

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

Розробка веб-застосування по замовленню таксі

**Текстова частина до курсової роботи
за спеціальністю „Програмна інженерія”**

Керівник курсової роботи
Кандидат фізико-
математичних наук,
Гречко А. В.

Виконала студентка
Шкаровська Н. С.

Київ – 2020

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Г. І. Малашонок
(підпис)

„_____” _____ 2019 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Шкаровська Н. С. факультету інформатики 3-го курсу

ТЕМА: Розробка веб-застосування по замовленню таксі

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Вступ

1. Аналіз предметної області
 2. Теоретичні відомості
 3. Опис реалізації програмного продукту
- Висновки

Перелік використаних джерел

Додатки

Дата видачі „_____” _____ 2019 р. Керівник _____

(підпис)

Завдання отримав _____ (підпис)

Календарний план виконання курсової роботи

Тема: Розробка веб-застосування по замовленню таксі

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	28.10.2019	
2.	Підбір матеріалів з HTML, CSS, JavaScript.	02.11.2019	
3.	Знайомство з середою розробки WebStorm та DataGrip.	08.11.2019	
4.	Знайомство з Node.js.	25.11.2019	
5.	Створення програм з використанням Node.js.	10.12.2019	
6.	Пошук літератури для вивчення програмування веб-застосувань клієнт-серверної архітектури.	23.12.2019	
7.	Створення методів обробки запитів за допомогою Express.js.	13.01.2020	
8.	Огляд аналогів розробки.	01.02.2020	
9.	Початок створення практичної частини.	06.03.2020	
10.	Початок написання текстової частини.	13.04.2017	
11.	Надання проміжної версії практичної частини.	22.04.2017	
12.	Аналіз практичної частини, її корегування.	26.05.2017	
13.	Остаточне завершення розробки веб-застосування.	02.05.2017	
14.	Остаточне завершення написання пояснювальної записки.	06.05.2017	

15.	Створення презентації.	07.05.2017	
16.	Захист курсової роботи.	11.05.2017	

Студент Шкаровська Н. С.

Керівник Гречко А. В.

“ _____ ” _____

Зміст

Зміст	5
Перелік умовних позначень.....	6
Анотація.....	7
Вступ	8
1 Аналіз предметної області	11
1.1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБҐРУНТУВАННЯ ТЕМИ	11
1.2 ОГЛЯД ІСНУЮЧИХ АНАЛОГІВ РОЗРОБКИ	13
1.3 ПОСТАНОВКА ЗАВДАННЯ	15
2 Теоретичні відомості	18
2.1 ПРЕДМЕТНА ОБЛАСТЬ	18
2.2 ОПИС ТЕХНІЧНИХ ВІДОМОСТЕЙ	19
3 Опис реалізації програмного продукту	23
3.1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	23
3.2 АРХІТЕКТУРА ВЕБ-ЗАСТОСУВАННЯ	26
3.3 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ	27
3.4 ОПИС РОЗРОБКИ ПРОГРАМИ	29
3.5 СТВОРЕННЯ ОБ'ЄКТІВ І РОЗРОБКА ГОЛОВНОЇ ПРОГРАМИ	30
3.6 ОПИС БАЗИ ДАНИХ ТА ІНТЕРФЕЙСУ ПРОГРАМИ	33
3.6.1. <i>Схема бази даних</i>	33
3.6.1. <i>Інтерфейс Програми</i>	34
3.7 ІНСТРУКЦІЯ КОРИСТУВАЧА	36
3.8 ПЕРСПЕКТИВИ РОЗРОБЛЕНОГО ВЕБ-ЗАСТОСУВАННЯ	39
Висновки.....	41
Перелік використаних джерел	43
Додатки	45
1 Додаток А – Додаткові ілюстрації	45
2 Додаток Б – Список ілюстрацій	53
3 Додаток В – Програмний код	54

Перелік умовних позначень

Taxiconn – власна назва веб-застосування по замовленню таксі.

Клієнт – користувач Taxiconn (або іншої системи по замовленню таксі), який користується послугами по замовленню таксі.

Водій – користувач Taxiconn (або іншої системи по замовленню таксі), який надає послуги по перевезенню на замовлення клієнтів.

Фрілансер – вільнонайманець, той хто сам шукає собі роботу і може одночасно працювати на декілька фірм.

Футер – структурний елемент, що розташований в нижній частині веб-сторінки (або веб-сторінок).

Контент – інформаційне наповнення сайту або сторінки.

Хмара – колекція цифрового контенту, що доступна з мережі Інтернет.

Localhost – локальний хост, зарезервоване доменне ім'я для приватних IP-адрес, недоступний з мережі Інтернет.

НаУКМА – Національний університет «Києво-Могилянська академія».

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер або інший персональний пристрій користувача.

SSR - server-side rendering (серверний рендеринг).

HTTP - Hyper Text Transfer Protocol (протокол передачі гіпертексту).

CORS - Cross-Origin Resource Sharing – технологія сучасних браузерів, що дозволяє надавати веб-сторінці доступ до ресурсів іншого домену.

AJAX - Asynchronous JavaScript And XML – підхід до побудови веб-сторінок, що не потребують перезавантаження сторінки при обробці запитів.

Селектор – html-тег, що дозволяє створити елемент інтерфейсу у вигляді випадального списку.

Анотація

У даній курсовій роботі передбачається розробка веб-застосування по замовленню таксі та створення автоматизованої системи, що не потребує участі диспетчера. Веб-застосування написане на Node.js з використанням додаткових ресурсів, зокрема фреймворків Bootstrap 4 та Express.js, а також шаблонізатора Pug, має клієнт-серверну архітектуру. До проекту також було підключено Google Maps та Google Autocomplete, для прокладання маршруту на картах. Система при замовленні таксі обраховує вартість керуючись власними алгоритмами та даними. Створений сайт містить увесь реалізований функціонал, вказаний у вимогах. У текстовій частині роботи міститься опис предметної області та алгоритми роботи таксопарків, теоретичні відомості про обрані засоби розробки, детальний опис процесу розробки веб-застосування, а також інструкцію користувача готовою системою.

Вступ

У стрімкому, сучасному світі актуальними залишаються питання швидких та комфортних переміщень. Більшість людей перебувають в постійних пошуках зручніших сервісів, що надають послуги пасажирських перевезень. Кожен хоче знаходити рішення власних потреб онлайн і з будь-якого пристрою, адже інтернет – це найзручніший спосіб доступу до інформації та менеджменту особистих та робочих справ. Рішенням таких проблем може стати веб-застосунок для замовлення таксі «Taxiconn», що представляє собою автоматизовану незалежну платформу з усім необхідним функціоналом, який дозволяє без участі диспетчерів надавати послуги для замовлення таксі по всій території України. Також даний сервіс зможе полегшити замовлення таксі для людей з вадами слуху і мовлення та інтровертів. Готовий веб-застосунок зможуть використовувати компанії, що надають послуги замовлення таксі, але не мають автоматизованої онлайн-системи.

Метою курсової роботи є створення автоматизованої системи по замовленню таксі, що надасть можливість зручно та швидко робити замовлення будь-де, в будь-який час та з будь-якого пристрою. Основним завданням проекту є дослідження системи роботи таксопарків для визначення найкращої стратегії розробки онлайн-застосування та безпосередньо реалізація готового веб-застосування, що містить повний набір функцій, необхідних для автоматизованого замовлення таксі клієнтами та виконання їхніх замовлень водіями.

Об'єктом дослідження є процес створення веб-сайту по замовленню таксі, що міг би автоматизувати роботу таксопарків та диспетчерських. Функціонал сайту базується на дослідженні форм роботи різних таксопарків (як онлайн так і поза мережею), з урахуванням їх переваг та недоліків.

Предметом дослідження стали актуальні алгоритми роботи таксопарків. Для розробки максимально ефективного та зручного веб-застосування, як для клієнтів, так і для водіїв, вивчалися алгоритми замовлення таксі для користувачів послуг, розподілу замовлень між водіями, збору даних про користувачів та

водіїв. Також було проаналізовано сучасні технології розробки веб-сайтів, що могли б забезпечити реалізацію обраних для розробки алгоритмів, адже система повинна коректно працювати у користувачів з різним програмним забезпеченням та підтримуватись різними браузерами різних версій. Основними методами дослідження є спостереження, аналіз сфери таксі для визначення основних проблем та пошук оптимальних рішень. Також використовувались методи прогнозування подальшого розвитку галузі.

Проведені дослідження сформували повну уяву про те, як повинен виглядати користувацький інтерфейс, який функціонал є найважливішим, у якому форматі його необхідно реалізовувати, які дані необхідно зберігати у базі даних, як розподілити виконання функцій між клієнтською та серверною частиною. Для коректної роботи системи та з метою уникнення проблем шахрайництвом у сфері таксі, рекомендовано, щоб у онлайн-системи був адміністратор, який би постійно відслідковував потік замовлень, незважаючи на те, що сайт містить певні правила та обмеження для мінімізації підозрілої діяльності. Реалізоване веб-застосування може використовуватись як незалежна автоматизована система замовлень таксі (без диспетчерських та таксопарків, де кожен водій працює як фрілансер) або програмне забезпечення для компаній, що вже надають послуги таксі для створення зручніших способів замовлень для клієнтів, управління поїздками для водіїв та збору статистики для менеджерів компаній.

Для створення сервісу «Taxiconn» використано сучасні засоби розробки, що дозволяють організувати зручну роботу з багатофайловими проектами. Клієнтську та серверну частини реалізовано за допомогою WebStorm 2020 – середовище розробки від JetBrains, що забезпечує роботу з HTML5, CSS та JavaScript, а також містить вбудовану систему керування версіями Git. Для управління базою даних використано MySQL Server 8.0 – вільну систему, що створює окремий локальний сервер для хостингу баз даних, та DataGrip 2020 – ПЗ від JetBrains для зручного управління базами даних. Також для відображення результатів роботи та перевірки повної працездатності веб-застосування

використовувались браузери Google Chrome та Opera (включаючи вбудовані інструменти розробника).

Робота, окрім переліку умовних позначень, вступу, висновку, переліку використаних джерел та додатків, містить розділи аналізу предметної області, теоретичних відомостей та опису реалізації веб-застосування. Перший розділ містить огляд сучасного стану питання та загальний опис технічного завдання, сформульованого в результаті аналізу систем аналогів. У другому розділі наведені шляхи пошуку оптимальних рішень щодо розробки веб-застосування, зокрема, визначення кращих підходів до розробки проекту. Третій розділ містить конкретизоване технічне завдання, описи розробки та інтерфейсу програми. Структура роботи повністю відповідає всім вимогам до курсових робіт.

1 Аналіз предметної області

1.1 Аналіз сучасного стану питання та обґрунтування теми

Сфера замовлення таксі достатньо розвинена в сучасному світі, зокрема в Україні. Громадський транспорт здебільшого не забезпечує всіх потреб різних верств населення, а власний транспортний засіб потребує не лише великих грошових витрат, але й повної участі в дорожньому русі людини, яка знаходиться за кермом. Безліч людей щодня користуються послугами таксі, замовляючи їх за телефоном (що не завжди зручно) або з мобільного застосування. Значна частина постійних користувачів послуг перевезень не вважає замовлення таксі за допомогою телефону зручним через такі фактори:

- проблеми з мобільним зв'язком: оператор може неправильно почути адресу посадки або час, на який замовлено таксі;
- диспетчери часто говорять дуже нерозбірливо, що ускладнює процес замовлення;
- деякі люди (зокрема інтроверти) відчують дискомфорт під час особистого спілкування, відповідно розмова з диспетчером при замовленнях також є неприйнятною для них;
- перевантажені лінії call-центрів служб замовлення таксі унеможлиблюють швидке прийняття замовлень, тому доводиться довго очікувати відповіді.

Також варто враховувати, що є група людей з вадами слуху та мовлення, які фізично не можуть замовити таксі за телефоном. В свою чергу, дзвінки водіїв про прибуття на місце посадки часто бувають недоречними або заважають. Більш зручний спосіб - замовлення таксі у спеціальних мобільних застосуваннях, але цей спосіб також має певні недоліки. Мобільні застосування потребують встановлення додаткового програмного забезпечення та постійно займають місце на пристрої, навіть коли не використовуються (більшість користувачів вважає це неприйнятним через брак місця на телефоні). Інший недолік мобільних застосувань полягає у «прив'язуванні» акаунту користувача до конкретного

пристрою. Окрім того, адміністратори великих компаній, які переважно замовляють таксі для своїх співробітників у корпоративних цілях, потребують можливості зручно робити замовлення з комп'ютера.

Алгоритм роботи звичайної служби замовлення таксі з диспетчером-людиною можна коротко описати так:

- a) людина, що хоче скористатись послугами сервісу таксі, телефонує або залишає заявку на дзвінок;
- b) один з диспетчерів відповідає на дзвінок клієнта, отримуючи (або уточнюючи) в розмові адресу посадки та висадки; іноді одразу називає орієнтовану вартість поїздки;
- c) сформоване замовлення на перевезення розсилається усім водіям;
- d) вільний водій приймає замовлення, про що сповіщає систему (диспетчера), яка у свою чергу надсилає sms-повідомлення людині-замовнику;
- e) водій вирушає за адресою, вказаною у замовленні, та виконує його, після чого повідомляє про це диспетчера.

Веб-застосування замовлення таксі, що розробляється, не повинне керуватись таким алгоритмом роботи, адже метою розробки є саме автоматизована система. Тому для даного проекту краще підійде алгоритм роботи служб замовлення таксі, що працюють без диспетчерів, а саме:

- a) людина, що хоче скористатись послугами сервісу таксі, залишає заявку із вказаними адресами висадки та посадки (при цьому система автоматично і одразу повідомляє орієнтовну вартість замовлення);
- b) водії відслідковують спеціальну сторінку, де відображаються замовлені поїздки, обирають найкраще на їх погляд замовлення та приймають його (при цьому користувач бачить зміну статусу замовлення);
- c) водій виконує замовлення, а потім позначає поїздку як завершену на сторінці замовлень (у системі зберігаються наведені замовлення, тому згодом оплата за надані послуги водію буде нарахована автоматично).

Варто зазначити, що при розробці даного веб-застосування (станом на період з березня по травень 2020-го року), актуальність системи, що розробляється значно виросла через світову пандемію COVID-19, адже громадський транспорт на території України було заборонено для загального користування, тому попит на таксі у цілому значно підвищився, адже інших шляхів пересування містом не існує, якщо немає особистого транспорту.

Отже, виникає потреба в універсальному веб-застосуванні для замовлення таксі, що не вимагатиме особистого спілкування з диспетчером, не потребуватиме завантажування додаткового програмного забезпечення на телефон та буде зручним для користування з будь-якого пристрою. При цьому на різних пристроях повинен бути незалежний доступ до єдиного акаунту користувача.

1.2 Огляд існуючих аналогів розробки

Сфера замовлень таксі онлайн надзвичайно велика, зокрема, в Україні є декілька великих конкуруючих компаній, що пропонують послуги онлайн-замовлення таксі, серед них:

- Uber – один з перших сервісів, що популяризував замовлення таксі онлайн не лише в Україні. Компанія пропонує замовлення таксі телефоном, у мобільному додатку або через сайт. Зокрема, замовити послугу через сайт достатньо важко, адже сторінка замовлення погано відображується пошуковими системами [1], а іноді після



Рисунок 1 – Логотип Uber

замовлення через сайт користувачам можуть телефонувати для підтвердження замовлення. Замовлення через мобільний додаток дуже зручне та містить багато додаткових опцій, декілька способів оплати та можливість зв'язатись з водієм через додаток за необхідності.

- Uklon – перший український сервіс для замовлення таксі онлайн [2]. В Uklon повністю відсутня можливість замовити таксі через телефон, зробити замовлення можна виключно в мобільному додатку.



Рисунок 2 - Логотип Uklon

Додаток має зручний інтерфейс, декілька можливих способів оплати, а також можливість змінювати ціну на поїздку. Проте, з особистого досвіду користування можна відмітити проблеми з визначенням адреси при замовленні поїздки, яка полягає в тому, що пошуковик додатку не завжди знаходить потрібну вулицю за частиною назви.

- Bolt – міжнародна компанія, що надає послуги замовлення таксі виключно в додатку, набула популярності в Україні останньою з вищезгаданих [3]. Компанія має найбільший вибір типів авто, які можна замовити, але порівняно незрозумілий інтерфейс з іншими мобільними додатками, натомість пропонує багато бонусів для своїх користувачів.

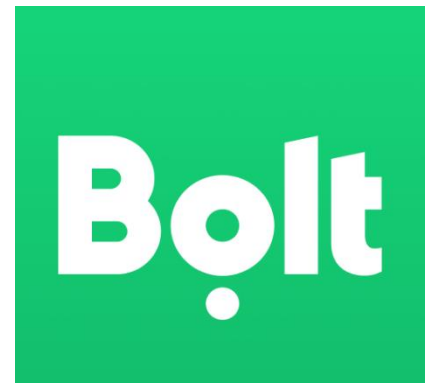


Рисунок 3 - Логотип Bolt

Існують також інші достатньо популярні сервіси, що пропонують послуги замовлення таксі за телефоном та онлайн. За останні роки більшість приватних компаній на ринку пасажирських перевезень намагаються створити та популяризувати свої способи замовлення таксі онлайн або в мобільних застосуваннях. Вдалим прикладом є служба замовлення таксі 838 [4], яка лише у 2019 році запустила повноцінний мобільний додаток для замовлення таксі. Хоча ця компанія на ринку досить давно, проте її популярність залишається порівняно низькою з іншими сервісами. Також існує безліч інших непопулярних онлайн-сервісів, які дозволяють «замовити» таксі онлайн. Але насправді після залишеної заявки телефонує диспетчер для уточнення адреси посадки та на вимогу називає

приблизну вартість поїздки телефоном, що знову ж таки є незручним для більшості користувачів.

Необхідно ще раз звернути увагу на те, що веб-сервіс буде розроблений таким чином, щоб зробити систему повністю автоматизованою, забезпечити замовлення таксі та реалізацію замовлень без участі диспетчера (лише клієнт та водій). Тому можна вважати, що зараз для більшості компаній в галузі замовлень таксі, аналогових до розробки, що стала об'єктом дослідження, реальним конкурентом є лише сервіс замовлень таксі онлайн від Uber. Проте, варто враховувати, що конкурентними є також і мобільні додатки, адже є багато прихильників таких способів замовлення таксі. Тому оцінка кількості конкурентів може варіюватися в залежності від погляду на проблему: а саме, чи враховувати служби, що пропонують замовлення таксі в мобільних додатках, як конкурентів?

Отже, наразі на ринку надання послуг пасажирських перевезень існує достатньо аналогів розробки, проте більшість з них не задовольняють повний перелік вимог користувачів до таких систем, особливо онлайн-сервіси, що лише імітують автоматизоване замовлення таксі. Частина користувачів цілком задоволена мобільними додатками, проте також існує пласт невдоволених користувачів. Таким чином, це ще раз доводить доцільність розробки такого веб-застосування.

1.3 Постановка завдання

У результаті ретельного аналізу предметної області та детального огляду конкурентних аналогів розробки сформовано повне уявлення про завдання, які необхідно виконати, та вимоги до системи, що розробляється.

Необхідно створити веб-застосування (сайт), що буде простим та зручним для всіх користувачів. При цьому повинен забезпечуватись доступ зареєстрованих користувачів до єдиного акаунту з будь-якого пристрою, в якому зберігатимуться персональні дані, а також дані про попередні замовлення та інформацію про їх виконання. Системою передбачається два типи користувачів:

клієнт та водій, що будуть виконувати різні задачі в межах сервісу. Також веб-застосування повинно реалізовувати основні завдання проекту, а саме: надання можливостей клієнтам зручно, швидко, а головне без участі диспетчера замовляти таксі, а водіям приймати замовлення без участі диспетчера, а згодом завершувати їх. Варто зазначити, що веб-сайт повинен бути доступний для незареєстрованих та неавторизованих користувачів, проте в цьому випадку повинні відображатись лише сторінки з загальнодоступною інформацією. Розроблене веб-застосування повинно повністю відповідати всім наведеним нижче вимогам та мати здатність реалізувати всі необхідні можливості.

Система повинна надавати наступні можливості:

- Реєстрація для клієнтів та водіїв з підтвердженням електронної пошти. Реєстрація водіїв, окрім персональних даних, повинна містити інформацію про транспортний засіб.
- Збереження та редагування персональної інформації про клієнтів та водіїв в особистому кабінеті.
- Формування замовлення на послуги перевезення для клієнтів «в один клік».
- Зручне відображення замовлених поїздок та можливість взяти замовлену поїздку для водіїв.
- Збереження даних про завершені та/або скасовані поїздки для можливості їх перегляду як для клієнтів, так і для водіїв.
- Перегляд інформації про компанію та контактні дані для всіх відвідувачів веб-сайту.
- Перегляд користувачами профілю водія (з фотографією, якщо вона завантажена), який прийняв замовлену поїздку.
- Перегляд водіями профілю клієнтів, що замовляють послуги таксі (включаючи фотографії клієнтів, якщо вони завантажені).

Користувацький інтерфейс веб-застосування повинен бути простий, лаконічний та інтуїтивно-зрозумілий. Сайт повинен мати зручну навігацію. Для кращого сприйняття сайт повинен мати виважений стиль в єдиній кольоровій

гамі. Реалізоване веб-застосування не повинне вимагати постійної підтримки та переписування коду для забезпечення його функціонування.

Зважаючи на вимоги до системи та проведений аналіз аналогових сервісів, варто зазначити, що веб-застосування також повинно містити наступні інформаційні сторінки:

- «Про нас» - короткий опис сайту з метою привернути увагу користувачів.
- «Контакти» - дані про компанію або власника, що надають послуги замовлення таксі з контактною інформацією для реалізації підтримки користувачів.
- Сторінка правил та рекомендацій з надання послуг у сфері таксі для водіїв (або відвідувачів веб-сайту, що хочуть стати водіями), із зазначенням переліку дозволених виробників та марок транспортних засобів.
- Сторінку правил користування сервісом замовлення таксі для клієнтів та всіх відвідувачів сайту.
- Окремі сторінки для реєстрації клієнтів та водіїв, а також зручне вікно авторизації (входу на сайт).
- Для авторизованих користувачів повинна бути доступна окрема сторінка «Профіль», що відображає персональну інформацію.

2 Теоретичні відомості

2.1 Предметна область

Сервіси, що надають послуги перевезень на приватному транспорті за домовленостями існували ще в Древньому Римі. В той же час виник термін «таксометр». Оплата за домовленостями зазвичай передбачала передачу певної товарної або грошової валюти. Більш звичного для сучасності вигляду таксі почало набувати в XVII столітті в Парижі [5], також відомо про англійські прототиби того ж періоду.

Як відомо, таксі переважно асоціюється з жовтим кольором. Вперше колір таксі обрали в Англії, транспортний засіб пофарбували у жовтий колір, адже компанія, що надавала послуги перевезень не хотіла асоціюватись з політичними партіями, а жовтий був єдиним «вільним» кольором [6]. За іншою версією, вперше таксі стали жовтими в Америці, адже їх фарбували в два найпопулярніші кольори: червоний та зелений, які при змішування утворили жовтий з специфічним оранжевим відтінком.

Таксі – це тип громадського транспорту, що використовується для перевезення пасажирів та грузів у будь-яке вказане місце з оплатою проїзду за таксометром [7], приладом, що обраховує вартість поїздки на основі певних формул та вимірюваних параметрів. Таксометр встановлюється в автомобілях, що надають послуги перевезень.

Обрахунок вартості поїздки здійснюється на основі тривалості або відстані, на яку було здійснено перевезення. При цьому кожна компанія встановлює власні значення вартості послуг. Окремі сервіси проводять обрахунки за сталими формулами: $\text{Ціна} = \text{кількість пройдених кілометрів} * \text{ціну за кілометр}$ (в якості вимірюваного параметру можуть виступати метри або хвилини). Більш розвинені служби таксі також враховують: поточну завантаженість їх водіїв, тип автомобіля, який замовили (економ, комфорт, бізнес, вантажні перевезення, тощо.), тип оплати (якщо доступна оплата не лише

готівкою), поточний час доби, тощо. Наведені показники впливають на ціну за кілометр або хвилину. Також багато служб таксі мають сталу мінімальну вартість поїздки на випадок замовлення таксі на короткі відстані, деякі компанії також додають вартість мінімальної поїздки до загальної:

$$\text{Ціна} = \text{число пройдених кілометрів} * \text{ціну за кілометр} + \text{мінімальна вартість.}$$

Всі послуги, що надає служба громадських перевезень, повинні відповідати вимогам чинного законодавства України [8]. Багато компаній також висувають власні вимоги до водіїв та їхніх транспортних засобів, а також встановлюють додаткові правила перевезень, проте внутрішні правила компанії не можуть суперечити законодавству.

2.2 Опис технічних відомостей

Розуміння роботи програми прямо пропорційно залежить від розуміння всіх її компонентів та взаємозв'язків. Веб-застосування по замовленню таксі має чітку клієнт-серверну архітектуру, що забезпечує логічне розподілення функцій між частинами програми: клієнтською, серверною та рівнем даних. Також невід'ємною складовою такої архітектури вважають мережу через яку комунікують клієнтська та серверна частини. Через дану мережу вони також обмінюються необхідними даними, при цьому лише сервер має доступ до рівня даних. Зазвичай мережею виступає всесвітня мережа Інтернет або локальна, недоступна ззовні, мережа. Клієнтський ПК надсилає запити на сервер (запит зазвичай включає певний персональний ідентифікатор користувача) для отримання контенту сайту, сервер обробляє запит, отримує з бази даних необхідні дані та надсилає сформовану відповідь [9].

Клієнт-серверна архітектура забезпечує:

- зменшення навантаження на клієнтський ПК при обробці даних;
- кращу безпеку при зберіганні даних, до яких користувачі не мають безпосереднього доступу;

- відсутність дублювання коду серверу на клієнтських програмах;
- реалізацію розділення ролей між користувачами.

У клієнт-серверній архітектурі існують дві основні моделі взаємодії: тонкий клієнт та товстий клієнт. При використанні моделі товстого клієнта основну обробку даних забезпечує користувацький ПК, а сервер використовується лише як спосіб доступу до даних. Модель тонкий клієнт – товстий сервер забезпечує повну реалізацію логіки застосування та управління даними на серверній частині, при цьому значно зменшується навантаження на ПК користувачів. Клієнтська частина використовується виключно як рівень представлення даних.

Даний проект втілює модель тонкого клієнта, для повної реалізації якого використовується серверний рендеринг. Серверний рендеринг (server-side rendering) - це генерування клієнтської частини на сервері в універсальний html-код [10]. В даному проекті сервер генерує сторінки на основі файлів шаблонізатора Pug, який дозволяє інтегрувати дані з зовнішніх ресурсів в html-код, створює цикли, що в свою чергу дозволяють створювати набори однотипних об'єктів на сторінках. Також в шаблонізаторі реалізована функція вторинного використання коду, що дозволяє компонувати різні сторінки з однакових об'єктів. Приклад використання шаблонізатора наведено на схемі 1.

Компоненти файлу all_orders.pug

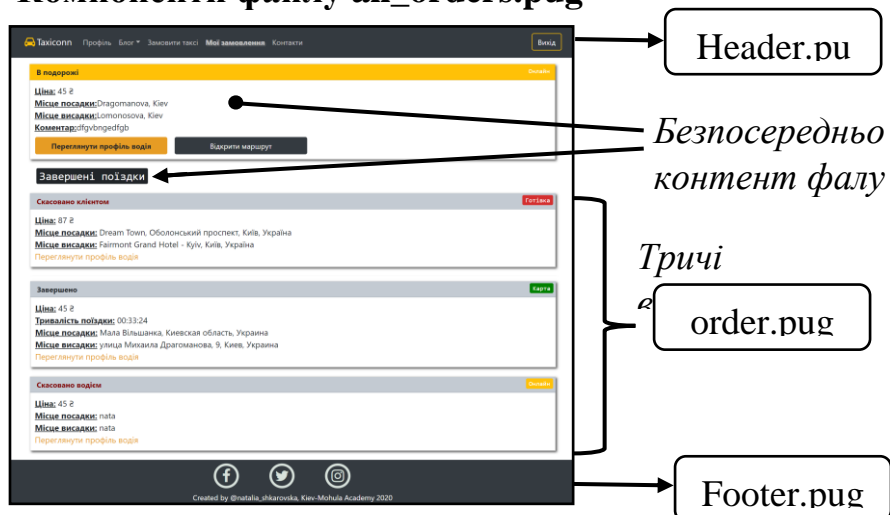


Схема 1 – Використання шаблонізатора

Користувачський ПК та сервер «спілкуються» між собою за допомогою протоколу передачі гіпертексту HTTP (в більш захищених мережах https). HTTP реалізує обмін через запити (request) та відповіді (response) [11]. Клієнтом зазвичай виступає браузер користувача, а сервером – обчислювальна машина з програмним кодом у хмарі або на локальному комп’ютері. Комунікація між частинами відбувається за алгоритмом, наведеним на схемі 2.



Схема 2 – Принцип роботи http

При першій спробі зайти на сайт браузер надсилає запит на html-сторінку, після цього надсилаються http-запити на файли стилів (.css), функцій (.js) та зображень або інших фалів, підключених до веб-сторінки. Якщо веб-сторінка повинна відображати не статичні дані, надсилаються додаткові запити на отримання необхідних даних. Переважно дані передаються в форматах json або xml.

Клієнт – це ПЗ на пристроях користувачів, що реалізує відображення інформації з зовнішніх ресурсів. В клієнт-серверній архітектурі клієнтами зазвичай виступають браузери або додатки, що надсилають запити за сервер відповідно до доменного імені та відображають отримані дані, зазвичай у вигляді html-сторінок.

Сервер – це ПЗ, що приймає запити клієнтів (браузерів або інших програм), виконує цифрову обробку інформації та має доступ до певних ресурсів (баз даних). Для реалізації досконалого серверу необхідний фреймворк, що має налаштовану систему роботи з даними різних типів, файлами cookies, заголовками CORS, тощо. Node.js – це програмна платформ для реалізації високопродуктивних веб-застосувань, написана мовою JavaScript [12]. Однією з

переваг Node.js є наявність найбільшої відкритої пакетної екосистеми (npm), тобто кожен вільно може використовувати будь-які бібліотеки та фреймворки, що допомагають оптимально реалізовувати необхідний функціонал. Зокрема, express – найпопулярніший фреймворк для веб-застосувань [13]. Express об'єднав у собі декілька бібліотек, утворивши мінімалістичну, гнучку та швидку систему для розробки http-серверів. Наразі актуальною версією є Express.js 4.17.1. Для розробки серверу за допомогою Node.js та Express.js їх необхідно встановити на локальному комп'ютері, підключивши менеджер пакетів npm. Розробивши систему маршрутизації (логічне визначення як застосування повинно відповідати на запит за конкретною адресою), можна створювати методи для обробки запитів за відповідними адресами.

3 Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

Враховуючи визначені в розділі 1.3 завдання з розробки, оперуючи технічними знаннями у сфері замовлення таксі та розробки веб-застосунків, можна сформулювати більш чіткі вимоги до програми та даних.

У базі даних необхідно в структурованому вигляді зберігати таку інформацію:

- Персональні дані клієнта: прізвище, ім'я, вік, електронна пошта, телефон (за бажанням), фото, примітки.
- Персональні дані водія: прізвище, ім'я, вік, електронна пошта, телефон (за бажанням), номер водійського посвідчення, номер авто, назву виробника та марку транспортного засобу, фото, примітки.
- Інформацію щодо замовлення: дані клієнта, що замовив, дані водія, що прийняв поїздку, адресу посадки, адресу висадки, тип оплати, вартість, поточний статус поїздки, дата та час подачі заявки, а також дата та час завершення поїздки (якщо поїздка набула статусу «Завершено», а не «Скасовано»).
- Перелік допустимих виробників та марок транспортних засобів у конкретній службі по замовленню таксі.

До інформації, яку необхідно зберігати, висуваються наступні корпоративні обмеження цілісності:

- Вік водія повинен бути в межах від 21-го до 65-років.
- Вік клієнта повинен перевищувати 16 років.
- Електронна пошта, номер телефону та логін у клієнтів та водіїв повинен бути унікальним. При цьому допускається, що однакові дані можуть бути у різних типів користувачів.
- Номери авто та ліцензії у водія повинні бути унікальними.
- Одночасно водій може мати лише одне замовлення зі статусом «В подорожі».

- Клієнт в один проміжок часу може мати не більше двох замовлень або поїздок зі статусом «В подорожі» (на випадок замовлень таксі для компанії або для рідних чи друзів).
- Адреси посадки та висадки повинні бути в межах України (для клієнтів).
- Способи оплати та тип поїздки (наприклад «economy» або «comfort») повинні бути обрані зі списку.
- Виробник та марка транспортного засобу водія повинна належати до переліку дозволених для використання автомобілів в конкретній службі таксі.
- Доменне ім'я для електронної пошти повинне обиратись зі списку.
- Номер телефону повинен відповідати шаблону номерів телефонів в межах України.

Для коректної реалізації всіх необхідних можливостей системи, необхідно, щоб система мала наступні функції:

- Швидка реєстрація для користувачів з надсиланням тимчасового паролю на електронну пошту (таким чином реалізується вимога до підтвердження електронної пошти), при першому вході пароль потрібно змінювати на обраний користувачем.
- Реєстрація водіїв із зазначенням повного переліку даних, обов'язкових для даного типу користувачів зі строгою валідацією цих даних (включаючи дані про автомобіль). Для завершення реєстрації водію повинен надсилатись код підтвердження.
- Перегляд особистої інформації в персональному кабінеті користувача (профілі) зі статистикою про попередні поїздки (завершені та/або скасовані).
- Редагування особистої інформації відповідно до корпоративних обмежень цілісності в профілі (в тому числі зміна пароля).
- Замовлення таксі для клієнтів з автозаповненням адрес та попереднім переглядом маршруту поїздки на карті.

- Прорахунок попередньої вартості замовлення для клієнтів.
- Перегляд замовлених поїздок користувачами з додатковими можливостями доступу до перегляду профіля користувача, маршруту.
- «Прийняти замовлення» - виконати замовлення користувача з відображенням відповідної інформації на сторінці замовлень як у водія, що прийняв замовлення, так і у клієнта, що подав заявку.
- «Скасувати замовлення» - клієнти можуть скасувати подане замовлення, поки воно не було прийняте водієм. Водії можуть скасовувати прийняті замовлення.
- «Завершити замовлення» - водії повинні змінювати статус замовлення на «Завершено» після його виконання.

Архітектура даної системи побудована за багаторівневим шаблоном з трьома рівнями: клієнтська частина, серверна частина та база даних. На клієнтській частині повинна бути реалізована навігація сторінками сайту, відображення даних на запити та виконання певних функцій через різноманітні форми та кнопки. Серверна частина повинна реалізовувати відповідні запити користувача та, за умови проходження перевірки рівня доступу до даних, надсилати дані користувачу. Також серверна частина повинна реалізовувати роботу з базою даних. У базі даних повинна зберігатись необхідна інформація.

Для комфортної роботи користувачів на сайті в звичному форматі необхідно реалізувати навігаційне меню в верхній частині екрану. Навігаційне меню повинне містити посилання на всі доступні для користувача сторінки та постійно відображатись, незалежно від контенту. Також сайт повинен містити звичну для сфери таксі кольорову гаму: відтінки жовтого (#ffc107), відтінки сірого (#343m40) та білий. Для додаткового виділення компонентів допускається використання зеленого (#008000) та червоного (#d42f2f) кольорів. Внизу кожної сторінки повинен міститись футер з інформацією про розробника.

Програма повинна працювати в окремому вікні браузера. Сайт повинен коректно відображатись на всіх типах пристроїв, де можна встановити браузер з розміром вікна від 450 пікселів, а також повинен підтримуватись переважною

більшістю сучасних популярних браузерів, включаючи їх старіші версії. Окрім того, програма повинна відповідати певним вимогам надійності.

Вимоги до надійності:

- Права доступу до інформації бази даних повинні розподілятися відповідно до їх ролей та авторизованості користувачів сайту (кожен користувач повинен мати змогу переглядати лише власну історію поїздок).
- Доступ до замовлених поїздок повинен бути виключно у водіїв. Після того як водій прийняв замовлення, воно повинно відображатись лише цьому водієві.
- При виявленні неправильних або некоректних дій користувачів, система повинна повідомляти клієнтів або водіїв про це.
- Замовляти таксі може лише зареєстрований клієнт з підтвердженою електронною поштою.
- Водій може мати доступ до системи лише після повної, коректної реєстрації та підтвердженої електронної пошти.
- При спробі переходу на неіснуючі сторінки або сторінки, що потребують іншого рівня авторизації, виконувати переадресацію на сторінку помилки (іншими словами 404).

3.2 Архітектура веб-застосування

Для реалізації веб-застосування було обрано стандартну клієнт-серверну архітектуру, що має чітко розподілені три рівні: рівень представлення даних (інтерфейс користувача), серверна частина, що забезпечує логічну обробку подій та база даних, як рівень управління даними. Таку архітектуру було обрано зважаючи не лише на вимоги до швидкодії програми на пристроях користувачів, але й враховуючи забезпечення розподілення рівнів доступу до інформації. Також варто зазначити, що була обрана клієнт-серверна архітектура з товстим сервером та тонким клієнтом. Таке розподілення логіки обробки інформації обумовлене вимогою, що сайт повинен підтримуватись будь-якими браузерами

та з будь-якого пристрою, тому в такий спосіб мінімізується використання ресурсів пристроїв користувачів. Окрім того, така архітектура забезпечує віддалене зберігання даних, незалежно від сесії користувача або роботи сервера. Схематично архітектура програми зображена на рисунку 4.

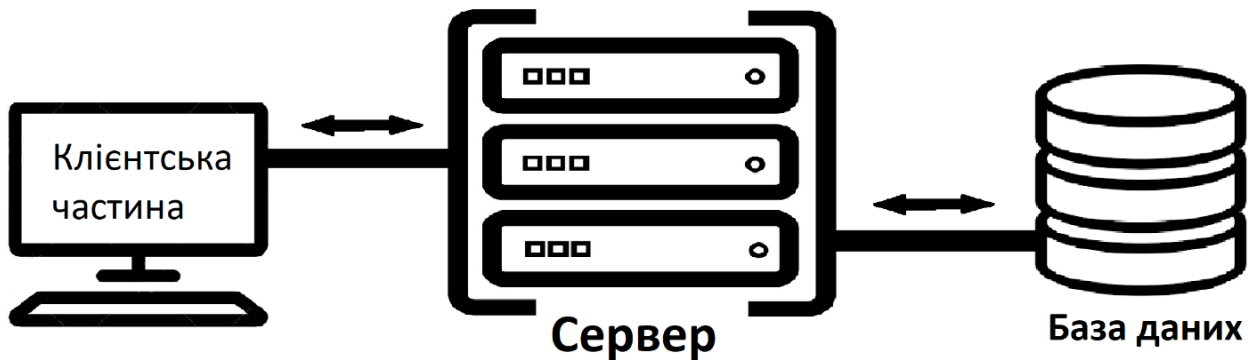


Рисунок 4 - Архітектура веб-застосування

3.3 Обґрунтування вибору засобів розробки

Оскільки розроблене веб-застосування має клієнт-серверну архітектуру, для реалізації різних рівнів використовувались різні мови програмування. Основна мова розробки - JavaScript, що використовувалась і на клієнтській і на серверній частинах. JavaScript — це динамічна, об'єктно-орієнтована мова програмування, яка підтримується абсолютною більшістю браузерів, що необхідно для розробки даного веб-застосування. Також мова підтримує технологію AJAX, що реалізує динамічне завантаження інформації, в тому числі використовується для реалізації запитів та отримання відповідей з серверної частини.

Основна обробка подій та запитів відбувається на сервері, написаному на Node.js, який в свою чергу реалізований на JavaScript. Основний функціонал сервера реалізований за допомогою наступних бібліотек та фреймворків:

- Express – мінімалістичний фреймворк, за допомогою якого створюється власне API для обробки http-запитів.

- Mysql – драйвер, що забезпечує підключення до бази даних та подальшу обробку sql-запитів.
- Pug – шаблонізатор, що динамічно створює html-сторінки на сервері, відповідно до вказаних параметрів.
- body-parser – бібліотека, що дозволяє отримувати із запитів дані у форматі json.
- cookie-parser – бібліотека, що забезпечує зручний доступ до даних cookies в заголовках запитів, що надходять на сервер.
- Multer – бібліотека, що реалізує завантаження файлів на сервер. Використовується для завантаження фотографій у профілях користувачів.

Клієнтська частина відображується за допомогою мов HTML5 (мова розмітки) та JavaScript. Для візуальних ефектів використовується мова стилю сторінок CSS. На клієнтській частині також використовуються:

- Bootstrap 4 – фреймворк, що містить набір готових компонентів для створення кращих візуальних ефектів.
- jQuery – відкрита бібліотека JavaScript для реалізації AJAX-запитів та спрощення написання програмного коду.

Для розробки клієнтської та серверної частин було обрано інтегроване середовище розробки від компанії JetBrains – WebStorm 2020. Це ПЗ дозволяє зручно встановлювати додаткові бібліотеки та фреймворки, управляти багатофайловими проектами, створювати та запускати написаний сервер локально. Також WebStorm містить вбудований контроль версій Git, що активно використовувався при розробці веб-застосування.

Як рівень даних було обрано mySQL базу даних, адже це одна з найдоступніших та відкритих баз даних, з якими можна працювати як на localhost, так і в хмарі. Для управління базою даних використовувались MySQL-сервер 8.0, що дозволяє створити окремий локальний сервер для баз даних, та DataGrip 2020 – програмне забезпечення для зручного керування базами даних.

3.4 Опис розробки програми

Веб-застосування являє собою набір файлів, що забезпечують роботу сервера та файли, що використовуються на рівні представлення даних. Умовно розробку програми можна поділити на такі етапи:

- Створення бази даних з таблицями відповідно до технічного завдання.
- Створення серверу на Node.js, підключення необхідних бібліотек та фреймворків.
- Написання методів, що реалізують відповіді на основні запити.
- Створення файлів-модулів та їх методів, підключення даних файлів до основного серверу.
- Написання файлів розміток сторінок та підключенням до них файлів функцій (в даному випадку файлів з розширенням .pug або .html).
- Створення функцій на клієнтській частині, що забезпечують навігацію сайтом, реалізують звернення до сервера та обробку відповідей.
- Створення стилю веб-застосування: підключення файлів .css та фреймворку Bootstrap 4.
- Тестування готової програми та її відлагодження.

Варто зазначити, що написання методів відповідей на сервері розбивається на модулі відповідно до груп методів на які вони відповідають. Зокрема, окремі модулі створено для обробки: запитів клієнтів (/user/...), запитів водіїв (/driver/...), запитів стосовно замовлень незалежно від типу користувача (/orders/...), а також окремий модуль, що відповідає за встановлення з'єднання з базою даних та використовується усіма іншими модулями. Також виокремлюється модуль функцій, які використовуються всередині методів та реалізують незначну частину обробки даних. Функції винесені в окремий файл для кращої декомпозиції коду та полегшення роботи з ним.

3.5 Створення об'єктів і розробка головної програми

Файлова структура проекту наведена на рисунку 5 [14]. У корені проекту зберігаються лише файли конфігурацій, логотип веб-застосування (favicon.ico) та основний файл (server.js), що запускає програму та забезпечує її працездатність. Директорія «config» містить файли конфігурації з даними про базу даних (включно з логіном та паролем), порт на якому за замовчуванням запускається веб-застосування, а також доменне ім'я. Директорія «node_modules» - це стандартний каталог, де зберігаються встановлені бібліотеки та з якої вони в подальшому імпортуються. Каталог «files» включає директорії з файлами шрифтів та зображень, що використовуються на сайті. Каталог «uploads» використовується для проміжного збереження зображень, які завантажують користувачі. Після завантаження в базу даних, непотрібні зображення видаляються.

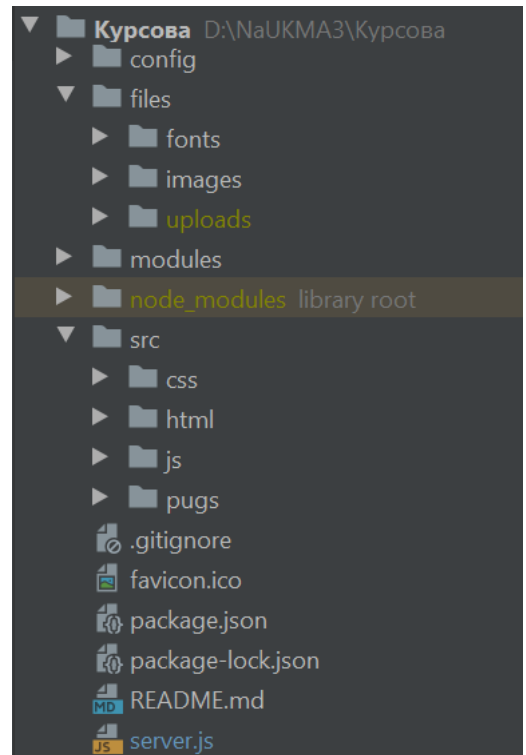


Рисунок 5 – Файлова структура проекту

Директорія «modules» містить файли модулів, що використовує сервер для обробки запитів (описані в розділі 3.5), всі файли мають розширення .js. Всі дані з директорії напряду імпортуються в основному файлі server.js. Також директорія містить вкладений каталог «scratch», яка використовується вбудованими бібліотеками localStorage та sessionStorage для зберігання даних (після використання дані видаляються). Каталог «src» містить всі ресурси, що забезпечують роботу клієнтської частини, включаючи: файли шаблонізатора Pug, які перетворюються сервером в звичний для браузерів html-код, файли з розширенням .html для сторінок, що не потребують передачі даних з сервера (файли зберігаються одразу готовими до відображення, що пришвидшує роботу сервера), а також директорії з файлами .css та .js, що імпортуються файлами

розмітки. Всі файли стилів та функцій в проекті підключаються у мінімізованому варіанті для пришвидшення їх імпортування (тобто використовуються файли *.min.js та *.min.css).

Логіку роботи програми можна описати достатньо просто. Спочатку користувач переходить на сайт за доменним ім'ям. В цей час сервер повертає згенеровану сторінку, яка відображається користувачу. Якщо користувач хоче перейти на іншу сторінку, на сервер надходить відповідний запит. Якщо така сторінка існує та роль користувача дозволяє її переглянути, сторінка генерується та відображується, інакше користувач бачить сторінку помилки. Якщо користувач хоче виконати певну дію (більшість дій супроводжуються натисканням на кнопку), викликається функція з клієнтської частини, що відповідає за дану дію. Деякі функції виконуються без запитів на сервер, інші потребують відповіді сервера. В обох випадках користувач бачить відображення результату його дій.

Механізм виконання запитів сервером можна описати декількома алгоритмами, відповідно до їх класифікації: запит без надісланих на сервер даних (рисунок 6) та з надісланими для обробки даними (рисунок 7).

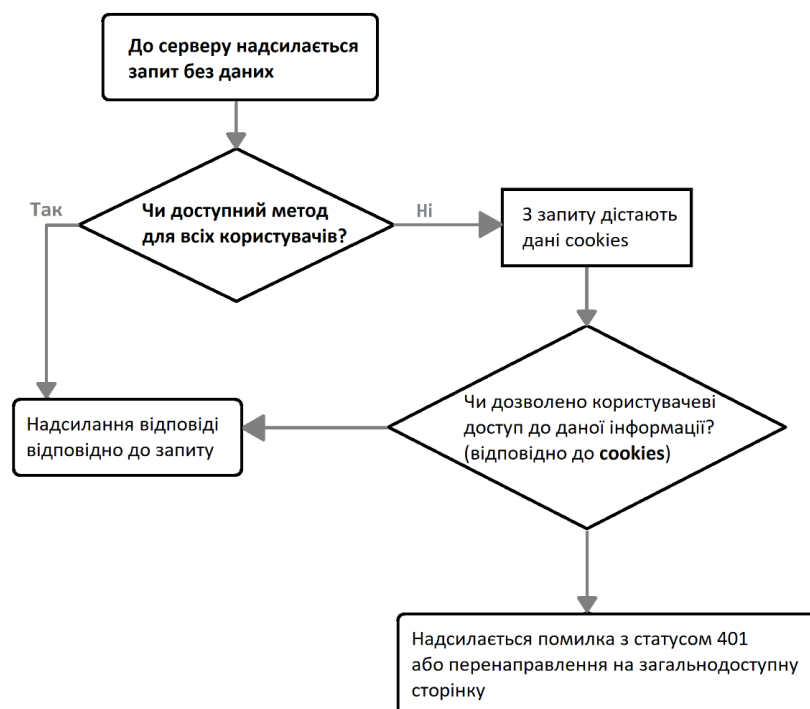


Рисунок 6 – Алгоритм роботи методу сервера без даних



Рисунок 7 – Алгоритм роботи запиту з даними

Варто зазначити, що наявні в програмі типи запитів “GET”, “POST” та “PUT” мають однакову логіку виконання, за винятком того, що методи типу “GET” спрямовані на надсилання даних та згенерованих сторінок без додаткової обробки даних. В той час як методи типів “POST” та “PUT” приймають дані, обробляють їх та надсилають відповідь про оброблену інформацію, якщо необхідно, або відповідний статус, що повідомляє клієнтську частину про успішне виконання запиту. Проте, запит будь-якого типу може приймати і надсилати у відповідь інформацію.

3.6 Опис бази даних та інтерфейсу програми

3.6.1. Схема бази даних

Відповідно до технічного завдання база даних містить всі необхідні параметри та має схему даних, представлену на рисунку 8. Всі дані, що зберігаються в базі даних, відповідають корпоративним обмеженням цілісності та проходять валідацію на клієнтській або серверній частині. Всі реляції знаходяться в четвертій нормальній формі.

Реляція «blogs», яка не має зв'язків із іншими таблицями, містить рекомендації з використання сервісу таксі для користувачів. Дані зберігаються в базі даних, адже ця інформація може неодноразово змінюватись. На сайті дані відображуються у вигляді коротких блогів, при цьому відображається заголовок, текст та дата.

Таблиця «drivers» зберігає персональну інформацію про водіїв таку як: прізвище та ім'я, вік, логін, електронну пошту та номер телефону, опис, номер водійського посвідчення (licence), фото, а також інформацію про авто: номер автомобіля, рік транспортного засобу та посилання на модель і виробника, адже водій не може використовувати транспортний засіб марка та виробник якого не підпадають під список дозволених. Вищевказані дані використовуються для відображення в профілі водія.

Таблиці «car_models» містить дані про дозвалені марки автомобілів та посилання на виробника. В свою чергу в реляції та «car_producer» міститься список виробників транспортних засобів, використання яких дозволене в межах сервісу. Саме з даних списків виробників та марок водії обирають виробника та

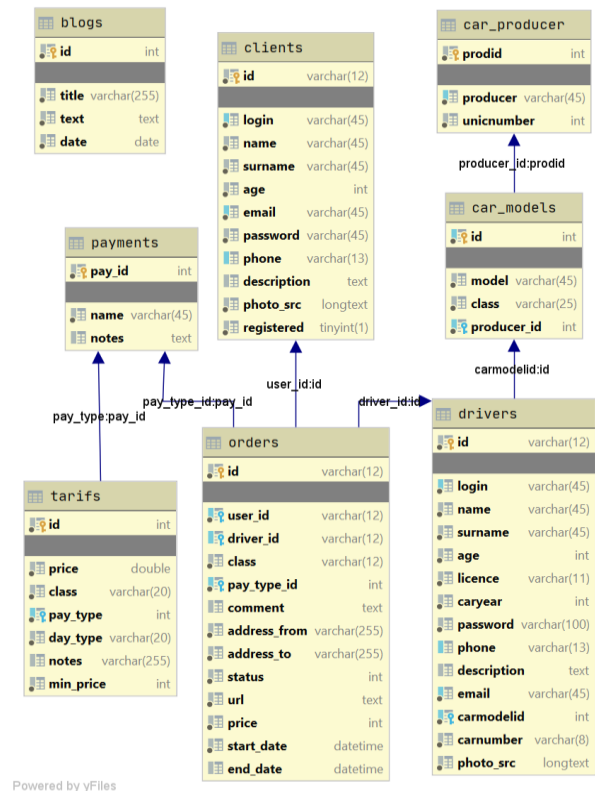


Рисунок 8 – Схема бази даних

марку транспортного засобу при реєстрації, тому у водія зберігається лише посилання на обрану марку.

У реляції «clients» зберігаються персональні дані користувачів, такі як: прізвище та ім'я, вік, опис, фото, логін, електронна пошта та номер телефону, а також чи здійснив користувач перший вхід в систему та змінив пароль (поле registered – використовується як підтвердження електронної пошти).

Варто зазначити, що в полях паролів водіїв та клієнтів зберігаються хеші паролів. Паролі хешуються алгоритмом md5 в три ітерації з додаванням солі (тексту, що не міститься в паролі, використовується для додаткового захисту).

Таблиця «orders» містить інформацію про всі замовлення, зокрема, має зовнішні ключі з унікальними ідентифікаторами водіїв, користувачів, типів оплати, а також містить статус поїздки та адресу висадки, адресу посадки, посилання на прокладений маршрут на гугл картах, час замовлення поїздки користувачем та час її завершення, якщо поїздка не була скасованою.

Таблиця «tarifs» містить інформацію, необхідну для обрахунку вартості поїздки. Для обрахунку вартості поїздки з таблиці береться значення ціни за кілометр (в гривнях), що відповідає встановленим параметрам типу оплати, класу транспортного засобу та часу доби, що визначає тариф («час пік», «звичайний», «нічний»). Ціна за кілометр множиться на відстань, що обраховується відповідно до прокладеного маршруту на карті.

3.6.1. Інтерфейс програми

На будь-якій сторінці веб-застосування в верхній частині екрану знаходиться навігаційне меню, що може видозмінюватись, в залежності від ролі користувача (неавторизований, клієнт або водій). На рисунку 9 показано, як навігація по сайту дозволяє переглядати інформацію різних сторінок, деякі навігаційні елементи, вбудовані у вигляді випадаючого списку (як наприклад «Водію»), а також сторінка містить кнопки «Увійти» або «Вихід».

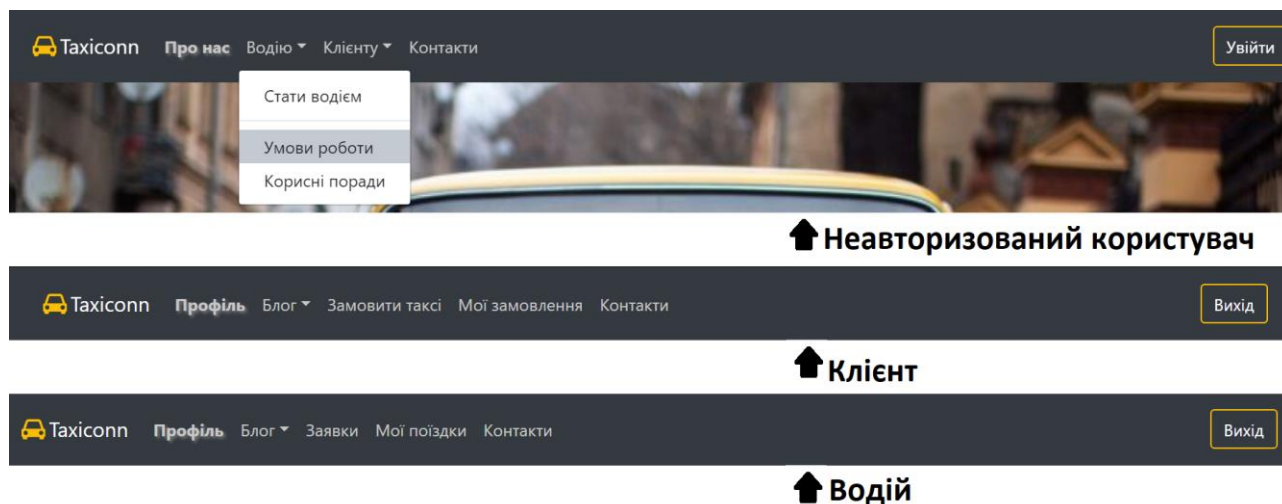


Рисунок 9 – Навігаційне меню

Форма для авторизації на сайті відображується у вигляді модального вікна, що також пропонує посилання на сторінки реєстрації. Натомість функція реєстрації реалізована у вигляді окремих сторінок для різних типів користувачів, тому що сторінки реєстрації повинні містити багато інформації та не зникати при випадкових натисненнях користувача поза модальним вікном.

Більшість сторінок лише відображають певну інформацію (більше прикладів сторінок в розділі 3.7), проте є сторінки, які динамічно змінюються. Наприклад, сторінка «Стати водієм», що є реалізацією реєстрації водія, при зміні значення селектору «Виробник авто» (список виробників авто також заповнюється з бази даних), змінює значення селектору «Марка авто», пропонуючи доступні для вибору марки авто обраного виробника та обраного класу (рисунок 10).

Рисунок 10 – Вибір марки авто

Сторінка «Заявки» у водіїв, що відображає замовлення таксі різних користувачів, асинхронно оновлюється для відображення нових заявок без

перезавантаження сторінки. На сторінці «Замовити таксі» у користувачів підключене автозаповнення адрес від Google (google autocomplete), яке пропонує варіанти адреси залежно від введених літер. Після введення адреси посадки та висадки без перезавантаження сторінки на карті будується маршрут поїздки (рисунок 13 в розділі 3.7). На сторінці «Контакти», що доступна навіть неавторизованим користувачам, також є карта, на якій повинен бути позначений головний офіс, але у пропонованій версії розробки позначена адреса НаУКМА (вулиця Григорія Сковороди, 2, Київ).

Більшість сторінок сайту мають статичну адресу (наприклад: /orders/my). Проте, для відображення профілів користувачів і водіїв, що доступні за посиланням в картках поїздок використовують динамічні адреси. Наприклад, /user/:user_id, де змінна user_id може набувати довільних значень та відображати профіль відповідного клієнта за запитом (за умови, що запит співпадає з унікальним ідентифікатором одного з користувачів).

Користувацький інтерфейс розроблено таким чином, щоб користуватися ним було зручно як клієнтам та і водіям. Деякі назви посилань та кнопок, особливо на сторінках для водіїв, названі з урахуванням розмовної лексики водіїв сфери таксі.

3.7 Інструкція користувача

При першому вході на сайт користувач бачить сторінку «Про нас», що містить навігаційне меню, контент (карусель зображень, список переваг використання та секцію «Думка розробника») та футер (Додаток А, рисунок 15). Користувач може ознайомитись з інформацією інших сторінок, використовуючи навігацію, зокрема сторінку контактів (Додаток А, рисунок 16), правила користування для клієнтів (Додаток А, рисунок 17), умови роботи для водіїв (Додаток А, рисунок 18), а також рекомендації для водіїв (Додаток А, рисунок 19).

Для зареєстрованих користувачів є можливість авторизації за допомогою зручного модального вікна (рисунок 11). Зареєструватись на сайті можна на відповідних сторінках реєстрації для водіїв (Додаток А, рисунок 20) або на сторінці швидкої реєстрації для клієнтів (Додаток А, рисунок 21). Перейти на вищевказані сторінки можна через відповідну кнопку «Зареєструватись» у вікні входу або через навігаційне меню (сторінки «Стати водієм» та «Стати клієнтом» відповідно). Реєстрація вимагає підтвердження електронної пошти, проте по-різному реалізована у водіїв та користувачів:

Рисунок 11 – вікно входу

- Користувач після швидкої реєстрації отримує на пошту тимчасовий пароль, який після першого входу система запропонує змінити (Додаток А, рисунок 22).
- Водій після завершення реєстрації переадресовується на сторінку (Додаток А, рисунок 23), на якій необхідно ввести код, надісланий системою на вказану електронну пошту.

При реєстрації користувачам встановлюється зображення профілю за замовчуванням, що складається з ініціалів та відповідає загальній стилістиці сайту.

Неавторизований користувач не може замовити таксі, проте може скористатись калькулятором цін на відповідній сторінці (Додаток А, рисунок 27). На сторінці також прокладається маршрут на карті, а вартість поїздки змінюється динамічно, залежно від обраних параметрів.

Після входу в систему (авторизації), відкривається сторінка профілю (рисунок 12), де можна переглянути вказану персональну інформацію, змінити її, завантажити фото або змінити пароль. Профіль водія додатково містить інформацію про автомобіль (Додаток А, рисунок 24).

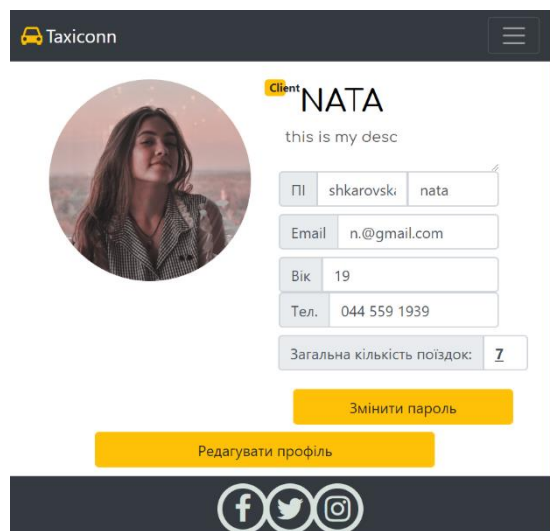


Рисунок 12 – профіль клієнта

Для замовлення таксі, необхідно перейти на сторінку «Замовити таксі» (рисунок 13), де потрібно ввести адреси посадки і висадки, змінити тип комфортності, якщо необхідно, а також обрати тип оплати. Тип комфортності автомобіля та спосіб оплати завжди встановлений за замовчуванням для зменшення часу, що витрачається на створення замовлення. Варто зазначити, що на автоматичне заповнення адрес визначене обмеження: адреси повинні бути в межах України, але вводити адреси можна будь-якою мовою. На сторінці також передбачено поле «Вартість», що динамічно обраховує вартість поїздки. Користувач не може мати одночасно більше двох незавершених замовлень, відповідно не може замовити поїздку, якщо має ще дві незавершені.

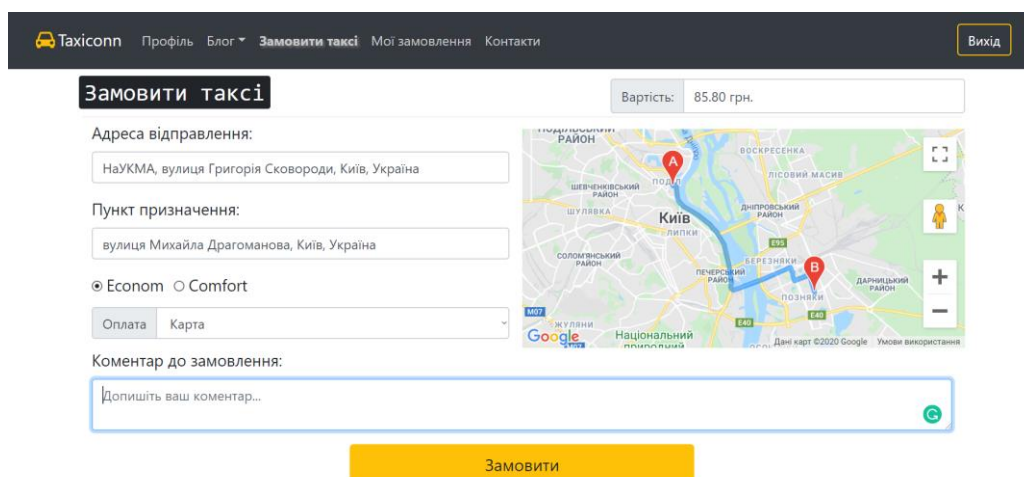


Рисунок 13 – Сторінка замовлення таксі

Після створення замовлення, система автоматично перенаправляє на сторінку «Мої замовлення», де можна відслідковувати поточну поїздку та

переглядати попередні замовлення (Додаток А, рисунок 25). Поки водій не прийняв замовлення, у користувача є можливість його скасувати, після прийняття водієм замовлення користувач може бачити як змінюється статус замовлення від «В подорожі» до «Завершено» (або «Скасовано»).

Для того, щоб прийняти перше замовлення, водій повинен перейти на сторінку «Заявки», де відображаються неприйняті замовлення (рисунок 14).

Перед прийняттям

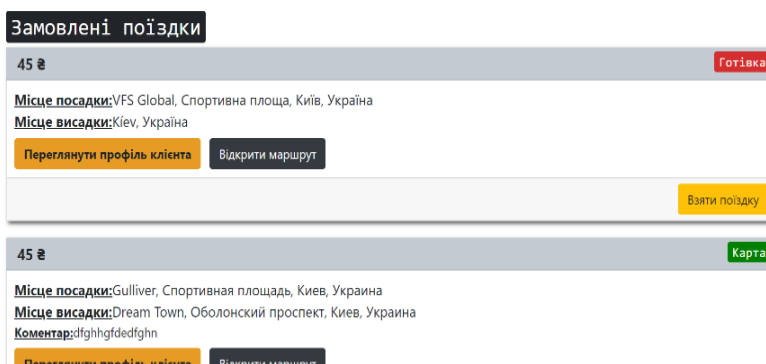


Рисунок 14 – Сторінка «Заявки»

замовлення водій може переглянути детальну інформацію про рекомендований маршрут та клієнта. Щоб взяти замовлення, необхідно натиснути кнопку «Взяти поїздку». Водій переходить на сторінку «Мої поїздки», де можна управляти поточною поїздкою, а також переглядати історію поїздок (Додаток А, рисунок 26). Водій одночасно може мати лише одну поїздку зі статусом «В подорожі».

Для того, щоб завершити роботу на сайті, необхідно натиснути кнопку «Вихід». Якщо користувач не вийде із системи, то при наступному вході на сайт він також буде авторизований, адже сайт використовує cookies. Необхідно обов'язково виходити з системи при відвідуванні сайту з чужого пристрою.

3.8 Перспективи розробленого веб-застосування

Даний проект може надалі розвиватися, розширюючи власний функціонал та додаючи нові можливості для покращення управління поїздками та збору статистики. Зокрема, у подальшому буде корисним розробити окрему частину веб-застосування для адміністраторів системи, щоб забезпечити зручний моніторинг підозрілої активності, а також додавання, редагування та видалення інформації в базі даних про виробників та марки автомобілів, правила користування сервісом, способи оплати, тощо.

Для клієнтів системи корисними та зручними будуть функції «Запланувати поїздку» (сказавши дату та час коли необхідно виїжджати) та створення спеціальних карток з адресами, що вводяться найчастіше для пришвидшення замовлення. Також можна надати користувачам можливість зберігати адреси з мітками, наприклад «дім», «робота», тощо. Також для пришвидшення реєстрації можна реалізувати авторизацію через гугл.

На сайті можна додати рейтингову систему оцінки поїздок та можливість обирати водіїв для поїздки, залежно від їх статусів. Додатково можна створити рейтинг клієнтів та реалізувати розумну систему підбору водіїв та клієнтів відповідно до їх рейтингу. В подальшому допускається реалізація оплати поїздок через веб-сайт та нарахування заробітної платні водіям автоматично, що підключили кредитну картку до свого акаунту. Також залежно від статусу та рейтингу водія можна обраховувати зарплатню за різними коефіцієнтами, а також реалізувати систему бонусів для водіїв з найкращим рейтингом.

Якщо система вийде на загальний ринок або буде викуплена компанією, що надає послуги таксі, можна додати систему промокодів для заохочення користувачів.

Висновки

У процесі роботи над розробкою веб-застосування по замовленню таксі було проведено дослідження сучасного стану сфери надання послуг з пасажирських перевезень в Україні. Аналіз алгоритмів роботи таксі в загальному та огляд аналогів розробки дозволили виокремити недоліки та переваги веб-застосувань, існуючих в сфері таксі. В свою чергу, це допомогло правильно визначити необхідний функціонал для веб-застосування, що розроблялося, а також чітко описати технічне завдання (включаючи корпоративні обмеження цілісності, вимоги до надійності та дизайну користувацького інтерфейсу). Вивчення сучасних засобів розробки дозволило побудувати оптимальну архітектуру веб-застосування та створити чітку логіку комунікації між різними частинами програми. Також це допомогло виявити, які додаткові бібліотеки та фреймворки варто підключати до проекту, а які будуть лише обтяжувати його. Загалом обрана клієнт-серверна архітектура була побудована на Node.js з використанням фреймворку Express.js, а обраний сервер-рендеринг дозволив реалізувати модель тонкого клієнта.

Розробка прикладного веб-застосування відбувалася поетапно з постійним покращенням попередніх версій програми. На проміжних етапах розробки проводилося ретельне тестування та відлагодження реалізованих функцій. Розроблене застосування, реалізоване у вигляді сайту, має декілька сторінок, наповнених статичним та динамічним контентом, два типи користувачів: клієнт та водій, підключені карти та автозаповнення від Google. Для користувачів передбачена реєстрація з підтвердженням електронної пошти та авторизація зі збереженням сесії користувача на ПК. В даному проекті реалізовані наступні функції: замовлення таксі для клієнтів, калькулятор цін поїздок, прийняття замовлення та завершення або скасування поїздки для водіїв, динамічна побудова маршруту на сайті та передбачення ціни поїздки при замовленні, завантаження фотографій користувачів в базу даних та управління персональною інформацією в акаунті, єдиному для будь-якого пристрою.

Отже, розроблене веб-застосування по замовленню таксі може використовуватись як автоматизована система, що не потребує участі диспетчера. Застосування доступне з будь-якого браузера, має інтуїтивно-зрозумілий інтерфейс та є конкурентоспроможним серед інших служб в сфері пасажирських перевезень, що надають послуги замовлень таксі онлайн. У подальшому функціонал системи може розширюватись додатковими можливостями для управління поїздками та збору статистики. Враховуючи отримані в ході напрацювань результати, можна переконливо стверджувати: визначена мета досягнена, а окреслені завдання проекту повністю виконані.

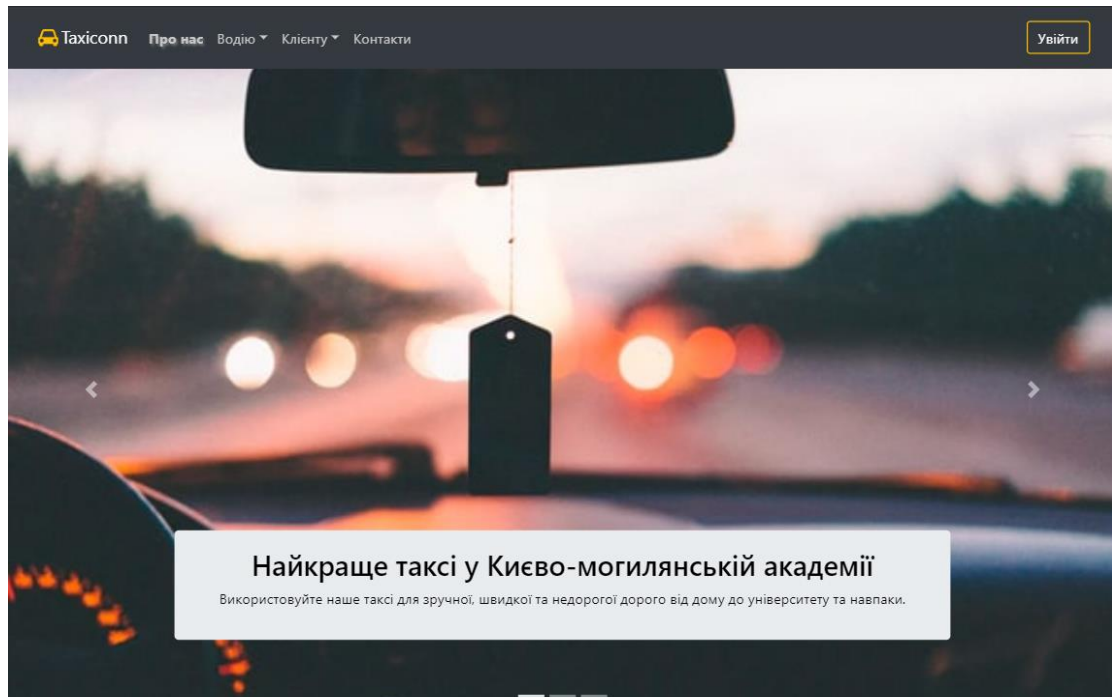
Перелік використаних джерел

- [1] Uber. (2 Березня 2020 р.). *Замовлення таксі*. Отримано з uber.com: <https://m.uber.com/looking>
- [2] Uklon. (25 Травень 2018 р.). *Головна сторінка*. Отримано з Uklon: <https://uklon.com.ua>
- [3] Bolt. (12 Березня 1029 р.). *Головна*. Отримано з Болт: <https://bolt-ua.com>
- [4] Таксі 838. (22 Листопад 2019 р.). *Місто Київ*. Отримано з taxi838: <https://taxi838.ua/kyiv>
- [5] Mary, B. (5 Лютий 2020 р.). *Hailing: History of the Taxi*. Отримано з thoughtco.com: <https://www.thoughtco.com/hailing-history-of-the-taxi-1992541>
- [6] FABRY, M. (2 Травень 2017 р.). *Now You Know: Why Are Taxi Cabs Yellow?* Отримано з Time: <https://time.com/4640097/yellow-taxi-cabs-history/>
- [7] Вікіпедія. (27 Липень 2019 р.). *Таксі*. Отримано з Вільна енциклопедія: <https://uk.wikipedia.org/wiki/Таксі>
- [8] ВВР. (5 Квітня 2001 р.). ЗАКОН УКРАЇНИ. *Про автомобільний транспорт*. Київ, Україна.
- [9] Harrington, J. L. (26 Квітень 2016 р.). *The Database Environment*. Отримано з sciencedirect: <https://doi.org/10.1016/B978-0-12-804399-8.00001-6>
- [10] Wagener, D. (14 Грудень 2019 р.). *Server-Side Rendering with Pug Templates*. Отримано з medium: <https://medium.com/@SigniorGratiano/server-side-rendering-with-pug-templates-e1e5947d4c1a>
- [11] W3S. (27 Вересень 2018 р.). *What is HTTP?* Отримано з w3schools: https://www.w3schools.com/whatis/whatis_http.asp
- [12] Patel, P. (14 Серпень 2018 р.). *Чем на самом деле является Node.js?* Отримано з Хабр: <https://habr.com/ru/post/420123/>




- [13] TJ Holowaychuk, StrongLoop. (26 Травень 2019 р.). *Початкова сторінка*. Отримано з Express: <https://expressjs.com/ru/>
- [14] Шкаровська, Н. (30 Березень 2020 р.). *Taxiconn*. Отримано з GitHub: <https://github.com/natahiko/Taxiconn>

Додатки

1 Додаток А – Додаткові ілюстрації



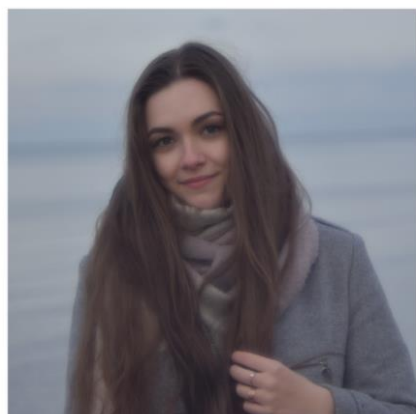
ПЕРЕВАГИ ВИКОРИСТАННЯ

 <p>Відображення заявок</p> <p>Зручно замовляйте таксі та переглядайте історію ваших поїздок в будь-який час.</p>	 <p>Доступні скрізь</p> <p>Замовляйте з будь-якої точки світу в будь-якому напрямку</p>	 <p>Без очікувань</p> <p>Мінімальний час очікування на опрацювання запитів, обробка запитів за лічені секунди!</p>
---	---	--

Цікаво знати

Таксі — засіб громадського транспорту, зазвичай легковий або вантажний автомобіль, використовується для перевезення пасажирів і вантажів у будь-яке вказане місце з оплатою проїзду машини за таксою за лічильником — таксометром.

Згідно з даними інтернет-порталу Prostoturist, станом на березень 2015 року середній тариф на таксі Києва становить 6 гривень за 1 кілометр. За основу бралася вартість поїздки з центру міста в спальний район. Протяжність маршруту склала 13 кілометрів.



Думка розробника

Я вважаю, що користування таким сервісом має приносити і водіям і клієнтам лише задоволення. Особисто я була надзвичайно захоплена його розробкою, сайт створений з чітко продуманою структурою, щоб користування було приємним та легким, тому сподіваюсь, що він знайде відгук в кожному серці.

P.S. Час плинний - Могилянка вічна



Рисунок 15 – головна сторінка «Про нас»

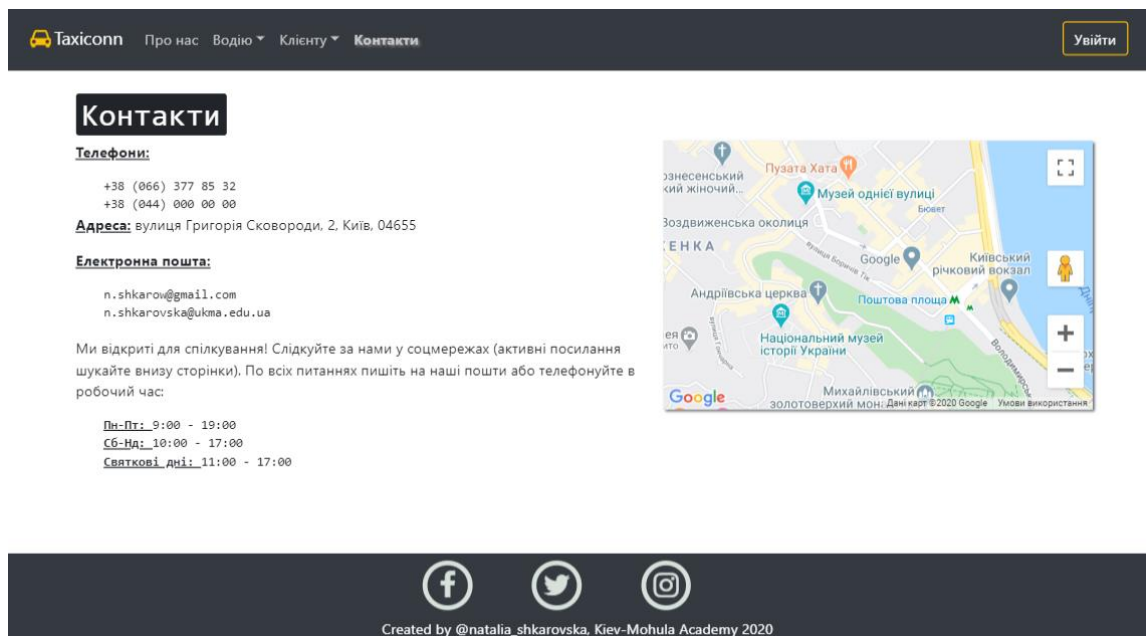


Рисунок 16 – сторінка «Контакти»

Правила користування для клієнтів

Щоб покращити нас сервіс та зробити його приємним і для водіїв і для клієнтів ми рекомендуємо дотримуватись рекомендацій та правил користування нашим сервісом таксі

Взаємодотримання правил зробить вашого водія кращим, а вашу поїздку приємнішою, окрім того, **ЗА НЕДОТРИМАННЯ ПРАВИЛ ВОДІЙ МАЄ ПОВНЕ ПРАВО ЗАВЕРШИТИ ПОЇЗДКУ ДОСТРОКОВО!** Ці правила є загальними правилами для більшості схожих сервісів нашої країни, тому їх повинен знати кожен

Етикет

Будьте ввічливими з водієм, вітайтеся та прощайтесь. За необхідності не соромтесь ставити питання, але не обтяжуйте водія зайвими розмовами. Ведіть себе чемно, адже це запорука гарного настрою вас та вашого водія!

Mon Mar 16 2020

Стриманість

Якщо ви потрапили в затори або спізнюєтесь на важливу зустріч, повідомляйте водія про це спокійно! Пам'ятайте, що водій не може впливати на ситуацію на дорогах або порушувати ПДД через те, що вам "потрібно швидше". Також дійте розважливо в надзвичайних ситуаціях, якщо такі виникли.

Thu Mar 26 2020

Чистота

Підтримуйте чистоту в салоні за можливості. Не трітьтесь взуттям об м'яку оббивку салону та не залишайте після себе сміття. Також не бажано їсти та пити під час поїздки, особливо якщо упаковка їжі не призначена для поїздки (для напоїв має бути посудина з кришкою). Також не варто в салоні перевозити та відкривати речі/продукти які мають їдкі запахи.

Fri Mar 20 2020

Пунктуальність

Не змушуйте водія чекати довше 5 хвилин на місці посадки. Пам'ятайте, що після очікування понад 5 хвилин водій має право скасувати поїздку. Також бажано не затримуватись при виході. Під'їждючи до назначеного місця заздалегідь почніть збирати свої речі та готувати гроші, якщо обрали готівковий вид розрахунку. Також зупинки під час руху за вимогою водія може відхиляти, якщо не бачить в цьому нагальної потреби.

Tue Mar 24 2020

Місткість


Будь-яке авто має свої параметри та вантажопід'ємність. Не варто використовувати таксі як грузовик. Якщо габарити ваших речей перевищують розміри багажника водія він не має права перевозити вас з цим багажем. Також заборонено перевозити людей в кількості більшій за встановлену кількістю пасажирських місць в авто.

Thu Mar 26 2020

Сподіваємось, ви дотримуватимесь вищевказаних правил та ваші поїздки з нашим сервісом будуть виключно приємними!!!

Рисунок 17 – сторінка «Правила користування»

A large fleet of yellow taxis parked in rows, serving as the background for the advertisement. The taxis are arranged in neat lines, filling the frame. The background image is a high-angle shot of a large parking lot or taxi stand filled with numerous yellow taxis. The taxis are parked in neat rows, and their bright yellow color stands out against the darker pavement. The perspective is from above, looking down at the fleet. The overall impression is one of a large, organized fleet of vehicles.

 Taxiconn Профіль Блог Замовити таксі Мої замовлення Контакти

Вихід

Робота в нас - це мрія будь-якого водія!!! Приєднуйтесь вже зараз

Кожен наш працівник - особистість, тому ми піклуємось про вас, вашу безпеку та надаємо найкращі умови праці

Стати водієм

Вимоги до автомобіля

- Авто має бути випущене після 2000-го року
- Автомобіль повинен мати не менше 4 дверей
- Повинне бути дійсне свідоцтво про реєстрацію транспортного засобу
- Вартість автомобіля економ-класу повинна бути більшою за 140 000 (сто сорок тисяч) грн

ОСНОВНІ ВИМОГИ ДО ВОДІЇВ

 **Посвідчення водія**
Водій завжди повинен при собі мати посвідчення відповідної категорії, а також бажано паспорт

 **Стаж від 3 років**
Ви маєте мати практику
водиння автомобілем від 3
років, при цьому без ДТП за останній рік

Документи

При собі завжди потрібно мати Ліцензію на перевезення пасажирів та Свідотство на авто

СПИСОК ДОЗВОЛЕНИХ МАШИН

Model	Category	Market Share (%)
1 Series	Економ	12.5
2 Series	Комфорт+	12.5
3 Series	Комфорт+	10.0
4 Series	Комфорт+	7.5
5 Series	Економ	7.5
6 Series	Економ	5.0
M2	Економ	2.5
M3	Економ	2.5

Рисунок 18 – сторінка «Умови роботи»

Корисні поради водіям **taxiconn.com.ua**

Щоб мати рекомендації серед користувачів, а також отримувати можливі додаткові чайові від клієнтів можете скористатися нашими порадами, що нададуть вам змогу покращити ваш сервіс

Також в цьому розділі ви можете знайти лайфхаки, що полегшать вашу роботу та зроблять її ще приємнішою для вас та вашого автомобіля, що є важливою складовою наших працівників

Автомобіль

- Слідкуйте за чистотою вашого авто
- Підтримуйте чистоту в салоні та не тримайте в салоні зайвих речей
- Регулярно проходите ТехОгляд
- Краще, щоб в салоні не було зайвих запахів: цигарки, ароматизатори, кава...



Водій

- Будьте охайними: чистий одяг, відсутність неприємних запахів...
- Не запізнюйтесь, а приїхавши заздалегідь не надзвонюйте клієнтові
- Користуйтеся навігатором (навіть якщо знаєте короткий шлях)
- Дотримуйтесь Правил Дорожнього Руху

Пасажир

- Обов'язково вітайтеся та прощайтеся з клієнтом



Рисунок 19 – сторінка «Корисні поради»

Стати водієм **taxiconn.com.ua**

Особиста інформація

Ім'я	<input type="text"/>		
Прізвище	<input type="text"/>		
Логін	<input type="text" value="nata"/>		<input type="button" value="Перевірити"/>
Email	Електронна пошта	<input type="text" value="@gmail.com"/>	
Вік	<input type="text"/>	Телефон	<input type="text"/>
<input style="height: 40px;" type="text" value="Опишіть себе (за бажанням)..."/>			

Водійське посвідчення та авто

Серія та номер	ВАН	<input type="text" value="000000"/>	
Категорія	Оберіть...	Рік випуску	2018
Номер авто	XX	0000	XX
<input checked="" type="radio"/> Економ <input type="radio"/> Комфорт			
Виробник	Оберіть виробника		
Марка	Оберіть марку		
Оберіть пароль для входу			
Пароль	<input type="password" value="...."/>		Повторіть пароль

Стати водієм

Рисунок 20 – сторінка реєстрації водіїв


Taxiconn
Про нас
Водію
Клієнту
Контакти

Увійти

Швидка реєстрація
taxiconn.com.ua

Ім'я

Ім'я

Прізвище

Прізвище

Логін

Логін

Перевірити

Email

Електронна пошта

@gmail.com

Вік

Телефон

Опишіть себе (за бажанням)...

Зареєструватись





Created by @natalia_shkarovska, Kiev-Mohula Academy 2020

Рисунок 21 – сторінка реєстрації для клієнтів


Taxiconn
Про нас
Водію
Клієнту
Контакти

Увійти

Змініть пароль

Старий пароль:

Новий пароль:

Повторіть пароль:


Зберегти зміни





Created by @natalia_shkarovska, Kiev-Mohula Academy 2020

Рисунок 22 – Сторінка зміни паролю після реєстрації для клієнтів





Taxiconn
Про нас
Водію
Клієнту
Контакти

Увійти

Підтвердіть реєстрацію


На вказану вами пошту було надіслано код підтвердження, скопіюйте його та вставте в поле нижче:

Підтвердити

Created by @natalia_shkarovska, Kiev-Mohula Academy 2020

Рисунок 23 – сторінка підтвердження реєстрації для водіїв



Driver

NATA

ПІ	Шкаровська	Наталія	
Email	n.shkarow@ukr.net		
Вік	35	Тел. 1234567890	
Загальна кількість поїздок:	3	Змінити пароль	
Номер	FF2345FF	Права dfg	Клас econom
Авто	Skoda	Rapid	2000

Редагувати профіль

Рисунок 24 – профіль водія

В подорожі

Онлайн

Ціна: 45 ₴

Місце посадки: Dragomanova, Kiev

Місце висадки: Lomonosova, Kiev

Коментар: dfgvbngedfgb

Переглянути профіль водія

Відкрити маршрут

Замовлено

Карта

Ціна: 28 ₴

Місце посадки: Fairmont Grand Hotel - Kyiv, Київ, Україна

Місце висадки: Win Bar, вулиця Хорива, Київ, Україна

Скасувати замовлення

Завершені поїздки

Скасовано клієнтом

Готівка

Ціна: 87 ₴

Місце посадки: Dream Town, Оболонський проспект, Київ, Україна

Місце висадки: Fairmont Grand Hotel - Kyiv, Київ, Україна

Переглянути профіль водія

Рисунок 25 – сторінка «Мої поїздки» клієнта

В подорожі

Онлайн

Ціна: 45 ₴

Місце посадки: Dragomanova, Kiev

Місце висадки: Lomonosova, Kiev

Коментар: dfgvbngedfgb

Переглянути профіль клієнта

Відкрити маршрут

Скасувати замовлення

Завершити поїздку

Завершені поїздки

Завершено

Карта

Ціна: 45 ₴

Тривалість поїздки: 00:33:24

Місце посадки: Мала Вільшанка, Киевская область, Украина

Місце висадки: улица Михаила Драгоманова, 9, Киев, Украина

Переглянути профіль клієнта

Рисунок 26 – сторінка «Мої поїздки» водія

Рисунок 27 – сторінка «Калькулятор цін»

2 Додаток Б – Список ілюстрацій

Рисунок 1 – ЛОГОТИП UBER	13
Рисунок 2 - ЛОГОТИП UKLON	14
Рисунок 3 - ЛОГОТИП VOLT	14
Рисунок 4 - АРХІТЕКТУРА ВЕБ-ЗАСТОСУВАННЯ	27
Рисунок 5 – ФАЙЛОВА СТРУКТУРА ПРОЕКТУ	30
Рисунок 6 – АЛГОРИТМ РОБОТИ МЕТОДУ СЕРВЕРА БЕЗ ДАНИХ	31
Рисунок 7 – АЛГОРИТМ РОБОТИ ЗАПИТУ З ДАНИМИ	32
Рисунок 8 – СХЕМА БАЗИ ДАНИХ	33
Рисунок 9 – НАВІГАЦІЙНЕ МЕНЮ	35
Рисунок 10 – ВИБІР МАРКИ АВТО	35
Рисунок 11 – ВІКНО ВХОДУ	37
Рисунок 12 – ПРОФІЛЬ КЛІЄНТА	38
Рисунок 13 – СТОРІНКА ЗАМОВЛЕННЯ ТАКСІ	38
Рисунок 14 – СТОРІНКА «ЗАЯВКИ»	39
Рисунок 15 – ГОЛОВНА СТОРІНКА «ПРО НАС»	46
Рисунок 16 – СТОРІНКА «КОНТАКТИ»	46
Рисунок 17 – СТОРІНКА «ПРАВИЛА КОРИСТУВАННЯ»	47

	54
РИСУНОК 18 – СТОРІНКА «УМОВИ РОБОТИ»	48
РИСУНОК 19 – СТОРІНКА «КОРИСНІ ПОРАДИ»	49
РИСУНОК 20 – СТОРІНКА РЕЄСТРАЦІЇ ВОДІЇВ	49
РИСУНОК 21 – СТОРІНКА РЕЄСТРАЦІЇ ДЛЯ КЛІЄНТІВ	50
РИСУНОК 22 – СТОРІНКА ЗМІНИ ПАРОЛЮ ПІСЛЯ РЕЄСТРАЦІЇ ДЛЯ КЛІЄНТІВ	50
РИСУНОК 23 – СТОРІНКА ПІДТВЕРДЖЕННЯ РЕЄСТРАЦІЇ ДЛЯ ВОДІЇВ	51
РИСУНОК 24 – ПРОФІЛЬ ВОДІЯ	51
РИСУНОК 25 – СТОРІНКА «МОЇ ПОЇЗДКИ» КЛІЄНТА	52
РИСУНОК 26 – СТОРІНКА «МОЇ ПОЇЗДКИ» ВОДІЯ	52
РИСУНОК 27 – СТОРІНКА «КАЛЬКУЛЯТОР ЦІН»	53

3 Додаток В – Програмний код

server.js

```

const express = require('express'),
    pug = require('pug'),
    bodyParser = require('body-parser'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'), path = require('path'),
    multer = require('multer'),
    db = require('./modules/database_pool'),
    functions = require('./modules/functions'),
    text = require('./config/main.json'),
    config = require('./config/conf.json');

const server = express();
server.use(cookieParser());
server.listen(config['port']);
server.use(express.static(__dirname));
server.use(express.static('public'));
server.use(express.static('files'));
server.use(bodyParser.urlencoded({ extended: true }));
server.use(bodyParser.json());
console.log('Server is running http://localhost:' + config['port']);

server.use('/user', require('./modules/users'));
server.use('/driver', require('./modules/drivers'));
server.use('/orders', require('./modules/orders'));
//use fot test
// server.use('/default', require('./modules/default'));

//storage for temporari photo saving
let storage = multer.diskStorage({
  destination: function (req, file, callback) {
    callback(null, "../files/uploads");
  },
  filename: function (req, file, cb) {
    cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname));
  }
});

```

```

    }
  });

// method for uploading photo
const upload_user_photo = multer({storage: storage});
server.post('/upload_user_photo', upload_user_photo.single('photo'), function (req, res) {
  if (!req.file) {
    console.log("No file received");
    res.json({ "empty": true });
  } else {
    //if fine found
    fs.readFile(req.file.path, (err, data) => {
      const text = "data:" + req.file.mimetype + ";base64," + data.toString('base64');
      res.json({ "src": text });
      res.end();
      //save in localStorage for use whe save
      functions.savePhotoSrc(req.cookies, text, req.file.path);
      const photo_url = functions.getPhotoUrlForDel(req.cookies.userid);
      try {
        //removing file
        fs.unlinkSync(__dirname + photo_url.substr(5));
      } catch (e) { console.log(e); }
    });
  }
});

//main page
server.get(['/aboutus', '/'], function (req, res) {
  text.header['nowpage'] = "nav_aboutus";
  res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
  res.write(fs.readFileSync(__dirname + '/src/html/aboutus.html', 'utf8'));
  res.end();
});

//defaul footer for including in html files
server.get('/getfooter', function (req, res) {
  res.statusCode = 200;
  res.write(pug.renderFile(__dirname + "/src/pugs/footer.pug"));
  res.end();
});

//users profiles only for autorised
server.get('/profile', function (req, res) {
  const user = req.cookies.authorised;
  const userid = req.cookies.userid;
  //set default page
  text.header['nowpage'] = "nav_profile";
  //check usertype
  if (user === 'drivers') {
    db.getCon().query(functions.getSQLProfileDriver(userid), function (err, result) {
      if (err) res.redirect("/404");
      else {
        //render resulted pug
        res.write(pug.renderFile(__dirname + "/src/pugs/header-driver.pug", text.header));
        db.getCon().query("SELECT prodid,producer FROM car_producer;", function (err, result2) {
          res.write(pug.renderFile(__dirname + "/src/pugs/profile-driver.pug", {
            producers: result2,
            info: result[0]
          }));
          res.end();
        });
      }
    });
  }
});

```

```

} else if (user === 'clients') {
  db.getCon().query(functions.getSQLProfileClient(userid), function (err, result) {
    if (err) res.redirect("/404");
    else {
      //render resulted pug
      res.write(pug.renderFile(__dirname + "/src/pugs/header-client.pug", text.header));
      res.write(pug.renderFile(__dirname + "/src/pugs/profile-client.pug", {
        info: result[0]
      }));
      res.end();
    }
  });
} else res.redirect("/");
});

//page where drivers enter code after registration
server.get('/confirmregistration', function (req, res) {
  text.header['nowpage'] = "";
  const usertype = req.cookies.authorised;
  if (usertype === 'clients' || usertype === 'drivers') {
    res.redirect("/404");
  } else {
    res.write(pug.renderFile(__dirname + "/src/pugs/header-unauthorized.pug", text.header));
    res.write(pug.renderFile(__dirname + "/src/pugs/confirmregistration.pug", {
      "email": req.query.email + req.query.emend
    }));
  }
  res.end();
});

//page for which redirect after enter confirm code
server.get('/confirmregistratcode', function (req, res) {
  text.header['nowpage'] = "";
  res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
  let code = req.query.code;
  let email = req.query.email;
  let carmodel = functions.getCarModelId(code);
  if (code === "" || email === "") {
    res.write(pug.renderFile(__dirname + "/src/pugs/unsucccessRegistered.pug"));
    res.write(pug.renderFile(__dirname + "/src/pugs/confirmregistration.pug", {
      "email": email
    }));
    res.end();
  } else {
    //addind confirmed friver to database
    let sql = functions.getRegisretDriverSQL(email, code, carmodel);
    db.getCon().query(sql, function (err) {
      if (err) {
        res.write(pug.renderFile(__dirname + "/src/pugs/unsucccessRegistered.pug"));
        res.write(pug.renderFile(__dirname + "/src/pugs/confirmregistration.pug", {
          "email": email
        }));
      } else
        res.write(pug.renderFile(__dirname + "/src/pugs/successRegistered.pug"));
      res.end();
    });
  }
});

// contact page
server.get('/contacts', function (req, res) {
  text.header['nowpage'] = "nav_contacts";
  res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
  res.write(pug.renderFile(__dirname + "/src/pugs/contacts.pug", {

```



```

    "googlemapapi": config.googlemapapi
  });
  res.end();
});

server.get('/thankspage', function (req, res) {
  text.header['nowpage'] = "";
  res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
  res.write(fs.readFileSync(__dirname + '/src/html/thanks.html', 'utf8'));
  res.end();
});

//methods using fot changing password for user first time after registration
server.get('/changepassword', function (req, res) {
  text.header['nowpage'] = "";
  res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
  res.write(pug.renderFile(__dirname + "/src/pugs/changePass.pug"));
  res.end();
});

//page for ordering taxi
server.get('/ordertaxi', function (req, res) {
  text.header['nowpage'] = "nav_ordertaxi";
  //set google Api token
  text.header['googlemapapi'] = config.googlemapapi;
  const usertype = req.cookies.authorised;
  if (usertype === 'drivers') res.redirect("/orders/my");
  else if (usertype === 'clients') {
    //get payments type
    db.getCon().query('SELECT * FROM payments', function (err, result) {
      res.write(pug.renderFile(__dirname + "/src/pugs/header-client.pug", text.header));
      res.write(pug.renderFile(__dirname + "/src/pugs/ordertaxi.pug", {
        "pay_types": result,
        "googlemapapi": config.googlemapapi
      }));
      res.end();
    });
  } else res.redirect("/user/registeruser");
});

server.get('/registered', function (req, res) {
  text.header['nowpage'] = "";
  db.getCon().query(functions.getSQLRedisterUser(req.query), function (err, result) {
    functions.sendUserMail(req.query.login, config.email, text.usermail);
    if (err) res.json({"data": false});
    else {
      res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
      res.write(pug.renderFile(__dirname + "/src/pugs/registered_client.pug"));
    }
    res.end();
  });
});

//methods for chech login
server.post('/login', function (req, res) {
  let login = req.body.login;
  let type = req.body.type;
  let password = functions.hashPassword(req.body.password);
  //chech for correct data in database
  db.getCon().query(functions.getSQLLogin(login, type, password), function (err, result) {
    if (result.length < 1) res.statusCode = 401;
    else {
      res.statusCode = 200;
    }
  });
});

```

```

    res.setHeader('Content-Type', 'application/json');
    //send user_id to set in cookies
    let responseBody = {
      'userid': result[0].id,
      'registered': result[0].registered
    };
    res.write(JSON.stringify(responseBody));
  }
  res.end();
});
});

//change password in user profile
server.put('/password', function (req, res) {
  let old = req.body.old;
  let now = req.body.now;
  let type = req.body.type;
  if (old === "" || old === undefined || now === "" || now === undefined) {
    res.statusCode = 406;
    res.write(JSON.stringify({"err": 'lost some data'}));
    res.end();
    return;
  }
  //password hashing
  now = functions.hashPassword(now);
  old = functions.hashPassword(old);
  let userid = req.cookies.userid;
  //check for correct old password
  db.getCon().query(functions.getSQLPassword(type, userid, old), function (err, result) {
    if (result.length < 1) {
      res.statusCode = 409;
      res.write(JSON.stringify({"err": 'uncorect old'}));
      res.end();
    } else {
      //set new password
      db.getCon().query("UPDATE " + type + " SET password='" + now + "' WHERE id='" + userid + "'", function
(err) {
        if (err) {
          res.write(JSON.stringify({
            "err": 'Внутрішня помилка на сервері'
          }));
          res.statusCode = 409;
        } else res.statusCode = 204;
        res.end();
      });
    }
  });
});

//methods for drivers to accept order
server.post('/acceptOrder', function (req, res) {
  if (req.cookies.authorised !== 'drivers') {
    res.end();
    return;
  }
  const userid = req.cookies.userid;
  db.getCon().query("SELECT * FROM orders WHERE driver_id='{ }' AND (status=1 OR status=2)".format(userid),
function (err0, result0) {
  if (result0.length >= 1) {
    res.statusCode = 405;
    res.end()
  } else {
    db.getCon().query("UPDATE orders SET driver_id='{ }', status=1 WHERE id='{ }'".format(userid,

```

```

req.body.orderid), function () {
    res.statusCode = 200;
    res.end();
});
}
});
});

//get all car models independ on producer
server.post('/carmodel', function (req, res) {
    let producer = req.body.producer;
    if (req.body.secret_key !== config.select_carmodel_token) {
        res.statusCode = 406;
        res.end();
        return;
    }
    db.getCon().query("SELECT id,model FROM car_models WHERE producer_id='{ }' AND
class='{ }';".format(producer, req.body.carclass), function (err, result) {
        res.statusCode = 202;
        res.setHeader('Content-Type', 'application/json');
        res.write(JSON.stringify({
            "res": result
        }));
        res.end();
    });
});

//checking free login
server.post('/isLoginFree', function (req, res) {
    let login = req.body.login;
    if (login === "") {
        res.statusCode = 406;
        res.end();
        return;
    }
    db.getCon().query(functions.getSQLIsLoginFree(req.body.type, login), function (err, result) {
        res.setHeader('Content-Type', 'application/json');
        let datares = { "free": false };
        if (result.length < 1) {
            datares['free'] = true;
        }
        res.write(JSON.stringify(datares));
        res.end();
    });
});

//checking free login, email, phone for one user
server.post('/isAllFree', function (req, res) {
    let login = req.body.login;
    let phone = req.body.number;
    let email = req.body.email;
    if (login === "" || phone === "" || email === "") {
        res.statusCode = 406;
        res.end();
        return;
    }
    //create sql request
    const sql = functions.getSQLIsAllFree(req.body.client, login, phone, email);
    db.getCon().query(sql, function (err, result) {
        res.setHeader('Content-Type', 'application/json');
        if (result.length < 1) {
            res.statusCode = 200;
            res.write(JSON.stringify({ }));
        }
    });
});

```

```

        res.end();
    } else {
        let user = result[0];
        res.statusCode = 409;
        let datares = { "email": false, "phone": false, "login": false };
        if (email === user.email) datares['email'] = true;
        if (phone === user.phone) datares['phone'] = true;
        if (login === user.login) datares['login'] = true;
        res.write(JSON.stringify(datares));
        res.end();
    }
});
});

//post request for chaging user password first time
server.post('/changepass', function (req, res) {
    db.getCon().query(functions.getSQLChageUserPassword(req.cookies, req.body), function (err) {
        if (err) res.json({ "data": false });
        else res.json({ "data": true });
        res.end();
    })
});

//sending main for drivers to confirm registration
server.post('/sendmail', function (req, res) {
    let emailTo = req.body.email;
    let code = functions.generateCode();
    functions.setCode(emailTo, code);
    if (functions.send(emailTo, code, config.email, text.email)) {
        //save driver info in local storage
        functions.setRegisretDriverInfo(code, req.body);
        res.setHeader('Content-Type', 'application/json');
        res.write(JSON.stringify({ "res": true }));
    } else res.statusCode = 400;
    res.end();
});

//update user profile info
server.put('/profile', function (req, res) {
    let sql;
    const type = req.cookies.authorised;
    if (type === 'drivers') sql = functions.getDriverSQLUpdate(req.cookies.userid, req.body);
    else if (type === 'clients') sql = functions.getClientSQLUpdate(req.cookies.userid, req.body);
    else {
        res.statusCode = 401;
        res.end();
        return;
    }
    db.getCon().query(sql, function () {
        const sql2 = functions.getSQLUploadPhoto(req.cookies);
        if (sql2 !== "") {
            db.getCon().query(sql2, function () {
                res.end();
            });
        }
    });
});

// default error page
server.use(function (req, res) {
    text.header['nowpage'] = "";
    res.write(pug.renderFile(__dirname + functions.getHeader(req.cookies.authorised), text.header));
    res.write(fs.readFileSync(__dirname + '/src/html/404.html', 'utf8'));
    res.end();
});

```

```
});

String.prototype.format = function () {
  var i = 0, args = arguments;
  return this.replace(/{} /g, function () {
    return typeof args[i] !== 'undefined' ? args[i++] : "";
  });
};
```

modules

database_pool.js

```
const mysql = require('mysql');
const config = require('../config/conf.json');
class Database_pool {
  static instance = new Database_pool();
  constructor() {
    //create connection
    this.con = mysql.createConnection(config.database);
  }
  static getCon(){
    //return cinnnection
    return Database_pool.instance.con;
  }
}
```

```
module.exports = Database_pool
```

drivers.js

```
const router = require('express').Router(),
  db = require('./database_pool'),
  text = require('../config/main.json'),
  functions = require('./functions'),
  pug = require('pug'),
  path = require('path'), fs = require('fs');

//return work cond page with info about car models
router.get('/workcond', function (req, res) {
  text.header['nowpage'] = "nav_driver";
  res.write(pug.renderFile(path.join(__dirname, '..') + functions.getHeader(req.cookies.authorised), text.header));
  db.getCon().query("SELECT * FROM car_models INNER JOIN car_producer ON car_producer.prodid = car_models.producer_id ORDER BY producer;", function (err, result) {
    res.write(pug.renderFile(path.join(__dirname, '../src/pugs/workcond.pug'), {
      allmodels: result
    }));
    res.end();
  });
});

//usefull tips html page
router.get('/usefultips', function (req, res) {
  text.header['nowpage'] = "nav_driver";
  res.write(pug.renderFile(path.join(__dirname, '..') + functions.getHeader(req.cookies.authorised), text.header));
  res.write(fs.readFileSync(path.join(__dirname, '../src/html/usefultips.html'), 'utf8'));
  res.end();
});

//become driver page
router.get('/becomedriver', function (req, res) {
  let user = req.cookies.authorised;
  if (user === 'drivers') res.redirect("/profile");
  else {
    text.header['nowpage'] = "nav_driver";
    res.write(pug.renderFile(path.join(__dirname, '..') + functions.getHeader(req.cookies.authorised), text.header));
    db.getCon().query("SELECT prodid,producer FROM car_producer;", function (err, result) {
```

```

        res.write(pug.renderFile(path.join(__dirname, '../src/pugs/becomedriver.pug'), {
            producers: result
        }));
        console.log(result);
        res.end();
    });
}
});

//pages with dynamic driver ids for show profile without editind
router.param('driverid', function (req, res, next, driverid) {
    req.userid = driverid;
    return next();
});
router.get('/:driverid', function (req, res) {
    text.header['nowpage'] = "";
    const userid = req.userid;
    db.getCon().query("SELECT * FROM drivers WHERE id='{ }'".format(userid), function (err, result) {
        if (err || result.length < 1) res.redirect("/404");
        else {
            db.getCon().query(functions.getSQLProfileDriver(userid), function (err2, result2) {
                text.header['nowpage'] = "/driverprofile/" + userid;
                res.write(pug.renderFile(path.join(__dirname, '../') + functions.getHeader(req.cookies.authorised),
text.header));
                res.write(pug.renderFile(path.join(__dirname, '../src/pugs/profile_driver_for_client.pug'), {
                    info: result[0],
                    producers: result2[0]
                }));
                res.end();
            });
        }
    });
});

String.prototype.format = function () {
    var i = 0, args = arguments;
    return this.replace(/{/g, function () {
        return typeof args[i] !== 'undefined' ? args[i++] : "";
    });
};
module.exports = router;

```

users.js

```

const router = require('express').Router(),
    db = require('./database_pool'),
    text = require('./config/main.json'),
    functions = require('./functions'),
    pug = require('pug'),
    path = require('path'), fs = require('fs');

//page with user rules from db
router.get('/userrules', function (req, res) {
    text.header['nowpage'] = "nav_client";
    res.write(pug.renderFile(path.join(__dirname, '../') + functions.getHeader(req.cookies.authorised), text.header));
    db.getCon().query("SELECT * FROM blogs;", function (err, result) {
        res.write(pug.renderFile(path.join(__dirname, '../src/pugs/userrules.pug'), {
            blogs: result
        }));
        res.end();
    });
});

//quick register page
router.get('/registeruser', function (req, res) {

```

```

let user = req.cookies.authorised;
if (user === 'clients') res.redirect("/profile");
else {
  text.header['nowpage'] = "nav_client";
  res.write(pug.renderFile(path.join(__dirname, '..') + functions.getHeader(req.cookies.authorised), text.header));
  db.getCon().query("SELECT prodid,producer FROM car_producer;", function (err, result) {
    res.write(pug.renderFile(path.join(__dirname, '../src/pugs/becomeclient.pug'), {
      producers: result
    }));
    res.end();
  });
}
});

//pages with dynamic client ids for show profile without editind
router.param('userid', function (req, res, next, userid) {
  req.userid = userid;
  return next();
});
router.get('/:userid', function (req, res) {
  text.header['nowpage'] = "";
  const userid = req.userid;
  db.getCon().query(functions.getSQLProfileClient(userid), function (err, result) {
    if (err || result.length < 1) res.redirect("/404");
    else {
      text.header['nowpage'] = "/userprofile/" + userid;
      res.write(pug.renderFile(path.join(__dirname, '..') + functions.getHeader(req.cookies.authorised), text.header));
      res.write(pug.renderFile(path.join(__dirname, '../src/pugs/profile_client_for driver.pug'), {
        "info": result[0]
      }));
      res.end();
    }
  });
});
String.prototype.format = function () {
  var i = 0, args = arguments;
  return this.replace(/{} /g, function () {
    return typeof args[i] !== 'undefined' ? args[i++] : " ";
  });
};
module.exports = router;

```

orders.js

```

const router = require('express').Router(),
db = require('./database_pool'),
text = require('./config/main.json'),
functions = require('./functions'),
pug = require('pug'),
path = require('path'), fs = require('fs'),
config = require('./config/conf.json');

router.get('/', function (req, res) {
  if (req.cookies.authorised !== 'drivers') res.redirect("/404");
  else {
    text.header['nowpage'] = "nav_orders";
    res.write(pug.renderFile(path.join(__dirname, '../src/pugs/header-driver.pug'), text.header));
    const userid = req.cookies.userid;
    db.getCon().query("SELECT class FROM car_models WHERE id IN (SELECT carmodelid FROM drivers WHERE id='{ }');".format(userid),
      function (err0, result0) {
        const clas = functions.setCarModelId(userid, result0);
        db.getCon().query("SELECT * FROM orders INNER JOIN payments ON payments.pay_id=orders.pay_type_id WHERE status=0 AND class='{ }';".format(clas), function (err, result) {
          res.write(pug.renderFile(path.join(__dirname, '../src/pugs/orders.pug"), {

```

```

        "orders": result
    });
    res.end();
});
});
}
});

router.post('/endDrive', function (req, res) {
    db.getCon().query(functions.getSQLEndDrive(req.body.order_id, req.cookies.userid), function () {
        res.json({ "data": true });
        res.end();
    })
});

router.post('/createorder', function (req, res) {
    const userid = req.cookies.userid;
    if (userid === undefined) {
        res.statusCode = 401;
        res.write(JSON.stringify({ "err": "no user id in cookies" }));
        res.end();
        return;
    }
    const from = req.body.address_from;
    const to = req.body.address_to;
    const clas = req.body.clas;
    const pay_type = req.body.pay_type;
    let price = req.body.price;
    price = parseFloat(price.substring(0, price.length - 5));
    if (from === "" || to === "" || clas === "" || pay_type === "") {
        res.statusCode = 402;
        res.write(JSON.stringify({ "err": "not enough parameters" }));
        res.end();
        return;
    }
    db.getCon().query("SELECT * FROM orders WHERE user_id='{ }' AND (status=1 OR status=0)".format(userid),
function (err, result) {
    if (result.length >= config.max_car_amount_for_client) {
        res.json({ "too_much": true, "count": config.max_car_amount_for_client });
        res.end();
    } else {
        const sql = functions.getSQLCreateOrder(userid, from, to, clas, pay_type, req.body.notes, price);
        db.getCon().query(sql, function (err) {
            if (err !== null) {
                res.redirect("/orders/createorder");
            } else {
                res.statusCode = 201;
                res.json({ "too_much": false });
                res.end();
            }
        });
    }
});
});

router.post('/calcprice', function (req, res) {
    const clas = req.body.clas;
    const pay_type = req.body.pay_type;
    const km = req.body.km;
    if (clas === "" || clas === undefined || km === "" || km === undefined || pay_type === "" || pay_type === undefined)
    {
        res.statusCode = 401;
        res.end();
    }
});

```



```

    }
    const day_type = functions.getDayType();
    const sql = `SELECT * FROM tariffs WHERE class='{ }' AND pay_type='{ }'
        AND day_type='{ }';`.format(clas, pay_type, day_type);
    db.getCon().query(sql, function (err, result) {
        if (err) res.statusCode = 401;
        else {
            let price = result[0]['price'];
            price *= (parseFloat(km.substring(0, km.length - 3).replace(',', '.')));
            price += result[0]['min_price'];
            res.json({ "price": price.toFixed(2) });
        }
        res.end();
    });
});

router.get('/calculator', function (req, res) {
    text.header['nowpage'] = "nav_client";
    res.write(pug.renderFile(path.join(__dirname, "..", functions.getHeader(req.cookies.authorised)), text.header));
    db.getCon().query('SELECT * FROM payments', function (err, result) {
        res.write(pug.renderFile(path.join(__dirname, "../src/pugs/calculator.pug"), {
            "pay_types": result,
            "googlemapapi": config.googlemapapi
        }));
        res.end();
    });
});

router.get('/all', function (req, res) {
    const clas = functions.getCarModelIdLocal(req.cookies.userid);
    if (clas === null) {
        res.statusCode = 403;
        res.json({ "err": "no user founded" });
        res.end();
    } else {
        db.getCon().query("SELECT * FROM orders INNER JOIN payments ON payments.pay_id=orders.pay_type_id
WHERE status=0 AND class='{ }';".format(clas), function (err, result) {
            res.statusCode = 200;
            res.json({ "data": pug.renderFile(path.join(__dirname, "../src/pugs/allorders.pug"), { "orders": result } ) });
            res.end();
        });
    }
});

router.post('/cancelDrive', function (req, res) {
    console.log("cancel");
    db.getCon().query(functions.getSQLCancelDrive(req.body.order_id, req.cookies), function () {
        res.json({ "data": true });
        res.end();
    });
});

router.get('/my', function (req, res) {
    const user_type = req.cookies.authorised;
    const userid = req.cookies.userid;
    text.header['nowpage'] = "nav_mydrives";
    if (user_type === 'drivers') {
        res.write(pug.renderFile(path.join(__dirname + "../src/pugs/header-driver.pug"), text.header));
        db.getCon().query(functions.getSQLMyDrives(userid, user_type), function (err, result) {
            res.write(pug.renderFile(path.join(__dirname + "../src/pugs/my_orders.pug"), {
                "all_orders": result,
                "type": "driver"
            }));
        });
    }
});

```

```

        res.end();
    });
} else if (user_type === 'clients') {
    res.write(pug.renderFile(path.join(__dirname + "../src/pugs/header-client.pug"), text.header));
    db.getConnection().query(functions.getSQLMyDrives(userid, user_type), function (err, result) {
        res.write(pug.renderFile(path.join(__dirname + "../src/pugs/my_orders.pug"), {
            "all_orders": result,
            "type": "client"
        }));
        res.end();
    });
} else res.redirect("/404");
});
String.prototype.format = function () {
    var i = 0, args = arguments;
    return this.replace(/\{ \}/g, function () {
        return typeof args[i] !== 'undefined' ? args[i++] : "";
    });
};
module.exports = router;

```

functions.js

```

let LocalStorage = require('node-localstorage').LocalStorage, localStorage = new LocalStorage('./scratch');
let JSONStorage = require('node-localstorage').JSONStorage, jsonStorage = new JSONStorage('./scratch');
let nodemailer = require('nodemailer');
var md5 = require('md5');

```

```

module.exports = {
    //return header name depend on autorised
    getHeader: function (autorised) {
        if (autorised === 'drivers') {
            return "/src/pugs/header-driver.pug";
        } else if (autorised === 'clients') {
            return "/src/pugs/header-client.pug";
        } else {
            return "/src/pugs/header-unauthorized.pug";
        }
    },
    hashPassword: function (passStr) {
        let result = md5(passStr);
        return md5(result + 'taxiconn');
    },
    generateCode: function () {
        return Math.random().toString(36).slice(2);
    },
    //save code
    setCode: function (email, code) {
        localStorage.setItem(email, code);
    },
    //save driver info into local storage
    setRegisretDriverInfo: function (code, json) {
        json['password'] = this.hashPassword(json.password);
        jsonStorage.setItem(code, JSON.stringify(json));
    },
    getCarModelId: function (code) {
        let json = jsonStorage.getItem(code);
        if (json === "") return "";
        json = JSON.parse(json);
        return json.car_model;
    },
    getRegisretDriverSQL: function (email, code, carmodelid) {
        if (localStorage.getItem(email) === null || localStorage.getItem(email) !== code) {
            return "";
        }
    }
}

```

```

//get driver info from local storage
let json = jsonStorage.getItem(code);
let photo_src = "https://eu.ui-avatars.com/api/?name=" + json.name + "%20" + json.surname
  + "&background=343a40&color=ffc107&bold=true&size=512";
if (json === "") return "";
json = JSON.parse(json);
const userid = this.generateCode();
//add driver to db
let sql = `INSERT INTO drivers (id, login, name, surname, age, licence, carmodelid, caryear,
  password, phone, description, email, carnumber, photo_src) VALUES
  ('{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}', '{}')
  `.format(userid, json.login, json.name, json.surname, json.age, json.licence, carmodelid,
  json.car_year, json.password, json.phone, json.description, json.email, json.autonum, photo_src);
localStorage.removeItem(email);
localStorage.removeItem(code);
return sql;
},
send: async function (emailTo, code, config, email) {
  //send email for driver
  let transporter = nodemailer.createTransport({
    host: config.host,
    port: config.port,
    secure: false,
    auth: config.auth
  });
  //content of email
  let mailOptions = {
    from: config.auth.user,
    to: emailTo,
    subject: config.subject,
    html: "<div style='background-color: #343A40; width: 100%;>" +
      "  <b style='margin: 15px; color: white; font-size: x-large>Taxiconn</b>" +
      "  <h2 style='text-align: center'></h2></div>" +
      "<span style='text-align: center; align-items: center; color: black'><h1>" + email.header + "</h1><p>" +
email.text + "</p>" +
      "<div><a style='color: #E69C24' href='" + email.link + "/?email=" + emailTo + "'>" + email.linktext +
"</a></div>" +
      "<input style='width: 50%; margin: 7px 25%; text-align: center; padding: 5px; font-size: x-large; " +
      "background: white; border: none' disabled type='text' value='" + code + "' id='code'>" +
      "</span>"
  };
  transporter.sendMail(mailOptions, function (error, info) {
    if (error) {
      console.log(error);
      return false;
    } else {
      console.log('Email sent: ' + info.response);
      return true;
    }
  });
},
getDriverSQLUpdate: function (userid, data) {
  return `UPDATE drivers SET login='${}', name='${}', surname='${}', age=${data.age},
    carmodelid=${data.model}, caryear=${data.year}, phone='${}', description='${}' WHERE id='${}'
  `;
  .format(data.login, data.name, data.surname, data.phone, data.desc, userid);
},
getClientSQLUpdate: function (userid, data) {
  return `UPDATE drivers SET login='${}', name='${}', surname='${}', age='${}',
    phone='${}', description='${}' WHERE id='${}'
  `;
  .format(data.login, data.name, data.surname, data.age, data.phone, data.desc, userid);
},
setCarModelId: function (userid, result) {
  const clas = result[0].class;

```

```

    localStorage.setItem(userid, clas);
    return clas;
  },
  getCarModelIdLocal: function (userid) {
    return localStorage.getItem(userid);
  },
  getSQLLogin: function (login, type, password) {
    if (type === 'clients')
      return `SELECT * FROM clients WHERE password='${password}'
      AND (email='${login}' OR login='${login}' OR phone='${login}');`.format(password, login, login, login);
    else
      return `SELECT * FROM drivers WHERE password='${password}'
      AND (email='${login}' OR login='${login}' OR phone='${login}');`.format(password, login, login, login);
  },
  getSQLPassword: function (type, userid, old) {
    if (type === 'clients')
      return "SELECT * FROM clients WHERE password='${old}' AND id='${userid}";.format(old, userid);
    else
      return "SELECT * FROM drivers WHERE password='${old}' AND id='${userid}";.format(old, userid);
  },
  getSQLIsLoginFree: function (type, login) {
    if (type === 'clients')
      return "SELECT * FROM clients WHERE login='${login}";.format(login);
    else
      return "SELECT * FROM drivers WHERE login='${login}";.format(login);
  },
  getSQLCreateOrder: function (userid, from, to, clas, pay_type, notes, price) {
    if (notes === "") notes = 'NULL';
    else notes = "" + notes + "";
    //create dir_url
    const dirurl = "https://www.google.com/maps/dir/?api=1&origin=" + encodeURI(from) + "&destination=" +
      encodeURI(to) + "&travelmode=driving&dir_action=navigate";
    const code = this.generateCode();

    return `INSERT INTO orders
    (id, user_id, class, pay_type_id, comment, address_from, address_to, url, start_date, price)
    VALUES ('${code}', '${userid}', '${clas}', '${pay_type}', ${notes}, '${start_date}', '${price}', NOW(), '${code}');`.format(code, userid, clas, pay_type,
from, to, dirurl, price);
  },
  getSQLUploadPhoto: function (cookies) {
    //get photo src from storage and remove
    const text = localStorage.getItem("photo_src_" + cookies.userid);
    localStorage.removeItem("photo_src_" + cookies.userid);
    if (cookies.userid === "" || cookies.userid === undefined || cookies.authorised === "" || cookies.authorised ===
undefined) {
      return "";
    }
    //return sql text
    const userid = cookies.userid;
    if (cookies.authorised === 'clients')
      return "UPDATE clients SET photo_src='${text}' WHERE id='${userid}";.format(text, userid);
    else
      return "UPDATE drivers SET photo_src='${text}' WHERE id='${userid}";.format(text, userid);
  },
  savePhotoSrc: function (cookies, text, file_path) {
    localStorage.setItem("photo_src_" + cookies.userid, text);
    localStorage.setItem("photo_path_" + cookies.userid, file_path);
  },
  getPhotoUrlForDel: function (userid) {
    return localStorage.getItem("photo_path_" + userid);
  },
  getSQLProfileClient: function (userid) {
    return `SELECT *, COUNT(CASE WHEN orders.status<>0 AND orders.status<>1 THEN orders.id END) AS

```

```

total_amount,
    COUNT(CASE WHEN orders.status=4 THEN orders.id END) AS finished_amount,
    COUNT(CASE WHEN orders.status=2 THEN orders.id END) AS cancel_driver_amount,
    COUNT(CASE WHEN orders.status=3 THEN orders.id END) AS cancel_client_amount
FROM clients LEFT OUTER JOIN orders ON clients.id = orders.user_id WHERE clients.id='{}'
GROUP BY user_id`.format(userid);
},
getSQLProfileDriver: function (userid) {
    return `SELECT *, COUNT(CASE WHEN orders.status<>0 AND orders.status<>1 THEN orders.id END) AS
total_amount,
    COUNT(CASE WHEN orders.status=4 THEN orders.id END) AS finished_amount,
    COUNT(CASE WHEN orders.status=2 THEN orders.id END) AS cancel_driver_amount,
    COUNT(CASE WHEN orders.status=3 THEN orders.id END) AS cancel_client_amount
FROM drivers LEFT OUTER JOIN car_models ON car_models.id=drivers.carmodelid LEFT OUTER JOIN
orders
    ON drivers.id = orders.driver_id WHERE drivers.id='{}' GROUP BY driver_id`.format(userid);
},
getSQLEndDrive: function (order_id, driver_id) {
    return `UPDATE orders SET status=4, end_date=NOW() WHERE driver_id='{}' AND id='{}';`.format(driver_id,
order_id);
},
getSQLCancelDrive: function (order_id, cookies) {
    const user_id = cookies.userid;
    if (cookies.authorised === 'clients') {
        return `UPDATE orders SET status=3, end_date=NOW() WHERE id='{}' AND user_id='{}';`.format(order_id,
user_id);
    } else {
        return `UPDATE orders SET status=2, end_date=NOW() WHERE id='{}' AND
driver_id='{}';`.format(order_id, user_id);
    }
},
getSQLMyDrives: function (user_id, user_type) {
    if (user_type === 'clients') {
        return `SELECT *, TIMEDIFF(end_date, start_date) AS time_diff FROM orders INNER JOIN payments
            ON orders.pay_type_id = payments.pay_id WHERE user_id='{}';`.format(user_id);
    } else {
        return `SELECT *, TIMEDIFF(end_date, start_date) AS time_diff FROM orders INNER JOIN payments
            ON orders.pay_type_id = payments.pay_id WHERE driver_id='{}';`.format(user_id);
    }
},
getSQLRedisterUser: function (data) {
    //get all parameters
    const code = this.generateCode();
    const password = this.generateCode();
    const passHash = this.hashPassword(password);
    const email = data.email + data.emend;
    const photo_src = "https://eu.ui-avatars.com/api/?name=" + data.first_name + "%20" + data.surname
        + "&background=343a40&color=ffc107&bold=true&size=512";
    //save in local storage
    jsonStorage.setItem(data.login, JSON.stringify({
        "full_name": data.first_name + " " + data.surname,
        "password": password,
        "email": email
    }));
    if (data.description === "")
        data.description = 'NULL';
    return `INSERT INTO clients (id, login, name, surname, age, email, password, phone, description, photo_src)
VALUES
    ('{}', '{}', '{}', '{}', '{}', '{}', '{}', ${data.phone}, ${data.description}, '{}')`
        .format(code, data.login, data.first_name, data.surname, data.age, email, passHash, photo_src);
},
getSQLChageUserPassword: function (cookies, body) {
    //hasing

```

```

const oldHash = this.hashPassword(body.old_pass);
const newHash = this.hashPassword(body.new_pass);
return `UPDATE clients SET password='{', registered=1 WHERE password='{ }' AND id='{ }';`.format(newHash,
oldHash, cookies.unauthorised_userid);
},
sendUserMail: async function (login, config, email) {
  //sed amain with temporary password for user
  let data = jsonStorage.getItem(login);
  jsonStorage.removeItem(login);
  data = JSON.parse(data);
  let transporter = nodemailer.createTransport({
    host: config.host,
    port: config.port,
    secure: false,
    auth: config.auth
  });
  //email content
  let mailOptions = {
    from: config.auth.user,
    to: data.email,
    subject: config.subject,
    html: "<div style='background-color: #343A40; width: 100%'>" +
      "  <b style='margin: 15px; color: white; font-size: x-large'>Taxiconn</b>" +
      "  <h2 style='text-align: center'></h2></div>" +
      "<span style='text-align: center; align-items: center; color: black'><h2>" + email.header.format(data.name) +
"</h2><p>" + email.text + "</p>" +
      "<div style='font-weight: bold; font-size: 20px; color: red'>" + email.alert + "</div>" +
      "<input style='width: 50%; margin: 7px 25%; text-align: center; padding: 5px; font-size: x-large; " +
      "background: white; border: none' disabled type='text' value='" + data.password + "' id='code'>" +
      "</span>"
  };
  transporter.sendMail(mailOptions, function (error, info) {
    if (error) console.log(error);
    else console.log('Email sent: ' + info.response);
  });
},
getSQLIsAllFree: function (client, login, phone, email) {
  if (client) return "SELECT * FROM clients WHERE login='{ }' OR phone='{ }' OR email='{ }';".format(login,
phone, email);
  else return "SELECT * FROM drivers WHERE login='{ }' OR phone='{ }' OR email='{ }';".format(login, phone,
email);
},
getDayType: function () {
  const time = new Date().getHours();
  if (time === 23 || (time >= 0 && time <= 7)) return "night";
  if ((time >= 8 && time <= 10) || (time >= 17 && time <= 20)) return "peak";
  return "default";
}
}
String.prototype.format = function () {
  var i = 0, args = arguments;
  return this.replace(/{/g, function () {
    return typeof args[i] !== 'undefined' ? args[i++] : "";
  });
};
};

```

src -> css

main.min.css

```

:root{--myblack:#343m40;--yellow:#ffc107;--light-grey:#C3CAD2;--grey:#636970;--yellow-darker:#E69C24;--yellow-
dark:rgb(194, 132, 34);--light-yellow:rgba(255, 223, 127, 0.5)}body{overflow-x:hidden;width:100%;padding-
top:75px}.black{color:#000}.text-dark-red{color:#7f0000}.hover-light: hover{background-color:var(--light-
yellow);font-weight:500}.section-header h2{font-size:32px;color:var(--myblack);text-transform:uppercase;font-
weight:700;position:relative;padding-bottom:20px}.section-header

```

```

h2::before{content:"";position:absolute;display:block;width:50px;height:3px;background:#eee;bottom:0;left:0}.box{padding:20px;height:90%;margin-bottom:35px;box-shadow:10px 10px 15px rgba(73,78,92,.1);background:#fff;transition:.4s}.carousel-item{height:90vh}.box h4:hover{text-shadow:1px 1px 1px rgba(29,33,28,.37)}.myicon{width:5em;padding:.5em;float:left}.wrapper{width:70%!important;margin:auto!important}.wrapper ul{list-style:none}.wrapper ul li{width:55px;height:55px;line-height:55px;margin:0 10px;text-align:center;cursor:pointer;border-radius:50%;border:5px solid #d8e2dc;float:left;transition:all .5s ease}.card-header{margin-right:-1px;margin-left:-1px}.wrapper ul li .fa{color:#d8e2dc;width:30px;margin-top:0;padding-top:0;transition:all .5s ease}.wrapper ul li: hover.facebook{border:5px solid #3b5998;box-shadow:0 0 15px #3b5998;transition:all .5s ease}.wrapper ul li: hover .fa-facebook{color:#3b5998;text-shadow:0 0 15px #3b5998;transition:all .5s ease}.wrapper ul li: hover.twitter{border:5px solid #00aced;box-shadow:0 0 15px #00aced;transition:all .5s ease}.wrapper ul li: hover .fa-twitter{color:#00aced;text-shadow:0 0 15px #00aced;transition:all .5s ease}.wrapper ul li: hover.instagram{border:5px solid #bc2a8d;box-shadow:0 0 15px #bc2a8d;transition:all .5s ease}.wrapper ul li: hover .fa-instagram{color:#bc2a8d;text-shadow:0 0 15px #bc2a8d;transition:all .5s ease}.wrapper ul li: hover.google{border:5px solid #dd4b39;box-shadow:0 0 15px #dd4b39;transition:all .5s ease}.wrapper ul li: hover .fa-google{color:#dd4b39;text-shadow:0 0 15px #dd4b39;transition:all .5s ease}footer .copyright{text-align:center;width:100%!important;margin-bottom:5px;color:#fff!important}.cc{bottom:2px;color:#acacac;text-align:center;font-size:medium;align-items:center;align-self:center}@font-face{font-family:FontAwesome;src:local("FontAwesome"),url(../files/fonts/fontawesome-webfont.woff?v=4.7.0) format('woff2'),url(/files/fonts/fontawesome-webfont.woff?v=4.7.0) format('woff'),url(../files/fonts/fontawesome-webfont.ttf?v=4.7.0) format('truetype');font-weight:400;font-style:normal}.a{color:var(--yellow-darker)!important}.fa{display:inline-block;font:normal normal normal 14px/1 FontAwesome;text-rendering:auto;-webkit-font-smoothing:antialiased;-moz-osx-font-smoothing:grayscale;font-size:2em!important}.fa-twitter:before{content:"\f099"}.fa-instagram:before{content:"\f16d"}.fa-facebook:before{content:"\f09a"}html{height:100%}body{display:flex;flex-direction:column;min-height:100%}.content{flex:1 0 auto}footer{bottom:0;width:100%;flex:0 0 auto}footer nav{height:95px}.pt-header{padding:75px 0 0;margin:0;width:100%;position:initial}.contacts{margin:30px;font-size:larger}.numbers{font-size:large;position:absolute;top:120px;right:40px}.googlemap{height:300px;box-shadow:2px 2px 3px var(--grey)}.error404{margin:15px 25px;padding:10px}.pict-with-caption{width:100%;background:url(../files/images/taxipark.jpg) no-repeat;background-size:cover;height:90vh;margin-top:-15px}.semi-transparent{background-color:rgba(230,230,230,.38);width:100%;height:100%;text-align:center;padding:15%}.become{margin:20px}.become button{width:50%!important;font-size:xx-large;font-weight:bolder}.condition{margin:20px 7%;font-size:larger}.tip-img{width:25%;height:100%;margin:0 5%;padding:0;object-fit:cover;float:right}.rect{width:100%;height:100px;fill:var(--yellow-dark)}.input-group-prepend span{width:100%!important}.invalid-feedback p{margin:0!important;padding:0!important}.wrap{column-count:2}.prod{background-color:var(--yellow-dark);text-align:right;padding-right:15px}.p-10{padding:10px!important}.p-10 .text-sm-right{font-size:smaller;font-style:italic}.driver_auto{margin:0 25px;padding:10px 25px 25px}.main_profile_part{width:100%}.photo_container{width:100%;padding-top:100%;position:relative}.profile_img{object-fit:cover;position:absolute;top:0;left:0;bottom:0;right:0;width:100%;height:100%;border-radius:50%}.driver_auto input[disabled],.driver_auto select[disabled],.personal_info input[disabled]{background-color:#fff;margin:7px 0;border:1px solid #ced4da}.driver_auto input,.driver_auto select,.driver_auto textarea,.personal_info input{background-color:rgba(230,156,36,.3);margin:7px 0}.login{background:0 0;border:0!important;font-weight:bolder;font-size:xx-large;color:#000;font-family:Comfortaa,serif;text-transform:uppercase;margin-bottom:0!important;padding-bottom:0!important}.status{width:100%;background:0 0;border:0;font-family:Comfortaa,serif;font-size:medium}.year{width:10px!important}.ordercart{margin-top:10px;background:var(--yellow);height:45vh;float:right}.alert a{color:#000;text-decoration:underline}.alert a: hover{color:#000;text-decoration:none;font-weight:bolder}.bg-light-grey{background-color:var(--light-grey)!important}.bg-red{background-color:rgba(212,47,47,.6)!important}.bg_paytype_1{background-color:#d42f2f!important}.bg_paytype_2{background-color:green!important}.bg_paytype_3{background-color:var(--yellow)!important}.dropdown-item: hover{background-color:var(--light-grey)!important}.bg-grey{background-color:var(--grey)!important}.bg-warning-dark{background-color:var(--yellow-darker)!important;border-color:var(--yellow-darker)!important;font-weight:bolder!important}.bg-warning-dark2{background-color:var(--yellow-dark)!important;border-color:var(--yellow-dark)!important;font-weight:bolder!important}.card-header{padding:5px!important}.card-body{padding:10px!important;font-size:large}.card{margin-bottom:20px!important}.card-footer{padding:7px!important}.card-footer a,.ord_buttons a{margin-right:12px!important}.kbdord{color:var(--myblack)!important;background:#fff!important;font-size:medium!important}.comment_order{font-size:medium;color:var(--myblack)!important}.order_price{margin-left:10px;font-weight:bolder}.shadow-small{box-shadow:3px 3px 5px rgba(0,0,0,.65)}::-webkit-scrollbar{width:10px;background:0 0}::-webkit-scrollbar-track{background-color:var(--myblack)}::-webkit-scrollbar-thumb{background:var(--yellow)}.brad-0{border-radius:0!important}.inputfile{width:.1px;height:.1px;opacity:0;overflow:hidden;position:absolute;z-index:-1}.upload_photo_profile{border-radius:50%;width:100%;height:0;padding-top:100%;z-

```

```

index:1000;position:relative;margin-top:-
100% }.upload_photo_label{position:absolute;top:0;left:0;bottom:0;right:0;width:100%;height:100%;content:url(../fil
es/images/plus.svg);opacity:40%;border-radius:50% }.nav-link{color:#ffc107}.nav-
link:hover{color:#cc9500}#mycarousel{margin-top:-15px}.nav-link{color:#c8c8c8!important}.active_nav{font-
weight:bold;text-shadow:2px 2px 3px #7c7c7c}.badge{height:fit-
content}.profile_drives{width:100%;color:#000;text-decoration:none;padding:0
5px;cursor:pointer}.profile_drives:hover>span,b{text-decoration:underline!important}.h-fit-cont{height:fit-
content!important}.xlarge{font-size:x-large!important}.b{font-weight:700!important}.font-small{font-
size:medium}.down-arrow{font-size:26px;line-height:.5;font-weight:900;display:inline}@media (max-
width:576px){.carousel-caption{position:static!important}}

```

src -> js

main.js

```

//logining user function
function loginUser() {
    let passSelector = $("#log_pass");
    let login = $("#log_login").val();
    let pass = passSelector.val();
    let type = $("#log_type").children("option:selected").val();
    passSelector.val("");
    if (type === "Оберіть як ви хочете увійти..." || login === "" || pass === "") {
        alert("Всі поля мають бути заповнені");
        return false;
    }
    const person = {
        "login": login,
        "password": pass,
        "type": type
    };
    //create request for check
    $.ajax({
        url: '/login',
        type: 'post',
        dataType: 'json',
        contentType: 'application/json',
        success: function (data) {
            if (!data.registered && person.type === 'clients') {
                //if client try to login first time need to change password
                document.cookie = "unauthorised_userid=" + data.userid;
                document.cookie = "unauthorised=" + type;
                window.location = '/changepassword';
            } else {
                //set cookies and redirect to profile
                document.cookie = "userid=" + data.userid;
                document.cookie = "authorised=" + type;
                window.location = '/profile';
            }
        },
        error: function () {
            //if not such data in db
            alert("Перевірте правильність даних та спробуйте ще!");
            return false;
        },
        data: JSON.stringify(person)
    });
}

//removing cookies for exit user from system
function exit() {
    document.cookie = "userid=null;max-age=-1";
    document.cookie = "authorised=null;max-age=-1";
    window.location = '/';
}

```



```

//function for change list of models depend on producer
function renew_car_model(value) {
  let producerid = $("#select_carproducer").children("option:selected").val();
  let classSel = $('#carclassprofile');
  //if is a becomedriver page
  let car_class = classSel.children("option:selected").val();
  //if is a driver profile page
  if (car_class === undefined) {
    car_class = $('input[name="car_class"]:checked').val();
  }
  let prods = {
    "producer": producerid,
    "carclass": car_class,
    "secret_key": "OiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4"
  };
  let carModelSelector = $("#select_carmodel");
  carModelSelector.empty();
  $.ajax({
    url: '/carmodel',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function (data) {
      let res = data.res;
      carModelSelector.append("<option value='default' class='default_option'>Оберіть марку</option>");
      for (let i = 0; i < res.length; i++)
        carModelSelector.append("<option value='" + res[i].id + "'>" + res[i].model + "</option>");
      $("#select_carmodel option[value=" + value + "]).attr('selected', true);
    },
    data: JSON.stringify(prods)
  });
}

//submitting function on becomedriver
function loadSubmit() {
  const emailSelector = $("#driver_email");
  let loginSelector = $("#driver_login");
  const selectorsids = ["#select_category", "#select_carproducer", "#select_carmodel"];
  //check all field
  const isValid = [!validateName("#driver_name"), !validateName("#driver_surname"),
    !validateAge("#driver_age"), !validateTel("#driver_tel"), !validSeria("#driver_seria"),
    !validSeriaNum("#driver_seria_num"), !validPassword("#driver_pass", "#driver_pass_conf"),
    !validSelectors(selectorsids), !validAutoNum("driver_autonum"), !validEmptyAndSetKeyup(emailSelector),
    !validEmptyAndSetKeyup(loginSelector)];
  if (isValid.includes(true)) {
    event.preventDefault();
    return false;
  }
  //get needed values that should be unique
  let login = loginSelector.val();
  const email = emailSelector.val() + $("#select_email").val();
  let licence = $("#driver_seria").val() + $("#driver_seria_num").val();
  let phoneSel = $("#driver_tel");
  let phone = phoneSel.val();
  const autonum = $("#driver_autonum1").val() + $("#driver_autonum2").val() + $("#driver_autonum3").val();
  //check unique
  $.ajax({
    url: '/isAllFree',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function () {

```

```

loginSelector.addClass("is-valid");
loginSelector.removeClass("is-invalid");
//send main is all is correct
$.ajax({
    url: '/sendmail',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    accept: 'application/json',
    success: function (data) {
        if (data.res) {
            return true;
        } else {
            emailSelector.addClass("is-invalid");
            event.preventDefault();
            return false;
        }
    },
    data: JSON.stringify({
        "email": email,
        "name": $("#driver_name").val(),
        "surname": $("#driver_surname").val(),
        "age": $("#driver_age").val(),
        "login": login,
        "phone": phone,
        "description": $("#driver_desc").val(),
        "licence": licence,
        "car_producer": $("#select_carproducer").val(),
        "car_model": $("#select_carmodel").val(),
        "car_year": $("#car_year").val(),
        "password": $("#driver_pass").val(),
        "car_class": $("input[name="car_class"]:checked").val(),
        "autonum": autonum
    })
});
},
error: function (data) {
    if (data.login) {
        loginSelector.addClass("is-invalid");
        loginSelector.removeClass("is-valid");
    }
    if (data.email) {
        emailSelector.addClass("is-invalid");
        emailSelector.removeClass("is-valid");
    }
    if (data.phone) {
        phoneSel.addClass("is-invalid");
        phoneSel.removeClass("is-valid");
    } else {
        alert("Перевірте ваш логін, номер телефону та пошту на унікальність!");
    }
    event.preventDefault();
    return false;
},
data: JSON.stringify({
    "login": login,
    "number": phone,
    "email": email,
    "client": false
})
});
}

```

```

//listener for selector to create grey color for none values
function setValues() {
  let docs = document.getElementsByTagName("select");
  for (let i = 0; i < docs.length; i++) {
    docs[i].addEventListener("change", checkValueSelect);
    if (docs[i].value === "default") {
      docs[i].style.color = '#afb1b4';
    } else {
      docs[i].style.color = '#222222';
    }
  }
}

function checkValueSelect() {
  let options = this.value;
  if (options === "default") {
    this.style.color = '#afb1b4';
  } else {
    this.style.color = '#222222';
  }
}

//check neq login in registered pages and profiles
function checkLogin(logSelector, type) {
  let login = logSelector.val();
  if (login === "") {
    return false;
  }
  logSelector.keyup(function () {
    logSelector.removeClass("is-valid");
    logSelector.removeClass("is-invalid");
  });
  $.ajax({
    url: '/isLoginFree',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    accept: 'application/json',
    success: function (data) {
      if (data.free) {
        logSelector.addClass("is-valid");
        logSelector.removeClass("is-invalid");
      } else {
        logSelector.addClass("is-invalid");
        logSelector.removeClass("is-valid");
      }
    },
    data: JSON.stringify({
      "login": login,
      "type": type
    })
  });
}

async function checkFreeLoginClient() {
  let logSelector = $("#profile_login");
  if (logSelector.val() === JSON.parse(sessionStorage.getItem("profile"))) {
    logSelector.addClass("is-valid");
    logSelector.keyup(function () {
      logSelector.removeClass("is-valid");
      logSelector.removeClass("is-invalid");
    });
  }
  return;
}

```

```

    }
    return checkLogin(logSelector, "clients");
}

function checkRegisterFreeLogin() {
    let logSelector = $("#client_login");
    logSelector.keyup(function () {
        logSelector.removeClass("is-valid");
        logSelector.removeClass("is-invalid");
    });
    return checkLogin(logSelector, "clients");
}

async function checkFreeLogin() {
    let logSelector = $("#driver_login");
    if (logSelector.val() === undefined) {
        logSelector = $("#profile_login");
        if (logSelector.val() === JSON.parse(sessionStorage.getItem("profile")).login) {
            logSelector.addClass("is-valid");
            logSelector.keyup(function () {
                logSelector.removeClass("is-valid");
                logSelector.removeClass("is-invalid");
            });
            return true;
        }
    }
    return checkLogin(logSelector, "drivers");
}

//change passwor in profile
function changePass(usertype) {
    const oldPassSel = $("#old_change_pass");
    const passSel = $("#change_pass");
    const passSel2 = $("#change_pass2");
    let oldPass = oldPassSel.val();
    let pass = passSel.val();

    if (oldPass.length < 4) {
        oldPassSel.addClass("is-invalid");
        oldPassSel.click(function () {
            oldPassSel.removeClass('is-invalid');
        });
        return false;
    }
    if (!validPassword("#change_pass", "#change_pass2")) {
        return false;
    }
    const sended = {
        "old": oldPass,
        "now": pass,
        "type": usertype
    };
    //create request
    $.ajax({
        url: '/password',
        type: 'put',
        dataType: 'json',
        contentType: 'application/json',
        success: function () {
            //set correct alert
            $("#profilesection").prepend("<div class='alert alert-success alert-dismissible'>" +
                "<button type='button' class='close' data-dismiss='alert'>&times;</button>" +
                "<strong>Ваш пароль успішно змінено!</strong></div>");

```

```

        $("#cancelchangePass").click();

        oldPassSel.val("");
        passSel.val("");
        passSel2.val("");
        return true;
    },
    error: function (data) {
        if (data.err === 'uncorect old') {
            oldPassSel.addClass("is-invalid");
            oldPassSel.click(function () {
                oldPassSel.removeClass('is-invalid');
            });
            passSel2.val("");
        } else {
            alert(data.err);
            passSel2.val("");
        }
        return false;
    },
    data: JSON.stringify(sended)
});
}

//functions for make fielda enable for editing
function showEditingProfile(val) {
    if (val) {
        //for editing
        $(".driver_auto input[type=number]").prop("disabled", true);
        $(".driver_auto select").prop("disabled", true);
        $(".main_profile_part input").prop("disabled", true);
        $("#change_password").prop("disabled", false);
        $("#profile_desc").prop("disabled", true);
        $("#checkLogin").hide();
        const editSel = $("#editprofilebutton");
        editSel.show();
        editSel.prop("disabled", false);
        $("#saveprofilebutton").hide();
        $("#canselprofilebutton").hide();
        $("#upload_photo").hide();
    } else {
        //cancel editing
        $(".driver_auto input[type=number]").prop("disabled", false);
        $(".driver_auto select").prop("disabled", false);
        $(".main_profile_part input").prop("disabled", false);
        $("#profile_desc").prop("disabled", false);
        $("#profile_email").prop("disabled", true);
        $("#carnumber").prop("disabled", true);
        $("#change_password").prop("disabled", true);
        $("#editprofilebutton").hide();
        $("#checkLogin").show();
        $("#saveprofilebutton").show();
        $("#canselprofilebutton").show();
        $("#upload_photo").show();
    }
}

function editClientProfile() {
    let profileData = {
        "login": $("#profile_login").val(),
        "name": $("#profile_name").val(),
        "surname": $("#profile_surname").val(),
        "age": $("#profile_age").val(),
    }
}

```

```

        "phone": $("#profile_phone").val(),
        "desc": $("#profile_desc").val(),
        "photo_src": $("#photo").attr("src")
    };
    sessionStorage.setItem("profile", JSON.stringify(profileData));
    showEditingProfile(false);
}

function cancelClientChanges() {
    let arr = JSON.parse(sessionStorage.getItem("profile"));
    $("#profile_login").val(arr.login);
    $("#profile_name").val(arr.name);
    $("#profile_surname").val(arr.surname);
    $("#profile_age").val(arr.age);
    $("#profile_phone").val(arr.phone);
    $("#profile_desc").val(arr.desc);
    $("#photo").attr("src", arr.photo_src);
    showEditingProfile(true);
    sessionStorage.removeItem("profile");
}

function editDriverProfile() {
    let profileData = {
        "login": $("#profile_login").val(),
        "name": $("#profile_name").val(),
        "surname": $("#profile_surname").val(),
        "age": $("#profile_age").val(),
        "phone": $("#profile_phone").val(),
        "desc": $("#profile_desc").val(),
        "class": $("#carclassprofile").children("option:selected").val(),
        "producer": $("#select_carproducer").children("option:selected").val(),
        "model": $("#select_carmodel").children("option:selected").val(),
        "year": $("#profile_carid").val(),
        "photo_src": $("#photo").attr("src")
    };
    console.log(profileData);
    sessionStorage.setItem("profile", JSON.stringify(profileData));
    showEditingProfile(false);
}

function cancelDriverChanges() {
    let arr = JSON.parse(sessionStorage.getItem("profile"));
    $("#profile_login").val(arr.login);
    $("#profile_name").val(arr.name);
    $("#profile_surname").val(arr.surname);
    $("#profile_age").val(arr.age);
    $("#profile_phone").val(arr.phone);
    $("#profile_desc").val(arr.desc);
    $("#profile_carid").val(arr.year);
    showEditingProfile(true);
    $("#carclassprofile").val(arr.class);
    $("#select_carproducer").val(arr.producer);
    $("#photo").attr("src", arr.photo_src);
    renew_car_model(arr.model);
    sessionStorage.removeItem("profile");
}

function cancelClientChanges() {
    let arr = JSON.parse(sessionStorage.getItem("profile"));
    $("#profile_login").val(arr.login);
    $("#profile_name").val(arr.name);
    $("#profile_surname").val(arr.surname);
    $("#profile_age").val(arr.age);

```

```

$("#profile_phone").val(arr.phone);
$("#profile_desc").val(arr.desc);
showEditingProfile(true);
sessionStorage.removeItem("profile");
}

function saveDriverChanges() {
    const profileData = {
        "login": $("#profile_login").val(),
        "name": $("#profile_name").val(),
        "surname": $("#profile_surname").val(),
        "age": $("#profile_age").val(),
        "phone": $("#profile_phone").val(),
        "desc": $("#profile_desc").val(),
        "class": $("#carclassprofile").children("option:selected").val(),
        "producer": $("#select_carproducer").children("option:selected").val(),
        "model": $("#select_carmodel").children("option:selected").val(),
        "year": $("#profile_carid").val(),
    };
    $.ajax({
        url: '/profile',
        type: 'put',
        contentType: 'application/json',
        accept: '*/*',
        success: function () {
            showEditingProfile(true);
            sessionStorage.removeItem("profile");
        },
        error: function () {
            cancelDriverChanges();
            alert("<b>На сервері виникла якась помилка. Ваші дані не було змінено!</b><br/><span>" +
                "Можливо номер телефону який ви намагалися вказати вже вказаний у іншого водія</span>");
            sessionStorage.removeItem("profile");
        },
        data: JSON.stringify(profileData)
    });
}

function saveClientChanges() {
    const profileData = {
        "login": $("#profile_login").val(),
        "name": $("#profile_name").val(),
        "surname": $("#profile_surname").val(),
        "age": $("#profile_age").val(),
        "phone": $("#profile_phone").val(),
        "desc": $("#profile_desc").val()
    };
    $.ajax({
        url: '/profile',
        type: 'put',
        dataType: 'json',
        contentType: 'application/json',
        success: function () {
            showEditingProfile(true);
            sessionStorage.removeItem("profile");
        },
        error: function () {
            cancelClientChanges();
            console.log("На сервері виникла якась помилка. Ваші дані не було змінено");
            sessionStorage.removeItem("profile");
        },
        data: JSON.stringify(profileData)
    });
}

```

```

}

//submit on ordertaxi
function ordertaxi() {
  const address_from = $("#address_from").val();
  const address_to = $("#address_to").val();
  if (address_to === "" || address_from === "") {
    alert("Всі поля мають бути заповнені!");
    return;
  }
  const data = {
    "address_from": address_from,
    "address_to": address_to,
    "pay_type": $("#pay_type").val(),
    "clas": $("input[name='class']:checked").val(),
    "notes": $("#notes").val(),
    "price": $("#km_val").val()
  };
  $.ajax({
    url: '/orders/createorder',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function (data2) {
      if (data2.too_much) {
        //alert in more than 2 active drives
        $("#foralert").prepend("<div class='alert alert-danger alert-dismissible'>" +
          "<button type='button' class='close' data-dismiss='alert'>&times;</button>" +
          "<strong>Перевищено ліміт!</strong> Ви вже маєте " + data2.count +
          " активні (незавершені) поїздки. Для нового замовлення спершу завершіть або скасуйте їх. " +
          "</div>");
      } else
        window.location = "/thankspage";
    },
    error: function (data2) {
      //server error
      $("#foralert").prepend("<div class='alert alert-danger alert-dismissible'>" +
        "<button type='button' class='close' data-dismiss='alert'>&times;</button>" +
        "<strong>Помилка!</strong> Чомусь не вдалося обробити вашу заявку. <a href='#" +
        onclick='ordertaxi()'>Спробуйте ще!</a>" +
        "</div>");
      console.log(data2.err);
    },
    data: JSON.stringify(data)
  });
}

function getOrder(id) {
  $.ajax({
    url: '/acceptOrder',
    type: 'post',
    success: function () {
      window.location = '/orders/my';
    },
    error: function () {
      $("#order_modal_alert").modal("toggle");
    },
    data: {"orderid": id}
  });
}

```

//functions for renew orders on deivers orders page


```

async function renew_orders_method() {
  $.ajax({
    url: '/orders/all',
    type: 'get',
    dataType: 'json',
    contentType: 'application/json',
    success: function (data) {
      $("#orders").html(data.data);
    },
    error: function (data) {
      console.log(data.err);
    }
  });
}

function setFooter() {
  $.ajax({
    url: '/getfooter',
    type: 'get',
    dataType: 'text',
    contentType: 'text',
    success: function (data) {
      $("#for_footer").html(data);
    },
    error: function (data) {
      console.log(data.err);
    }
  });
}

function sendNewProfilePhoto() {
  var formData = new FormData($("#upload_photo")[0]);
  $.ajax({
    url: '/upload_user_photo',
    type: "POST",
    data: formData,
    success: function (data) {
      console.log(data);
      if (!data.empty) {
        document.getElementById("photo").setAttribute("src", data.src);
      }
    },
    error: function () {
      alert('Sorry something happened on server');
    },
    contentType: false,
    processData: false
  });
}

function endDrive(order_id) {
  $.ajax({
    url: '/orders/endDrive',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function () {
      window.location = '/orders/my';
    },
    data: JSON.stringify({"order_id": order_id})
  });
}

```

```

function cancelDrive(order_id) {
  $.ajax({
    url: '/orders/cancelDrive',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function () {
      window.location = '/orders/my';
    },
    data: JSON.stringify({"order_id": order_id})
  });
}

function showRegisterModal() {
  $('#register').modal('toggle');
  $('#login').modal('toggle');
}

function firstChangePass() {
  if (!validPassword('#new_pass', '#new_pass2')) {
    return;
  }
  $.ajax({
    url: '/changepass',
    type: 'post',
    dataType: 'json',
    contentType: 'application/json',
    success: function (data) {
      if (!data.data) {
        $('#uncorrect_alert').style("display", "block");
        $('#new_pass2').val("");
        $('#old_pass').val("");
      } else {
        document.cookie = 'unauthorised_userid=null;max-age=-1';
        document.cookie = 'unauthorised=null;max-age=-1';
        window.location = '/aboutus';
      }
    },
    data: JSON.stringify({
      "old_pass": $('#old_pass').val(),
      "new_pass": $('#new_pass').val()
    })
  });
}

function registerUser() {
  const emailSelector = $("#client_email");
  let loginSelector = $("#client_login");
  const isValid = [!validateName("#client_name"), !validateName("#client_surname"),
    !validateAge("#client_age"), !validateTel("#client_tel"),
    !validEmptyAndSetKeyup(emailSelector), !validEmptyAndSetKeyup(loginSelector)];
  console.log(isValid);
  if (isValid.includes(true)) {
    event.preventDefault();
    return false;
  }
  let login = loginSelector.val();
  const email = emailSelector.val() + $("#select_email").val();
  let phoneSel = $("#client_tel");
  let phone = phoneSel.val();
  $.ajax({
    url: '/isAllFree',
    type: 'post',

```

```

dataType: 'json',
contentType: 'application/json',
success: function () {
    loginSelector.addClass("is-valid");
    loginSelector.removeClass("is-invalid");
},
error: function (data) {
    if (data.login) {
        loginSelector.addClass("is-invalid");
        loginSelector.removeClass("is-valid");
    }
    if (data.email) {
        emailSelector.addClass("is-invalid");
        emailSelector.removeClass("is-valid");
    }
    if (data.phone) {
        phoneSel.addClass("is-invalid");
        phoneSel.removeClass("is-valid");
    } else {
        alert("Перевірте ваш логін, номер телефону та пошту на унікальність!");
    }
    event.preventDefault();
    return false;
},
data: JSON.stringify({
    "login": login,
    "number": phone,
    "email": email,
    "client": true
}))
});
}

```

```

function calcPrice() {
    const d = sessionStorage.getItem("distance");
    if (d === null) return;
    $.ajax({
        url: '/orders/calprice',
        type: 'post',
        dataType: 'json',
        contentType: 'application/json',
        success: function (data) {
            document.getElementById('km_val').value = data.price + " грн.";
        },
        data: JSON.stringify({
            "km": d,
            "pay_type": $("#pay_type").val(),
            "clas": $("input[name='class']:checked").val(),
        })
    });
}

```

maps.min.js

```

let main_zoom=15;function myMap(){let e={center:new
google.maps.LatLng(50.46000699004913,30.52081012508188),zoom:15,mapTypeControl:!1,draggable:!1,scaleContro
l:!1},t=new google.maps.Map(document.getElementById("googleMap"),e),o=new
google.maps.LatLng(50.46000699004913,30.52081012508188),n=new
google.maps.Marker({position:o,animation:google.maps.Animation.BOUNCE,icon:"../images/map.png"});n.setMap(t),
google.maps.event.addListener(n,"click",function(){15===main_zoom?(t.setZoom(9),main_zoom=9):(t.setZoom(15),m
ain_zoom=15),t.setCenter(n.getPosition())})}function initAutocomplete(){new AutocompleteDirectionsHandler(new
google.maps.Map(document.getElementById("map"),{mapTypeControl:!1,center:{lat:50.46000699004913,lng:30.5208
1012508188},zoom:13}))}function
AutocompleteDirectionsHandler(e){this.map=e,this.originPlaceId=null,this.destinationPlaceId=null,this.travelMode="
DRIVING",this.directionsService=new google.maps.DirectionsService,this.directionsRenderer=new

```

```

google.maps.DirectionsRenderer,this.directionsRenderer.setMap(e);var
t=document.getElementById("address_from"),o=document.getElementById("address_to"),n=new
google.maps.places.Autocomplete(t);n.setComponentRestrictions({country:["ua"]}),n.setFields(["place_id"]);var i=new
google.maps.places.Autocomplete(o);i.setComponentRestrictions({country:["ua"]}),i.setFields(["place_id"]),this.setupP
laceChangeListener(n,"ORIG"),this.setupPlaceChangeListener(i,"DEST"))AutocompleteDirectionsHandler.prototype
e.setupPlaceChangeListener=function(e,t){var
o=this;e.bindTo("bounds",this.map),e.addListener("place_changed",function(){var
n=e.getPlace();"ORIG"===t?o.originPlaceId=n.place_id:o.destinationPlaceId=n.place_id,o.route()}}),AutocompleteDir
ectionsHandler.prototype.route=function(){if(this.originPlaceId&&this.destinationPlaceId){var
e=this;this.directionsService.route({origin:{placeId:this.originPlaceId},destination:{placeId:this.destinationPlaceId},tra
velMode:this.travelMode},function(t,o){if("OK"===o){e.directionsRenderer.setDirections(t);var
n=t.routes[0].legs[0].distance.text;sessionStorage.setItem("distance",n),calcPrice()}else window.alert("Directions
request failed due to "+o)}});

```

validator.min.js

```

function validName(i){return/^[\a-zA-ZА-Яа-яІ-і]+$/i.test(i)}function validateName(i){let
a=$(i).val();return""!==a&&validName(a)?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid"),$(i).keyup(function(){ $(i).removeClass("is-invalid"),!0:( $(i).removeClass("is-invalid"),$(i).addClass("is-
invalid"),$(i).keyup(function(){let a=$(i).val();""!==a&&validName(a)?( $(i).removeClass("is-
invalid"),$(i).addClass("is-valid")):( $(i).addClass("is-invalid")),!1})function validTel(i){return/^[\+]?([0-9]{3})?[-
\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/i.test(i)}function validateTel(i){let
a=$(i).val();return""!==a&&validTel(a)?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid"),$(i).keyup(function(){ $(i).removeClass("is-invalid"),!0:( $(i).removeClass("is-invalid"),$(i).addClass("is-
invalid"),$(i).keyup(function(){let a=$(i).val();a.length>0&&a.length<10&&( $(i).removeClass("is-
valid"),$(i).removeClass("is-invalid"),""!==a&&validTel(a)?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid")):( $(i).addClass("is-invalid")),!1})function validSeria(i){let a=$(i).val();return
3===a.length&&validateName(a)?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid"),$(i).keyup(function(){ $(i).removeClass("is-invalid"),!0:( $(i).removeClass("is-invalid"),$(i).addClass("is-
invalid"),$(i).keyup(function(){let a=$(i).val();3===a.length||validateName(a)?( $(i).removeClass("is-
invalid"),$(i).addClass("is-valid")):( $(i).addClass("is-invalid")),!1})function
validateSeriaNum(i){return!(i.length<5||i.length>10)&& /^[0-9]+$/.test(i)}function validSeriaNum(i){return
validateSeriaNum($(i).val())?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid"),$(i).keyup(function(){ $(i).removeClass("is-invalid"),!0:( $(i).removeClass("is-invalid"),$(i).addClass("is-
invalid"),$(i).keyup(function(){ validateSeriaNum($(i).val())?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid")):( $(i).addClass("is-invalid")),!1})function validatePass(i){return!(i.length<6)&&(!/[0-9]+$/.test(i)&&!/^[a-zA-
ZA-Яa-яІ-і]+$/i.test(i)&&!(i===i.toLowerCase()||i===i.toUpperCase()))}function validPassword(i,a){let
l=$(i).val();return validatePass(l)?$(a).val()===l||( $(a).addClass("is-
invalid"),$(a).keyup(function(){ $(a).removeClass("is-invalid"),$(a).val(""),!1):( $(i).addClass("is-
invalid"),$(i).keyup(function(){ validatePass($(i).val())?( $(i).removeClass("is-invalid"),$(i).addClass("is-
valid")):( $(i).addClass("is-invalid")),$(a).val(""),!1})function validAge(i){let
a=$(i).val();return""===a||a<21||a>65?( $(i).addClass("is-invalid"),$(i).click(function(){ $(i).removeClass("is-
invalid")),!1):( $(i).removeClass("is-invalid"),$(i).addClass("is-valid"),$(i).click(function(){ $(i).removeClass("is-
invalid")),!0})function validSelectors(i){for(let a=0;a<i.length;a++){let l=i[a];if("default"=== $(l).val())return
$(l).addClass("is-invalid"),$(l).click(function(){ $(l).removeClass("is-invalid")),!1}return!0}function
validAutoNum(i){const a=$("#"+i+"2");let l=a.val();const
s=validAutoLetter($("#"+i+"1"),e=validAutoLetter($("#"+i+"3"));return""!==l&&4===l.length&& /^[0-
9]+$/.test(l)?s&&e:(a.addClass("is-invalid"),a.click(function(){a.removeClass("is-invalid")),!1})function
validAutoLetter(i){const a=i.val();return""!==a&&2===a.length&& /^[a-zA-ZА-Яа-яІ-і]+$/i.test(a)?(i.addClass("is-
valid"),i.click(function(){i.removeClass("is-valid")),!0:(i.addClass("is-invalid"),i.click(function(){i.removeClass("is-
invalid")),!1})function validEmptyAndSetKeyup(i){return i.keyup(function(){emailSelector.removeClass("is-
valid"),emailSelector.removeClass("is-invalid")),""!==i.val()||(i.addClass("is-invalid"),!1)}

```

src -> html

404.html

```
<section class="error404 content"><p class="display-1"><b>404</b></p>
```

```
<p>Сторінка на яку ви намагаєтесь перейти недоступна або вимагає іншого рівня авторизації, перепрошуємо
за
```

```
незручності</p>
```

```
<p>Переірте правильність вашого посилання або скористайтесь навігацією на сайті, щоб знайти необхвдну
вам сторінку.
```

```
Або увійдіть за допомогою кнопки у верхньому правому куті екрану</p>
```

```
<form action="/" method="get">
```

```
<button class="btn-warning btn btn-lg" type="submit">На головну</button>
```

```
</form>
```

```
</section>
```

```
<div id="for_footer">
```

```
</div>
```

```
<script type="text/javascript"> setFooter();</script>
```

aboutus.html

```
<div class="content">
```

```
<div class="carousel slide pointer-event" id="mycarousel" data-ride="carousel">
```

```
<ol class="carousel-indicators">
```

```
<li class="" data-target="#mycarousel" data-slide-to="0"></li>
```

```
<li data-target="#mycarousel" data-slide-to="1" class=""></li>
```

```
<li data-target="#mycarousel" data-slide-to="2" class="active"></li>
```

```
</ol>
```

```
<div class="carousel-inner" style="height: 95vh">
```

```
<div class="carousel-item">
```

```
<div class="carousel-caption d-md-block jumbotron"><h2 class="black">Найкраще таксі у Києво-могилянській
```

```
академії</h2>
```

```
<p class="black">Використовуйте наше таксі для зручної, швидкої та недорогої дороги від дому до університету та навпаки.</p></div>
```

```
</div>
```

```
<div class="carousel-item">
```

```
<div class="carousel-caption d-md-block jumbotron"><h2 class="black">Безпека - понад усе</h2>
```

```
<p class="black">Ми надзвичайно стурбовані тим, щоб наші користувачі не потрапляли в ДТП, тому всі
```

```
поїздки виключно віртуальні!<span class="black"> Віртуальність гарантує вам фізичну безпеку та запобігає отриманню неприємного досвіду в поїздах.</span>
```

```
</p></div>
```

```
</div>
```

```
<div class="carousel-item active">
```

```
<div class="carousel-caption d-md-block jumbotron"><h3 class="black">Цінність у кожному клієнтові та водієві!!!</h3>
```

```
<p class="black">Ми цінуємо кожного, хто приєднується до нашого сервісу та створюємо комфортне середовище для роботи і користування</p></div>
```

```
</div>
```

```
</div>
```

```
<a class="carousel-control-prev" href="#mycarousel" role="button" data-slide="prev"><span class="carousel-control-prev-icon" aria-hidden="true"></span><span class="sr-
```

```
only">Previous</span></a><a
```

```
class="carousel-control-next" href="#mycarousel" role="button" data-slide="next"><span
```

```
class="carousel-control-next-icon" aria-hidden="true"></span><span class="sr-only">Next</span></a></div>
```

```
<section class="mt-5">
```

```
<div class="container">
```

```
<div class="section-header"><h2>Переваги використання</h2>
```

```
<p></p></div>
```

```
<div class="row">
```

```
<div class="col-lg-4">
```

```
<div class="box"><h4>Відображення
```

```
заявок</h4>
```

```
<p>Зручно замовляйте таксі та переглядайте історію ваших поїздок в будь-який час.</p></div>
```

```
</div>
```

```
<div class="col-lg-4">
```

```
<div class="box"><h4>Доступні скрізь</h4>
```

```
<p>Замовляйте з будь-якої точки світу в будь-якому напрямку</p></div>
```

```
</div>
```

```
<div class="col-lg-4">
```

```
<div class="box"><h4>Без
```

```

        очікувань</h4>
        <p>Мінімальний час очікування на опрацювання запитів, обробка запитів за лічені секунди!</p>
    </div>
</div>
</div>
</div>
</section>
<div class="container"><h1><kbd>Цікаво знати</kbd></h1><br>
    <p style="font-size: large">Таксі — засіб громадського транспорту, зазвичай легковий або вантажний
автомобіль,
        використовується для перевезення пасажирів і вантажів у будь-яке вказане місце з оплатою проїзду
машини за
        такою за лічильником — таксометром.</p>
    <p style="font-size: large">Згідно з даними інтернет-порталу Prostoturist, станом на березень 2015 року
середній
        тариф на таксі Києва становить 6 гривень за 1 кілометр. За основу бралася вартість поїздки з центру
міста в
        спальний район. Протяжність маршруту склала 13 кілометрів.</p></div>
<div class="card" style="margin: 30px; padding: 10px">
    <div class="card-body row">
        <div class="col-12 col-sm-5">
            
        </div>
        <div class="col-12 col-sm-7">
            <h2 class="card-title">Думка розробника</h2>
            <hr style="left: 2.5%; position: absolute; width: 55%;">
            <br>
            <p class="card-text" style="font-size: large">Я вважаю, що користування таким сервісом має приносити і
                водіям
                і клієнтам лише задоволення. Особисто я була надзвичайно захоплена його розробкою, сайт
створений з
                чітко
                продуманою структурою, щоб користування було приємним та легким, тому сподіваюсь, що він
знайде
                відгук в
                кожному серці.</p>
            <p class="card-text float-right text-secondary">P.S. Час плинний - Могилянка вічна</p>
        </div>
    </div>
</div>
</div>
<div id="for_footer">
</div>
<script type="text/javascript"> setFooter();</script>

thanks.html
<section class="condition content mt-1"><p class="display-4"><b>Вашу заявку успішно прийнято!</b></p>
    <p>Очікуйте, ми шукаємо для вас водія.</p>
    <p>Щойно водій прийме вашу заявку ви отримаєте лист на електронну пошту з детальною інформацією про
вашого водія та
        поїздки.<span>Переглянути ваші поїздки (включаючи поточну) ви можете за кнопкою нижче</span></p>
    <button class="btn-warning btn btn-lg d-inline-block" onclick="window.location='/orders/my';">Переглянути мої
        поїздки
    </button>
    <button class="btn btn-outline-warning btn-lg d-inline-block" onclick="window.location='/';">На головну
    </button>
</section>

<div id="for_footer">
</div>
<script type="text/javascript"> setFooter();</script>

usefulltips.html

```

```

<section class="condition content"><h2><kbd>Корисні поради водіям</kbd><b><i> taxiconn.com.ua</i></b></h2>
  <p>Щоб мати рекомендації серед користувачів, а також отримувати можливі додаткові чайові від клієнтів
  можете
    скористатися нашими порадами, що нададуть вам змогу покращити ваш сервіс</p>
  <p>Також в цьому розділі ви можете знайти лайфхаки, що полегшать вашу роботу та зроблять її ще
  приємнішою для вас та
    вашого автомобіля, що є важливою складовою наших працівників</p>
  <hr>
  <div><h4>Автомобіль</h4>
    <ul>
      <li>Слідкуйте за чистотою вашого авто</li>
      <li>Підтримуйте чистоту в салоні та не тримайте в салоні зайвих речей</li>
      <li>Регулярно проходите ТехОгляд</li>
      <li>Краще, щоб в салоні не було зайвих запахів:<i> цигарки, ароматизатори, кава...</i></li>
    </ul>
  </div>
  <hr>
  <div><h4>
  Водій</h4>
    <ul>
      <li>Будьте охайними:<i> чистий одяг, відсутність неприємних запахів...</i></li>
      <li>Не запізнюйтесь, а приїхавши заздалегідь не надзвонюйте клієнтові</li>
      <li>Користуйтеся навігатором<i> (навіть якщо знаєте короткий шлях)</i></li>
      <li>Дотримуйтесь<b> Правил Дорожнього Руху</b></li>
    </ul>
  </div>
  <hr>
  <div><h4>Пасажир</h4>
    <ul>
      <li>Обов'язково вітайтеся та прощайтесь з клієнтом</li>
      <li>Допомагайте грузити особисті речі до багажника</li>
      <li>Дотримуйтесь<i> правил етикету</i></li>
      <li>Завжди відповідайте на питання користувачів, але не втомлюйте бесідою</li>
    </ul>
  </div>
</section>

<div id="for_footer">
</div>
<script type="text/javascript"> setFooter();</script>

```

src -> pugs

allorders.pug

```

each order in orders
  div.card.brad-0.shadow-small
    div.card-header.bg-light-grey.brad-0
      span.order_price #{order.price} ₪
      span.float-right
        kbd(class='bg_paytype_' + order.pay_type_id) #{order.name}
    div.card-body
      b Місце посадки:
      span #{order.address_from}
      br
      b Місце висадки:
      span #{order.address_to}
      if order.comment !== null
        br
        span.comment_order
          b Коментар:
          span #{order.comment}
    div.mt-2.ord_buttons
      a.btn.btn-warning.bg-warning-dark(href='/user/' + order.user_id) Переглянути профіль клієнта
      a.btn.btn-dark(href='#' + order.url}' target='blank') Відкрити маршрут

```

```
div.card-footer
  button.btn.btn-warning.float-right(onclick="getOrder('{ order.id }')") Взяти поїздку
```

becomeclient.pug

```
div.container.content(id='becomedriverpage')
  div.row
    div.col-12.col-sm-12.mb-3
      h2
        kbd Швидка реєстрація
        i taxiconn.com.ua
      form.col-12.row.needs-validation(novalidate action='/registered' method='get' onSubmit='return registerUser()')
        div.input-group.col-sm-10.col-12.mb-2
          div.input-group-prepend
            span.input-group-text Ім'я
            input.form-control(required type="text" placeholder="Ім'я" maxlength=45 id='client_name'
name='first_name')
            span.invalid-feedback Ім'я не повинно містити чисел чи символів

          div.input-group-prepend
            span.input-group-text Прізвище
            input.form-control(required type="text" placeholder="Прізвище" maxlength=45 id='client_surname'
name='surname')
            span.invalid-feedback Прізвище не повинно містити чисел чи символів
          div.input-group.col-sm-10.col-12.mb-2
            div.input-group-prepend
              span.input-group-text Логін
              input.form-control(required type="text" placeholder="Логін" maxlength=20 id='client_login' name='login')
            div.input-group-append
              input.btn.btn-outline-warning(type='button' data-toggle="tooltip" title="Перевірте чи вільний ваш логін"
value='Перевірити' onclick="checkRegisterFreeLogin()")
              span.invalid-feedback Ваш логін зайнятий! Оберіть новий
          div.input-group.col-sm-10.col-12.mb-2
            div.input-group-prepend
              span.input-group-text Email
              input.form-control(required name='email' type="text" placeholder="Електронна пошта" maxlength=45
id='client_email')

          select.btn-outline-warning.input-group-append.form-control(name='emend' id='select_email')
            option.form-control.btn-outline-info @gmail.com
            option.form-control.btn-outline-info @ukma.edu.ua
            option.form-control.btn-outline-info @ukr.net
            option.form-control.btn-outline-info @mail.com
          span.invalid-feedback Перевірте вашу пошту
        div.col-sm-10.col-12.mb-2
          div.row.input-group
            div.input-group.col-12.col-sm-6
              div.input-group-prepend
                span.input-group-text Вік
                input.form-control(required type="number" min=16 id='client_age' name='age')
            div.col.col-sm.pr-0(style="margin-right: -15px")
              div.input-group
                div.input-group-prepend
                  span.input-group-text Телефон
                  input.form-control(required type="tel" placeholder="" id='client_tel' name='phone')
                span.invalid-feedback Номер телефону повинен відповідати одному з шаблонів:
                p (123) 456-7890
                p (123)456-7890
                p 123-456-7890
                p 123.456.7890
                p 1234567890
                p +31636363634
                p 075-63546725
            div.col-sm-10.col-12.mb-2
```



```

        textarea.form-control(id='client_desc' name='description' rows=3 placeholder="Опишіть себе (за бажанням)...")
    div.col-sm-10.col-12
        div.input-group.align-content-center.mb-0(style='width: 100%')
            button.btn.btn-md.btn-warning(style='margin: auto; width: 50%' type='submit') Зареєструватись
include footer
becomedriver.pug
div.container.content(id='becomedriverpage')
    div.row
        div.col-12.col-sm-12
            h2
                kbd Стати водієм
                i taxiconn.com.ua
            hr
            form.form-row.needs-validation(novalidate action='/confirmregistration' method='get' onSubmit='return loadSubmit()')
                h3.col-12.col-sm-12 Особиста інформація
                div.input-group.col-sm-10.col-12.mb-2
                    div.input-group-prepend
                        span.input-group-text Ім'я
                        input.form-control(required type="text" placeholder="Ім'я" maxlength=45 id='driver_name')
                        span.invalid-feedback Ім'я не повинно містити чисел чи символів
                div.input-group.col-sm-10.col-12.mb-2
                    div.input-group-prepend
                        span.input-group-text Прізвище
                        input.form-control(required type="text" placeholder="Прізвище" maxlength=45 id='driver_surname')
                        span.invalid-feedback Прізвище не повинно містити чисел чи символів
                div.input-group.col-sm-10.col-12.mb-2
                    div.input-group-prepend
                        span.input-group-text Логін
                        input.form-control(required type="text" placeholder="Логін" maxlength=20 id='driver_login')
                    div.input-group-append
                        input.btn.btn-outline-warning(type='button' data-toggle="tooltip" title="Перевірте чи вільний ваш логін" value='Перевірити' onclick="checkFreeLogin()")
                        span.invalid-feedback Ваш логін зайнятий! Оберіть новий
                div.input-group.col-sm-10.col-12.mb-2
                    div.input-group-prepend
                        span.input-group-text Email
                        input.form-control(required name='email' type="text" placeholder="Електронна пошта" maxlength=45 id='driver_email')

                select.btn-outline-warning.input-group-append.form-control(name='emend' id='select_email')
                    option.form-control.btn-outline-info @gmail.com
                    option.form-control.btn-outline-info @ukma.edu.ua
                    option.form-control.btn-outline-info @ukr.net
                    option.form-control.btn-outline-info @mail.com
                span.invalid-feedback Перевірте вашу пошту
            div.col-sm-10.col-12.mb-2
                div.row.input-group
                    div.input-group.col-12.col-sm-6
                        div.input-group-prepend
                            span.input-group-text Бік
                            input.form-control(required type="number" max=65 min=21 id='driver_age')
                    div.col.col-sm.pr-0(style="margin-right: -15px")
                        div.input-group
                            div.input-group-prepend
                                span.input-group-text Телефон
                                input.form-control(required type="tel" placeholder="" id='driver_tel')
                                span.invalid-feedback Номер телефону повинен відповідати одному з шаблонів:
                                    p (123) 456-7890
                                    p (123)456-7890
                                    p 123-456-7890
                                    p 123.456.7890

```

```

        p 1234567890
        p +31636363634
        p 075-63546725
div.col-sm-10.col-12.mb-2
    textarea.form-control(id='driver_desc' rows=3 placeholder="Опишіть себе (за бажанням)...")

h3.col-12.col-sm-12 Водійське посвідчення та авто
div.col-sm-10.col-12.mb-2
    div.input-group
        div.input-group-prepend
            span.input-group-text Серія та номер
            input.form-control(required type="text" placeholder="BAH" maxlength=3 id='driver_seria')
            input.form-control(required type="text" placeholder="000000" id="driver_seria_num")
div.col-sm-10.col-12.mb-2
    div.row.input-group
        div.input-group.col-12.col-sm-6
            div.input-group-prepend
                span.input-group-text Категорія
                select.input-group-append.form-control(id='select_category')
                    option.default_option(value="default") Оберіть...
                    option.form-control.btm-outline-info B
                    option.form-control.btm-outline-info B1
                    option.form-control.btm-outline-info C
                    option.form-control.btm-outline-info C1
                    option.form-control.btm-outline-info D
                    option.form-control.btm-outline-info D1
                    option.form-control.btm-outline-info BE
                    option.form-control.btm-outline-info CE
                    option.form-control.btm-outline-info C1E
                    option.form-control.btm-outline-info DE
                    option.form-control.btm-outline-info D1E
            div.input-group.col-12.col-sm.pr-0(style="margin-right: -15px")
                div.input-group-prepend
                    span.input-group-text Рік випуску
                    input.form-control(id='car_year' required type="number" placeholder="2018" min=2000 max=2020)

div.col-sm-10.col-12.mb-2
    div.input-group
        div.input-group-prepend
            span.input-group-text Номер авто
            input.form-control(type="text" placeholder="XX" maxlength=2 id='driver_autonum1')
            input.form-control(type="text" placeholder="0000" maxlength=4 id='driver_autonum2')
            input.form-control(type="text" placeholder="XX" maxlength=2 id='driver_autonum3')
div.col-sm-10.col-12.mb-2
    div.input-group(style='font-size: xx-large;')
        div.form-check-inline
            label.form-check-label.input-group
                div.input-group-prepend
                    div.input-group-text
                        input.form-check-input(type="radio" onclick='renew_car_model()' checked name="car_class"
value='econom')
                    span.form-control Економ
            div.form-check-inline
                label.form-check-label.input-group
                    div.input-group-prepend
                        div.input-group-text
                            input.form-check-input(type="radio" onclick='renew_car_model()' name="car_class"
value='comfort')
                    span.form-control Комфорт
div.col-sm-10.col-12.mb-2
    div.input-group
        div.input-group-prepend

```

```

    span.input-group-text Виробник
    select.input-group-append.form-control(onChange='renew_car_model()' id='select_carproducer')
    option.default_option(value="default" selected) Оберіть виробника
    each prod in producers
        option.form-control(value=prod.prodid) #{prod.producer}
div.col-sm-10.col-12.mb-2
    div.input-group
        div.input-group-prepend
            span.input-group-text Марка
            select.input-group-append.form-control(id="select_carmodel")
            option.default_option(value="default" selected) Оберіть марка

h5.col-12.col-sm-12 Оберіть пароль для входу
div.col-sm-10.col-12.mb-2
    div.input-group
        div.input-group-prepend
            span.input-group-text(data-toggle="tooltip" title="Пароль повинен бути мінімум 6 символів, має містити цифри") Пароль
            input.form-control(required type="password" placeholder="Введіть пароль" maxlength=45 id='driver_pass')
            span.invalid-feedback Пароль повинен містити літери нижнього та вищого регістру, цифри та може містити символи
            input.form-control(required type="password" placeholder="Повторіть пароль" maxlength=45 id='driver_pass_conf')
            span.invalid-feedback Паролі не збігаються
div.col-sm-10.col-12.mb-2
    div.input-group.mb-3.align-content-center(style='width: 100%')
        button.btn.btn-md.btn-warning(style='margin: auto; width: 50%' type='submit') Стати водієм

script(type='text/javascript') setValues();
script(type='text/javascript') renew_car_model();
include footer

```

calculator.pug

```

section.condition.content.mt-1.mb-2
    div(id='foralert')
    div.row
        h2.col-12
            kbd Калькулятор цін

hr.m-1
div.container
    div.address_info.row
        div.col-12.col-sm-6
            div.form-group
                label(for="address_from") Адреса відправлення:
                input.form-control(type=text id='address_from' placeholder='Звідки?')
            div.form-group
                label(for="address_to") Пункт призначення:
                input.form-control(type=text id='address_to' placeholder='Куди їдемо?')
            div(style='margin-bottom: 12px')
                div.form-check-inline
                    label.form-check-label
                        input.form-check-input(type='radio' name='class' checked value='econom' onclick='calcPrice()')
                        span Econom
                div.form-check-inline
                    label.form-check-label
                        input.form-check-input(type='radio' name='class' value="comfort" onclick='calcPrice()')
                        span Comfort
            div.input-group
                div.input-group-prepend
                    span.input-group-text Оплата
                select.input-group-append.form-control(id="pay_type" onchange='calcPrice()')
                each type in pay_types

```

```

        option.default_option(value=type.pay_id data-toggle="tooltip" title=type.notes) #{type.name}
    div.col-12.col-sm-6.ordercart(id="map")
    div.col-12.input-group.mt-2
        div.input-group-prepend
            span.input-group-text Вартість:
            input.input-group-append.form-control.bg-white(id='km_val' value='-' disabled=true)

```

include footer

```

script(type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=" + googlemapapi +
"&libraries=places&callback=initMap")
script(src='.././src/js/maps.min.js' type="text/javascript")
script(type="text/javascript" initAutocomplete());

```

changePass.pug

section.error404.content

```

    div.d-none(id='incorrect_alert')
        div.alert.alert-danger.font-small
            button.close(type="button" data-dismiss="alert") &times;
            strong Некоректні дані
            span Пароль не було змінено, через невідповідність даних. Вкажіть правильний пароль та встановіть
новий для входу

```

div.row

```

    h1.col-12 Змініть пароль
    div.input-group.col-10.mb-2
        div.input-group-prepend
            span.input-group-text Старий пароль:
            input.form-control.m-0(type="password" name='old_password' id='old_pass')
        div.input-group.col-10.mb-2
            div.input-group-prepend
                span.input-group-text Новий пароль:
                input.form-control.m-0(type="password" name='new_password' id='new_pass')
            div.input-group.col-10.mb-2
                div.input-group-prepend
                    span.input-group-text Повторіть пароль:
                    input.form-control.m-0(type="password" id='new_pass2')
            button.btn-warning.btn.btn-md.pr-5.pl-5.b(onclick='firstChangePass()') Зберегти зміни

```

include footer

confirmregisrtation.pug

section.error404.content

```

    p.display-3 <b>Підтвердіть реєстрацію</b>
    p На вказану вами пошту було надіслано код підтвердження, скопіюйте його та вставте в поле нижче:
    form(action='/confirmregistrcode' method='get')
        input.form-control(type='text' name='code')
        input.form-control( style='display: none' type='email' name='email' value=email)
        button.btn-warning.btn.btn-lg(type='submit') Підтвердити

```

include footer

contacts.pug

section.container.content

```

    br
    div.row
        h1.col-sm-12.col-12
            kbd Контакти
        div.col-12.col-sm-7
            div
                p
                    b Телефони:
                    pre.mb-0 +38 (066) 377 85 32
                    pre.mb-0 +38 (044) 000 00 00
                p
                    b Адреса:
                    span вулиця Григорія Сковороди, 2, Київ, 04655

```

```

p
  b Електронна пошта:
  code
    pre.mb-0 n.shkarow@gmail.com
    pre.mb-0 n.shkarovska@ukma.edu.ua
  p Ми відкриті для спілкування! Слідкуйте за нами у соцмережах (активні посилання шукайте внизу
сторінки). По всіх питаннях пишiть на наші пошти або телефонуйте в робочий час:
    pre.mb-0 <b>Пн-Пт: </b>
      span 9:00 - 19:00
    pre.mb-0 <b>Сб-Нд: </b>
      span 10:00 - 17:00
    pre.mb-0 <b>Святкові дні: </b>
      span 11:00 - 17:00
  div.googlemap.col-sm-5.col-12(id="googleMap")
include footer
script(src="https://maps.googleapis.com/maps/api/js?key=" + googlemapapi + "&callback=myMap")
script(src='../src/js/maps.min.js' type="text/javascript")
script(type="text/javascript") myMap();

footer.pug
footer.footer.bg-dark
  div.container
    div.row
      div.col-12.col-sm-12
        nav.navbar.navbar-dark(style='position: initial; height: fit-content')
          div.row.wrapper
            ul.col-12.col-sm-12.row.mb-0
              a.col-3.col-sm-2.offset-1.offset-sm-3(href='https://www.facebook.com/natalia.shkarovska'
target='_blank')
                li.facebook
                  i.fa.fa-facebook.fa-2x(aria-hidden="true")
                a.col-3.col-sm-2(href='https://twitter.com/natahiko' target='_blank')
                  li.twitter
                    i.fa.fa-twitter.fa-2x(aria-hidden="true")
                a.col-3.col-sm-2(href='https://www.instagram.com/natahiko/' target='_blank')
                  li.instagram
                    i.fa.fa-instagram.fa-2x(aria-hidden="true")
            div.col-sm-12.col-12.copyright Created by @natalia_shkarovska, Kiev-Mohula Academy 2020

header-client.pug
head
  title=www
  meta(charset='utf-8')
  include outsource
body
  nav.navbar.navbar-expand-md.navbar-dark.fixed-top.bg-dark
    div.container(style="max-width: 100%")
      a.navbar-brand(href="#")
        img.d-inline-block.align-top(src='../files/images/logo.png' width="30" height="30" alt="")
        span #{title}
      button.navbar-toggler(data-toggle="collapse" data-target="#navbarContent" aria-controls="navbarContent"
aria-expanded="false")
        span.navbar-toggler-icon
      div.collapse.navbar-collapse(id="navbarContent")
        ul.navbar-nav.mr-auto
          li.nav-item
            a.nav-link(id='nav_profile' href="/profile") Профiль
          li.nav-item.dropdown(id='nav_client')
            a.nav-link.dropdown-toggle(id='nav_aboutus' href="#" data-toggle="dropdown" aria-expanded="false")
Блог
            div.dropdown-menu
              a.dropdown-item(href="/aboutus") Про нас
              a.dropdown-item(href="/user/userrules") Правила користування
            li.nav-item

```

```

        a.nav-link(id='nav_ordertaxi' href="/ordertaxi") Заовити таксі
    li.nav-item
        a.nav-link(id='nav_mydrives' href="/orders/my") Мої замовлення
    li.nav-item
        a.nav-link(id="nav_contacts" href="/contacts") Контакти
    span.navbar-text
        a.btn.btn-outline-warning(onclick="exit()") Вихід
script(type='text/javascript') document.getElementById("#{nowpage}").className += " active_nav ";
header-driver.pug
head
    title=www
    meta(charset='utf-8')
    include outsource
body
    nav.navbar.navbar-expand-md.navbar-dark.fixed-top.bg-dark
        div.container(style="max-width: 100%")
            a.navbar-brand(href="#")
                img.d-inline-block.align-top(src="../../files/images/logo.png" width="30" height="30" alt="")
                span #{title}
            button.navbar-toggler(data-toggle="collapse" data-target="#navbarContent" aria-controls="navbarContent"
aria-expanded="false")
                span.navbar-toggler-icon
            div.collapse.navbar-collapse(id="navbarContent")
                ul.navbar-nav.mr-auto
                    li.nav-item
                        a.nav-link(id='nav_profile' href="/profile") Профіль
                    li.nav-item.dropdown(id='nav_driver')
                        a.nav-link.dropdown-toggle(id='nav_aboutus' href="#" data-toggle="dropdown" aria-expanded="false")

```

Блог

```

                div.dropdown-menu
                    a.dropdown-item(href="/aboutus") Про нас
                    a.dropdown-item(href="/driver/workcond") Умови роботи
                    a.dropdown-item(href="/driver/usefultips") Корисні поради
                li.nav-item
                    a.nav-link(id="nav_orders" href="/orders") Заявки
                li.nav-item
                    a.nav-link(id='nav_mydrives' href="/orders/my") Мої поїздки
                li.nav-item
                    a.nav-link(id="nav_contacts" href="/contacts") Контакти
            span.navbar-text
                a.btn.btn-outline-warning(onclick="exit()") Вихід
script(type='text/javascript') document.getElementById("#{nowpage}").className += " active_nav ";

```

header-unauthorised.pug

```

head
    title=www
    meta(charset='utf-8')
    include outsource
body.pt-header
    nav.navbar.navbar-expand-md.navbar-dark.fixed-top.bg-dark
        div.container(style="max-width: 100%")
            a.navbar-brand(href='/')
                img.d-inline-block.align-top(src="../../files/images/logo.png" width="30" height="30" alt="")
                span.text-white #{title}
            button.navbar-toggler(data-toggle="collapse" type="button" data-target="#navbarContent")
                span.navbar-toggler-icon
            div.collapse.navbar-collapse(id="navbarContent")
                ul.navbar-nav.mr-auto
                    li.nav-item
                        a.nav-link(id="nav_aboutus" href="/aboutus") Про нас
                    li.nav-item.dropdown
                        a.nav-link.dropdown-toggle(id="nav_driver" href="#" data-toggle="dropdown" aria-expanded="false")

```

Водію

```


dropdown-menu
  a.dropdown-item(href="/driver/becomedriver") Стати водієм
  div.dropdown-divider
  a.dropdown-item(href="/driver/workcond") Умови роботи
  a.dropdown-item(href="/driver/usefultips") Корисні поради
li.nav-item.dropdown
  a.nav-link.dropdown-toggle(id="nav_client" href="#" data-toggle="dropdown" aria-expanded="false")
Клієнту
  div.dropdown-menu
    a.dropdown-item(href="/user/registeruser") Стати користувачем
    a.dropdown-item(href="/user/userrules") Правила користування
    div.dropdown-divider
    a.dropdown-item(href="/orders/calculator") Калькулятор цін
  li.nav-item
    a.nav-link(id="nav_contacts" href="/contacts") Контакти
span.navbar-text
  a.btn.btn-outline-warning(type="button" data-toggle="modal" data-target="#login" href="#") Увійти
script(type="text/javascript") document.getElementById("#{nowpage}").className += " active_nav ";
div.modal.fade(id="login" role="dialog" aria-hidden="true")
  div.modal-dialog(role="document")
    div.modal-content
      div.modal-header
        h5.modal-title Вхід
        button.close(id="login_modal_dismiss" data-dismiss="modal" aria-label="Close")
        span(aria-hidden="true") &times;
      div.modal-body
        form
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text(data-toggle="tooltip" data-placement="left" title="Введіть ваш логін, намер
телефону або електронну пошту") Логін:
            input.form-control(type="text" id="log_login" placeholder="login, number or email...")
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Пароль:
            input.form-control(type="password" id="log_pass" )
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Увійти як:
            select.form-control(id="log_type")
              option(selected='true') Оберіть як ви хочете увійти...
              option(value='clients') Клієнт
              option(value='drivers') Водій
      div.modal-footer
        button.btn.btn-secondary(onclick="showRegisterModal()") Зареєструватись
        button.btn.btn-warning(onclick="loginUser()") Увійти
        span(style="display: none;" id="nowpage") #{nowpage}
div.modal.fade(id="register" role="dialog" aria-hidden="true")
  div.modal-dialog(role="document")
    div.modal-content
      div.modal-header
        h5.modal-title Реєстрація
        button.close(id="register_modal_dismiss" data-dismiss="modal" aria-label="Close")
        span(aria-hidden="true") &times;
      div.modal-body.row.p-5
        h2.col-12.text-center.mb-3 Як ви хочете зареєструватись???
        a.btn.btn-warning.col-5(href="/driver/becomedriver") Водій
        a.btn.btn-warning.col-5.offset-2(href="/user/registeruser") Користувач
my_orders.pug
div.container.content.mt-2

each order in all_orders
  if order.status === 1


```

```

div.row
  div.col-12.card.brad-0.shadow-small.p-0
    div.card-header.bg-warning.brad-0
      span.order_price В подорожі
      span.float-right
        kbd(class='bg_paytype_' + order.pay_type_id) #{order.name}
    div.card-body.row
      div.col-12
        b Ціна:
        span #{order.price} ₪
        br
        b Місце посадки:
        span #{order.address_from}
        br
        b Місце висадки:
        span #{order.address_to}
        if order.comment != null
          br
          span.comment_order
          b Коментар:
          span #{order.comment}
      div.col-12.mt-2
        if type === 'client'
          a.btn.btn-warning.bg-warning-dark.col-3.ml-3(href='/driver/#{order.user_id}') Переглянути профіль
        водія
        else
          a.btn.btn-warning.bg-warning-dark.col-3.ml-3(href='/user/#{order.driver_id}') Переглянути профіль
        клієнта
        a.btn.btn-dark.col-3.ml-3(href='#{order.url}' target='blank') Відкрити маршрут
      if type === 'driver'
        div.card-footer
          button.btn.btn-warning.float-right(onclick="endDrive('' + order.id + '')") Завершити поїздку
          button.btn.btn-secondary.float-right.mr-3(onclick="cancelDrive('' + order.id + '')") Скасувати
        замовлення

    if type === 'client'
      div.row
        each order in all_orders
          if order.status === 0
            div.col-12.card.brad-0.shadow-small.p-0
              div.card-header.brad-0.bg-grey
                span.order_price Замовлено
                span.float-right
                  kbd(class='bg_paytype_' + order.pay_type_id) #{order.name}
              div.card-body.row
                div.col-12
                  b Ціна:
                  span #{order.price} ₪
                  br
                  b Місце посадки:
                  span #{order.address_from}
                  br
                  b Місце висадки:
                  span #{order.address_to}
                  if order.comment != null
                    br
                    span.comment_order
                    b Коментар:
                    span #{order.comment}
              div.card-footer
                button.btn.btn-warning.float-right(onclick="cancelDrive('' + order.id + '')") Скасувати замовлення
      div.row
        h3.col-sm-12.col-12.mb-3

```



```

    kbd Завершені поїздки
  each order in all_orders
    if order.status === 2 || order.status === 3 || order.status === 4
      div.col-12.card.brad-0.shadow-small.p-0
        div.card-header.bg-light-grey.brad-0
          if order.status === 4
            span.order_price Завершено
          else if order.status === 3
            span.order_price.text-dark-red Скасовано клієнтом
          else
            span.order_price.text-dark-red Скасовано водієм
          span.float-right
            kbd(class='bg_paytype_' + order.pay_type_id) #{order.name}
        div.card-body
          b Ціна:
          span #{order.price} ₪
          if order.status === 4
            br
            b Тривалість поїздки:
            span #{order.time_diff}
          br
          b Місце посадки:
          span #{order.address_from}
          br
          b Місце висадки:
          span #{order.address_to}
          if order.comment !== null
            br
            span.comment_order
            b Коментар:
            span #{order.comment}
          br
          if type === 'client'
            a(href='/driver/' + order.driver_id) Переглянути профіль водія
          else if driver_id !== null
            a(href='/user/' + order.user_id) Переглянути профіль клієнта
          else
            div.a Замовлення було скасовано раніше, ніж його взяв водій

include footer

orders.pug
section.condition.content
  h2
    kbd Замовлені поїздки
  div.orders(id='orders')
    include allorders

div.modal.fade(id='order_modal_alert' tabindex="-1" role="dialog" aria-hidden="true")
  div.modal-dialog(role='document')
    div.modal-content
      div.modal-header
        h5.modal-title Увага
        button.close(type="button" data-dismiss="modal" aria-label="Close")
          span(aria-hidden="true") &times;
      div.modal-body
        p Ви не можете брати нові поїздки, адже у вас є активні, незавершені раніше поїздки.
        p Перевірте статуси ваших поїздок на сторінці
          a(href='orders/my') Мої поїздки
      div.modal-footer
        button.btn.btn-warning(type="button" data-dismiss="modal") Зрозуміло!

include footer
script(type="text/javascript") setInterval(renew_orders_method, 3000);

```

ordertaxi.pug

```

section.condition.content
  div(id='foralert')
  div.row
    h2.col-7
      kbd Замовити таксі
    div.col-5.input-group
      div.input-group-prepend
        span.input-group-text Вартість:
      input.input-group-append.form-control.bg-white(id='km_val' value='' disabled=true)
  hr.m-1
  div.container
    div.address_info.row
      div.col-12.col-sm-6
        div.form-group
          label(for="address_from") Адреса відправлення:
          input.form-control(type=text id='address_from' placeholder='Звідки?')
        div.form-group
          label(for="address_to") Пункт призначення:
          input.form-control(type=text id='address_to' placeholder='Куди їдемо?')
        div(style='margin-bottom: 12px')
          div.form-check-inline
            label.form-check-label
              input.form-check-input(type='radio' name='class' checked value='econom' onclick='calcPrice()')
              span Econom
          div.form-check-inline
            label.form-check-label
              input.form-check-input(type='radio' name='class' value="comfort" onclick='calcPrice()')
              span Comfort
        div.input-group
          div.input-group-prepend
            span.input-group-text Оплата
          select.input-group-append.form-control(id="pay_type" onchange='calcPrice()')
          each type in pay_types
            option.default_option(value=type.pay_id data-toggle="tooltip" title=type.notes) #{type.name}

      div.col-12.col-sm-6.ordercart(id="map")
        div.col-12.col-sm-12
          div.form-group
            label(for="address_from") Коментар до замовлення:
            textarea.form-control(id='notes' placeholder='Допишіть ваш коментар...')

        div.input-group.align-content-center(style='width: 100%')
          button.btn.btn-lg.btn-warning.mb-0(style='margin: auto; width: 40%' type='button' onclick='ordertaxi()')

Замовити
include footer
script(type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=" + googlemapapi +
"&libraries=places&callback=initMap")
script(src='../src/js/maps.min.js' type="text/javascript")
script(type='text/javascript') initAutocomplete();

```

outsource.pug

```

link(href = '../src/css/main.min.css', rel = 'stylesheet')
link(href = '../node_modules/bootstrap/dist/css/bootstrap.min.css', rel = 'stylesheet')
script(src='../node_modules/jquery/dist/jquery.min.js' type="text/javascript")
script(defer src='../node_modules/bootstrap/dist/js/bootstrap.min.js' type="text/javascript")
script(defer src='../src/js/validator.min.js' type="text/javascript")
script(src='../src/js/main.min.js' type="text/javascript")

```

profile-client.pug

```

section.mt-3.content(id='profilesection')
  div.main_profile_part.row
    div.col-10.col-sm-5.col-md-4.col-lg-3.offset-1
      div.photo_container

```

```

    img.profile_img(src=info.photo_src id="photo" alt="")

    form(style="display: none" enctype="multipart/form-data" id='upload_photo')
    input.inputfile(id="profile_photo" name="photo" type="file" onchange='sendNewProfilePhoto()')
    label.upload_photo_profile(for="profile_photo")
    span.upload_photo_label

div.personal_info.col-12.col-sm-6.col-md-7.col-lg-8.row
div.col-1.badge.badge-warning Client
div.col-11.col-sm-6
div.input-group
    input.login.m-0(type="text" id='profile_login' value=info.login disabled)
    span.invalid-feedback Ваш логін зайнятий! Оберіть новий
    div.input-group-append(id='checkLogin' style='display: none')
        input.btn.btn-outline-secondary.m-0(type='button' data-toggle="tooltip" title="Перевірте чи вільний
ваш логін" value='Перевірити' onclick="checkFreeLoginClient()")

if info.description
div.col-12
    textarea.status(disabled id='profile_desc') #{info.description}

div.input-group.col-12.col-sm-12.mb-2
div.input-group-prepend
    span.input-group-text III
    input.form-control.m-0(type="text" id='profile_surname' value=info.surname disabled)
    input.form-control.m-0(type="text" id='profile_name' value=info.name disabled)
div.input-group.col-12.col-sm-12.mb-2
div.input-group-prepend
    span.input-group-text Email
    input.form-control.m-0(type="text" id='profile_email' value=info.email disabled)
div.form-inline.row.col-12.col-sm-12.mb-2.pr-0
div.input-group.col-12.col-sm-12.col-md-4.pr-0
div.input-group-prepend
    span.input-group-text Bik
    input.form-control.m-0(type="number" id='profile_age' value=info.age disabled)
div.input-group.col-12.col-sm-12.pr-0.col-md-8
div.input-group-prepend
    span.input-group-text Тел.
    input.form-control.m-0(type="text" id='profile_phone' value=info.phone disabled)
div.col-12.col-sm-6.col-md-7.col-lg-8.offset-sm-6.offset-md-5.offset-lg-4.row
div.col-12.col-md-6.accordion.p-0.offset-sm-0
div.card.border-0
div.card-header.border-0.bg-white.p-0
h5.mb-0
a.profile_drives.p-0.input-group(type="button" data-toggle="collapse" data-target="#collapseOne" aria-
expanded="true" aria-controls="collapseOne")
div.input-group-prepend
    span.input-group-text Загальна кількість поїздок:
    b.form-control.m-0 #{info.total_amount}
div.collapse(id="collapseOne" aria-labelledby="headingOne")
div.card-body.p-0
div.input-group.mb-2
div.input-group-prepend
    span.input-group-text Завершених:
    b.form-control.m-0 #{info.finished_amount}
div.input-group.mb-2
div.input-group-prepend
    span.input-group-text Скасованих водіями:
    b.form-control.m-0 #{info.cancel_driver_amount}
div.input-group.mb-2
div.input-group-prepend
    span.input-group-text Скасованих клієнтом:
    b.form-control.m-0 #{info.cancel_client_amount}

```

```

    div.input-group.col-md-6.offset-sm-0.col-12.mb-2
      input.btn.btn-warning.form-control.h-fit-cont(type="button" id='change_password' value="Змінити пароль"
data-toggle="modal" data-target="#changepass")
    div.input-group.col-8.offset-2
      input.btn.btn-warning.mb-2.form-control(type="button" id='editprofilebutton' value="Редагувати профіль"
onclick='editClientProfile()')
    div.input-group.col-8.offset-2
      input.btn.btn-outline-secondary.mb-2.form-control(style='display:none' type="button" id='canselprofilebutton'
value="Скасувати" onclick='canselClientChanges()')
      input.btn.btn-warning.mb-2.form-control(style='display:none' type="button" id='saveprofilebutton'
value="Зберегти зміни" onclick='saveClientChanges()')

div.modal.fade(id="changepass" tabindex="-1" role="dialog" aria-hidden="true")
  div.modal-dialog(role="document")
    div.modal-content
      div.modal-header
        h5.modal-title Змінити пароль
        button.close(data-dismiss="modal" aria-label="Close")
          span(aria-hidden="true") &times;
      div.modal-body
        form
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Старий пароль:
            input.form-control(type="password" id="old_change_pass" )
            span.invalid-feedback Неправильний старий пароль
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Новий пароль:
            input.form-control(type="password" id="change_pass" )
            span.invalid-feedback Пароль повинен містити літери нижнього та вищого регістру, цифри та може
містити символи
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Підтвердіть пароль:
            input.form-control(type="password" id="change_pass2" )
            span.invalid-feedback Паролі повинні збігатись
        div.modal-footer
          button.btn.btn-secondary(data-dismiss="modal" id='cancelchangepass') Назад
          button.btn.btn-warning(onclick="changePass('clients')") Зберегти
include footer

```

profile-driver.pug

```

div.content
  section.mt-3(id='profilesection')
    div.main_profile_part.row
      div.col-10.col-sm-5.col-md-4.col-lg-3.offset-md-1.offset-sm-0.offset-1
        div.photo_container
          img.profile_img(src=info.photo_src id="photo" alt="")

          form(style="display: none" enctype="multipart/form-data" id='upload_photo')
            input.inputfile(id="profile_photo" name="photo" type="file" onchange='sendNewProfilePhoto()')
            label.upload_photo_profile(for="profile_photo")
            span.upload_photo_label

        div.personal_info.col-12.col-sm-7.col-ld-8.row
          div.col-1.badge.badge-warning Driver
          div.col-11.col-sm-6
            div.input-group
              input.form-control.login.m-0(type="text" id='profile_login' value=info.login disabled)
              span.invalid-feedback Ваш логін зайнятий! Оберіть новий
              div.input-group-append(id='checkLogin' style='display: none')
                input.btn.btn-outline-secondary.m-0(type="button" data-toggle="tooltip" title="Перевірте чи вільний

```

ваш логін" value='Перевірити' onclick="checkFreeLogin()")

```

    if info.description
      div.col-12
        textarea.status(disabled id='profile_desc') #{info.description}
    div.input-group.col-12.col-sm-12.mb-2
      div.input-group-prepend
        span.input-group-text ІІІ
        input.form-control.m-0(type="text" id='profile_surname' value=info.surname disabled)
        input.form-control.m-0(type="text" id='profile_name' value=info.name disabled)
    div.input-group.col-12.col-sm-12.mb-2
      div.input-group-prepend
        span.input-group-text Email
        input.form-control.m-0(type="text" id='profile_email' value=info.email disabled)
    div.form-inline.row.col-12.col-sm-12.mb-2.pr-0
      div.input-group.col-12.col-sm-12.col-md-4.pr-0
        div.input-group-prepend
          span.input-group-text Вік
          input.form-control.m-0(type="number" min=21 max=65 id='profile_age' value=info.age disabled)
      div.input-group.col-12.col-sm-12.pr-0.col-md-8
        div.input-group-prepend
          span.input-group-text Тел.
          input.form-control.m-0(type="text" id='profile_phone' value=info.phone disabled)
    div.col-12.col-sm-6.col-md-7.col-lg-8.offset-sm-6.offset-md-5.offset-lg-4.row
      div.col-12.col-md-6.accordion.p-0.offset-sm-0
        div.card.border-0
          div.card-header.border-0.bg-white.pl-0.pr-0
            h5.mb-0
              a.profile_drives.p-0.input-group(type="button" data-toggle="collapse" data-target="#collapseOne"
aria-expanded="true" aria-controls="collapseOne")
                div.input-group-prepend
                  span.input-group-text Загальна кількість поїздок:
                  b.form-control.m-0 #{info.total_amount}
          div.collapse(id="collapseOne" aria-labelledby="headingOne")
            div.card-body.p-0
              div.input-group.mb-2
                div.input-group-prepend
                  span.input-group-text Завершених:
                  b.form-control.m-0 #{info.finished_amount}
              div.input-group.mb-2
                div.input-group-prepend
                  span.input-group-text Скасованих водієм:
                  b.form-control.m-0 #{info.cancel_driver_amount}
              div.input-group.mb-2
                div.input-group-prepend
                  span.input-group-text Скасованих клієнтами:
                  b.form-control.m-0 #{info.cancel_client_amount}

          div.input-group.col-md-6.offset-sm-0.col-12.mb-2
            input.btn.btn-warning.form-control.h-fit-cont(type="button" id='change_password' value="Змінити
пароль" data-toggle="modal" data-target="#changepass")

    div.driver_auto.pb-1.m-0.ml-lg-2
      div.form-inline.row
        div.input-group.col-12.mb-3.col-md-5.col-sm-12
          div.input-group-prepend
            span.input-group-text Номер
            input.form-control.m-0(type="text" id='carnumber' value=info.carnumber disabled)
        div.input-group.col-12.mb-3.col-md-3.col-sm-4
          div.input-group-prepend
            span.input-group-text Права
            input.form-control.m-0(type="text" value=info.licence disabled)
        div.input-group.col-12.mb-3.col-md-4.col-sm-8

```

```

div.input-group-prepend
  span.input-group-text Клас
select.form-control.m-0(disabled onchange='renew_car_model()' id='carclassprofile')
  if info.class === 'econom'
    option(selected) econom
    option comfort
  else
    option econom
    option(selected) comfort
div.input-group.col-12.mb-3.row.pr-0
  div.input-group-prepend.col-3.col-sm-2.pr-0
    span.input-group-text Авто
  select.form-control.m-0.col-9.col-sm-4(disabled onchange='renew_car_model()' id='select_carproducer')
    each prod in producers
      if prod.prodid === info.producer_id
        option.form-control(selected value=prod.prodid) #{prod.producer}
      else
        option.form-control(value=prod.prodid) #{prod.producer}
  select.form-control.m-0.col-8.col-sm-4(disabled id="select_carmodel")
  input.form-control.m-0.col-4.col-sm-2(type="number" id='profile_carid' min=2000 max=2020
value=info.caryear disabled)
  div.row
    div.input-group.col-8.mb-2.offset-2
      input.btn.btn-warning.mb-2.form-control(type="button" id='editprofilebutton' value="Редагувати профіль"
onclick='editDriverProfile()')
    div.input-group.col-8.mb-2.offset-2
      input.btn.btn-outline-secondary.mb-2.form-control(style='display:none' type="button" id='canselprofilebutton'
value="Скасувати" onclick='canselDriverChanges()')
      input.btn.btn-warning.mb-2.form-control(style='display:none' type="button" id='saveprofilebutton'
value="Зберегти зміни" onclick='saveDriverChanges()')

div.modal.fade(id="changepass" tabindex="-1" role="dialog" aria-hidden="true")
  div.modal-dialog(role="document")
    div.modal-content
      div.modal-header
        h5.modal-title Змінити пароль
        button.close(data-dismiss="modal" aria-label="Close")
        span(aria-hidden="true") &times;
      div.modal-body
        form
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Старий пароль:
            input.form-control(type="password" id="old_change_pass" )
            span.invalid-feedback Неправильний старий пароль
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Новий пароль:
            input.form-control(type="password" id="change_pass" )
            span.invalid-feedback Пароль повинен містити літери нижнього та вищого регістру, цифри та може
містити символи
          div.input-group.mb-3
            div.input-group-prepend
              span.input-group-text Підтвердіть пароль:
            input.form-control(type="password" id="change_pass2" )
            span.invalid-feedback Паролі повинні збігатись
        div.modal-footer
          button.btn.btn-secondary(data-dismiss="modal" id='cancelchangepass') Назад
          button.btn.btn-warning(onclick="changePass('drivers')") Зберегти

script(type='text/javascript') renew_car_model(`#{info.carmodelid}`)
include footer

```

registered-client.pug

section.error404.content

h1 Ви успішно зареєстровані

p.xlarge На вказану вами пошту було надіслано <u>Пароль</u>, який ви можете використати для першого входу на сайт, а потім змінити його на власний:

button.btn-warning.btn.btn-md.pr-5.pl-5.b(type="button" data-toggle="modal" data-target="#login" href="#")

Увійти

include footer

successRegistered.pug

div(style='width: 80%; margin: 15px 10%')

div.alert.alert-success

button.close(type="button" data-dismiss="alert") ×

strong Вітаємо, ви зареєстровані!

span Ви можете увійти в систему скориставшись логіном та паролем, вказаними при реєстрації

р або почитати корисні поради для водіїв

include workcond

userrules.pug

div.content

section.condition

h2

kbd Правила користування

i для клієнтів

р Щоб покращити нас сервіс та зробити його приємним і для водіїв і для клієнтів ми рекомендуємо дотримуватись рекомендацій та правил користування нашим сервісом таксі

р Взаємодотримання правил зробить вашого водія добрішим, а вашу поїздку приємнішою, окрім того,

б ЗА НЕДОТРИМАННЯ ПРАВИЛ ВОДІЙ МАЄ ПОВНЕ ПРАВО ЗАВЕРШИТИ ПОЇЗДКУ

ДОСТРОКОВО!

span Ці правила є загальними правилами для більшості схожих сервісів нашої країни, тому

i їх повинен знати кожен

div.album.py-5.bg-light

div.container

div.row

each blog in blogs

div.col-sm-6.col-lg-4.col-12.mb-3

div.card.shadow-sm.h-100.mb-0

svg.bd-placeholder-img.card-img-top(width="100%" height="50")

rect.rect

div.card-body.row

h2.col-12 #{blog.title}

p.card-text.col-12 #{blog.text}

div.col-10.offset-2.text-sm-right.align-self-end

small.text-muted #{blog.date.toString() }

section.condition

p(style='text-align: center') Сподіваємось, ви дотримуватимесь вищевказаних правил та ваші поїздки з нашим сервісом будуть виключно приємними!!!

include footer

workcond.pug

div.content

section.pict-with-caption

div.semi-transparent

div.card(style='padding: 20px')

h2 Робота в нас - це мрія будь-якого водія!!! Приєднуйтеся вже зараз

р Кожен наш працівник - особистість, тому ми піклуємось про вас, вашу безпеку та надаємо найкращі умови праці

form.become(action='/driver/becomedriver' method='get')

button.btn.btn-warning(type='submit') Стати водієм

section.condition

р Ми приймаємо на роботу лише найкращих! Щоб працювати у нас ви та ваше авто повинні відповідати деяким мінімальним вимогам, що наведені нижче

h2 Вимоги до автомобіля

ul

li Авто має бути випущене після 2000-го року

```

    li Автомобіль повинен мати не менше 4 дверей
    li ПОвинне бути дійсне свідоцтво про реєстрацію транспортного засобу
    li Вартість автомобіля економ-класу повинна бути більшою за 140 000 (сто сорок тисяч) грн
section
  div.container
    div.section-header
      h2 Основні вимоги до водіїв
      p
    div.row
      div.col-lg-4
        div.box
          img.myicon(src='../files/images/icon4.png')
          h4 Посвідчення водія
          p Водій завжди повинен при собі мати посвідчення відповідної категорії, а також бажано паспорт
      div.col-lg-4
        div.box
          img.myicon(src='../files/images/icon5.png')
          h4 Стаж від 3 років
          p Ви маєте мати практику водіння автомобілем від 3 років, при цьому без ДТП за останній рік
      div.col-lg-4
        div.box
          img.myicon(src='../files/images/icon6.png')
          h4 Документи
          p При собі завжди потрібно мати
            b Ліцензію на перевезення пасажирів
            span та
            b Свідоцтво на авто
section.mt-5
  div.container
    div.section-header
      h2 Список дозволених машин
      p
    ul.nav.nav-tabs(style='font-size:x-large')
      li.nav-item
        a.nav-link.active(data-toggle="tab" href="#economTable") Економ
      li.nav-item
        a.nav-link(data-toggle="tab" href="#comforttable") Комфорт +
    div.tab-content
      div.tab-pane.container.active(id='economTable')
        div.row
          each mod in allmodels
            if mod.class === "econom"
              div.col-6.col-sm-4.col-md-3.p-10
                div.card.hover-light
                  div.card-body.row
                    div.col-12 #{mod.model}
                    div.col-12.text-sm-right #{mod.producer}
      div.tab-pane.container(id='comforttable')
        div.row
          each mod in allmodels
            if mod.class === "comfort"
              div.col-6.col-sm-4.col-md-3.p-10
                div.card.hover-light
                  div.card-body.row
                    div.col-12 #{mod.model}
                    div.col-12.text-sm-right #{mod.producer}
include footer

```