Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ З АЛГОРИТМОМ АНАЛІЗУ ТОВАРІВ

Текстова частина до кваліфікаційної роботи

за спеціальністю „Інженерія програмного забезпечення” 121

Керівник кваліфікаційної роботи

с.в. Борозенний С.О.

(прізвище та ініціали)

_____

(підпис)

“____” _____ 2024 р.

Виконала студентка

Михайленко О.В.

(прізвище та ініціали)

“____” _____ 2024 р.

Київ 2024

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,

доцент, к.ф-м.н.

_____О. П. Жежерун (підпис)

„____”_____2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студентці Михайленко Олександрі Валеріївні факультету інформатики 4-го курсу

ТЕМА Розробка веб-застосунку для електронної комерції

з алгоритмом аналізу товарів

Зміст ТЧ до кваліфікаційної роботи:

Індивідуальне завдання

Вступ

1 Огляд предметної області

2 Огляд існуючих аналогових систем

3 Опис реалізації веб застосунку електронної комерції з рекомендаційною системою

Висновки

Список літератури

Додатки (за необхідністю)

Дата видачі „___” _____ 2024 р. Борозенний О.С. _____ (підпис)

Завдання отримав _____ (підпис)

Тема: Розробка веб-застосунку для електронної комерції з алгоритмом аналізу товарів

Календарний план виконання роботи:

№ п/п	Назва етапу кваліфікаційної роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу	жовтень 2023	
2.	Аналіз матеріалів за темою	14.10.2023	
3.	Розробка Веб-застосунку для електронної комерції з алгоритмом аналізу товарів	15.03.2024	
4.	Написання текстової частини до кваліфікаційної роботи	30.04.2024	
5.	Створення презентації для доповіді	09.05.2024	
6.	Коригування виконаної роботи	20.05.2024	
7.	Остаточне оформлення роботи	25.05.2024	
8.	Захист кваліфікаційної роботи	27.05.2024	

Михайленко О. В. _____

Борозенний О. С. _____

“_____”_____

Зміст

Анотація 5
Вступ 6
Розділ 1. Огляд предметної області 8
1.1. Аналіз сучасної проблематики та обґрунтування теми 8
1.2. Постановка задачі 9
Розділ 2. Огляд існуючих аналогових систем 9
2.1. Огляд наявних веб-застосунків електронної комерції 9
2.2. Огляд рекомендаційних систем 15
Висновок 20
Розділ 3. Опис реалізації веб-застосунку електронної комерції з рекомендаційною системою 20
3.1. Аналіз технічного завдання 20
3.2. Опис архітектури застосунку 23
3.3. Опис технологій, використаних для створення веб-застосунку 25
3.3.1. Технології клієнтської частини 26
3.3.2. Технології серверної частини 27
3.3.3. Безпека системи 28
3.3.4. Засоби створення бази даних 30
3.3.5. Засоби розгортання веб-застосунку у мережі Інтернет 31
3.4. Структура бази даних 32
3.5. Структура проекту 33
3.6. Опис рекомендаційного алгоритму 37
3.7. Інтерфейс програми 39
Висновок 50
Список використаних джерел 51
Список використаних зображень 53

Анотація

Кваліфікаційна робота присвячена розробці веб-застосунку для електронної комерції з алгоритмом аналізу товарів.

Front-end створення засобами бібліотеки React. Для розробки back-end було використано: Node, Express. База даних: PostgreSQL.

Веб застосунок розгорнуто засобами Amazon Web Services.

Ключові слова: електронна комерція, алгоритм аналізу товарів, веб розробка, AWS

Вступ

Актуальність:

Електронна комерція за останні двадцять років стала важливою складовою сучасного бізнесу. Розвиток технологій допоміг людям комунікувати знаходячись на довгих відстанях, а бізнесу поширювати свої місця збуту на великі території, в тому числі відправляти товари по всьому світу. Маючи офлайн точку бізнес охоплює увагу обмеженої кількості людей, але якщо створити мультимовний веб сайт, товари будуть доступні для людей будь-якої національності і будь-де у світі.

Розвиток бізнесу впливає не тільки на прибутки конкретної організації, а й на економічний стан країни, і як наслідок, на якість життя населення. Варто зазначити, що технології більш адаптивні до аномальних робочих умов, зокрема під час пандемії наявність онлайн частини бізнесу виявилась вкрай важливою для виживання компаній у таких несприятливих обставинах. За один рік з 2019-ого по 2020 рік, світовий ринок електронної комерції зріс на 77%. Показники українського ринку також стрімко зросли на 40%.[1] Тож розвиток технологій у сфері електронної комерції або e-commerce є важливою складовою розвитку економіки України та в цілому світової економіки.

Наразі у світі є дуже багато пропозицій, конкуруючих організацій та компаній. Для того щоб кожен міг знайти свого покупця, стало необхідно створення рекомендаційних систем. Такі алгоритми несуть користь як бізнесу, допомагаючи збільшити прибуток, так і клієнтові – адже система знаходить те, що цікавить конкретну особу, підшукуючи актуальні варіанти й економлячи час та сили.

Мета дослідження:

Розробка веб-застосунку електронної комерції з рекомендаційною системою товарів.

Об’єкт дослідження:

Рекомендаційні системи по пошуку релевантних товарів для користувачів, а також веб-застосунки та методи хостингу.

Розділ 1. Огляд предметної області

1.1. Аналіз сучасної проблематики та обґрунтування теми

Наразі сфера електронної комерції стрімко розвивається. Після пандемії стало зрозуміло: щоб вижити бізнесу потрібно бути гнучким, адаптуватись під незвичні ситуації та підлаштовуватись під клієнтів. Через те, що населення планети масово мало сидіти вдома, а економіка і бізнес не можуть застигнути, e-commerce сфера стрімко розрослась. Як наслідок, збільшилась конкуренція у цій галузі, тож нові рішення, альтернативні підходи і покращення наявних є дуже важливим для розвитку і підвищення конкурентоспроможності бізнесу на сьогодні.

Більш того, наразі час є одним з найважливіших ресурсів, тож коли потенційному клієнту не потрібно їхати у магазин, щоб щось придбати, а достатньо взяти в руки свій смартфон і натиснути кілька разів на екран, це значно полегшує досвід користувача.

З розвитком технологій покращується сервіс. Якщо в минулому сторіччі люди стикались з дефіцитом товарів і відсутністю вибору, то зараз кількість різновидів однієї одиниці товару є такою великою, що користувачі майже ніколи не догортають до останніх сторінок електронних каталогів. При цьому, потреба знайти найбільш релевантний товар для себе все ще є актуальною. І в цьому допомагають пошукові та рекомендаційні системи. Пошук по запиту користувача, а також фільтрація звужують кількість знайдених одиниць контенту, проте рекомендаційний алгоритм здатний передбачити забаганки покупця ще до того, як він за це задумається. Такі системи економлять час, несуть додаткову користь у вигляді позитивних емоцій для кінцевого споживача та допомагають отримувати додатковий прибуток виробникам.

Підсумовуючи варто сказати, що галузь електронної комерції швидко розвивається, а отже потребує свіжих ідей і рішень. Дослідження і покращення рекомендаційних алгоритмів допомагає поліпшити взаємодію потенційного покупця зі застосунками для продажу товарів та послуг.

1.2. Постановка задачі

У ході аналізу предметної області було виділено такі етапи дослідження даної кваліфікаційної роботи:

1. провести аналіз популярних існуючих додатків у галузі e-commerce;
2. дослідити сучасні підходи до реалізації рекомендаційних систем;
3. розробити веб-застосунок електронної комерції з рекомендаційним алгоритмом на основі проведеного дослідження.

Розділ 2. Огляд існуючих аналогових систем

2.1. Огляд наявних веб-застосунків електронної комерції

Для того щоб детально дослідити вже наявні рішення, було обрано три популярні сервіси електронної комерції:

- Rozetka – найпопулярніший український маркетплейс товарів з різних ніш;
- Prom – сервіс, що дозволяє порівнювати однакові товари між собою по ціні та відгуках;
- Instagram – донедавна відомий як соціальна мережа, але наразі цей сервіс тестує можливість здійснювати покупки одразу у застосунку, ця опція доступна наразі тільки у США.

Усі застосунки вище є посередниками між бізнесом та покупцем, тобто це платформи де продавець може виставити свій товар, а покупець його придбати. Сфера електронної комерції також включає і не великі веб-вітрини, що представляють продукцію одного виробника, або магазину. Оскільки це не масштабні застосунки, які наразі можна створити використовуючи конструктор сайтів, розглядатися такі рішення у контексті даної кваліфікаційної роботи не будуть.

Серед зазначених веб-сервісів є 2 класичні маркетплейси: Rozetka та Prom, і альтернативний, новий підхід у сфері електронних продажів – Instagram. Функціонал останнього сервісу знаходиться на стадії запуску, і тестується тільки у деяких регіонах. Аналіз різнопланових рішень допоможе розібратись, який з них є найкращим.

У вищезгаданих сервісах будуть оглядатись та оцінюватись такі критерії:

- SEO оптимізація
- UX – зручність і відповідність до стандартних норм створення інтерфейсу для покупців;
- Додаткові можливості для продавців;
- Робота рекомендаційної системи.

Rozetka

1. SEO

Сервіс використовує проплачену рекламу від Google, що дозволяє йому бути першим по запиту «Rozetka» з приписом «реклама». При цьому в них добре налаштоване SEO, тож безоплатне перше посилання також належить цьому сервісу.

У додачу, веб-застосунок піклується про просування окремих товарів у Google, тож якщо шукати конкретні категорії товарів, які представлені у продажі на Rozetka, посилання на ці товари або категорії з’являються у топ 10 перших посилань, на різних місцях, в залежності від запиту.

Варто зазначити, що у просуванні своїх товарів Rozetka також використовує рекламу у Google і вона є унікальною для кожного користувача. Адже в рекламі з’являються саме ті товари, що вже цікавили клієнта раніше, але покупка не була здійснена.

2. User Experience

Досвід користувача відповідає інтуїтивним очікуванням потенційного клієнта. Є зручний розподіл товарів по категоріям, обширні можливості по фільтрації товарів, і всі логічні способи сортування.

На головній сторінці представлені популярні товару у різних категоріях, та акційні пропозиції. Такі оголошення є загально популярними і або не адаптуються, або дуже слабко адаптуються під унікальні потреби користувача.

Зберігається історія замовлень користувача, є можливість створювати списки побажань та порівнянь. Також зберігається історія перегляду товарів, чати з продавцями і можна передивитись всі відгуки, що користувач залишав в одному місці.

Особливо корисним можна виділити можливість налаштувати систему розсилки під себе: відмовитись від не потрібної інформації і лишити тільки корисну.

В інтерфейсі легко розібратись новому користувачу. Він коректно адаптується під будь-який розмір екрану комп’ютера, планшета та телефона.

3. Можливості для продавців

Продавці можуть скористуватись кількома способами покращення просування свого товару:

- Встановити знижку на товар, у такому разі маркетплейс буде автоматично частіше показувати покупцям такий товар
- Створити комплекти товарів, що разом будуть обходитись дешевше
- Проплатити участь у промоакціях
- Створити рекламну кампанію для свого товару

Всі ці методи є або платними(останні два), або приносять потенційно менше прибутку при продажі(як перші два). [2]

4. Рекомендаційна система

Потенційний покупець має можливість відфільтрувати та відсортувати товари по своїм потребам. Проте, якщо покупка не була здійснена, але користувач проявив зацікавленість до товару, алгоритм це запам’ятає і створить рекламу у Google, де з’явиться товар, що вже привернув увагу, та популярні речі з тієї ж категорії.

Prom

1. SEO

Так само, як і сервіс вище, Prom має проплачене просування у пошуковій системі Google та правильно налаштоване SEO. Що дозволяє йому займати перше місце з позначкою «реклама» та перше місце у рекомендованих посиланнях.

Також, якщо шукати окремі товари по конкретним назвам(модель, виробник, тощо), то оголошення з Prom будуть у перші топ 10 посилань, якщо такий товар присутній у їхньому каталозі.

2. User Experience

Має подібний досвід користувача, як в Rozetka.

Проте через велику кількість підкатегорій, при наведені на категорію виводиться величезний список, що складно розгледіти на малорозмірному екрані комп’ютера.

На головному екрані висвічуються товари, в залежності від того, які товари вже були куплені з цього акаунту. Видно аналіз унікальних потреб користувача.

3. Можливості для продавців

Prom надає можливість рекламувати товари за відсоток з продажі, або за плату за кожен клік, що призводить до розгортання сторінки товару. Метод просування залежить від категорії товару.[3]

4. Рекомендаційна система

Prom є яскравим представником колаборативної фільтрації(докладніше про цей вид рекомендаційного алгоритму написано у наступному розділі), адже під час огляду конкретного товару, він показує, які речі беруть інші покупці у комплекті з даним. Таким чином, вони розширюють кількість куплених товарів в одного користувача, показуючи які ще товари можуть бути вам цікаві.

Instagram

1. SEO

По деяким запитам в пошуковій системі Google можна отримати посилання на конкретне відео з соціальної мережі. Переходячи за таким посиланням, користувач потрапляє на веб-версію Instagram, проте це, на відміну від попередніх сервісів, саме додаток, тож веб-версія не призначена для

постійного користування і закликає перейти у додаток, щоб продовжити перегляд.

2. User Experience

З недоліків можна виділити, що веб-версія не призначена для постійного користування, тож щоб здійснювати покупку користувачу потрібно мати завантажений додаток.

Зручно те, що як тільки користувач побачив товар, що йому сподобався, він швидко може його придбати, не виходячи з соціальної мережі і не витрачаючи час на додатковий пошук.

3. Можливості для продавців

Продавці можуть купити просування свого відео в залежності від свого бюджету. Це корисна функція, адже просування одного ролику або посту не є дорогим і можна корегувати бюджет на рекламу у своєму електронному кабінеті бізнес сторінки Instagram.

У самій системі можна співпрацювати з інфлюєренсами, що є одним з найпоширеніших видів реклами останнього десятиліття.

І найголовніше можна самостійно впливати на продажі шляхом покращення якості свого контенту. Тобто є безкоштовний шлях поширення свого товару: кращий контент, він більше подобається людям, він отримує більші охоплення.

4. Рекомендаційна система

Підхід відрізняється від вище згаданих веб-сервісів електронної комерції. Instagram аналізує вподобання користувачів – потенційних клієнтів, фіксуючи які короткі відео вони додивляються до кінця, а які - ні. Таким чином

вибудовується унікальний підхід до кожного, завдяки чому товари будуть показані дійсно зацікавленій аудиторії.

Проте, щоб така рекомендаційна система змогла визначити, кому цікавий той чи інший контент – потрібно витратити час на постійне ведення своєї бізнес сторінки. Через якийсь час алгоритм знайде аудиторію, яка зацікавлена у такій продукції і зможе просувати і збільшувати продажі ефективніше.

2.2. Огляд рекомендаційних систем

Якщо раніше була проблема світового масштабу - неосвіченість людей, обмеженість доступності навчання, а головне складність зі зберіганням, копіюванням та передачею інформації на великі відстані, то зараз, в еру стрімкого розвитку технологій, людство зіштовхнулось з проблемою перенасиченості інформаційного простору. Зараз даних на різну тему на стільки багато, що користувач фізично не може витратити стільки часу, щоб дослідити чи хоча б ознайомитись з усіма матеріалами. На допомогу приходять пошукові та рекомендаційні системи.

Оскільки пошукові системи, є важливою складовою електронної комерції, розглянемо спочатку підходи до створення алгоритмів знаходження релевантних пропозицій за запитом користувача. Головною задачею таких систем є економія часу потенційного клієнта, тобто знайти найбільш релевантний варіант відповідно до запиту користувача.

Найяскравіший, найпопулярніший приклад пошукової системи це Google. Він не тільки знаходить найбільш відповідні веб-сайти, а навіть допомагає користувачу дописати свій запит. Тобто надає підказки під час введення фрази для пошуку. Щоб обробити таку кількість запитів Google використовує потужні

сервери, що здатні обробити запити великої кількості користувачів з усього світу одночасно.

Для того, щоб зробити добірку найбільш релевантних веб-сайтів, пошукова система аналізує їх текстовий вміст, а також альтернативний текст зображень і знаходить сторінки на яких є згадування пошукового запиту(пряме або видозмінене) кілька разів. Після цього знайдені сайти необхідно розподілити, створивши певний порядок, це робиться завдяки аналізу попереднього досвіду взаємодії інших користувачів з цією сторінкою. Google фіксує як багато людей відвідують цей сайт і скільки часу вони на ньому проводять. Такий аналіз відбувається постійно, тож якщо веб-сторінка була популярна два останні місяці, з якихось причин, а потім припинила бути цікавою, пошукова система це помітить і знизить рейтинг цього сайту, оскільки він тепер не відповідає потребам користувачів.

Google також враховує об’ємність сайту, чи є в ньому посилання на інші зовнішні ресурси, а також внутрішні – тобто інші підсторінки цього ж домену. Більш того, пошукова система відслідковує чи зсилаються на цю сторінку інші веб-сайт, або чи згадано її у постах в соціальних мережах, тощо. Таким чином вибудовується рейтинг довіри до веб-ресурсу, Google упевнюється що це не спамерська сторінка і що вона варта уваги користувачів.

Існує окрема галузь – SEO (англ. search engine optimization) просування, спеціалісти з якої займаються оптимізацією веб-сайтів, щоб вивести їх у топ перших сторінок у пошуку. Тобто Google має безліч правил, обмежень і внутрішньої логіки, що породили окрему сферу діяльності, професію і посаду у світі. При цьому достеменно не відома повноцінна логіка роботи найбільшої пошукової системи і навіть досвідчені спеціалісти часом не можуть передбачити дію алгоритмів. SEO не є темою даної кваліфікаційної роботи, тож

не будемо на цьому зупинятись. Проте варто зазначити, що SEO просування тісно пов’язано з сферою електронної комерції, адже потенційні клієнти інколи шукають товари одразу в Google, а не на конкретних маркетплейсах. Тому e-commerce веб-ресурсам необхідно налаштовувати своє SEO таким чином, щоб користувачам вибивалась саме їх сторінка у Google.

Системи пошуку використовуються в багатьох галузях. Наразі поле для введення запитів, можна знайти на безлічі різноманітних сайтів. Пошук інтегрують як для взаємодії з великими базами даних, так і у малих каталогах, для зручності потенційного покупця. Проте рекомендаційні системи в більшості використовуються саме для сфери електронної комерції. Також їх широко використовують у рекламі, яскравий приклад це таргетована реклама у соціальних мережах. Такі алгоритми інтегруються у застосунках, що надають можливість перегляду кіно та відео, публікують новини.

Рекомендаційні системи мають перед собою складнішу задачу, адже їх головним завданням є передбачити потреби користувача. У цьому випадку потенційний клієнт не вказує свій запит, але має певну проблему, яку йому складно пояснити програмі. Або, часто, людина не має конкретної потреби, але певна система, наприклад YouTube рекомендує якесь відео для перегляду, що є релевантним для користувача. Як наслідок, людина отримала цікаву або корисну інформацію, а YouTube отримав задоволеного клієнта. Рекомендаційний алгоритм аналізує дії (поточні і попередні), враховує наявні данні про користувача, щоб передбачити його потреби і підшукує відповідні варіанти товарів абощо. Розглянемо основні типи рекомендаційних систем:

1. Колаборативна фільтрація (англ. collaborative filtering)[4]

Ця система є найпопулярнішим рекомендаційним методом. Алгоритм аналізує певні данні про людину і знаходить інших, схожих користувачів

системи. На основі того, що подобається клієнтам з подібними даними, він робить припущення, що ця, нова, людина матиме такі самі вподобання і рекомендує їй відповідні товари, відео абощо, відсортовані за різними критеріями, зокрема популярністю серед інших. Такі алгоритми зазвичай засновані на методах машинного навчання і здатні аналізувати певні закономірності, що допомагає покращувати підхід до рекомендацій.

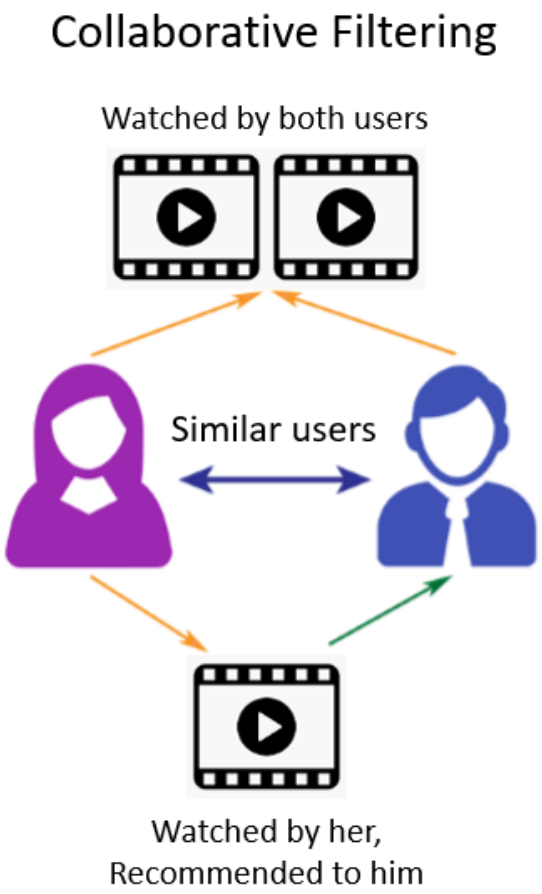


Рисунок 2.1. Схема колаборативної фільтрації

Негативним фактором колаборативної фільтрації є її потреба у збиранні особистих даних користувачів. Люди зазвичай не хочуть ділитись інформацією про себе, а чим менше інформації тим складніше розподілити користувачів на кластери і прорахувати, що їм буде цікаво.

2. Контентна фільтрація (англ. content-based filtering)[5]

У цьому випадку алгоритм враховує не тільки дані про користувача, а ще аналізує інформацію про, наприклад, товари. Цей метод співставляє людей з речами, що їм подобались, або були ними куплені і на основі такого зв’язку рекомендує подібні товари. Сортування рекомендаційного контенту відбувається на основі їх рейтингу у системі, що може включати сукупність факторів, як наприклад середній рейтинг відгуків, кількість переглядів, покупок, часу, який витрачають люди на огляд даного товару, тощо.

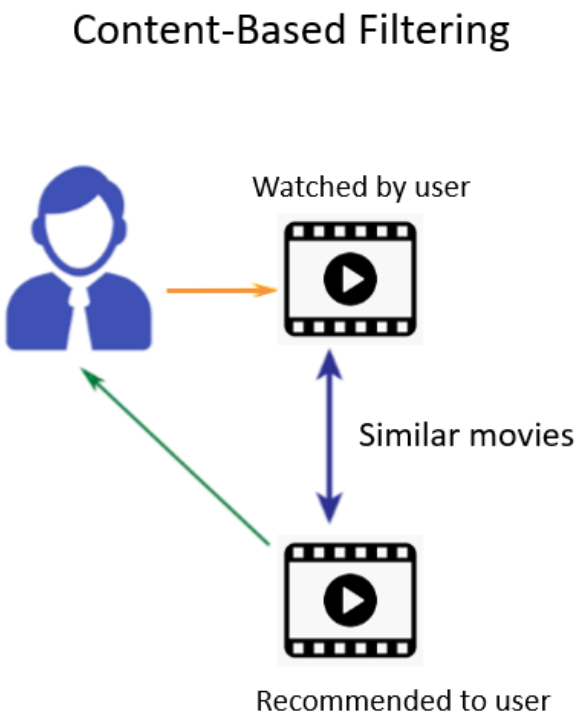


Рисунок 2.2. Схема контентної фільтрації

Недоліком такої системи є її некоректна робота по відношенню до нових користувачів та товарів. Адже перші ще не мають історії замовлень, а другі не мають оцінок на основі яких можна зробити аналіз зацікавленості.

Підсумовуючи, варто сказати, що наявні рекомендаційні системи не є ідеальними. Кожен новий застосунок електронної комерції ставить собі за мету бути кращим за своїх конкурентів, а отже, розробляє і тестує нові способи покращення досвіду користувача (зокрема економії часу пошуку). Поки люди

стикаються із тим, що застосунки пропонують їм не релевантні пропозиції, рекомендаційні системи є куди розвивати і покращувати.

Висновок

У ході аналізу було виявлено сильні сторони вже існуючих лідерів сфери електронної комерції:

- Інтуїтивно зрозумілий інтерфейс, що адаптується під будь-який екран. Це забезпечує комфорт потенційних покупців, та економить їх час.
- SEO оптимізація для просування популярних товарів у пошуковій системі Google. Це допомагає збільшити продажі, що додає лояльність продавців і призводить до довгострокової співпраці.
- Рекомендаційна система – це потужний інструмент, що допомагає підібрати для користувача найактуальніший для нього товар. Варто зазначити, що це є найбільш важливим інструментом, адже він поєднує у собі переваги: економить час пошуку покупця, та збільшує кількість продажів у продавців.

Також було проведено дослідження існуючих підходів до створення рекомендаційних систем. Для реалізації у даній кваліфікаційні роботі було обрано тип - контентної фільтрації. Обґрунтування такого вибору, а також деталі реалізації докладно описані у пункті 3.6.

Розділ 3. Опис реалізації веб-застосунку електронної комерції з рекомендаційною системою

3.1. Аналіз технічного завдання

Мета:

Створити веб-застосунок маркетплейс та реалізувати алгоритм аналізу товарів у магазині для розуміння успішності, затребуваності товару, а також просування його для інших користувачів, для яких цей товар буде актуальним.

Функціональні вимоги:

У системі має бути 4 ролі: адміністратор, власник(головний адміністратор), виробник і користувач. Кожен з цих ролей має мати можливість виконувати певні дії, а саме:

Адміністратор має право або можливість

- додавати/редагувати/видаляти категорії, підкатегорії та під-під категорії
- погоджувати/відхиляти нові товари, що додають виробники
- відправляти/поновлювати виробників в чорному списку
- керувати своїм обліковим записом: редагувати інформацію про себе, додавати аватар
- налаштувати систему оповіщень під власні потреби, а саме отримувати листи з інформацією на пошту, або у телеграм у чат-боті. Мають відправлятися наступні повідомлення:
    - про те, що було залишено відгук на товар з рейтингом 3 зірки або менше
    - про те, що було залишено питання від клієнта про певний товар
    - про відповідь на відгук, що залишив адміністратор
    - про звернення до служби підтримки, з текстом звернення і контактними даними автора
    - про завантаження нового товару, що потребує узгодження

Власник має право:

- все що й адміністратор

- додавати/видаляти адміністраторів
- редагувати дані про адміністратора

Виробник(продавець) має право:

- зареєструватись, та авторизуватись в подальшому у системі
- подавати нові товари на узгодження (заповнювати про них інформацію)
- бачити список вже опублікованих своїх товарів
- встановити можливість самовивозу(вписавши місто, адресу, як туди дійти)
- налаштувати систему оповіщень під власні потреби, а саме отримувати листи з інформацією на пошту, або у телеграм у чат-боті. Мають відправлятися наступні повідомлення:
    - про те, що на їх товар залишили відгук
    - про те, що на їх товар залишили відгук-запитання
    - про те, що заявку на додавання товару затвердили, або навпаки відхилили
    - про те, що зробили нове замовлення товару
- виконувати пошук по своїм товарам
- бачити статистику своїх товарів: кількість покупок кожного товару, у скількох користувачів лежать в кошику, і в скількох у списках бажань
- редагувати інформацію про наявні товари
- бачити таблицю замовлень, редагувати статус замовлення
- встановити товар, як тимчасово не доступний для покупців.

Користувач має право:

- дивитись перелік товарів, по категоріям
- користуватись пошуком товарів по: артикулу та назві
- зв'язатися з службою підтримки - адміністратором
- залишати відгуки та відгуки-запитання на товари

- додавати товари (або кілька екземплярів одного товару) в кошик та свій список побажань
- переглядати історію своїх покупок
- мати свої бонуси та застосовувати їх під час покупок
- редагувати інформацію про себе у своєму профілі
- побачити на головній сторінці рекомендовані для неї товари
- побачити на головній сторінці топ-3 найбільш популярних товари цього тижня
- налаштувати систему оповіщень під власні потреби, а саме отримувати листи з інформацією на пошту, або у телеграм у чат-боті. Мають відправлятися наступні повідомлення:
    - про те, що товар є знову в наявності (про товари, що лежать у списку бажань, або кошику клієнта, якщо вони не були в наявності)

3.2. Опис архітектури застосунку

Проаналізувавши технічне завдання та різні архітектури веб-застосунків, було прийнято рішення реалізовувати даний проект на клієнт-серверній архітектурі[6].

У даному контексті клієнт – це користувач, що заходить з будь-якого пристрою на веб-сайт і відправляє запити для отримання якоїсь інформації, або виконання певних дій на сервер. Сервер обробляє запити від клієнта, використовуючи back-end програмного коду та, після цього, надає відповідь клієнтові.

Існують дві концепції реалізації клієнт-серверної архітектури:

1. Потужний клієнт. У такому разі на клієнтській стороні відбувається певна обробка дій, а сервер виступає більше як

сховище даних і передає необхідну інформацію за запитом клієнта.

2. Слабкий клієнт, потужний сервер. У такому разі всі процеси: відтворення дій, обробка даних, тощо, відбуваються на сервері. У той час як клієнт тільки відтворює інтерфейс додатку, відправляє запити на сервер та обробляє відповіді.

Після огляду технічного завдання було прийнято рішення реалізовувати другу концепцію. Адже такий підхід знімає навантаження з пристроїв користувачів додатку. Також варто зазначити, що друга концепція потребує наявності потужного сервера, особливо зі збільшенням кількості користувачів, що одночасно користуються даним сервісом.

Клієнт-серверна архітектура має два типи:

- Дворівнева, тобто складається з двох ярусів, а саме сервера і клієнта.
- Трирівнева, тобто складається з трьох ярусів, а саме серверу, як прикладного компонента, клієнта і сервера бази даних

У контексті даного додатка, найбільш релевантним є створення трирівневої архітектури. Розподіл операцій знижує навантаження на сервер, що допомагає веб-застосунку бути більш стійким. Також трирівнева архітектура може бути легше перероблена на багаторівневу, тобто перейти на використання більше ніж двох серверів, що дозволить швидше адаптувати систему до збільшення навантаження через зріст зацікавленості клієнтів до сервісу. Використання кількох серверів допоможе оптимізувати розподіл навантаження, а також підвищити ефективність системи (як приклад, швидше відправляти відповіді на запити клієнтів). Більш того, в подальшому можна розподілити монолітний код на мікросервіси, де кожен сервер буде незалежно від інших підтримувати роботу одного сервісу. Такий підхід допоможе системі бути стресостійкою: якщо один з

сервісів дасть збій, припинить працювати тільки він, а не вся система. Схематичне зображення взаємодії пристроїв у клієнт-серверного застосунку зображено на рис. 3.1.

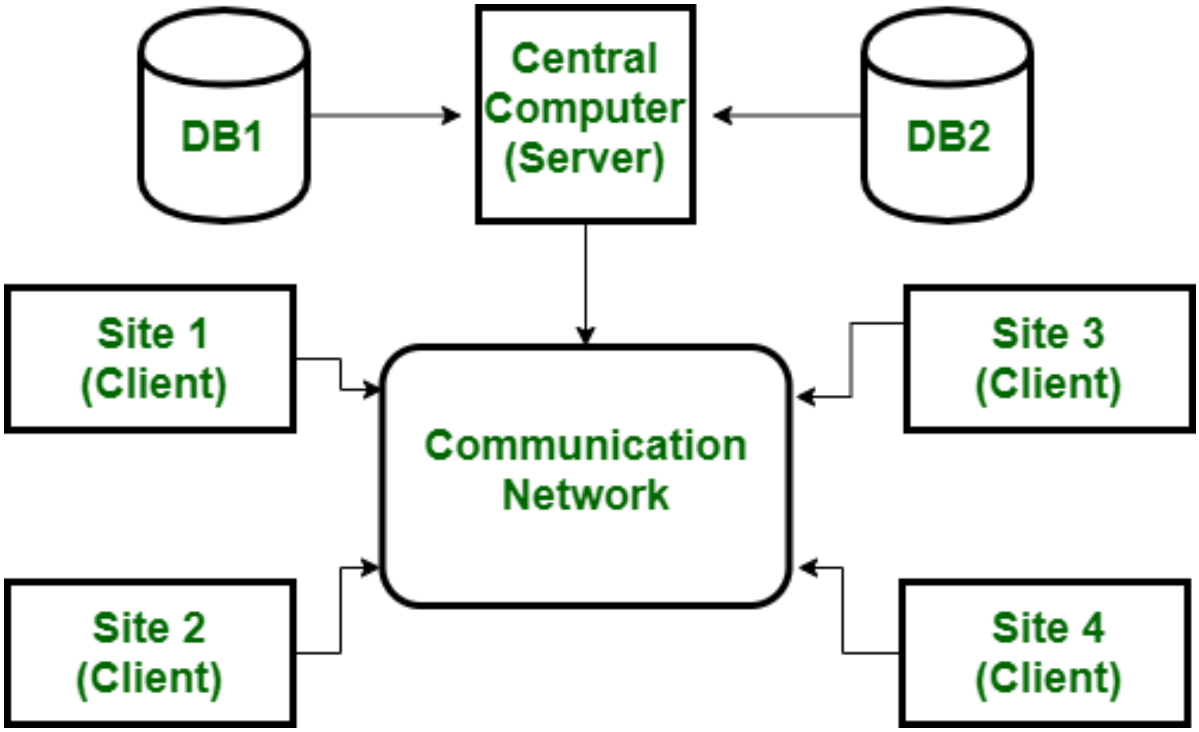


Рисунок 3.1. Структура клієнт серверної архітектури

Клієнт і сервер(а також сервер бази даних) взаємодіють між собою через мережеві протоколи – систему правил, використовуючи які пристрої взаємодіють через мережу.

Отже, обрана трирівнева клієнт-серверна архітектура з потужним сервером допомагає розподілити навантаження між пристроєм клієнта – замовника, та сервером. Може бути легше адаптована і покращена при масштабуванні проекту.

3.3. Опис технологій, використаних для створення веб-застосунку

3.3.1. Технології клієнтської частини

Для розробки частини інтерфейсу користувача веб-додатку було обрано JavaScript бібліотеку – React. Цей декларативний інструмент для створення клієнтської частини заснований на компонентах, що дозволяє уникнути дублювання коду при побудові складних великих веб-застосунків. React є найрозповсюдженішим рішенням для розробки веб-інтерфейсу. Унікальною перевагою інструменту є можливість використовувати його при розробці мобільних застосунків на React Native.

Оскільки React саме бібліотека, а не фреймворк, він не накладає архітектурні обмеження і дозволяє створювати унікальний код під свої потреби. Засобами React можна створити будь-який інтерфейс(адаптивний, з анімаціями, різноманітними ефектами тощо), адже для стилізації використовується спеціалізована мова стилів – CSS. Більш того, React компонент має життєвий цикл, і розробник отримує доступ впливати на нього на кожному етапі: від його рендерингу, до вилучення з екрану. Для створення елементів використовується JSX – XML-подібний синтаксис, що дозволяє писати JavaScript методи посеред HTML розмітки. JSX не є обов'язковим інструментом, проте він значно спрощує і пришвидшує процес написання коду.

На додачу, React дозволяє використовувати безліч додаткових бібліотек, що надають готові UI або UX рішення і заощаджують час розробників.

Також, відсутність архітектурних обмежень дозволяє розробляти додатки, що потребуватимуть масштабування в майбутньому. Таким чином, додаток може бути адаптованим під розробку архітектурного рішення – мікрофронтендів[7]. Це мікросервісна архітектура тільки для структуризації коду інтерфейсу користувача. Такий підхід створено для того, щоб частини інтерфейсу могли розроблятись незалежно один від одного і не перетворювались на один

монолітний фронт. Наразі дедалі популярнішим стає розробка SPA (Single Page Application).Така сторінка з часом може збільшуватись у розмірах і перетворюватись на моноліт. Мікрофронтенд вирішує цю проблему, перетворюючи сторінку на кілька веб-аплікацій. Таке розділення відбувається за виконуваними бізнес-функціями та/або можливостями, що вони мають перекривати. Такі веб-аплікації можна розробляти паралельно.

3.3.2. Технології серверної частини

Для створення серверної частини застосунку було обрано фреймворк Express[8]. Це гнучкий інструмент для розробки Node.js веб та мобільних застосунків.

Node.js - це платформа, що надає можливість запуску JavaScript коду. Він з’явився у 2009 році і створив революцію, адже раніше JavaScript використовувався тільки як механізм браузера Google Chrome (він користувався V8 – рушієм JavaScript з відкритим кодом). Node.js перевикористав V8, надавши можливість йому працювати на серверах, що дозволило створювати серверний код на тій же мові програмування, що й frontend методи. У результаті, розробники тепер мають змогу застосовувати однакові парадигми програмування на сервері та клієнті.

Більш того, Node.js підтримує асинхронний стиль кодування, V8 є потужним інструментом, що в сукупності надає можливість писати швидкодіючий код (без використання багатопоточності). Також, JavaScript має велику кількість додаткових бібліотек та широку спільноту, що спрощує процес кодування і пошуку відповідей на різні запитання.

Фреймворк Express спрощує Node.js API та додає нові функції. Зокрема, він робить менш складним HTTP- сервер, наприклад: пересилання .jpeg файлу є

досить затратною по потужності операцією у чистому Node.js, Express у свою чергу дозволяє вмістити таку операцію лише в один рядок. На рис 3.2. зображено спосіб взаємодії Express з HTTP- сервером Node.js.

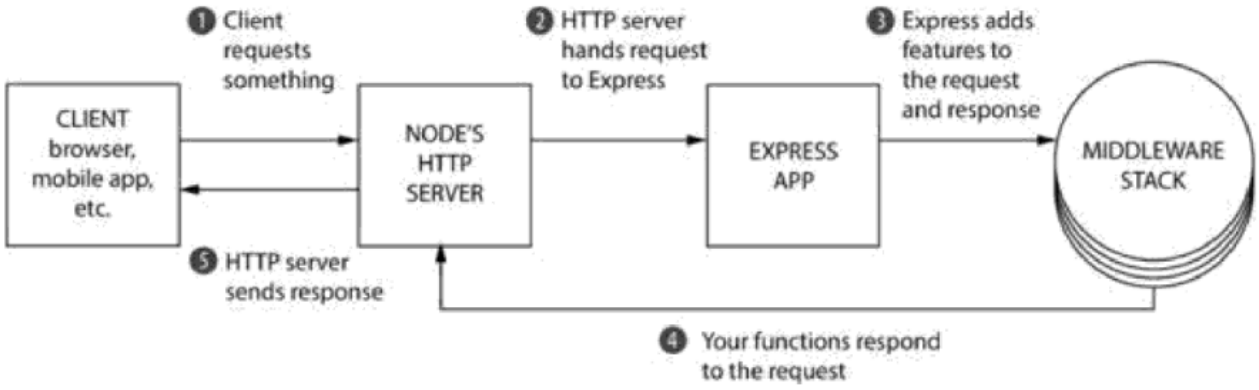


Рисунок 3.2. Схема взаємодії Express та Node.js

3.3.3. Безпека системи

Розробляючи велику систему, що буде зберігати особисту інформацію користувачів, важливо створити безпечний сервіс авторизації. Для захисту даних користувачів системи у проекті цієї кваліфікаційної роботи використовується JWT - JSON Web Token.

JWT складається з 3 частин: header, payload та signature.

Header обов’язково зберігає в собі тип алгоритму кодування. Також може містити тип токену та додаткові допоміжні данні.

Payload містить інформацію про користувача. Наповнюваність цієї частини задає розробник.

Попередні 2 частини зберігаються у зашифрованому вигляді, але зловмисник може їх розшифрувати. Для того щоб забезпечити безпеку системі на сервері міститься секретний ключ, який не відомий для користувачів. За

допомогою ключа header, payload кодується і утворює signature. Signature допомагає переконатись в тому, що токен не є підробкою зловмисників.

Токени поділяються на 2 типи: access token і refresh token.

Тип токена	**Access token**	**Refresh token**
Час існування	15-30 хвилин, але цей час може змінюватись і до кількох днів, в залежності від потреби системи у захисті	15-60 днів, в залежності від потреби системи
Мета	Використовується для доступу до сервера	Використовується для оновлення доступу
Місце збереження	Зберігається в localstorage	Зберігається в httpOnly cookies

Коли користувач авторизується в системі, введені email та пароль відправляються на сервер. Сервер генерує access token і refresh token. Під час використання системи, якщо користувач доступається до індивідуальних даних, то від клієнта надсилається запит на сервер. У запиті міститься access token, якщо він коректний, тоді сервер повертає status 200 та інформацію, що була запитана. Якщо access token втратив свою чинність, тоді сервер повертає status 401. Необхідно передбачити таку відповідь завчасно і на клієнтській частині обробити таку відповідь: відправити refresh token, який зберігається у cokies. Отримавши refresh token сервер звірить його з наявним у базі даних, створить нову пару access token і refresh token і відправить його клієнту. Таким чином клієнт зможе знову користуватись системою протягом часу існування нового access token.

Для того щоб убезпечити паролі користувачів системи, в проекті використовується хешування паролів. Тобто паролі в базі даних зберігаються у захешованому вигляді. Перевагою такого методу є те що захешовані паролі неможливо перетворити назад у звичайний рядок. Таким чином, навіть якщо зловмисники отримали доступ до бази даних, інформація про паролі для них не буде корисною, адже її ніяк не можна використати для доступу до профілів користувачів системи.

3.3.4. Засоби створення бази даних

Для збереження інформації було обрано PostgreSQL.[9] Це об’єктно-реляційна база даних, основною перевагою якої є відкритий вихідний код. Вона використовує і навіть розширює структуровану мову запитів (англ. SQL). PostgreSQL має перевірену архітектуру, яка базується на моделі клієнт-сервер, забезпечує надійне збереження і цілісність даних, відома своєю відмовостійкістю.

Об’єктно-реляційна база даних має підхід як у реляційних базах, а саме, дані зберігаються у вигляді сукупності таблиць (формально описаних). Інформація з цих таблиць може бути зібрана різними засобами за допомогою мови запитів, і при цьому не потрібно перероблювати структуру бази. У той же час, PostgreSQL вважається об'єктною через використання об'єктно-орієнтованої системи управління базами даних - object-oriented database management system (OODBMS). У подібних системах база даних виступає як стійкий об'єкт, сховище для програмного забезпечення, написаного об'єктно-орієнтованою мовою програмування, з API для зберігання та вилучення об'єктів.

PostgreSQL має велику спільноту користувачів, адже була створена аж 1986 році. Таким чином можна швидко знайти вирішення проблем, з якими

стикається програміст під час розробки. Більш того, вона може бути інтегрована у проект будь-якої мови програмування, що робить її універсальним рішенням для різних проектів. А вивчення роботи саме цієї бази даних дозволяє програмісту легше адаптуватись під різні проекти.

Ще однією перевагою PostgreSQL, є можливість утворення зв'язку з середовищем розробки IntelliJ IDEA, що пришвидшує процес написання запитів і зменшує кількість помилок через одруківки. А також, pgAdmin – програма з зручним інтерфейсом, що дозволяє контролювати стан таблиць і перевіряти їх наповнення у режимі реального часу.

3.3.5. Засоби розгортання веб-застосунку у мережі Інтернет

Для розгортання застосунку у мережі Інтернет було обрано платформу веб-сервісів – Amazon Web Services (далі AWS)[10]. Вона надає широкий спектр можливостей, такі як: зберігання даних, обчислення, робота у мережі на різних рівнях абстракції.

До переваг використання саме AWS можна віднести:

- Зручність у використанні – на кожен інструмент є обширна документація і навіть відео-інструкції, що допомагають швидше їх опанувати;
- Швидка робота захощеної системи – одна з головних переваг серед конкурентів;
- Гнучкість – AWS надає велику кількість інструментів які можна налаштувати унікально під свої потреби;
- Безпечність – AWS має зхищену інфраструктуру і можливість спостерігати на різними рівнями безпеки, а також налаштовувати безпеку

застосунку відповідно до матеріальних ресурсів і нагальних потреб компанії;

- Наявність безкоштовного пробного періоду на 1 рік. Це дозволяє новим спеціалістам розібратись з різними інструментами та специфікою роботи AWS.

Під час розгортання застосунку даної кваліфікаційної роботи було використано сервіс EC2 (Elastic Compute Cloud), що відповідає за створення віртуальної машини. Також, для розгортання бази даних застосовано інструмент RDS (Amazon Relational Database Service).

3.4. Структура бази даних

Щоб оптимізувати час створення бази даних необхідно спочатку намалювати схему (див. рис. 3.3.). На схемі зображено всі сутності, зв’язки між ними, атрибути та їх характеристики.

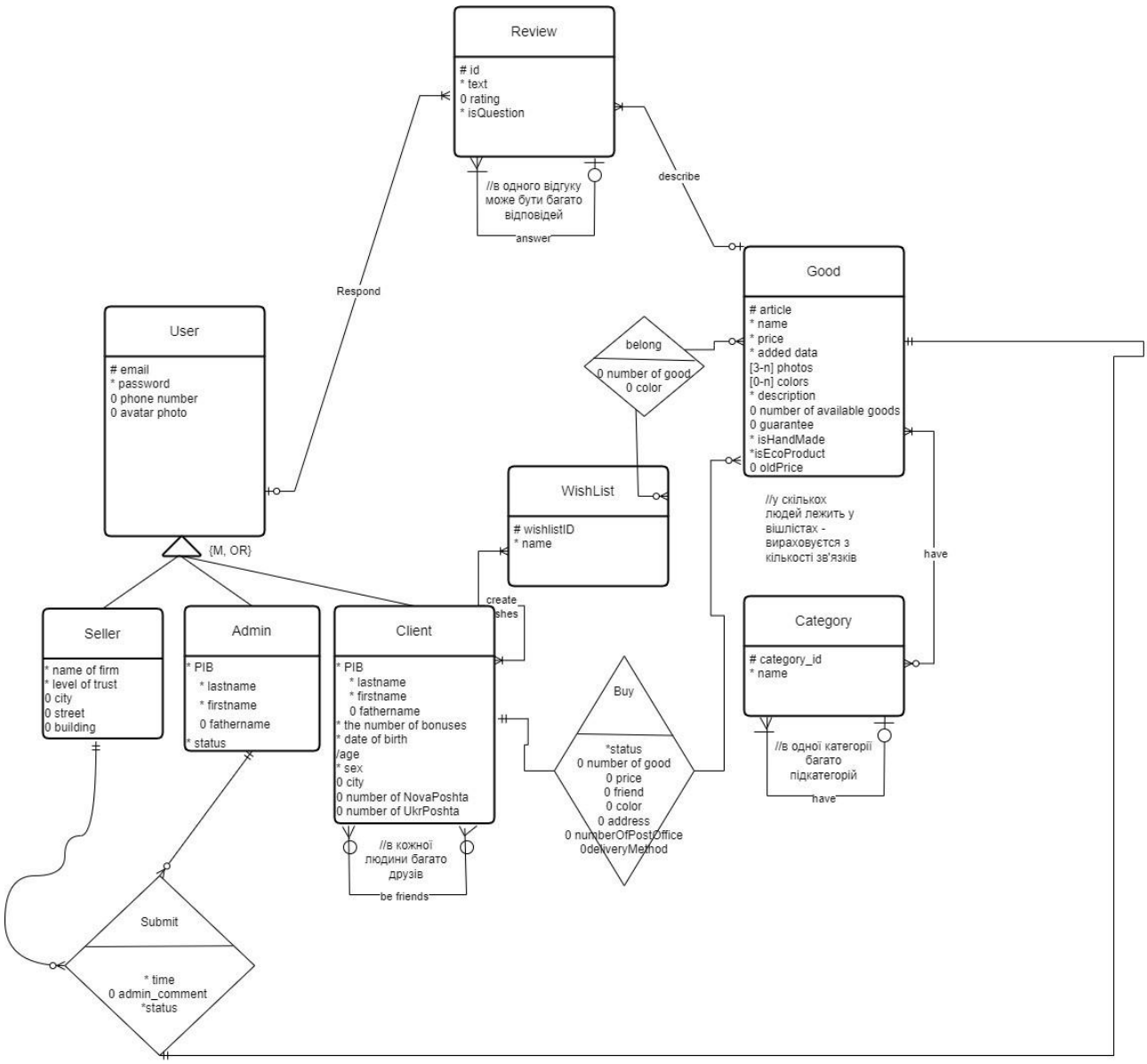


Рисунок 3.3. Схема бази даних

Схема бази даних дозволяє структурувати пункти технічного завдання, створити візуальне сприйняття. Спираючись на неї, далі створюється план побудови бази даних з розподілом на окремі таблиці. Визначаються обмеження, а також тип даних для кожного атрибуту.

3.5. Структура проекту

Оскільки проект побудований використовуючи правила клієнт-серверної архітектури, він містить 2 головні директорії «client» і «server».

Каталог «client» відповідає за створення інтерфейсу програми і використовує для цього сукупність бібліотек React. Стартове наповнення генерується автоматично, для цього достатньо написати команду:

npx create-react-app my-app

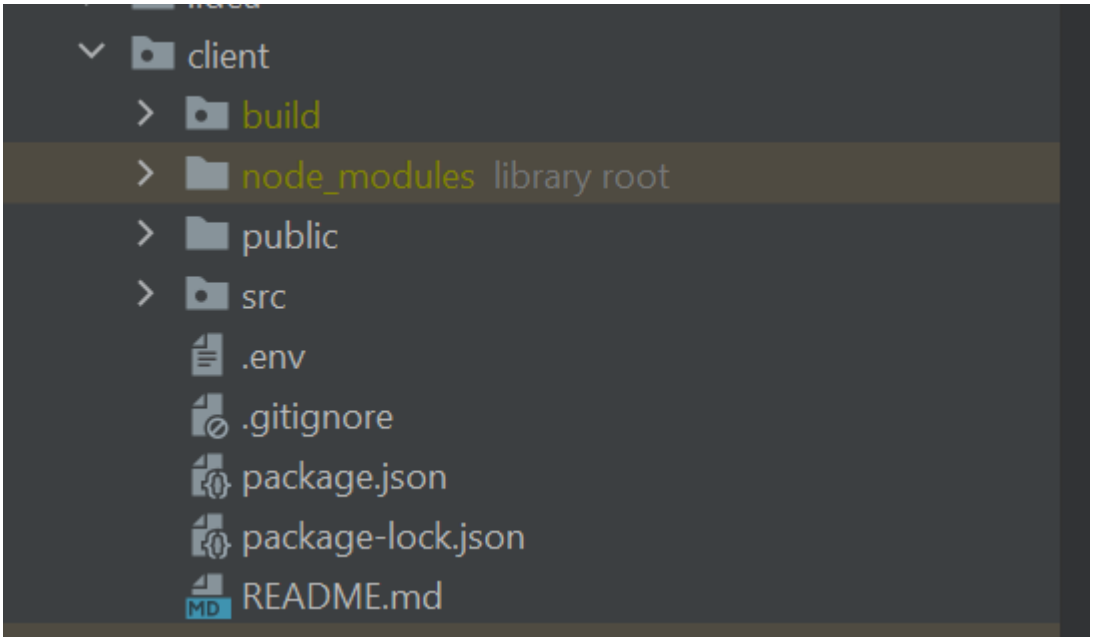


Рисунок 3.4. Структура каталогу «client»

На рис. 3.4. представлено скріншот структури frontend частини проекту. Розберемо детальніше кожну піддиректорію:

«build» - каталог, що генерується автоматично, якщо у терміналі написати команду npm run build. Після відтвореної даної команди відбувається автоматична генерація скрипта з написаного коду. Ця директорія створюється на останньому етапі, після закінчення написання frontend коду, щоб використовувати написаний код у інших проектах, або для хостингу.

node_modules - генерується автоматично під час створення проекту. Тут міститься усі встановлені залежності для frontend частини проекту, а також, сюди завантажуються додаткові бібліотеки.

public - містить статичні іконки (тобто зображення, які використовуються постійно під час роботи застосунку), а також файли: index.html, та manifest.json.

src – містить файли з основним кодом frontend

.env – файл, що зберігає змінні

package.json та package-lock.json – конфігураційні файли

Каталог «server» містить у собі логіку програми, backend частину. Працює на основі Node. Стартове наповнення генерується автоматично, для цього достатньо написати команду:

```
npm init
```

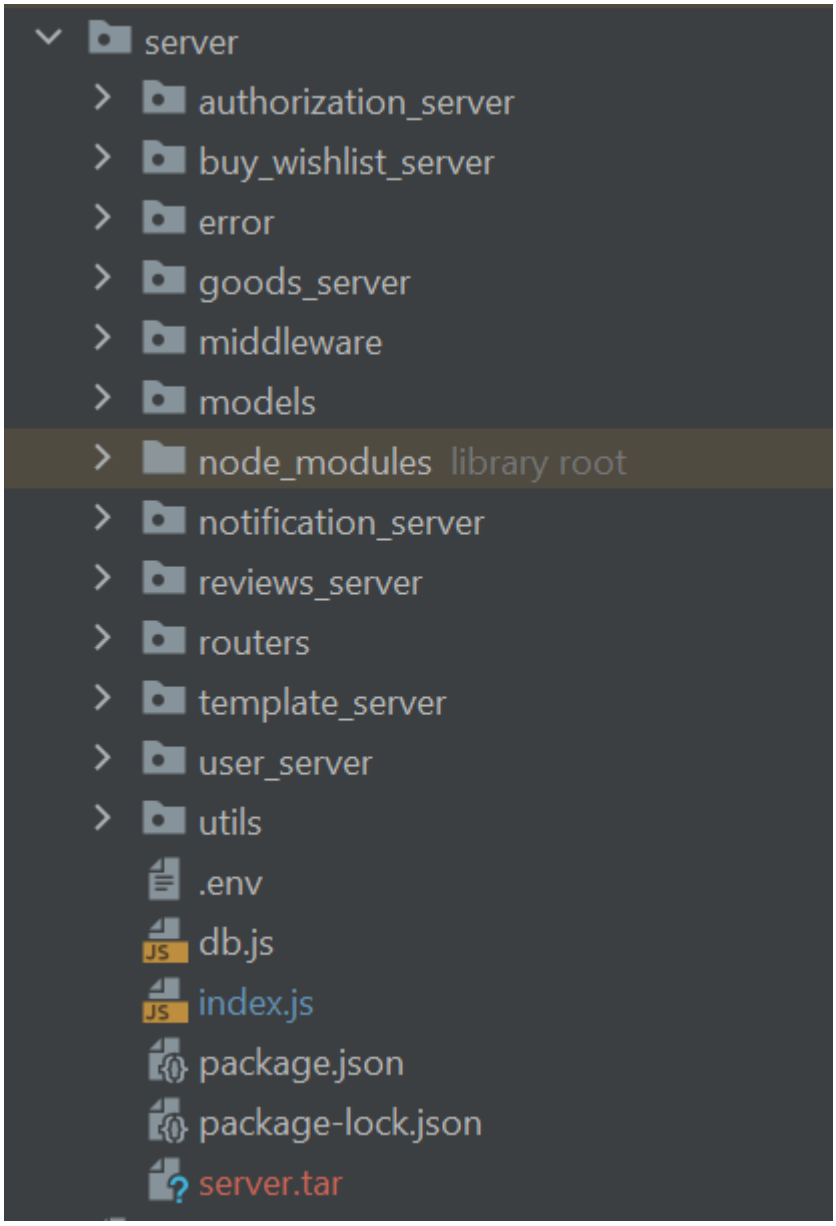


Рисунок 3.5. Структура каталогу «server»

node_modules - містить усі встановлені залежності для backend частини проекту, а також, сюди завантажуються додаткові Node бібліотеки.

Інші директорії містять код backend (відокремлений на підкаталоги за логічними принципами)

.env – містить константні зовнішні змінні середовища

db.js – містить налаштування для бази даних

index.js – файл з якого запускається backend.

package.json та package-lock.json – файли конфігурацій.

server.tar – згенерований архів для хостингу серверної частини.

3.6. Опис рекомендаційного алгоритму

Вище було зазначено основні види рекомендаційних систем. Для поглибленого вивчення можливостей алгоритмів та подальшій реалізації його у практичній частині було обрано тип - контентної фільтрації. Адже він, по-перше, дозволяє проаналізувати більше даних, бо враховується вся інформація про товари, якою веб-застосунок може необмежено користуватись, на відміну від приватних даних користувачів. По-друге, цей підхід є більш персоналізованим, бо спирається на поведінку конкретної особи, а не на думку інших людей, які нібито схожі на головного об’єкта.

Під час розробки власного алгоритму, варто пам’ятати, що є необхідність мінімізувати основну проблему даного підходу. Вона полягає у тому, що система не може створити рекомендацію для нового клієнта, а також про новий товар, адже не має інформації про них.

Таким чином було розроблено алгоритм, який спирається на наступні логічні правила:

Після авторизації користувача в системі, алгоритм знаходить які категорії товарів були цікаві цій особі. Він проходиться по товарам, що були куплені і збережені, знаходить категорії, до яких вони відносяться, далі прибирає дублі категорій з масиву. А потім сортує від тих, що зацікавили нещодавно до найдавніших.

Після такого аналізу ми отримуємо масив з кількома категоріями, але з них потрібно обрати товари, що будуть найбільш релевантні для користувача. Для цього, спочатку, потрібно пройтись по масиву всіх товарів і прибрати ті, що вже були куплені даним користувачем, а також ті, що вже лежать у його списку побажань.

Товари, що лишились, важливо відсортувати від тих, що з найбільшою вірогідністю можуть сподобатись, і це робиться таким чином: контент сортується спочатку по кількості здійснених вдалих покупок, потім за середнім рейтингом відгуків(від найбільшого), після цього за кількістю збережень кожної одиниці зі списку і в кінці за рейтингом довіри до продавця(тобто якщо продавець має високий рівень довіри, його товари будуть стояти вище).

Щоб мінімізувати головну проблематику контентної фільтрації, було виконано такі дії:

Додано систему збереження товарів у списку побажань і включено категорії цих товарів у аналіз рекомендаційним алгоритмом. Придбати нову річ, може і не зареєстрована особа, але щоб зберегти якийсь товар у списку своїх побажань людина обов’язково має мати акаунт. При цьому реєстрація і збереження для користувача є морально набагато простішою дією, а ніж замовлення й оплата.

Таким чином, система знатиме які товари радити новій особі після моменту першого збереження. Такий підхід є швидшим, а ніж чекати першої покупки. Також, для зовсім нового користувача (ще не зареєстрованого, або з щойно створеним акаунтом), програма буде створювати топ найпопулярніших товарів тижня, тобто тих, що замовляли найбільшу кількість разів за останні 7 днів.

Щоб не залишити у тіні нові товари, що ще не мають ніякого внутрішнього рейтингу в системі, створено додаткове перемішування відсортованого масиву товарів. Вище зазначено, як саме алгоритм відсортовує потенційно найцікавіші товари на перші місця. Щоб дати шанс новим товарам на продажі, відбувається додаткове проходження по масиву і на 3-4 товари з найбільшим рейтингом вставляється картка нового товару(яка при попередньому сортуванні опиняється внизу масиву). Важливо, що ця картка не з поганим рейтингом і тому опинилась в кінці списку, а саме немає поки ніяких даних для аналізу.

У підсумку можна зазначити: щоб покращити розповсюджену систему фільтрації та рекомендації товарів, було створено рейтинг довіри до продавця, що впливає на сортування і як наслідок, допомагає убезпечити клієнтів від нових продавців, що можуть виявитись недоброчесними. Було впроваджено додаткове сортування масиву товарів, результатом якого, ново додані речі до системи отримують можливість бути поміченими. А також, пришвидшено здатність системи до аналізу нових користувачів, завдяки можливості збереження товарів і включенні їх категорій у аналіз.

3.7. Інтерфейс програми

Як вже було зазначено вище, у технічному завданні, у веб-застосунку є чотири ролі: адміністратор, власник, продавець і покупець. Розглянемо кожну з них.

Адміністратор.

Панель адміністратора має 4 вкладки:

1. Товари

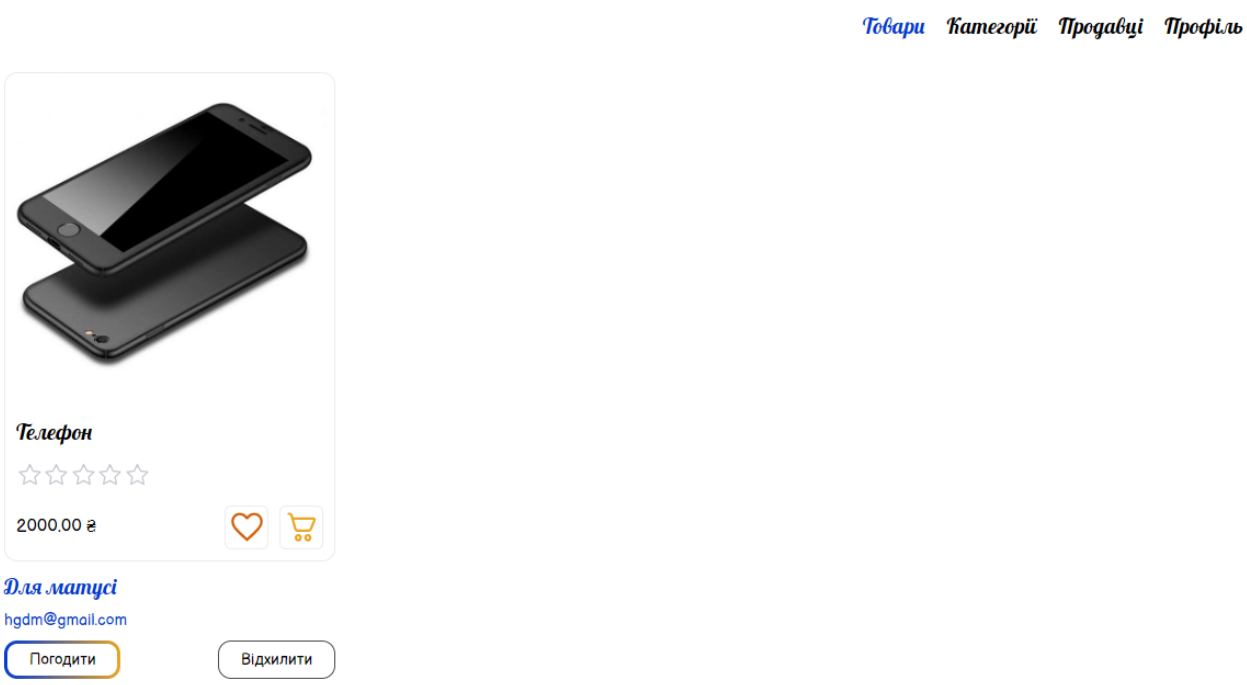


Рисунок 3.6. Сторінка «Товари» у панелі адміністратора.

Тут адміністратор може побачити нові товари, що завантажують продавці. Він може погодити товари, або навпаки відхилити, вказавши причину відмови.

2. Категорії

Рисунок 3.7. Сторінка «Категорії» у панелі адміністратора.

Тут адміністратор може відкрити кожну категорію, продивитись які товари їй належать. А також додавати, редагувати, видаляти категорії та редагувати приналежність товару до певної категорії.

3. Продавці

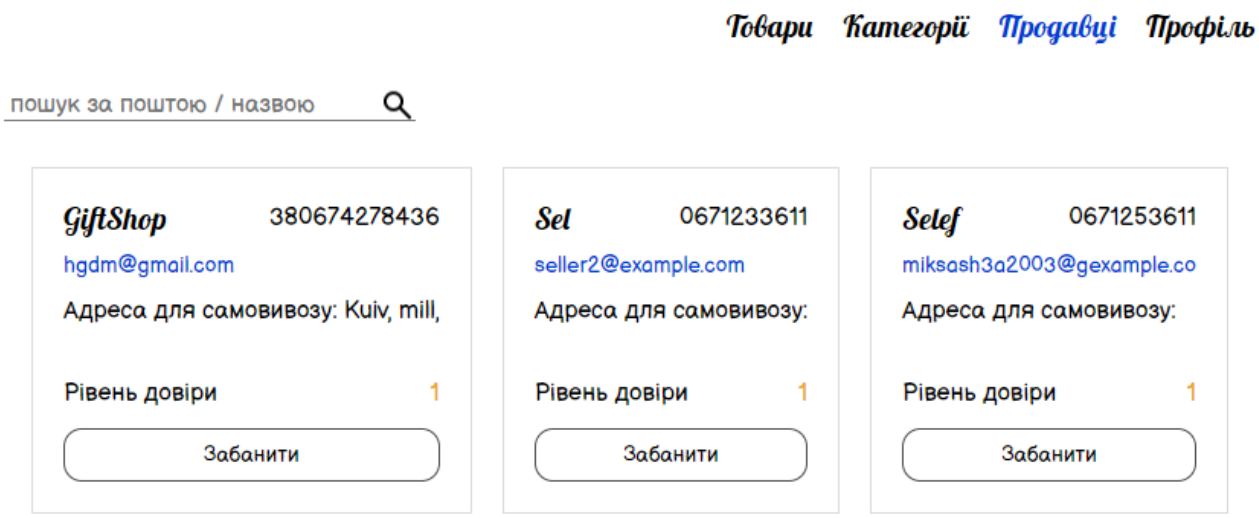


Рисунок 3.8. Сторінка «Продавці» у панелі адміністратора.

На даній вкладці адміністратор може керувати продавцями, зареєстрованими у системі. Він може користуватись пошуком по пошті та назві виробництва, а також банити профіль продавця, якщо він порушує правила.

4. Профіль

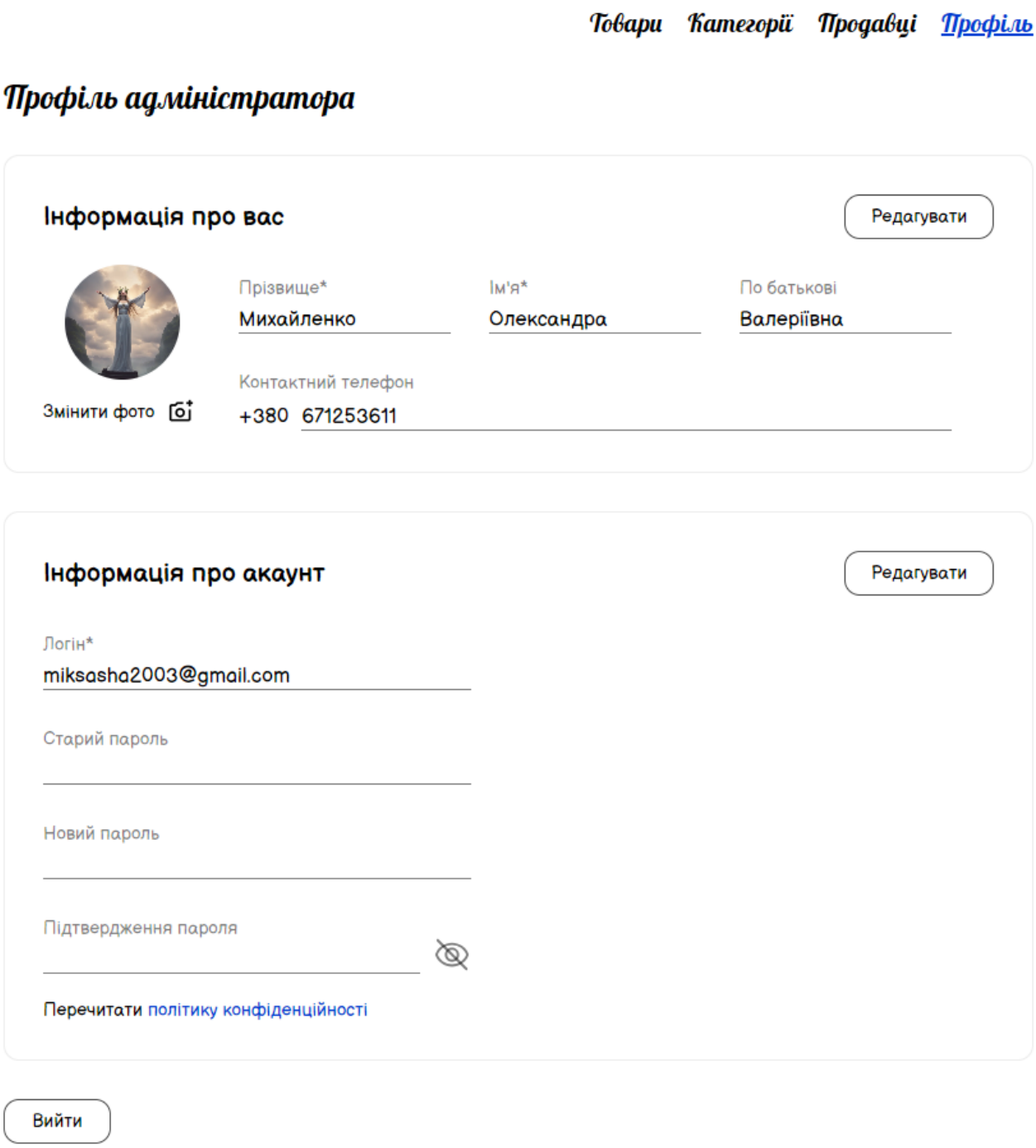


Рисунок 3.9. Сторінка «Профіль» у панелі адміністратора.

Де адміністратор може ввести особисту інформацію, поновити свій пароль, або змінити аватар.

Власник.

Ця роль у системі має всі ті самі права, що й адміністратор, але має додаткову можливість створювати акаунти нових адміністраторів:

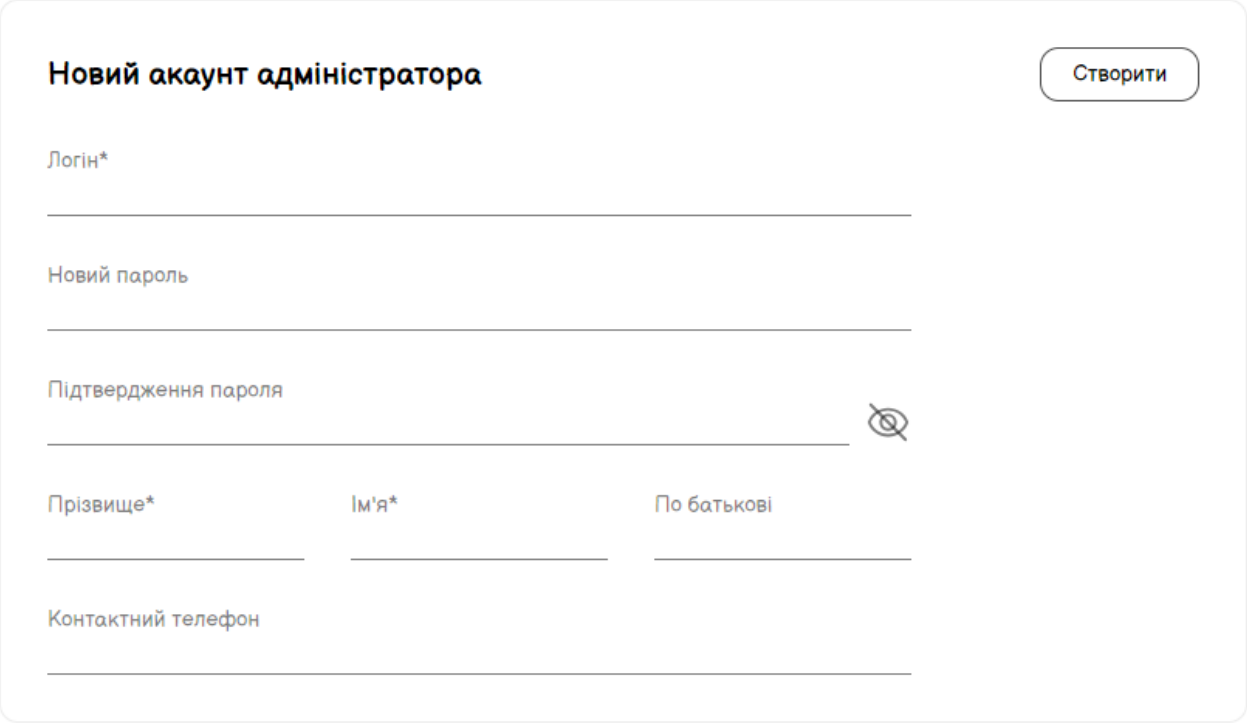


Рисунок 3.10. Створення акаунту нового адміністратора у панелі власника.

Продавець.

Панель продавця має теж 4 сторінки:

1. Мої товари

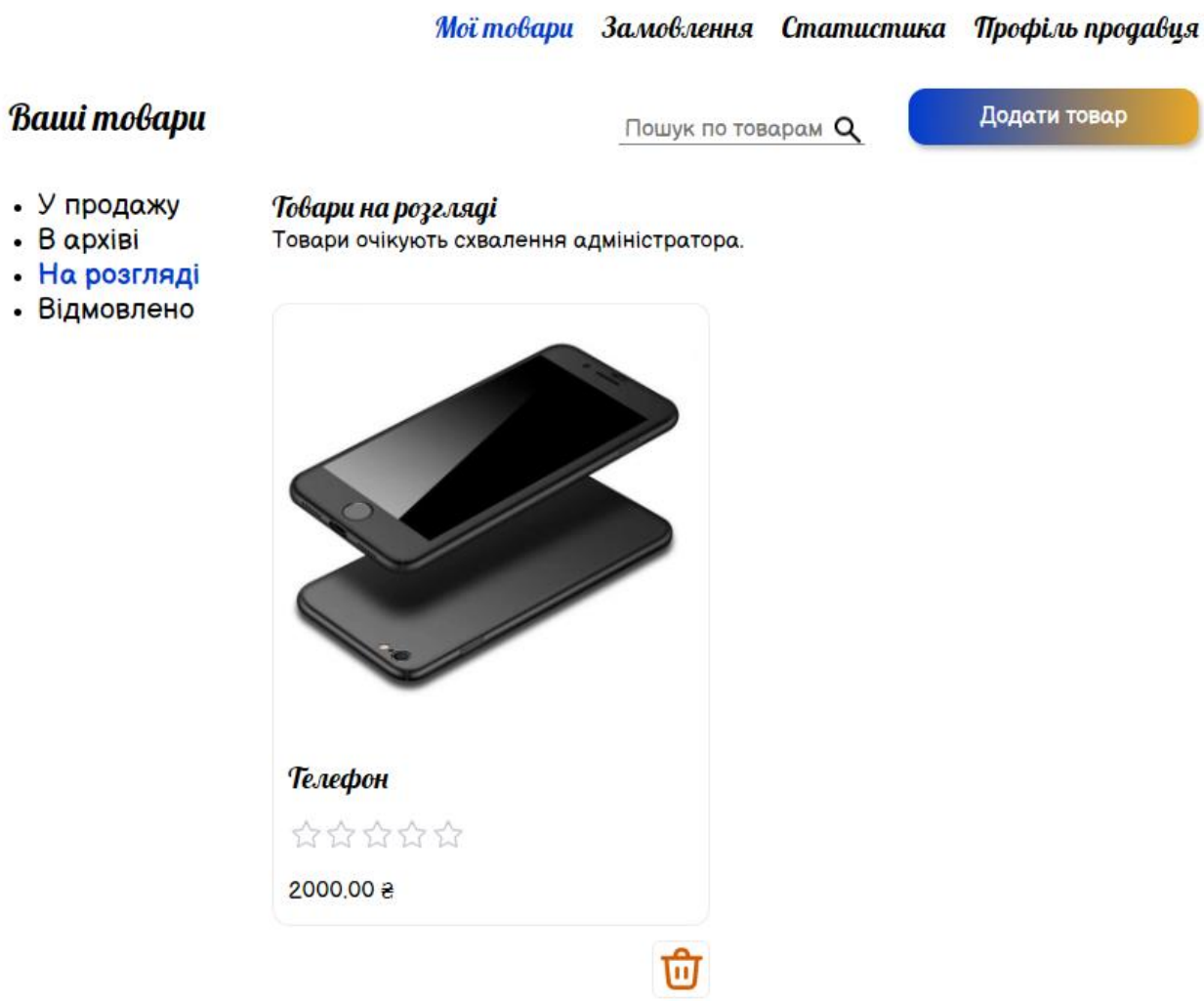
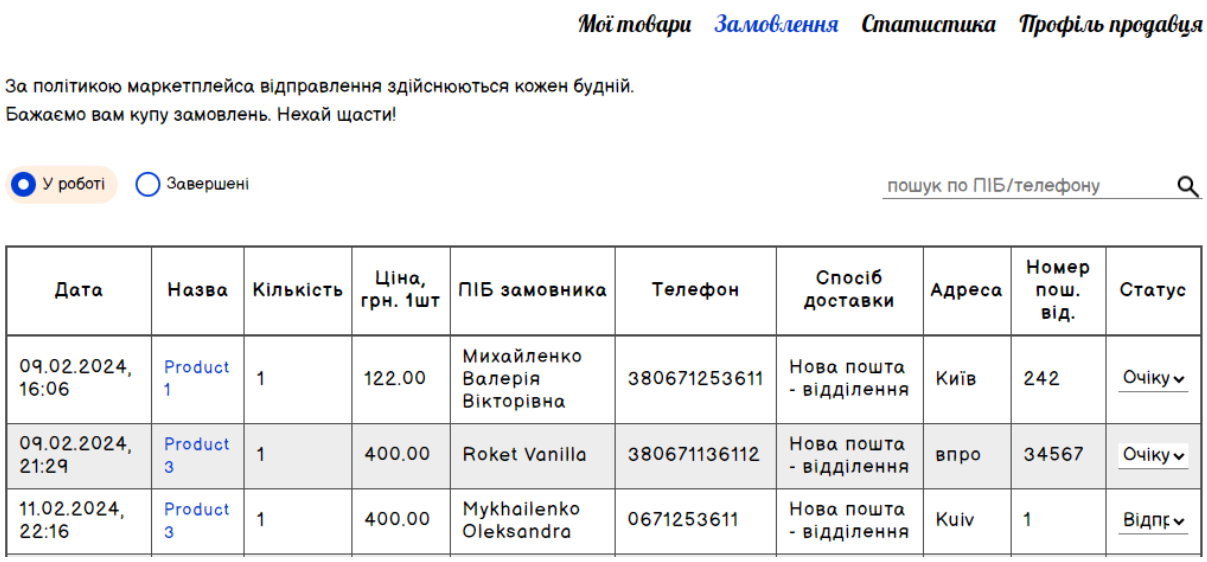


Рисунок 3.11. Сторінка «Мої товари» у панелі продавця.

У даній вкладці користувач може скористатись пошуком по товарах ввівши назву, або артикль товару. Є кнопка додавання нового товару, після натискання на неї продавець потрапляє у простір, де може ввести всю необхідну інформацію про товар, а також підвантажити від 3 до 10 фотографій у форматах jpeg, webp або png. Щоб зекономити місце на сервері, усі фотографії конвертуються у формат .webp пі час збереження картки товару. Такий функціонал допомагає не тільки зменшити об’єм займаної пам’яті, а ще й допомагає у SEO просуванні, адже до такого формату фотографій Google алгоритми є більш лояльними.

Ліворуч розташоване підменю, де користувач у ролі продавця може оглянути товари у різних статусах, а саме: «у продажі»(товари, що зараз можуть придбати покупці), «в архіві» (ці картки бачить тільки їх власник - продавець), «на розгляді» (товари, що чекають узгодження адміністратором) та «відмовлено»(картки, що були відхилені адміністратором).

2. Замовлено



Дата	Назва	Кількість	Ціна, грн. 1шт	ПІБ замовника	Телефон	Спосіб доставки	Адреса	Номер пош. від.	Статус
09.02.2024, 16:06	Product 1	1	122.00	Михайленко Валерія Вікторівна	380671253611	Нова пошта - відділення	Київ	242	Очіку
09.02.2024, 21:29	Product 3	1	400.00	Roket Vanilla	380671136112	Нова пошта - відділення	впро	34567	Очіку
11.02.2024, 22:16	Product 3	1	400.00	Mykhailenko Oleksandra	0671253611	Нова пошта - відділення	Kuiv	1	Відпр

Рисунок 3.12. Сторінка «Замовлення» у панелі продавця.

У цій вкладці продавець може подивитись нові замовлення, та змінити їх статус, якщо вони вже були відправлені клієнту. Також у верхній панелі зліва можна перемкнути у стан «Завершені», щоб подивитись історію своїх відправлень. У додачу, є можливість пошуку по ПІБ замовника та його номеру телефона.

3. Статистика

Назва	Ціна, грн	У наявності, шт	Хочуть купити	В обробці	Повернено	Продано
Product 1, Рожевий	122.00	1	4	4	0	2
Product 1, Помаранчевий		9				
Product 3	400.00	7	1	12	1	1
Product 2	120.00	0	1	0	0	0

Рисунок 3.13. Сторінка «Статистика» у панелі продавця.

Можна подивитись статистику по своїм товаром, що знаходяться у продажі зараз, а також відсортувати цю інформацію по будь-якому полю, натиснувши на назву колонки.

4. Профіль продавця

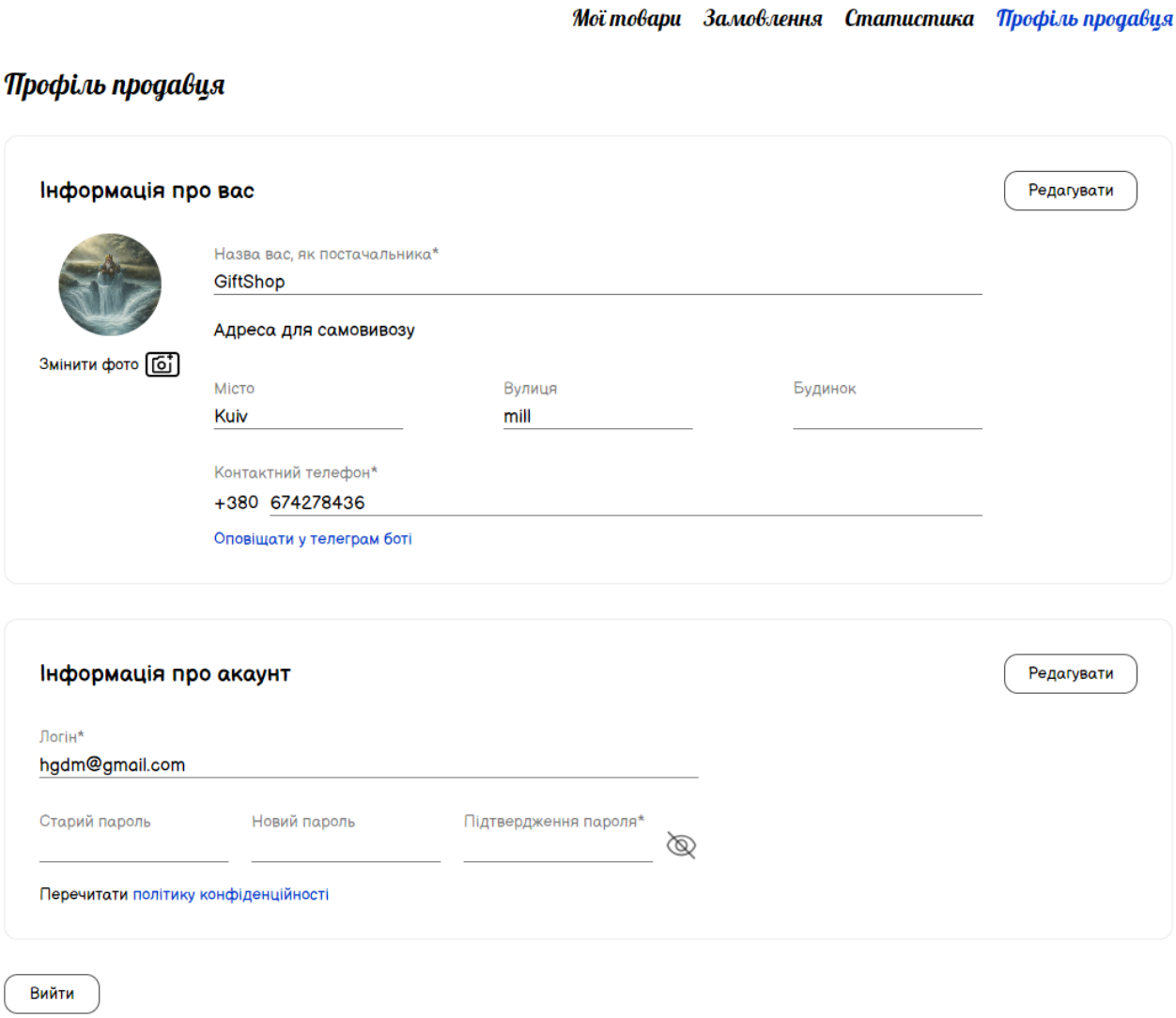


Рисунок 3.14. Сторінка «Профіль продавця» у панелі продавця.

На даній сторінці можна ввести інформацію про себе, а також налаштувати оповіщення через телеграм бот. У разі необхідності, є можливість змінити старий пароль.

Покупець.

На головній сторінці авторизованого покупця з’являється список рекомендованих для нього товарів. У не зареєстрованого користувача на цьому місці буде видно три найбільш популярних товарів тижня.

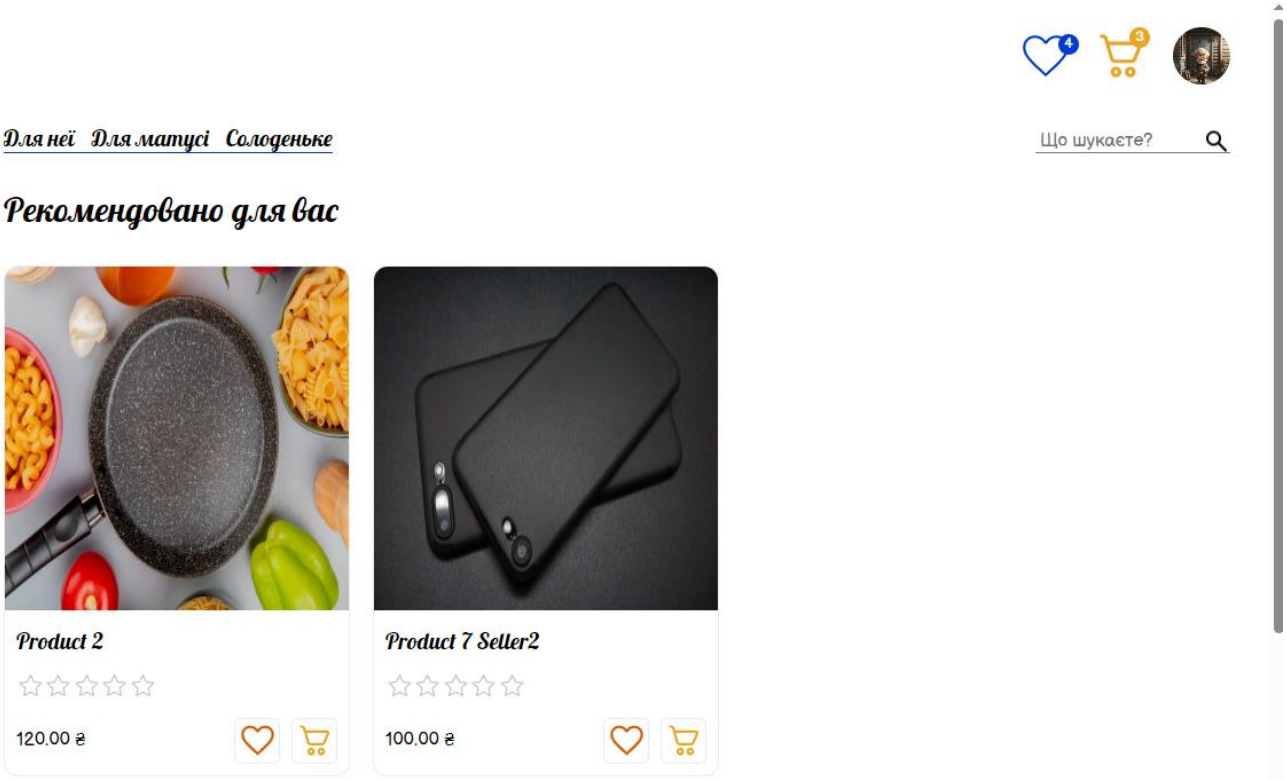


Рисунок 3.15. Головна сторінка у панелі покупця.

У покупця є вкладка побажань, де зберігаються всі товари, що сподобались користувачу, а також кошик:

Рисунок 3.16. Сторінка кошика у панелі покупця.

Натиснувши на кнопку «Оформити замовлення» (див. рис. 3.16.), користувач перейде у на сторінку де можна ввести всю необхідну інформацію для доставки і завершити процес замовлення.

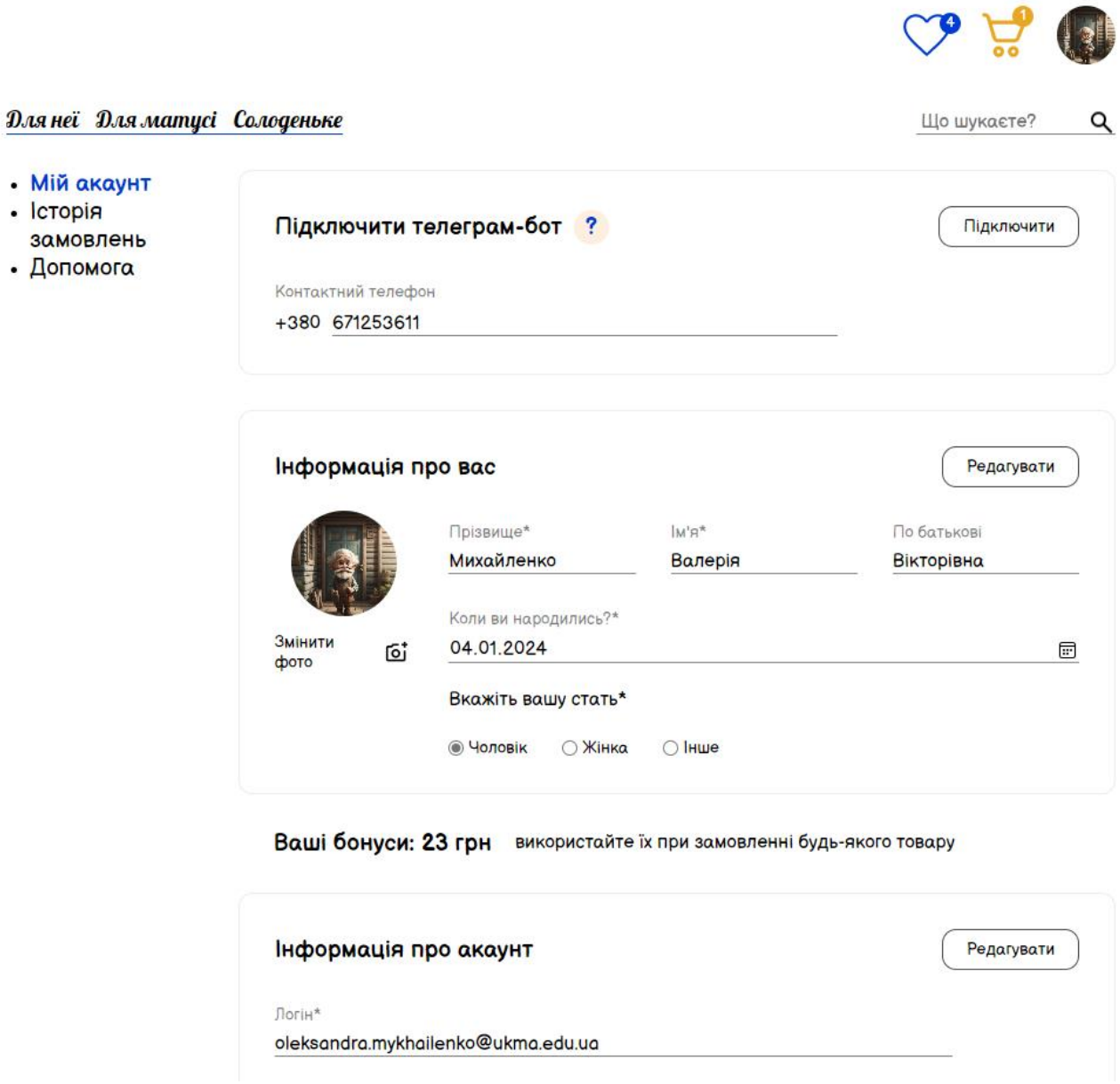


Рисунок 3.17. Сторінка «Профіль покупця» у панелі покупця.

На рис. 3.17. видно, що у профілі покупця, так само можна змінити вказану інформацію про себе, підключити телеграм бот для оповіщень, а також оновити

пароль, якщо є така необхідність. Ліворуч можна побачити підменю, завдяки якому є можливість перевірити історію своїх замовлень і звернутись у технічну підтримку, у разі виникнення складнощів.

Весь інтерфейс програми є адаптивним, тож підлаштовуються під будь-який необхідний розмір екрану.

Висновок

Щоб розробити якісний застосунок, що відповідатиме очікуванням і потребам замовника необхідно перш за все створити технічне завдання. Відповідно до нього розробляється архітектура майбутнього застосунку та підбирається стек технологій, що буде використовуватись під час розробки. У випадку поточного веб-додатку було обрано клієнт-серверну архітектуру. Для розробки використовувався PERN стек, що включає в себе:

- PostgreSQL – об’єктно-реляційна база даних, що підтримує структуровану мову запитів;
- Express – Node.js веб фреймворк, який в даному випадку обробляє серверну частину програми, забезпечуючи маршрутизацію, обробку HTTP запитів;
- React – JavaScript бібліотека, що відповідає за створення інтерфейсу користувача;
- Node - середовище виконання JavaScript для розробки веб-серверів.

Усі зазначені frontend та backend інструменти використовують мову програмування - JavaScript. Це спрощує налагодження передачі даних між сервером та інтерфейсом (PostgreSQL підтримує інтеграцію в усі мови програмування). На рис. 3.5. зображено схему взаємодії технологій PERN стеку.

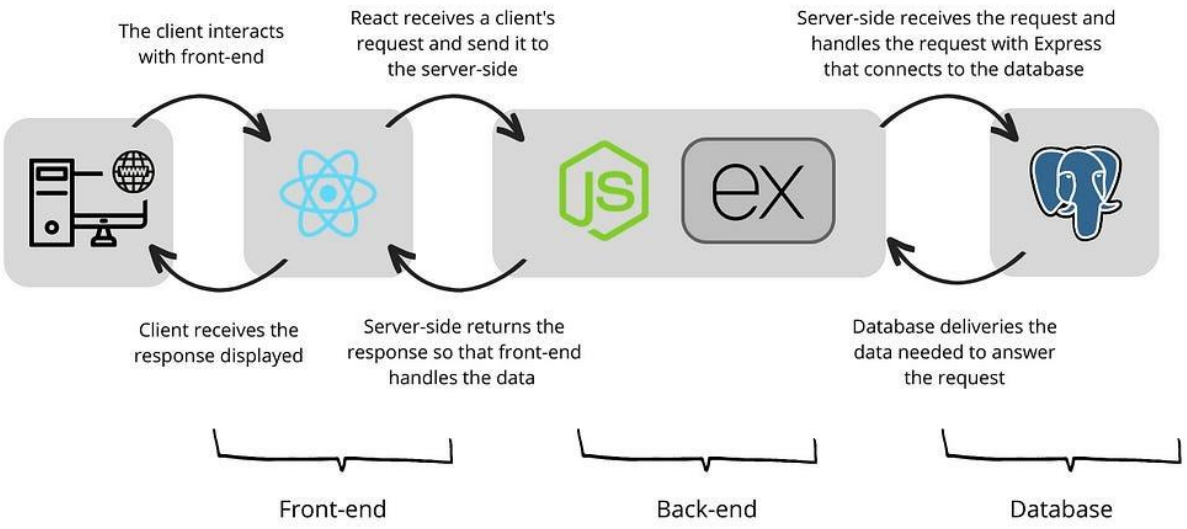


Рисунок 3.18. Схематичне зображення обміну даними у PERN стеку

Також, було проведено дослідження вже існуючих підходів до розробки рекомендаційних алгоритмів. Після аналізу, було запропоновано способи мінімізації слабких місць методу і розроблено відповідну систему.

Список використаних джерел

1. «THE ROLE OF E-COMMERCE IN STIMULATING INNOVATIVE BUSINESS DEVELOPMENT IN THE CONDITIONS OF EUROPEAN INTEGRATION». URL: https://d1wqtxts1xzle7.cloudfront.net/104496204/3930_330_340-libre.pdf?1690215852=&response-content-disposition=inline%3B+filename%3DTHE_ROLE_OF_E_COMMERCE_IN_STIMULATING_IN.pdf&Expires=1715075625&Signature=PM4ddG8JB5nqQ0XBvP6ezIVbQMmq6xLD~wzuF6FOGjcZX92s7ctV2j2sN8yX3JsuR3mSrVEilrPoucltUFoHZQMYK-

nlhflBZkYC8vKmX0Qkm~hbkUiwHYnw3x87CByBxXMpvqPfKW5ywQfeNK0wt33MeKXjeC2hNlBRPPdM0UrLYWo8YurUvypDCyh~V5-BQ2OyvbFiWMMBWKymyQxwKh7LG4sXRs5fyyEb5qMNkVlSlDvmEi0~OJ8~Foqt~T0DFFXloeoIbvZNO3Eigcvz60SxfOuhgQ~14DL8ICUb~eZ5pop4ETtdlaKXQVDzDPaINyiCq0gtoaHaHl7dz3cF5Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

2. Детальніше про умови рекламування у Rozetka. URL: https://sellerhelp.rozetka.com.ua/p324-advertising-and-promotion.html
3. Детальніше про умови рекламування у Prom. URL: https://support.prom.ua/hc/uk/articles/360004966437-%D0%94%D0%BE%D1%81%D1%82%D1%83%D0%BF-%D1%82%D0%B0-%D0%9F%D1%80%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D1%83-%D0%9A%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B7%D1%96-ProSale
4. «A Survey of Collaborative Filtering Techniques». URL: https://www.hindawi.com/journals/aai/2009/421425/
5. «Using Content-Based Filtering for Recommendation». URL: https://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf
6. «What is Client-Server System: Architecture, Issues and Challenge of Client - Server System». URL: https://www.researchgate.net/profile/Sawsan-Ali-6/publication/339340046_What_is_Client-Server_System_Architecture_Issues_and_Challenge_of_Client_-Server_System_Review/links/5e4c64c5458515072da88cd7/What-is-Client-Server-System-Architecture-Issues-and-Challenge-of-Client-Server-System-Review.pdf

7. «Micro frontend: Microservice implementation on Web Development». URL: https://www.theseus.fi/bitstream/handle/10024/511484/Son_Bui.pdf?sequence=2&isAllowed=y
8. «Express in Action: Writing, building, and testing Node.js applications». URL: https://books.google.com.ua/books?hl=uk&lr=&id=czkzEAAAQBAJ&oi=fnd&pg=PT13&dq=express+node&ots=8BiIQmBqk7&sig=z4p_Bvr9JEBXg-zj70EWSQfX9DU&redir_esc=y#v=onepage&q=express%20node&f=false
9. About PostgreSQL: https://www.postgresql.org/about/
10. «Amazon Web Services in Action, Third Edition: An In-depth Guide to AWS» URL: https://books.google.com.ua/books?hl=uk&lr=&id=joK3EAAAQBAJ&oi=fnd&pg=PA1&dq=aws+hosting&ots=DsNnPuatp_&sig=B2AX93oM_rga4feoZwhz8Kr0ycc&redir_esc=y#v=onepage&q=aws%20hosting&f=false

Список використаних зображень

2.1. https://www.kdnuggets.com/2019/11/content-based-recommender-using-natural-language-processing-nlp.html

2.2. https://www.kdnuggets.com/2019/11/content-based-recommender-using-natural-language-processing-nlp.html

3.1. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fdifference-between-client-server-and-distributed-dbms%2F&psig=AOvVaw2THXdKDL0VwbAArM8ubs4E&ust=1714589918625000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNCnnKzP6oUDFQAAAAAdAAAAABAI

3.2. https://books.google.com.ua/books?hl=uk&lr=&id=czkzEAAAQBAJ&oi=fnd&pg=PT13&dq=express+node&ots=8BiIQmBqk7&sig=z4p_Bvr9JEBXg-zj70EWSQfX9DU&redir_esc=y#v=onepage&q=express%20node&f=false

3.18. https://medium.com/@ritapalves/get-started-with-the-pern-stack-an-introduction-and-implementation-guide-e33c55d09994