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

НАВЧАЛЬНА ПЛАТФОРМА СИСТЕМИ ОРГАНІЗАЦІЇ РОБОТИ РЕПЕТИТОРІВ

Текстова частина до курсової роботи
за спеціальністю „Інженерія програмного забезпечення” 121

Керівник курсової роботи
кандидат фізико-математичних наук,
доцент Нагірна А. М.
_____
(підпис)
“____” _____ 2025 р.

Виконала студентка ІПЗ-3
Скіп Ю. Я.
“____” _____ 2025 р.

Київ 2025

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

ЗАТВЕРДЖУЮ
Зав. кафедри інформатики
доцент, к.ф-м.н.
_____ Гороховський С.С.
(підпис)
“____” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Скіп Юлії Ярославівні факультету інформатики 3 курсу
ТЕМА: Навчальна платформа системи організації роботи репетиторів

Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Вступ
1 Дослідження та аналіз предметної області
2 Розробка веб-застосунку для організації та планування роботи репетитора
Висновки
Список літератури

Дата видачі “____” _____ 2024 р. Керівник Нагірна А.М.
_____
(підпис)
Завдання отримала _____
(підпис)

Тема: Навчальна платформа системи організації роботи репетиторів

Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	10.10.2024	
2.	Дослідження репетиторської галузі	15.10.2024 – 15.11.2024	
3.	Аналіз існуючих застосунків для планування роботи репетиторів	15.11.2024 – 30.11.2024	
3.	Окреслення основних вимог до веб-застосунку	30.11.2024 – 07.12.2024	
4.	Проєктування бази даних	15.12.2024 – 20.12.2024	
5.	Створення інтерактивного прототипу веб-застосунку	21.12.2024 – 21.01.2025	
6.	Написання програмного коду	01.02.2025 – 04.05.2025	
7.	Написання текстової частини роботи	02.03.2025 – 04.05.2025	

Студент Скіп Ю.Я. _____

Керівник Нагірна А.М. _____

“_____”_____

ЗМІСТ

АНОТАЦІЯ **5**

ВСТУП **6**

РОЗДІЛ 1: ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ **8**

1.1 Дослідження актуальності теми та потреб користувачів 8
1.2 Аналіз існуючих платформ для організації та планування роботи репетитора 15
1.2.1 EasyWeek *16*
1.2.2 Google Calendar *17*
1.2.3 Notion *18*
1.2.4 План тижня та Тижневий планувальник *19*
1.3 Висновки до Розділу 1 20

РОЗДІЛ 2: РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ ТА ПЛАНУВАННЯ РОБОТИ РЕПЕТИТОРА. **21**

2.1 Постановка задачі 21
2.2 Проєктування та створення бази даних 22
2.2.1 Концептуальна ER-модель бази даних *22*
2.2.2 Реляційна модель бази даних *24*
2.2.3 Обґрунтування вибору системи керування базами даних PostgreSQL *27*
2.3 Створення інтерактивного прототипу веб-застосунку 28
2.3.1 Огляд сервісу Figma *28*
2.3.2 Опис створеного прототипу *28*
2.4 Реалізація програмного коду веб-застосунку 31
2.4.1 Опис серверної частини веб-застосунку *31*
2.4.2 Опис клієнтської частини веб-застосунку *35*
2.4.3 Опис реалізованого функціоналу *36*
2.4.4 Апробація реалізованого веб-застосунку *44*
2.5 Висновки до Розділу 2 45

ВИСНОВКИ **46**

СПИСОК ЛІТЕРАТУРИ **48**

Анотація

У роботі продемонстровано процес розробки веб-застосунку для організації та планування роботи репетитора. Описано та проаналізовано результати опитування, проведеного серед потенційних користувачів. Досліджено існуючі рішення і виявлено їхні недоліки. Розглянуто етапи створення веб-застосунку. Описано процес розробки бази даних, створення інтерактивного прототипу інтерфейсу користувача, побудову серверної та клієнтської частин. Мета роботи – створення комплексного рішення, яке змогло би задовільнити запити більшості репетиторів. Результатом є веб-застосунок, що містить модулі, необхідні для організації робочого процесу репетиторів, який підвищує ефективність їхньої роботи та покращує взаємодію з учнями.

Ключові слова: веб-застосунок, організація, планування, репетитор, база даних, інтерфейс, сервер, клієнт.

Вступ

У зв’язку із стрімким розвитком технологій попит на онлайн-навчання значно зріс, що призвело до вагомого збільшення кількості репетиторів на ринку праці.

На даний момент існує багато застосунків, які можуть слугувати засобами для полегшення роботи репетитора, проте вони покривають лише конкретні вузькі області цієї комплексної потреби. Одні допомагають із плануванням, інші зі збереженням матеріалів, обліком учнів або ж обліком доходів. Також існують завдання, засобів для вирішення яких ще немає в мережі. Через таку розосередженість репетитори витрачають багато часу для того, щоб зорієнтуватись в своїх записах та розпланувати або ж проаналізувати свою роботу.

Виходячи з потреби користувачів у зручному ресурсі, який міг би значно полегшити їхній досвід у організації та плануванні робочого процесу, та відсутності повних аналогів такого ресурсу у мережі, за мету даної роботи було поставлено детальний аналіз потреб користувачів, порівняння існуючих рішень та розробку комплексного веб-застосунку, який зміг би задовільнити запити більшості репетиторів.

Створення такого веб-застосунку забезпечить зменшення витраченого часу на процес планування та обліку, який займає левову частку робочого дня. Це допоможе репетиторам більше зосередитись на навчальному процесі, що, зокрема, покращить ефективність навчання для учнів. Додатковою перевагою слугуватиме збільшення кількості часу для відпочинку викладачів, що значно вплине на покращення їхнього загального стану здоров’я та надасть можливість більше уваги приділити собі та близьким, що у наш час є, безперечно, важливим та цінним.

Робота складається з двох розділів, кожен з яких містить у собі декілька підрозділів.

Перший розділ присвячено дослідженню потреб репетиторів та аналізу існуючих рішень для поставленої задачі. Наведено результати опитування потенційних користувачів та на його основі розглянуто застосунки, якими вже користуються опитані, описано їхні проблеми та побажання.

У другому розділі коротко сформовано постановку задачі та описано процес розробки веб-застосунку. Розглянуто процес проєктування та створення бази даних. Описано інтерактивний прототип інтерфейсу користувача та обґрунтовано вибір дизайнерських рішень. Наведено опис серверної та клієнтської частин веб-застосунку, та інструментів, використаних для їхнього створення. Підсумовано та проаналізовано реалізовані можливості. Описано результати апробації створеного веб-застосунку серед цільової аудиторії.

Створено програмний продукт, що містить у собі функціонал, запитаний потенційними користувачами, та призначений для організації та планування роботи репетитора.

РОЗДІЛ 1: Дослідження та аналіз предметної області

1.1 Дослідження актуальності теми та потреб користувачів

Щоб остаточно підтвердити актуальність та затребуваність розробки застосунку для організації та планування роботи репетитора, а також детальніше дослідити потреби потенційних користувачів, перед початком розробки додатково було проведено опитування серед людей, що працюють у репетиторській галузі.

Першим кроком було обрання способу проведення опитування та, зважаючи на специфіку роботи репетиторів та щільний графік, надано перевагу онлайн-опитуванню, яке є швидким, зручним, та не потребує особистої присутності учасників. Для його створення обрано сервіс Google Forms [1], який надає можливість створювати форми, що містять запитання різного типу, ділитись покликанням для їх проходження та переглядати статистику на основі відповідей.

Наступним кроком було формування питань, які допоможуть детальніше проаналізувати запити та настрої потенційних користувачів. [2] Основна увага зосереджена на дослідженні їхнього попереднього та поточного досвіду, а також на визначенні побажань щодо покращення робочого процесу.

Фінальним етапом було проведення опитування, в ньому взяли участь 20 репетиторів. Результати його проведення наведено нижче.

Питання №1. «Які способи Ви використовуєте для планування робочого дня?» (рисунок 1.1).

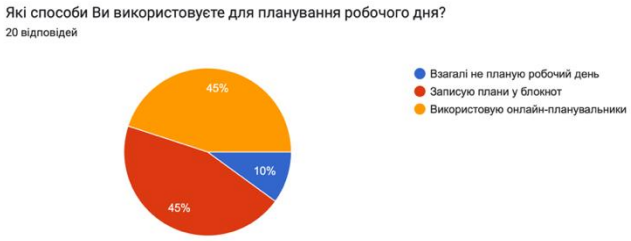


Рисунок 1.1 – Діаграма відповідей репетиторів на питання №1

Питання №2. «Які способи планування є найбільш зручними на Вашу думку?» (рисунок 1.2).

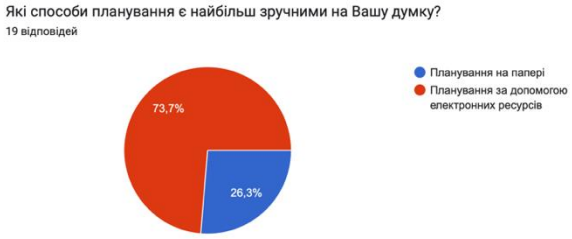


Рисунок 1.2 – Діаграма відповідей репетиторів на питання №2

Питання №3. «Які ресурси Ви використовуєте для планування? Чи є у них щось, що потребує покращення?» (рисунок 1.3), (рисунок 1.4).

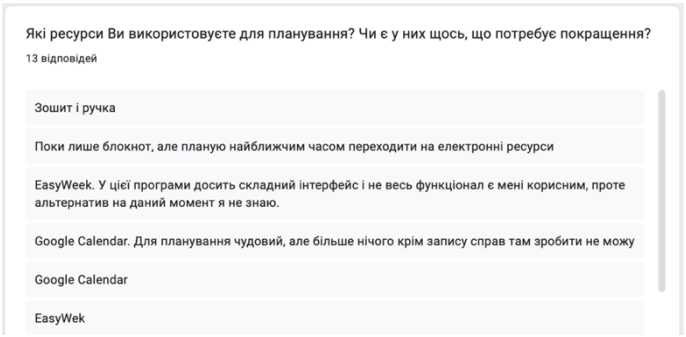


Рисунок 1.3 – Розгорнуті відповіді репетиторів на питання №3

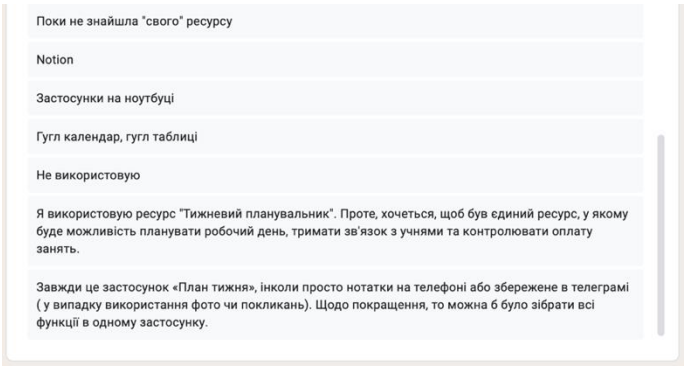


Рисунок 1.4 – Розгорнуті відповіді репетиторів на питання №3 (продовження)

Питання №4. «Як Ви ведете облік своїх учнів?» (рисунок 1.5).

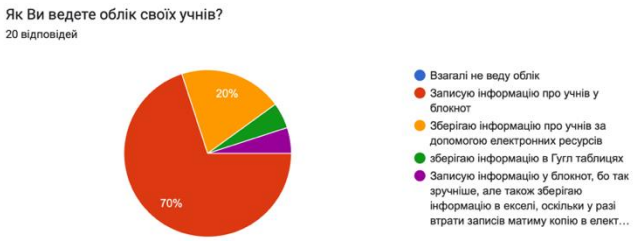


Рисунок 1.5 – Діаграма відповідей репетиторів на питання №4

Питання №5. «Що у процесі обліку учнів Вам здається незручним або таким, що потребує покращення?» (рисунок 1.6), (рисунок 1.7).

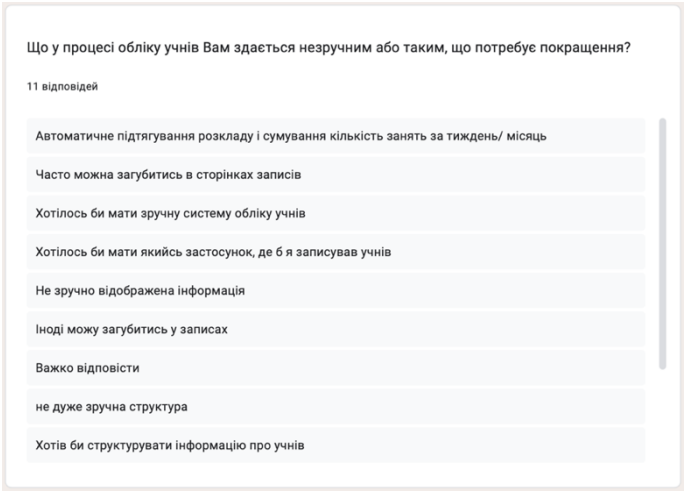


Рисунок 1.6 – Розгорнуті відповіді репетиторів на питання №5

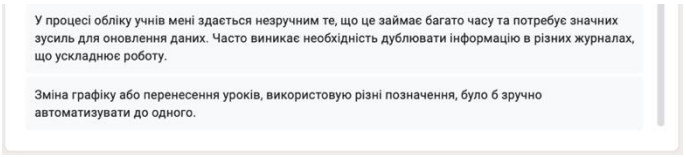


Рисунок 1.7 – Розгорнуті відповіді репетиторів на питання №5 (продовження)

Питання №6. «Як Ви відстежуєте свої доходи від репетиторської діяльності?» (рисунок 1.8).

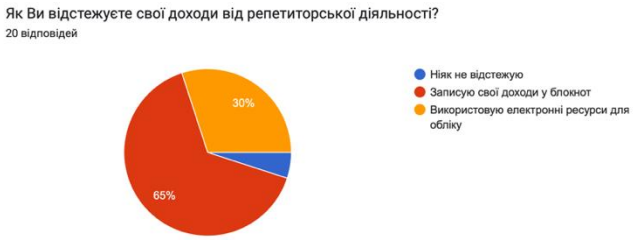


Рисунок 1.8 – Діаграма відповідей репетиторів на питання №6

Питання №7. «Чи виникали у Вас труднощі під час ведення обліку доходів? На Вашу думку, як можна їх уникнути?» (рисунок 1.9), (рисунок 1.10).

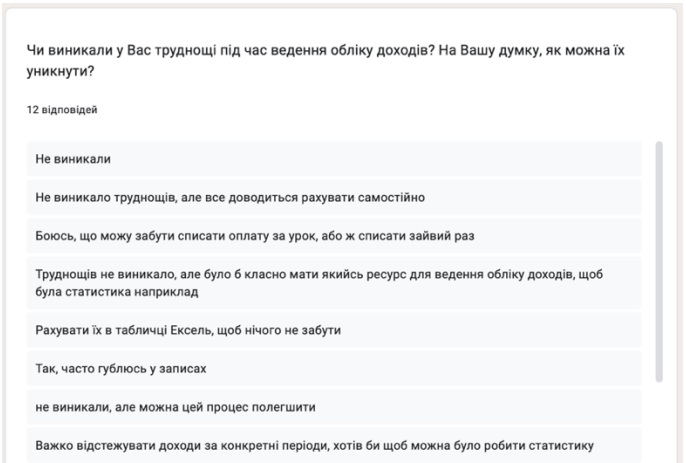


Рисунок 1.9 – Розгорнуті відповіді репетиторів на питання №7

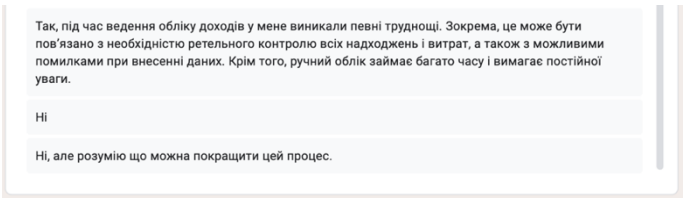


Рисунок 1.10 – Розгорнуті відповіді репетиторів на питання №7 (продовження)

Питання №8. «Як Ви задаєте домашні завдання учням?» (рисунок 1.11).

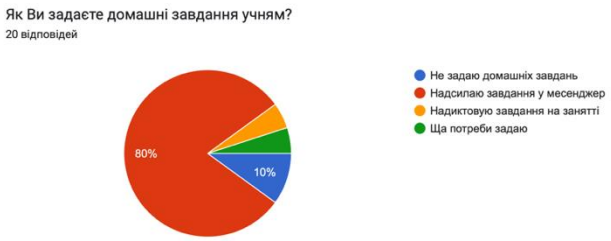


Рисунок 1.11 – Діаграма відповідей репетиторів на питання №8

Питання №9. «Чи є у Вас побажання щодо спрощення або покращення процесу надсилання та перевірки домашніх завдань?» (рисунок 1.12), (рисунок 1.13).

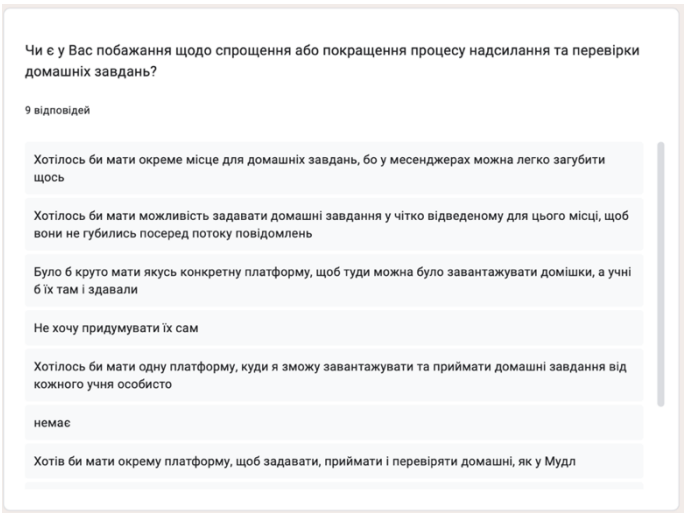


Рисунок 1.12 – Розгорнуті відповіді репетиторів на питання №9

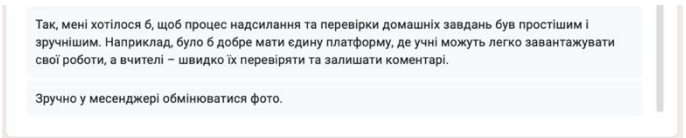


Рисунок 1.13 – Розгорнуті відповіді репетиторів на питання №9 (продовження)

Питання №10. «Яким чином Ви зберігаєте та впорядковуєте навчальні матеріали?» (рисунок 1.14).

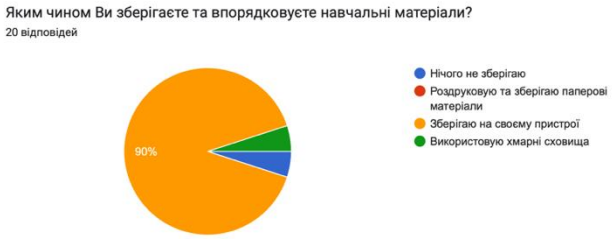
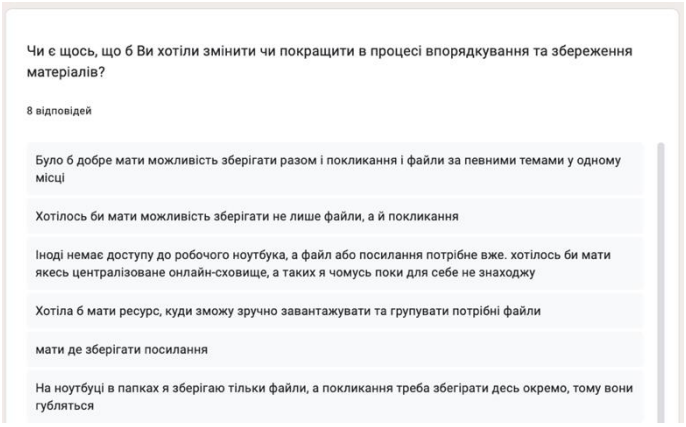


Рисунок 1.14 – Діаграма відповідей репетиторів на питання №10

Питання №11. «Чи є щось, що б Ви хотіли змінити чи покращити в процесі впорядкування та збереження матеріалів?» (рисунок 1.15), (рисунок 1.16).



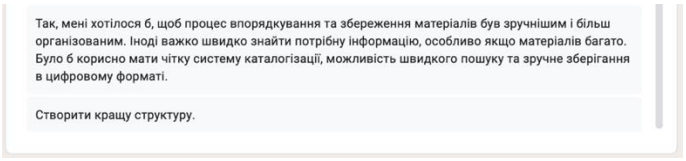


Рисунок 1.15 – Розгорнуті відповіді репетиторів на питання №11

Рисунок 1.16 – Розгорнуті відповіді репетиторів на питання №11(продовження)

Питання №12. «Які аспекти роботи Ви б хотіли автоматизувати або оптимізувати за допомогою цифрових інструментів?» (рисунок 1.17), (рисунок 1.18).

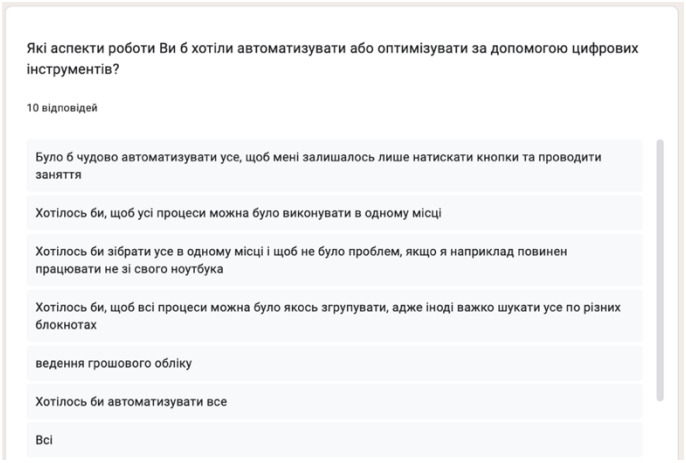


Рисунок 1.17 – Розгорнуті відповіді репетиторів на питання №12

Рисунок 1.18 – Розгорнуті відповіді репетиторів на питання №12 (продовження)

Отже, підсумувавши, можна зробити висновки, що існуючі методи організації роботи репетиторів мають низку недоліків та не повністю задовольняють потреби користувачів. Незважаючи на те, що багато респондентів використовують електронні планувальники, зокрема «EasyWeek» [3] , «Google Calendar»[4], «Notion»[5], «План тижня»[6] та «Тижневий планувальник»[7], значна частина все ще віддає перевагу паперовим блокнотам. Водночас більшість зазначає, що наявні цифрові інструменти не завжди відповідають їхнім потребам.

Одними з ключових проблем, з якими стикаються користувачі, є труднощі в обліку учнів та плануванні занять. Репетитори використовують різні методи, проте їм бракує єдиного зручного інструменту для структурованого збереження інформації. Схожу ситуацію можна побачити і з обліком доходів – більшість веде його вручну, що забирає багато часу та несе низку загроз як щодо втрати даних, так і щодо порушення їхньої

несуперечливості. Ще одним важливим аспектом роботи репетиторів є надсилання та перевірка домашніх завдань. Більшість користується месенджерами, однак такий підхід не дозволяє централізовано керувати процесом. Водночас опитані висловили бажання мати платформу, яка б спрощувала надсилання, прийом та перевірку завдань. Наступним болючим місцем учасників опитування є збереження та впорядкування навчальних матеріалів. Репетитори переважно зберігають матеріали на своїх пристроях, що несе за собою низку незручностей та загрозу втрати даних. Вони зазначають, що хотіли б мати платформу для збереження матеріалів із можливістю організації файлів та покликань й швидкого доступу до них з будь-якого пристрою.

Загалом, значна частина опитаних висловила бажання автоматизувати більшість робочих процесів, що дозволило б їм заощаджувати час та підвищити ефективність роботи. Результати опитування підтверджують актуальність розробки застосунку для організації та планування роботи репетиторів, який об’єднає ключові функції в одному місці та зробить процес підготовки до занять швидшим та простішим.

1.2 Аналіз існуючих платформ для організації та планування роботи репетитора

Проведене опитування підтвердило актуальність нашої розробки, вказавши на нагальні потреби та труднощі, з якими кожного дня стикаються опитані нами репетитори. Однак, щоб створити дійсно ефективний та зручний інструмент, необхідно проаналізувати вже наявні рішення, щоб зрозуміти їхні сильні та слабкі сторони. Саме тому було проведено аналіз платформ, якими вже користуються опитані нами репетитори а саме: «EasyWeek», «Google Calendar», «Notion», «План тижня» та «Тижневий планувальник». Під час

аналізу увагу звернено на такі аспекти як інтерфейс, наявність необхідного функціоналу, та зручність користування.

1.2.1 EasyWeek

Першим для аналізу обрано ресурс «EasyWeek» (рисунок 1.19). Як зазначено на офіційному сайті, «EasyWeek» – це «Практична система CRM та додаток для онлайн-запису, що допомагає репетиторам керувати розкладом та бронюванням занять».

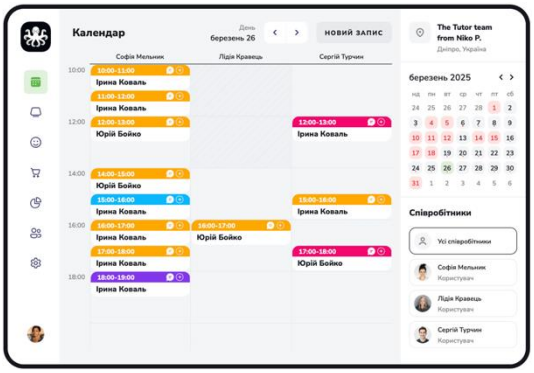


Рисунок 1.19 – Приклад інтерфейсу застосунку «EasyWeek»

Перш за все варто зазначити, що інтерфейс є досить простим та привабливим. Кольори приємні для сприйняття, шрифти чіткі та читабельні, елементи прості та логічно впорядковані. Щодо наявності функціоналу, який є необхідним, на думку, опитаних раніше репетиторів варто зазначити, що присутні такі можливості, як:

– можливість планування робочого дня та занять;
– можливість ведення обліку учнів;
– можливість ведення грошового обліку (лише для преміум версії);

Водночас відсутні такі функції, як:

– можливість надсилання, прийому та перевірки домашніх завдань;
– наявність централізованого сховища для вкладень;

– можливість перегляду статистики проведених занять;

З усіх вищевказаних застосунків «EasyWeek» найбільше підходить під концепцію, описану опитаними репетиторами. Зручність користування цим застосунком можна оцінити як середню. Інтерфейс хоч і досить зрозумілий, проте містить багато складних елементів та функцій, які часто важко знайти. Враховуючи, що репетиторами працюють люди з різними рівнями вміння користування електронними ресурсами, функціонал таких застосунків має бути більш простим.

1.2.2 Google Calendar

Наступним проаналізовано «Google Calendar» (рисунок 1.20). Насамперед слід зазначити, що «Google Calendar» не був створений спеціально для репетиторів, а є універсальним інструментом для планування різних завдань.

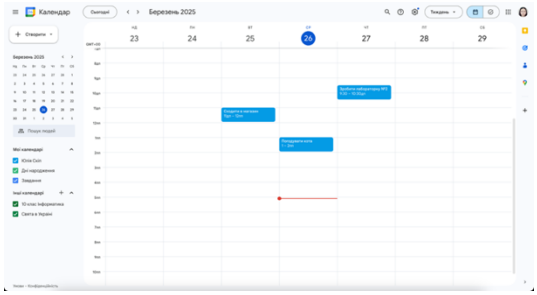


Рисунок 1.20 – Приклад інтерфейсу застосунку «Google Calendar»

Інтерфейс цього ресурсу є максимально простим. Кольори приємні, шрифти чіткі та читабельні, елементи логічно впорядковані. Щодо функціоналу, то як вже було зазначено вище, «Google Calendar» створений виключно для планування, тому він покриває лише один аспект потреб користувачів – планування робочого дня. Оскільки, опитані репетитори

висловили потребу в ресурсі, який об’єднував би всі їхні щоденні завдання, то, хоча «Google Calendar» і є хорошим інструментом, у цьому випадку він не підходить для комплексного вирішення задач репетиторів. Зручність користування ним, як планувальником є максимально високою, проте через інші потреби репетиторів він задовольняє їх лише частково.

1.2.3 Notion

Далі розглянуто ресурс «Notion» (рисунок 1.21) та оцінено його відповідність потребам репетиторів. «Notion» – це застосунок для ведення нотаток, побудови систем управління та організації життя.

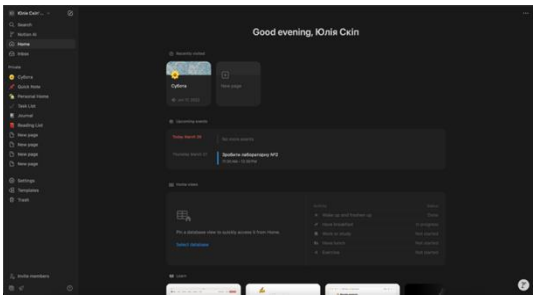
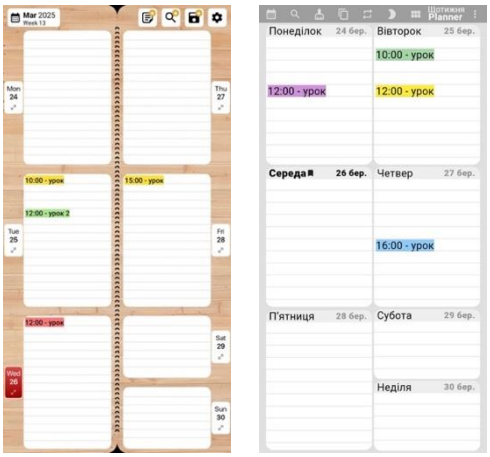


Рисунок 1.21 – Приклад інтерфейсу застосунку «Notion»

Інтерфейс є дуже простим, привабливим та інтуїтивно зрозумілим. Кольори нейтральні, шрифти чіткі. Основною перевагою цього ресурсу є можливість кастомізації власних записів, що дає простір для творчості користувачів. Щодо функціоналу, то це знову ж таки лише планувальник, хоча, враховуючи вищезазначену можливість кастомізації, тут можна не лише планувати робочий день, а й вести облік доходів та учнів. Однак виконання таких процесів, як організація процесу надсилання домашніх завдань, створення бібліотеки вкладень та ведення статистики, у цьому застосунку неможливе. Загалом «Notion» є зручним у користуванні, проте, враховуючи запити репетиторів, він не повністю відповідає їхнім потребам для комплексної організації та планування роботи.

1.2.4 План тижня та Тижневий планувальник

Наступними для огляду обрано одразу два мобільні застосунки – «План тижня» (рисунок 1.22 (а)) та «Тижневий планувальник». (рисунок 1.22 (б)) Ми аналізуватимемо їх разом, оскільки вони дуже схожі за функціоналом.



а *б*

Рисунок 1.22 – Приклад інтерфейсів застосунків «План тижня»(а) та «Тижневий планувальник»(б)

Інтерфейси цих двох застосунків схожі, проте у «Тижневому планувальнику» він простіший та водночас зручніший завдяки збільшеному масштабу елементів та кращому розташуванню тексту. Структура елементів в обох додатках досить логічна, шрифти читабельні та чіткі. Щодо функціоналу, як і в «Notion», тут можна лише додавати нотатки. Однак, на відміну від попереднього, ці застосунки не мають широких можливостей кастомізації, що унеможливлює одночасне ведення обліку учнів чи фінансів. Загалом ці додатки є досить зручними у використанні, але їхній функціонал недостатній, щоб мати змогу повністю організувати роботу репетитора.

1.3 Висновки до Розділу 1

Опираючись на проведене опитування, ми дослідили, що багато репетиторів досі користуються традиційними методами планування, а саме – плануванням на папері. Проте, значна частина із них незадоволена таким способом, але через відсутність спеціалізованих застосунків, не має іншого виходу. Інша частина репетиторів, що все-таки користується електронними ресурсами для організації своєї роботи, також має зауваження до них та хотіла б покращити цей процес. Ми проаналізували застосунки, якими уже користуються опитані репетитори та визначили, що жоден з них повністю не відповідає їхнім вимогам. Зважаючи на ці факти, ми довели актуальність розробки застосунку для планування та організації роботи репетиторів.

Розділ 2: Розробка веб-застосунку для організації та планування роботи репетитора.

2.1 Постановка задачі

В попередньому розділі розглянуто актуальність розробки та визначено потреби потенційних користувачів. Проаналізувавши відповіді репетиторів, було сформовано ключові вимоги, щодо функціоналу веб-застосунку, а саме:

– планування робочого дня;
– ведення обліку учнів;
– перегляд розкладу учнів;
– ведення обліку доходів;
– централізована задача, прийом та перевірка домашніх завдань;
– централізоване збереження навчальних матеріалів;
– перегляд статистики на основі доходів та проведених занять.

Отже, вище вказані вимоги необхідно реалізувати в наступні модулі програми:

– Сторінка реєстрації. Користувачам надаватиметься можливість створити власні профілі для подальшого керування процесом організації та планування роботи.

– Сторінка входу. У випадку наявності власного профілю у нашому веб-застосунку, користувачі матимуть змогу увійти у систему, використовуючи вказані при реєстрації електронну адресу або ім’я користувача та пароль.

– Профіль користувача, який включатиме особисті дані та дані для входу у систему. Користувачу надаватиметься змога переглядати та редагувати усю інформацію.

– Сторінка зі списком учнів. Користувач матиме можливість переглядати повний список своїх учнів та інформацію про них, їхній розклад та історію оплат, виконувати операції зі збереженими даними.

– Модуль для планування занять. Користувачу надаватиметься функціонал для планування робочого процесу.

– Модуль для керування домашніми завданнями. Користувачу надаватиметься повний список його учнів та можливість керування домашніми завданнями для кожного учня окремо.

– Сторінка зі списком справ. Користувач матиме змогу у даному веб-застосунку планувати не лише заняття, а й повсякденні справи у межах поточного місяця.

– Особиста бібліотека. Користувач зможе організовувати необхідні файли та покликання за власними категоріями.

– Сторінка статистики. Користувач матиме можливість переглядати статистику доходів та проведених занять.

2.2Проєктування та створення бази даних

Однією з ключових вимог до будь-якого застосунку є збереження його стану між сеансами. Найбільш зручним способом для досягнення цієї мети є використання баз даних. Завдяки ним можна структуровано зберігати інформацію, необхідну для роботи додатка, а також редагувати, видаляти та завантажувати її для подальшого відображення.

2.2.1 Концептуальна ER-модель бази даних

Важливим кроком у проєктуванні бази даних є створення концептуальної моделі «сутність-зв’язок» (ER-моделі) [8], яка допомагає краще зрозуміти предметну область, візуалізувавши дані, які ми будемо зберігати, та зв’язки між ними. Опираючись на функціонал, описаний у

постановці задачі, було проведено аналіз того, яка саме інформація має зберігатись у нашій базі, та визначено сутності, які будуть представленими у моделі, а саме:

– Tutor (репетитор) – сутність, що містить деталі про користувача.
– Student (учень) – сутність, що містить деталі про учня.
– Event (подія) – сутність, що містить деталі, про заплановане заняття.
– Homework (домашнє завдання) – сутність, що містить інформацію про домашнє завдання.
– Task (завдання) – сутність, що містить деталі про повсякденні справи користувача.
– Attachment (вкладення) - сутність, що містить файли або покликання, збережені користувачем.
– Payment (оплата) - сутність, що містить деталі про оплату за заняття.
– PasswordReset (зміна паролю) – допоміжна сутність, що містить інформацію для зміни паролю користувача у разі, якщо він його забув.

Для створення концептуальної ER-моделі використано інструмент draw.io [9], побудовано усі сутності, вказано їхні ключі, атрибути та відношення між ними (рисунок 2.1).

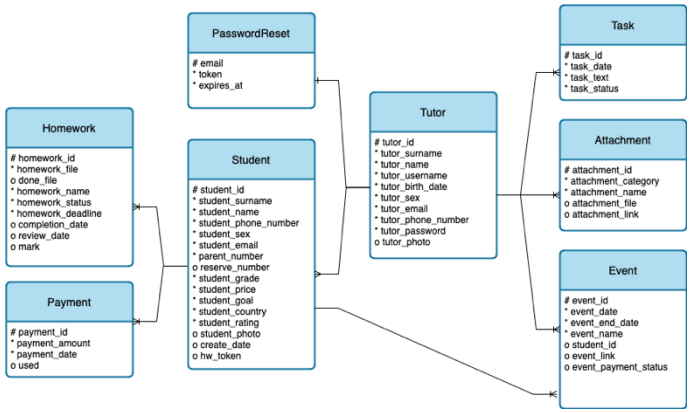


Рисунок 2.1 – Концептуальна ER-модель бази даних

2.2.2 Реляційна модель бази даних

Після побудови концептуальної ER-моделі, її було переведено у реляційну модель [10], яка є ближчою до практичного представлення в системі керування базами даних.

Таблиця «Tutors» (таблиця 2.1) відповідає типу сутності «Tutor».

Назва атрибуту	**Тип**	**NULL/NOT NULL**	**FK: ON DELETE, ON UPDATE**
PK tutor_id	INT	NOT NULL	
tutor_surname	VARCHAR(30)	NOT NULL	
tutor_name	VARCHAR(30)	NOT NULL	
tutor_username	VARCHAR(30)	NOT NULL	
tutor_birth_date	DATE	NOT NULL	
tutor_sex	VARCHAR(6)	NOT NULL	
tutor_email	VARCHAR(50)	NOT NULL	
tutor_phone_number	VARCHAR(13)	NOT NULL	
tutor_password	VARCHAR(30)	NOT NULL	
tutor_photo	VARCHAR(200)	NULL	

Таблиця 2.1 – таблиця «Tutors»

Таблиця «Students» (таблиця 2.2) відповідає типу сутності «Student».

Назва атрибуту	**Тип**	**NULL/NOT NULL**	**FK: ON DELETE, ON UPDATE**
PK student_id	INT	NOT NULL	
FK tutor_id	INT	NOT NULL	CASCADE, CASCADE
student_surname	VARCHAR(30)	NOT NULL	
student_name	VARCHAR(30)	NOT NULL	
student_phone_number	VARCHAR(13)	NOT NULL	
student_sex	VARCHAR(6)	NOT NULL	
student_email	VARCHAR(50)	NOT NULL	
parent_number	VARCHAR(13)	NOT NULL	
reserve_number	VARCHAR(13)	NULL	
student_grade	INT	NOT NULL	
student_price	INT	NOT NULL	
student_goal	VARCHAR(50)	NOT NULL	
student_country	VARCHAR(50)	NOT NULL	
student_rating	VARCHAR(50)	NOT NULL	
student_photo	VARCHAR(200)	NULL	
create_date	DATE	NOT NULL	
hw_token	VARCHAR(200)	NOT NULL	

Таблиця 2.2 – таблиця «Students»

Таблиця «Events» (таблиця 2.3) відповідає типу сутності «Event».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK event_id	INT	NOT NULL	
FK tutor_id	INT	NOT NULL	CASCADE, CASCADE
FK student_id	INT	NULL	SET NULL, CASCADE
event_date	DATE	NOT NULL	
event_end_date	DATE	NOT NULL	
event_name	VARCHAR(200)	NOT NULL	
event_link	VARCHAR(200)	NULL	
payment_status	BOOLEAN	NULL	

Таблиця 2.3 – таблиця «Events»

Таблиця «Homeworks» (таблиця 2.4) відповідає типу сутності «Homework».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK homework_id	INT	NOT NULL	
FK student_id	INT	NOT NULL	CASCADE, CASCADE
homework_file	VARCHAR(200)	NOT NULL	
done_file	VARCHAR(200)	NULL	
homework_name	VARCHAR(50)	NOT NULL	
homework_status	VARCHAR(50)	NOT NULL	
homework_deadline	DATE	NOT NULL	
completion_date	DATE	NULL	
review_date	DATE	NULL	
mark	INT	NULL	

Таблиця 2.4 – таблиця «Homeworks»

Таблиця «Tasks» (таблиця 2.5) відповідає типу сутності «Task».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK task_id	INT	NOT NULL	
FK tutor_id	INT	NOT NULL	CASCADE, CASCADE
task_date	DATE	NOT NULL	
task_text	VARCHAR(200)	NOT NULL	
task_status	BOOLEAN	NOT NULL	

Таблиця 2.5 – таблиця «Tasks»

Таблиця «Attachments» (таблиця 2.6) відповідає типу сутності «Attachment».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK attachment_id	INT	NOT NULL	
FK tutor_id	INT	NOT NULL	CASCADE, CASCADE
attachment_category	VARCHAR(50)	NOT NULL	
attachment_name	VARCHAR(50)	NOT NULL	
attachment_file	VARCHAR(200)	NULL	
attachment_url	VARCHAR(200)	NULL	

Таблиця 2.6 – таблиця «Attachments»

Таблиця «Payments» (таблиця 2.7) відповідає типу сутності «Payment».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK payment_id	INT	NOT NULL	
FK student_id	INT	NULL	SET NULL, CASCADE
payment_amount	INT	NOT NULL	
payment_date	DATE	NOT NULL	
used	INT	NULL	

Таблиця 2.7 – таблиця «Payments»

Допоміжна таблиця «PasswordResets» (таблиця 2.8) відповідає типу сутності «PasswordReset».

Назва атрибуту	Тип	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK tutor_email	VARCHAR(50)	NOT NULL	CASCADE, CASCADE
token	INT	NOT NULL	
expires_at	DATE	NOT NULL	

Таблиця 2.8 – таблиця «PasswordResets»

2.2.3 Обґрунтування вибору системи керування базами даних PostgreSQL

На сьогодні існує багато різних систем керування базами даних (СКБД), кожна з яких має свої переваги, недоліки та оптимальні сценарії використання. Вибір конкретної СКБД залежить від потреб застосунку, вимог до продуктивності, безпеки та масштабованості.

У нашому випадку було обрано PostgreSQL [11] – об’єктно-реляційну систему керування базами даних. Такий вибір зумовлений тим, що ця система не тільки підтримує велику кількість стандартів SQL, а й має багато переваг, зокрема:

- підтримка великої кількості вбудованих типів даних;
- можливість виконання складних SQL запитів;
- можливість створення тригерів для автоматизованого керування даними;
- підтримка одночасної модифікації БД декількома користувачами, за рахунок механізму Multiversion Concurrency Control (MVCC);
- висока продуктивність та масштабованість у великих проєктах;
- зручна інтеграція з сучасними вебтехнологіями за допомогою великої кількості клієнтських бібліотек.

2.3 Створення інтерактивного прототипу веб-застосунку

Прототипування є дуже важливим етапом у створенні цифрового продукту. За допомогою інтерактивного прототипу можна не лише визначити зовнішній вигляд застосунку, а й змоделювати та протестувати його функціональність ще до початку розробки. Основною перевагою прототипування є можливість виявити й виправити недоліки на ранньому етапі, що дозволяє заощадити значну кількість часу та ресурсів.

2.3.1 Огляд сервісу Figma

Для розробки інтерактивного прототипу інтерфейсу користувача було обрано сервіс Figma [12], який надає можливість створювати векторні дизайни сторінок застосунку та додавати інтерактивні переходи між ними, щоб відтворити шлях користувача.

Цей сервіс є безкоштовним для індивідуальних користувачів та пропонує широкий спектр можливостей у дизайні. Figma підтримує велику кількість шрифтів та різні колірні схеми, дає можливість додавати на полотно геометричні фігури, зображення, піктограми та керувати візуальними ефектами для їх відображення. Figma дає можливість експортувати як окремі елементи дизайну, так і цілі вікна у різних форматах, зокрема, векторному, що є дуже зручним під час розробки. Також він дозволяє відтворювати навігаційні шляхи користувача, що є корисним у прототипуванні.

2.3.2 Опис створеного прототипу

Опираючись на постановку задачі та структуру бази даних було визначено ключові сторінки веб-застосунку та інформацію, що має на них відображатись. Інтерфейс вирішено зробити максимально простим, сучасним та інтуїтивно зрозумілим, оскільки наша цільова аудиторія – репетитори,

можуть бути будь-якого віку та з різними навичками користування електронними ресурсами.

Вибір основного кольору в нашому дизайні ґрунтується на дослідженнях, що описують те, які емоції викликають певні кольори та як їх доцільно використовувати в інтерфейсах різних типів застосунків [13]. Основна кольорова гама включає блакитний колір і його відтінки (рисунок 2.2), оскільки саме ці кольори асоціюються з довірою, спокоєм і безпекою, які є важливими у контексті нашого веб-застосунку.

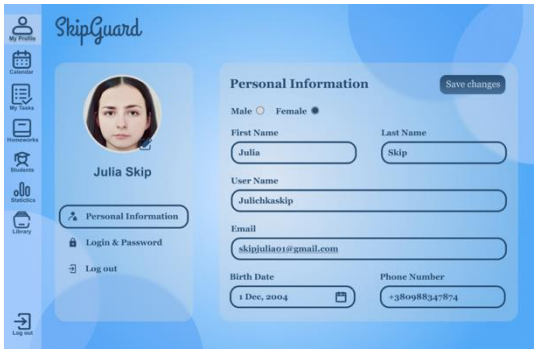


Рисунок 2.2 – Приклад дизайну профілю користувача

Додатково для покращення інтуїтивності інтерфейсу, використано елементи червоного, зеленого та жовтого кольорів (рисунок 2.3).

Рисунок 2.3 – Приклад використання додаткових кольорів у дизайні

Шрифти обрано прості та читабельні. Для логотипу (рисунок 2.4) використано декоративний шрифт Grand Hotel, який надає проєкту

індивідуальності та візуальної виразності. Назва нашого веб-застосунку містить гру слів: «Skip» – англійський варіант мого прізвища, який також перекладається як «пропустити», та «guard» – «охорона». Разом вони формують назву «SkipGuard», яку можна інтерпретувати як «охоронець від пропусків». Це відображає головну ідею проєкту: допомога репетиторам у плануванні, організації роботи та уникненні пропусків уроків та інших справ, пов’язаних з репетиторством.

Рисунок 2.4 – Логотип веб-застосунку

Для основного текстового контенту застосовано поєднання гарнітур Arial та Georgia з варіаціями кеглів, накреслень, міжрядкового інтервалу та відстаней між символами (рисунок 2.5). Таке оформлення створює чітку візуальну ієрархію та покращує сприйняття інформації.

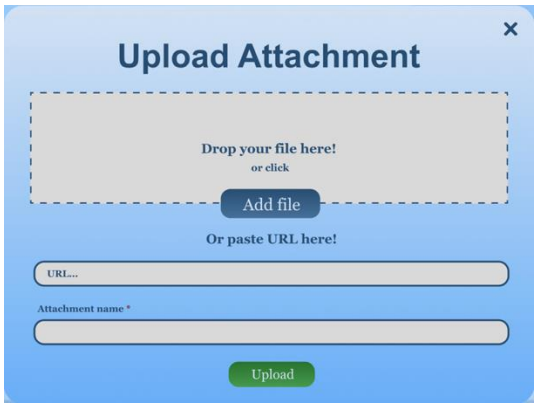


Рисунок 2.5 – Приклад поєднання гарнітур в дизайні

Даний прототип не лише демонструє елементи інтерфейсу, а й дозволяє відтворити повноцінний шлях користувача в межах нашого веб-застосунку (рисунок 2.6).

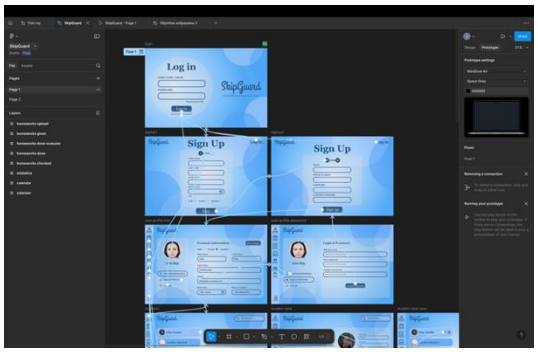


Рисунок 2.6 – Приклад реалізації навігаційного потоку між сторінками прототипу

Для реалізації цього було використано функціонал Figma, а саме – можливість додавання взаємодій. До інтерактивних елементів інтерфейсу було додано ініціатори дій та відповідні події, що дозволило реалізувати складну навігаційну логіку між сторінками прототипу.

2.4 Реалізація програмного коду веб-застосунку

Ключовим етапом створення застосунку є написання коду. На основі всіх попередніх досліджень, а також спроєктованої бази даних та створеного прототипу інтерфейсу користувача, було реалізовано повнофункціональний веб-застосунок, який складається з двох основних частин – серверної та клієнтської. Структуру, функціональні можливості та реалізацію цих частин розглянуто детальніше в наступних підрозділах.

2.4.1 Опис серверної частини веб-застосунку

Серверна частина веб-застосунку реалізована за допомогою Node.js [14]. Основним фреймворком для створення сервера є Express [15], який забезпечує гнучкий механізм маршрутизації, обробки HTTP-запитів і побудови REST API.

Під час розробки використано такі модулі npm [16] :

– express – фреймворк, що використовується для створення основи веб-сервера, він дозволяє легко створювати маршрути (GET, POST, PUT, DELETE), обробляти запити та надсилати відповіді.

– cors – модуль для Express, що дозволяє увімкнути CORS (спільне використання ресурсів з різних джерел). Це корисно, коли клієнтська та серверна частини працюють на різних портах, як у нашому випадку.

– pg-promise – модуль для взаємодії з базою даних PostgreSQL. Він надає можливість автоматичного підключення та транзакцій, і містить в собі механізм форматування запитів.

– dotenv – модуль, що використовується для завантаження змінних оточення з файлу .env. За допомогою цього механізму чутливі дані, такі як API-ключі, або паролі від бази даних можна приховати, щоб не зберігати їх безпосередньо у коді.

– jsonwebtoken – модуль для безпечної автентифікації. За його допомогою можна запобігти несанкціонованому доступу до сторінок шляхом генерування унікального сесійного токену.

– multer – проміжне програмне забезпечення, яке в основному використовується для завантаження файлів, що надходять з клієнтської частини.

– cloudinary – модуль, що використовується для з’єднання із хмарним сховищем Cloudinary. За його допомогою можна завантажувати та видаляти файли зі сховища, безпосередньо у коді програми.

– multer-storage-cloudinary – модуль для інтеграції multer та cloudinary, що дозволяє напряму завантажувати файли у хмарне сховище.

– crypto – модуль, що використовується для генерації токенів та криптографічних операцій.

Код серверної частини містить такі ключові компоненти як: взаємодія з хмарним сховищем та базою даних.

Взаємодія зі сховищем Cloudinary реалізована через три окремі маршрути (ендпоінти) для завантаження файлів:

– /upload – для завантаження зображень профілів користувачів та їхніх учнів (у форматах .jpg, .png, .jpeg) у папку «profile_pictures» у хмарному сховищі.

– /upload-file – для збереження вкладених файлів загального типу з особистих бібліотек користувачів у папці «attachments» хмарного сховища.

– /upload-homework – для завантаження файлів загального типу із домашніми завданнями, у хмарну папку «homeworks».

Кожен маршрут використовує проміжне програмне забезпечення multer з адаптером CloudinaryStorage для інтеграції з API Cloudinary. Через об'єкт params встановлюються такі параметри, таких як папка збереження та допустимі формати або тип ресурсу. До обробки відповіді додатково перевіряється наявність файлу, а результатом успішного запиту є повернення покликання на завантажений файл, яке потім записується у відповідну комірку бази даних (рисунок 2.7).

Рисунок 2.7 – Приклад взаємодії зі сховищем Cloudinary

Взаємодія з базою даних містить велику кількість маршрутів, кожен з яких відповідає за певну CRUD операцію (читання, створення, оновлення,

видалення) [17]. Для кожної таблиці бази даних створено маршрути, які забезпечують повноцінне керування інформацією у ній. Зокрема, у випадку таблиці «Tasks» реалізовано такі можливості як:

– читання записів (рисунок 2.8);

Рисунок 2.8 – Приклад читання інформації з бази даних

– додавання нового запису (рисунок 2.9);

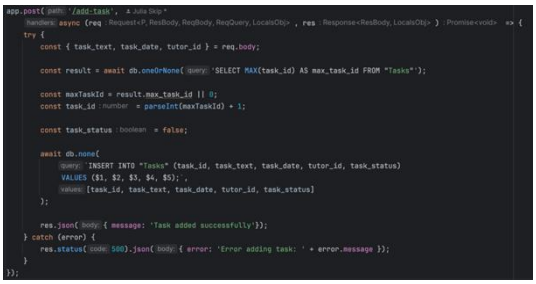


Рисунок 2.9 – Приклад створення нового запису в базі даних

– оновлення існуючого запису (рисунок 2.10).

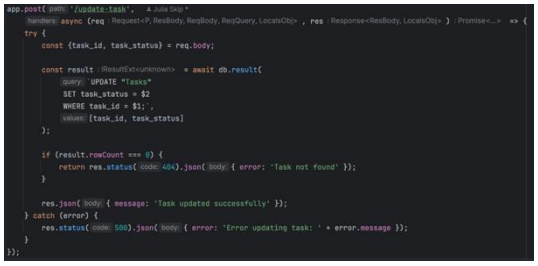


Рисунок 2.10 – Приклад оновлення інформації у базі даних

– видалення конкретного запису (рисунок 2.11);

Рисунок 2.11 – Приклад видалення інформації з бази даних

Обробка запитів здійснюється з використанням асинхронних функцій, що гарантує стабільну взаємодію з базою даних та обробку помилок.

2.4.2 Опис клієнтської частини веб-застосунку

Клієнтська частина веб-застосунку реалізована на базі JavaScript-бібліотеки для створення інтерфейсів користувача – React [18].

Проєкт побудований за компонентним підходом, де кожен функціональний блок представлений окремим React-компонентом. Компоненти згруповані у відповідні директорії згідно з їх функціональністю та для кожного з них створено власний CSS файл зі стилями (рисунок 2.12). Також використовуються глобальні стилі для таких елементів, які повторюються у різних модулях програми, наприклад – меню та вікно підтвердження дії.

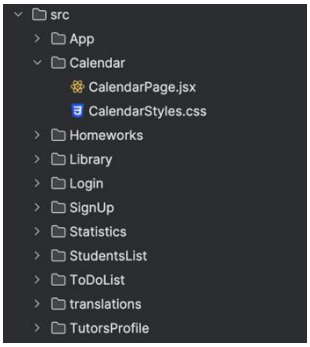


Рисунок 2.12 – Структура директорій проєкту

Для реалізації клієнтської частини веб-застосунку, окрім стандартних можливостей React, додатково використано такі інструменти:

- Для створення компонентів та управління станом використано хуки useState та useEffect із бібліотеки react.
- Для реалізації маршрутизації між сторінками використано хуки useParams та useNavigate з бібліотеки react-router-dom.
- Для зв'язку з бекендом використовується метод fetch, який дозволяє здійснювати HTTP-запити до серверу з клієнтської частини додатку. За допомогою fetch можна надсилати запити та отримувати або відправляти дані у форматі JSON.
- Для підтримки багатомовного інтерфейсу використано хук useTranslation з пакету react-i18next та об'єкт i18next з бібліотеки i18next.
- Для візуалізації статистичних даних використовується бібліотека recharts, яка дозволяє створювати інтерактивні графіки у React. Зокрема, компоненти BarChart, Bar, XAxis, YAxis, Tooltip та ResponsiveContainer використовуються для побудови адаптивних стовпчикових діаграм зі зручною навігацією та підказками при наведенні.
- Для хешування паролів використано бібліотеку bcryptjs, яка реалізує алгоритм bcrypt. Вона забезпечує безпечне зберігання паролів шляхом хешування з використанням солі, що робить дані захищеними від атак методом перебору.
- Для відправки електронних листів користувачам використовується бібліотека emailjs, яка дозволяє інтегрувати функціонал відправки повідомлень з клієнтської частини без необхідності налаштування серверної логіки.

2.4.3 Опис реалізованого функціоналу

В попередніх підрозділах ми розглянули те, яким чином був реалізований наш веб-застосунок для організації та планування роботи

репетитора. У цьому підрозділі ми розглянемо конкретний функціонал, який пропонує користувачам наш ресурс.

Сторінки входу та реєстрації

При відкритті нашого веб-застосунку, користувач бачить перед собою сторінку входу у свій особистий кабінет (рисунок 2.13), йому пропонується здійснити вхід, ввівши свої електронну адресу або ім’я користувача та пароль, вказані при реєстрації.

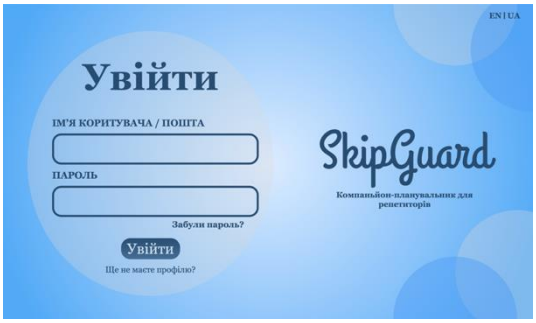


Рисунок 2.13 – Сторінка входу у систему

У випадку, якщо користувач забув свій пароль, він має можливість надіслати запит на його відновлення. Після натискання на текст «Забули пароль?», його перенаправить на форму, де потрібно ввести свою електронну адресу, на яку прийде покликання на форму для скидання паролю (рисунок 2.14).

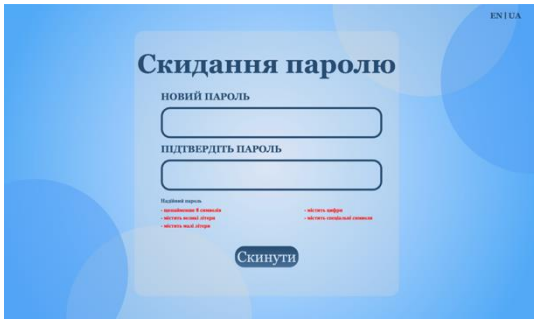
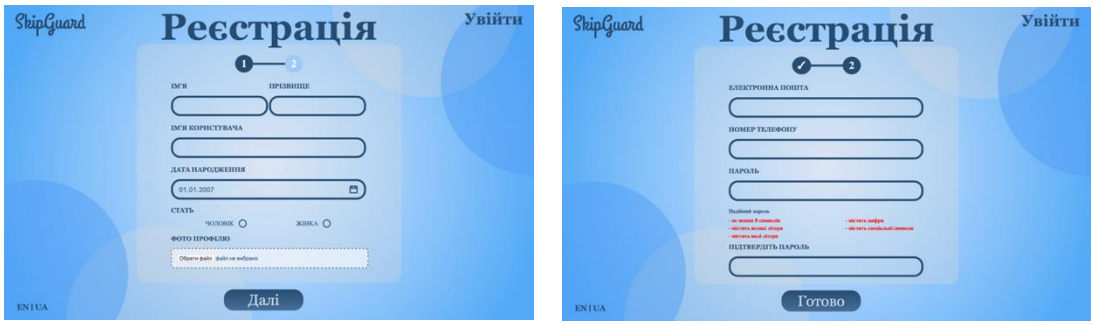


Рисунок 2.14 – Форма для скидання паролю

Покликання є унікальним та дійсним лише протягом 15 хвилин, або до моменту зміни паролю в межах цього часу.

Якщо у користувача ще немає створеного профілю, то він може зареєструватись, натиснувши на текст «Ще не маєте профілю?» та заповнивши усі необхідні поля форми реєстрації (рисунок 2.15).



а *б*

Рисунок 2.15 – Перший крок реєстрації (а) та другий крок реєстрації(б)

Профіль користувача

Після реєстрації, у кожного користувача з'являється особистий профіль (рисунок 2.16), де зберігається контактна інформація про нього та інформація для входу. Користувачу надається можливість переглядати та редагувати особисті дані та змінювати пароль.

а *б*

Рисунок 2.16 – Профіль користувача: особиста інформація (а) та інформація для входу(б)

Сторінка з інформацією про учнів

На сторінці з інформацією про учнів відображається список усіх учнів репетитора. Він може додавати нових, видаляти та редагувати інформацію про вже існуючих учнів, здійснювати пошук за прізвищем або іменем. Для кожного з учнів вказана контактна інформація та інформація, що стосується навчального процесу (рисунок 2.17).

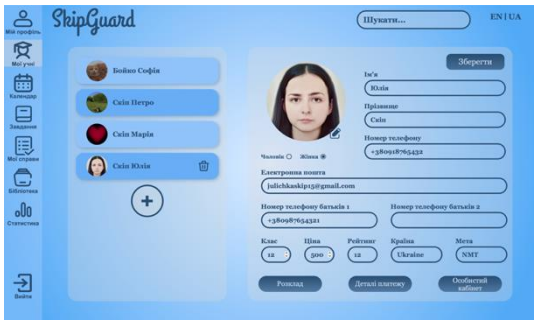
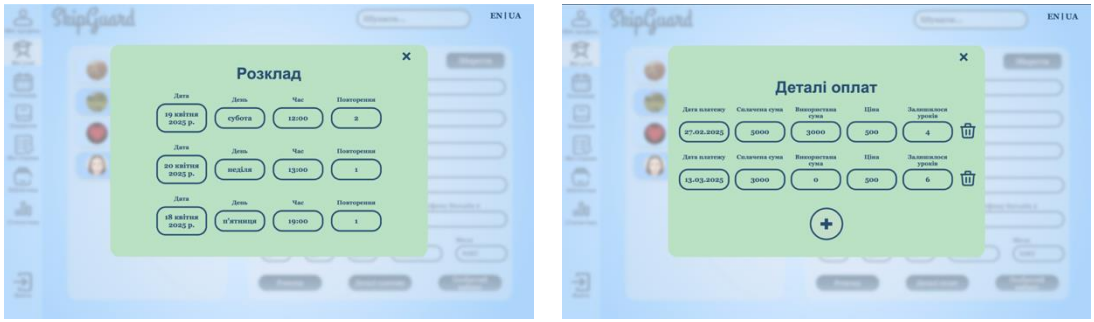


Рисунок 2.17 – Сторінка з інформацією про учнів

Додатково можна отримати детальну інформацію щодо розкладу запланованих занять (рисунок 2.18 (а)) та оплат (рисунок 2.18 (б)), покликання на особистий кабінет для здачі домашніх робіт конкретного учня. Розклад підтягується автоматично з календаря. Деталі оплати репетитор вносить вручну, проте списування коштів за уроки відбувається автоматично.



а *б*

Рисунок 2.18 – Відображення розкладу (а) та деталей оплат учня (б)

Сторінка для планування занять

На сторінці для планування занять користувач може переглянути свій розклад на поточний день (рисунок 2.19 (а)), тиждень (рисунок 2.19 (б)).

а *б*

Рисунок 2.19 – Перегляд планів на день (а) та тиждень (б)

Також доступний перегляд розкладу на місяць (рисунок 2.20 (а)) та деталі конкретної події (рисунок 2.20 (б)).

а *б*

Рисунок 2.20 – Перегляд планів та місяць (а) та конкретної події (б)

Додатково на сторінці відображається відлік до найближчої запланованої події, календар та годинник.

Репетитор може додавати нові плани (рисунок 2.21 (а)), видаляти чи редагувати (рисунок 2.21 (б)) існуючі.

а *б*

Рисунок 2.21 – Форми для додавання (а) та редагування (б) подій

Сторінка з домашніми завданнями

На сторінці з домашніми завданнями відображається повний список учнів користувача та деталі домашніх завдань кожного учня окремо. Репетитор може завантажувати нові роботи (рисунок 2.22 (а)), переглядати, редагувати та видаляти задані (рисунок 2.22 (б)).

а *б*

Рисунок 2.22 – Форма завантаження нової роботи (а) та перегляд заданих домашніх робіт (б)

Також надається можливість перевіряти та оцінювати здані завдання (рисунок 2.23).

а *б*

Рисунок 2.23 – Перегляд зданих (а) та оцінених (б) робіт

Учні можуть переглядати в особистому кабінеті задані та оцінені завдання й завантажувати виконані.

Сторінка з особистими справами

На сторінці з особистими справами (рисунок 2.24) користувач може планувати повсякденні справи.

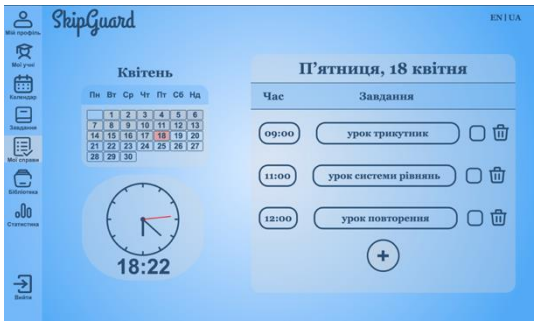


Рисунок 2.24 – Сторінка для перегляду особистих планів на день

Тут репетитор може організовувати свої дні в межах поточного місяця, додаючи, видаляючи та позначаючи як виконані, повсякденні завдання. Також у цей список автоматично підтягуються заплановані заняття, щоб уникнути накладок у графіку.

Особиста бібліотека користувача

На сторінці бібліотеки (рисунок 2.25) користувач може створювати «полиці» для вкладень та завантажувати туди файли або покликання.

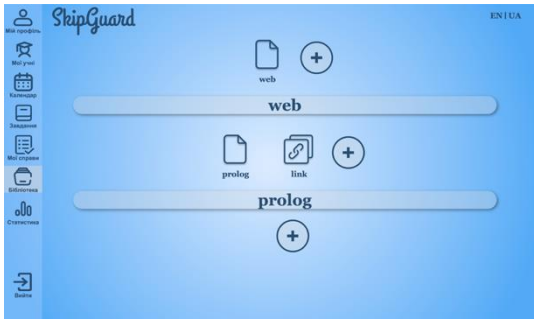


Рисунок 2.25 – Особиста бібліотека користувача

Надається можливість видаляти вкладення, переглядати покликання (рисунок 2.26 (а)) в браузері та завантажувати файли (рисунок 2.26 (б)) на свій пристрій.

а *б*

Рисунок 2.26 – Опції керування покликанням (а) та файлом (б)

Сторінка статистики

На сторінці статистики користувач може переглядати графіки з даними про доходи (рисунок 2.27 (а)) та проведені заняття (рисунок 2.27 (б)) за поточні тиждень, місяць та рік.

а *б*

Рисунок 2.27 – Перегляд статистики доходів (а) та уроків (б)

Наш веб-застосунок в першу чергу спрямований на український ринок, проте ми не виключаємо його використання закордоном, тому прийнято рішення реалізувати його, використовуючи дві локалізації – українську та англійську.

2.4.4 Апробація реалізованого веб-застосунку

Успішно проведено апробацію створеного веб-застосунку серед цільової аудиторії. У її процесі взяли участь двоє репетиторів з математики, які у своїх відгуках (рисунок 2.28) зазначили, що завдяки користуванню нашим ресурсом вони значно полегшили процес підготовки до занять та збільшили свою продуктивність. Це позитивно позначилось на процесі навчання та загальному самопочутті респондентів.

а

б

Рисунок 2.28 – Відгук першої репетиторки (а) та другої репетиторки (б)

2.5 Висновки до Розділу 2

Отже, було сформовано постановку задачі, визначивши ключові можливості, як основні вимоги, що мають бути реалізованими у веб-застосунку для організації та планування роботи репетитора, який буде задовільняти запити користувачів. Розглянуто та детально описано процес проєктування бази даних, зокрема, побудову моделі «сутність-зв’язок» і реляційної моделі, та обгрунтувано вибір СКБД для створення бази даних. Представлено створений інтерактивний прототип інтерфейсу користувача веб-застосунку та обгрунтовано рішення щодо дизайну. Описано процес розробки среверної та клієнтської частин веб-застосунку, з урахуванням необхідних вимог до створеного веб-застосунку. Підсумовано та проаналізовано реалізований функціонал. Проведено апробацію створеного веб-застосунку серед цільової аудиторії.

Висновки

У результаті виконаної роботи було реалізовано програмний продукт для організації та планування роботи репетиторів, що відповідає актуальним потребам користувачів.

Проведене опитування потенційних користувачів та аналіз ринку існуючих рішень підтвердили актуальність теми та наявність запиту на комплексний веб-застосунок, який би поєднував функціонал для планування, обліку учнів та доходів, управління домашніми завданнями, структурованого збереження матеріалів та аналітики у єдиному ресурсі.

Під час розробки було сформульовано чітку постановку задачі, визначено ключові вимоги до продукту, спроєктовано і створено базу даних, розроблено інтерактивний прототип користувацького інтерфейсу та реалізовано серверну і клієнтську частини веб-застосунку із застосуванням вибраних технологій та інструментів. Реалізований веб-застосунок в першу чергу спрямований на український ринок, але ми не виключаємо можливість його використання закордоном, тому використано дві локалізації – українську та англійську.

Проведено успішну апробацію створеного веб-застосунку серед цільової аудиторії та наведено її результати. Наш веб-застосунок дозволяє репетиторам ефективніше організовувати свою роботу, витрачаючи менше часу на адміністративні завдання, що сприяє покращенню якості навчального процесу та забезпечує краще балансування між професійною діяльністю та особистим життям.

Результати виконаної роботи відкривають перспективи подальшого розвитку веб-застосунку – удосконалення існуючого та додавання нового

функціоналу відповідно до змін потреб користувачів та інтеграцію з іншими сервісами для ще більшої зручності.

Таким чином, поставлену мету було досягнуто, а результати розробки мають високий потенціал практичного застосування серед репетиторів різних напрямків.

Список літератури

1. Google Forms [Електронний ресурс]. – Режим доступу до ресурсу: https://workspace.google.com/products/forms/
2. UXPUB. Продуктовий дизайн. Інтерв'ю з користувачами (на основі прото-персони) та гіпотез [Електронний ресурс] / UXPUB. – Режим доступу до ресурсу: https://ux.pub/zhmikhov/intierviu-z-koristuvachami-na-osnovi-proto-piersoni-ta-ghipotiez-5b5n
3. EasyWeek. Програма для репетитора [Електронний ресурс]. – Режим доступу до ресурсу: https://easyweek.com.ua/solutions/tutor
4. Google Calendar [Електронний ресурс]. – Режим доступу до ресурсу: https://calendar.google.com
5. Notion [Електронний ресурс]. – Режим доступу до ресурсу: https://www.notion.com
6. План тижня – Щоденник [Електронний ресурс] / Google Play. – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.weeklyplannerapp.weekplan&hl=uk
7. Тижневий планувальник [Електронний ресурс] / Google Play. – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.jonasbernardo.developer.planejamento_semanal&hl=uk
8. Chen P. P. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned [Електронний ресурс] / Peter P. Chen // Computer Science Department, Louisiana State University. – Режим доступу до ресурсу: http://bit.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf

9. draw.io [Електронний ресурс]. – Режим доступу до ресурсу: https://app.diagrams.net
10. Converting ERD to a relational model [Електронний ресурс] // Runestone Academy. – Режим доступу до ресурсу: https://runestone.academy/ns/books/published/practical_db/PART2_DATA_MODELING/03-ERD-to-relational/ERD-to-relational.html
11. PostgreSQL 17 Documentation [Електронний ресурс] / PostgreSQL Global Development Group. – Режим доступу до ресурсу: https://www.postgresql.org/files/documentation/pdf/17/postgresql-17-A4.pdf
12. Figma [Електронний ресурс]. – Режим доступу до ресурсу: https://www.figma.com/design/
13. Cao J. Colour psychology for web design: 14 examples [Електронний ресурс] / J. Cao ; за участю J. Foley // Creative Bloq. – Features. – Режим доступу до ресурсу: https://www.creativebloq.com/web-design/12-colours-and-emotions-they-evoke-61515112
14. Node.js v23.11.0 documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://nodejs.org/docs/latest/api/
15. Express – офіційна документація [Електронний ресурс]. – Режим доступу до ресурсу: https://expressjs.com/uk/
16. npm [Електронний ресурс]. – Режим доступу до ресурсу: https://www.npmjs.com
17. DAO CRUD operations – Exposed documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://www.jetbrains.com/help/exposed/dao-crud-operations.html
18. React – офіційна документація [Електронний ресурс]. – Режим доступу до ресурсу: https://react.dev