

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: «Програмний комплекс забезпечення процесу тестування»

Здобувача (ки) групи ІТ-91 Акименка Віталія Віталійовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Віталій АКІМЕНКО
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доц., к.т.н., доц. Світлана ВАЩЕНКО _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО
«_____» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Акименку Віталію Віталійовичу

1 Тема роботи Програмний комплекс забезпечення процесу тестування

керівник роботи Ващенко Світлана Михайлівна, к.т.н., доцент,

затверджені наказом по університету від « 29 » травня 2023 р. №0588-VI

2 Строк подання студентом роботи « 7 » червня 2023 р.

3 Вхідні дані до роботи технічне завдання на розробку програмного комплексу

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, аналіз предметної області, проектування програмного комплексу, розробка програмного комплексу; висновки

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

Актуальність, постановка задачі, аналіз комплексів-аналогів, порівняння аналогів, структура проекту, вимоги до продукту, структурно-функціональна модель, варіанти використання веб-додатку, варіанти використання мобільного додатку, фізична модель бази даних, засоби реалізації, демонстрація сторінок веб-застосунку, демонстрація сторінок мобільного застосунку, апробація, висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 02.01.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	02.01.2023 – 25.01.2023	Виконано
2	Планування робіт	26.01.2023 – 15.02.2023	Виконано
3	Розробка веб-додатку програмного комплексу	16.02.2023 – 01.04.2023	Виконано
4	Тестування роботи веб-додатку	02.04.2023 – 13.04.2023	Виконано
5	Розробка мобільного додатку програмного комплексу	14.04.2023 – 17.05.2023	Виконано
6	Тестування роботи мобільного додатку	18.05.2023 – 31.05.2023	Виконано
7	Створення документації проекту	01.06.2023 – 07.06.2023	Виконано

Студент

(підпис)

Віталій АКІМЕНКО

Керівник роботи

(підпис)

к.т.н., доц. Світлана ВАЩЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Програмний комплекс забезпечення процесу тестування».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 14 найменувань, додатків. Загальний обсяг роботи – 117 сторінок, у тому числі 53 сторінки основного тексту, 2 сторінки списку використаних джерел, 62 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці програмного комплексу забезпечення процесу тестування учнів в школах України.

В роботі проведено дослідження області програмних комплексів для тестування, виконано аналіз та порівняння програмних комплексів-аналогів, сформовано технічне завдання на розробку програмного комплексу, описано архітектуру програмного комплексу та його компонентів, формування список робіт та етапів розробки, а також демонстрацію роботи програмного комплексу як його компонентів.

Результатом проведеної роботи є програмний комплекс забезпечення процесу тестування учнів в школах України, що складається з мобільного та веб-додатків.

Практичне значення роботи полягає у автоматизації процесу створення, формування тестів, їх проходження та збору результатів тестувань, підвищенні якості та простоти процесу тестування в школах України, а також зменшення впливу людських факторів на результати тестування.

Ключові слова: програмний комплекс, мобільний додаток, веб-додаток, Xamarin.Forms, тестування, школа.

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Актуальність проблеми	7
1.2 Аналіз програмних комплексів-аналогів	8
1.3 Постановка задачі.....	13
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ	15
2.1 Моделювання роботи веб-додатку програмного комплексу в нотації IDEF0	15
2.2 Моделювання роботи мобільного додатку програмного комплексу в нотації IDEF0	18
2.3 Діаграма варіантів використання веб-додатку програмного комплексу	20
2.4 Діаграма варіантів використання мобільного додатку програмного комплексу	21
2.5 Проектування моделі БД програмного комплексу	23
3 РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ	25
3.1 Архітектура програмного комплексу.....	25
3.2 Архітектура веб-додатку	26
3.3 Архітектура мобільного додатку.....	26
3.4 Програмна реалізація веб-додатку	28
3.5 Програмна реалізація мобільного додатку	34
3.6 Використання веб-додатку.....	38
3.7 Використання мобільного додатку	45
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А	57
ДОДАТОК Б	80
ДОДАТОК В	87

ВСТУП

По всьому світу навчальні заклади автоматизують процеси навчання та тестування за допомогою електронних пристроїв та платформ, і така тенденція є результатом впливу багатьох факторів, не тільки подіями останніх років, а й у цілому простотою використання сучасних технологій, які економлять ресурси як учнів, так і вчителів.

Традиційні методи тестування значною мірою покладаються на ручні, паперові процеси, часто з багатьма обмеженнями. Але нікому не секрет, що вони мають безліч недоліків, таких як трудомісткий процес створення тестів та перевірки результатів, можливість помилок під час оцінювання, а також відсутність гнучкості в адаптації оцінювання до різних навчальних потреб.

З появою програмних комплексів забезпечення процесу тестування, під беззаперечним тиском умов останніх років, такі популярні методи тестувань учнів почали відходити на другий план. Такі комплекси є зручною та доступною повноцінною заміною звичним кожному тестів на листочках. Учителям більше не потрібно кожного разу проводити та перевіряти однотипні тести, які іноді мають тенденцію не змінюватись роками.

Тому тема даної роботи визначена як розробка програмного комплексу, який дозволить спростити процес організації тестування зі сторони вчителя, а також надасть учням зручний сучасний засіб для виконання тестових завдань.

Досягнути поставленої мети можна за рахунок вирішення таких задач:

- детальне дослідження предметної області програмного комплексу, а також аналіз програмних аналогів;
- визначення функціональних вимог до програмного комплексу та його компонентів, а також укладення календарного плану проекту;
- проведення структурно-функціонального моделювання програмного комплексу та його компонентів;

- дослідження та вибір інструментів та середовищ для програмної реалізації компонентів програмного комплексу;
- проектування та створення спільної для комплексу бази даних;
- розробка сучасного, простого та інтуїтивно зрозумілого інтерфейсу компонентів програмного комплексу;
- тестування роботи веб- та мобільного застосунків як готового програмного комплексу.

За результатами представленої роботи було опубліковано тези доповіді на конференції «ІМА – 2023» [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

У сучасному освітньому середовищі, який з кожним днем розвивається все швидше, беззаперечною основою для трансформації традиційних методів навчання стали інтеграції комп'ютерних технологій. Поява цифрових інструментів і платформ заклала фундамент до створення методів оцінювання, які покращують та полегшують процес тестування для вчителів і учнів.

Чи не найпопулярнішим варіантом цифрових платформ є саме повноцінні програмні комплекси, що задовольняють потреби всіх користувачів. Наприклад, веб-частина платформи є зручним інструментом для викладачів, що мають постійно переглядати, редагувати та доповнювати матеріали, а мобільна версія – вдалим знаряддям для учнів, що можуть в будь-який момент взяти телефон і засвоїти матеріал.

Як повідомляє GitNux, 95% учнів приходять на заняття з мобільними пристроями, а 94% - мають на меті використати його в академічних цілях [2]. В доповнення до цієї статистики DataReportal повідомляє, що середньостатистична людина використовує мобільний пристрій близько 5 годин на добу [3]. Все це є беззаперечними доказами того, що можливість та бажання використовувати мобільні пристрої в навчальних цілях буде не тільки зручно для сучасних учнів, а й вже є потребою, навіть не зважаючи на події останніх років.

Формат дистанційного навчання з використанням програмних комплексів був підтриманий МОН України, який ініціював створення платформи «Всеукраїнська школа онлайн», що запрацював в грудні 2020 року [4]. Вже в травні 2020 році «Всеукраїнська школа онлайн» запустила мобільний додаток, як логічне продовження платформи у зручному та завжди

доступному мобільному форматі. Таке рішення було прийняте, оскільки, як повідомляє МОН України, «за результатами дослідження Державної служби якості освіти, 89,3% учнів для дистанційного навчання використовують саме смартфони» [5].

Функціонал вже створених програмних комплексів постійно оновлюється, що однозначно допомагає підтримувати попит до них. Кожен користувач має можливість написати відгук на платформі, звідки додаток був завантажений. Ці коментарі оброблюються командою розробників, які потім доповнюють та покращують розроблюваний продукт.

1.2 Аналіз програмних комплексів-аналогів

На даний момент можна знайти велику кількість різноманітних платформ для створення тестів. Більшість з них має застарілий дизайн та програють у функціональності сучаснішим рішенням, які постійно оновлюються та розвиваються [6].

У рамках роботи над проектом було оглянуто існуючі програмні комплекси для забезпечення процесу тестування.

Перший з них це «Kahoot!» - ігрова навчальна платформа, що дозволяє одним користувачам створювати тести на сайті, а іншим – проходити їх за допомогою смартфона [7]. Даний аналог має інтуїтивно зрозумілий дизайн та інтерфейс як в веб-, так і в мобільному додатку.

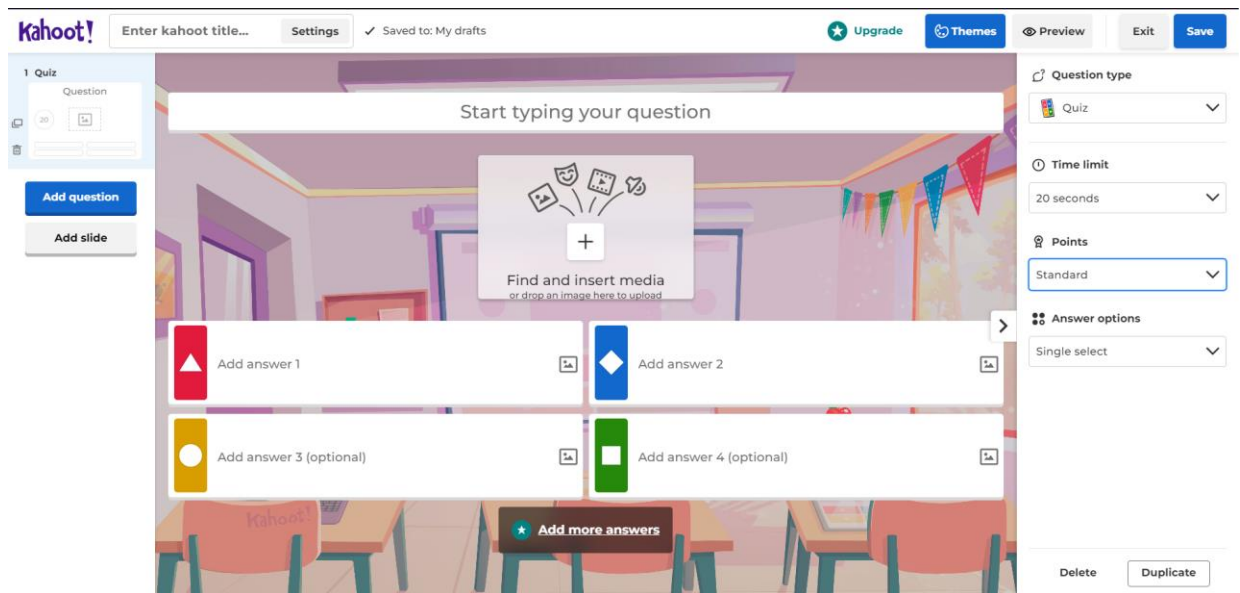


Рисунок 1.1 - Сторінка створення тесту в веб-додатку «Kahoot!»

Функціонал платформи дозволяє створювати на сайті або в мобільному застосунку тести з будь-якою кількістю запитань типу «Quiz» або «True or False» зі змінними часовими обмеженнями, типами оцінювання, та кількістю відповідей на кожне запитання (рис.1.1). Створені тести доступні для проходження за унікальним номером в додатку для смартфона (рис.1.2).

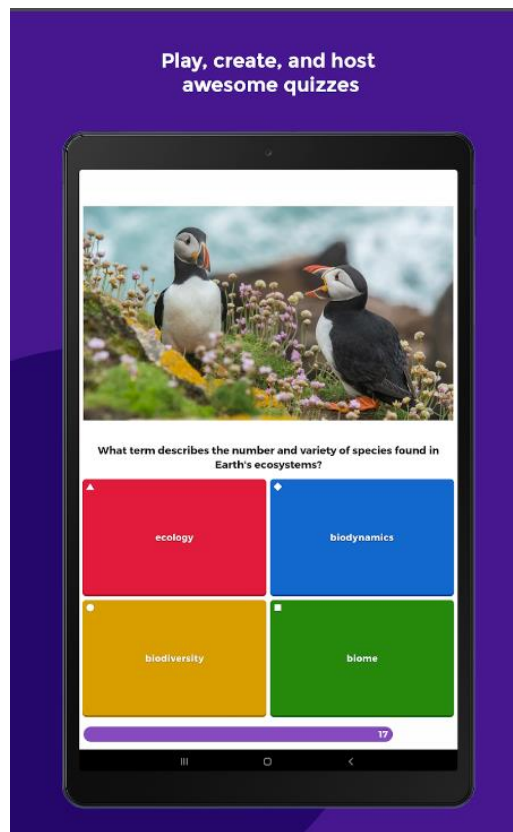


Рисунок 1.2 – Сторінка проходження тесту в мобільному додатку «Kahoot!»

До недоліків платформи відносяться максимальна обмеженість безкоштовного плану, за яким кількість одночасних проходжень тесту обмежена до 40 і відсутня більшість функціоналу платформи (рис. 1.3)., а також відсутність формування статистики проходження тесту учнями.

	Basic Free	Kahoot!+ Start for teachers €3.99 <small>per teacher per month €48 billed annually</small>	Kahoot!+ Premier for teachers €5.99 <small>per teacher per month €72 billed annually</small>	Kahoot!+ Max for teachers €9.99 <small>per teacher per month €108 billed annually</small>
Player limit			Most popular	Best value
Synchronous player limit	40	100	200	400
Creation set				
Create private kahoots	?	×	✓	✓
Premium image library	?	×	✓	✓
Teaching guides	?	×	✓	✓
Audio in questions	?	×	✓	✓
Review set				

Create engaging year-end review sessions with Kahoot!+ Premier for only €5.99/month. [Buy now](#)

Рисунок 1.3 - Порівняння планів платформи «Kahoot!»

Наступний комплекс-аналог «Quizizz» – платформа проходження тестів, що позиціонує себе як «інструмент для самостійного навчання, який допомагає кожному учневі відстежувати свої досягнення» [8]. Веб- та мобільні застосунки притримуються лаконічності та сучасності інтерфейсу, що буде зрозумілим будь-якому користувачу. Тести можуть бути створені на сайті (рис. 1.4), після чого надається можливість пройти їх в мобільному застосунку (рис. 1.5).

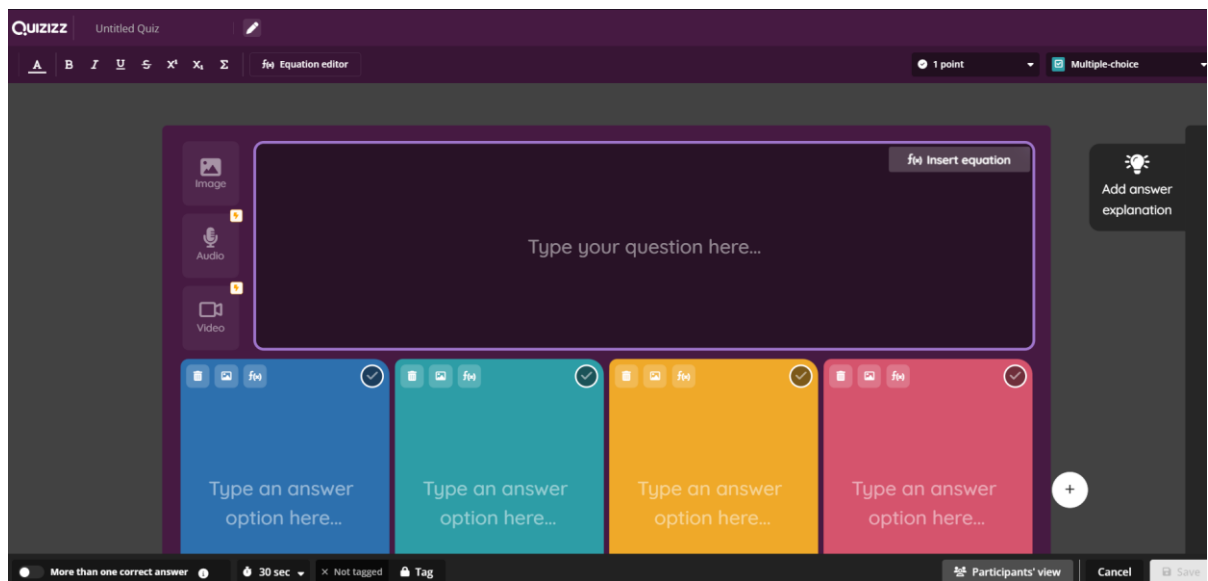


Рисунок 1.4 - Сторінка створення тесту в веб-додатку «Quizizz»



Рисунок 1.5 – Сторінка проходження тесту в мобільному додатку «Quizizz»

За результатами аналізу програмних комплексів було складено порівняльну таблицю аналогів та розроблюваного програмного комплексу, що представлена в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика комплексів-аналогів

Критерій порівняння	«Kahoot!»	«Quizizz»	Розроблюваний комплекс
Наявність кросплатформенного мобільного додатку	-	+	+
Безкоштовний доступ до всього функціоналу	-	-	+
Варіативна кількість відповідей на запитання	+	+	+
Можливість додати фото до запитання	+	+	+
Позакласні проходження тестів	+	+	+
Інтерфейс українською мовою	-	-	+
Збір статистики з результатами тестувань для кожного учня	-	+	+

1.3 Постановка задачі

Проаналізувавши предметну область та оглянувши найпопулярніші аналоги інформаційних систем було визначено мету проекту – розробка програмного комплексу, що складається з веб- та кросплатформенного мобільного додатків, для організації процесу тестування учнів в школах України.

Кожна з частин комплексу має бути орієнтована на відповідного користувача: для веб-частини це вчитель, для мобільної – учень.

Функціонал веб-додатку комплексу має містити наступні функції для користувачів-вчителів:

- створення тестів двох типів: «Вікторина» та «Хто швидше»;
- перегляд та редагування створених тести
- створення запитань до тестів з 4-6 варіантами відповіді
- перегляд та редагування створених запитань
- отримання доступу до звітності про проходження тестів учнями
- надання доступу учням до роботи над помилками за завданнями, на які він дав неправильну відповідь.
- редагування відсоткове зниження результуючого балу за повторну відповідь учнем на неправильне запитання.

Що стосується користувачів-учнів, мобільний додаток комплексу повинен мати наступні функції для них:

- створення тестів двох типів: «Вікторина» та «Хто швидше»;
- перегляд та редагування створених тести
- створення запитань до тестів з 4-6 варіантами відповіді
- перегляд та редагування створених запитань
- отримання доступу до звітності про проходження тестів учнями
- надання доступу учням до роботи над помилками за завданнями, на які він дав неправильну відповідь.

— редагування відсоткове зниження результуючого балу за повторну відповідь учнем на неправильне запитання.

Розробка веб-додатку програмного комплексу буде виконуватися в середовищі WebStorm мовами програмування HTML, CSS, JavaScript, PHP.

Мобільний додаток буде розроблятися в середовищі VisualStudio з використанням фреймворку Xamarin.Forms, що дозволяє створювати одночасно застосунок для мобільних платформ Android та iOS.

Повні переліки вимог до розроблюваного комплексу та його частин наведено в технічному завданні в додатку А.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ

Етап проектування програмного комплексу є логічним продовженням процесу розробки програмного комплексу після проведення робіт з визначення актуальності проблеми та мети проекту, а також постановки задачі для проекту.

2.1 Моделювання роботи веб-додатку програмного комплексу в нотації IDEF0

Основний процес веб-додатку представляє собою організацію процесу тестування вчителем, що зображено на рисунку 2.1 у вигляді діаграми A-0 нульового рівня декомпозиції в нотації IDEF0.

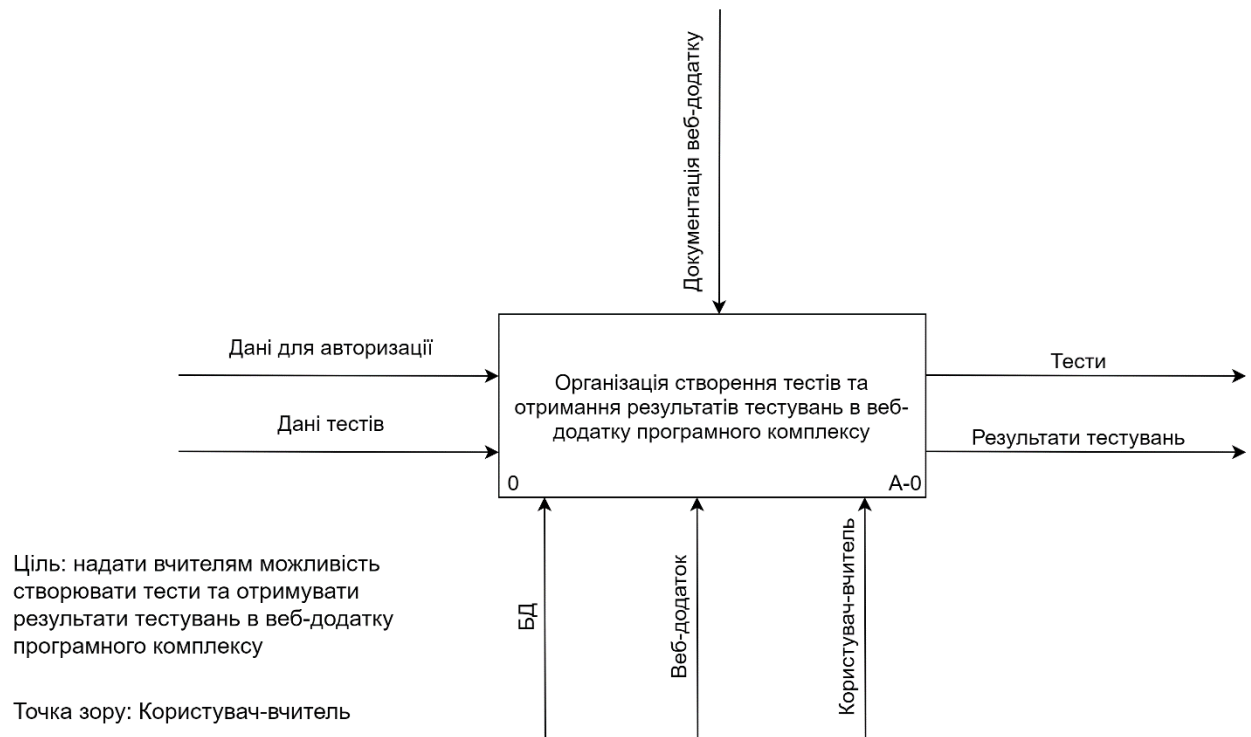


Рисунок 2.1 – Контекстна діаграма основного процесу веб-додатку

Наступним етапом було виконано декомпозицію розробленої контекстної діаграми на підпроцеси, а саме:

- авторизація користувача-вчителя;
- підготовка тестів;
- збір результатів тестувань.

Діаграма декомпозиції першого рівня контекстної діаграми веб-додатку зображена на рисунку 2.2.

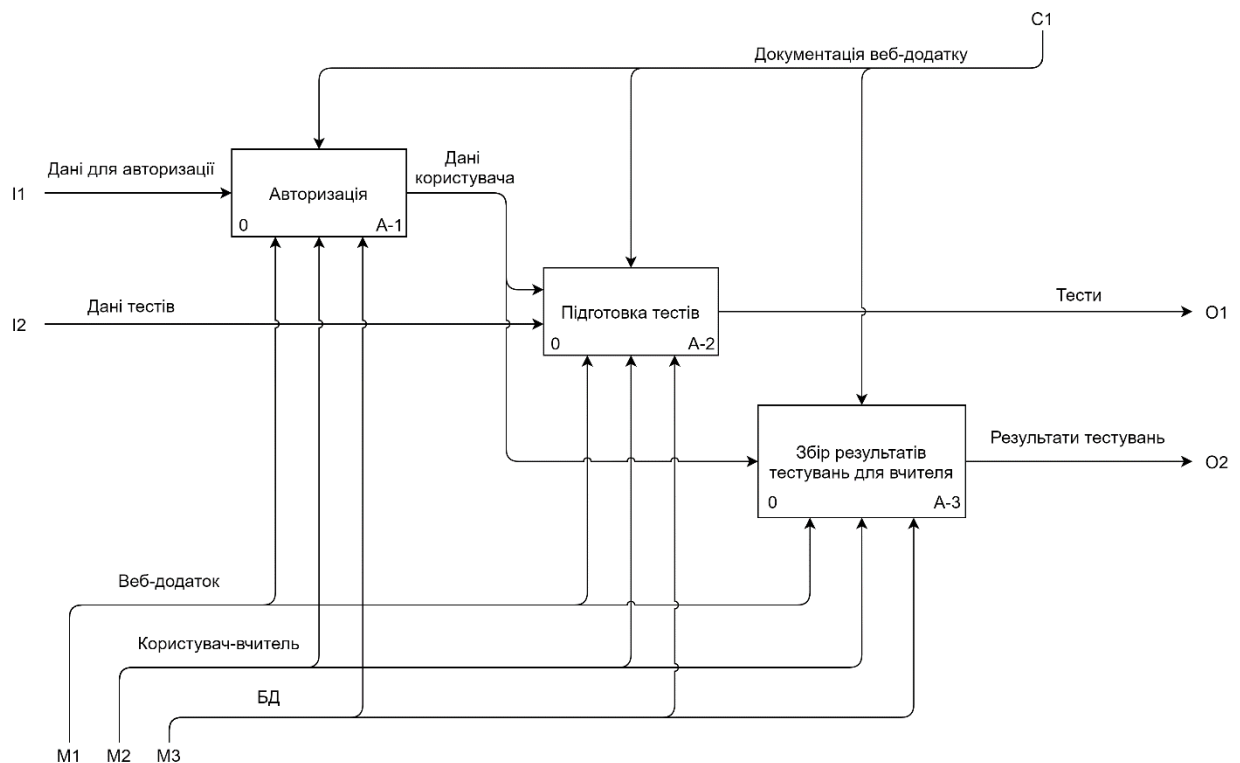


Рисунок 2.2 – Декомпозиція першого рівня основного процесу веб-додатку

Для деталізації процесу «Підготовка тестів» було виконано діаграму декомпозиції другого рівня. Даний процес був розділений на 6 логічних процесів, а саме:

- створення тесту;
- пошук вже створеного тесту;
- редагування тесту
- створення запитань;

- вибір вже створених запитань;
- редагування запитань.

Діаграма декомпозиції зображена на рисунку 2.3.

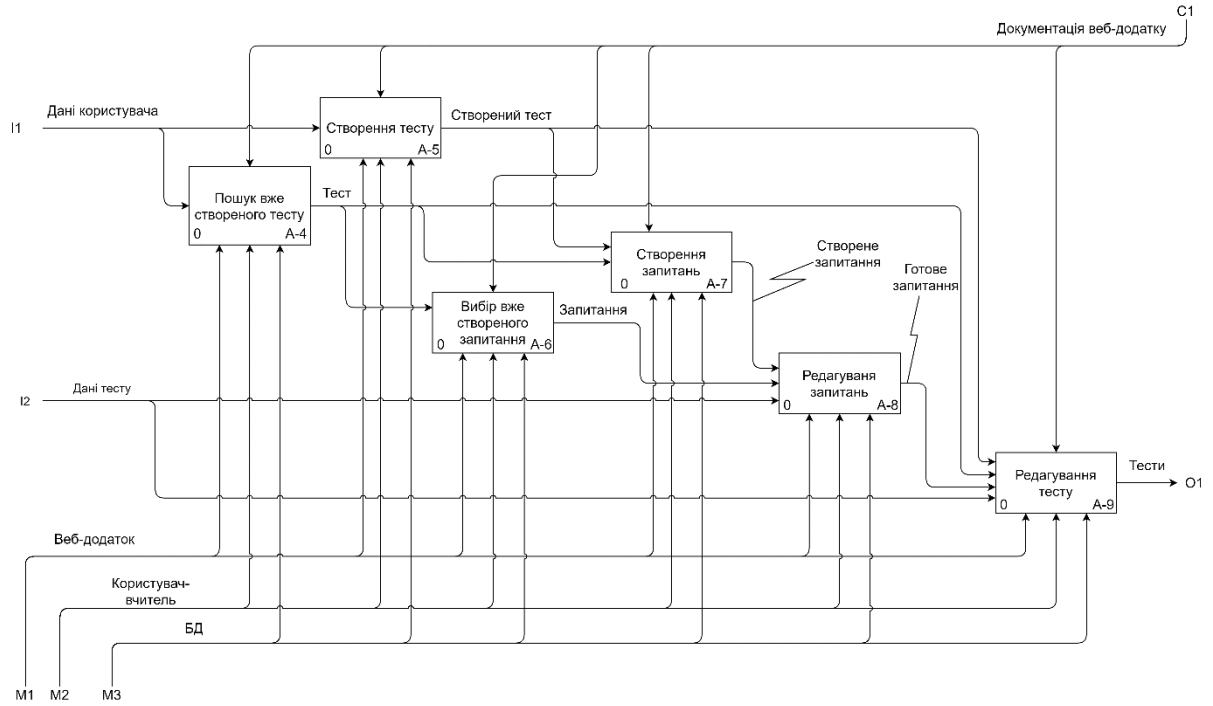


Рисунок 2.3 – Декомпозиція процесу «Підготовка тестів» основного процесу веб-додатку програмного комплексу

2.2 Моделювання роботи мобільного додатку програмного комплексу в нотації IDEF0

Основний процес мобільного додатку представляє собою організацію процесів проходження тестів та збору статистики учнем, що зображено на рисунку 2.4 у вигляді діаграми Б-0 нульового рівня декомпозиції в нотації IDEF0.

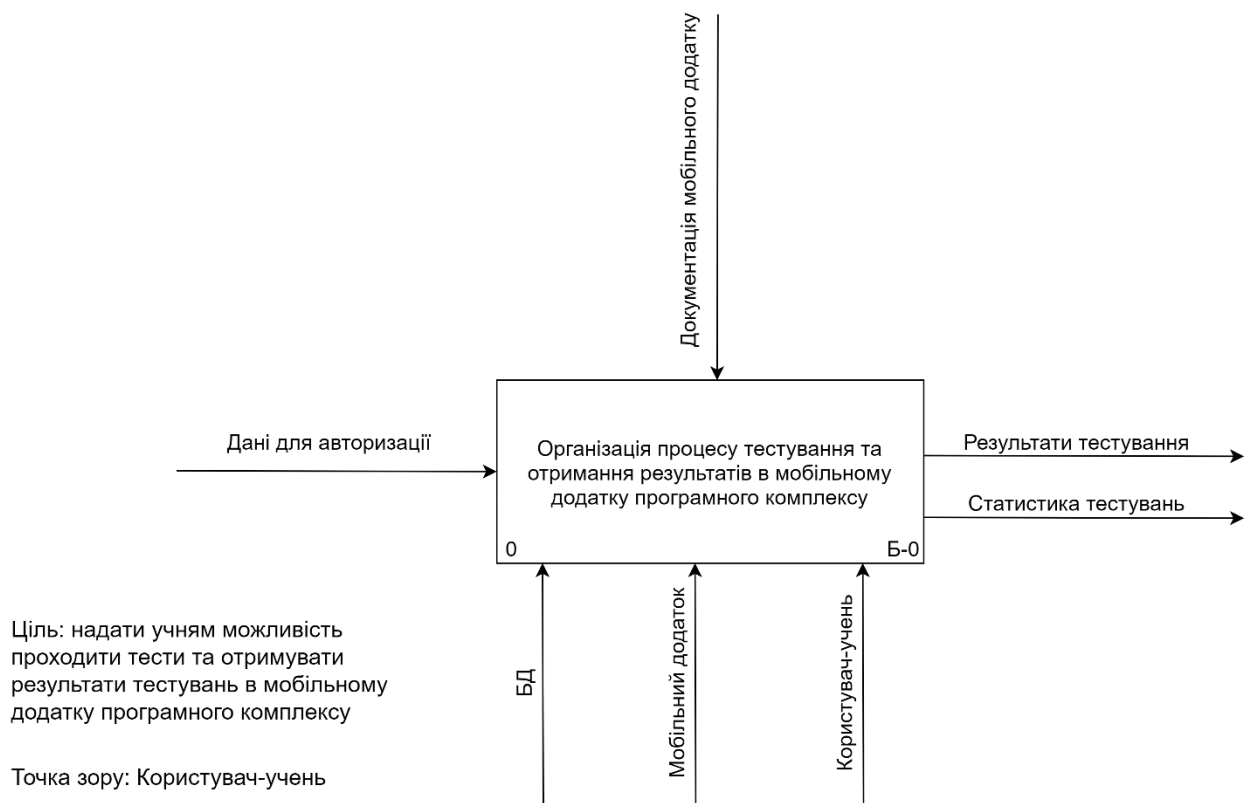


Рисунок 2.4 – Контекстна діаграма основного процесу мобільного додатку

Наступним етапом було виконано декомпозицію розробленої контекстної діаграми на підпроцеси, а саме:

- авторизація користувача-учня;
- проходження тестів;
- збір результатів тестувань для учня.

Діаграма декомпозиції першого рівня контекстної діаграми мобільного додатку зображена на рисунку 2.5.

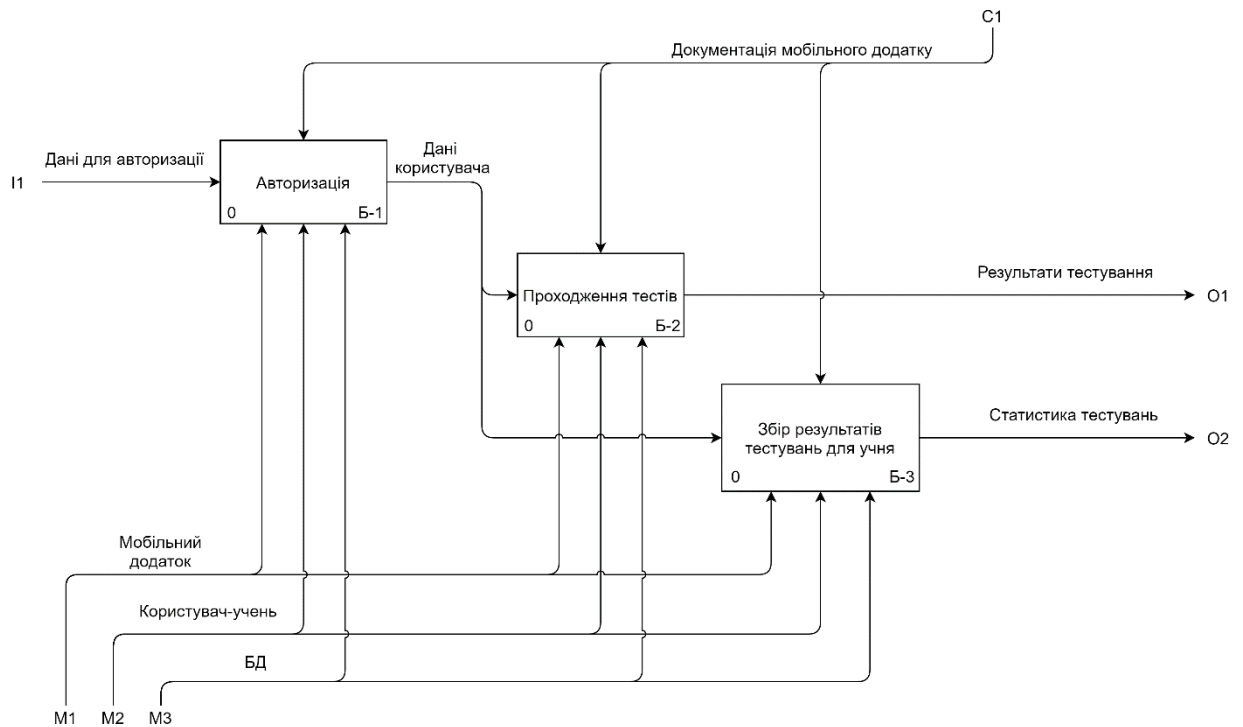


Рисунок 2.5 – Декомпозиція першого рівня основного процесу мобільного додатку

Для деталізації процесу «Проходження тесту» було виконано діаграму декомпозиції другого рівня. Даний процес був розділений на 4 логічних процесів, а саме:

- пошук тесту для проходження;
- надання відповідей на запитання тесту;
- перегляд результатів тестування;
- робота над помилками.

Діаграма декомпозиції зображена на рисунку 2.6.

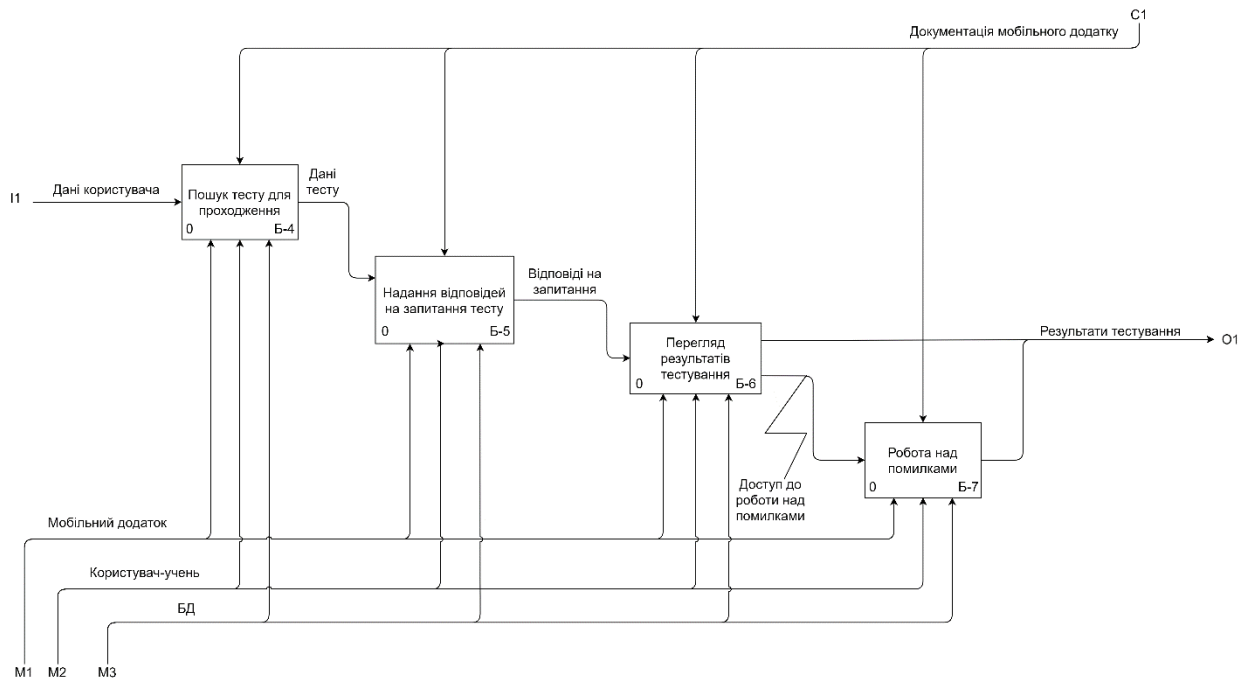


Рисунок 2.6 – Декомпозиція процесу «Проходження тестів» основного процесу мобільного додатку програмного комплексу

2.3 Діаграма варіантів використання веб-додатку програмного комплексу

Веб-додаток має розмежування надання доступу до інформації та функціоналу, що базується на даних авторизації користувача. Отже, акторами веб-додатку є «Зареєстрований користувач-вчитель» та «Незареєстрований користувач-вчитель». До операцій актора «Незареєстрований користувач-вчитель» відноситься лише реєстрація. До операцій актора «Зареєстрований користувач-вчитель» належать наступні:

- авторизація;
- створення, редагування та видалення тестів;
- створення, редагування та видалення запитань;
- збір статистики проходження тесту.

Діаграма варіантів використання веб-додатку зображена на рисунку 2.7.



Рисунок 2.7 – Діаграма варіантів використання веб-додатку

2.4 Діаграма варіантів використання мобільного додатку програмного комплексу

Мобільний додаток також має розмежування надання доступу до інформації та функціоналу користувачу-учню на основі авторизації. Під час тестування мобільним пристроєм фіксуються дії користувача, дані про які використовуються мобільним додатком для контролю проходження тесту. Отже, акторами мобільного додатку є «Зареєстрований користувач-учень», «Незареєстрований користувач-учень» та «Мобільний пристрій». До операцій актора «Незареєстрований користувач-учень» відноситься лише реєстрація. До операцій актора «Зареєстрований користувач-учень» належать наступні:

- авторизація;
- проходження тестів;

- надання відповідей на запитання;
- робота над помилками;
- збір статистики тестувань.

До операцій актора «Мобільний пристрій» належить фіксація дій користувача.

UML діаграма варіантів використання зображена на рисунку 2.8.

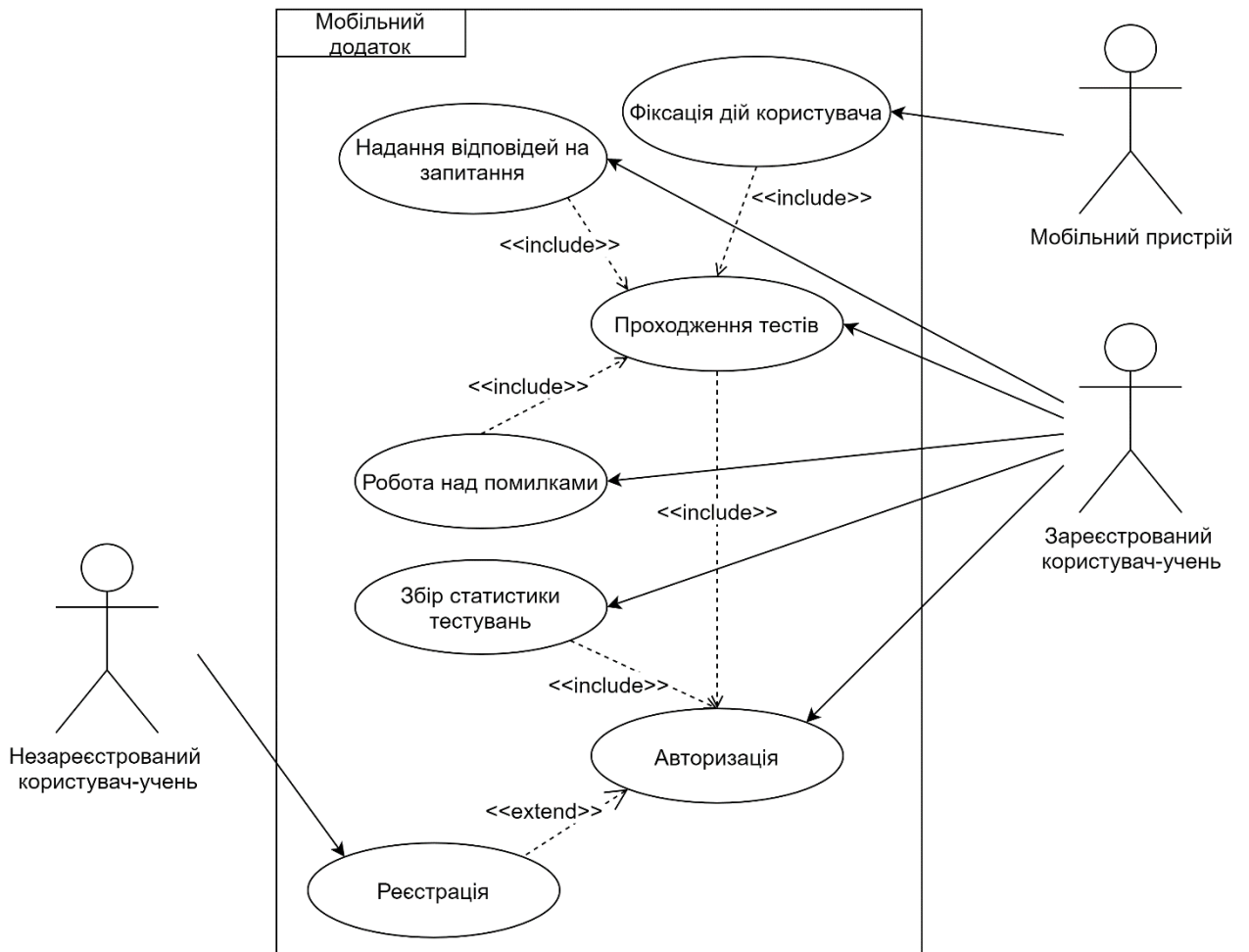


Рисунок 2.8 – Діаграма варіантів використання мобільного додатку

2.5 Проектування моделі БД програмного комплексу

База даних є спільною як для мобільного, так і для веб-додатку програмного комплексу. Вона складається з таблиць, що безпосередньо зберігають дані для відображення, та таблиць що необхідні для зв'язку записів між собою.

Опис таблиць бази даних наведений в таблиці 2.1.

Таблиця 2.1 – Таблиці бази даних

Назва таблиці	Опис даних
«Tests»	Записи з основною інформацією про тести
«TestsData»	Записи з властивостями та обмеженнями тестів
«TestTypes»	Записи з типами тестів
«Questions»	Записи з основною інформацією про запитання
«Answers»	Записи з варіантами відповідей на запитання
«TeacherUsers»	Записи з даними авторизації користувачів-вчителів
«TeacherData»	Записи з даними користувачів-вчителів
«PupilUsers»	Записи з даними авторизації користувачів-учнів
«PupilsData»	Записи з даними користувачів-учнів
«PupilTestCompletions»	Записи з інформацією про проходження тестів користувачами-учнями
«TeacherTests»	Записи для зв'язку тестів з користувачами-вчителями, що їх створили
«TestQuestions»	Записи для зв'язку питань з відповідними їм тестами
«QuestionAnswers»	Записи для зв'язку варіантів відповідей з відповідними їм запитаннями

Продовження таблиці 2.1

Назва таблиці	Опис даних
«QuestionResults»	Записи з результатами відповідей на запитання
«PupilResults»	Записи для зв'язку результатів відповідей на запитання з учнями, що їх дали

Логічна модель бази даних зображена на рисунку 2.6 у виді моделі «сутність-зв'язок».

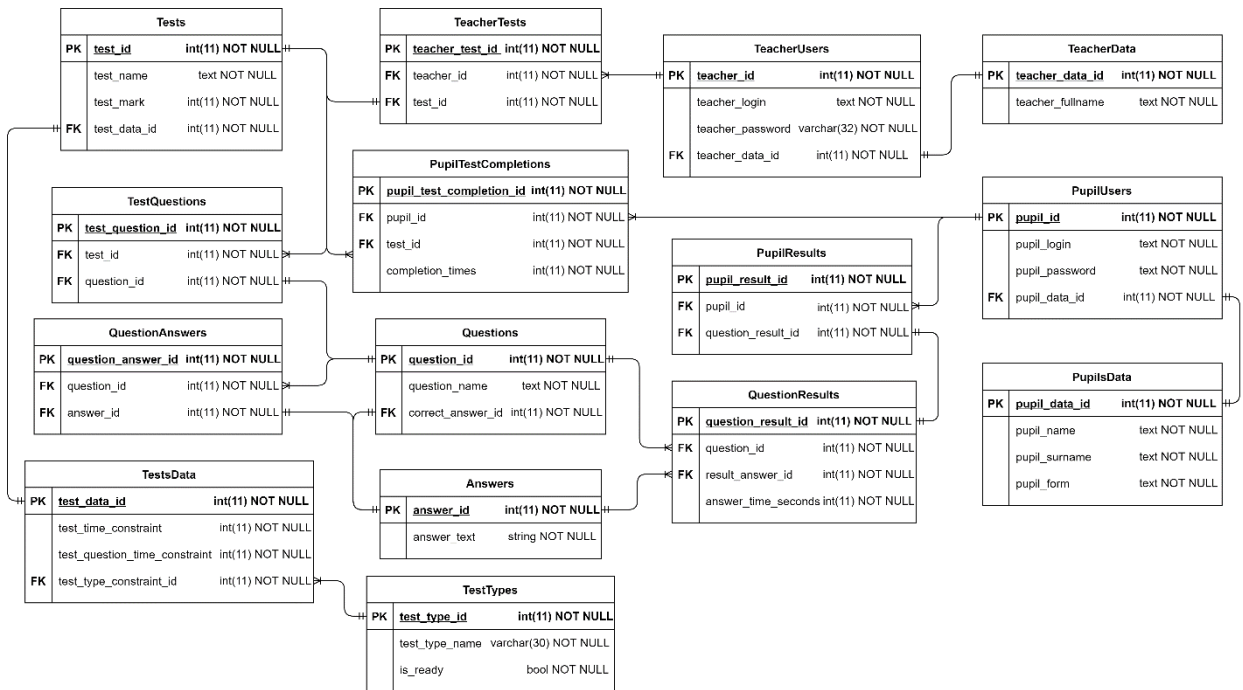


Рисунок 2.6 – ER-діаграма бази даних програмного комплексу

3 РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Архітектура програмного комплексу

Програмний комплекс буде складатися з мобільного та веб-додатків, що є орієнтованими під свій тип користувача, для веб-додатку це вчитель, для мобільного додатку – учень. Кожен з додатків використовує для обробки та відображення дані, що зберігаються у спільній базі даних, розташованій на серверній частині комплексу (рисунок 3.1). Таким чином дані, що тільки були занесені до БД вчителем за допомогою веб-додатку можуть бути використані учнями в мобільному додатку, і навпаки.

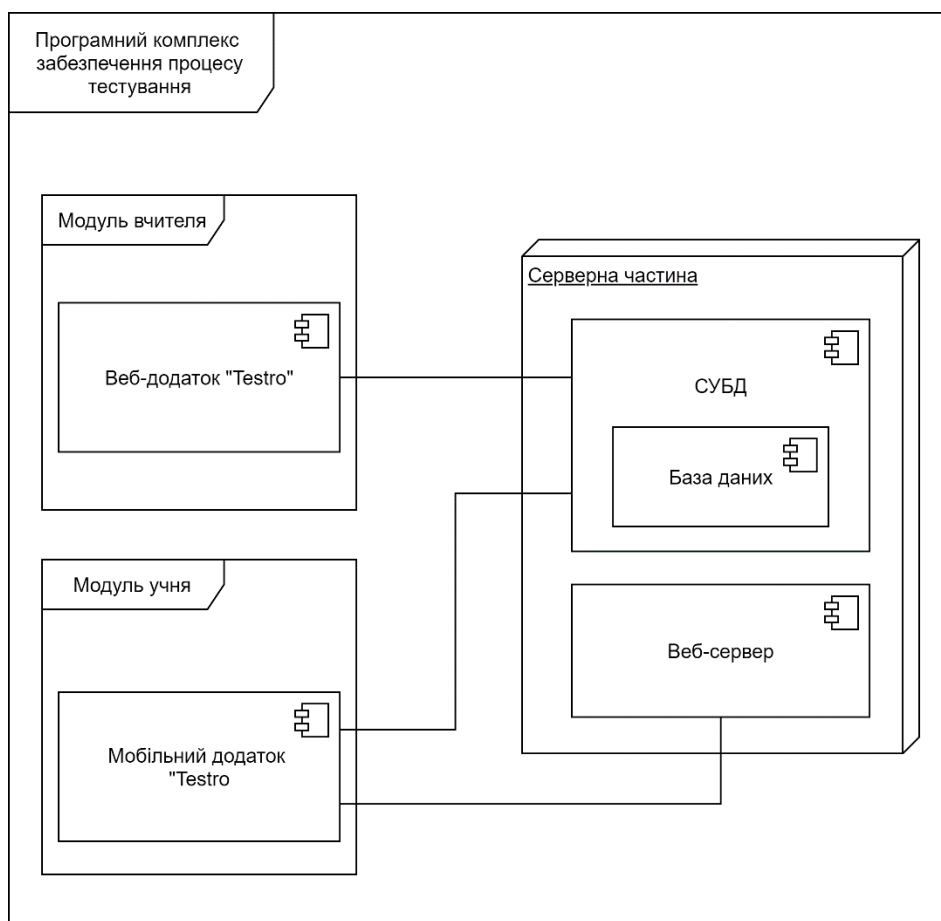


Рисунок 3.1 – Архітектура програмного комплексу

3.2 Архітектура веб-додатку

Веб-додаток розділений на клієнтську частину – Frontend, що відповідає за відображення даних користувачу та створення інтерфейсу для взаємодії з ними, та серверну частину – Backend, що відповідає за обробку запитів користувача до бази даних, відправку результатів запитів до клієнтської частини, а також підвантаження даних з файлової системи серверу [9].

Архітектура розроблюваного веб-додатку програмного комплексу зображена на рисунку 3.2.

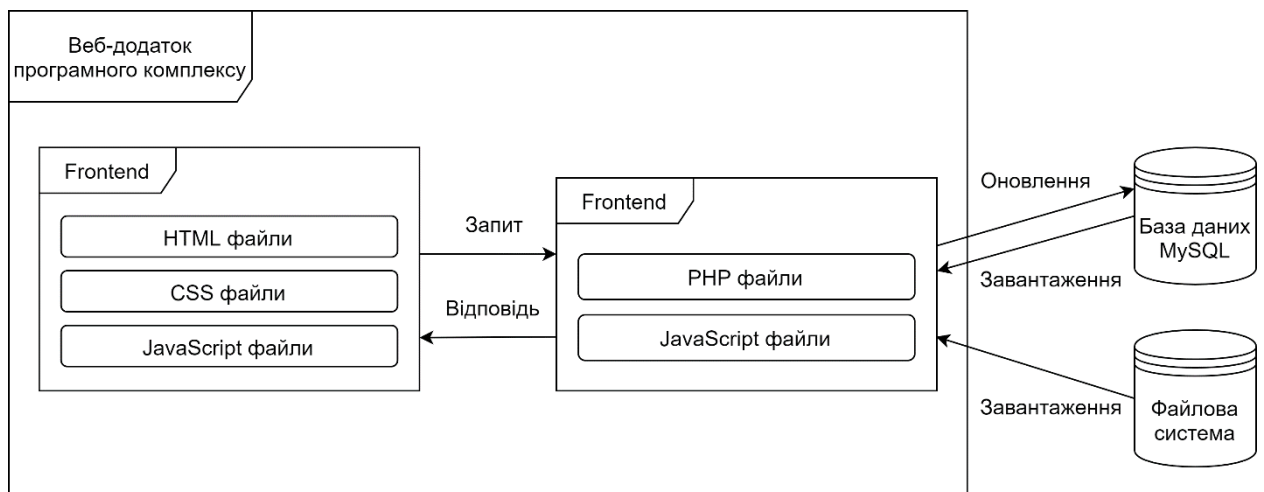


Рисунок 3.2 – Архітектура веб-додатку програмного комплексу

3.3 Архітектура мобільного додатку

З метою поліпшення роботи розроблюваного програмного продукту було обрано паттерн проектування MVVM (Model-View-ViewModel), який дозволяє виділити кожен з компонентів в окремі сутності моделі, інтерфейсу, та програмної логіки, що відповідають за збереження, відображення, а також обробку необхідних для роботи додатку даних [10].

Основними концепціями зазначеного підходу до розробки застосунку є:

- 1) користувач взаємодіє з View, який відповідає за відображення даних та обробку дій користувача, що впливає безпосередньо на нього;
- 2) ViewModel не має жодної інформації про View, існуючи автономно;
- 3) View містить посилання на ViewModel, отримуючи з нього дані та передаючи необхідні запити від користувача;
- 4) між View та ViewModel зберігається двосторонній зв'язок даних;
- 5) Model є контейнером для зберігання даних, які оброблюються за допомогою ViewModel.

Архітектура мобільного додатку зображена на рисунку 3.3.

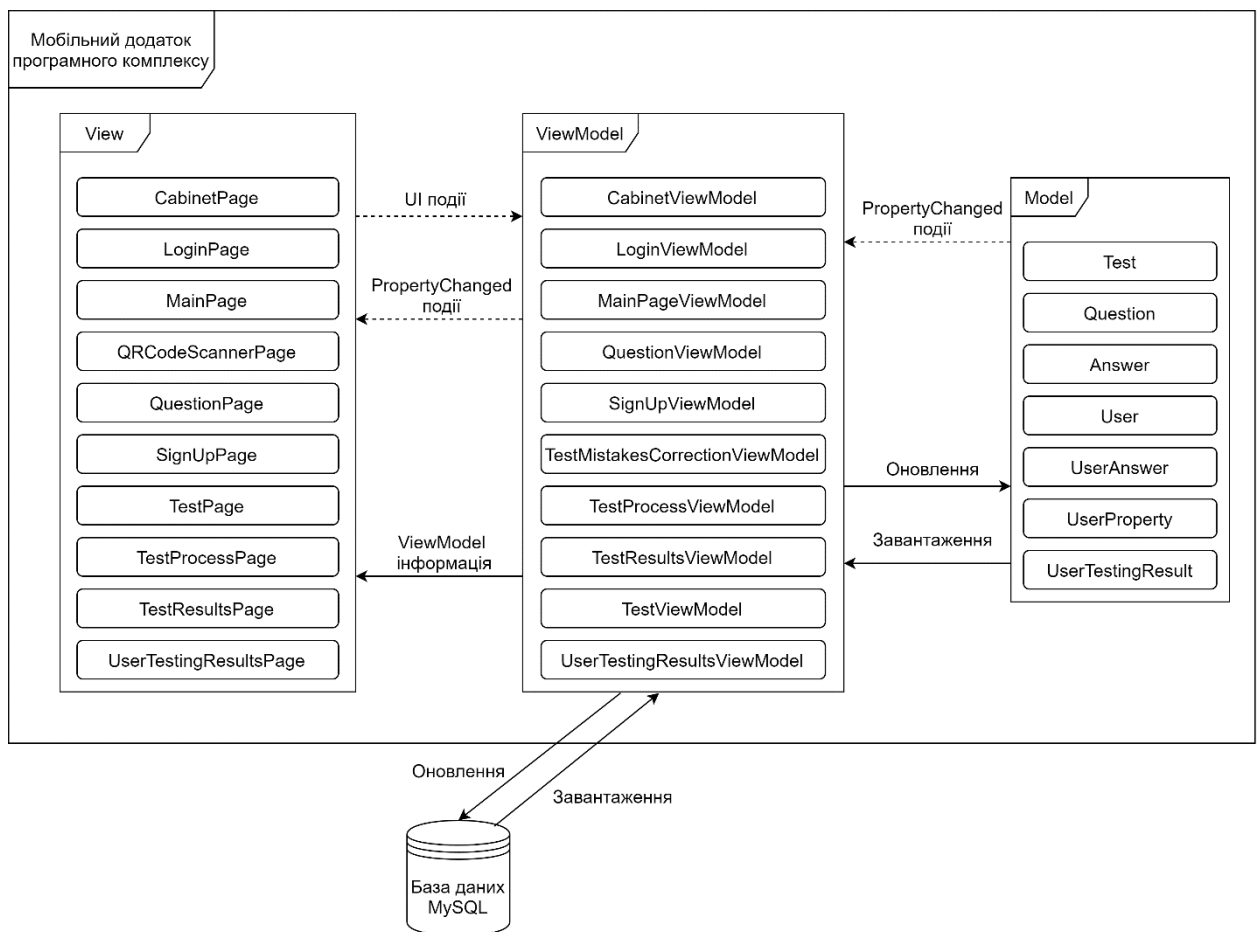


Рисунок 3.3 – Архітектура мобільного додатку програмного комплексу

3.4 Програмна реалізація веб-додатку

Для розробки веб-додатку в якості IDE було обрано WebStorm та звичний для веб-розробки набір мов програмування: HTML, CSS, JavaScript, PHP. Для локальної імітації роботи веб-серверу було обрано застосунок XAMPP, що в собі містить інструменти адміністрування як серверу, так і бази даних, а також їх створення та запуску в один клік (рис 3.4).

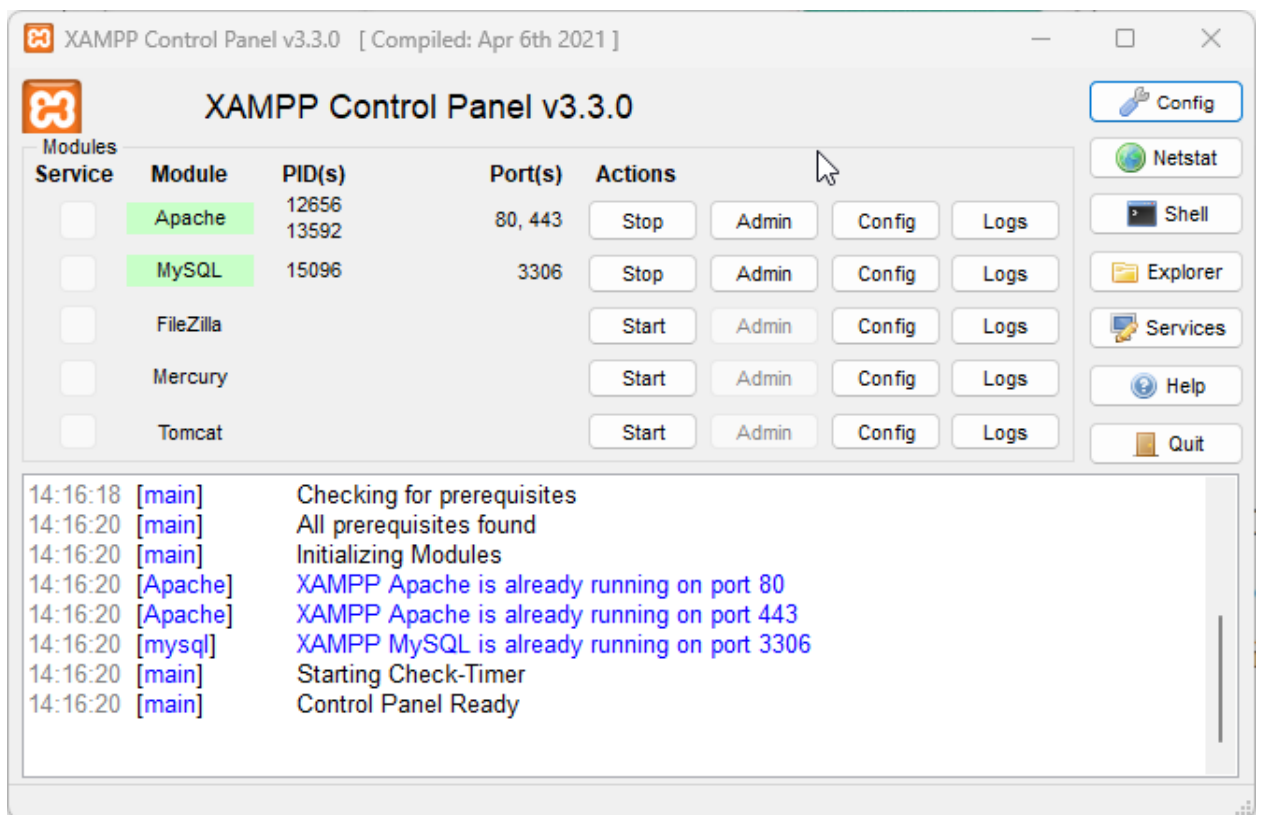


Рисунок 3.4 – Інтерфейс додатку XAMPP

Розробка починається зі створення пустого проекту, що знаходиться в файлового середовищі локального серверу, що був автоматично створений додатком XAMPP (рис 3.5).

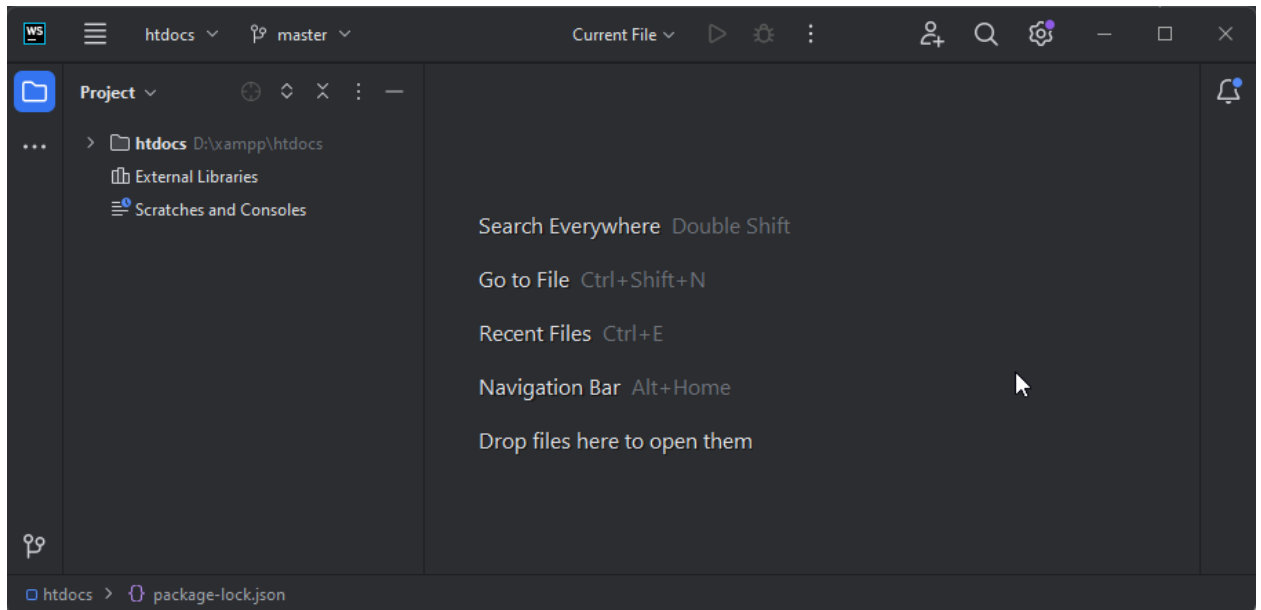


Рисунок 3.5 – Результат створення проекту веб-додатку

Першим етапом було організовано структуру проекту для розподілу файлів за їх типом та способом використання на окремі папки. Опис структури проекту зазначено в таблиці 3.1.

Таблиця 3.1 – Опис файлів в папках файлового середовища серверу

Папка	Опис файлів
Корінь проекту	PHP та HTML файли, що використовуються для відображення інформації в веб-додатку
«images»	Зображення в веб-додатку
«scripts»	JS скрипти веб-додатку
«styles»	CSS файли веб-додатку
«utils»	PHP файли для обробки запитів користувача

Результуюча структура проекту зображена на рисунках 3.6-3.7.

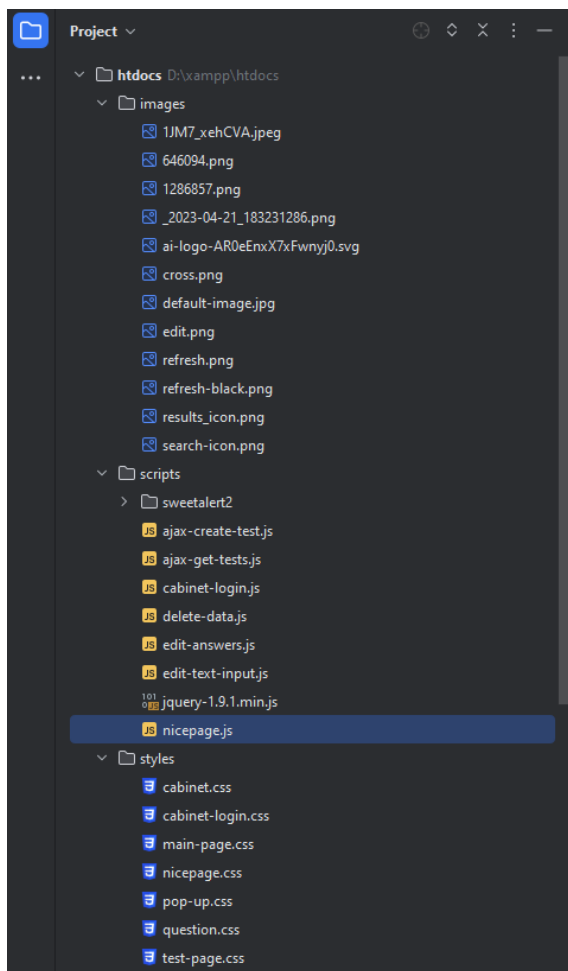


Рисунок 3.6 – Структура проекту.

Частина 1

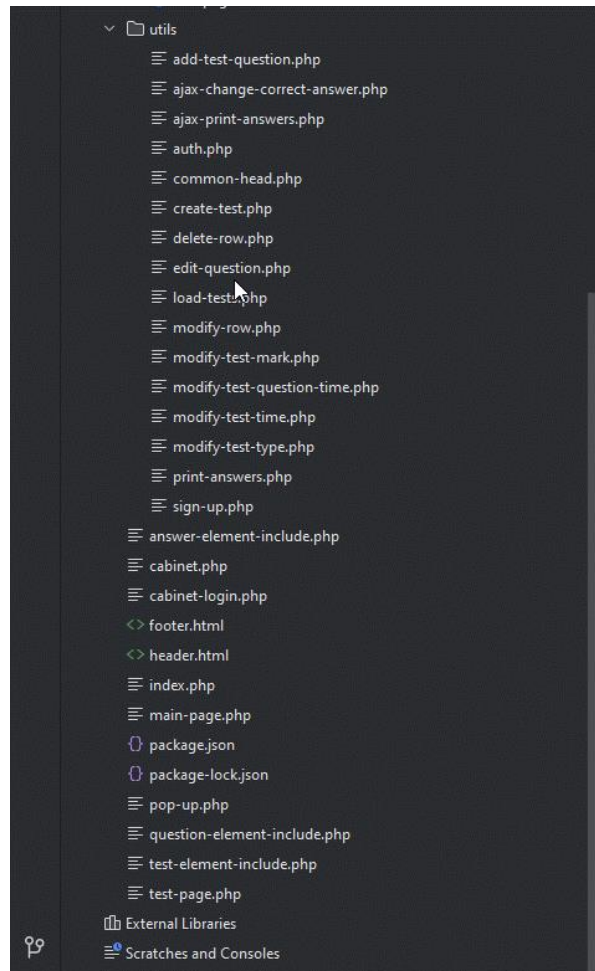


Рисунок 3.7 Структура проекту.

Частина 2

Опис файлів з програмними кодами веб-додатку наведено в таблиці 3.2.

Таблиця 3.2 – Опис файлів з програмними кодами веб-додатку

Файл	Опис файлів
scripts/sweetalert2.js	Відображення поп-ап діалогових вікон користувачу
scripts/ajax-create-test.js	Асинхронне створення тесту
scripts/ajax-delete-data.js	Асинхронне видалення тестів та питань
scripts/ajax-edit-answers.js	Асинхронне редагування варіантів відповідей на запитання

Продовження таблиці 3.2

Файл	Опис файлів
scripts/ajax-edit-text-input.js	Асинхронне редагування текстових полів, що відображуються на сторінках
scripts/ajax-get-tests.js	Асинхронне отримання списку тестів з або без фільтрації
scripts/cabinet-login.js	Обробка дій користувача на сторінці входу в кабінет
styles/cabinet.css	Стилі сторінки «Кабінет»
styles/cabinet-login.css	Стилі сторінки «Вхід»
styles/main-page.css	Стилі сторінки «Головна»
styles/pop-up.css	Стилі секції «Запитання» на сторінці «Тест»
styles/question.css	Стилі елемента «Запитання» на сторінці «Тест»
styles/test-page.css	Стилі сторінки «Тест»
utils/add-test-question.php	Обробка запиту на додавання нового питання до тесту на сторінці «Тест»
utils/auth.php	Обробка авторизації на сторінці «Вхід»
utils/change-correct-answer.php	Обробка запиту на зміну правильної відповіді в запитанні тесту
utils/common-head.php	Файл з підключенням загальних стилів та скриптів
utils/create-test.php	Обробка запиту на створення нового тесту на сторінці «Кабінет»
utils/delete-row.php	Обробка запиту на видалення запису у відповідності до наданих обмежень
utils/modify-row.php	Обробка запиту на редагування запису у відповідності до наданих обмежень

Продовження таблиці 3.2

Файл	Опис файлів
utils/ modify-drop-down-row.php	Обробка запиту на редагування запису, що відображений на сайті у вигляді випадаючого списку, у відповідності до наданих обмежень
utils/print-answers.php	Відображення варіантів відповідей на запитання в секції «Запитання» на сторінці «Тест»
utils/sign-up.php	Обробка реєстрації на сторінці «Вхід»
answer-element-include.php	Елементу «Варіант відповіді» в секції «Запитання» на сторінці «Тест»
cabinet.php	Сторінка «Кабінет»
cabinet-login.php	Сторінка «Вхід»
footer.html	Елемент «Footer» сторінок веб-додатку
header.html	Елемент «Header» сторінок веб-додатку
index.php	Початкова сторінка, що використовується для переходу на сторінку «Головна»
main-page.php	Сторінка «Головна»
pop-up.php	Спливаючий елемент «Запитання» на сторінці «Тест»
question-element-include.php	Елемент «Запитання» сторінки «Тест»
test-element-include.php	Елемент «Тест» сторінки «Кабінет»
test-page.php	Сторінка «Тест»

Для забезпечення оновлення даних на сторінках додатку без їх оновлення був використаний підхід асинхронної відправки запитів на сервер та підвантаження отриманих результатів засобами Ajax [11].

З метою уникнення проблем відображення сторінок веб-додатку на різних пристроях в процесі розробки було використано підхід із створення динамічних макетів сторінок за допомогою flexbox контейнерів, що підлаштовуються властивостей браузеру та власного наповнення з метою забезпечення коректного вирівнювання елементів під час їх безпосереднього відображення [12].

Порівняння відображення наповнення сайту на пристроях з різним розміром екрану зображено на рисунках 3.8-3.9.

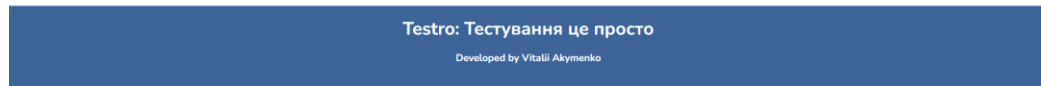
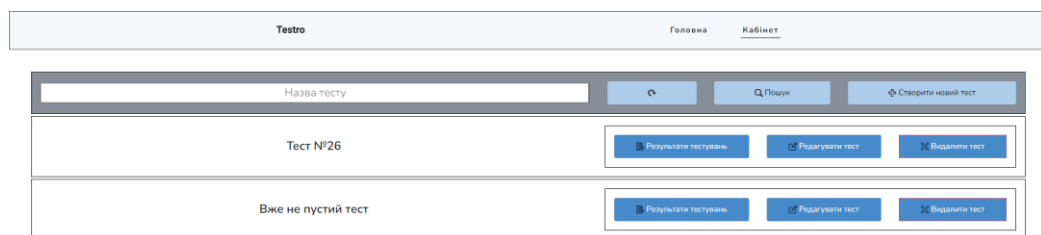


Рисунок 3.8 – Вигляд сайту на пристрої з широким екраном

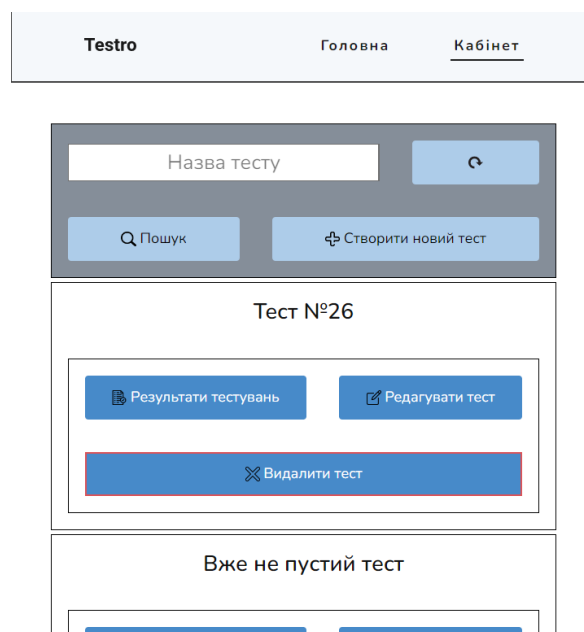


Рисунок 3.9 – Вигляд сайту на пристрої з вузьким екраном

3.5 Програмна реалізація мобільного додатку

Для розробки мобільного додатку в якості IDE було обрано VisualStudio та кроссплатформенний фреймворк Xamarin.Forms, який розширяє функціонал .Net платформи, створюючи зручний інструмент для розробки мобільних застосунків під платформи Android, Windows та iOS [13].

Розробка починається зі створення пустого проекту шляхом вибору шаблону додатку та потрібних платформ. Обираємо Tabbed, Android та iOS (рис. 3.10) і продовжуємо.

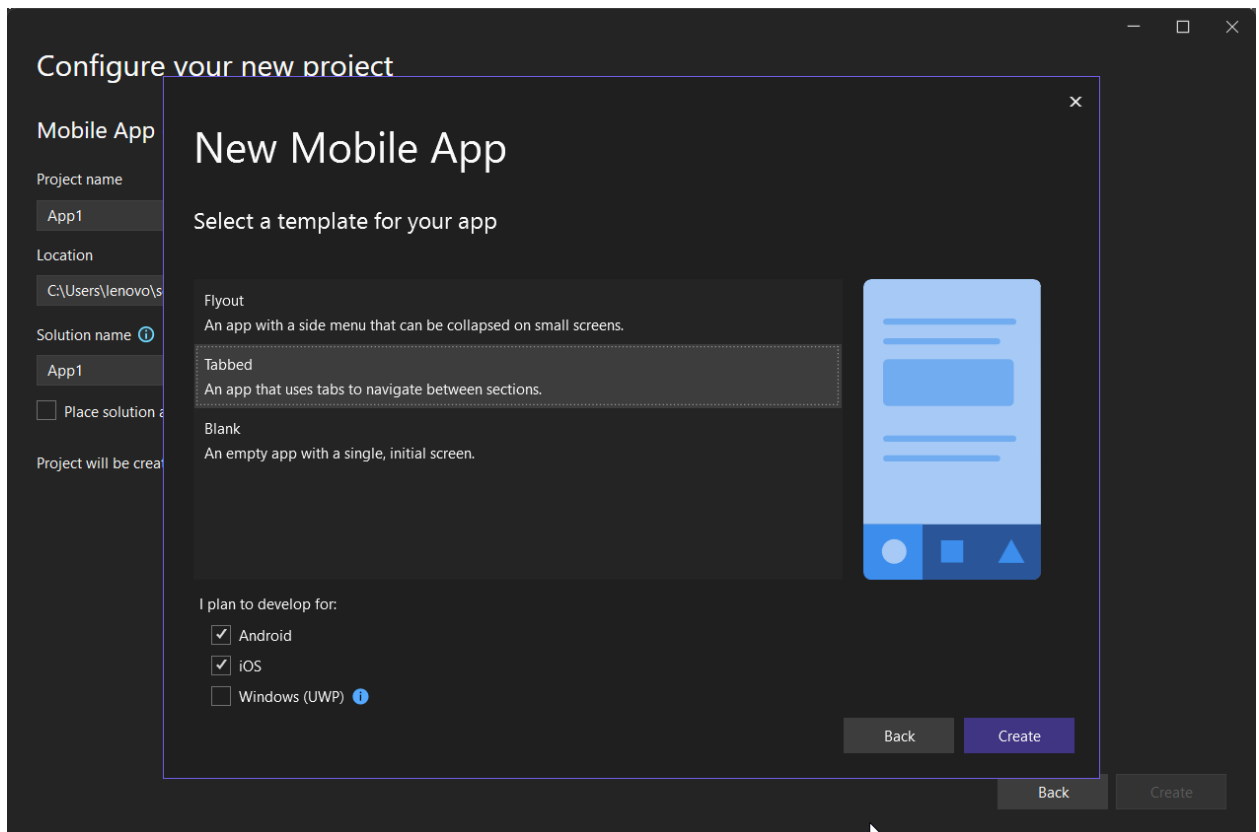


Рисунок 3.10 – Створення пустого проекту

В результаті був автоматично створений шаблон проекту з правильною файловою структурою, орієнтованою під розробку MVVM застосунків, готовою до заповнення (рис. 3.11).

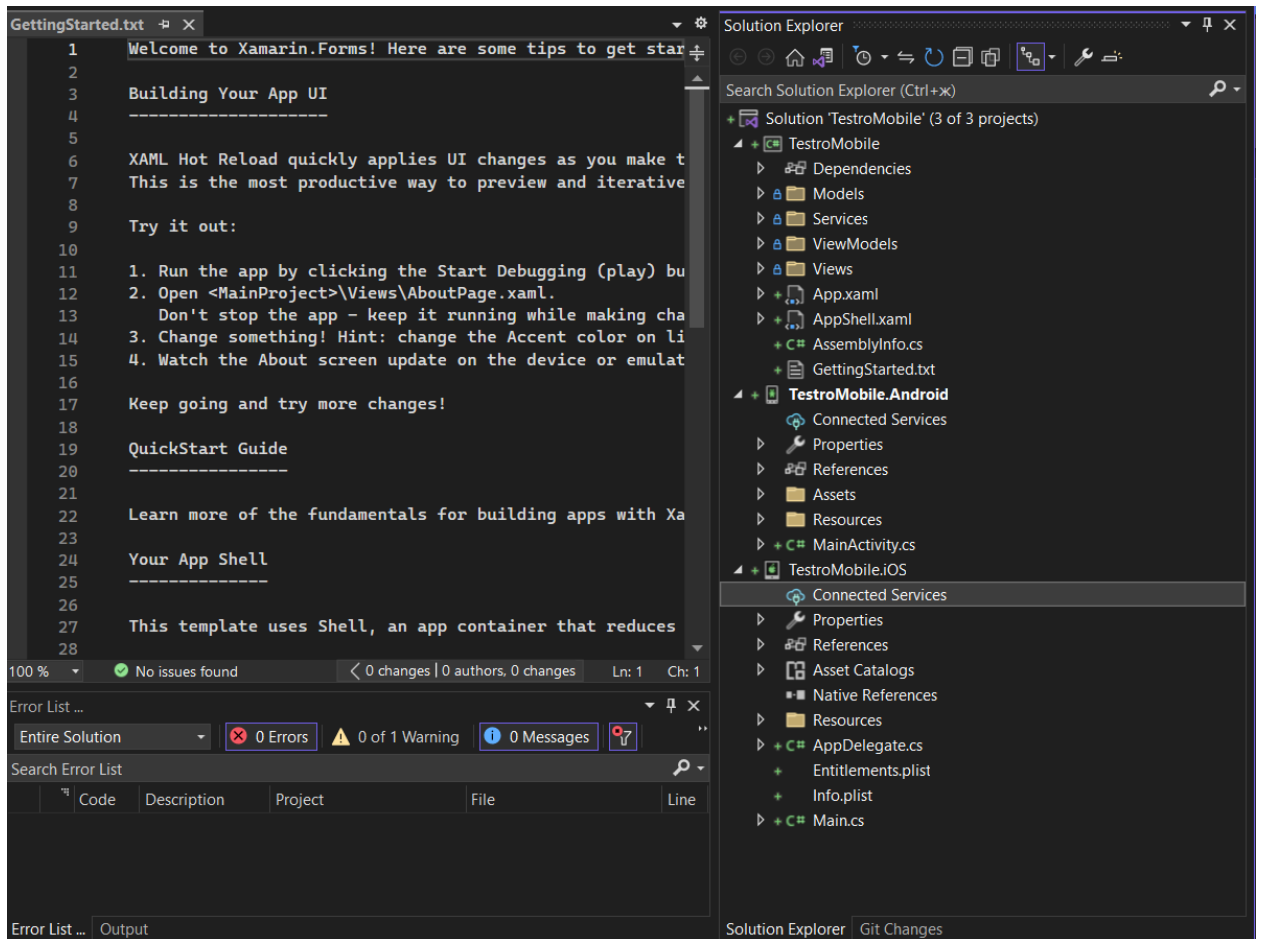


Рисунок 3.11 – Створений проект

У результаті роботи були розроблені файли моделей, сторінок, а також видів сторінок. Опис розроблених файлів з програмними кодами наведено в таблиці 3.3.

Таблиця 3.3 – Опис програмних файлів мобільного застосунку

Файл	Опис файлів
BaseNotify.cs	Базовий опис сутності, що створює двосторонній зв'язок між Model та ViewModel
Answer.cs	Опис сутності даних відповіді на запитання
Question.cs	Опис сутності даних запитання
Test.cs	Опис сутності даних тесту
User.cs	Опис сутності даних користувача

Продовження таблиці 3.3.

Файл	Опис файлів
UserAnswer.cs	Опис сутності відповіді користувача на запитання
UserProperty.cs	Опис сутності даних користувача
UserTestingResult.cs	Опис сутності результатів проходження тесту користувачем
BaseViewModel.cs	Базовий опис сутності для створення інструменту зі взаємодією між ViewModel та Model/View сутностями
BaseTestModel.cs	Базовий опис сутності моделі перегляду, що працює з даними тесту
CabinetViewModel.cs	Опис моделі перегляду сторінки «Кабінет»
LoginViewModel.cs	Опис моделі перегляду сторінки «Вхід»
SignUpViewModel.cs	Опис моделі перегляду сторінки «Реєстрація»
MainPageViewModel.cs	Опис моделі перегляду сторінки «Головна»
QuestionViewModel.cs	Опис моделі перегляду сторінки «Запитання»
TestMistakesCorrection ViewModel.cs	Опис моделі перегляду сторінки «Робота над помилками»
TestProcessViewModel.cs	Опис моделі перегляду сторінки проходження тесту
TestResultsViewModel.cs	Опис моделі перегляду сторінки результатів поточного тестування
TestViewModel.cs	Опис моделі перегляду сторінки «Тест»
UserTestingResultsViewM odel.cs	Опис моделі перегляду сторінки «Результати користувача»
CabinetPage.xaml	Сторінка «Кабінет»
LoginPage.xaml	Сторінка «Вхід»
MainPage.xaml	Сторінка «Головна»

Продовження таблиці 3.3.

QRCodeScannerPage.xaml	Сторінка сканеру QR-кодів
QuestionPage.xaml	Сторінка «Запитання»
SingUpPage.xaml	Сторінка «Реєстрація»
TestPage.xaml	Сторінка «Тест»
TestProcessPage.xaml	Сторінка підтримки процесу проходження тесту
TestResultsPage.xaml	Сторінка результатів поточного тестування
UserTestingResultsPage.xaml	Сторінка «Результати користувача»
App.xaml	Сутність застосунку
AppShell.xaml	Сутність обгортки застосунку

Всі сторінки мобільного застосунку були розроблені декларативною мовою розмітки XAML, яка за допомогою деревовидної структури з комбінації стандартних або вручну створених елементів описує об'єкт динамічного виду сторінки [14]. Приклад сторінки мобільного застосунку мовою XAML представлено на рисунку 3.12.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3      xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4      x:Class="Testro.Views.MainPage"
5      NavigationPage.HasBackButton="False"
6      xmlns:vm="clr-namespace:Testro.ViewModels" xmlns:zxing="clr-namespace:ZXing.Net.Mobile.Forms;assembly=ZXing.Net.Mobile.Forms"
7      Title="{Binding Title}">
8
9      <ContentPage.BindingContext>
10         <vm:MainPageViewModel />
11     </ContentPage.BindingContext>
12
13     <ContentPage.Resources>
14         <ResourceDictionary>
15             <Color x:Key="Accent" Color="#96d1ff"/>
16         </ResourceDictionary>
17     </ContentPage.Resources>
18
19     <ContentPage.Content>
20         <StackLayout Padding="10,0,10,0" VerticalOptions="Center" Spacing="10">
21             <Grid>
22                 <Grid.ColumnDefinitions>
23                     <ColumnDefinition Width="*" />
24                     <ColumnDefinition Width="*" />
25                 </Grid.ColumnDefinitions>
26                 <Entry Text="{Binding ActiveTest}"
27                     Placeholder="Запитання"
28                     FontSize="Title"
29                     Grid.Column="0" />
30                 <Button VerticalOptions="Center"
31                     Margin="0,10,0,0"
32                     Text="Сканувати QR код"
33                     Command="{Binding ScanQRCodeCommand}"
34                     BackgroundColor="{StaticResource Primary}"
35                     TextColor="White"
36                     Grid.Column="1" />
37             </Grid>
38             <Button VerticalOptions="Center"
39                 Margin="0,10,0,0"
40                 Text="Нехайти до тесту"
41                 Command="{Binding GoToTestCommand}"
42                 IsEnabled="{Binding GoToTestButtonEnabled}"
43                 BackgroundColor="Green"
44                 TextColor="White" />
45         </StackLayout>
46     </ContentPage.Content>
47 </ContentPage>
48
49

```

Рисунок 3.12 – Програмний код сторінки «Головна»

3.6 Використання веб-додатку

Після запуску додатку з'являється головна сторінка веб-додатку, що містить інформацію про платформу (рис.3.13).

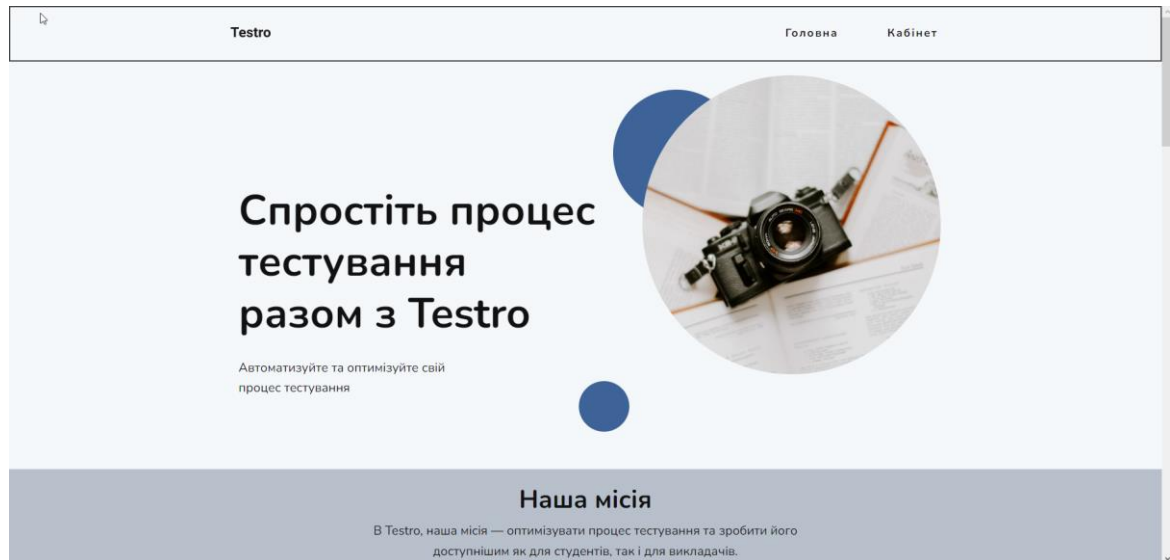


Рисунок 3.13 – Головна сторінка веб-додатку

Для продовження роботи з додатком необхідно перейти до пункту меню «Кабінет», що знаходиться в шапці сайту. Після переходу, з'являється сторінка входу в акаунт користувача (рис. 3.14). Якщо користувач не є зареєстрованим в системі – йому доступна форма реєстрації, перейти на яку можна перемикачем вгорі основного блоку сторінки (рис. 3.15).

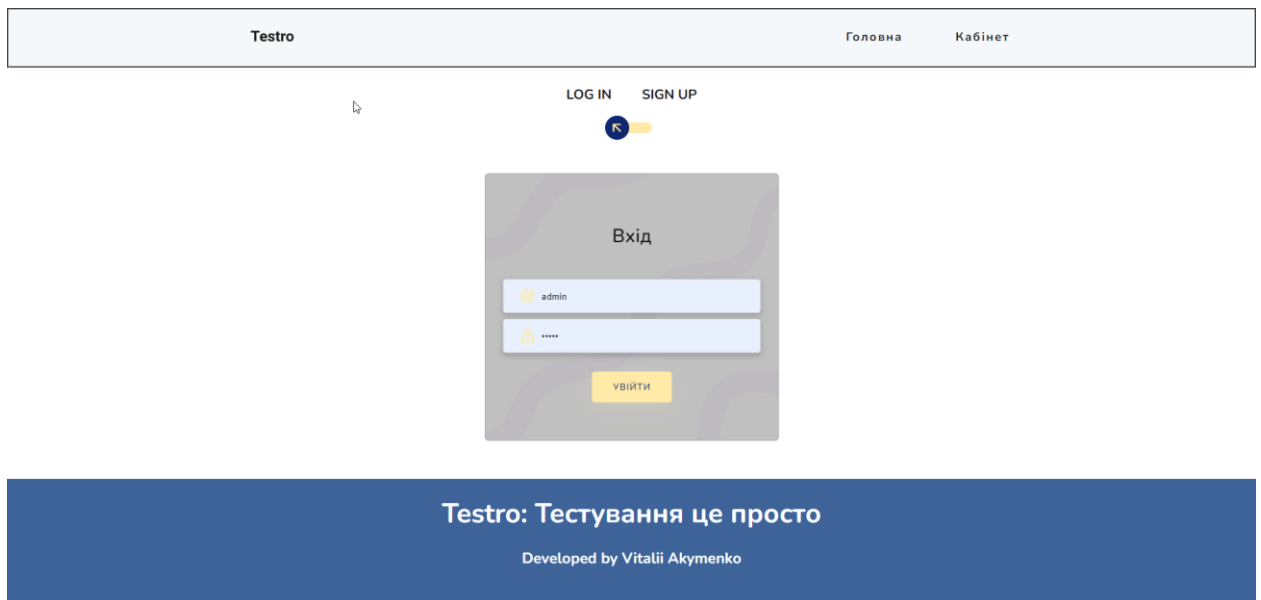


Рисунок 3.14 – Сторінка входу в аккаунт

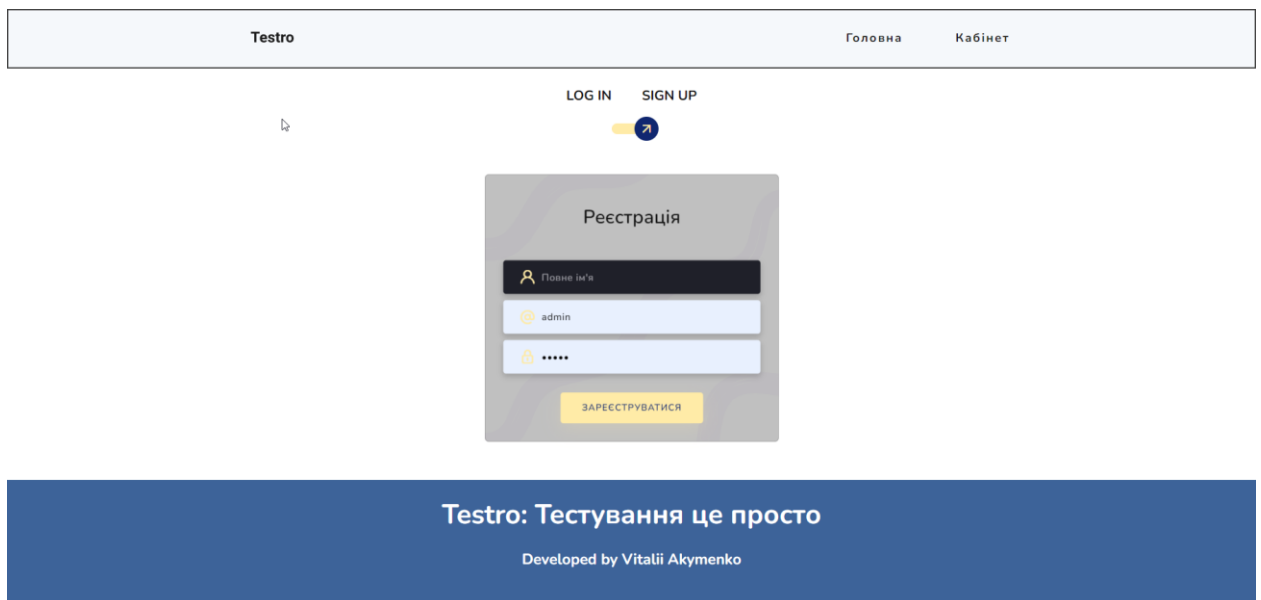


Рисунок 3.15 – Форма реєстрації

Інформація про користувача зберігається та використовується при повторних входах до кабінету.

Після успішного входу користувача буде переадресовано на сторінку кабінету, де зображені його тести та форма пошуку тестів. Якщо жодного тесту не було створено – користувач побачить повідомлення про це (рис. 3.16).

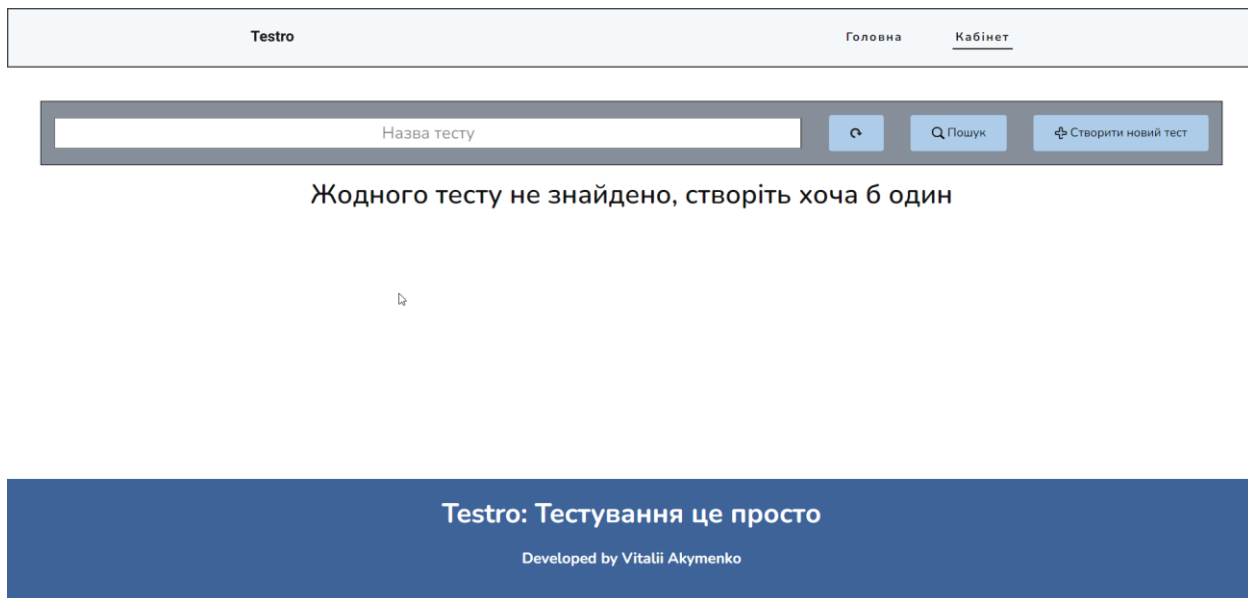


Рисунок 3.16 – Кабінет користувача

Скориставшись кнопкою в верхній частині основного блоку сторінки користувач може створити новий пустий тест (рис. 3.17) та одразу перейти до його редагування.

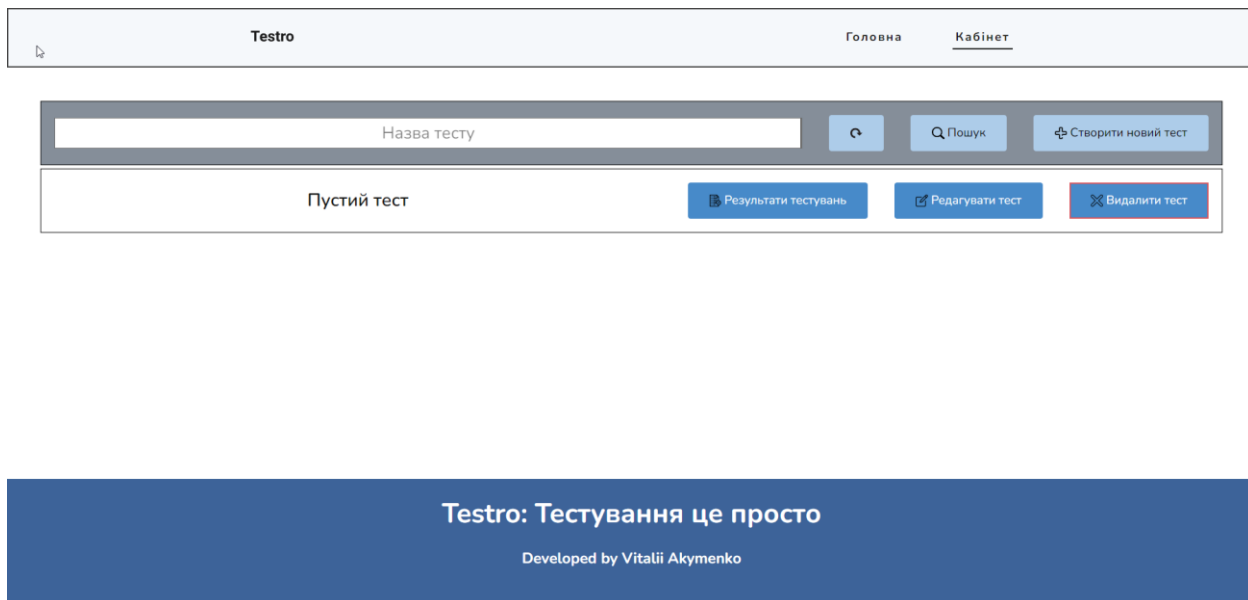


Рисунок 3.17 – Результат створення тесту

Сторінка «Тест» містить випадаючі списки редагування властивостей тесту, а також кнопки створення запитань та видалення тесту (рис. 3.18).

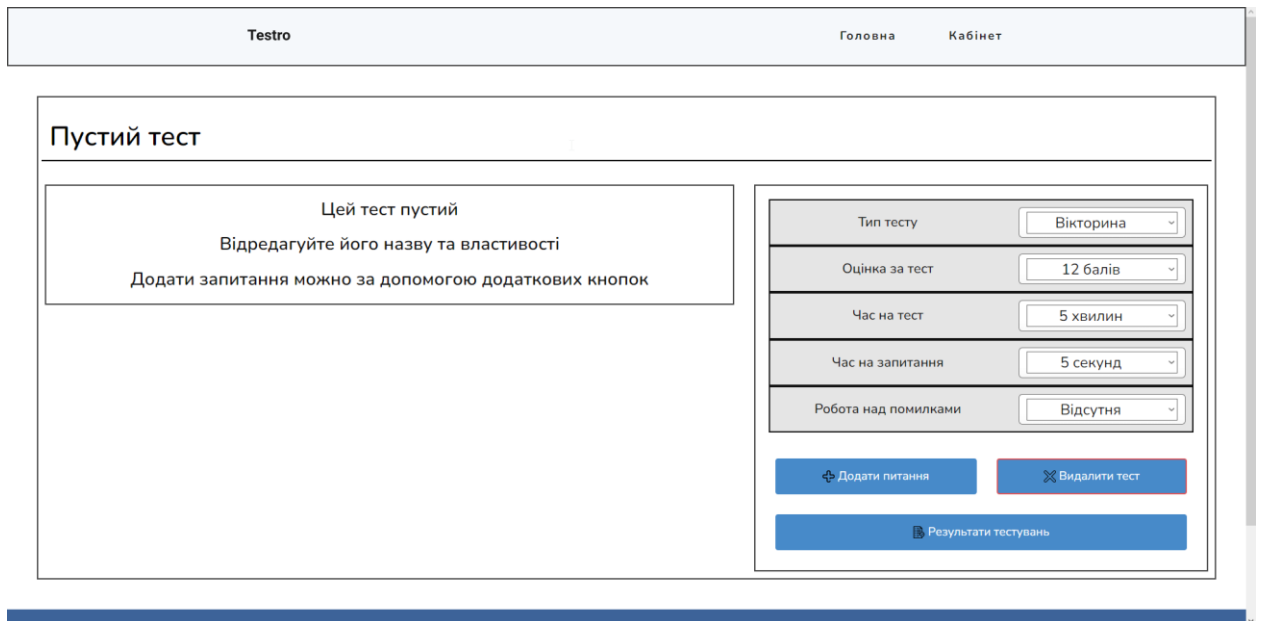


Рисунок 3.18 – Сторінка редагування тесту

Так само як з тестом, користувач може редагувати та видалити запитання відповідними кнопками, що відображаються окремо для кожного запитання (рис. 3.19).

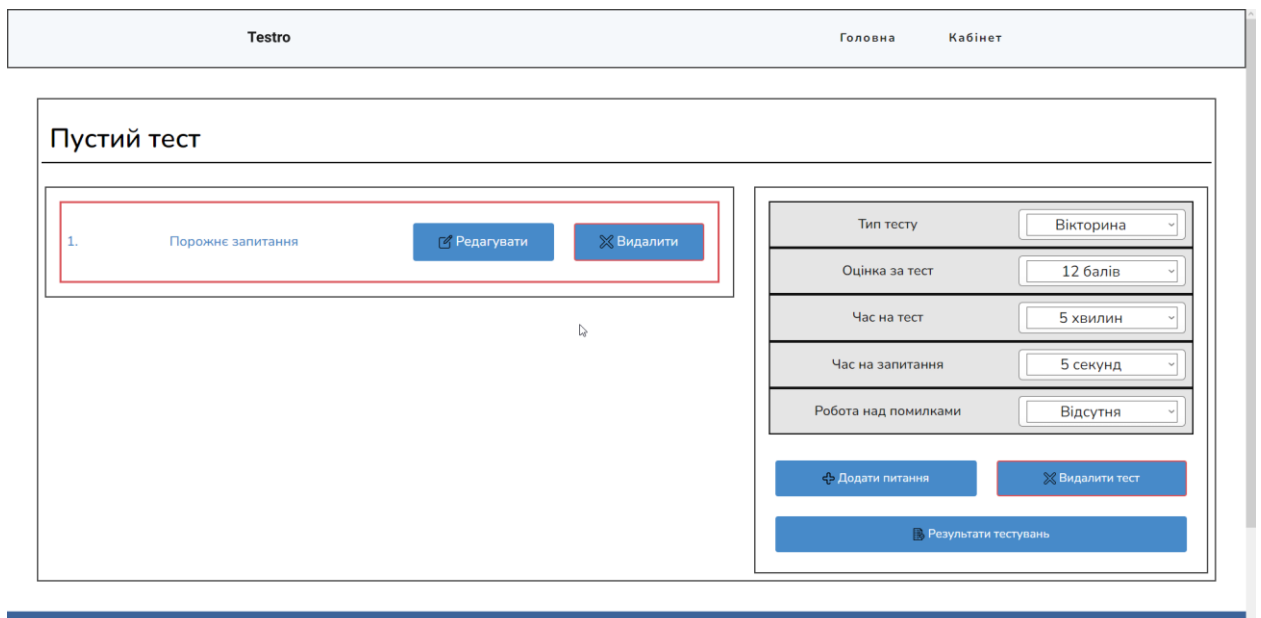


Рисунок 3.19 – Результат додавання запитання

Сторінка редагування тесту представляє собою модульне вікно, що зображене поперх сторінки тесту (рис. 3.20).

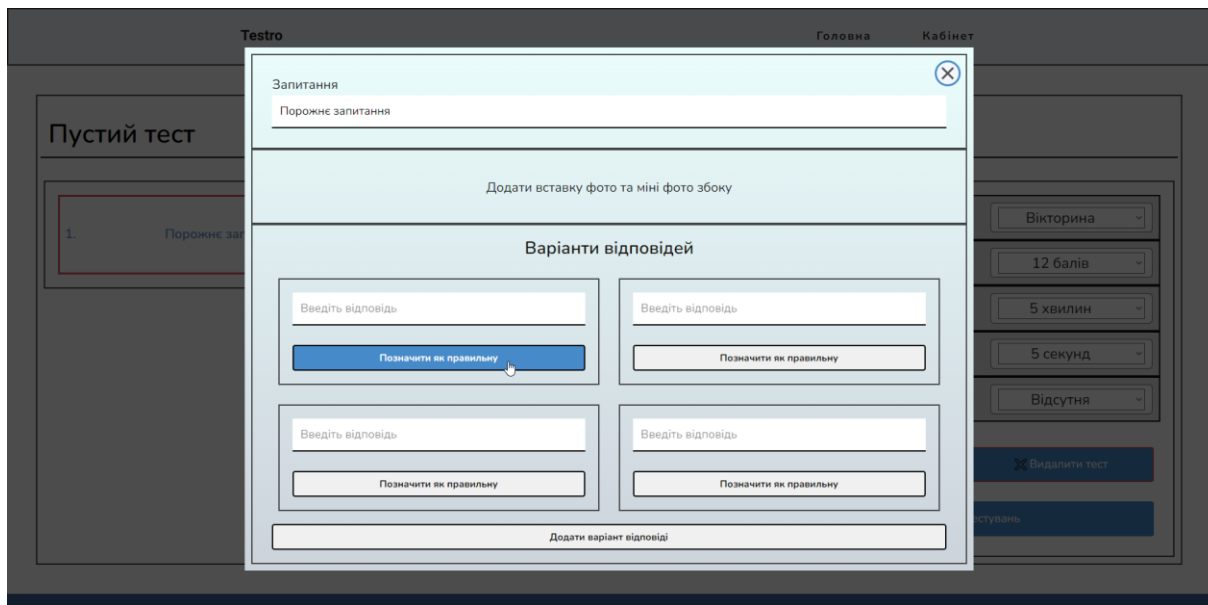


Рисунок 3.20 – Сторінка редагування тесту

Для редагування текстових полів користувачу достатньо ввести необхідне значення, після чого поле підсвітиться зеленим, сигналізуючи про успішність операції (рис. 3.21).

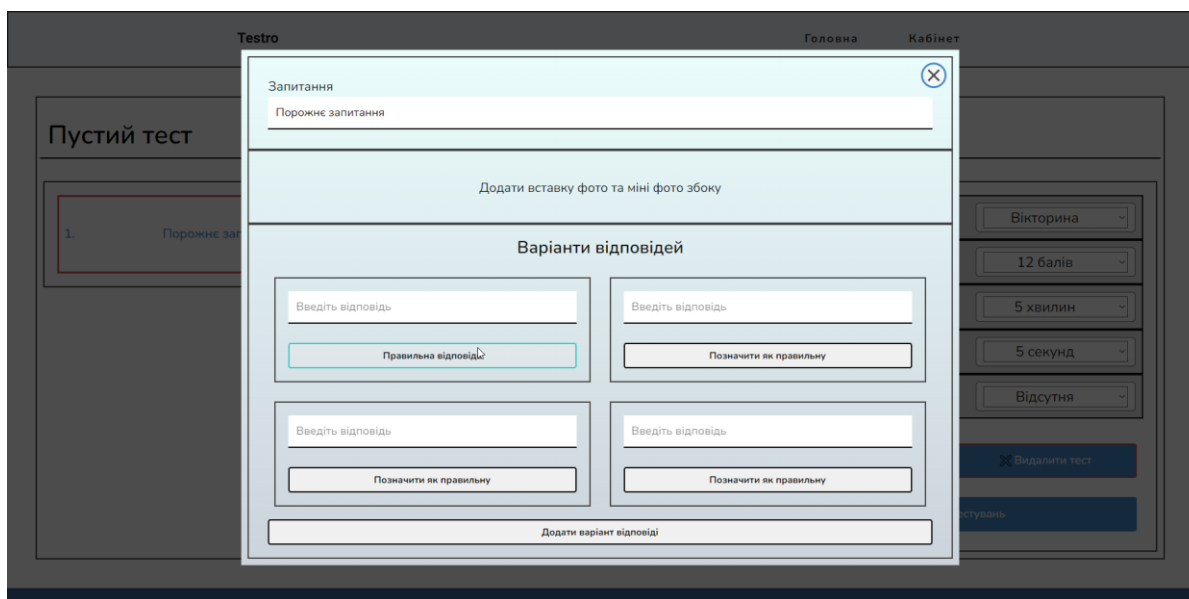


Рисунок 3.21 – Результат встановлення правильної відповіді та редагування назви запитання

Відповідно видалення запитань та тестів супроводжується спливаючими вікнами підтвердження операції. Приклад такого вікна зображено на рисунку 3.22.

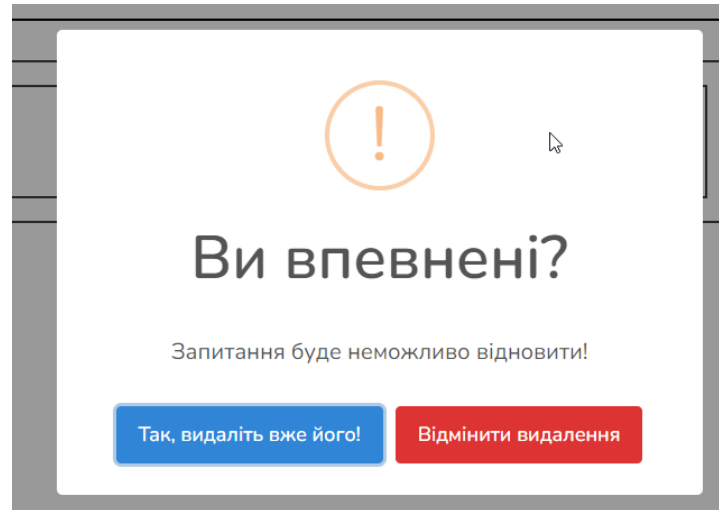


Рисунок 3.22 – Приклад вікна підтвердження запитання тесту

Користувачу надана можливість змінювати кількість варіантів відповідей від 4 до 6, що можна зробити за допомогою натискання на кнопки «Додати варіант відповіді» та «Видалити відповідь» (рис. 3.23).

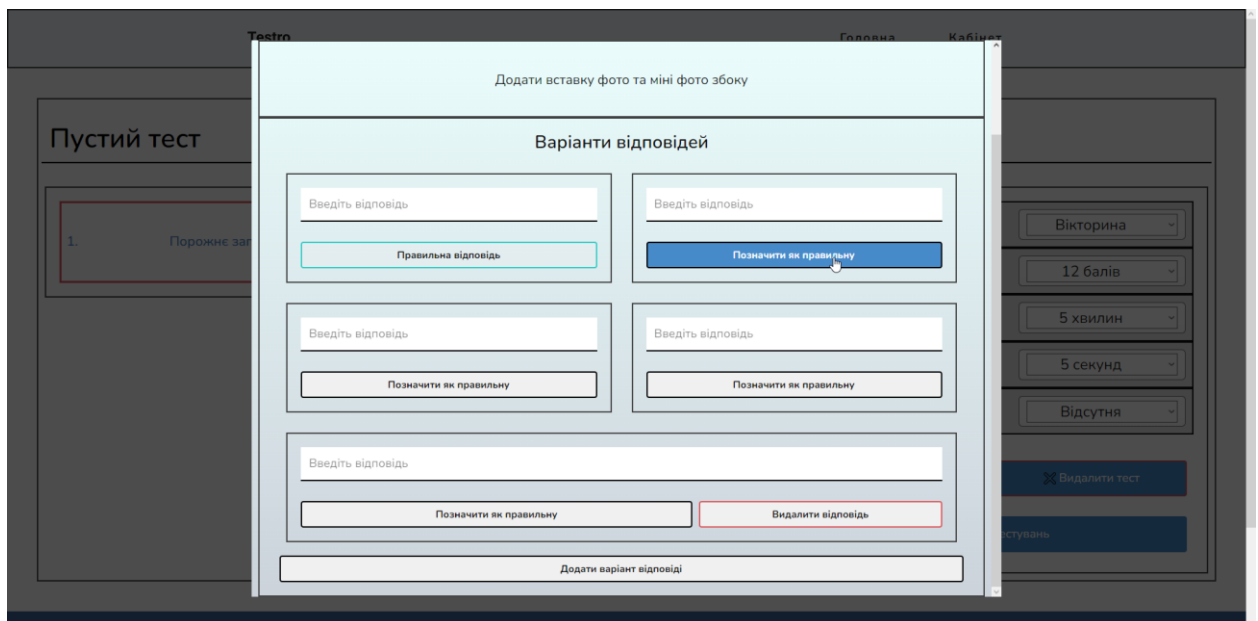


Рисунок 3.23 – Приклад додавання варіанту відповіді до запитання тесту

До кожного тесту користувач-вчитель може переглядати результати тестувань учнів у вікні «Результати тестування» (рис. 3.24).

Testro		Головна	Кабінет
<input type="text" value="Ім'я учня для пошуку"/>			
		<input type="button" value="↻"/>	<input type="button" value="🔍 Пошук"/>
Artem Akymenko	2 / 2	9	<input type="button" value="📄 Детальні результати учня"/>

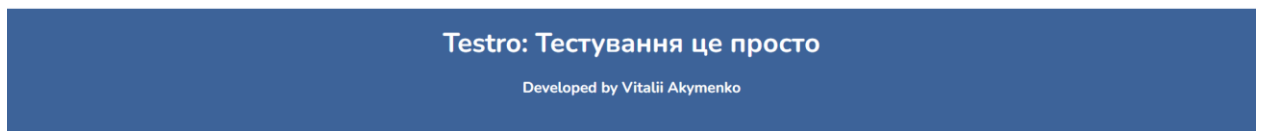


Рисунок 3.24 – Приклад результатів тестування

3.7 Використання мобільного додатку

Після запуску мобільного додатку користувача зустрічає сторінка авторизації в кабінет користувача-учня. У разі якщо користувач не є зареєстрованим, він може перейти до сторінки реєстрації внизу сторінки авторизації (рис. 3.25). Сторінка реєстрації зображена на рисунку 3.26.

На кожній зі сторінок присутня валідація даних, що представляє собою недоступність кнопок «Увійти» та «Зареєструватись» без наявності даних в необхідних полях сторінок.

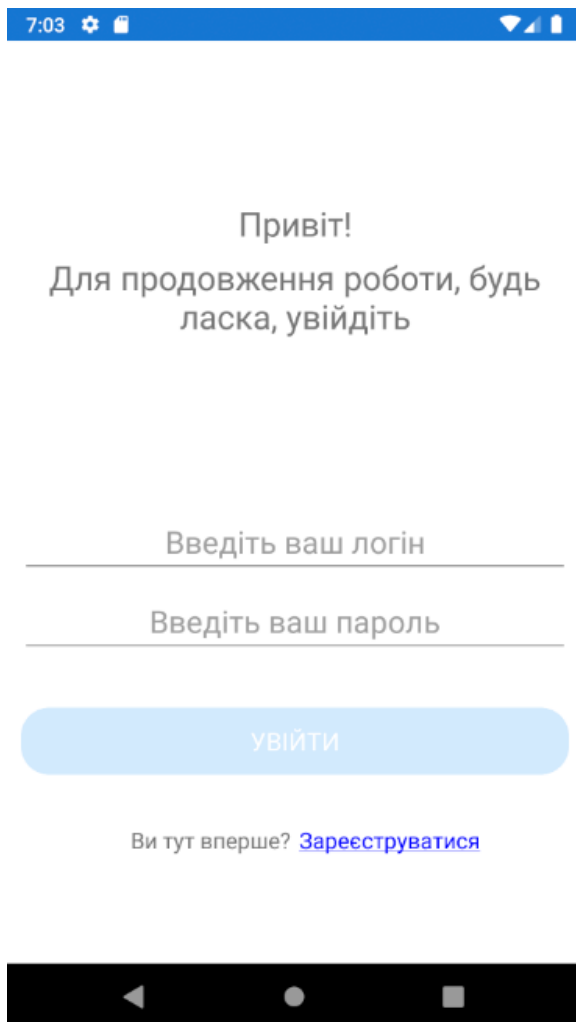


Рисунок 3.25 – Сторінка авторизації



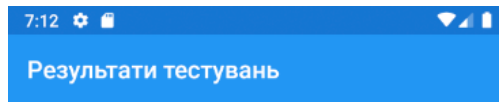
Рисунок 3.26 – Сторінка реєстрації

В результаті успішного входу користувача буде переадресовано на головну сторінку зі збереженням даних для наступних входів.

На головній сторінці знаходяться поля вводу унікального номеру тесту та кнопка переходу до тесту. Номер тесту може бути автоматично отриманий з QR-коду шляхом його сканування після натискання відповідної кнопки на сторінці. Внизу сторінки знаходиться меню для швидкої навігації між сторінками «Головна» (рис. 3.27), «Результати», що містить результати тестувань користувача (рис. 3.28), та «Кабінет», що містить інформацію користувача (рис 3.29). Дані користувача можуть бути змінені, про результат операції користувача буде повідомлено спливаючим вікном (рис. 3.30).



Рисунок 3.27 – Головна сторінка мобільного додатку



Ви ще не проходили жодних тестів

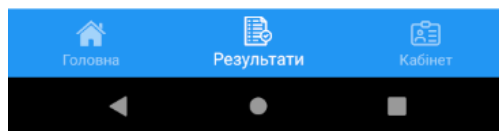


Рисунок 3.28 – Сторінка з результатами тестувань

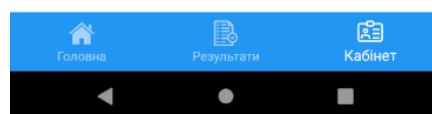
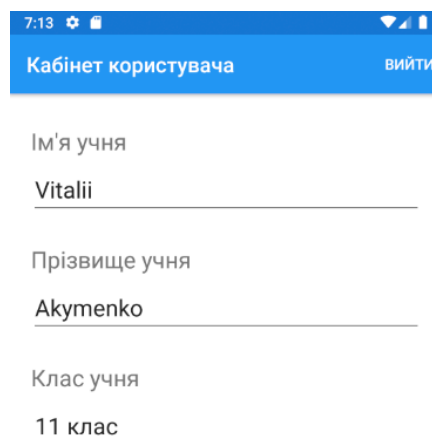


Рисунок 3.29 – Сторінка з даними користувача

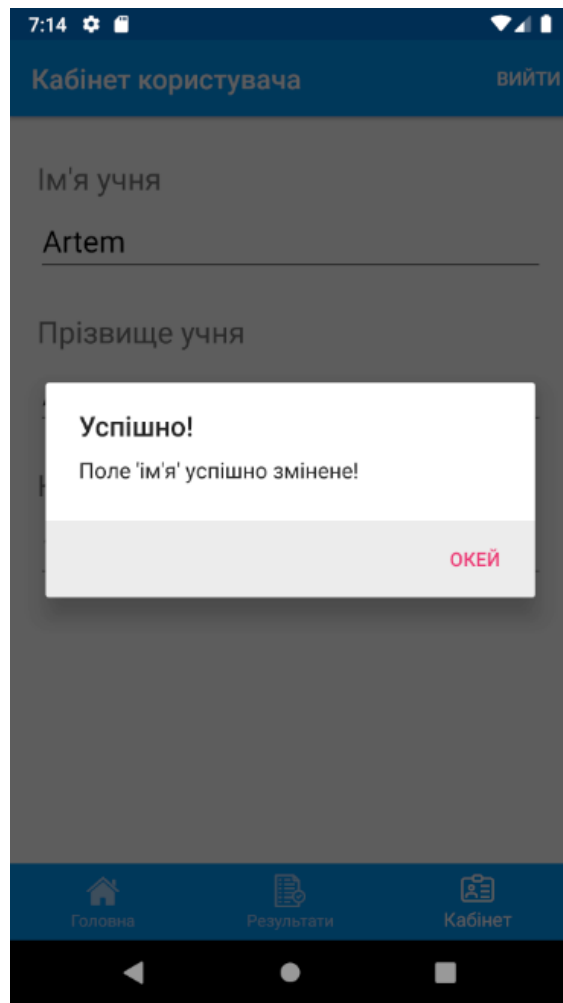


Рисунок 3.30 – Результат редагування даних користувача

Після знаходження тесту за унікальним номером користувача буде переадресовано на сторінку з властивостями тесту, де він зможе розпочати тест натиснувши відповідну кнопку внизу екрану (рис. 3.31).

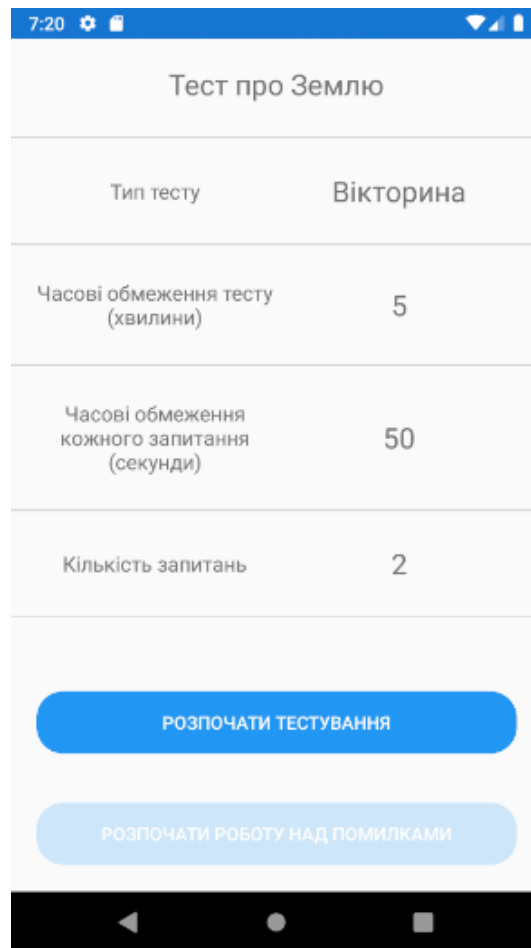


Рисунок 3.31 – Сторінка з властивостями тесту перед початком тестування

В залежності від встановлених обмежень тесту, користувачу надається час для надання відповіді на запитання та на тест в цілому. Якщо учень не встиг надати відповідь на запитання у встановлений час – елементи вибору відповідей стають неактивними. Якщо не встиг завершити тест – він завершується встановленням пустих відповідей незавершеним запитанням.

Сторінка запитання під час проходження тесту зображена на рисунку 3.32.

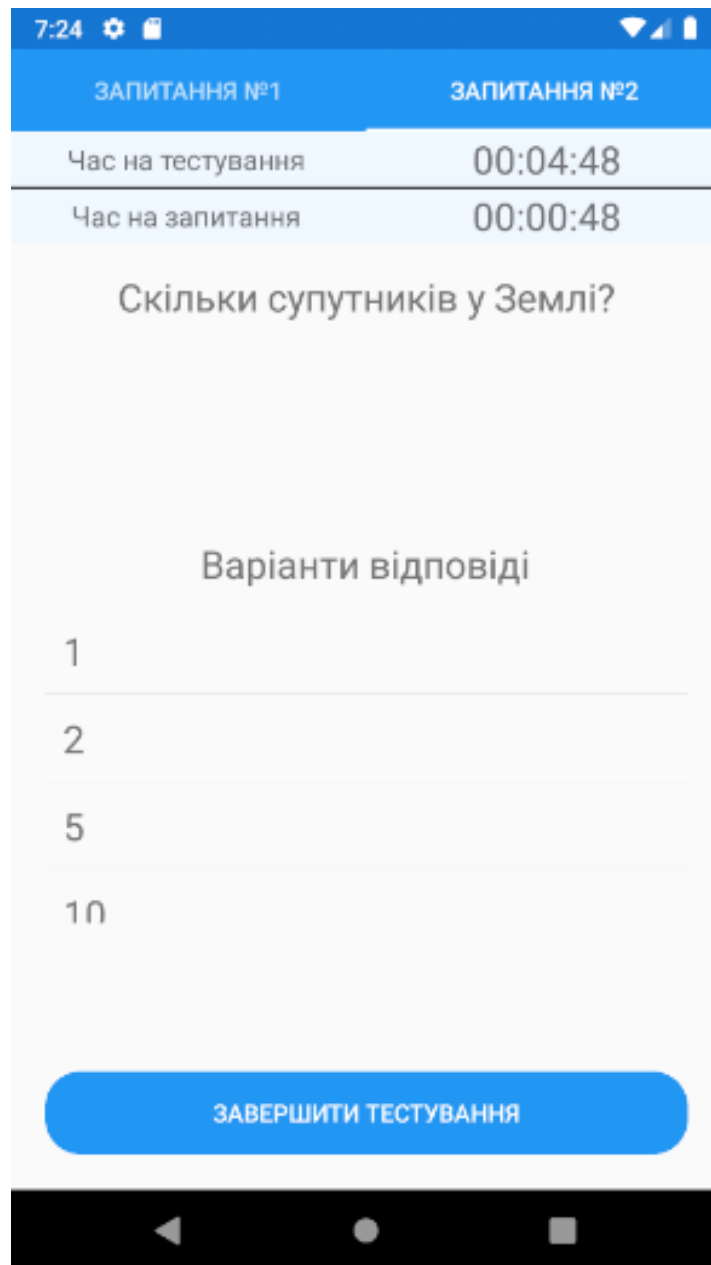


Рисунок 3.32 – Сторінка запитання під час тестування

Після завершення тестування користувачу відображується сторінка з результатами тестування: кількість правильних відповідей та загальна оцінка, яку він отримав (рис. 3.33).

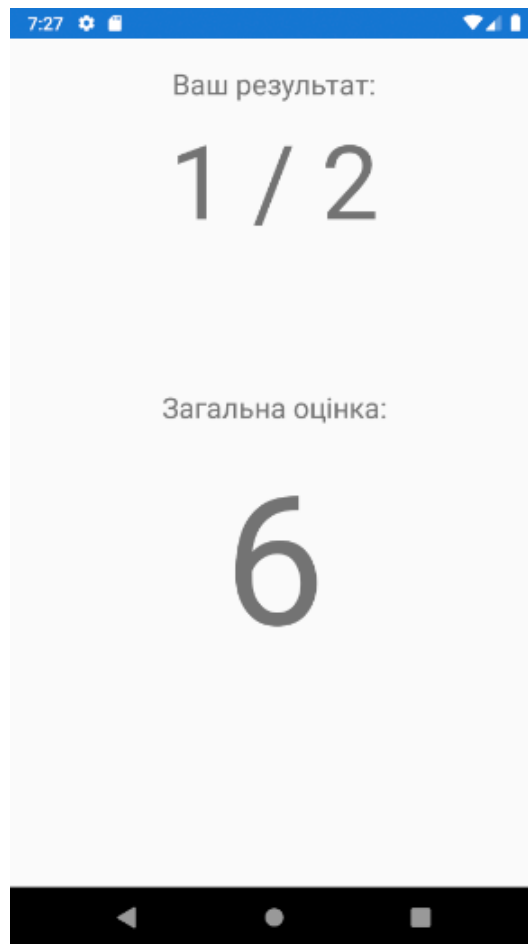


Рисунок 3.33 – Сторінка з результатами тестування

Якщо вчитель дозволив учням проводити роботу над помилками – після першого проходження тесту учень буде мати можливість ще раз пройти тест зі зниженням максимальної оцінки за кожен попередньо надану неправильну відповідь. Під час роботи над помилками тест містить лише ті запитання, на які учень надав неправильну відповідь, або зовсім не надав.

Результати повторного проходження тестування зі зниженням оцінки за неправильні запитання зображені на рисунку 3.34.

Після проходження учнем тесту він може переглянути їх на сторінці «Результати» у вигляді таблиці, де зазначені всі результати його тестувань (рис. 3.35).

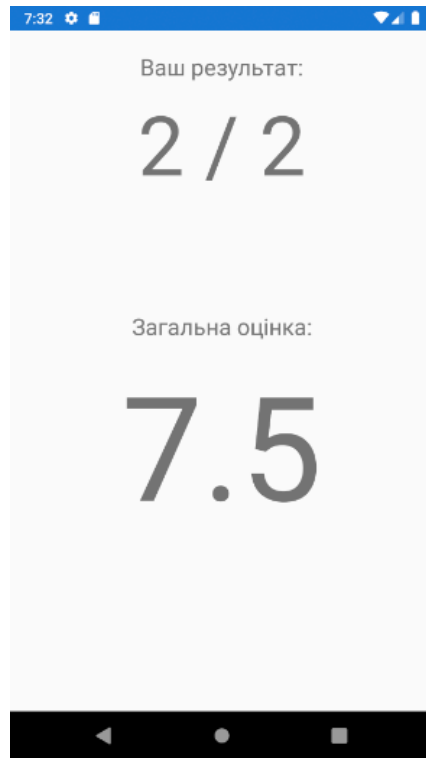


Рисунок 3.34 – Сторінка з результатами тестування після роботи над помилками

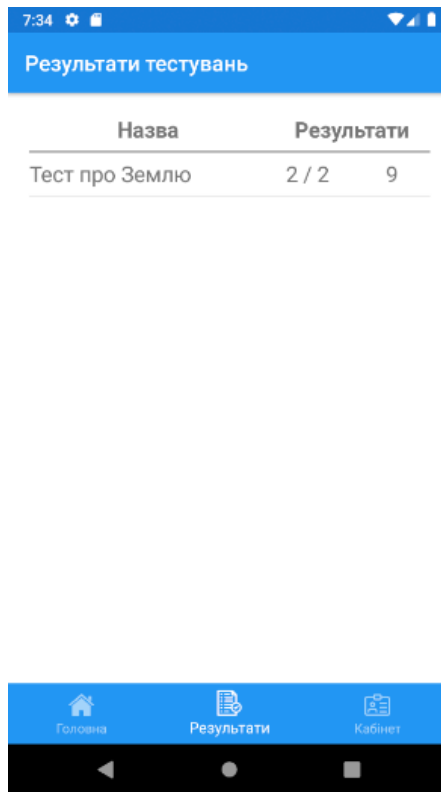


Рисунок 3.35 – Сторінка з результатами тестувань учня

ВИСНОВКИ

У рамках роботи над кваліфікаційною роботою бакалавра було виконано аналіз предметної області та визначено актуальність проблеми. В рамках етапу аналізу програмних комплексів для кожного розглянутого програмного комплексу було визначено переваги та недоліки, результати зведено в порівняльну таблицю, а також сформовано технічне завдання на виконання власної розробки програмного комплексу.

Наступним етапом було сформовано мету та завдання проєкту, визначено інструменти процесу розробки. Мета проєкту - розробка програмного комплексу, що складається з веб- та крос-платформного мобільного додатків, для організації процесу тестування учнів в школах України. Для розробки веб-додатку програмного комплексу було обрано IDE WebStorm та мови програмування HTML, CSS, JavaScript, PHP, для мобільного додатку - IDE VisualStudio та фреймворк Xamarin.Forms.

У рамках проєктування програмного комплексу було створено діаграми в нотації IDEF0 процесів роботи мобільного та веб-додатків, які наступним етапом були декомпозовані на підпроцеси. Для кожного з додатків розроблено діаграми варіантів використання, які наглядно в графічному форматі дозволили розмежувати та виділити необхідний та обов'язковий функціонал застосунків.

Розроблена реляційна модель бази даних була використана для створення фізичної моделі бази даних, яка в свою чергу забезпечує інформаційну підтримку робочих процесів мобільного та веб-додатків з метою збереження та подальшої обробки даних.

Результатом виконання роботи є програмний комплекс, що складається з крос-платформного мобільного додатку, а також веб-додатку, що в повній мірі відповідають вимогам та надають цільовим користувачам необхідний функціонал.

Веб-додаток був розміщений на хостинг-сервісі. Кожен з розроблених мобільних додатків був завантажений до відповідного його платформі магазину додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Акименко В.В., Ващенко С.М. Програмний комплекс забезпечення процесу тестування. // «Інформатика, математика, автоматика»: матеріали та програма науково-технічної конференції, м. Суми, 274 – 28 квітня 2023 р. – Суми: Сумський державний університет, 2023. – С. 159 – 160.
2. Cell Phone Usage in Schools 2023: Statistics And Insights [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://blog.gitnux.com/cell-phone-usage-statistics/>.
3. DIGITAL 2023: GLOBAL OVERVIEW REPORT [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://datareportal.com/reports/digital-2023-global-overview-report>.
4. ВШО – Всеукраїнська школа онлайн [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://lms.e-school.net.ua/>.
5. Дистанційка у смартфоні: Всеукраїнська школа онлайн запускає мобільний додаток [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://mon.gov.ua/ua/news/distancijka-u-smartfoni-vseukrayinska-shkola-onlajn-zapuskae-mobilnij-zastosunok>.
6. Створення тестів для онлайн-навчання: огляд платформ. – 2021. – Режим доступу до ресурсу: https://osvita.ua/vnz/high_school/80118/.
7. Kahoot! [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://kahoot.com>.
8. Quizizz [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://quizizz.com/join>.
9. Frontend vs Backend [Електронний ресурс]. – 2023. Режим доступу до ресурсу: <https://www.geeksforgeeks.org/frontend-vs-backend/>
10. Model-View-ViewModel (MVVM) [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.techtarget.com/whatis/definition/Model-View-ViewModel>.

11. What is AJAX and How Does It Work? [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.hostinger.com/tutorials/what-is-ajax>.
12. A Complete Guide to Flexbox [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>.
13. Xamarin Forms [Электронный ресурс]. – 2022. – Режим доступа до ресурсу: <https://devopedia.org/xamarin-forms>.
14. XAML overview [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-7.0>

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Програмний комплекс забезпечення процесу тестування»

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ

1.1 Призначення програмного комплексу

Програмний комплекс призначений для створення шкільними вчителями тестів з їх наступним проходженням учнями в школі під час уроку або дистанційно за допомогою мобільного додатку.

1.2 Мета створення програмного комплексу

Метою створення програмного комплексу є автоматизація однотипних, звичних для тестувань в школах, процесів, таких як безпосереднє проведення тестувань учнями, перевірка результатів, а також формування звітів як для учнів, так і для вчителів.

1.3 Цільова аудиторія

До цільової аудиторії належать учні та вчителі шкіл, що знаходяться в будь-яких областях України

2 ВИМОГИ ДО ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Вимоги до програмного комплексу в цілому

2.1.1 Вимоги до структури й функціонування програмного комплексу

Програмний комплекс має складатися з веб- та мобільного застосунків. Кожен з компонентів має представляти собою автономний додаток зі спільною базою даних, доступ до яких надається окремо в залежності від типу користувачів.

2.1.2 Вимоги до персоналу

До необхідних навичок персоналу відносяться досвід в користуванні персональним комп'ютером, мобільним пристроєм, а також веб-браузером. Особливі технічні вимоги, що необхідні для підтримки роботи комплексу, відсутні.

2.1.2 Вимоги до збереження інформації

Інформація, необхідна для роботи комплексу, має зберігатися у базі даних, що повинна бути реалізована засобами СУБД.

2.1.3 Вимоги до розмежування доступу

Розроблюваний комплекс має бути доступним в загальній мережі Інтернет. Функціонал, що надається користувачу, залежить від компоненту комплексу, який використовується.

2.2 Вимоги до веб-додатку комплексу в цілому

2.2.1 Вимоги до структури й функціонування веб-додатку

Веб-додаток повинен бути загальнодоступним в мережі Інтернет. Доменне ім'я, що повинно використовуватись для веб-додатку – Testro.com.ua

Кожен з компонентів веб-додатку має відповідати чітко визначеному функціоналу із дотриманням взаємозалежності один з одним.

2.2.2 Вимоги до персоналу

До необхідних навичок персоналу веб-додатку відносяться досвід в користуванні персональним комп'ютером, а також веб-браузером. Особливі технічні вимоги, що необхідні для підтримки роботи веб-додатку, відсутні.

2.2.3 Вимоги збереження інформації

Інформація, необхідна для роботи веб-додатку, має зберігатися у загальній для комплексу базі даних, реалізованій засобами СУБД.

2.2.4 Вимоги до розмежування доступу

У залежності від наданих прав, користувачі веб-додатку поділяються на адміністратора та вчителя.

Вчителям надається можливість доступу до всіх компонентів веб-додатку, створювати, редагувати та видаляти тести, а також переглядати та завантажувати результати проходження тестів у вигляді таблиць та діаграм.

Адміністратори мають доступ до всіх тестів, можуть редагувати та видаляти їх. Доступ до прав адміністратора надається автоматично після авторизації користувача-адміністратора.

2.3 Вимоги до мобільного додатку комплексу в цілому

2.3.1 Вимоги до структури й функціонування мобільного додатку

Мобільний додаток має кросплатформенним, розробленим під платформи Android та iOS.

Мобільні додатки повинні бути розміщеними у відповідним їм магазинам додатків в залежності від цільової платформи.

Кожен з компонентів мобільного додатку має дотримуватися принципу взаємозалежності та відповідати чітко визначеному функціоналу.

2.3.2 Вимоги до персоналу

До необхідних навичок персоналу відносяться досвід в користуванні мобільним телефоном. Особливі технічні вимоги, що необхідні для підтримки роботи мобільного додатку, відсутні.

2.3.3 Вимоги збереження інформації

Інформація, що використовується та створюється за допомогою мобільного додатку, повинна зберігатися у загальній для комплексу базі даних, реалізованій засобами СУБД.

2.3.4 Вимоги до розмежування доступу

Оскільки мобільний додаток необхідний для проходження тестів, єдиним видом його користувачів з повним доступом до функціоналу є учні.

Учням надається можливість доступу до всіх сторінок мобільного додатку, пошук та проходження тестів за унікальним номером, а також перегляд та збереження результатів тестувань у вигляді таблиць та діаграм.

2.4 Структура програмного комплексу

2.4.1 Загальна інформація про структуру програмного комплексу

Програмний комплекс складається з трьох основних компонентів:

- Веб-сайт – для створення та редагування тестів, перегляду результатів тестування вчителями;
- Кросплатформенний мобільний застосунок – для проходження тестів, а також перегляду результатів тестувань учнями;
- Сервер – для збереження даних у базі даних, необхідних для роботи компонентів комплексу

Структура програмного комплексу зображена на рисунку А.1.

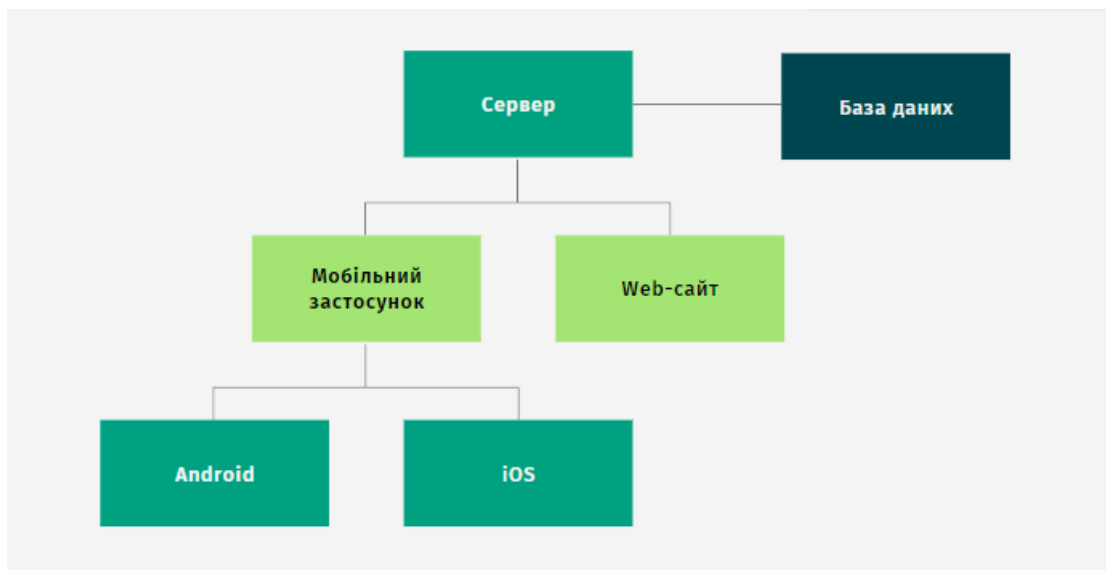


Рисунок А.1 – Структура програмного комплексу

2.4 Структура веб-додатку програмного комплексу

2.4.1 Загальна інформація про структуру веб-додатку

Веб-додаток складається з визначеного набору з 6 сторінок:

- Головна – основна інформація про комплекс, а також форма відправки повідомлень до персоналу веб-додатку;
- Вхід – сторінка з формою авторизації або реєстрації;
- Кабінет – при успішному вході в систему відображає список за доступних для користувача тестів за вказаними фільтрами, а також кнопку створення нового тесту;
 - Тест – містить властивості, з можливістю їх редагування, запитання відповідного тесту з можливістю їх додавання або переходу на сторінку редагування, а також кнопку переходу до статистики обраного тесту;
 - Запитання – сторінка властивостей обраного запитання для його редагування або видалення;
 - Результати – сторінка результатів обраного тесту, містить можливість збереження в файл у вигляді таблиць або діаграм.

2.4.2 Навігація

Для навігації в шапці має бути створено меню з двома елементами, що надають можливість переходу до головної сторінки та кабінету користувача. Меню має бути доступним на кожній сторінці для надання користувачу можливості в будь-який момент перейти до головних сторінок веб-додатку.

Навігація між залежними один від одних сторінок, таких як «Тест», «Запитання», «Результати», відбувається з попередніх їм сторінок для збереження цілісності відображення даних та переходу між ними.

2.4.3 Дизайн та структура додатку веб-додатку

Основою інтерфейсу та дизайну веб-додатку мають слугувати простота, зрозумілість та інтуїтивність, адже основними його користувачами мають бути викладачі різного рівня користування комп'ютером та веб-браузером в цілому.

Основними кольорами було обрано білий та відтінки синього, оскільки вони максимально прості для сприйняття і не відволікають від елементів інтерфейсу.

Макети сторінок веб-додатку зображено на рисунках А.2 та А.3.

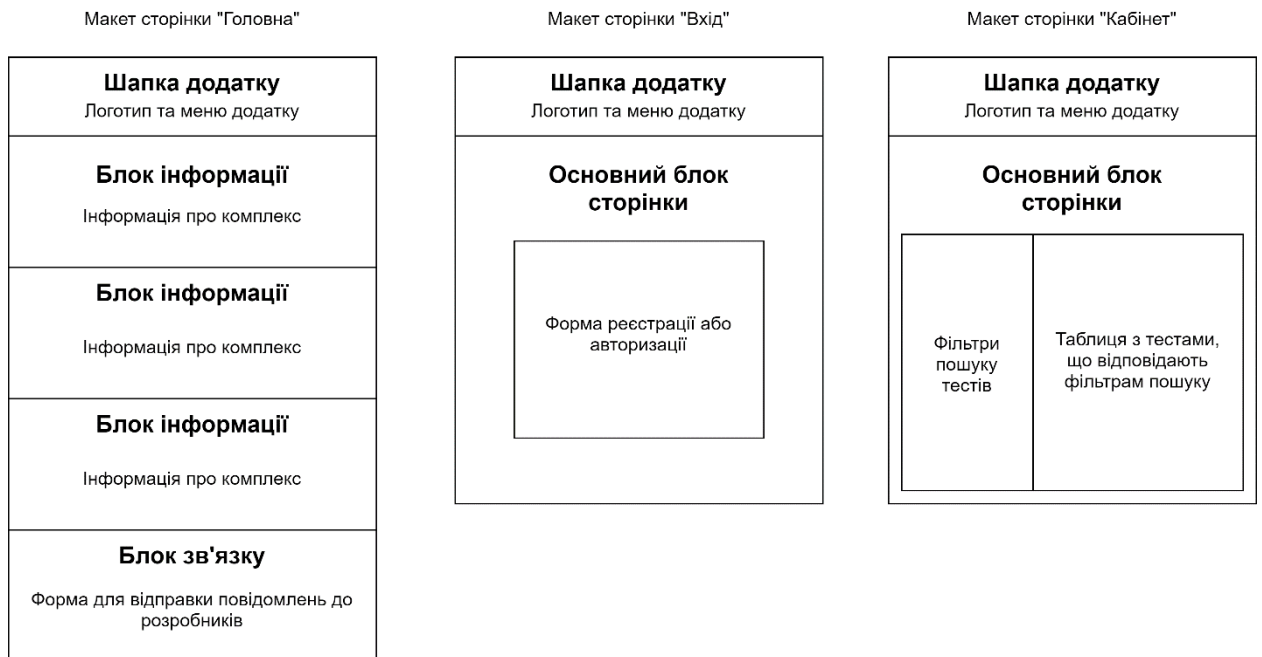


Рисунок А.2 – Макети сторінок «Головна», «Вхід» та «Кабінет» веб-додатку

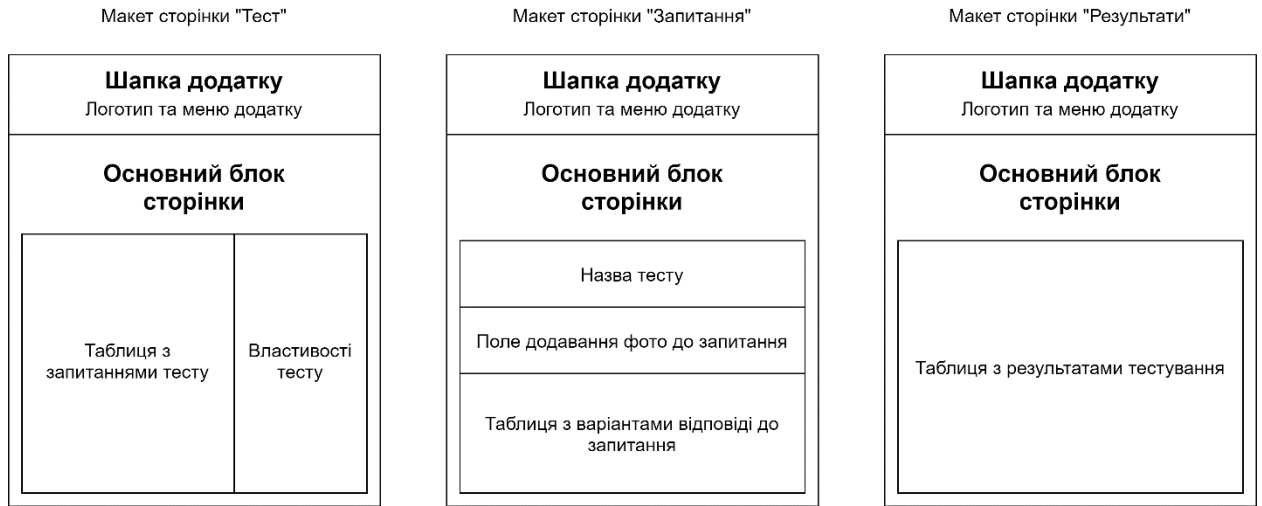


Рисунок А.3 – Макети сторінок «Тест», «Запитання» та «Результати» веб-додатку

2.4.4 Система навігації

Карта веб-додатку зображена на рисунку А.4.

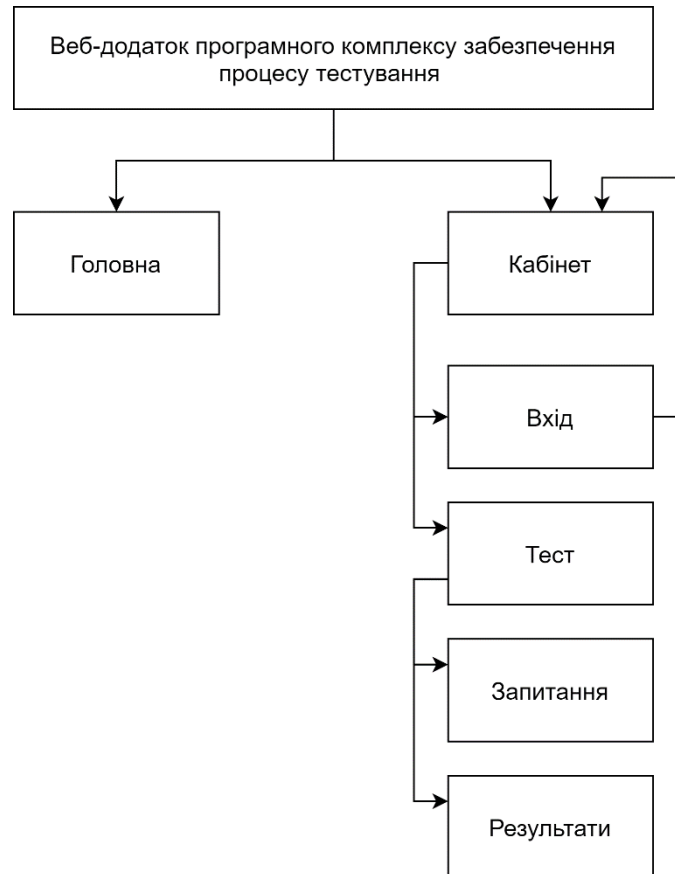


Рисунок А.4 – Карта веб-додатку програмного комплексу

2.5 Структура мобільного додатку програмного комплексу

2.5.1 Загальна інформація про структуру мобільного додатку

Веб-додаток складається з визначеного набору з сторінок:

- Авторизація – сторінка авторизації існуючого учня;
- Реєстрація – сторінка реєстрації нового учня;
- Головна – при успішному вході в систему відображає текстове поле для вводу унікального номеру тесту для проходження або його автоматичного введення за допомогою сканування QR-коду;
- Тест – містить інформацію про тест: кількість запитань, оцінку, часові обмеження, а також кнопку початку тестування;
- Запитання – сторінка для надання відповіді на запитання, містить текст запитання, фото в разі його наявності, а також варіанти відповіді;
- Результати тесту – сторінка з результатами щойно пройденого тесту, відображає кількість правильних відповідей, а також кнопку повторного проходження тесту зі зниженим оцінюванням;
- Кабінет – сторінка з даними користувача та кнопкою виходу з аккаунту користувача на пристрої;
- Результати – сторінка з результатами пройдених користувачем тестів.

2.5.2 Навігація

Для навігації внизу екрану додатку має бути створене меню для переходу між головною сторінкою, кабінетом, а також сторінкою результатів користувача. Меню має бути доступним на кожній з перелічених сторінок, для надання користувачу можливості в будь-який момент перейти до основних сторінок мобільного додатку.

Навігація між сторінками «Тест», «Запитання», «Результати тесту» відбувається поступово в процесі тестування, що зменшує інформаційне навантаження на учня під час проходження тесту.

2.5.3 Дизайн та структура додатку мобільного додатку

Мобільний додаток має бути орієнтований на досвідченого користувача смартфона, отже його інтерфейс та дизайн мають бути сучасними, але одночасно простими з метою зменшення інформаційного навантаження на учня та фокусування його уваги на проходженні тесту.

Макети сторінок мобільного додатку зображено на рисунках А.5-А7.



Рисунок А.5 – Макети сторінок «Авторизація», «Реєстрація» та «Головна» мобільного додатку

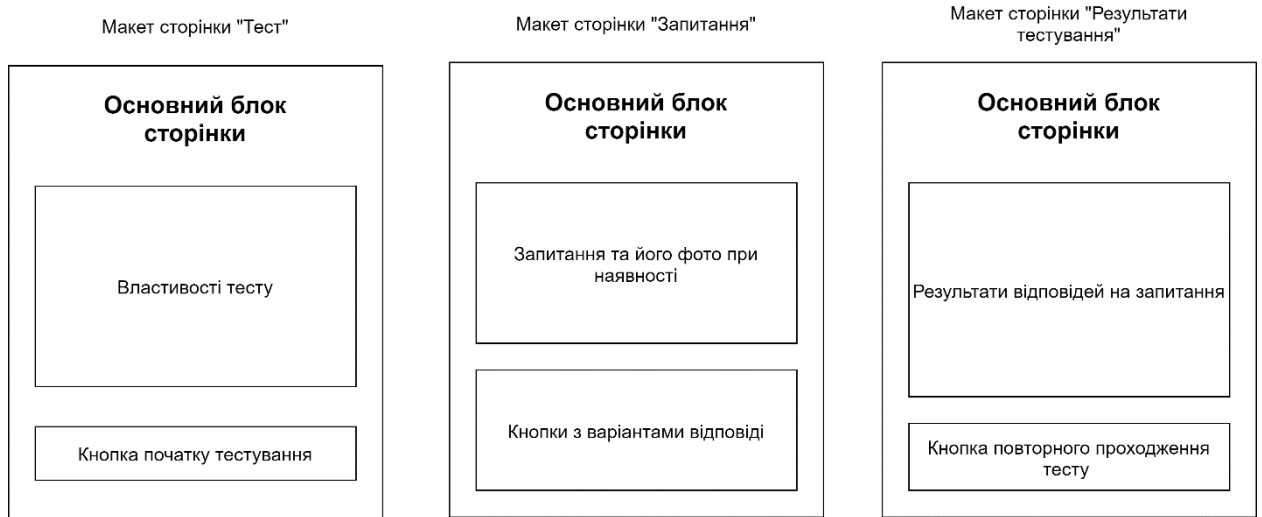


Рисунок А.6 – Макети сторінок «Тест», «Запитання» та «Результати тестування» мобільного додатку

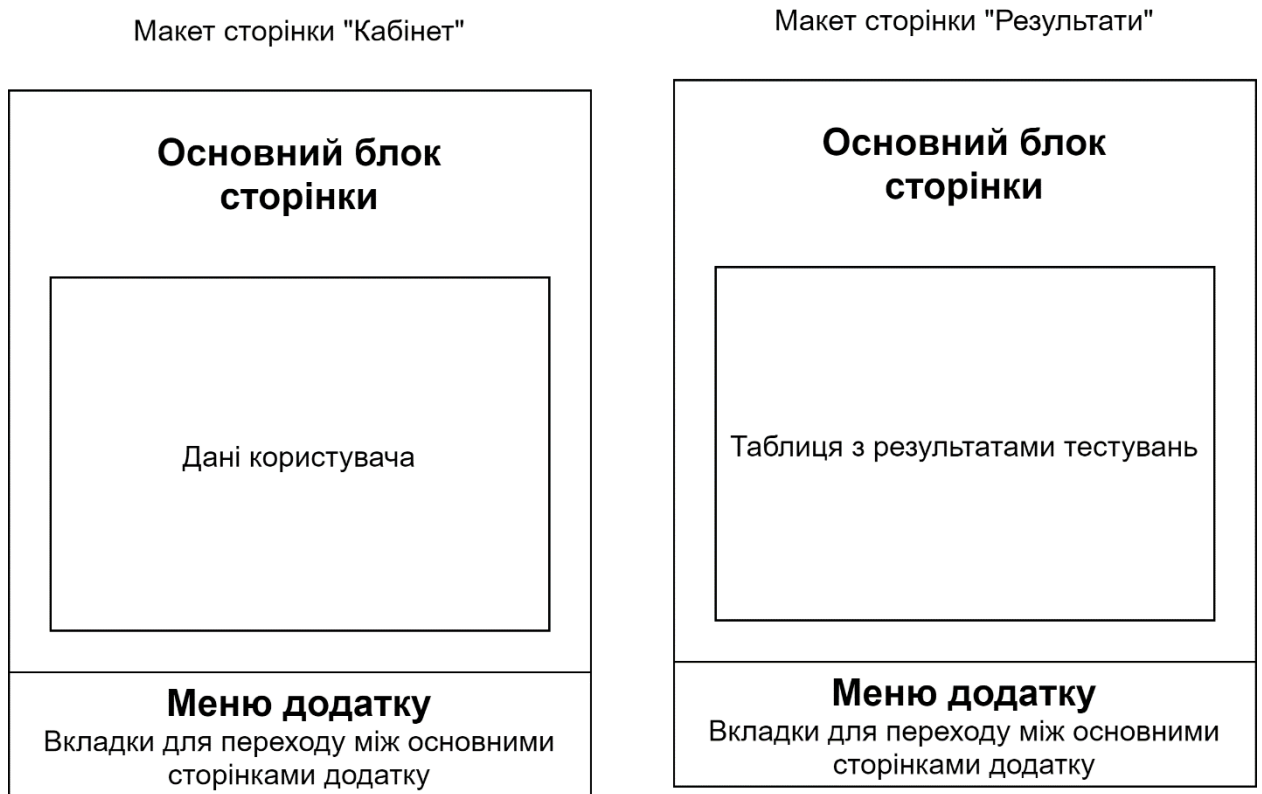


Рисунок А.7 – Макети сторінок «Кабінет» та «Результати» мобільного додатку

2.5.4 Система навігації

Карта мобільного додатку зображена на рисунку А.8.



Рисунок А.8 – Карта мобільного додатку програмного комплексу

2.6 Вимоги до функціонування веб-додатку

2.6.1 Потреби користувача

На основі поставлених вимог замовника було створено список потреб користувача, зазначений в таблиці А.1.

Таблиця А.1 – Потреби користувача веб-додатку

ID	Потреба користувача	Джерело
UN-01	Створення тестів різних типів	Клієнт
UN-02	Редагування часових обмежень, оцінки, типу тестів, а також можливості роботи над помилками та його оцінки для тестів	Клієнт
UN-03	Видалення тестів	Клієнт, Адміністратор
UN-04	Створення, редагування та видалення запитань тесту	Клієнт
UN-05	Перегляд створених тестів	Клієнт, Адміністратор
UN-06	Збір результатів проходження тесту	Клієнт
UN-07	Збереження результатів тестувань у вигляді таблиць та діаграм	Клієнт
UN-08	Можливість зворотнього зв'язку з персоналом	Клієнт

2.6.2 Функціональні вимоги

Враховуючи вищезазначені потреби користувача, було визначено наступний перелік функціональних вимог:

- Реєстрація та авторизація користувачів-вчителів;

- Можливість зворотного зв'язку за допомогою форми;
- Створення тестів типів «Вікторина» та «Хто швидше» (табл. А.2);
- Редагування властивостей тесту, а саме часових обмежень, оцінки, типу тесту;
 - Додавання можливості роботи над помилками тесту зі встановленням зниження максимальної оцінки за кожну повторну відповідь;
 - Відображення таблиці з використанням фільтрів створених користувачем тестів;
 - Додавання, редагування та видалення запитань до тесту;
 - Редагування кількості варіантів відповіді на запитання;
 - Відображення статистики проходження тесту у вигляді таблиці;
 - Збереження статистики тестувань у вигляді таблиць та діаграм;
 - Адміністрування створених тестів, їх видалення.

Таблиця А.2 – Перелік доступних видів тестів для створення

Тип	Особливості
Вікторина	Користувач читає запитання та обирає на нього відповідь серед запропонованих варіантів (від 4 до 6)
Хто швидше	Користувачі синхронно отримують запитання та дають на нього відповідь серед запропонованих варіантів: хто відповів правильно і найшвидше – отримує найбільше балів рейтингу. По завершенню формується рейтингова таблиця з переможцями

2.6.3 Системні вимоги

Перелік вимог, визначений розробником, зазначено в таблиці А.3.

Таблиця А.3 – Функціональні вимоги веб-додатку

ІД	Системна вимога	Пріоритет	Опис
SR-01	Наявність модуля зворотного зв'язку	S	Надає вчителю можливість відправки зворотної форми
SR-02	База даних з тестами	M	Відповідає за наповнення списку тестів
SR-03	База даних з запитаннями	M	Відповідає за наповнення списку запитань тесту
SR-04	База даних з відповідями на запитання	M	Відповідає за наповнення варіантів відповідей
SR-05	База даних з властивостями тестів	M	Відповідає за присвоєння тестам відповідних властивостей
SR-06	База даних з результатами тестувань	M	Відповідає за наповнення таблиці результатів тестування
SR-07	База даних з результатами відповідей на запитання	M	Відповідає за наповнення таблиці результатів відповідей на запитання
SR-08	База даних вчителів	M	Надає можливість реєстрації та авторизації користувачів
SR-09	Панель адміністратора	C	Надає можливість адміністратору видаляти тести

Пояснення використаних в таблиці А.2 умовних позначень:

- Must have (M) – вимоги, що обов'язково повинні бути реалізовані в веб-додатку;

- Should have (S) – вимоги, що повинні бути реалізовані, але мають нижчий пріоритет реалізації;
- Could have (C) – вимоги, що можуть бути реалізовані, але не є обов'язковими для досягнення мети проекту.

2.7 Вимоги до функціонування мобільного додатку

2.7.1 Потреби користувача

Список потреб користувача, що було сформовано відповідно до вимог, зазначено в таблиці А.4.

Таблиця А.4 – Потреби користувача мобільного додатку

ІД	Потреба користувача	Джерело
UN-01	Проходження тестів	Клієнт
UN-02	Перегляд результатів тестування	Клієнт
UN-03	Перегляд результатів відповіді на запитання	Клієнт
UN-04	Робота над помилками тесту	Клієнт
UN-05	Перегляд статистики проходження тестів	Клієнт
UN-06	Збереження результатів тестувань у вигляді таблиць та діаграм	Клієнт

2.7.2 Функціональні вимоги

Враховуючи вищезазначені потреби користувача, було визначено наступний перелік функціональних вимог:

- Реєстрація та авторизація користувачів-учнів;
- Проходження тестів;
- Блокування переходу на інші додатки, спроб створити скріншот або вимкнути телефон;
- Відображення результатів тестування за запитаннями і в цілому за тест;
- Робота над помилками тесту;
- Перегляд статистики проходження тестів;

- Збереження статистики проходження тестів у вигляді таблиць та діаграм.

2.7.3 Функціональні вимоги

Перелік системних вимог, визначений розробником, зазначено в таблиці А.5.

Таблиця А.5 – Системні вимоги мобільного додатку

ІД	Системна вимога	Пріоритет	Опис
SR-01	База даних з тестами	М	Відповідає за доступ до тесту
SR-02	База даних з запитаннями	М	Відповідає за наповнення списку запитань тесту
SR-03	База даних з відповідями на запитання	М	Відповідає за наповнення статистики проходження тестів
SR-04	База даних з властивостями тестів	М	Відповідає на наповнення властивостей тесту
SR-05	База даних учнів	М	Відповідає за авторизацію та реєстрацію учнів

Пояснення використаних в таблиці А.5 умовних позначень:

- Must have (М) – вимоги, що обов'язково повинні бути реалізовані в веб-додатку.

2.8 Вимоги до видів забезпечення веб-додатку

2.8.1 Вимоги до інформаційного забезпечення

Веб-додаток має бути реалізований з використанням:

- HTML 5;
- CSS;
- JQuery 1.7.1;
- PHP 7.4.4;
- MySQL 8.0.

2.8.2 Вимоги до лінгвістичного забезпечення

Всі модулі веб-додатку мають бути реалізовані українською мовою.

2.8.3 Вимоги до програмного забезпечення

Клієнтська частина програмного забезпечення веб-додатку програмного комплексу має використовуватися в браузері Chrome 2 і вище, або Opera 9.5 і вище, або Firefox 3.5 і вище.

2.9 Вимоги до видів забезпечення мобільного додатку

2.9.1 Вимоги до інформаційного забезпечення

Веб-додаток має бути реалізований з використанням наступних мов програмування та фреймворків:

- С#;
- Xamarin.Forms.

2.9.2 Вимоги до лінгвістичного забезпечення

Всі модулі мобільного додатку мають бути реалізовані українською мовою.

2.9.3 Вимоги до програмного забезпечення

Клієнтська частина програмного забезпечення мобільного додатку розроблюваного програмного комплексу має використовуватися в мобільних пристроях Android 9.0 та вище або iOS 12 та вище.

3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ

Етапи робіт із реалізації програмного комплексу та його компонентів наведено в таблиці А.6.

Таблиця А.6 – Етапи розробки програмного комплексу та його компонентів

№	Склад і зміст роботи	Терміни розробки в робочих днях
1	Постановка мети та цілей розробки програмного комплексу	2 дні
2	Формулювання технічного завдання	4 дні
3	Створення прототипу веб-додатку	2 дні
4	Створення макетів дизайну веб-додатку	3 днів
5	Верстка	3 дні
6	Розробка модулів веб-додатку	3 дні
7	Тестування веб-додатку	2 дні
8	Створення прототипу мобільного додатку	2 дні
9	Створення макетів дизайну мобільного додатку	3 днів
10	Написання сторінок мобільного додатку	3 дні
11	Розробка модулів мобільного додатку	3 дні
12	Тестування мобільного додатку	2 дні
13	Завершення робіт	1 день
	Загальна тривалість робіт	28 днів

4 ВИМОГИ ДО СКЛАДУ Й ЗМІСТУ РОБІТ ІЗ ВВЕДЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ В ЕКСПЛУАТАЦІЮ

Для коректної роботи програмного комплексу та правильної взаємодії його компонентів між собою, мобільний додаток та веб-додаток мають бути введені в роботу одночасно.

Веб-додаток має бути розміщений в мережі Інтернет з використанням хостинг-сервісу на заздалегідь придбаному доменному імені “testro.com.ua”. Для коректності переносу веб-додатку на хостинг потрібно щоб останній відповідав вимогам, що були зазначені у розробленому технічному завданні.

Кожен з реалізованих мобільних додатків під платформи Android та iOS має бути розміщений у відповідному платформі магазині додатків.

Наповнення бази даних при переносі на хостинг має містити лише дані про користувачів адміністраторів для веб-додатку, а також незмінні таблиці з типами та властивостями тестів. Інші дані будуть додані користувачами в процесі роботи з компонентами програмного комплексу. Для коректності переносу бази даних на хостинг він обов’язково повинен відповідати сформованим в технічному завданні вимогам.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

Мета проекту: Розробити програмний комплекс, що складається з веб-та кросплатформенного мобільного додатків, для організації процесу тестування учнів в школах України. Проект буде реалізований вчасно, що підтверджується розробленим календарним планом проекту. Зазначена мета проекту була деталізована методом SMART, що зображено в таблиці Б.1.

Таблиця Б.1 – Специфікація мети проекту методом SMART

Критерій SMART	Ціль
Specific (конкретна)	Розробити програмний комплекс, що складається з веб-та кросплатформенного мобільного додатків, для організації процесу тестування учнів в школах України
Measurable (вимірювана)	Програмний комплекс повністю забезпечує учнів та вчителів функціоналом з організації процесу тестування в школах
Achievable (досяжна)	Ціль проекту є досяжною, адже розробник має досвід розробки веб-додатків мовами HTML, CSS, JS, PHP, так само як і мобільних додатків на фреймворкі Xamarin.Forms. Мета проекту відповідає його вимогам та потребам учнів та вчителів в школах.
Relevant (значуща)	Проект створить зручні умови роботи учням та вчителям в рамках створення, проходження, а також перевірки результатів тестування
Time-bound (обмежена в часі)	Програмний комплекс, як і всі його частини розроблюються з дотриманням визначених термінів на основі визначених календарних планів.

Планування змісту структури робіт. Робоча структура проекту (WBS) представляє собою ієрархічний опис робіт, що будуть виконані в рамках реалізації проекту. На WBS процеси проекту представляються в виді дерева з різними ступенями деталізації по кожному з процесів проекту. Розроблена робоча структура проекту зображена на рисунку Б.1.

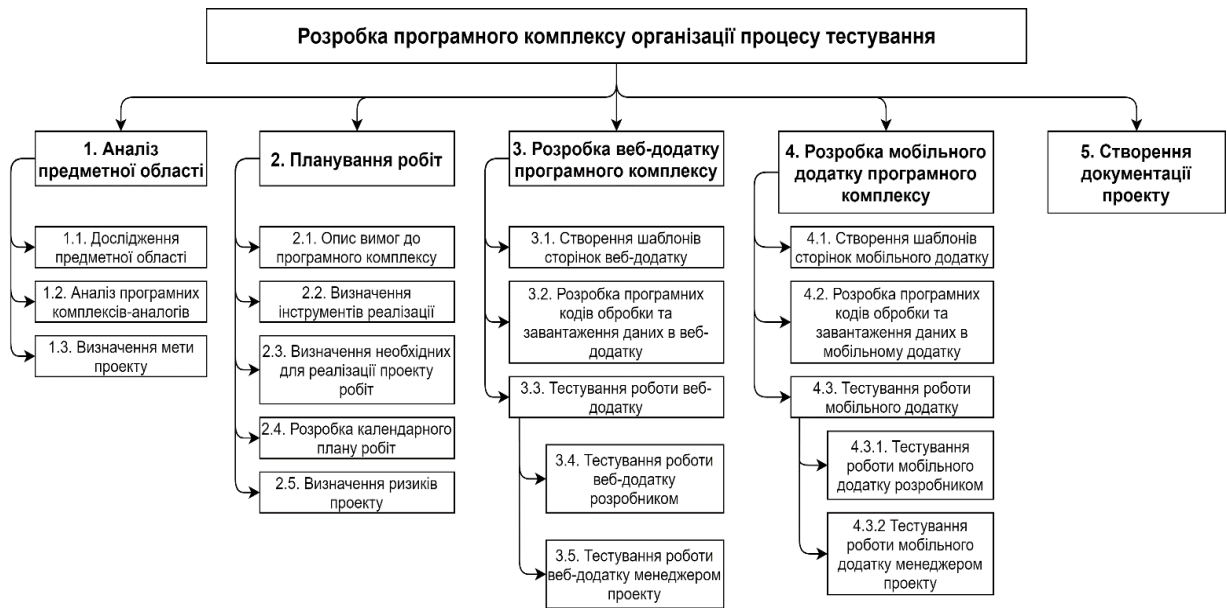


Рисунок Б.1 – Робоча структура проекту

Організаційна структура проекту (OBS). Наступним етапом робіт після створення робочої структури проекту є створення організаційної структури проекту (OBS). Вона є графічним відображенням учасників та керівників проекту, задіяних в його реалізації, як його ресурсів. В таблиці Б.2 зазначений список виконавців проекту.

Таблиця Б.2 – Виконавці проекту

Ініціали виконавця	Роль виконавця	Роль в рамках проекту
Акименко В.В	Розробник, тестувальник	Відповідає за реалізацією основного функціоналу, його тестуванням, поширенням програмного комплексу, а також плануванням робіт.
Ващенко С.М	Менеджер проекту	Головний керівник проекту, відповідає визначення мети проекту та перевірку готового програмного забезпечення на відповідність вимогам.

Після формування списку виконавців проекту було створено організаційну структуру проекту (рис. Б.2).

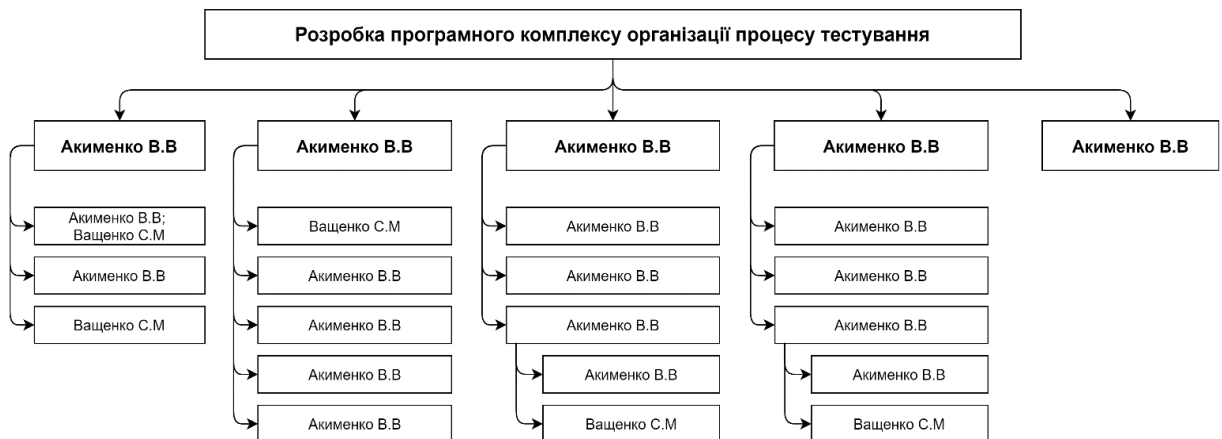


Рисунок Б.2 – Робоча структура проекту

Діаграма Ганта. Основними інструментами, що допомагають виконавцям проекту виконувати відповідні завдання є різноманітні графіки та діаграми. Одним з таких засобів планування та управління проектами є діаграма Ганта. Вона представляє кожен з видів робіт у вигляді відрізків, що розташовані у хронологічній послідовності із зазначенням залежностей між

завданнями, що є основним допоміжним ресурсом під час відстеження дотримання плану виконавцями.

В процесі виконання робіт кожне статус виконання відповідних завдань фіксуються у відсотках для наочності. Діаграма Ганта з роботами по проекту та відповідними їм ресурсами зображена на рисунку Б.3.

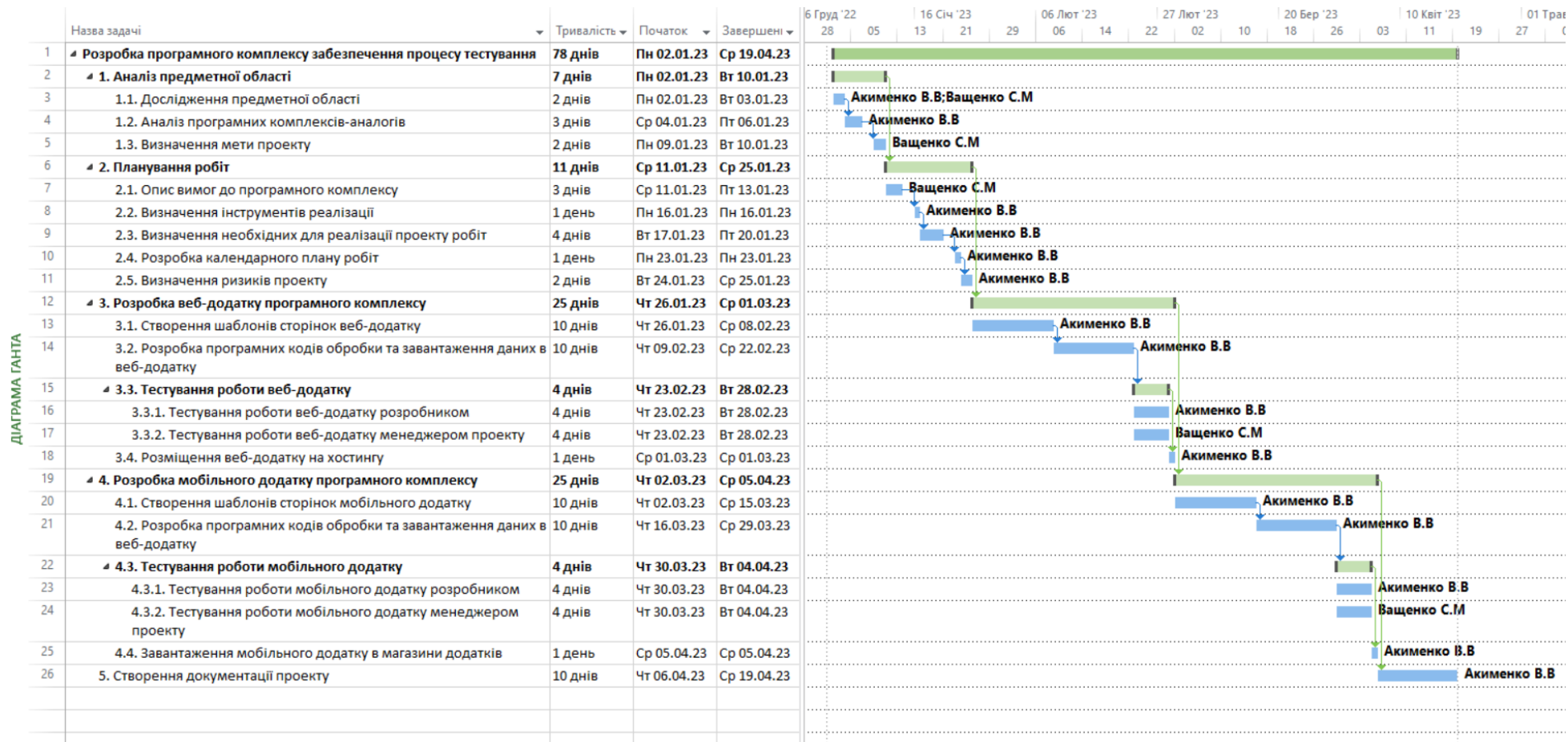


Рисунок Б.3 – Діаграма Ганта з роботами проекту

Аналіз ризиків. Ідентифікація ризиків проекту дозволяє визначити та оцінити перелік конкретних проблем, що можуть виникнути в ході роботи над проектом, і, в результаті, вплинути на кінцевий продукт та його якість.

Оскільки даний процес вимагає комплексності, до нього мають бути залучені всі виконавці та учасники проекту протягом всього процесу його реалізації.

Ймовірність виникнення ризиків, так само як можливі об'єми збитків від ризиків, поділяються на п'ять категорій: мінімальна, низька, середня, висока, максимальна.

З використанням зазначених даних була проведена класифікація ризиків проекту (табл. Б.3).

Таблиця Б.3 – Класифікація ризиків проекту

Назва ризику	Ймовірність	Питома вага втрат
Невідповідність ТЗ вимогам до проекту	Мінімальна	Висока
Недотримання календарного плану робіт	Низька	Середня
Некоректність роботи ПЗ	Середня	Висока
Некоректність в роботі сервера	Середня	Висока
Втрата даних	Середня	Максимальна
Хвороба розробника	Низька	Низька
Нестача апаратних ресурсів	Низька	Мінімальна

З використанням розробленої класифікації ризиків було визначено матрицю ризиків проекту (табл. Б.4).

Таблиця Б.4 – Матриця ризиків проекту

Ймовірність	Питома вага втрат				

На основі класифікації ризиків проекту та визначеної матриці ризиків проекту було сформовано план реагування на ризики (табл. Б.5).

Таблиця Б.5 – Плани реагування на ризики

Назва ризику	План реагування
Невідповідність вимогам до проекту ТЗ	Виконати необхідні правки для відповідності необхідним вимогам
Недотримання календарного плану робіт	Скоригувати календарний план шляхом оптимізації роботи над окремими етапами
Некоректність роботи ПЗ	Перезапустити програму, якщо результат незмінний - повернутися до попередньої збереженої робочої версії веб-додатку. В найшвидший термін виявити та усунути проблеми в ПЗ
Некоректність в роботі сервера	Перезапустити сервер, якщо результат незмінний – звернутися до постачальника послуг з хостингу для вирішення проблеми.
Втрата даних	Регулярно створювати резервні копії даних
Хвороба розробника	Скоригувати календарний план робіт шляхом оптимізації роботи над окремими етапами
Нестача апаратних ресурсів	Підвищити план хостингу з більшим об'ємом ресурсів

ДОДАТОК В

Лістинг коду сторінок веб-додатку

Файл: scripts/ajax-create-test.js

```
$(document).ready(function(){

function createTest(_data) {
  var test_id = -1;
  $.ajax({
    type: "POST",
    url: 'utils/create-test.php',
    data: _data,
    success: function(data) {
      test_id = data;
    }
  }).done(function(){

    console.log(test_id);
    window.location.replace("test-page.php?testId=" + test_id);
  });
}

$('body').on("click", "#add-test-submit", function(e){
  e.preventDefault();
  createTest({});
});
});
```

Файл: scripts/ajax-delete-data.js

```
$(document).ready(function() {
function deleteData(object, name, _tableName, _condition, afterSuccess) {
  Swal.fire({
    title: 'P'Pë PIPiPµPIPSµPSC-?',
    text: name + " P±CfPrPµ PSPµPjPsP¶P»PëPIPs PIC-PrPSPsPIPëC,Pë!",
    icon: 'warning',
    showCancelButton: true,
    confirmButtonColor: '#3085d6',
    cancelButtonColor: '#d33',
    confirmButtonText: 'PÿP°Pє, PIPëPrP°P»C-C,CH PIP¶Pµ P№PsPiPs!',
    cancelButtonText: 'P'C-PrPjC-PSPëC,Pë'
  }).then((result) => {
    if (result.isConfirmed){
      $.ajax({
        type: "POST",
```



```

        url: 'utils/delete-row.php',
        data: {
            tableName: _tableName,
            condition: _condition
        },
        success: afterSuccess
    });
}
});
}

$(".delete-question").on("click", function(event){
    var _tableName = 'questions';
    var _condition = {
        name: 'question_id',
        value: $(this).attr('id')
    };

    var message = "P—P°PïPëC,P°PSPSCII";
    deleteData($(this), message, _tableName, _condition, function() {
        window.location.reload();
    });
});

$(".delete-test").on("click", function(event){
    var _tableName = 'tests';
    var _condition = {
        name: 'test_id',
        value: $(this).attr('id')
    };

    var message = "PÿPµCÍC,";
    deleteData($(this), message, _tableName, _condition, function(){
        window.location.replace("cabinet.php");
    });
});

$("body").on("click", ".delete-answer", function(e){
    e.preventDefault();

    var elementToDelete = "#answer-element-" + $(this).val();
    var questionId = $(this).closest(".question-data").attr('id').replace("answers-", "");

    $.ajax({
        type: "POST",
        url: 'utils/delete-row.php',
        data: {
            tableName: 'answers',
            condition: {
                name: 'answer_id',
                value: $(this).val()
            }
        }
    });
});

```

```

    },
    success: function(data) {
        console.log(questionId);
        var data = {
            'question': questionId
        };
        showAnswers(data);
    }
});
});
});

```

Файл: scripts/ajax-edit-answers.js

```

$(document).ready(function(){
    $('body').on("click", ".assign-answer", function(e){
        e.preventDefault();
        var data = {
            'answer': $(this).val(),
            'question': $("#question-answer-" + $(this).val()).val()
        };
        changeAnswers(data);
    });

    $('body').on("click", ".add-answer", function(e){
        e.preventDefault();
        var data = {
            'question_id': $(this).attr("id"),
            'question': $(this).attr("id")
        };
        addAnswer(data);
    });
});

function changeAnswers(_data){
    $.ajax({
        type: "POST",
        url: 'utils/ajax-change-correct-answer.php',
        data: _data
    }).done(function(){
        showAnswers(_data);
    });
}

function addAnswer(_data){
    $.ajax({
        type: "POST",
        url: 'utils/add-question-answer.php',
        data: _data
    }).done(function(){
        showAnswers(_data);
    });
}

```

```

}

function showAnswers(_data) {
  var resultHTML = "Error!";
  $.ajax({
    type: "POST",
    url: 'utils/ajax-print-answers.php',
    data: _data,
    success: function(data) {
      resultHTML = data;
      $('#answers-' + _data['question']).html(resultHTML).show()
    }
  }).done(function(){
    $('#answers-' + _data['question']).html(resultHTML).show();
  });
}

```

Файл: scripts/ajax-edit-text-input.js

```

$(document).ready(function() {
  function setColorDamping(object, color){
    object.css({'border-color': color + "0"});
    setTimeout(() => {
      object.css({'border-color': color + "255"});
    }, 500);
  }

  function colorDamping(object, color, originalColor){
    setColorDamping(object, color);
    setTimeout(() => {
      setColorDamping(object, originalColor);
    }, 2500);
  }

  function modify(object, _tableName, _result, _condition){
    $.ajax({
      type: "POST",
      url: 'utils/modify-row.php',
      data: {
        tableName: _tableName,
        result: _result,
        condition: _condition
      },
      success: function(data){
        colorDamping(object, 'rgba(133,224,133,', 'rgba(0,0,0,');
      }
    });
  }

  $(".editable-test-name").on("focusout", function(event){
    var _tableName = 'tests';
    var _result = {

```

```

        name: 'test_name',
        value: $(this).val()
    };
    var _condition = {
        name: 'test_id',
        value: $(this).attr('id')
    };

    modify($(this), _tableName, _result, _condition);
});

$(".editable-test-name").on("keydown", function(event){
    if(event.key == "Enter") {
        $(this).blur();
    };
});

$(".editable-answer-name").on("focusout", function(event){
    var _tableName = 'answers';
    var _result = {
        name: 'answer_text',
        value: $(this).val()
    };
    var _condition = {
        name: 'answer_id',
        value: $(this).attr('id')
    };

    modify($(this), _tableName, _result, _condition);
});

$(".editable-answer-name").on("keydown", function(event){
    if(event.key == "Enter") {
        $(this).blur();
    };
});

$(".editable-question-name").on("focusout", function(event){
    var _tableName = 'questions';
    var _result = {
        name: 'question_name',
        value: $(this).val()
    };
    var _condition = {
        name: 'question_id',
        value: $(this).attr('id')
    };

    modify($(this), _tableName, _result, _condition);
});

$(".editable-question-name").on("keydown", function(event){

```

```

    if(event.key == "Enter") {
        $(this).blur();
    };
});

$(".close-module-window-button").on("click", function(event){
    window.location.reload();
});
});

```

Файл: scripts/ajax-get-pupil-results.js

```

$(document).ready(function(){
    function isEmpty(value) {
        return typeof value == 'string' && !value.trim() || typeof value == 'undefined' || value ===
        null;
    }

    function showTestResults(_data) {
        var resultHTML = "Error!";
        $.ajax({
            type: "POST",
            url: 'utils/load-test-results.php',
            data: _data,
            success: function(data) {
                resultHTML = data;

                if(isEmpty(resultHTML))
                    resultHTML = "<h3 class='first-item'>P–PsPrPSPsPiPs CтPμP·CfP»CHC,P°C,Cf
                    PSPμ P·P°PN°PrPμPSPs</h3>";

                $('#test-results-list').html(resultHTML).show();
            }
        }).done(function(){
            $('#test-results-list').html(resultHTML).show();
        });
    }

    $("body").on("click", "#find-pupil-results-submit", function(e){
        e.preventDefault();
        var data = {
            'test-id' : $('#test-id').val(),
            'pupil-to-find': $('#pupil-to-find').val()
        };
        showTestResults(data);
    });

    $("body").on("click", "#refresh-pupil-results-submit", function(e){
        e.preventDefault();
        var data = {
            'test-id' : $('#test-id').val(),
            'pupil-to-find': ""
        };
    });

```

```

        showTestResults(data);
        $('#pupil-to-find').val("");
    });
});

```

Файл: scripts/ajax-get-tests.js

```

$(document).ready(function(){
    function isEmpty(value) {
        return typeof value == 'string' && !value.trim() || typeof value == 'undefined' || value ===
        null;
    }

    function showTests(_data) {
        var resultHTML = "Error!";
        $.ajax({
            type: "POST",
            url: 'utils/load-tests.php',
            data: _data,
            success: function(data) {
                resultHTML = data;

                if(isEmpty(resultHTML))
                    resultHTML = "<h3>P-PsPrPSPsPiPs C,PμCΓC,Cf PSpμ
P·P°PNᵇPrPμPSPs</h3>";

                $('#tests_list').html(resultHTML).show();
            }
        }).done(function(){
            $('#tests_list').html(resultHTML).show();
        });
    }

    $('body').on("click", "#find-test-submit", function(e){
        e.preventDefault();
        var data = {
            'test-to-find': $('#test-to-find').val()
        };
        showTests(data);
    });

    $('body').on("click", "#refresh-test-submit", function(e){
        e.preventDefault();
        var data = {
            'test-to-find': ""
        };
        showTests(data);
        $('#test-to-find').val("");
    });
});

```

Файл: utils/print-answers.php

```

<?php
    $mysql = new mysqli("localhost", "root", "", "testro_db");
    $mysql->autocommit(true);

    $query = "SELECT * FROM `answers` WHERE `answer_id` IN (SELECT `answer_id`
FROM `question_answers` WHERE `question_id` = " . $question['question_id'] . ")";
    $result = $mysql->query($query);

    echo '<div class="u-expanded-width u-list u-list-1">
        <div class="container">';

    $answer_idx = 0;

    while ($answer_data = $result->fetch_assoc())
    {
        $answer_idx++;
        include("{$_SERVER['DOCUMENT_ROOT']}/answer-element-include.php");
    }

    echo '</div>
        </div>';

    if($answer_idx < 6)
        echo '<button class="add-answer u-btn-round u-radius-4 u-btn-3 u-text-hover-white u-
hover-palette-1-base item active" id=" . $question['question_id'] . "'>
            P”PsPrP°C,Pë PIP°CЂC–P°PSC, PIC–PrPiPsPIC–PrC–
        </button>';

    $mysql->close();
?>

```

Файл: utils/sign-up.php

```

<?php

function userWithLoginExists($mysql, $login) {
    $query = "SELECT * FROM `teacher_users` WHERE `teacher_login` = " . $login . " . ";";
    $result = $mysql->query($query);
    return $result->num_rows != 0;
}

function insertNewUserAndGetId($mysql, $login, $password, $fullname){
    $query = "INSERT INTO `teacher_data` (`teacher_full_name`) VALUES(" . $fullname .
    ") . ";";
    $mysql->query($query);
    $query = "INSERT INTO `teacher_users` (`teacher_login`, `teacher_password`,
`teacher_data_id`) VALUES(" . $login . ", " . $password . ", " . $mysql->insert_id . ") . ";";
    $mysql->query($query);
    return $mysql->insert_id;
}

```

```

session_start();

if(isset($_POST["sign-up-submit"])){
    $login = $_POST["sign-up-login"];
    $password = md5($_POST["sign-up-password"]);
    $teacherFullName = $_POST["sign-up-fullname"];

    $mysql = new mysqli("localhost", "root", "", "testro_db");

    if(userWithLoginExists($mysql, $login))
        header("location: ../cabinet-login.php?userExists=true");
    else{
        $_SESSION['user_id'] = insertNewUserAndGetId($mysql, $login, $password,
$teacherFullName);
        header("location: ../cabinet.php");
    }

    $mysql->close();
    exit();
}

?>

```

Файл: /pupil-result-element-include.php

```

<?php
include_once("utils/get-pupil-results-data.php");
$mysql = new mysqli("localhost", "root", "", "testro_db");
$mysql->autocommit(true);

$pupil_data = GetPupilData($mysql, $pupil_id);
$pupil_result = array(
    "total_mark" =>    GetPupilTotalMark($mysql, $test_id, $pupil_id),
    "correct_answers" => GetPupilCorrectAnswersCount($mysql, $test_id, $pupil_id),
);
?>

```

```

<div class="u-align-center half-width-item">
    <div class="container with-border">
        <h4 class="u-text first-item">
            <a class="u-product-title-link">
                <?php echo $pupil_data['pupil_name'] . " " . $pupil_data['pupil_surname']; ?>
            </a>
        </h4>
        <div class="container first-item">
            <h4 class="u-text first-item">
                <a style="margin: 10px">
                    <?php echo $pupil_result['correct_answers']; ?>
                </a>
            </h4>
            <h4 class="u-text first-item">
                <a style="margin: 10px">

```



```

        <?php echo $pupil_result['total_mark']; ?>
    </a>
</h4>
<a class="u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-text-hover-
white item" id="<?=$pupil_id;?>">
    <span class="u-file-icon u-icon">
        
    </span>
    P”PμC,P°P»CHΨPSC– CЪPμP·CίP»CHΨC,P°C,Pë CίC‡PSCΠ
</a>
</div>

</div>

<?php include("{$_SERVER['DOCUMENT_ROOT']}/test-result-pop-up.php"); ?>
</div>

```

Файл: /question-element-include.php

```

<?php
    $border_class = is_null($question['correct_answer_id']) ? "u-border-3 u-border-palette-2" : "u-
border-2 u-border-grey-75";
?>
<div>
    <div class=" <?php echo $border_class;?> u-shape-rectangle u-list-item-1">
        <div>
            <ol class="u-align-center u-text u-text-2 container">
                <label class="u-valign-middle u-text-palette-1-base first-item">
                    <li value="<?=$list_item_number?>">
                        <?php echo $question['question_name'];?>
                    </li>
                </label>
                <div class="first-item container">
                    <a class="u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-text-
hover-white item u-button-link u-dialog-link" href="#question-<?=$question['question_id']?>">
                        <span class="u-file-icon u-icon">
                            
                        </span>
                        P PμPrP°PiCίPIP°C,Pë
                    </a>
                    <a class="delete-question u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-
btn-5 u-border-palette-2-base u-hover-palette-2-base u-text-hover-white item"
id="<?=$question['question_id']?>">
                        <span class="u-file-icon u-icon">
                            
                        </span>
                        P’PëPrP°P»PëC,Pë
                    </a>
                </div>
            </ol>

```

```

    </div>
  </div>
  <?php
    include("test-question-pop-up.php");
  ?>
</div>

```

Файл: /test-element-include.php

```

<div class="u-align-center" style="margin: 5px 0;">
  <div class="container with-border">
    <h4 class="u-text first-item">
      <a class="u-product-title-link" style="margin: 10px">
        <?php echo $test_data['test_name']; ?>
      </a>
    </h4>
    <div class="first-item container">
      <a class="u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-text-hover-
white item" href="/test-results.php?testId=<?=$test_data['test_id'];?>">
        <span class="u-file-icon u-icon">
          
        </span>
        P PμP·CfP»CЪC,P°C,Pë C,PμCfC,CfPIP°PSCЪ
      </a>
      <a class="u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-text-hover-
white item" href="/test-page.php?testId=<?=$test_data['test_id'];?>">
        <span class="u-file-icon u-icon">
          
        </span>
        P PμPrP°PiCfPIP°C,Pë C,PμCfC,
      </a>
      <a class="delete-test u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-
border-palette-2-base u-hover-palette-2-base u-text-hover-white item"
id="<?=$test_data['test_id'];?>">
        <span class="u-file-icon u-icon">
          
        </span>
        P°PëPrP°P»PëC,Pë C,PμCfC,
      </a>
    </div>
  </div>
</div>

```

Файл: /test-page.php

```

<?php
  session_start();
  function getTestInfo($mysql) {
    $query = "SELECT * FROM `tests` WHERE `test_id` = " . $_GET['testId'];
    $result = $mysql->query($query);
  }

```

```

    return $result->num_rows > 0 ? $result->fetch_assoc() : null;
}

function getTestData($mysql, $testDataId) {
    $query = "SELECT * FROM `tests_data` WHERE `test_data_id` = " . $testDataId;
    $result = $mysql->query($query);
    return $result->num_rows > 0 ? $result->fetch_assoc() : null;
}

function getTestQuestions($mysql) {
    $query = "CALL GetTestQuestions(" . $_GET['testId'] . ")";
    $result = $mysql->query($query);
    $questions = array();
    while ($question = $result->fetch_assoc()) {
        $questions[] = $question;
    }
    return $questions;
}

function getTestTypes($mysql){
    $query = "SELECT * FROM `test_types` WHERE `is_ready` = 1";
    mysqli_next_result($mysql);
    $result = $mysql->query($query);
    $testTypes = array();
    while ($type = $result->fetch_assoc()) {
        $testTypes[] = $type;
    }
    return $testTypes;
}

function getMarkSelectHTML($testInfo){
    echo '<select name="test-mark" onchange="this.form.submit()">';
    for($i = 5; $i <= 12; $i++)
    {
        $selectedText = $i == $testInfo['test_mark'] ? ' selected' : "";
        echo '<option value="' . $i . '" . $selectedText . '>' . $i . ' P±P°P»C–PI</option>';
    }
    echo '</select>';
}

function getTimeSelectHTML($testData){
    echo '<select name="time-constraint" onchange="this.form.submit()">';
    for($i = 5; $i <= 50; $i += 5)
    {
        $selectedText = $i == $testData['test_time_constraint'] ? ' selected' : "";
        echo '<option value="' . $i . '" . $selectedText . '>' . $i . ' C...PIPëP»PëPS</option>';
    }
    echo '</select>';
}

function getQuestionTimeSelectHTML($testData){
    echo '<select name="question-time-constraint" onchange="this.form.submit()">';

```

```

for($i = 5; $i <= 50; $i += 5)
{
    $selectedText = $i == $testData['test_question_time_constraint'] ? ' selected' : "";
    echo '<option value="" . $i . "" . $selectedText . '>' . $i . ' CÍPμPeCfPSPr</option>';
}
echo '</select>';
}

function getTypeSelectHTML($testTypes, $testData){
    echo '<select name="type-constraint" onchange="this.form.submit()">';
    foreach($testTypes as $type){
        $selectedText = $type['test_type_id'] == $testData['test_type_constraint_id'] ? ' selected' :
";
        echo '<option value="" . $type['test_type_id'] . "" . $selectedText . '>' .
$type['test_type_name'] . '</option>';
    }
    echo '</select>';
}

function getMistakesCorrectionSelectHTML($testData){
    echo '<select name="mistakes-correction" onchange="this.form.submit()">';
    for($i = 0; $i <= 1; $i++)
    {
        $selectedText = $i == $testData['test_mistakes_correction'] ? ' selected' : "";
        $optionText = $i == 0 ? "P’C–PrCÍCfC,PSCÍ" : "PκP°CÍPIPIPSP°";
        echo '<option value="" . $i . "" . $selectedText . '>' . $optionText . '</option>';
    }
    echo '</select>';
}

$mysql = new mysqli("localhost", "root", "", "testro_db");
$mysql->autocommit(true);

$testInfo = getTestInfo($mysql);
$testData = !empty($testInfo) ? getTestData($mysql, $testInfo['test_data_id']) : null;
$testQuestions = !empty($testInfo) ? getTestQuestions($mysql) : null;
$testTypes = getTestTypes($mysql);
$mysql->close();
?>

<!DOCTYPE html>
<html lang="ru" style="font-size: 18px;">
<head>
<title>
    PÿPμCÍC,
</title>
<?php include_once("utils/common-head.php"); ?>
<link href="/styles/test-page.css" rel="stylesheet" type="text/css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
<script class="u-script" type="text/javascript" src="scripts/ajax-edit-answers.js"
defer=""></script>

```

```

    <script class="u-script" type="text/javascript" src="scripts/ajax-edit-text-input.js"
defer=""></script>
    <script class="u-script" type="text/javascript" src="scripts/ajax-delete-data.js"
defer=""></script>
</link>
</head>
<body class="u-body u-xl-mode" data-lang="ru">
    <?php include_once("header.html"); ?>
    <section class="u-section-1" id="sec-498f">
        <div class="">
            <div class="u-clearfix u-expanded-width u-gutter-10 u-layout-wrap u-layout-wrap-1">
                <div class="u-gutter-0 u-layout">
                    <div class="u-border-2 u-border-grey-75" >
                        <div class="u-size-30">
                            <div class="u-layout-col">
                                <div class="u-container-style u-layout-cell u-size-60 u-layout-cell-1">
                                    <div class="u-container-layout u-container-layout-1">
                                        <h3 class="u-align-center u-text">
                                            <input class="editable-test-name u-border-2 u-border-black u-
border-no-left u-border-no-right u-border-no-top u-input u-input-rectangle u-input-1"
name="question-name" placeholder="P'PIPμPrC-C,CH P·P°PīPēC,P°PSPSCI" required=""
type="text" value="<?=$testInfo['test_name']?>" id="<?=$testInfo['test_id']?>">
                                        </input>
                                        </h3>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            <div class="u-size-30">
                <div class="container">
                    <div class="first-item">
                        <div class="u-border-2 u-border-grey-75 u-container-layout u-container-
layout-3">
                            <div class="u-list u-list-1">
                                <div>
                                    <?php
                                        if(count($testQuestions) == 0){
                                            echo '<h4 class="u-align-center u-text">PīPμPNē C,PμCÍC,
PīCfCÍC,PēPNē</h4>';
                                            echo '<h4 class="u-align-center u-text">P'C-
PrCḂPμPrP°PīCfPNēC,Pμ PNēPsPīPs PSP°P·PICf C,P° PIP»P°CÍC,PēPIPSCÍC,C-</h4>';
                                            echo '<h4 class="u-align-center u-text">P"PsPrP°C,Pē
P·P°PīPēC,P°PSPSCI PjPsP¶PSPs P·P° PrPsPīPsPjPsPīPsCḂ PrPsPrP°C,PēPsPIPēC...
PēPSPsPīPsPē</h4>';
                                        }
                                        else{
                                            $list_item_number = 1;
                                            foreach ($testQuestions as $question)
                                            {
                                                include("question-element-include.php");
                                                $list_item_number++;
                                            }
                                        }
                                    </?php
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
</body>

```

```

    }
    ?>
  </div>
</div>
</div>
</div>
<div class="max-third-width-item">
  <div class="u-border-2 u-border-grey-75 u-list-item-2">
    <div class="u-grey-10 u-border-2">
      <div class="container">
        <label class="u-text u-valign-middle u-align-center first-item">
          PŷPëPï C,PμCÍC,Cř
        </label>
        <div class="select item">
          <form method="POST" action="/utils/modify-test-
type.php?testId=<?=$_GET['testId'];?>">
            <?php getTypeSelectHTML($testTypes, $testData); ?>
          </form>

        </div>
      </div>
    </div>
  </div>
</div>
<div class="u-grey-10 u-border-2">
  <div class="container">
    <label class="u-text u-valign-middle u-align-center first-item">
      PĥC†C–PSPeP° P·P° C,PμCÍC,
    </label>
    <div class="select item">
      <form method="POST" action="/utils/modify-test-
mark.php?testId=<?=$_GET['testId'];?>">
        <?php getMarkSelectHTML($testInfo); ?>
      </form>
    </div>
  </div>
</div>
<div class="u-grey-10 u-border-2">
  <div class="container">
    <label class="u-text u-valign-middle u-align-center first-item">
      P§P°CÍ PSP° C,PμCÍC,
    </label>
    <div class="select item">
      <form method="POST" action="/utils/modify-test-
time.php?testId=<?=$_GET['testId'];?>">
        <?php getTimeSelectHTML($testData); ?>
      </form>
    </div>
  </div>
</div>
<div class="u-grey-10 u-border-2">
  <div class="container">
    <label class="u-text u-valign-middle u-align-center first-item">
      P§P°CÍ PSP° P·P°PïPëC,P°PSPSCŪ

```

```

        </label>
        <div class="select item">
            <form method="POST" action="/utils/modify-test-question-
time.php?testId=<?=$_GET['testId'];?>">
                <?php getQuestionTimeSelectHTML($testData); ?>
            </form>
        </div>
    </div>
</div>
<div class="u-grey-10 u-border-2">
    <div class="container">
        <label class="u-text u-valign-middle u-align-center first-item">
            P PsP±PsC,P° PSP°Pr PiPsPjPëP»PeP°PjPë
        </label>
        <div class="select item">
            <form method="POST" action="/utils/modify-test-mistakes-
correction.php?testId=<?=$_GET['testId'];?>">
                <?php getMistakesCorrectionSelectHTML($testData); ?>
            </form>
        </div>
    </div>
</div>
<br>
<div class="container">
    <a class="u-border-2 u-btn u-btn-round u-radius-4 u-btn-3 u-text-
hover-white item" href="/utils/add-test-question.php?testId=<?=$_GET['testId'];?>">
        <span class="u-file-icon u-icon">
            
        </span>
        P”PsPrP°C,Pë PiPëC,P°PSPSCI
    </a>

    <a class="delete-test u-border-2 u-btn u-btn-round u-radius-4 u-btn-
3 u-border-palette-2-base u-hover-palette-2-base u-text-hover-white
id="<?=$_GET['testId'];?>">
        <span class="u-file-icon u-icon">
            
        </span>
        P°PëPrP°P»PëC,Pë C,PμCíC,
    </a>

    <a class="u-align-center u-border-2 u-btn u-btn-round u-radius-4 u-
btn-3 u-text-hover-white max-width-item href="/test-
results.php?testId=<?=$_GET['testId'];?>">
        <span class="u-file-icon u-icon">
            
        </span>
        P PμP·CíP»CHC,P°C,Pë C,PμCíC,CíPIP°PSCĤ
    </a>
</div>

```



```

        return BaseViewModel.GetFirstValue<string>(query, connection, "answer_text");
    }
}
}

```

Файл: /Models/BaseNotify.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Runtime.CompilerServices;

namespace Testro.Models
{
    public class BaseNotify : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;
        protected virtual void OnPropertyChanged(string propertyName)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
        }

        protected bool SetProperty<T>(ref T backingStore, T value,
            [CallerMemberName] string propertyName = "",
            Action onChanged = null)
        {
            if (EqualityComparer<T>.Default.Equals(backingStore, value))
                return false;

            backingStore = value;
            onChanged?.Invoke();
            OnPropertyChanged(propertyName);
            return true;
        }
    }
}

```

Файл: /Models/Question.cs

```

using MySqlConnection;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using Testro.ViewModels;

namespace Testro.Models
{
    public class Question : BaseNotify
    {
        public readonly static long UNKNOWN_ID = -1;
    }
}

```

```

public long QuestionId { get; set; } = UNKNOWN_ID;
public string QuestionName { get; set; } = string.Empty;
public long CorrectAnswerId { get; set; } = UNKNOWN_ID;
public long QuestionResultId { get; set; } = UNKNOWN_ID;
public ObservableCollection<Answer> Answers { get; set; }

public static Question CreateQuestion(long questionId)
{
    Question question = new Question();

    if (questionId != UNKNOWN_ID)
    {
        question.QuestionId = questionId;
        question.QuestionName
BaseViewModel.GetDataBaseRequestResult(question.GetQuestionName);
        question.CorrectAnswerId
BaseViewModel.GetDataBaseRequestResult(question.GetCorrectAnswerId)
UNKNOWN_ID;
        question.Answers = question.GetAnswers();
    }

    return question;
}

protected string GetQuestionName(MySqlConnection connection)
{
    string query = "SELECT * FROM `questions` WHERE `question_id` = " + QuestionId +
    """;
    return BaseViewModel.GetFirstValue<string>(query, connection, "question_name");
}

protected long? GetCorrectAnswerId(MySqlConnection connection)
{
    string query = "SELECT * FROM `questions` WHERE `question_id` = " + QuestionId +
    """;
    return BaseViewModel.GetFirstValue<long>(query, connection, "correct_answer_id");
}

protected ObservableCollection<Answer> GetAnswers()
{
    List<long> answerIds
BaseViewModel.GetDataBaseRequestResult(GetQuestionAnswerIds);
    ObservableCollection<Answer> answers = new ObservableCollection<Answer>();

    answerIds.ForEach(delegate (long answerId)
    {
        answers.Add(Answer.CreateAnswer(answerId));
    });

    return answers;
}

```

```

private List<long> GetQuestionAnswerIds(MySqlConnection connection)
{
    string query = "SELECT * FROM `question_answers` WHERE `question_id` = '" +
QuestionId + "'";
    return BaseViewModel.GetAllValues<long>(query, connection, "answer_id");
}
}
}

```

Файл: /Models/Test.cs

```

using MySqlConnection;
using System.Collections.Generic;
using Testro.ViewModels;

```

```

namespace Testro.Models

```

```

{
    public class TestData : BaseNotify
    {
        public readonly static long UNKNOWN_ID = -1;

        public long TestDataId { get; private set; } = UNKNOWN_ID;
        public long TestTimeConstraint { get; private set; } = 0;
        public long TestQuestionTimeConstraint { get; set; } = 0;
        public long TestTypeConstraintId { get; private set; } = UNKNOWN_ID;
        public string TestTypeName { get; private set; }

        public static TestData CreateTestData(long testDataId)
        {
            TestData testData = new TestData();

            if (testDataId != UNKNOWN_ID)
            {
                testData.TestDataId = testDataId;
                testData.TestTimeConstraint
                BaseViewModel.GetDataBaseRequestResult(testData.GetTimeConstraintValue) ?? 0;
                testData.TestQuestionTimeConstraint
                BaseViewModel.GetDataBaseRequestResult(testData.GetQuestionTimeConstraintValue) ?? 0;
                testData.TestTypeConstraintId
                BaseViewModel.GetDataBaseRequestResult(testData.GetTypeConstraintValue)
                UNKNOWN_ID;
                testData.TestTypeName
                BaseViewModel.GetDataBaseRequestResult(testData.GetTypeConstraintName);
            }

            return testData;
        }

        private long? GetTimeConstraintValue(MySqlConnection connection)
        {

```

```

        string query = "SELECT * FROM `tests_data` WHERE `test_data_id` = " + TestDataId +
        """,
        return BaseViewModel.GetFirstValue<long>(query, connection, "test_time_constraint");
    }

    private long? GetQuestionTimeConstraintValue(MySqlConnection connection)
    {
        string query = "SELECT * FROM `tests_data` WHERE `test_data_id` = " + TestDataId +
        """,
        return BaseViewModel.GetFirstValue<long>(query, connection,
        "test_question_time_constraint");
    }

    private long? GetTypeConstraintValue(MySqlConnection connection)
    {
        string query = "SELECT * FROM `tests_data` WHERE `test_data_id` = " + TestDataId +
        """,
        return BaseViewModel.GetFirstValue<long>(query, connection,
        "test_type_constraint_id");
    }

    private string GetTypeConstraintName(MySqlConnection connection)
    {
        string query = "SELECT * FROM `test_types` WHERE `test_type_id` = " +
        TestTypeConstraintId + """;
        return BaseViewModel.GetFirstValue<string>(query, connection, "test_type_name");
    }
}

public class Test : BaseNotify
{
    public readonly static long UNKNOWN_ID = -1;

    public long TestId { get; set; } = UNKNOWN_ID;
    public TestData TestData { get; set; } = new TestData();

    private string _testName = string.Empty;
    public string TestName
    {
        get => _testName;
        set
        {
            SetProperty(ref _testName, value);
        }
    }

    public int TestMark { get; set; }

    public List<Question> Questions { get; set; }

    public static Test CreateTest(long testId, bool withMistakesOnly = false)
    {

```

```

Test test = new Test();

if (testId != UNKNOWN_ID)
{
    test.TestId = testId;
    test.TestMark = BaseViewModel.GetDataBaseRequestResult(test.GetTestMark);
    test.TestName = BaseViewModel.GetDataBaseRequestResult(test.GetTestName);
    test.TestData = test.GetTestData();
    test.Questions = test.GetQuestions(withMistakesOnly);
}
return test;
}

private long? GetTestDataId(MySqlConnection connection)
{
    string query = "SELECT * FROM `tests` WHERE `test_id` = " + TestId + "";
    return BaseViewModel.GetFirstValue<long>(query, connection, "test_data_id");
}
protected string GetTestName(MySqlConnection connection)
{
    string query = "SELECT * FROM `tests` WHERE `test_id` = " + TestId + "";
    return BaseViewModel.GetFirstValue<string>(query, connection, "test_name");
}

protected int GetTestMark(MySqlConnection connection)
{
    string query = "SELECT * FROM `tests` WHERE `test_id` = " + TestId + "";
    return BaseViewModel.GetFirstValue<int>(query, connection, "test_mark");
}

protected TestData GetTestData()
{
    long testDataId = BaseViewModel.GetDataBaseRequestResult(GetTestDataId) ??
UNKNOWN_ID;
    return TestData.CreateTestData(testDataId);
}

protected virtual List<Question> GetQuestions(bool withMistakesOnly)
{
    List<long> questionIds = withMistakesOnly ?

BaseViewModel.GetDataBaseRequestResult(GetTestQuestionIdsWithMistakes) :
    BaseViewModel.GetDataBaseRequestResult(GetTestQuestionIds);
    List<Question> questions = new List<Question>();

    questionIds.ForEach(delegate (long questionId)
    {
        questions.Add(Question.CreateQuestion(questionId));
    });

    return questions;
}

```

```

protected List<long> GetTestQuestionIds(MySqlConnection connection)
{
    string query = "SELECT * FROM `test_questions` WHERE `test_id` = " + TestId + """;
    return BaseViewModel.GetAllValues<long>(query, connection, "question_id");
}

protected List<long> GetTestQuestionIdsWithMistakes(MySqlConnection connection)
{
    string query = "CALL GetTestQuestionsWithMistakes(" + TestId + ", " + User.UserId +
    """);";
    return BaseViewModel.GetAllValues<long>(query, connection, "question_id");
}
}
}

```

Файл: /CabinetViewModel.cs

```

using MySqlConnection;
using Testro.Models;
using Xamarin.Essentials;
using Xamarin.Forms;

namespace Testro.ViewModels
{
    public class CabinetViewModel : BaseViewModel
    {
        public Command SignOutCommand { get; }

        public CabinetViewModel()
        {
            Title = "РъР±С-РSPµС, РєРсСЪРёСГС, СфPIP°С‡Р°";
            SignOutCommand = new Command(OnSignOutClicked);
            UserName = new UserProperty("С-Рј'СІ", "pupil_name");
            UserSurname = new UserProperty("РіСЪС-Р·PIPёС%Рµ", "pupil_surname");
            UserForm = new UserProperty("РєР»Р°СІ", "pupil_form");
        }

        public void OnAppearing()
        {
            GetDataBaseRequestResult(UserName.SetValueFromDataBase);
            GetDataBaseRequestResult(UserSurname.SetValueFromDataBase);
            GetDataBaseRequestResult(UserForm.SetValueFromDataBase);
        }

        async void OnSignOutClicked()
        {
            if (await DisplayConfirmAlert("Р'РёС...С-Рr Р· Р°РєРєР°СфPSC,Сф", "Р'Рё
            РІРµРµPIPSPµPSC-, С%Рs С...РсС‡РµС, Рµ РІРёР№С,Рё?"))
            {
                Preferences.Remove("USER_ID");
            }
        }
    }
}

```

```

        await Shell.Current.GoToAsync($"://{nameof(Views.LoginPage)}", true);
    }
}

private bool? ChangeUserDataInDataBase(MySqlConnection connection, string fieldName,
string fieldValue)
{
    string query = "UPDATE `pupils_data` SET `" + fieldName + "` = " + fieldValue +
        " WHERE `pupil_data_id` = " + User.UserDataId + """;
    MySqlCommand command = new MySqlCommand(query, connection);
    MySqlDataReader reader = command.ExecuteReader();
    bool changedInDataBase = reader.RecordsAffected == 1;
    reader.Close();
    return changedInDataBase;
}

private UserProperty _userName;
public UserProperty UserName
{
    get => _userName;
    set
    {
        SetProperty(ref _userName, value);
    }
}

private UserProperty _userSurname;
public UserProperty UserSurname
{
    get => _userSurname;
    set
    {
        SetProperty(ref _userSurname, value);
    }
}

private UserProperty _userForm;
public UserProperty UserForm
{
    get => _userForm;
    set
    {
        SetProperty(ref _userForm, value);
    }
}

#if false
async void OnItemSelected(Item item)
{
    if (item == null)
        return;
}

```

```

        // This will push the ItemDetailPage onto the navigation stack
        await
        Shell.Current.GoToAsync($"{nameof(ItemDetailPage)}?{nameof(ItemDetailViewModel.ItemId)
    )}={item.Id}");
    }
#endif
}
}

```

Файл: /LoginViewModel.cs

```

using MySqlConnection;
using System.Threading.Tasks;
using Testro.Models;
using Testro.Views;
using Xamarin.Forms;

namespace Testro.ViewModels
{
    public class LoginPageViewModel : BaseViewModel
    {
        public Command LoginCommand { get; }
        public Command SignUpCommand { get; }

        public LoginPageViewModel()
        {
            LoginCommand = new Command(OnLoginClicked);
            SignUpCommand = new Command(OnSignUpClicked);
        }

        private async void OnLoginClicked(object obj)
        {
            if (GetDataBaseRequestResult(DoesUserExist) ?? false)
            {
                long userId = GetDataBaseRequestResult(GetUserId) ?? -1;

                if (userId != -1)
                {
                    User.UserId = userId;
                    User.UserId = GetDataBaseRequestResult(GetUserDataId) ?? -1;
                    await Shell.Current.GoToAsync($"//{nameof(Views.MainPage)}", true);
                    await Task.Delay(2000);
                    User.Login = string.Empty;
                    User.Password = string.Empty;
                }
                else
                {
                    DisplayErrorAlert("PíPµPíCßP°PIPëP»CHßPSPëP№ PíP°CßPsP»CHß,
CÍPíCßPsP±CfPNëC,Pµ P·PSPsPICf!");
                }
            }
            else

```



```

        DisplayErrorAlert("PљPsCЂPёCЃC,CfPIP°C‡ P· C,P°PePёPj P»PsPiC–PSPsPj PSPµ
C–CЃPSCfC"!");
    }

    private async void OnSignUpClicked(object obj)
    {
        await Shell.Current.GoToAsync($"{nameof(SignUpPage)}");
    }

    private long? GetUserId(MySqlConnection connection)
    {
        string query = "SELECT * FROM `pupil_users` WHERE `pupil_login` = '" + UserLogin
+
        "' AND `pupil_password` = '" + GetHash(UserPassword) + "'";
        return GetFirstValue<long>(query, connection, "pupil_id");
    }

    private long? GetUserDataId(MySqlConnection connection)
    {
        string query = "SELECT * FROM `pupil_users` WHERE `pupil_login` = '" + UserLogin
+
        "' AND `pupil_password` = '" + GetHash(UserPassword) + "'";
        return GetFirstValue<long>(query, connection, "pupil_data_id");
    }

    private bool? DoesUserExist(MySqlConnection connection)
    {
        string query = "SELECT * FROM `pupil_users` WHERE `pupil_login` = '" + UserLogin
+
        "'";
        return GetHasRows(query, connection);
    }

    private string _userLogin = string.Empty;
    public string UserLogin
    {
        get => _userLogin;
        set
        {
            SetProperty(ref _userLogin, value.Trim());
            LoginButtonEnabled = UserPassword != string.Empty && UserLogin != string.Empty;
        }
    }

    private string _userPassword = string.Empty;
    public string UserPassword
    {
        get => _userPassword;
        set
        {
            SetProperty(ref _userPassword, value.Trim());
            LoginButtonEnabled = UserPassword != string.Empty && UserLogin != string.Empty;
        }
    }

```

```

    }

    private string _errorText = string.Empty;
    public string ErrorText
    {
        get => _errorText;
        set
        {
            SetProperty(ref _errorText, value.Trim());
        }
    }

    private bool _loginButtonEnabled = false;
    public bool LoginButtonEnabled
    {
        get => _loginButtonEnabled;
        set => SetProperty(ref _loginButtonEnabled, value);
    }
}
}

```

Файл: /MainPageViewModel.cs

```

using MySqlConnection;
using Testro.Views;
using Xamarin.Forms;

namespace Testro.ViewModels
{
    public class MainPageViewModel : BaseViewModel
    {
        public MainPageViewModel()
        {
            Title = "P“PsP»PsPIPSP° CÍC,PsCЂC–PSPeP°";
            ScanQRCodeCommand = new Command(OnScanQRCodeClicked);
            GoToTestCommand = new Command(OnGoToTestClicked);
        }

        protected async void OnScanQRCodeClicked()
        {
            await PushModalAsync(new QRCodeScannerPage(this));
        }

        protected async void OnGoToTestClicked()
        {
            if (GetDataBaseRequestResult(CheckIfTestExists) ?? false)
                await PushModalAsync(new TestPage(long.Parse(ActiveTest)));
            else
                DisplayErrorAlert("PŷP°PePěPNž C,PμCÍC, PSPμ C–CÍPSCfC”!");
        }

        private bool? CheckIfTestExists(MySqlConnection connection)

```

```

    {
        string query = "SELECT * FROM `tests` WHERE `test_id` = " + ActiveTest + "";
        return GetHasRows(query, connection);
    }

    public Command ScanQRCodeCommand { get; set; }
    public Command GoToTestCommand { get; set; }

    public string ActiveTest
    {
        get => _activeTest;
        set
        {
            SetProperty(ref _activeTest, value);
            GoToTestButtonEnabled = value.Length > 0;
        }
    }
    private string _activeTest;

    public bool GoToTestButtonEnabled
    {
        get => _goToTestButtonEnabled;
        set => SetProperty(ref _goToTestButtonEnabled, value);
    }
    private bool _goToTestButtonEnabled = false;
}
}

```

Файл: /QuestionViewModel.cs

```

using System;
using System.Collections.ObjectModel;
using System.Linq;
using Testro.Models;
using Xamarin.Forms;

namespace Testro.ViewModels
{
    public class QuestionViewModel : BaseViewModel
    {
        static readonly TimeSpan SECOND = TimeSpan.FromSeconds(1);

        public ObservableCollection<Answer> Answers { get; set; }
        public Question Question { get; set; }
        public int QuestionIndex { get; set; }
        public string ContinueButtonText { get; private set; }
        public Command ContinueTestingCommand { get; private set; }
        public TestProcessViewModel TestProcessViewModel { get; set; }
        private int TestQuestionTime { get; set; }

        private bool _canSetAnswers = true;
        public bool CanSetAnswers

```

```

{
    get => _canSetAnswers;
    set
    {
        if (_canSetAnswers)
        {
            SetProperty(ref _canSetAnswers, value);

            if (!value)
                QuestionTimeColor = Color.Red;
        }
    }
}

private TimeSpan _timeToEndTest;
public TimeSpan TimeToEndTest
{
    get => _timeToEndTest;
    private set { SetProperty(ref _timeToEndTest, value); }
}

private TimeSpan _timeToEndQuestion;
public TimeSpan TimeToEndQuestion
{
    get => _timeToEndQuestion;
    private set { SetProperty(ref _timeToEndQuestion, value); }
}

private Color _questionTimeColor = (Color)Label.TextColorProperty.DefaultValue;
public Color QuestionTimeColor
{
    get => _questionTimeColor;
    set { SetProperty(ref _questionTimeColor, value); }
}

public QuestionViewModel(TestProcessViewModel testViewModel, int questionIndex)
{
    TestProcessViewModel = testViewModel;
    Question = TestProcessViewModel.Test.Questions.ElementAt(questionIndex);
    QuestionIndex = questionIndex;
    Answers = Question.Answers;
    Title = "P—P°PîPëC,P°PSPSCII B,-" + (QuestionIndex + 1).ToString();
    ContinueButtonText = GetContinueButtonText();
    ContinueTestingCommand = new Command(TestProcessViewModel.ContinueTesting);

    TestQuestionTime = (int)testViewModel.Test.TestData.TestQuestionTimeConstraint;
    TimeToEndQuestion = new TimeSpan(0, 0, TestQuestionTime);
}

public void OnPageAppearing()
{
    Device.StartTimer(SECOND, OnTimerTick);
}

```

```

private string GetContinueButtonText()
{
    long questionsLeft = TestProcessViewModel.Test.Questions.Count - QuestionIndex - 1;
    return questionsLeft > 0 ? "PμPμCḂPμPNḂC,Pë PrPs PSp°CÍC,CfPìPSPsPiPs
P·P°PìPëC,P°PSPSCII" : "P—P°PIPμCḂCëPëC,Pë C,PμCÍC,CfPIP°PSPSCII";
}

public void AddUserAnswer(object sender, EventArgs e)
{
    Answer answer = (sender as ListView).SelectedItem as Answer;
    int answerTime = TestQuestionTime - (int)TimeToEndQuestion.TotalSeconds;
    UserAnswer userAnswer = new UserAnswer(Question.QuestionId, answer.AnswerId,
answerTime);
    TestProcessViewModel.AddUserAnswer(QuestionIndex, userAnswer);
}

private bool OnTimerTick()
{
    if (DateTime.Now > TestProcessViewModel.TestEndTime)
    {
        TestProcessViewModel.EndTesting();
        return false;
    }

    if (TestQuestionTime > 0 && CanSetAnswers)
    {
        CanSetAnswers = TimeToEndQuestion > SECOND;
        TimeToEndQuestion = CanSetAnswers ? TimeToEndQuestion.Subtract(SECOND) :
TimeSpan.Zero;
    }

    TimeSpan timeLeft = TestProcessViewModel.TestEndTime - DateTime.Now;
    TimeToEndTest = new TimeSpan(timeLeft.Hours, timeLeft.Minutes, timeLeft.Seconds);
    return true;
}
}
}

```