

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Ісамутдінову Даніїлу Артуровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмний застосунок гри-вікторини _____

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17 червня 2025 р.

3. Вхідні дані до роботи _____ 1) мова програмування Dart;

_____ 2) фреймворк Flutter;

_____ 3) середовище розробки Android studio.

4. Перелік питань, що потрібно опрацювати у роботі _____ 1)Актуальність теми;

_____ 2) Аналіз існуючих рішень;

_____ 3) Формулювання вимог;

_____ 4) Вибір мови програмування та інструментів розробки;

_____ 5) Формування стратегії розробки;

_____ 6) Обробка типових помилок;

_____ 7) Тестування застосунку;

_____ 8) Формування висновків.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Слайд-презентація – 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
	Аналіз проблеми та огляд існуючих рішень	27.05.25-28.05.25	
	Вибір мови програмування для розробки	29.05.25-30.05.25	
	Вибір бібліотек та віджетів	31.05.25-01.06.25	
	Розробка макету застосунку	02.06.25-03.06.25	
	Написання коду	04.06.25-07.06.25	
	Виправлення помилок	08.06.25-09.06.25	
	Оформлення матеріалів кваліфікаційної роботи	10.06.25-11.06.25	
	Подання кваліфікаційної роботи керівникові та її попередній захист	12.06.25-13.06.25	
	Подання кваліфікаційної роботи на рецензування	14.06.25-16.06.25	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Наталія ЄРЬОМІНА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 81 с., 28 рис., 2 дод., 12 джерел.

МОБІЛЬНИЙ ЗАСТОСУНОК, ГРА-ВІКТОРИНА, DART, FLUTTER, ANDROID STUDIO, ІНТЕРФЕЙС, РЕДАКТОР ПИТАНЬ, ГЕЙМІФІКАЦІЯ, ОСВІТА, РОБОТА З ФАЙЛАМИ.

Метою кваліфікаційної роботи є розробка програмного застосунку у вигляді гри-вікторини, який дозволяє користувачу проходити тематичні тести у вигляді інтерактивної гри з підрахунком балів. Застосунок має інтуїтивний інтерфейс, реалізований із використанням технології Flutter [1], та підтримує два основні режими роботи: відкриття зовнішнього файлу з тестами або створення власного набору запитань безпосередньо у застосунку.

У ході виконання кваліфікаційної роботи було розроблено та протестовано повнофункціональний мобільний застосунок під назвою «Challenging Game». Програмний продукт реалізовано з використанням мови програмування Dart [2] і фреймворку Flutter, що забезпечує адаптивний дизайн та швидку взаємодію з користувачем. Особливу увагу було приділено зручності інтерфейсу, реалізації ігрової логіки, а також структурі збереження даних.

Ігрова логіка включає розбиття тесту на теми, в кожній з яких є набір питань з певною вартістю (балами). Користувач має можливість відповідати на запитання в інтерактивному форматі, отримуючи зворотний зв'язок про правильність відповіді, а також загальну статистику після завершення гри. Відповіді перевіряються з урахуванням нечутливості до регістру і пробілів.

ABSTRACT

Bachelor's thesis: 81 pages, 28 figures, 2 appendices, 12 sources.

MOBILE APPLICATION, QUIZ GAME, DART, FLUTTER, ANDROID STUDIO, INTERFACE, QUESTION EDITOR, GAMIFICATION, EDUCATION, WORKING WITH FILE.

The major goal of this thesis is to develop a software application in the form of a quiz game, which allows the user to take thematic tests in the form of an interactive game with scoring. The application has an intuitive interface, implemented using Flutter[1] technology, and supports two main operating modes: opening an external file with tests or creating your own set of questions directly in the application.

During the qualification work, a fully functional mobile application called «Challenging Game» was developed and tested. The software product was implemented using the Dart [2] programming language and the Flutter framework, which provides cross-platform, adaptive design, and fast interaction with the user. Particular attention was paid to the convenience of the interface, the implementation of game logic, as well as the structure of data storage.

The game logic includes dividing the test into topics, each of which has a set of questions with a certain value (points). The user has the opportunity to answer questions in an interactive format, receiving feedback on the correctness of the answer, as well as general statistics after the game is completed. Answers are checked taking into account case insensitivity and spaces.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ	10
1.1 Актуальність теми.....	10
1.2 Аналіз існуючих аналогів.....	11
1.2.1 Ігрова навчальна платформа «Kahoot!»	11
1.2.2 Навчальна платформа «Quizizz».....	13
1.2.3 Навчальна платформа «Mentimeter»	15
1.2.4 Навчальний застосунок «Socrative»	16
1.2.5 Програмне забезпечення для опитування «Google Forms»	18
1.2.6 Гра-вікторина «Party Quiz»	20
1.2.7 Навчальна платформа «Quizlet»	21
1.3 Аналіз розробленого застосунка	23
2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ ТА ФОРМУВАННЯ СТРАТЕГІЇ РОЗРОБКИ.....	26
2.1 Мова програмування Dart.....	26
2.2 Фреймворк Flutter.....	26
2.3 Середовище розробки Android Studio	27
2.4 Сторонні бібліотеки	27
2.5 Вибір технологій та інструментів.....	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	29
3.1 Структура проекту	29
3.2 Опис файлів проекту.....	29
3.2.1 main.dart.....	29
3.2.2 pick_file.dart	30
3.2.3 questions.dart	32

3.2.4 question_editor_screen.dart	34
3.3 Тестування та результати	37
3.3.1 Методи тестування.....	37
3.3.2 Виявлені помилки та усунення.....	38
3.3.3 Оцінка результатів	38
4 ІНСТРУКЦІЯ КОРИСТУВАЧА	39
4.1 Екран «Меню»(main.dart)	39
4.2 Екран «Завантаження файлу»(pick_file.dart).....	40
4.3 Екран «Ігровий»(questions.dart)	42
4.4 Екран «Редактор набору питань»(question_editor_screen.dart).....	45
ВИСНОВКИ.....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	50
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	52
ДОДАТОК Б ПРОГРАМНИЙ КОД	57
Б.1 main.dart	57
Б.2 pick_file.dart	62
Б.3 questions.dart	65
Б.4 question_editor_screen.dart	74

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Баг — помилка або несправність у програмі, що призводить до неправильної роботи.

Валідний файл — текстовий файл, що повністю відповідає структурним вимогам програми.

Віджет — графічний компонент інтерфейсу, наприклад, кнопка, поле вводу чи список.

Парсер — механізм, який аналізує текстовий файл і перетворює його у структуровані дані.

Скрол — прокручування вмісту екрана вертикально або горизонтально.

APK (Android Package) — файл встановлення застосунку на пристроях Android.

MVP (Minimum Viable Product) — мінімальна реалізація продукту, достатня для тестування ідей.

SDK (Software Development Kit) — набір інструментів для розробки програмного забезпечення.

UI/UX (User Interface/User Experience) — зовнішній вигляд програми та враження користувача від її використання.

ВСТУП

Мобільні застосунки у формі вікторин та ігор на логіку користуються великою популярністю серед молоді, студентів, викладачів та звичайних користувачів. Вони мають не лише розважальний, але й навчальний характер, що робить їх ефективним інструментом для розвитку пам'яті, логічного мислення та засвоєння нових знань.

Через умови COVID-19 та війни таких як: часті та далекі переїзди, віддалена робота чи навчання, перебої із світлом – усе це підвищило попит на зручні, прості, надійні та автономні сервіси. Все більше людей користуються сучасними технологіями, що створює сприятливе середовище для розвитку мобільних застосунків.

Метою цієї роботи є створення застосунку на Flutter для гри-вікторини з можливістю як завантаження готових тестів з файлу, так і створення нових тестів вручну у графічному інтерфейсі. У застосунку передбачена обробка відповідей користувача, підрахунок балів та виведення результатів. Додатково реалізовано підтримку портретної та альбомної орієнтації, а також вивід проєкту та автора.

У результаті був реалізований повноцінний програмний продукт, який може бути використаний для навчання, тренування пам'яті, організації змагань або самоперевірки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ

1.1 Актуальність теми

У сучасному світі цифрові технології відіграють ключову роль у навчанні, саморозвитку та професійному зростанні. Водночас зростає інтерес до інтерактивних методів подачі інформації, зокрема до гейміфікації — використання ігрових елементів у неігровому середовищі. Одним із найпоширеніших і найефективніших способів реалізації гейміфікації є гри-вікторини, які поєднують у собі елементи навчання, розваг і змагальності.

Актуальність розробки програмного застосунку гри-вікторини зумовлена декількома факторами. По-перше, традиційні способи контролю знань часто не викликають достатньої мотивації у користувачів, особливо серед молоді. Вікторини ж стимулюють інтерес до навчального процесу, сприяють повторенню матеріалу, закріпленню знань і розвитку мислення. По-друге, у зв'язку з масовим переходом на дистанційні форми навчання, особливо після пандемії COVID-19, виникає потреба у нових цифрових інструментах, що забезпечують інтерактивність і зворотний зв'язок між користувачем і системою.

Додатковим фактором актуальності є воєнний стан в Україні, через який освітній процес часто переривається або переноситься у формат асинхронного навчання. У таких умовах стають надзвичайно важливими автономні рішення, які не залежать від постійного підключення до Інтернету, прості у використанні, і можуть бути запущені на будь-якому мобільному пристрої. Програмний застосунок, розроблений у межах цієї кваліфікаційної роботи, не потребує реєстрації, інсталяції додаткових компонентів чи доступу до серверів. Єдиною умовою є наявність підготовленого .txt-файлу з питаннями. Це дозволяє використовувати застосунок навіть в укриттях,

сільській місцевості або під час аварійного відключення зв'язку.

Окрему актуальність має функціональність, яка дозволяє користувачам самостійно створювати власні набори запитань. Така можливість забезпечує гнучкість використання застосунку в найрізноманітніших сферах: від шкільних уроків та університетських занять до корпоративних тренінгів і неформального навчання. Це дозволяє адаптувати зміст вікторини під конкретну аудиторію, навчальну програму чи індивідуальні потреби, а також сприяє активному залученню користувачів до процесу створення контенту.

1.2 Аналіз існуючих аналогів

Попри наявність низки популярних вікторин, більшість із них є або комерційними, або обмеженими у можливостях редагування і створення власного контенту. Також спостерігається брак локалізованих рішень, адаптованих під конкретні освітні чи культурні контексти. Тому створення доступного, універсального програмного застосунку, що дозволяє користувачам самостійно формувати тематичні вікторини, є особливо актуальним.

Крім того, з розвитком інформаційних технологій зростає попит на програмні продукти, які є не лише функціональними, а й простими у використанні та адаптованими до різних категорій користувачів. Які саме переваги чи обмеження мають популярні аналоги, довелось мені дослідити.

1.2.1 Ігрова навчальна платформа «Kahoot!»

У цьому розділі розглянуто інтерфейс та функціонал платформи «Kahoot!».(рис. 1.1)



Рисунок 1.1 – Логотип платформи «Kahoot!»

Kahoot![3] – один із найпопулярніших сервісів для проведення вікторин у режимі реального часу. Він широко використовується у школах та вишах для залучення студентів до інтерактивного навчання. Проте платформа орієнтована переважно на синхронне використання (усі учасники мають відповідати одночасно) та потребує підключення до інтернету (рис. 1.2). Створення запитань також обмежене вільним функціоналом (рис. 1.3) – повна кастомізація доступна лише у платних версіях.

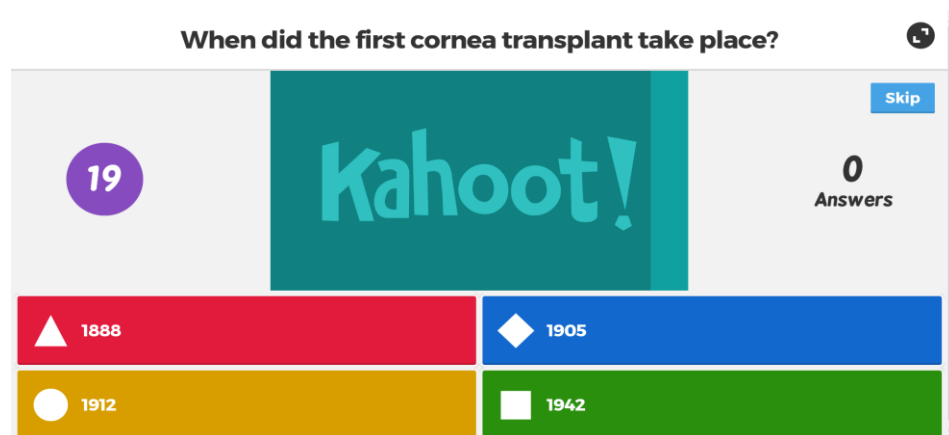


Рисунок 1.2 – Приклад процесу відповіді

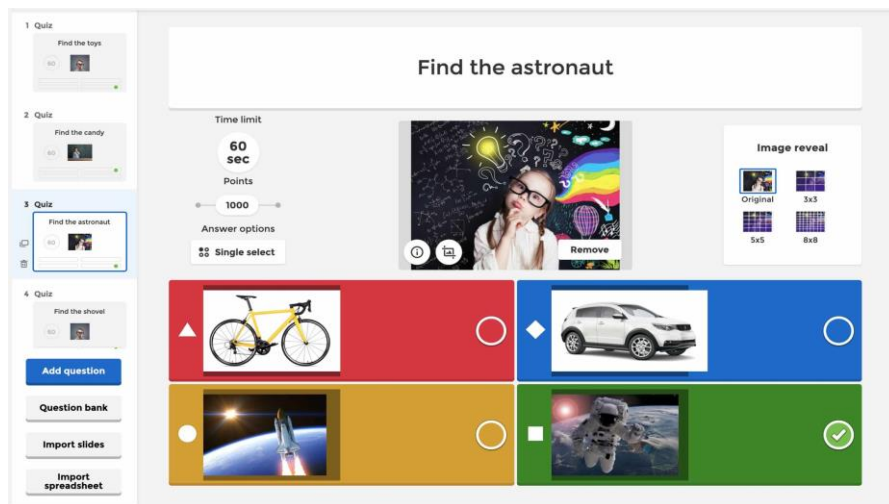


Рисунок 1.3 – Приклад редактора питань

Переваги:

- інтуїтивно зрозумілий інтерфейс;
- підтримка мультимедійного контенту (зображення, відео);
- можливість створювати загальнодоступні та приватні вікторини;
- рейтингова система для мотивації учасників.

Недоліки:

- основна функціональність орієнтована на синхронне використання;
- більшість додаткових можливостей доступні лише в платній версії;
- обмежена кастомізація оформлення та логіки гри.

1.2.2 Навчальна платформа «Quizizz»

У цьому розділі розглянуто інтерфейс та функціонал платформи «Quizizz» (рис.1.4).

Рисунок 1.4 – Логотип платформи «Quizizz»

Quizizz [4] пропонує ширший асортимент форматів, у тому числі індивідуальне проходження тестів (рис. 1.5, а), однак також має обмеження безкоштовної версії та складніший інтерфейс для нових користувачів (рис 1.5, б). Платформа дозволяє створювати власні запитання, проте не надає гнучкого механізму повторного використання наборів та категоризації тем (рис 1.5, в).

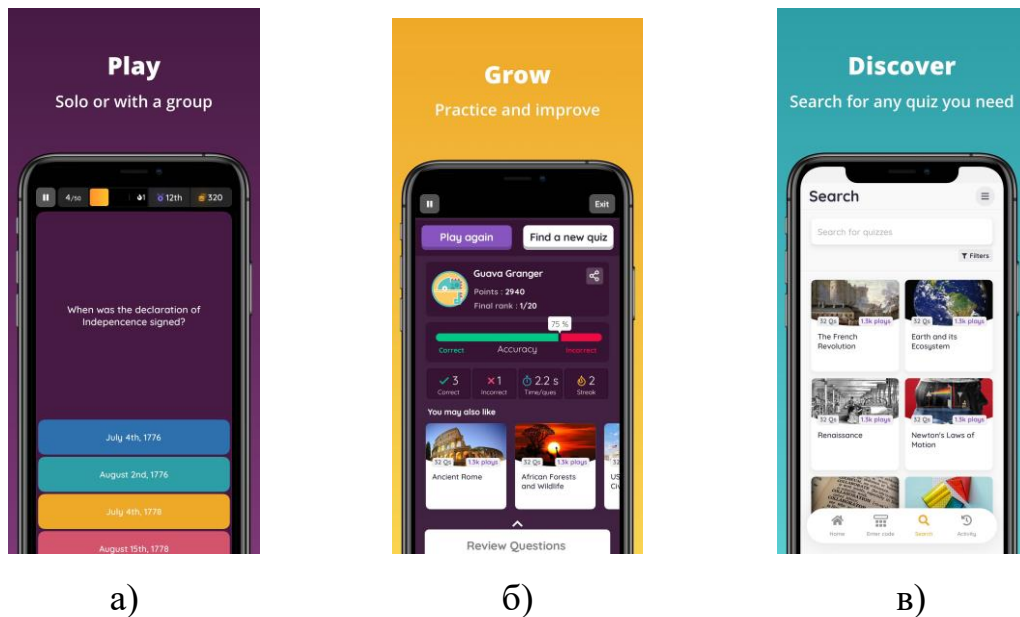


Рисунок 1.5 – Скриншоти застосунка: а) процес відповіді;
б) результати користувача; в) каталог тестів

Переваги:

- підтримка індивідуального та групового режимів гри;
- автоматичне збереження статистики відповідей та балів;
- велика бібліотека вже готових вікторин;

Недоліки:

- деякі функції, як-от імпорт запитань, аналітика тощо, доступні лише у платній версії;
- менш зручне налаштування структури вікторини порівняно з іншими сервісами.;
- відсутність повної свободи у створенні кастомного оформлення.

1.2.3 Навчальна платформа «Mentimeter»

У цьому розділі розглянуто інтерфейс та функціонал платформи «Mentimeter» (рис.1.6).



Рисунок 1.6 – Логотип компанії/застосунка «Mentimeter»

Mentimeter [5] більше орієнтований на презентації та голосування під час виступів (рис. 1.7), а не на повноцінні вікторини. Його вікторина функціональність є допоміжною та обмеженою за варіативністю типів запитань, способами взаємодії з користувачем та доступності функцій безкоштовної версії (рис. 1.8).

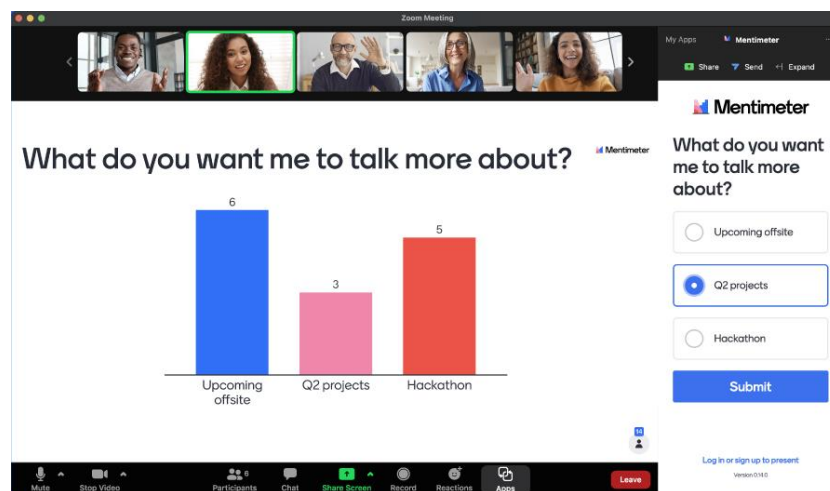


Рисунок 1.7 – Скриншот застосунка «Zoom» з інтегрованим застосунком «Mentimeter»[6]

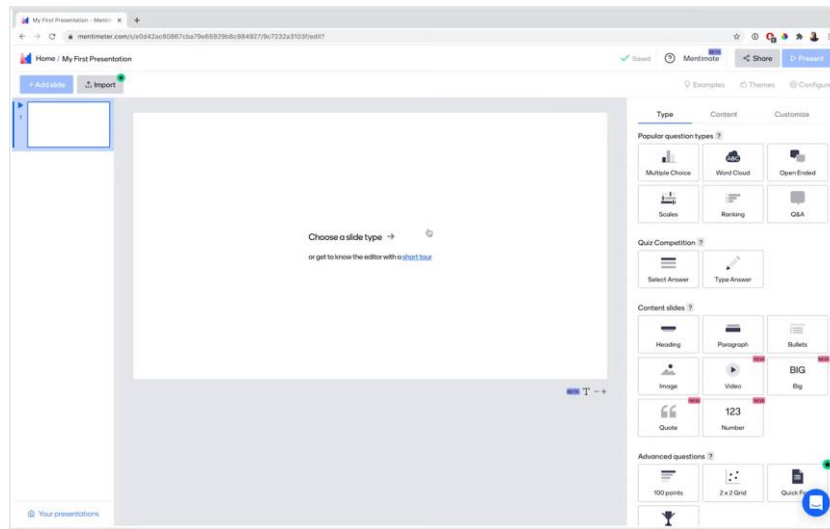


Рисунок 1.8 – Скриншот редактора презентацій «Mentimeter» зі списком доступних функцій

Переваги:

- гарна інтеграція з презентаційними матеріалами;
- різноманітні формати взаємодії з аудиторією;
- простота використання в режимі «питання-відповідь» у прямому ефірі.

Недоліки:

- вікторинна функціональність є вторинною й обмеженою;
- безкоштовна версія має обмеження на кількість слайдів та учасників;
- немає зручної структури для навчального чи самостійного використання.

1.2.4 Навчальний застосунок «Socrative»

У цьому розділі розглянуто інтерфейс та функціонал застосунка «Socrative»(рис.1.9).



Рисунок 1.9 – Логотип компанії/застосунку «Socrative»

Socrative [7] надає зручний інструмент (рис. 1.10) для вчителів з можливістю створення квізів, проте більшість функцій орієнтовані на освітній процес (рис. 1.11), і інтерфейс не адаптований для широкого загалу або неформального використання. Крім того, Socrative має обмеження на кількість активних користувачів у безкоштовній версії.

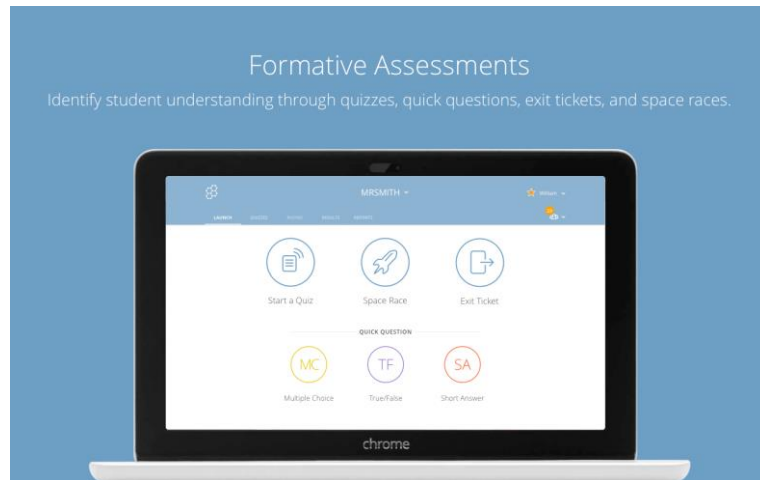


Рисунок 1.10 – Головний екран застосунка «Socrative»

Переваги:

- підтримка різних типів запитань (тестові, відкриті, правда/неправда);
- реальний час перевірки відповідей та аналітики;
- збереження результатів у форматі звітів.

The image shows the Socrative results screen for a 'Math Quiz'. It includes a 'FINISH' button and two toggle switches for 'Show Names' and 'Show Answers'. Below is a table with columns for Name, Progress, and four question numbers (#1, #2, #3, #4). The table shows individual student results and a 'Class Total' row.

Name	Progress	#1	#2	#3	#4
Noble, Donna	100% ✓	A	False	True	True
Pond Amy	100% ✓	A	False	True	True
Smith Mickey	100% ✓	C, A	True	True	True
Williams, Rory	100% ✓	A	True	True	True
Class Total		0%	50%	100%	100%

Click on Question #s or Class Total %s for a detailed question view

Рисунок 1.11 – Перегляд результатів учнів вчителем

Недоліки:

- обмеження на кількість активних класів у безкоштовній версії;
- орієнтованість лише на освітні установи;
- відсутність елементів гейміфікації, таких як рейтинг, анімація чи інтерактивність гри.

1.2.5 Програмне забезпечення для опитування «Google Forms»

У цьому розділі розглянуто інтерфейс та функціонал програмного забезпечення «Google Forms» (рис.1.12).



Рисунок 1.12 – Логотип «Google Forms»

Google Forms [8] – універсальний онлайн-інструмент для створення (рис 1.13) опитувань і тестів. Його часто використовують як простий спосіб контролю знань або збору інформації (рис. 1.14). Хоч він і дозволяє створювати тести, проте не передбачає гейміфікаційних елементів (таких як бали, таймер, рейтинг), які суттєво впливають на мотивацію користувачів.

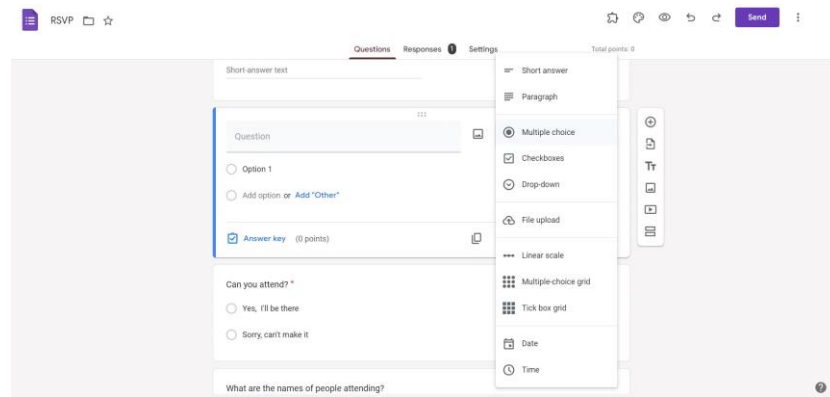


Рисунок 1.13 – Скриншот редактора питань «Google Forms»

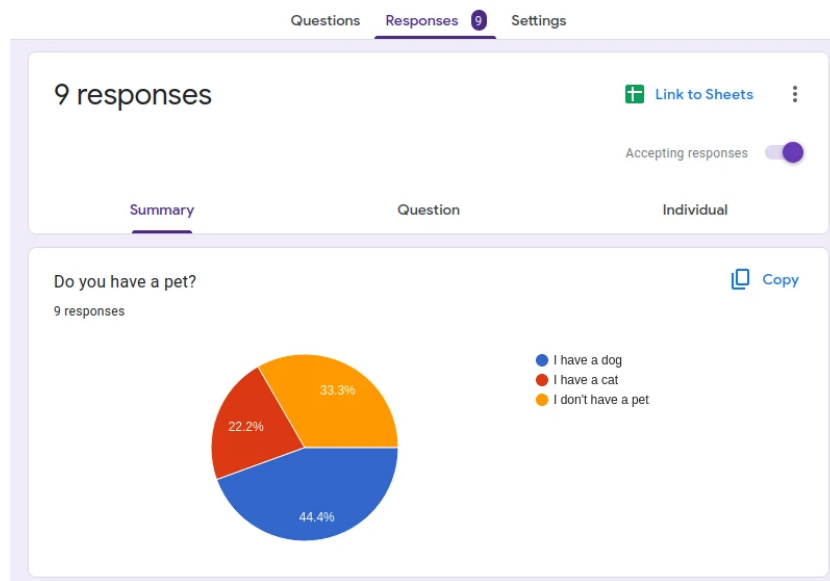


Рисунок 1.14 – Скриншот результатів відповідей

Переваги:

- простота створення форм;
- безкоштовний доступ для всіх користувачів Google;
- можливість автоперевірки тестів (із правильними відповідями).

Недоліки:

- відсутність гейміфікаційних елементів;
- немає інтерактивного ігрового інтерфейсу;
- обмежені можливості візуального налаштування.

1.2.6 Гра-вікторина «Party Quiz»

У цьому розділі розглянуто інтерфейс та функціонал гри-вікторини «Party Quiz» (рис.1.15).



Рисунок 1.15 – Логотип «Party Quiz»

Party Quiz [9] – це локальна вікторина для вечірок, доступна на платформі Steam. Гравці можуть створювати власні вікторини за допомогою вбудованого конструктора (рис. 1.16) або завантажувати готові набори запитань через Steam Workshop. Гра підтримує локальний мультиплеєр, а також можливість гри онлайн через спільний екран (рис. 1.17).



Рисунок 1.16 – Скриншот редактора питань «Party Quiz»



Рисунок 1.17 – Скриншот демонстрації питання гравцям

Переваги:

- можливість створення власних вікторин або використання спільноти для завантаження готових наборів;
- підтримка локального та онлайн-мультиплеєру;
- інтеграція з Steam Workshop для обміну контентом;
- позитивні відгуки користувачів на платформі Steam.

Недоліки:

- доступна лише через Steam, що обмежує аудиторію користувачів.
- відсутність мобільної версії;
- потребує встановлення додаткового програмного забезпечення для гри онлайн.

1.2.7 Навчальна платформа «Quizlet»

У цьому розділі розглянуто інтерфейс та функціонал платформи «Quizlet» (рис.1.18).

Quizlet

Рисунок 1.18 – Логотип «Quizlet»

Quizlet [10] – це онлайн-платформа для навчання, яка дозволяє створювати картки (рис. 1.19) з термінами й визначеннями, тестами, іграми (рис. 1.20), а також проходити вже готові навчальні набори. Основна аудиторія – школярі та студенти. Сервіс особливо популярний серед тих, хто вивчає іноземні мови, історію, біологію тощо.

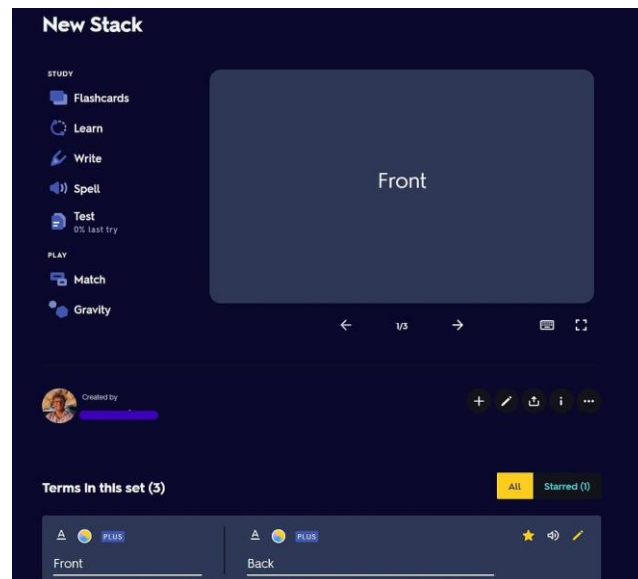


Рисунок 1.19 – Скриншот демонстрації карточек в «Quizlet»

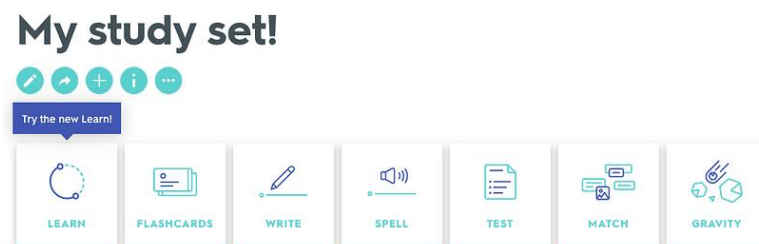


Рисунок 1.20 – Види тестування доступного в «Quizlet»

Переваги:

- простота створення навчальних наборів у вигляді флеш-карток;
- велика база вже створених матеріалів користувачами;
- різноманітні режими вивчення: тести, матчинг, «Spell», «Gravity» (гейміфікована гра), тощо;

- можливість створення навчальних груп та спільного використання матеріалів;

- наявність мобільного застосунку.

Недоліки:

- основний фокус на запам'ятовування інформації, а не на класичні вікторини з балами й змаганнями;

- частина функцій, зокрема докладна статистика й деякі режими гри, доступні лише у платній версії;

- обмежена кастомізація візуального оформлення та логіки ігрового процесу.

1.3 Аналіз розробленого застосунка

Розроблений застосунок гри-вікторини – це мобільна програма, створена для платформи Android, що поєднує в собі простоту використання, базову гейміфікацію та можливість повної автономної роботи. Головною ідеєю проекту стало забезпечення користувача гнучким інструментом для проходження й створення вікторин у зручному форматі, незалежно від наявності підключення до мережі Інтернет або зовнішніх сервісів.

Інтерфейс застосунка вирізняється зрозумілістю та лаконічністю. Його побудовано з орієнтацією на звичайного користувача – інтуїтивно зрозуміле розташування елементів дозволяє легко створювати нові тести, додавати запитання та запускати проходження без необхідності читати додаткові інструкції. Однією з ключових функцій стала можливість користувача самостійно створювати власні набори питань, що відкриває широкі можливості для індивідуалізованого використання програми – як у навчальному процесі, так і в розважальному середовищі, наприклад, під час тематичних вікторин у колі друзів чи родичів.

У застосунку реалізовано елементи гейміфікації, зокрема нарахування балів за правильні відповіді та підрахунок відсотка успішності, що сприяє

підтримці інтересу користувача до проходження тестів, отриманню нових знань і дає змогу оцінити власний результат. Усі перевірки відповідей виконуються автоматично – після кожної відповіді користувач одразу бачить зміну кількості балів та поточну оцінку своєї гри без необхідності ручної перевірки.

Окремо слід відзначити автономний характер застосунка. Його можна повноцінно використовувати без доступу до Інтернету, що робить його особливо зручним для мобільного застосування в дорозі, на заняттях або в умовах обмеженого зв'язку. Для повноцінної роботи достатньо лише наявності локальних файлів з наборами питань, які створюються й зберігаються користувачами. Жодні сторонні сервіси чи додаткові компоненти не потрібні.

На поточному етапі реалізовано лише текстові запитання, однак у подальшому планується розширення функціоналу з підтримкою мультимедійного контенту – зображень, аудіо та відео. Це значно збільшить дидактичний потенціал програми, дозволяючи створювати комплексніші та візуально привабливіші вікторини.

Ще однією особливістю є те, що розробка виконувалась одним розробником, що з одного боку дозволило зберегти цілісне бачення продукту та оперативно втілювати ідеї, а з іншого – дещо обмежило масштаб реалізації, особливо в частині розробки функцій синхронної гри між кількома користувачами. Така функціональність поки відсутня, однак розглядається як можливе розширення в майбутньому. Аналогічно, хоч наразі програма доступна лише для Android, у перспективі планується реалізація кросплатформної версії для iOS та інших систем.

Застосунок є повністю безкоштовним, не містить реклами чи платного контенту, що робить його максимально доступним для широкої аудиторії користувачів.

Таким чином, розроблений застосунок є функціональним, адаптивним і перспективним рішенням, яке вже на стартовому етапі може знайти

практичне застосування, а в подальшому – розвиватися в повноцінну мультиплатформну систему з розширеними інтерактивними можливостями.

2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ ТА ФОРМУВАННЯ СТРАТЕГІЇ РОЗРОБКИ

У процесі розробки програмного застосунку було обрано фреймворк Flutter як основну технологічну платформу, а також мову програмування Dart. Такий вибір зумовлений низкою факторів, які безпосередньо впливають на ефективність, масштабованість і гнучкість розробки.

2.1 Мова програмування Dart

Dart — це сучасна, об'єктно-орієнтована мова програмування, що має зручний синтаксис, підтримку асинхронного програмування через `async/await`, а також гарну інтеграцію з Flutter. Dart дозволяє легко створювати моделі даних та обробляти події у взаємодії з користувачем. Простота у використанні та хороша документація роблять його зручним вибором для мобільної розробки.

2.2 Фреймворк Flutter

Flutter забезпечує високу швидкість створення інтерфейсів користувача завдяки великій кількості вбудованих віджетів та можливості повного контролю над виглядом інтерфейсу. Його архітектура дозволяє розробляти UI з високим рівнем деталізації, адаптивністю до різних розмірів екранів та підтримкою анімацій без значного навантаження на систему.

Також важливо зазначити, що хоча Flutter декларує мультиплатформеність, поточна реалізація застосунку була протестована виключно на Android-платформі. Однак структура коду, використання універсальних віджетів та відмова від платформи-специфічних бібліотек дозволяють у майбутньому адаптувати застосунок до інших платформ без

суттєвих змін у логіці. Таким чином, реалізовано технологічний заділ для подальшого розширення функціональності та охоплення ширшої аудиторії.

2.3 Середовище розробки Android Studio

З метою зручності розробки та тестування було обрано середовище Android Studio[11], яке має повну інтеграцію з Flutter SDK та Android SDK. Android Studio забезпечило зручний відлагоджувальний процес, управління залежностями, повний контроль над збиранням APK-файлів, а також зручне візуальне середовище для написання та налагодження коду. Для тестування застосунку використовувалися виключно фізичні Android-пристрої, оскільки через технічні обмеження (нестача ресурсів та помилки запуску) не було можливості користуватися віртуальними емуляторами.

2.4 Сторонні бібліотеки

Додатково були використані такі сторонні бібліотеки:

- `file_picker` — для інтеграції з файловою системою пристрою. Дана бібліотека надає можливість відкривати локальні файли з пам'яті телефону, обмежуючи вибір лише певними типами (наприклад, `.txt`). Вона дозволила реалізувати зручний механізм завантаження тестових файлів;

- `auto_size_text` — забезпечує автоматичне масштабування розміру шрифту тексту відповідно до розміру контейнера. Це особливо важливо для гри, де питання різної довжини мають вписуватися у клітинки таблиці без порушення структури інтерфейсу.

2.5 Вибір технологій та інструментів

Розробка програмного забезпечення вимагала попереднього планування та поетапного впровадження функціональності. Для

забезпечення структурованості процесу було обрано поетапну стратегію розробки, яка включала кілька ключових фаз.

1. Аналіз вимог та постановка задач.

На цьому етапі було сформульовано основні функціональні вимоги до застосунку: підтримка структури з темами та питаннями, введення відповідей, підрахунок балів, автономна робота без інтернету, мінімалістичний інтерфейс та можливість створення власних тестів.

2. Розробка структури даних.

Було визначено базові моделі об'єктів: Question (питання), ThemeBlock (тематичний блок), а також формат вхідного файлу у вигляді .txt. Створено базовий парсер, який дозволяє конвертувати текстовий файл у внутрішню структуру.

3. Реалізація MVP.

На першому етапі створено базову гру: екран вибору файлу, логіка парсингу, відображення питань та введення відповідей. Це дозволило швидко протестувати основну функціональність.

4. Додавання користувацького інтерфейсу.

Було реалізовано графічне меню, оформлення кнопок, кольорову палітру, а також підтримку горизонтальної орієнтації. Для підвищення зручності взаємодії були застосовані функціональні віджети.

5. Розширення функціональності.

На цьому етапі було додано вікно створення нових тестів, діалог введення параметрів, обробку винятків при парсингу, фінальну статистику результатів гри та логіку завершення.

6. Тестування та оптимізація.

Після завершення основної реалізації проведено перевірку працездатності застосунку на фізичних пристроях. Усунено виявлені баги, удосконалено зручність користування.

Такий підхід дозволив поступово розширювати функціонал, не втрачаючи контроль над якістю та відповідністю поставленим вимогам.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Програмна реалізація застосунку «Challenging Game» охоплює розробку ключових функціональних модулів, організацію структури проекту, реалізацію логіки гри, механізмів зчитування файлів, а також побудову графічного інтерфейсу користувача.

3.1 Структура проекту

Застосунок побудовано у вигляді Flutter-проекту, що має чітке розділення логіки на кілька основних файлів:

- `main.dart` — точка входу в програму та реалізація екрану головного меню;
- `pick_file.dart` — екран з логікою вибору, відкриття і відображення текстових файлів з питаннями;
- `questions.dart` — екран з головним ігровим інтерфейсом, парсером та функціональними моделями даних;
- `question_editor_screen.dart` — екран для створення та збереження нових вікторин.

Кожен модуль відповідає за окрему логічну частину, що полегшує підтримку і масштабування проекту в майбутньому.

3.2 Опис файлів проекту

3.2.1 `main.dart`

Файл `main.dart` є початковою точкою входу у застосунок. У ньому визначено головний клас `SGameApp`, що ініціалізує `MaterialApp` з відповідною темою оформлення та головним екраном `Menu`.

Клас `Menu` є `StatelessWidget` і відповідає за побудову головного меню гри. Він містить назву гри (розділену на два елементи — «Challenging» та «game» для зберігання стилістичної узгодженості з використанням фірмового шрифту), а також основні кнопки, кожна з яких оформлена через `InkWell` та `Ink` для візуального ефекту натискання.

Передбачені такі функції:

- відкрити — перехід до екрана `FilePickerS` для вибору текстового файлу з питаннями;
- створити — відкриває діалогове вікно з полями для введення кількості тем та кількості питань, після чого здійснюється перехід до редактора `QuestionEditorScreen`;
- автор — викликає вікно `AboutDialog`, яке містить інформацію про версію програми та автора;
- вихід — реалізовано через `FloatingActionButton`, що викликає `SystemNavigator.pop()` для закриття застосунку.

Дизайн оформлений за допомогою кольорових градієнтів (зелена палітра), великих кнопок, фірмових шрифтів та вертикального компонування. Інтерфейс оптимізовано для портретної орієнтації. Через це ініціалізація орієнтації (`SystemChrome.setPreferredOrientations`) виконується при запуску програми, щоб зафіксувати початкову вертикальну орієнтацію.

3.2.2 `pick_file.dart`

Файл `pick_file.dart` реалізує функціональність вибору та попереднього перегляду тестового файлу, що містить питання вікторини. Він представлений як `StatefulWidget` під назвою `FilePickerS`, який забезпечує повний цикл взаємодії з локальною файловою системою пристрою.

Основні компоненти:

- кнопка «Обрати потрібний файл», яка відкриває файловий діалог за допомогою пакета `file_picker`;

- віджет Text, у якому відображається вміст обраного файлу або повідомлення про те, що файл не обраний;
- кнопка запуску гри (FloatingActionButton), яка стає активною лише після обрання валідного файлу;
- метод pickAndReadFile() — відкриває діалог вибору та зчитує файл у вигляді рядка через await file.readAsString();
- метод validateAndOpen() — перевіряє коректність структури файлу та викликає екран QuestionsScreen у разі успіху.

Передбачено обробку типових помилок:

- якщо файл не відповідає структурі, користувач бачить діалог з описом помилки (наприклад, відсутність теми, закритих дужок, або неправильне форматування);
- якщо файл не обрано, показується попередження з проханням спочатку обрати файл;
- усі дії користувача супроводжуються асинхронним оновленням інтерфейсу через setState();
- дизайн також містить AppBar з градієнтним оформленням, кнопкою повернення на головну (Navigator.of(context).pop());
- уся структура оптимізована під портретну орієнтацію та підтримує вертикальне прокручування для великого обсягу тексту за допомогою віджету SingleChildScrollView.

Модуль pick_file.dart забезпечує роботу з файловою системою користувача. За допомогою бібліотеки file_picker відкривається діалог для вибору .txt-файлу, після чого його вміст зчитується асинхронно. Якщо файл відповідає заданому формату (містить мітки #QUIZ_FILE_V1, themes=, questions=, [Тема:], {Питання:}, = та \$), виконується перехід до гри.

Функція parseSimpleQuiz із questions.dart відповідає за обробку даних. Вона перетворює текстовий вміст у список об'єктів ThemeBlock, кожен з яких містить питання та відповіді. Передбачено виявлення типових помилок структури, з відповідним повідомленням користувачу через AlertDialog.

Таким чином, `pick_file.dart` виконує функцію вхідного шлюзу до гри, перевіряючи правильність структури даних і забезпечуючи базову інтеракцію з користувачем.

3.2.3 `questions.dart`

Файл `questions.dart` є центральною складовою застосунку, оскільки він об'єднує логіку парсингу вікторини, визначення моделей даних, відображення ігрового інтерфейсу та обробки користувацьких відповідей.

Моделі даних:

У файлі оголошено дві ключові структури:

а) `Question`: зберігає одне питання вікторини. Містить такі поля:

- 1) `text` — текст питання;
- 2) `answer` — правильна відповідь;
- 3) `price` — вартість у балах;
- 4) `answered` — булеве значення, яке вказує, чи вже було це питання.

б) `ThemeBlock`: модель теми, яка включає назву та список об'єктів `Question`. Служить для логічного групування питань за спільною темою. Ці класи використовуються як базові елементи для парсингу й рендерингу гри.

Парсер текстових файлів:

Функція `parseSimpleQuiz(String rawText)` виконує розбір файлу формату `.txt`. Вона читає рядки по черзі, шукаючи мітки:

- `#QUIZ_FILE_V1` — заголовок файлу;
- `themes=..., questions=...` — кількість тем і питань;
- `[Тема: ...]` — початок нової теми;
- `{Питання: ...}` — початок питання;
- `=...` — відповідь;
- `$...` — кількість балів.

Функція створює список `ThemeBlock`, кожен з яких має власну назву та список `Question`. Якщо файл не відповідає очікуваному формату, парсер

кидає виняток, який перехоплюється і відображається користувачеві через `AlertDialog`. Також перевіряє правильну кількість тем і запитань. Це дозволяє забезпечити стабільність програми навіть при спробі завантажити некоректні файли.

Інтерфейс гри (`QuestionsScreen`).

Клас `QuestionsScreen` — це `StatefulWidget`, що реалізує головний ігровий інтерфейс. Він будує сітку тем і питань, які користувач проходить у форматі вікторини. Основні елементи:

- кнопка повернення до меню;
- поточний рахунок балів порівняно з максимально доступною кількістю балів;
- кнопка фіксації/розблокування зміни орієнтації застосунку;
- вертикальна колонка з назвами тем;
- сітка питань, представлених як кнопки з ціною;
- при натисканні — з'являється діалог з полем для введення відповіді.

Після введення:

- відповідь порівнюється з правильною без врахування регістру та пробілів, для простоти надання та перевірки відповіді;
- якщо вона вірна — додаються та показуються нараховані бали;
- якщо ні — знімаються бали, і показується правильна відповідь.

Додаткова логіка:

- кожні клітинки блокуються не залежно від відповіді;
- коли відповіді надані на всі питання — гра завершується і виводиться діалог із підсумками;
- підраховується кількість правильних відповідей та поточний рахунок.

Технічні особливості:

- синхронізація прокрутки тем та ігрового поля — `ScrollController`;
- вбудована адаптація до горизонтальної чи вертикальної орієнтації для покращення огляду сітки;

- кольорове маркування клітинок (сині — ще не пройдені, сірі — пройдені, темно-зелені — пусті);
- усі взаємодії побудовано через `setState()` з відображенням змін в реальному часі.

Файл `questions.dart` є ядром ігрової логіки. Він не лише керує інтерфейсом гри, а й забезпечує надійний механізм обробки даних, перевірку структури тестів, адаптивне відображення питань і зручну навігацію. Його структура дозволяє легко масштабувати застосунок — наприклад, додати таймер, рейтингову систему чи мультимовну підтримку.

3.2.4 `question_editor_screen.dart`

Файл `question_editor_screen.dart` реалізує повноцінний редактор тестів, що дозволяє створювати нові вікторини безпосередньо всередині застосунку. Він побудований як динамічний інтерфейс введення даних з можливістю експорту результату у `.txt`-файл відповідного формату.

Основні класи:

- `EditableTheme` — модель теми з назвою та списком питань;
- `EditableQuestion` — модель одного запитання з текстом, правильною відповіддю та балами;
- `QuestionEditorScreen` — головний `StatefulWidget`, який відповідає за всю логіку введення, перевірки та збереження даних.

Ініціалізація та структура.

У `initState()` створюється список тем (`themes`), кожна з яких має попередньо згенеровану кількість порожніх питань. Структура побудована динамічно залежно від кількості тем і запитань, переданих із головного меню лістинг 1.1.

Лістинг 1.1 – Створення списку тем (файл question_editor_screen.dart)

```

themes = List.generate(widget.themeCount, (i) {
  return EditableTheme(
    title: 'Тема ${i + 1}',
    questions: List.generate(widget.maxQuestionsPerTheme, (_) =>
EditableQuestion()),
  );
});

```

Також реалізовано синхронізацію вертикального скролу для колонок із назвами тем і блоків запитань за допомогою двох ScrollController.

Графічний інтерфейс.

Інтерфейс побудовано у вигляді двох частин:

- а) ліва колонка: перелік тем, які редагуються (назва теми + AutoSizeText);
- б) права область: сітка Card-блоків для кожного запитання з полями:
 - 1) текст запитання (TextField);
 - 2) правильна відповідь (TextField);
 - 3) кількість балів (TextField з keyboardType: number).

Вертикальна та горизонтальна прокрутка дозволяє працювати з великою кількістю тем і запитань.

Управління орієнтацією.

Передбачено можливість зміни орієнтації екрана під час редагування:

- кнопка в AppBar перемикає між тільки горизонтальною та вільною орієнтацією;
- статус зміни орієнтації зберігається у змінній isLandscape.

Збереження у .txt.

Після введення всіх потрібних питань користувач натискає кнопку «Зберегти як TXT», що відкриває діалог для введення імені файлу, після вибору якого Функція _exportToTxt():

- формує структуру тексту згідно зі специфікацією #QUIZ_FILE_V1;
- рахує загальну кількість запитань;
- будує файл у вигляді лістинг 1.2.

Лістинг 1.2 – Приклад оформлення файлу набору (файл `question_editor_screen.dart`)

```
#QUIZ_FILE_V1
themes=3
questions=15

[Тема: Математика]
{Питання: Скільки буде 2+2?}
=4
$100
...
```

- зберігає файл у вибрану директорію за допомогою `file_picker.getDirectoryPath()`;

- показує `AlertDialog` з підтвердженням шляху та статусом збереження;

- через 10 секунд закриває діалог та повертає користувача назад до меню редагування набору питань;

- та ,якщо користувач не захоче нічого міняти, може натиснути кнопку `home` для повернення до меню.

Особливості:

а) автоматичне очищення полів після повернення до меню;

б) поля, що залишились порожніми:

- 1) теми – зберігаються пустими;

- 2) питання – автоматично ігноруються при експорті;

- 3) відповіді – зберігаються пустими;

- 4) ціна – зберігаються як нуль;

в) інтеграція з буфером обміну тимчасово вимкнена (залишено у коді як коментар);

г) UI-елементи використовують стандартні Material-компоненти (`Card`, `TextField`, `ListTile`) та адаптивне масштабування (`AutoSizeText`).

`question_editor_screen.dart` є ключовим інструментом розширення функціональності застосунку. Він надає користувачеві простий, але

потужний спосіб створювати власні вікторини без використання сторонніх текстових редакторів. Цей модуль значно підвищує гнучкість і автономність гри, дозволяючи налаштувати контент відповідно до навчальних або розважальних потреб користувача.

3.3 Тестування та результати

Тестування програмного застосунку «Challenging Game» проводилося з метою перевірки стабільності роботи, відповідності функціональності до технічного завдання, а також зручності користування. Особлива увага приділялася обробці помилок, адаптивності інтерфейсу та коректності логіки гри.

3.3.1 Методи тестування

Для перевірки функціональності було застосовано ручне тестування на фізичних Android-пристроях. Емулятори не використовувались через технічні обмеження робочої машини.

Проведено наступні типи тестів:

- функціональне тестування — перевірка коректної роботи кожного екрану;
- тестування помилок — перевірка поведінки застосунку при неправильному форматі вхідних даних (відсутність міток, неправильний формат питань тощо);
- UI/UX тестування — аналіз зручності інтерфейсу, адаптивності в портретному й альбомному режимах, логіки скролу;
- краєві випадки — перевірка роботи при дуже довгих питаннях, відсутніх відповідях, нульовій ціні;
- тестування збереження — перевірка генерації .txt-файлу в редакторі, перевірка шляху збереження, коректності структури.

3.3.2 Виявлені помилки та усунення

У процесі тестування було виявлено такі проблеми:

- некоректне збереження файлу при незаповненому імені — виправлено через діалог введення назви;
- неможливість повернення у вертикальну орієнтацію — реалізовано перемикач орієнтації в AppBar;
- проблеми зі скролом тем і гри — синхронізовано через два ScrollController;
- некоректне відображення відповіді при помилці — додано діалог з правильною відповіддю.

Та дедалі більше, але під час налагодження майже відразу усувались. Та після усунення виявлених помилок застосунок демонстрував стабільну роботу та правильну логіку у всіх перевірених сценаріях.

3.3.3 Оцінка результатів

Усі основні функції застосунку реалізовано та протестовано. Робота з файлами, обробка помилок, введення відповідей, редактор та збереження виконуються згідно з очікуваннями. Застосунок адаптивний до розмірів екрана, працює стабільно в альбомній орієнтації під час гри та не потребує доступу до Інтернету.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

Для того щоб почати роботу із застосунком, треба отримати та встановити APK-файл на мобільному пристрої користувача. Після процесу встановлення на екрані мобільного пристрою потрібно знайти ярлик даного застосунку, відкрити його та побачити перед собою перший екран.

4.1 Екран «Меню»(main.dart)

Після запуску застосунку користувач потрапляє на орієнтований вертикально головний екран (рисунок 4.1, а) та бачить:

- 1) у «шапці» зазначено назву поточного екрану – «Меню»;
- 2) у «тілі» знаходяться:
 - а) назва застосунку «Challenging game» - оформлена фірмовим шрифтом (Notable);
 - б) кнопка «Відкрити» - яка відкриває екран «Завантаження файлу»;
 - в) кнопка «Створити» (рисунок 4.1, б) - яка відкриває діалогове вікно, в якому треба ввести кількість тем та максимальний розмір теми для переходу на екран «редактор набору питань» в якому відкриється шаблон набору питань із заданими параметрами;
 - г) кнопка «Автор»(рисунок 4.1, в) - яка відкриває діалог в якому зазначений автор, версія застосунку та використані ліцензії;
 - д) кнопка «Exit» - яка закриває застосунок.

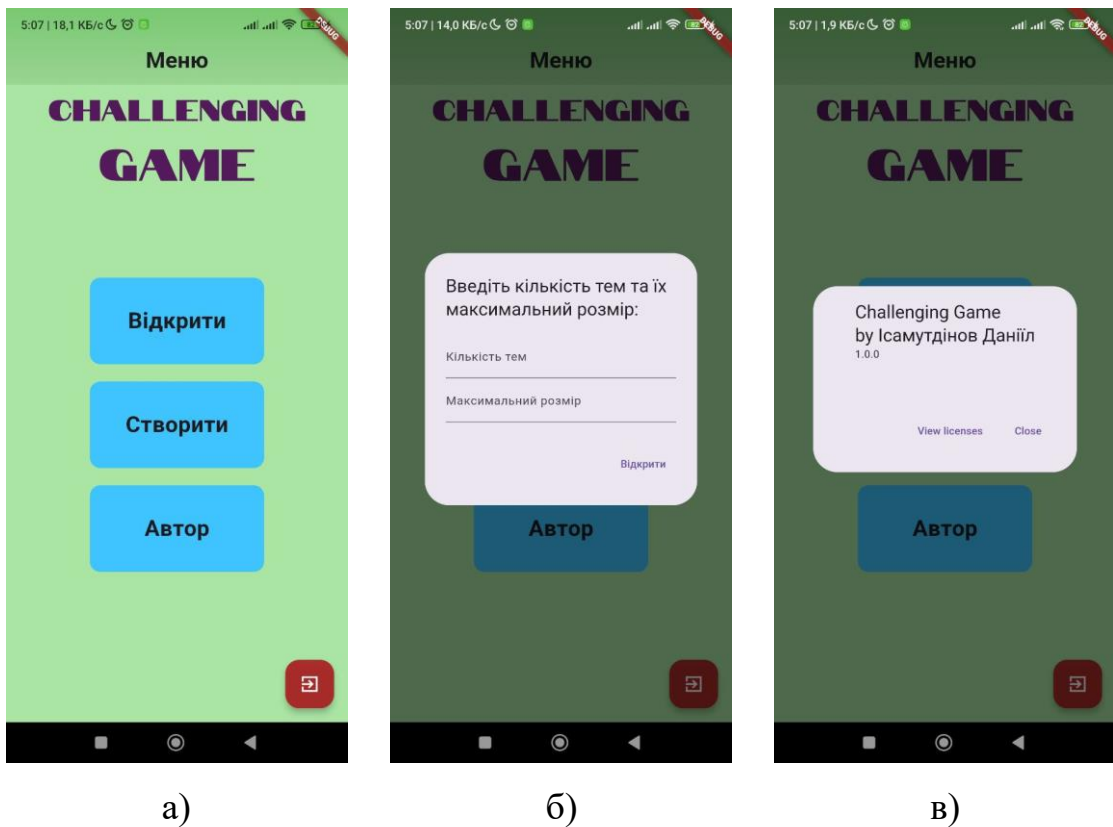


Рисунок 4.1 – Екран «Меню»: а) загальний вигляд; б) кнопка «Автор»; в) кнопка «Створити»

4.2 Екран «Завантаження файлу»(pick_file.dart)

Після натискання кнопки «Відкрити» в «Меню», користувач потрапляє на Екран «Завантаження файлу» (рисунок 4.2, а) та бачить:

1) у «шапці» зазначено назву поточного екрану – «Завантаження файлу»

2) у «тілі» знаходяться:

а) кнопка «Обрати потрібний файл» – після натискання якої, відкривається вікно, в якому користувач знаходить та вибирає потрібний йому файл (рисунок 4.2, б);

б) текстове поле – в якому спочатку буде написано «Текст файлу буде відображатись тут», якщо користувач почав відкривати файл але повернувся з нічим, то буде написано «Файл не обраний» (рисунок 4.2, в), якщо був відкритий txt файл, то буде відображатись його вміст (рисунок 4.3,

а);

в) кнопка «Let's play» – яка, якщо файл не обраний, буде сірою та при натисканні відобразиться діалог з текстом «Ви ще не обрали потрібний файл!» (рисунок 4.3, б), а якщо файл було відкрито, то вона окраситься в зелений, та при натисканні запуститься валідатор в результаті якого, якщо виникла помилка, вона виведеться на екран (рисунок 4.3, в), а якщо все пройшло, відкриється екран самої гри з обраним набором питань.

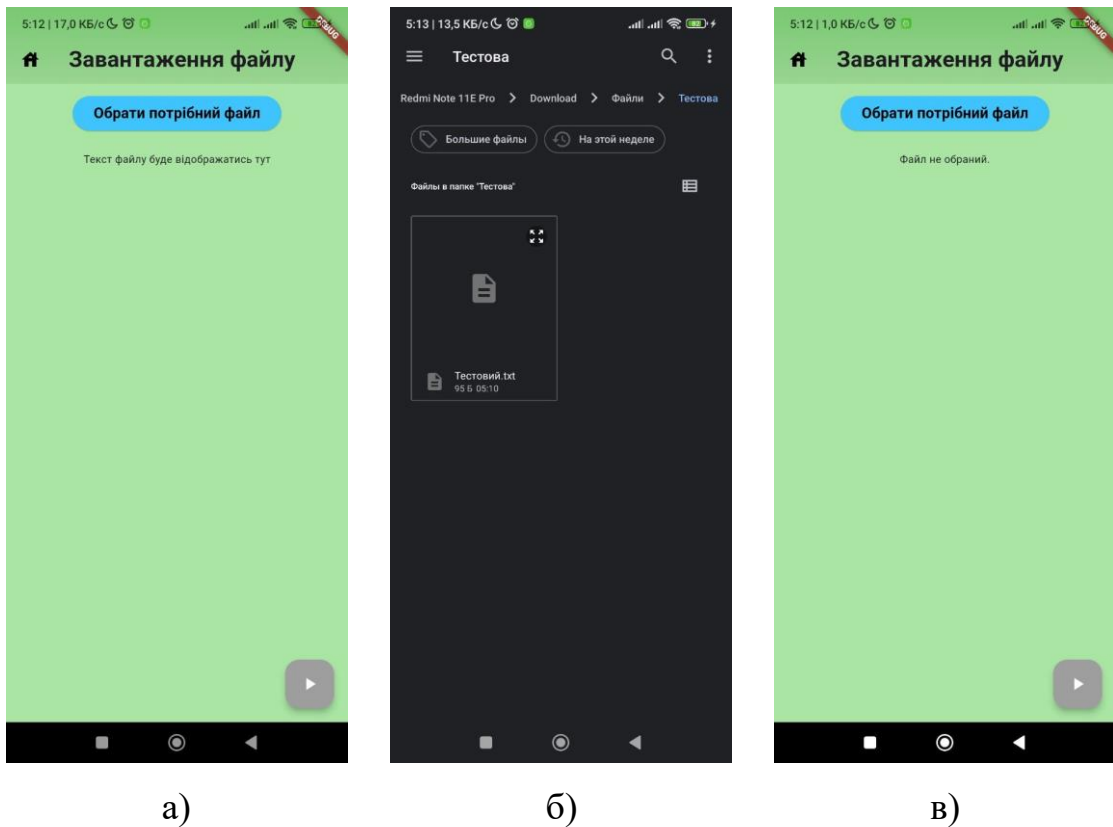


Рисунок 4.2 – Екран «Завантаження файлу»: а) загальний вигляд; б) вибір потрібного файлу; в) текстове поле, якщо не обраний файл;

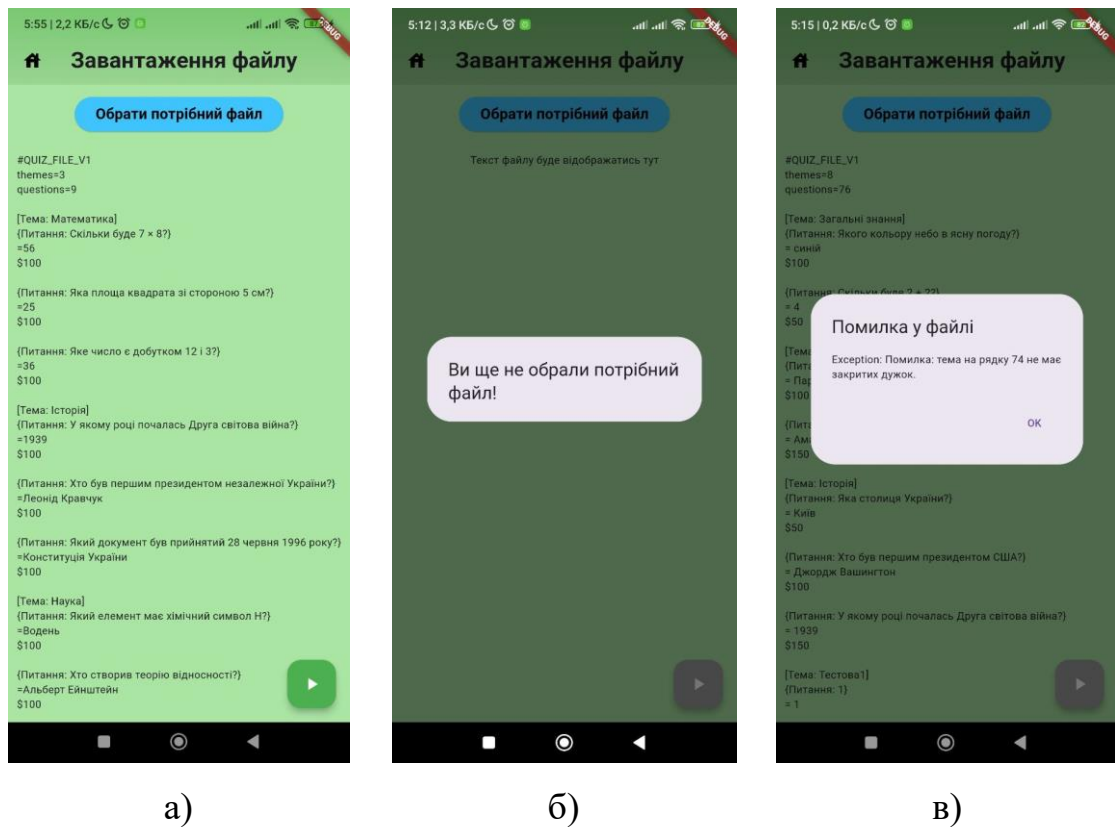


Рисунок 4.3 – Екран «Завантаження файлу»: а) з текстом із файлу; б) кнопка «Let's play», якщо не обраний файл; в) помилка від валідатора

4.3 Екран «Ігровий»(questions.dart)

Після натискання кнопки «Let's play» в екрані «Завантаження файлу» та успішного проходження валідатора, користувач потрапляє на Екран «Ігровий»(рисунок 4.4, а) та бачить:

1) у «шапці» знаходяться:

а) кнопка «home» - яка при натисканні закриває файл та повертається до екрану «Меню»;

б) текстове поле – в якому відображається поточні та максимальні бали;

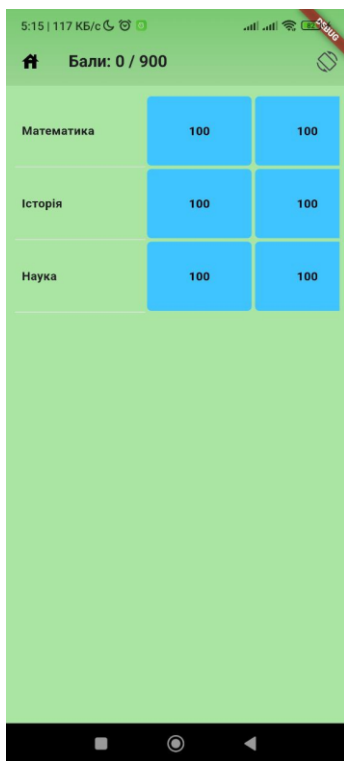
в) кнопка «Змінити орієнтацію» - яка фіксує або розблоковує зміну орієнтації застосунку(рисунок 4.4, б);

2) у «тілі» знаходяться:

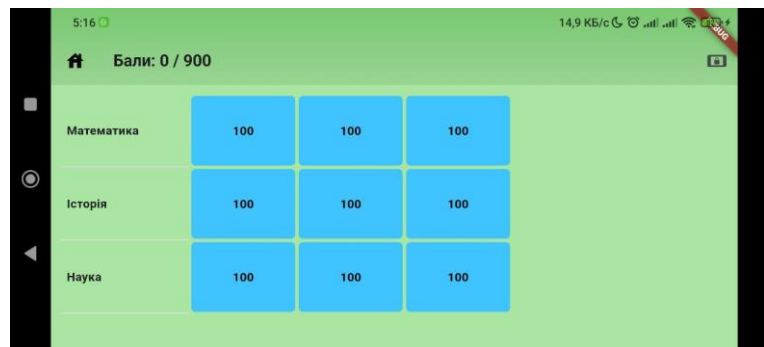
а) зліва колонка з темами;

б) сітка із кнопок з ціною – які при натисканні відкривають діалогові вікна відповідних питань (рисунок 4.5, а). Користувач повинен у полі вводу, що з'явився написати відповідь на питання що відображається у вікні. Після того як користувач ввів відповідь, він повинен натиснути кнопку «Відповісти», щоб отримати результат: «відповів вірно +бали» (рисунок 4.5, б) чи «відповів не вірно - бали, ось така правильна відповідь» (рисунок 4.5, в).

Також після того, як користувач пройде усі питання, відкриється вікно в якому буде сказано, що користувач завершив гру, отримав свої бали з максимальної кількості та на скільки питань відповів правильно (рисунок 4.6). Після цього, якщо натисне кнопку «ОК», повернеться до вікна «Меню».



а)



б)

Рисунок 4.4 – Екран «Ігровий»: а) загальний вигляд; б) після зміни орієнтації

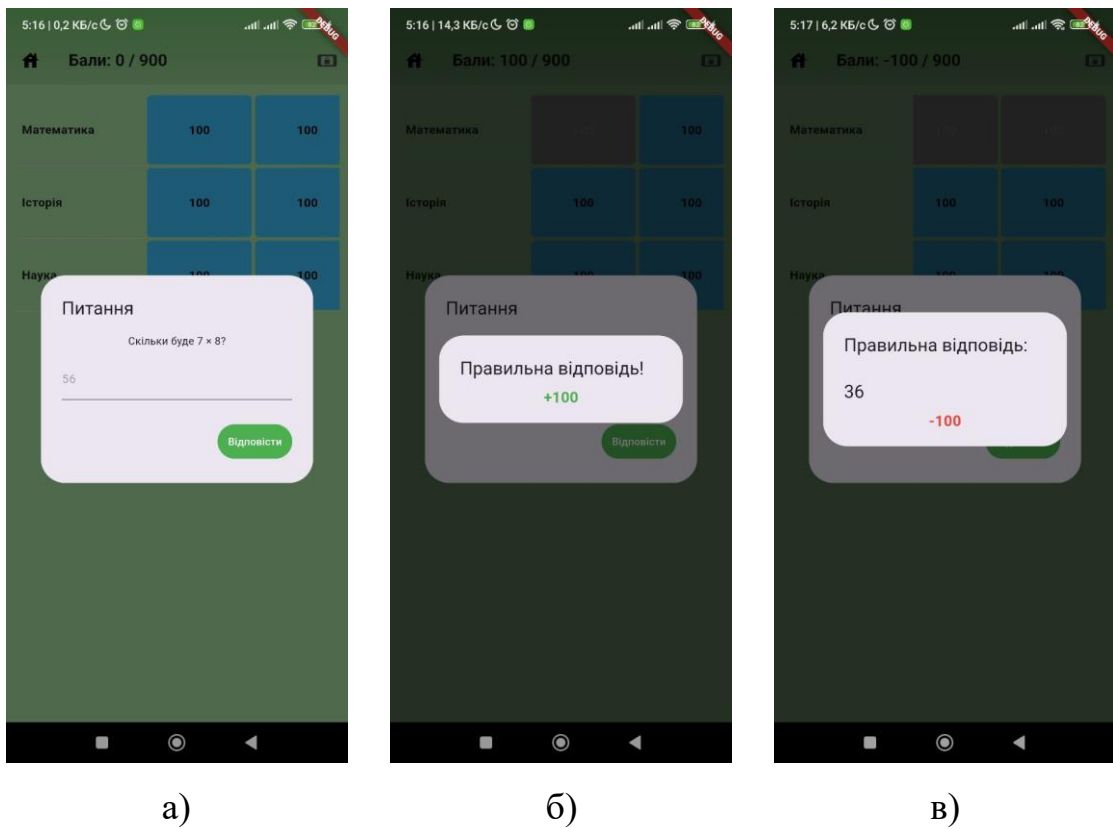


Рисунок 4.5 – Екран «Ігровий»: а) діалогове вікно з питанням; б) правильна відповідь; в) неправильна відповідь

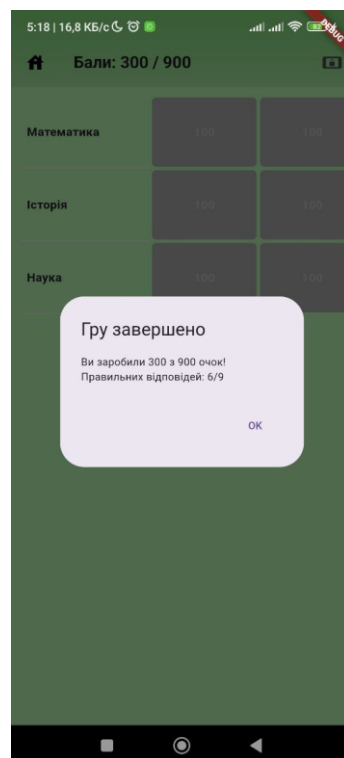


Рисунок 4.6 – Екран завершення гри та результати

4.4 Екран «Редактор набору питань»(question_editor_screen.dart)

Після введення кількості тем та максимального розміру теми у вікні, який відкривається після натискання кнопки «Створити», користувач перейде на екран «Редактору набору питань» в якому відкриється шаблон набору питань із заданими параметрами (рисунок 4.7, а) та користувач бачить:

1) у «шапці» знаходяться:

а) кнопка «home» – яка при натисканні закриває і не зберігає файл та повертається до екрану «Меню»;

б) текстове поле – в якому зазначено назву поточного екрану – «Редактор набору питань»;

в) кнопка «Зберег як TXT» – при натисканні на яку, відкриється діалогове вікно (рисунок 4.7, б), в якому користувач повинен ввести назву файлу, яку він хоче дати редагованому в даний час файлу. Після введення потрібної назви, користувач натискає на кнопку «Куди зберігати?» та обирає куди буде збережено файл (рисунок 4.7, в). Та після успішного збереження вилізе діалогове вікно, в якому буде сказано, що по обраному користувачем шляху було збережено файл (рисунок 4.8, а).

г) кнопка «Змінити орієнтацію» - яка фіксує або розблоковує зміну орієнтації застосунку (рисунок 4.8, б);

2) у «тілі» знаходяться:

а) зліва колонка з текстовими полями – в які користувач буде вписувати назву відповідної теми;

б) сітка із карток – в яких є: номер питання по порядку теми, три текстових поля в які користувач вписує текст, відповідь та ціну відповідного питання.

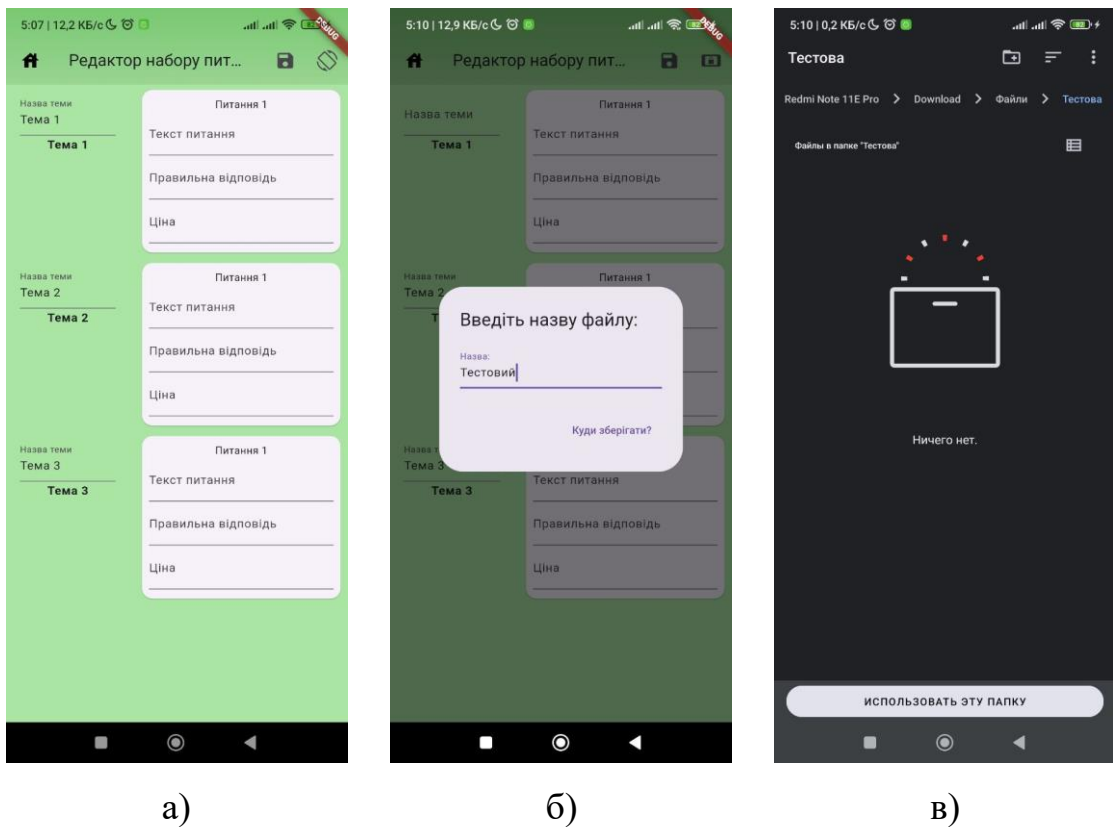


Рисунок 4.7 – Екран «Редактору набору питань»: а) загальний вигляд; б) ввід назви файлу; в) вибір шляху збереження файлу

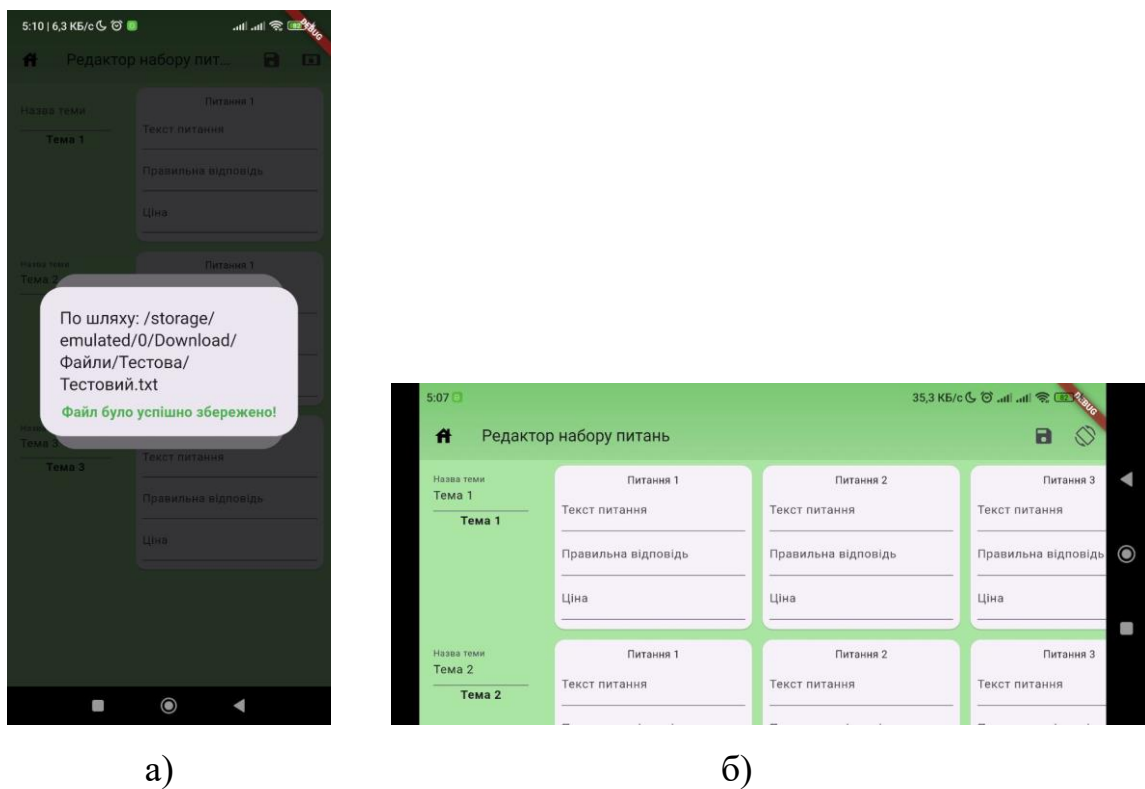


Рисунок 4.8 – Екран «Редактору набору питань»: а) Вікно успішно збереженого файлу; б) після зміни орієнтації

Застосунок не вимагає реєстрації, не використовує мережу Інтернет та працює повністю автономно. Увесь зміст вікторин зберігається у відкритому текстовому форматі.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було реалізовано повнофункціональний мобільний застосунок у вигляді гри-вікторини, що поєднує навчальні та розважальні елементи. Основна увага приділялася автономності, простоті використання, адаптивності інтерфейсу та можливості самостійного створення користувацького контенту.

Основні досягнення:

- розроблено застосунок на базі Flutter/Dart з можливістю запуску на Android-пристроях;
- реалізовано зручне головне меню, екран вибору тесту, повноцінну логіку гри, а також графічний редактор для створення нових вікторин;
- забезпечено підтримку імпорту та експорту файлів у текстовому форматі, що спрощує поширення і повторне використання тестів;
- проведено тестування на фізичних пристроях, виявлені недоліки були усунені, а застосунок продемонстрував стабільну роботу;
- забезпечено повну автономність: застосунок не потребує доступу до Інтернету та сторонніх сервісів.

Актуальність реалізованого застосунку підтверджується потребами в інтерактивних, автономних навчальних інструментах, особливо в умовах нестабільного інтернет-з'єднання (як після COVID-19 чи в умовах війни в Україні).

Практична цінність проекту полягає в тому, що він може використовуватись не лише для індивідуального навчання, а й у закладах освіти, на тренінгах або навіть у корпоративному середовищі.

Подальші напрями розвитку: додати підтримку багатокористувацького режиму або командної гри; зберігати статистику проходжень і персональні результати; розширити редактор: дозволити зберігати кілька наборів питань, змінювати мову тощо; забезпечити підтримку інших платформ (iOS, Web)

завдяки мультиплатформенності Flutter; розширити функціонал пакетів питань: різні типи питань, підтримка мультимедіа та багаторядковість.

Застосунок «Challenging Game» став прикладом того, як за допомогою сучасних інструментів можна реалізувати гнучке, надійне та корисне мобільне рішення з освітнім потенціалом.

Додатково, основні тези дослідження гейміфікації в освітньому процесі викладені в матеріалах конференції [12].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Flutter [Електронний ресурс]. – Режим доступу: <https://flutter.dev> (дата звернення: 26.05.2025).
2. Dart [Електронний ресурс]. – Режим доступу: <https://dart.dev> (дата звернення: 26.05.2025).
3. Kahoot! [Електронний ресурс]. – Режим доступу: <https://kahoot.com> (дата звернення: 26.05.2025).
4. Quizizz [Електронний ресурс]. – Режим доступу: <https://quizizz.com> (дата звернення: 26.05.2025).
5. Mentimeter [Електронний ресурс]. – Режим доступу: <https://www.mentimeter.com> (дата звернення: 26.05.2025).
6. Mentimeter app for Zoom [Електронний ресурс]. – Режим доступу: <https://www.mentimeter.com/integrations/zoom> (дата звернення: 26.05.2025).
7. Socrative [Електронний ресурс]. – Режим доступу: <https://www.socrative.com> (дата звернення: 26.05.2025).
8. Google Forms [Електронний ресурс]. – Режим доступу: <https://workspace.google.com/products/forms/> (дата звернення: 26.05.2025).
9. Party Quiz [Електронний ресурс]. – Режим доступу: https://store.steampowered.com/app/2024070/Party_Quiz/ (дата звернення: 26.05.2025).
10. Quizlet [Електронний ресурс]. – Режим доступу: <https://quizlet.com> (дата звернення: 26.05.2025).
11. Середовище розробки Android Studio [Електронний ресурс]. – Режим доступу: <https://developer.android.com/studio> (дата звернення: 26.05.2025).
12. Єр'оміна Н., Ісамутдінов.Д. Дослідження гейміфікації в освітньому процесі // Scientific research: modern challenges and future prospects. Proceedings of the 11th International scientific and practical conference. MDPC

Publishing. Munich, Germany. 2025. Pp. 293-296. URL: <https://sci-conf.com.ua/xi-mizhnarodna-naukovo-praktichna-konferentsiya-scientific-research-modern-challenges-and-future-prospects-9-11-06-2025-myunhen-nimechchina-arhiv/>.