

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

«До захисту допущено»

Завідувач кафедри

В. М. Терещенко

_____ (підпис)

«___» _____ 20__ р.


**Дипломна робота
на здобуття ступеня бакалавра**

за спеціальністю 122 Комп'ютерні науки

на тему:

**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ "СИСТЕМА ТЕСТУВАННЯ"
ПІД ANDROID**

Виконала студентка 4 курсу
Медовщук Віта Віталіївна

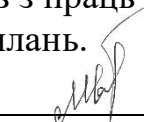

_____ (підпис)

Науковий керівник:
доцент кафедри математичної інформатики,
кандидат фіз.-мат. наук
Дервянченко Олександр Валерійович


_____ (підпис)

Засвідчую, що в цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент


_____ (підпис)

РЕФЕРАТ

Обсяг роботи 45 сторінок, 24 ілюстрації, 15 джерел посилань.

ІНТЕРФЕЙС ПРОГРАМНОГО ПРОДУКТУ, ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОДУКТУ, МОБІЛЬНИЙ ДОДАТОК, ОПИТУВАННЯ, ПЛАТФОРМА ДЛЯ НАВЧАННЯ.

Об'єктом роботи постає процес розроблення мобільного додатку, який дає змогу формувати й публікувати опитування різного ступеня складності з можливістю перегляду отриманих відповідей. Предметом роботи є програмний засіб для побудови опитувань із можливістю проходження та перегляду результатів.

Метою роботи є створення навчального програмного засобу для опитувань із ціллю автоматизувати процес перевірки знань та полегшити дистанційне навчання для учнів.

Інструменти розроблення: Android Studio [1] - інтегроване середовище розробки (IDE) для роботи з платформою Android; статично типізована, об'єктно-орієнтована мова програмування Kotlin [2]; Cloud Firestore [3] – гнучка, масштабована хмарна база даних NoSQL для зберігання й синхронізації даних для клієнтської і серверної розробки; Firebase [4] - це платформа розроблена Google для створення мобільних та веб-додатків, постачальник хмарних послуг.

Результати роботи: виконано загальний огляд веб-сервісів для конструкторів опитувань у режимі реального часу, проаналізовано переваги та недоліки використання програмних засобів у процесі перевірки знань, розроблено програмний продукт «Система для проведення онлайн-опитувань та тестувань».

Програмний продукт «Система для проведення онлайн-опитувань та тестувань» може застосовуватися в навчальному процесі шкіл, університетів та інших закладів навчання.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	5
ВСТУП.....	6
1 ОГЛЯД ІСНУЮЧИХ НА РИНКУ СИТЕМ.....	9
1.1 Google Forms.....	9
1.2 Quizizz.....	10
1.3 Surveymonkey.....	11
2 ОСНОВНІ ЕТАПИ СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ	12
2.1 Ідея	12
2.2 Побудова стратегії.....	12
2.3 Створення попередньої версії продукту	13
2.4 UI / UX дизайн.....	13
2.5 Розробка.....	14
2.6 Тестування	14
2.7 Розгортання.....	15
2.8 Підтримка.....	15
3 ВИКОРИСТАНІ ПРОГРАМНІ ЗАСОБИ, МОВИ ПРОГРАМУВАННЯ ТА ФРЕЙМВОРКИ.....	16
3.1 Android Studio	16
3.2 Kotlin	17
3.3 Firebase.....	18
3.3.1 Cloud Firestore	19
3.3.2 Cloud Storage	20
3.3.3 Firebase Authentication	21
4 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ.....	22
4.1. Призначення системи	22
4.2 Цілі створення системи	22
5 ВИМОГИ ДО СИСТЕМИ.....	23
5.1 Вимоги до функцій, які виконуються системою	23
5.2 Технічні вимоги	23
6 ОПИС ОРГАНІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	24
6.1 Логічна структура бази даних.....	24
7 ІНСТРУКЦІЯ КОРИСТУВАЧА.....	26

7.1 Іконка та екран-заставка	26
7.2 Сторінка входу.....	27
7.3 Сторінка реєстрації.....	28
7.4 Сторінка відновлення паролю	29
7.5 Сторінка доповнення профілю	30
7.6 Сторінка даних профілю	31
7.7 Сторінка приєднання до тесту	32
7.8 Сторінка проходження тесту	33
7.9 Результат тесту	34
7.10 Сторінка з власними тестами.....	34
7.11 Сторінка створення нового питання.....	35
ВИСНОВОК	37
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	40
ДОДАТОК А	42
Лістинг коду деяких функцій в додатку	42

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

IDE - Integrated Design Environment, інтегроване середовище розробки;

EOM - електронно-обчислювальна машина;

UI - User Interface, призначений для користувача інтерфейс;

UX - User Experience, це те, що людина відчуває при користуванні продуктом, системою чи сервісом (послугою);

ОС - операційна система;

SDK - software Development Kit, набір із засобів розробки, утиліт і документації;

JSON - JavaScript Object Notation, запис об'єктів JavaScript;

MVP - Model-View-Presenter, MVP, Модель–Представлення–Пред'явник;

MVVM - Model-View-ViewModel, Модель–Представлення–Модель Представлення;

iOS - iPhone operating system, мобільна операційна система від Apple;

AWS - Amazon Web Services

API - Application Programming Interface, інтерфейс прикладного програмування;

JVM - Java virtual machine, віртуальна машина Java;

XML - eXtensible Markup Language, розширювана мова розмітки;

БД - база даних;

HTTP - HyperText Transfer Protocol, протокол передачі гіпертексту.

ВСТУП

Оцінка сучасного стану об'єкта дослідження або розробки. Цілодобово все більше і більше відбувається науково-технічний прогрес. Він набрав шалений темп. Кожна людина сучасного суспільства має якийсь електронний пристрій (смартфон, планшет тощо). У XXI столітті люди проводять більше часу в Інтернеті, ніж у живому спілкуванні, в соціальних мережах і віртуальній реальності. Сидячи за електронним пристроєм, є можливість знайти роботу, оплатити рахунки, замовити одяг, їжу, шкільне приладдя, будь-що, і навіть, навчатися.

У зв'язку з пандемією, яка заповонила весь світ, значно виріс відсоток людей, що працюють та навчаються вдома. Поточний епідеміологічний стан країн спонукають до подальших пошуків можливостей застосування ЕОМ у навчальному процесі. Тому зросла потреба в інструментах онлайн-навчання та перевірки знань. На даний момент, комп'ютерне тестування використовується в освітніх закладах таких, як: школи, університети і т. д. Дуже важко перевіряти велику кількість робіт простому вчителю в очний час. Але за допомогою програм, які автоматизують процес опитування, це можна зробити одразу після написання роботи.

У сучасному світі є безліч вебсервісів для проведення опитувань. Кожен додаток по-своєму особливий, має багато вбудованих можливостей. На даний момент, у таких продуктах основними функціями є різні режими відповідей (одна або декілька зі списку, текстова відповідь, відповідь файлом та ін.), можливість анонімного опитування, опитування з обмеженням у часі, вивід результатів на графіку тощо. Прикладом одного з найвідоміших програмних засобів є Google Forms.

Актуальність роботи та підстави для її виконання. Існує думка, що смартфонами користуються тільки молоде покоління. Насправді ж, за статистикою, люди віком від 45 років почали частіше користуватись мобільними додатками. Мобільна розробка активно розвивається, адже кількість мобільних пристроїв є

значно більшою за кількість персональних комп'ютерів (ПК) і ця тенденція буде тільки рости.

У сьогоднішні є чимала кількість сайтів, де можна створити форму контролю знань. Але не всі вони зроблені безпечно. Наприклад, людина, яка добре володіє комп'ютером, може за допомогою своїх здібностей знайти відповіді. Також не всі такі сайти здатні на великі навантаження. Тому почали створювати додатки для перевірки, за допомогою яких, ви зможете опитати понад 1000 людей буквально за декілька хвилин, не боячись, що хтось зміг обхитрити систему.

Будь-хто, навіть учень, може створювати тест для своїх потреб, адже кроки його створення доволі прості. На мою думку, це найбільша перевага програм даного типу. Набагато легше та швидше створити опитування за 5 хвилин, ніж вступити в бесіду з кожним. Також цей додаток можна використовувати в побутових ситуаціях. Наприклад, поїздка з класом (яка вартість поїздки влаштує, у яке місце хотів би поїхати, які умови проживання хотів би), перевірка знань робітника (створення тесту за класифікацією роботи).

Мета й завдання роботи. Метою дипломної роботи є розробка програмного засобу для створення тестів, що можуть допомогти людям у швидкій перевірці знань. Для досягнення цієї мети поставлені такі завдання:

- а) дослідити існуючі сервіси для онлайн-тестів у всесвітній мережі;
- б) розробити технічне завдання до продукту;
- в) розробити дизайн та інтерфейс програмного продукту «Система для проведення онлайн-опитувань та тестувань»;
- г) розробити серверну частину додатку;
- д) розробити клієнтську частину додатку;
- е) здійснити тестування та налагодження;
- ж) поглибити знання в роботі з базами даних;
- з) познайомитись із розробкою мобільних додатків та мовою програмування Kotlin.

Об'єкт, методи й засоби розроблення. Об'єктом розроблення програмного засобу «Система для проведення онлайн-опитувань та тестувань» процес

розроблення сервісу, в функціонал якого входить конструювання тестів із різними можливими варіантами відповідей (одна, декілька, текстова тощо) з можливістю перегляду отриманих відповідей.

У процесі розробки спочатку визначаються функціональні вимоги, концепт продукту, далі відбувається проектування інтерфейсу, визначення дизайну, розробка серверної та клієнтської частини, тестування й налагодження сервісу.

Написання програми відбувалось на інтегрованому середовищі розробки (IDE) для роботи з платформою Android - Android Studio. Проект було розроблено на ОС Windows 10 [5].

Можливі сфери застосування. Програмний продукт націлений на полегшення роботи вчителів під час дистанційного навчання. Крім того, його може використовувати будь-хто. Можна проводити опитування на будь-яку тему, тому програма використовується в різних сферах. Наприклад, офіційна перевірка знань в освітніх закладах, опитування стажерів та робітників для перевірки їхньої кваліфікації тощо.

1 ОГЛЯД ІСНУЮЧИХ НА РИНКУ СИТЕМ

1.1 Google Forms

Google Forms (рисунок 1.1) – один із найпопулярніших сервісів для створення форм для опитування, що входить до веб-пакету Google Docs Editors, пропонуваного Google. [6] У цьому році він неодноразово зазнав деяких змін, які сприяли поліпшенню праці цього додатку. В нього є ряд переваг, а саме:

- а) ця платформа є безоплатною;
- б) є різні види відповідей (одна, декілька з списку, розгорнута текстова відповідь, відповідь файлом тощо);
- в) відповіді та питання можна перемішувати;
- г) можливість мати не обов'язкові питання;
- д) функція призначення кількості балів для відповідей та включення автоматичного оцінювання;
- е) можливість відправляти респондентам відповіді;
- ж) можливість зробити частину перевірки вручну, і тоді надіслати результат на вказану користувачем поштову адресу;
- з) також присутнє автоматичне зберігання інформації у інтерактивну таблицю.

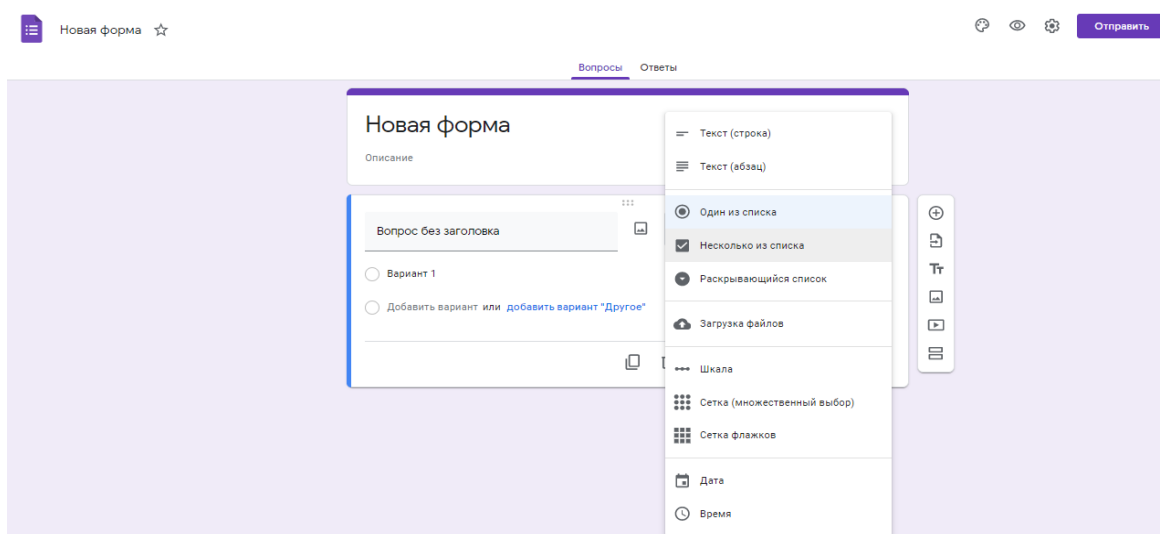


Рисунок 1.1 – Сторінка створення форми для опитування в «Google Forms»

1.2 Quizizz

Quizizz [7] (рисунок 1.2) - додаток, створений для проведення вікторин, які можна розробити на власному гаджеті. Ви можете вчитися самостійно або брати участь у групових вікторинах. Quizizz щомісяця використовують більше 20 мільйонів людей у школах, будинках та офісах по всьому світу. Переваги:

- а) щоб пройти тестування, потрібно всього лиш створити безкоштовний обліковий запис на веб-сайті www.quizizz.com;
- б) є можливість приєднуватися до ігор з будь-якого пристрою без додатка на joinmyquiz.com;
- в) проходження дистанційне (можна проходити з будь якої точки Світу);
- г) є доступ до вже готових вікторин, що охоплюють кожен тему, що теж є безкоштовними;
- д) є можливість кидати виклик своїм друзям та змагатися у рівні знань;
- е) також присутні повні опитування та інтернет навчання;
- ж) можливість брати участь у тренувальних заняттях;
- з) функція отримання даних, щоб побачити те, що ви знаєте зараз, і те, що вам потрібно переглянути;
- и) викладач має повну картину успішності в таблиці Excel.

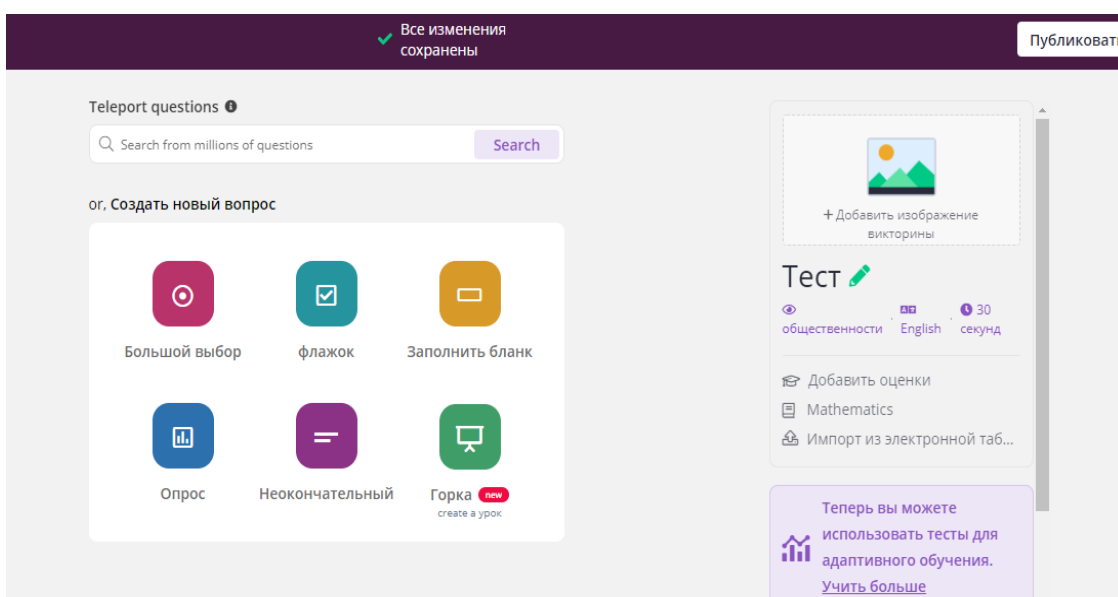


Рисунок 1.2 - Сторінка створення форми для опитування в «Quizizz»

1.3 Surveymonkey

Surveymonkey [8] (рисунок 1.3) – це сервіс для конструювання як простих і невеликих опитувань, так і для масової розсилки. SurveyMonkey є світовим лідером в програмному забезпеченні для проведення опитувань. Переваги:

- а) редагувати та створювати тест можна на будь-якому пристрої: комп'ютері, планшеті і смартфоні;
- б) безкоштовне завантаження та користування додатком;
- в) є можливість оформлення платної підписки, щоб скористатися розширеними можливостями;
- г) надання і заборона доступу до опитувань в вашому профілі;
- д) існує редагування вже створених користувачами опитувань, де ви зможете виправити помилки і додати пропущені запитання;
- е) доступ до опитування через посилання, відправлену в SMS-повідомленні, поштою або в соціальних мережах;
- ж) налаштування спливаючих повідомлень для відстеження результатів.

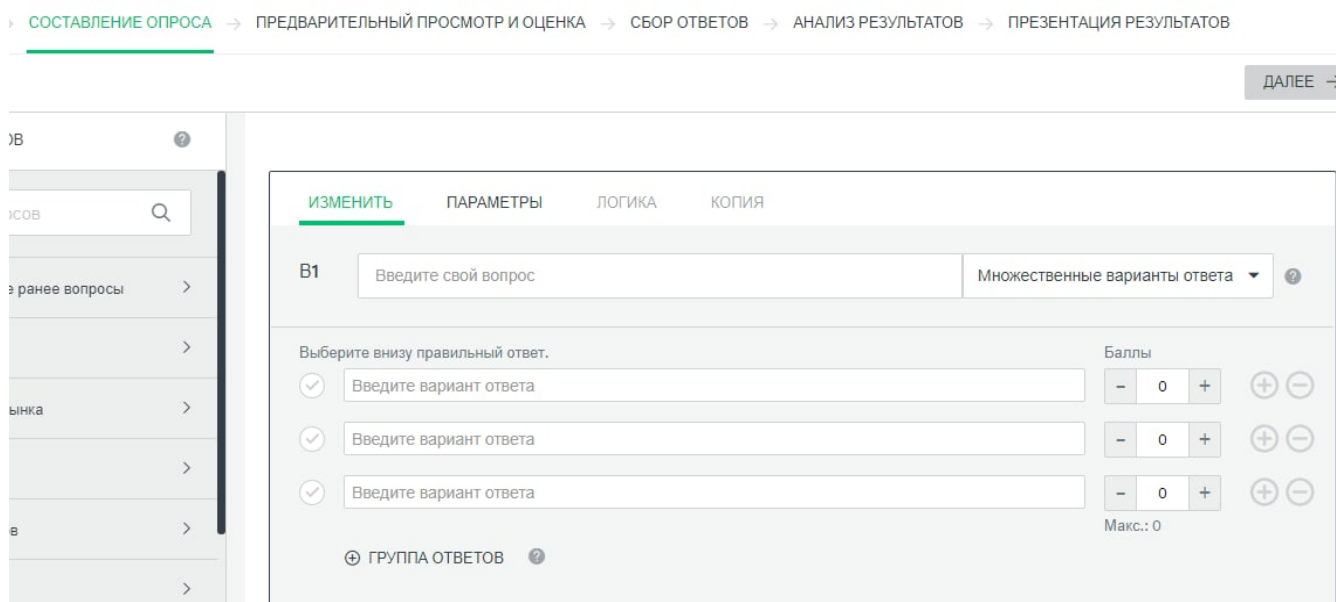


Рисунок 1.3 - Сторінка створення форми для опитування в «Surveymonkey»

2 ОСНОВНІ ЕТАПИ СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ

Процес розробки мобільного додатку, як і будь-якого сервісу, є досить громіздкий та має багато етапів на шляху створення. Нижче наведено основні стадії створення мобільних засобів.

2.1 Ідея

Перш за все, створення застосунку починається з ідеї. Ви чітко маєте знати чого ви хочете і дати відповідь на питання:

- а) з якою метою я збираюсь робити додаток, яка його ціль?
- б) які проблеми я збираюся вирішити за допомогою свого засобу?
- в) чи існують інші програми, які мають такий же функціонал?
- г) чим мій додаток буде кращим інших?
- д) яка цільова аудиторія моєї програми?

Відповідаючи на ці питання ви зможете визначити свої цілі і почати стратегію розвитку в правильному напрямку.

2.2 Побудова стратегії

Значним чином допоможе аналіз існуючих на ринку систем. Вивчаючи аналогічні додатки, ви можете:

- а) визначити потреби користувачів через відгуки, які можна переглянути в описі мобільних додатків;
- б) подивитись будову та форму інших системних програм;
- в) визначити функції, яких їм не вистачає;
- г) побачити додатки, яких більше завантажують на пристрої, дізнатись їх переваги;
- д) подивитись реалізацію певних ідей;
- е) переглянувши інші додатки, у вас є змога створити щось особливе (те, що буде відрізняти вас від інших).

Також важливим є встановити для себе часові рамки. Коли немає чіткого плану виконання, то процес може значно затягнутись.

2.3 Створення попередньої версії продукту

Це перший етап розробки мобільного додатку, коли ваша ідея перетворюється у невеличкий проект. Це не створення повноцінного засобу, але лише його ескіз, спрощений план майбутнього продукту [12]. На цьому етапі не є необхідним робити деталізований дизайн, достатньо зробити просту схематичну конструкцію. Прототип може бути створений за допомогою ручки і паперу, також це може бути прототип коду, який виглядає як повноцінний інтерактивний засіб. При створенні прототипів можна використовувати інструменти з всесвітньої мережі. Завдяки цьому ви можете:

- а) наглядно показати свої уявлення про вигляд програми;
- б) показати навігацію всередині додатку;
- в) краще зрозуміти свій засіб;
- г) випробувати свої ідеї;
- д) впевнитись, що ваше технічне завдання не перевищує ваші можливості та є здійсненним;

2.4 UI / UX дизайн

Народне прислів'я «Зустрічають по одягу» можна застосувати і у випадку мобільного додатку. Розробка дизайну є надзвичайно важливим етапом, він є одним з найнадійніших способів втримати користувачів і, відповідно, від цього може залежати успіх вашого застосунку. Ціль не в тому, щоб зробити яскравий і креативний дизайн для того, щоб додаток виглядав красиво, а в тому, щоб забезпечити приємний для користувача досвід з моменту відкриття додатка до моменту його закриття. Критерії, за якими розробляється дизайн:

- а) легкість – інтерфейс має бути зрозумілим у використанні, який нескладно дослідити без інструкції для користувачів;

- б) сучасний дизайн - хороша стратегія, оскільки ви знаєте, що це те, що людям подобається в даний момент, і що їм буде приємно цим користуватися;
- в) регулярні компоненти користувальницького інтерфейсу - використовуючи базові компоненти, ви можете внести ясність у користуванні вашим програмним засобом своїм клієнтам. Користувачі хочуть швидко знайти в додатку те, що шукають.

2.5 Розробка

Цей етап зазвичай займає найбільше часу. В процесі розробки будуть наступні стадії:

- а) визначення архітектури додатку в залежності від функціоналу, для Android та iOS існують основні типи архітектури програми: MVP, MVVM та ін.;
- б) вибір мови програмування. Додатки для iOS можна розробляти з використанням мови програмування Objective-C або Swift. Додатки Android в основному створюються з використанням Java або Kotlin;
- в) вибір бібліотек, фреймворків і готових рішень, які будуть використовуватися в додатку;
- г) проектування бази даних;
- д) налаштування серверної частини. Основний спосіб - це SaaS в якості серверної частини - наприклад, AWS Mobile Hub, CloudKit, Kinvey, Parse, Rest API для MVP базової програми;
- е) налаштування клієнтської частини.

2.6 Тестування

Тестування є не менш необхідним процесом, він дозволяє команді розробників нарешті побачити, чи відповідає мобільний додаток очікуванням і поставленим цілям. За допомогою цього етапу ви можете зробити свій продукт стабільним, безпечним, перевірити його на зручність в користуванні, його користувацький інтерфейс, виявити помилки, підвищити ймовірність того, що

додаток буде працювати правильно при будь-яких обставинах. Крім того, це зменшує ризик втрати потенційних клієнтів. Є такі види тестування:

- а) тестування установки;
- б) тестування сумісності;
- в) стресове тестування;
- г) тестування навантаження;
- д) функціональне тестування;
- е) тестування локалізації;
- ж) юзабіліті тестування;
- з) автоматизоване тестування [15];

2.7 Розгортання

Для публікації власного мобільного застосунку необхідно відправити ваш засіб до магазинів додатків, Apple App Store для додатків iOS і Google Play для додатків Android. Кожна програма перед публікацією перевіряється командами Google і Apple. Випуск в магазині додатків вимагає підготовки метаданих, в тому числі:

- а) назва вашого додатка;
- б) опис;
- в) категорія;
- г) ключові слова;
- д) значок запуску;
- е) скріншоти.

2.8 Підтримка

На цьому етапі необхідно підтримувати працездатність додатку, в разі необхідності розширювати функціонал. Важливо оновлювати функції відповідно до запитів цільової аудиторії, відслідковувати тенденції, обробляти відгуки користувачів для майбутніх покращень [11]. Оперативна підтримка для кінцевих користувачів і часте внесення виправлень в додаток з поліпшеннями будуть мати життєво необхідне значення для підтримки зацікавленості користувачів.

3 ВИКОРИСТАНІ ПРОГРАМНІ ЗАСОБИ, МОВИ ПРОГРАМУВАННЯ ТА ФРЕЙМВОРКИ

3.1 Android Studio

Android Studio [1] (рисунок 3.1) — інтегроване середовище розробки (IDE) для платформи Android, яке було анонсоване Еллі Паверс, менеджером по продукції корпорації Google, 16 травня 2013 року на конференції Google I/O. Перша стабільна версія Android Studio 1.0 була випущена Google восьмого грудня 2014 року. Середовище створювалось на базі IntelliJ IDEA [9] (комерційне інтегроване середовище розробки для різних мов програмування від компанії JetBrains.).

Особливості:

- а) знаходиться у вільному доступі;
- б) гнучка система збирання на основі Gradle;
- в) рефакторинг коду;
- г) розширений редактор макетів, що дозволяє користувачам працювати з UI компонентами за допомогою drag-and-drop, функція перегляду макетів одночасно на різних конфігураціях екранів.
- д) вбудована підтримка Google Cloud Platform та ін.

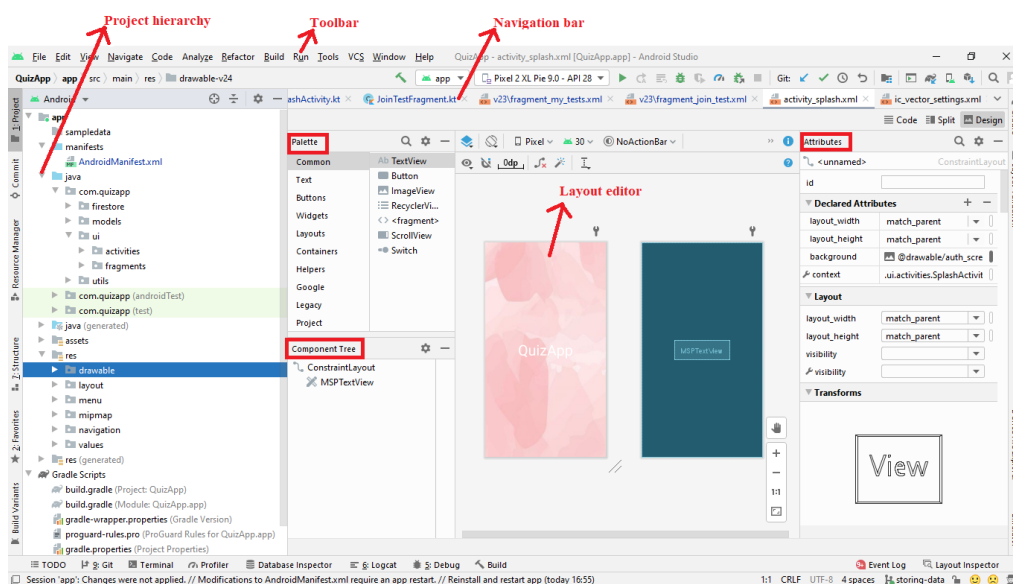


Рисунок 3.1 – Інтерфейс Android Studio

3.2 Kotlin

Kotlin [2] — статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Він підтримує як об'єктно-орієнтовне, так і процедурне програмування. Також компілюється в JavaScript. Мову Kotlin було названо на честь острова Котлін у Фінській затоці, де розміщена частина Кронштадту. Головною метою авторів було створити більш типо-безпечну та лаконічну мову, ніж Java та більш простою, ніж Scala.

Переваги:

- а) вихідний код відкритий, впровадження безкоштовні;
- б) програми застосовують існуючі Java-фреймворки та бібліотеки;
- в) доступні компіляції в байткод JVM, JavaScript;
- г) рівень безпеки високий (наприклад, спроба надати null-значення призводить до помилки. Kotlin покликаний виключити помилки подібного роду з вашого коду);
- д) процес вивчення простий;
- е) перетворення Java-коду в Kotlin і навпаки відбувається автоматично.

Недоліки:

- а) швидкість компіляції, але це не постійна недоробка;
- б) пошук розробників, тому що мова відносно нова;
- в) підтримка - початківцям розробникам доведеться нелегко, адже вирішувати проблеми доведеться поодиночі. У Kotlin невелика спільнота і мало ресурсів для вивчення.

Приклади коду:

```
fun main() {
    val scope = "world"
    println("Hello, $scope!")
}
fun sayHello(maybe: String?, neverNull: Int) {
    // use of elvis operator
    val name: String = maybe ?: "stranger"
    println("Hello $name")
}
```

3.3 Firebase

Firebase [4] - американська компанія, постачальник хмарних послуг, заснована в 2011 році Ендрю Лі і Джеймсом Темпліном, і поглинена в 2014 році корпорацією Google.

Firebase надає детальну документацію і кросплатформені SDK, які допоможуть вам створювати мобільні додатки. На сьогоднішній день її застосовують для створення найкращих додатків на декількох платформах. Особливостями цієї платформи є те, що вона дозволяє відволікати увагу розробників від створення серверних систем (тобто програмних частин проєктів, які приховані від користувачів, наприклад, коду сервера). Це спрощує та пришвидшує створення мобільних додатків, таким чином повністю зосереджуючись на UX / UI (тобто користувацькому інтерфейсі та досвіді).

Firebase - одне з рішень BaaS (серверна частина як послуга), яке пропонує безліч можливостей для розробників.

Це сервер, база даних, хостинг та аутентифікація на платформі. Наприклад, Firebase Realtime Database надає розробникам API, який може синхронізувати дані додатків між клієнтами і зберігати їх в хмарі.

Відвідавши сторінку продукту, ви виявите, що Firebase пропонує три категорії послуг, доступних для клієнтських додатків. (рисунок 3.3.1).



Рисунок 3.3.1 - Будівельні блоки Firebase

3.3.1 Cloud Firestore

Це база даних NoSQL (БД, яка забезпечує механізм видобування та зберігання даних, що відрізняється від підходу таблиць-відношень в реляційних БД), яка дозволяє легко зберігати, синхронізувати та запитувати дані для мобільних та веб-програм.

На відміну від бази даних Firebase Realtime, Cloud Firestore призначений для корпоративного використання, що передбачає масштабованість, складні моделі даних та розширені параметри запиту. Консоль Firebase можна використовувати для перегляду даних в обох базах даних. Ще однією спільною рисою є наявність SDK для роботи з кодом сервера обох баз даних. Вони доступні для Python, Node.js, Golang, Ruby, PHP, Java, .NET та C #.

Можливості Cloud Firestore:

- а) запитувати і легко структурувати дані за допомогою колекцій і документів;
- б) створювати ієрархії для зберігання пов'язаних даних і легко виймати потрібні дані за допомогою запитів;
- в) всі запити масштабуються в залежності від розміру вашого набору результатів (примітка: не набору даних);
- г) Cloud Firestore дозволяє автоматично синхронізувати дані додатків між пристроями.

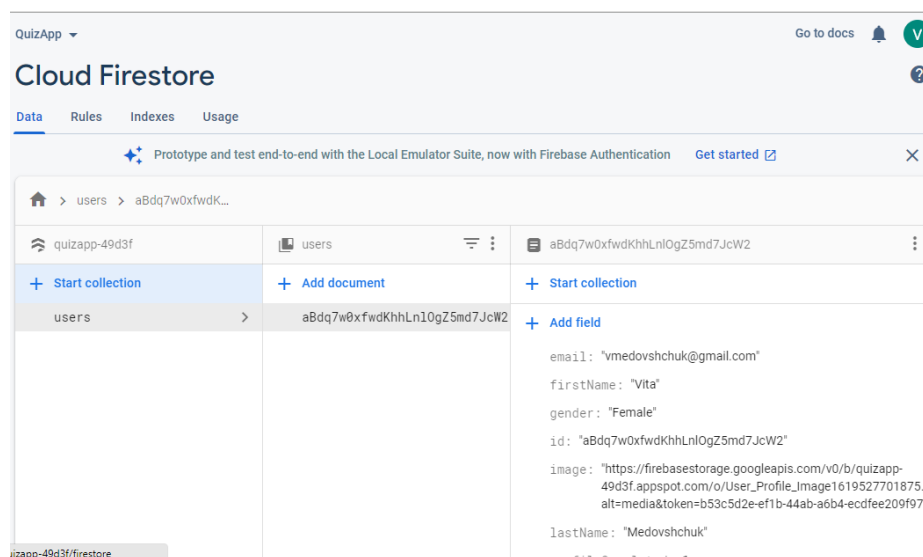


Рисунок 3.3.1 – Консоль Cloud Firestore

3.3.2 Cloud Storage

Cloud Storage – є хмарним сховищем Firebase. Воно є доволі простим у користуванні та допомагає програмістам зберігати та підтримувати контент користувачів (наприклад, фотографії, аудіо чи відео).

Переваги Cloud Storage:

- а) SDK Firebase дозволяє завантажувати файли для програми і при цьому не важливо яка якість мережі. Клієнт може при поганій характеристиці мережі повторити операцію завантаження з місця переривання, тим самим заощаджуючи час і пропускну здатність користувачів;
- б) дані, які зберігаються в Cloud Storage доступні через Google Cloud та Firebase;
- в) найприємнішим є те, що хмарне сховище може масштабуватись автоматично;
- г) можна фільтрувати зображення чи перекодувати відео на стороні сервера;
- д) для доступу до файлів необхідно мати ім'я об'єкта, яке має структуру каталогів (наприклад, /users/vita/profile/photo.jpg), але насправді вони не існують.
- е) щоб забезпечити контроль доступу, SDK Firebase для Cloud Storage інтегрований з автентифікацією Firebase.

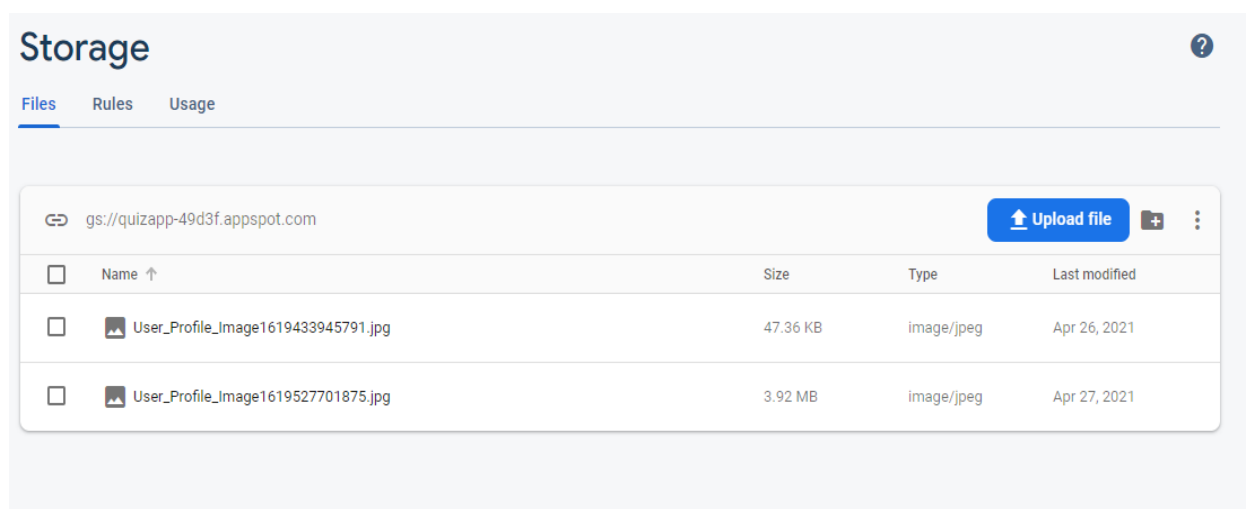


Рисунок 3.3.2 – Консоль Cloud Storage

3.3.3 Firebase Authentication

Дуже часто в мобільних додатках необхідно реєструвати користувача для того, щоб безпечно зберігати їх особисті дані. В моєму додатку я це роблю для того, щоб була можливість створювати власний тест та зберігати їх.

Firebase надає готове рішення для реалізації автентифікації користувачів у вашому додатку. Він дозволяє здійснювати автентифікацію за допомогою постачальників автентифікації (Facebook, Google та ін.) та електронної пошти з паролем. Також Firebase має функцію скидування паролю за допомогою електронної пошти, відправки SMS на телефон для автентифікації, анонімної авторизації.

Приклад коду реєстрації нового користувача:

```
auth.createUserWithEmailAndPassword(email, password)
  .addOnCompleteListener(this) { task ->
    if (task.isSuccessful) {
      // Sign in success, update UI with the signed-in user's information
      Log.d(TAG, "createUserWithEmail:success")
      val user = auth.currentUser
      updateUI(user)
    } else {
      // If sign in fails, display a message to the user.
      Log.w(TAG, "createUserWithEmail:failure", task.exception)
      Toast.makeText(baseContext, "Authentication failed.",
        Toast.LENGTH_SHORT).show()
      updateUI(null)
    }
  }
```

4 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ

4.1. Призначення системи

Даний веб-застосунок призначений для простішого ведення відомостей про успішність респондентів, а саме: здатність створювати власні тести, наявність детальної інформації про результати проходження тестів, збір статистики, внесення даних до БД та автоматизація процесів оцінювання. Мобільний додаток може використовуватись в сфері освіти та спрощувати освіту під час дистанційного навчання. Також його можна застосовувати в інших галузях для перевірки знань. Наприклад, для перевірки правильного перекладу слів англійської мови чи для тесту на знання мови програмування.

4.2 Цілі створення системи

Система була створена з метою:

- а) проаналізувати сучасні застосунки для онлайн-тестів у мережі Інтернет;
- б) навчитись створювати технічне завдання до продукту;
- в) удосконалити навички дизайну програмного продукту;
- г) покращити знання у розробці серверної частини додатку та клієнтської частини;
- д) навчитись тестувати та налагоджувати додаток;
- е) поглибити знання в роботі з базами даних;
- ж) познайомитись із розробкою мобільних додатків;
- з) краще познайомитись з хмарними технологіями;
- и) дослідити платформу Firebase;
- к) навчитись верстати XML представлення для Android;
- л) вивчити мову програмування Kotlin;
- м) створити додаток для створення тестів з метою полегшення дистанційного навчання.

5 ВИМОГИ ДО СИСТЕМИ

5.1 Вимоги до функцій, які виконуються системою

Перелік функцій, які має виконувати мобільний застосунок:

- а) змога реєструвати користувача – введення імені, паролю, електронної пошти, гендеру;
- б) здійснення процесу відновлення паролю, відповідно розсилка інформації на пошту;
- в) можливість входу до власного аккаунту;
- г) редагування даних аккаунту;
- д) можливість створювати тест;
- е) функція редагування тесту;
- ж) здатність переглядати тест;
- з) можливість видаляти тест;
- и) перегляд результатів тесту;
- к) можливість проходити тест.

5.2 Технічні вимоги

Обсяг таблиць БД не регламентується. Перелік полів таблиць БД не є вичерпним. Діалог англomовний. Під час тестування провадиться введення нових даних у таблиці бази даних. Веб-застунок має орієнтуватися на потреби кінцевого користувача (а не програміста). Мають генеруватися відповідні повідомлення про помилки англійською мовою на некоректні дії користувача, пов'язані з введенням невірних даних, не заповненням обов'язкових полів введення в формах та інші, які можуть бути оброблені системою. Мова програмування Kotlin. Застосунок має бути створений для платформи Android. Необхідне застосування хмарних технологій за допомогою Firebase. Операційна система: Windows 10. Мобільний додаток повинен виконувати всі функції наведені вище.

6 ОПИС ОРГАНІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

6.1 Логічна структура бази даних

В своєму проекті я використовувала Cloud Firestore. Ця база даних є документо-орієнтовною базою NoSQL (БД, яка забезпечує механізм видобування та зберігання даних, що відрізняється від підходу таблиць-відношень в реляційних БД), тобто нема ні таблиць, ні рядків [13].

В Cloud Firestore одиницею зберігання є документи - тут зберігаються дані в документах, які організовані в колекції. Кожен документ зберігає набір пар ключ-значення. Документи - це полегшені записи, які містять поля, зіставлені зі значеннями, він ідентифікується унікальним іменем. Також, документам можна надати власні ключі або дозволити автоматично їх назначати. До речі, документи дуже схожі на JSON [10] — це текстовий формат, який використовується для обміну даними між комп'ютерами. JSON базується на тексті, він є легким для читання людиною.

У Cloud Firestore існує оптимізація для зберігання великих колекцій невеликих документів. Колекції являються просто контейнерами для документів. В документи можуть бути вкладені об'єкти та вкладені колекції. Обидва з них можуть включати примітивні поля (наприклад, рядки) або складні об'єкти (наприклад, списки).

Ви можете вибрати будь-які поля для розміщення в кожному документі та будь-які типи даних, що зберігаються в цих полях, оскільки у Cloud Firestore немає схеми. В документах однієї колекції можуть міститись різні поля або зберігатись різні типи даних у цих полях. Але це не варто робити для простоти використання. Немає необхідності колекції видаляти чи створювати. При створенні першого документу в колекції – колекція існує, при видаленні всіх документів – вона більше не існує.

Типи даних у полях: Array, Boolean, Bytes, Date and time, Floating-point number, Geographical point, Integer, Map, Null, Reference, Text string.

Нижче наведено приклад колекції users з документом (рисунок 6.1), які містить поля про дані користувача, а саме:

- а) електронна пошта;
- б) ім'я;
- в) прізвище;
- г) стать;
- д) картинка профіля;
- е) індикатор заповнення профілю.

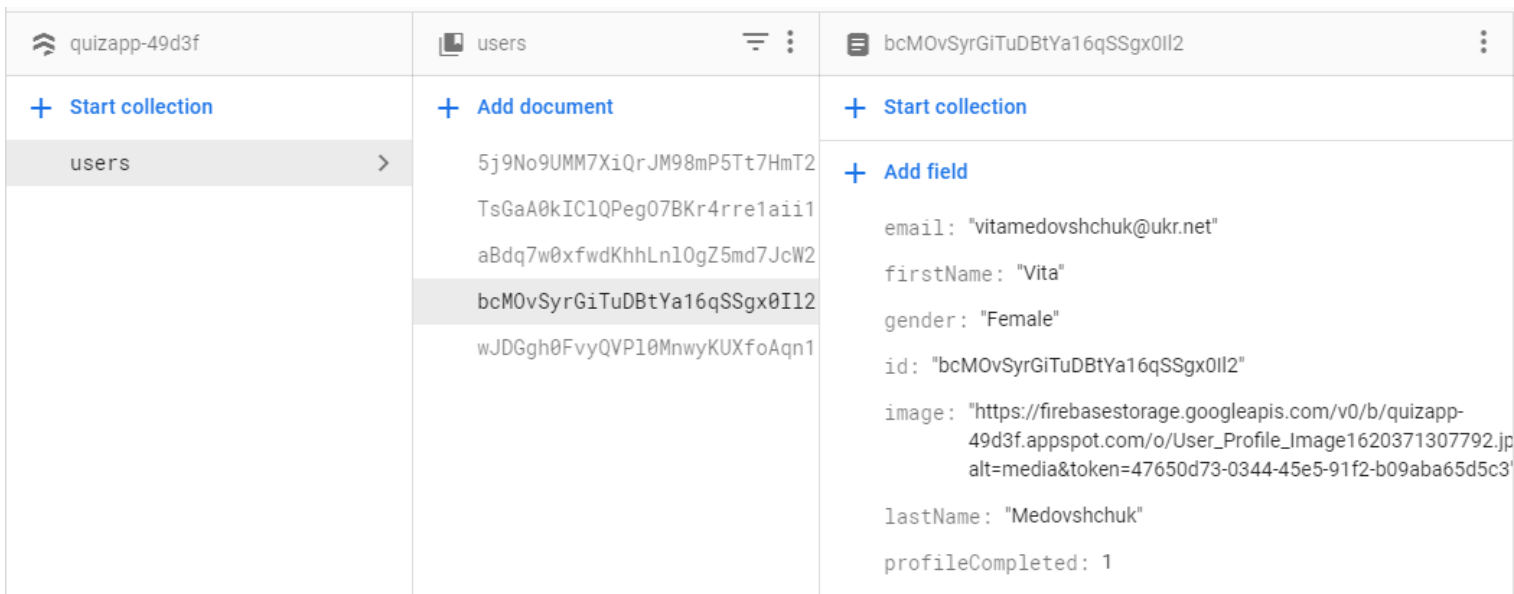


Рисунок 6.1 – Колекція Users

7 ІНСТРУКЦІЯ КОРИСТУВАЧА

7.1 Іконка та екран-заставка

Перше, на що звертають увагу користувачі при встановленні додатку, це іконка (рисунок 7.1). Бажано її зробити під свій продукт, таку, щоб виділялась серед інших, а не залишити стандартну. В Android Studio є вбудовані засоби для створення значків – Image Asset.

Додаток було названо «QuizApp» - від англійських слів «quiz» - вікторина, «app» - додаток.

При запуску програми з'являється екран-заставка (рисунок 7.1), яка вмикається на час - 2 секунди. Це ввідна сторінка в додатку. Вона займає весь екран, має на фоні картинку та у центрі назва засобу. В логотипі можна додати ще версію додатку.

Для всього додатку визначений єдиний шрифт - Montserrat-Bold. Використання єдиного шрифту надає вашому додатку привабливості та дизайн стає не таким їдким. Майже весь текст в мобільному засобі є білим. Кольорова гамма додатку – відтінки рожевого, білий.



Рисунок 7.1 – QuizApp: а – іконка QuizApp; б – екран-заставка QuizApp

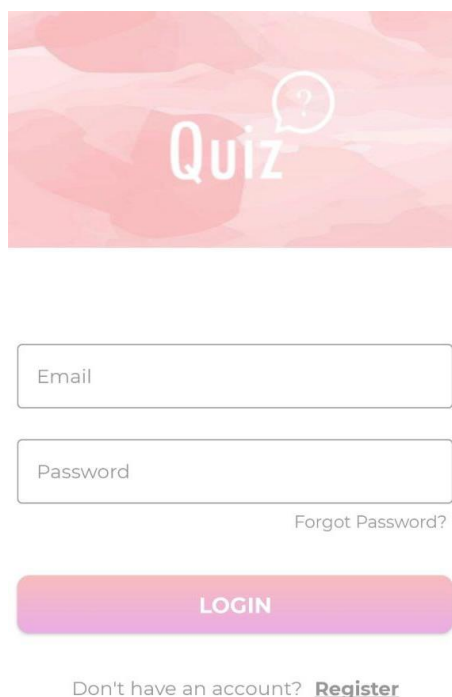
7.2 Сторінка входу

Далі при запуску додатку, після екрану-заставки, йде сторінка логіну (рисунок 7.2) – входу в свій профіль. Верхню частину займає картинка, той же фон, що і на екрані-заставці, посередині логотип додатку білого кольору. Далі зображені два поля вводу. Перше поле для електронної пошти, друге – для паролю.

Перш за все, для користування засобом, необхідно мати власний аккаунт. Якщо електронну пошту ввести не відповідну загальноприйнятому стандарту (логін, собачка «@», домен), то виведеться помилка, про невалідну електронну пошту. Якщо ви не зареєстрований користувач і введете неіснуючі дані, то з'явиться помилка про те, що такого користувача не існує. Якщо помилитесь в написанні логіну чи паролю, то буде аналогічне повідомлення. Якщо не введете в якесь поле дані, то буде помилка про необхідність введення всіх даних

Також на сторінці логіну є клікабельні два посилання, за допомогою яких можна перейти на сторінку реєстрації чи сторінку відновлення паролю.

Всі кнопки однаково стилізовані. Вони мають лінійний окрас, заокруглені кути та білий колір тексту.



The image shows a login page for an application named 'Quiz'. At the top, there is a header with the word 'Quiz' in a white, sans-serif font, accompanied by a speech bubble icon containing a question mark. Below the header, there are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. To the right of the password field, there is a link that says 'Forgot Password?'. Below these fields is a large, rounded rectangular button with a pink-to-purple gradient, labeled 'LOGIN'. At the bottom of the page, there is a link that says 'Don't have an account? Register'.

Рисунок 7.2 – Сторінка входу

7.3 Сторінка реєстрації

При натисканні на посилання «Register» на сторінці входу, йде перехід на сторінку створення аккаунту (рисунок 7.3).

На ній присутні такі поля, як: ім'я, прізвище, електронна пошта, пароль, повторне введення паролю та погодження з умовами користування. Крім того, на цій сторінці є посилання «Login» на випадок, якщо ви вже зареєстрований користувач.

Перелік помилок (відповідний код у додатку А):

- а) програма не дозволить реєстрацію, якщо хоча б одне поле буде порожнє;
- б) в полі електронної пошти йде перевірка пошти згідно міжнародного стандарту її написання (ім'я_користувача@ім'я_домена);
- в) пароль має складатись не менше, ніж з 6 символів;
- г) повторний пароль має збігатись з попереднім.

Після натискання кнопки «Register» при правильному заповненні всіх полей користувача перенаправляють на сторінку логіну та видається повідомлення про успішну реєстрацію (додаток А).

< CREATE AN ACCOUNT

First name

Last name

Email

Password

Confirm Password

I Agree To The [Terms & Condition](#).

REGISTER

Already have an account? [Login](#)

Рисунок 7.3 – Сторінка реєстрації

7.4 Сторінка відновлення паролю

При натисканні на посилання «Forgot Password?» на сторінці входу, йде перехід на сторінку відновлення паролю (рисунок 7.4).

На цій сторінці присутнє лише поле введення електронної пошти.

Як і на інших сторінках присутні:

- а) перевірка наявності введення пошти;
- б) перевірка правильного введення пошти;
- в) перевірка існування користувача з такою поштою.

Після того, як користувач ввів свою пошту для відновлення паролю, йому висвічується повідомлення про те, що на електронну пошту було надіслано лист з посиланням для зміни паролю (додаток А).

Для відновлення паролю необхідно: відкрити свою електронну пошту, знайти відповідний лист, перейти за посиланням – відкриється нова сторінка (рисунок 7.4), ввести новий пароль.

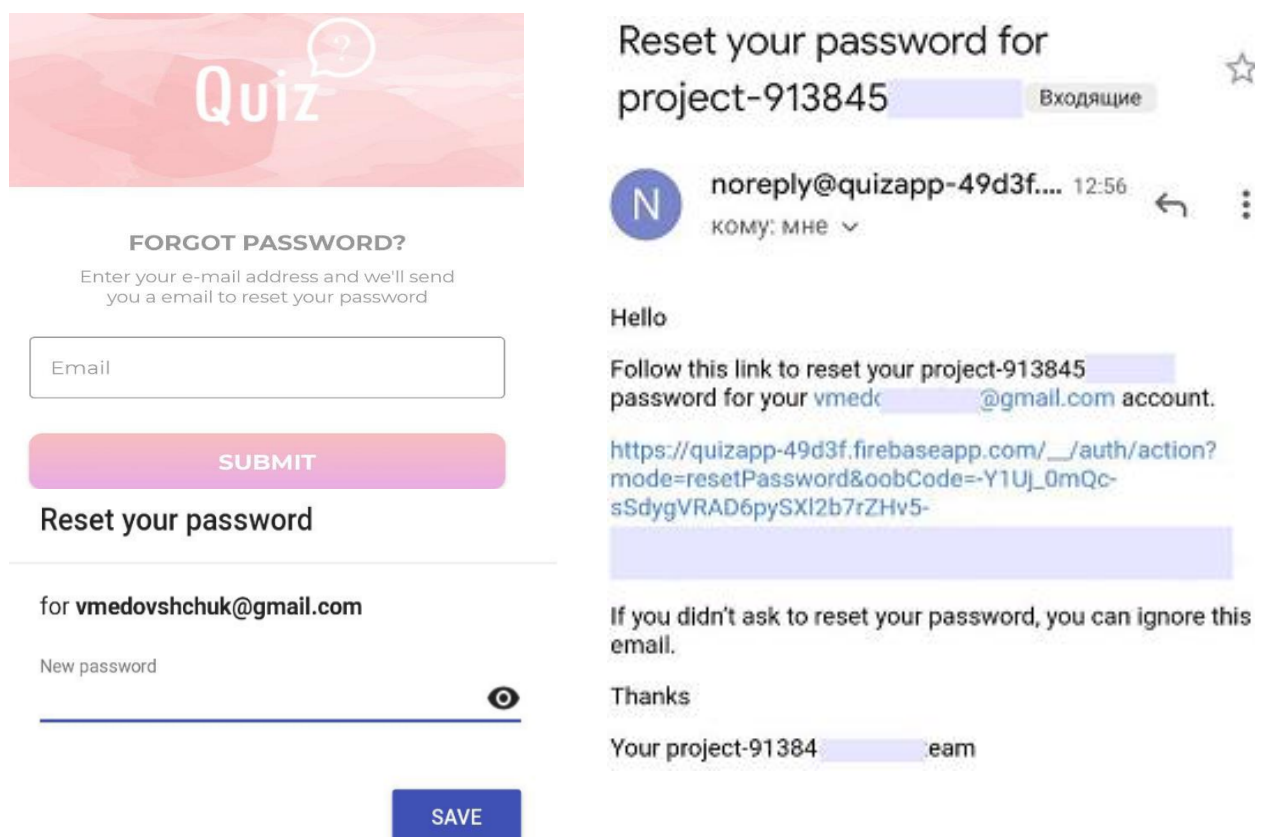


Рисунок 7.4 – QuizzApp: а – сторінка відновлення паролю в додатку; б – лист про відновлення паролю; в – сторінка відновлення паролю в браузері

7.5 Сторінка доповнення профілю

Після першого входу в профіль після реєстрації з'являється сторінка доповнення профілю (рисунок 7.5).

На сторінці зображено:

- а) форма для вставки рисунка для профілю – при натисканні йде запит на дозвіл до даних користувача і далі можна вибрати з галереї фотографію;
- б) ім'я, яке можна змінювати;
- в) прізвище, також можна змінювати;
- г) електронна пошта – недоступна для змін;
- д) радіокнопка – елемент графічного інтерфейсу, де можна вибрати тільки одну опцію (гендер).

Після того, як всі поля будуть заповнені, дані можна зберегти та перейти до головного меню мобільного засобу (додаток А).

COMPLETE PROFILE

First name
Vita

Last name
Medovshchuk

Email
vitamedovshchuk@ukr.net

MALE FEMALE

SAVE

Рисунок 7.5 – Сторінка доповнення профілю

7.6 Сторінка даних профілю

Після того, як користувач зайшов в свій профіль, в додатку зображено три вкладки: «My Tests», «Join Test», «Create Test».

На сторінці «My Tests» в верхньому правому куті зображена шестерня – перехід до даних свого профілю (рисунок 7.6).

На ній зображені такі дані:

- а) картинка;
- б) ім'я та прізвище;
- в) гендер;
- г) електронна пошта.

Також є можливість редагування профілю – кнопка «Edit». Натискаючи на цю сторінку, можна перейти на сторінку корегування профілю. В неї схожий функціонал на сторінку доповнення профілю (рисунок 7.6).

Крім того, є функцію виходу користувача з профілю – кнопка «LOGOUT».

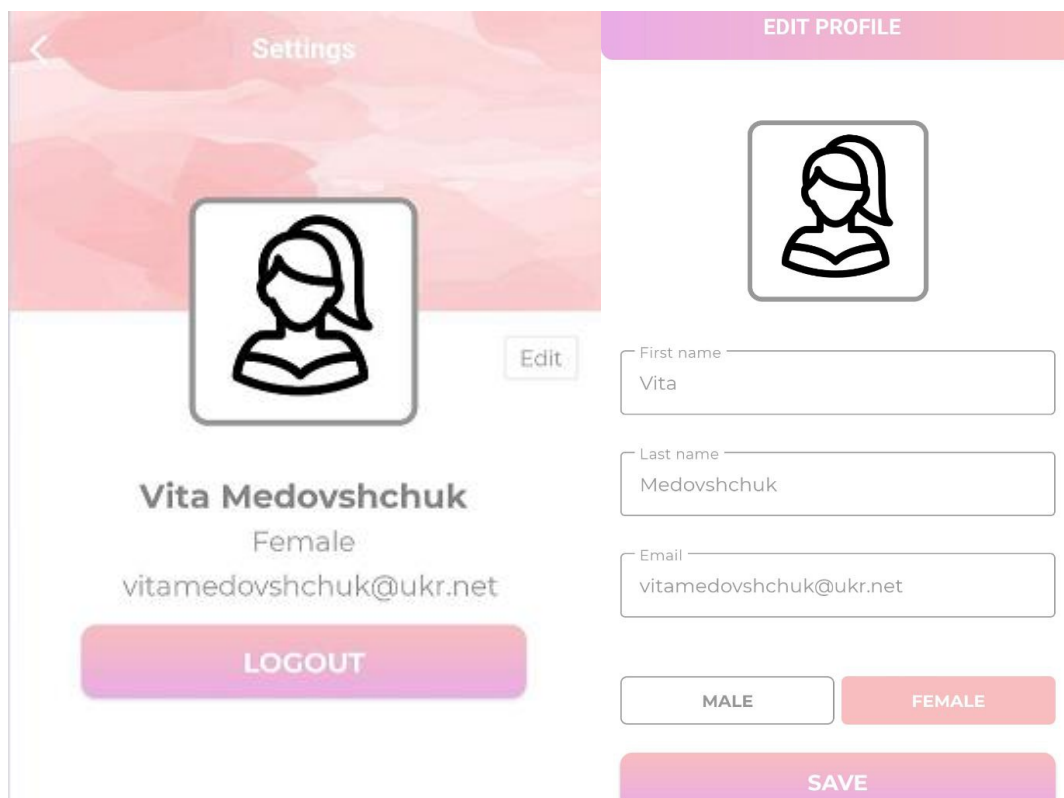


Рисунок 7.6 – QuizApp: а – сторінка даних користувача; б – сторінка редагування профілю

7.7 Сторінка приєднання до тесту

Якщо перейти на сторінку «Join Test», то відкриється вкладка, де буде можливість пройти тест (рисунок 7.7).

Це ввідна сторінка для проходження тесту. На цій вкладці зображене віконце з тінню навкруги «Welcome» з проханням ввести ваше повне ім'я та ідентифікатор тесту. Відповідно в віконці є поле введення повного імені та поле введення ідентифікатора тесту. Нижче, під полями, зображена кнопка з градієнтним забарвленням «Start».

Помилки при користуванні, які можуть з'явитись:

- а) не введене ім'я;
- б) не введений ідентифікатор тесту;
- в) введений неіснуючий ідентифікатор тесту.

При правильно заповнених даних користувач натискаючи на кнопку «Start» переходить на сторінку тесту, де може вже проходити його, відповідаючи на питання.

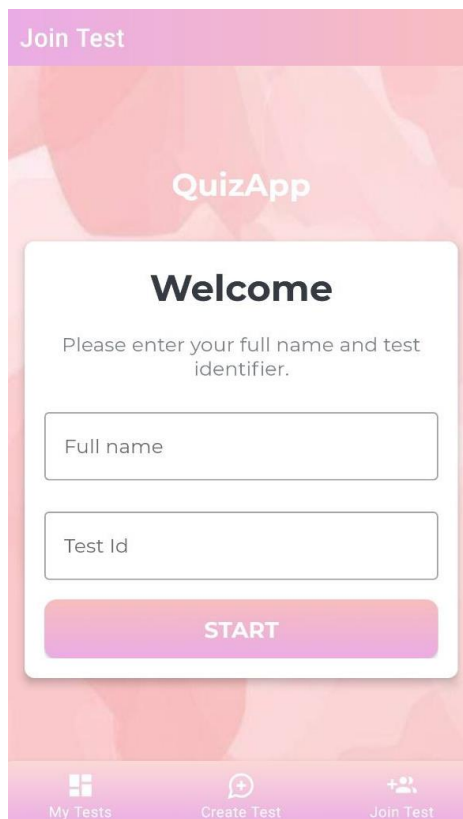


Рисунок 7.7 – Сторінка приєднання до тесту

7.8 Сторінка проходження тесту

На рисунку 7.8 зображене одне з питань, які попадаються в тесті.

Проходження тесту (додаток А):

- а) питання випадають один за одним;
- б) при виборі якогось питання, воно виділяється рожевою рамкою, а текст стає жирним;
- в) в кожному питанні зображений індикатор процесу – яке по рахунку питання та скільки вже пройдено;
- г) при натисканні на кнопку «Submit» виділяються правильні відповіді зеленим, а неправильні відповідно червоним;
- д) після відповіді необхідно натиснути на кнопку «Go to next questions»;
- е) в питанні може бути присутня картинка;
- ж) відповідь може бути не одна.

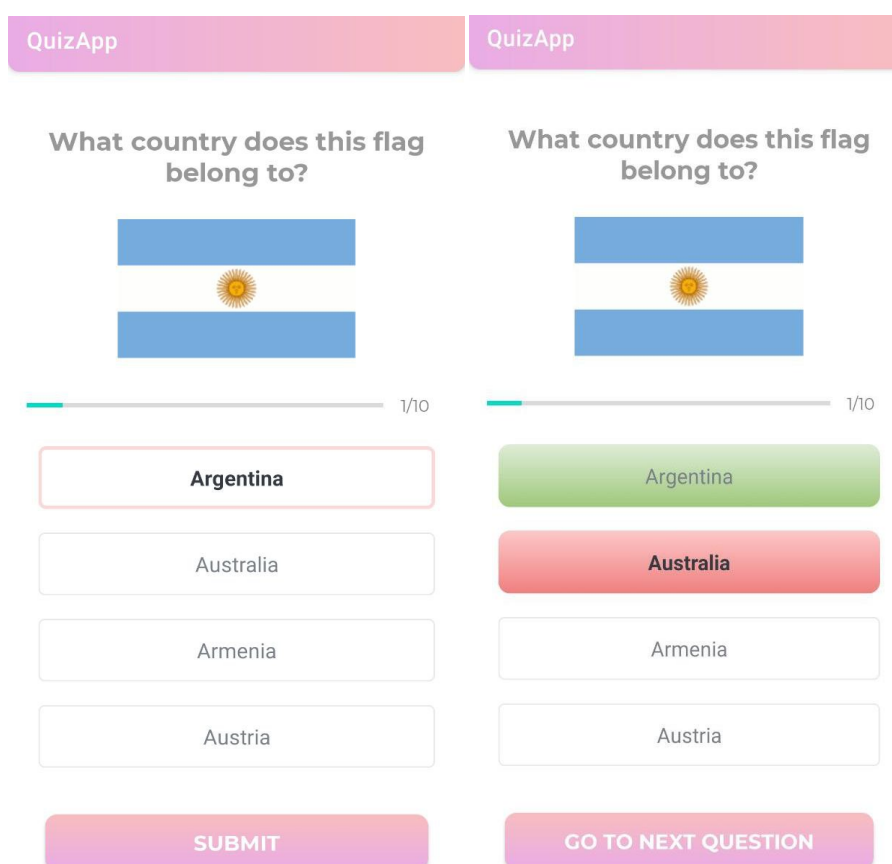


Рисунок 7.8 - QuizApp: а – Вибір варіанту до натискання кнопки «Submit»; б – Сторінка зображення правильності відповіді

7.9 Результат тесту

На останньому запитанні необхідно замість кнопки «Go to next question» випадає кнопка «Finish», на яку необхідно натиснути для того, щоб побачити свій результат. Після цього вмикається нова сторінка – сторінка твого результату за весь пройдений тест. На ній зображений трофей, вітання з проходженням тесту та оцінка, яку ти отримав – певна кількість балів від максимальної. Програма автоматично підраховує правильні відповіді за кожне питання в тесті та виводить її на екран.

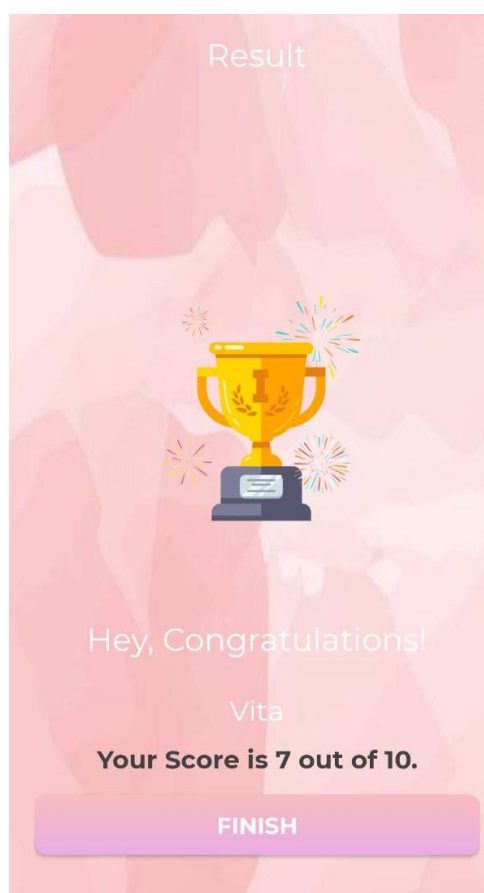


Рисунок 7.9 – Результат проходження тесту

7.10 Сторінка з власними тестами

Якщо перейти на вкладку «My Tests», то на ній буде зображено всі тести, які створив користувач (рисунок 7.10). Крім того, в верхньому правому куті буде шестерня – перехід до налаштувань профілю, про яке написано вище.

На цій сторінці можна переглядати свій тест, редагувати і видаляти

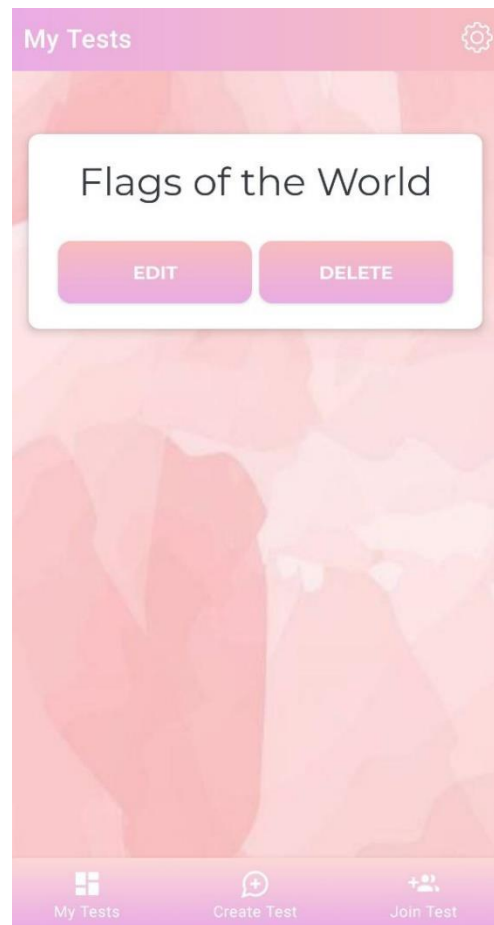


Рисунок 7.10 – Сторінка власних тестів

7.11 Сторінка створення нового питання

Наступною сторінкою, яку ми розглянемо, це сторінка створення нового питання (рисунок 7.11).

За допомогою цієї вкладки, можна легко додати нове питання в власний тест. Зверху присутня можливість додавання при необхідності нової картинки. При додаванні картинки, додаток запитує дозвіл до даних користувача на пристрої, інакше картинку додати заборонено. Далі обов'язково потрібно ввести текст питання. Також є можливість вибрати варіант відповідей – одна чи декілька. Після цього користувач має ввести по черзі можливі відповіді на питання. За допомогою значка «+» можна додати ще одну відповідь. По закінченню введення відповідей необхідно натиснути на кнопку «Submit».

< ADD QUESTION

+

Question

Single Select Multiple Select

Option 1

Option 2

+

SUBMIT

Рисунок 7.11 – Сторінка створення питання

ВИСНОВОК

На сьогоднішній день у зв'язку з пандемією коронавірусу, яка наводить жах на людей у кожній країні, в світі стала популярною онлайн освіта. Відповідно зріс попит на допоміжні програми пов'язані з дистанційним навчанням, які дозволять вчителям проводити заняття та перевіряти отримані навички учнів. Тому це мене нашттовхнуло на ідею створення програмного засобу, який здатний автоматизувати процес оцінювання знань людей з метою зменшення навантаження роботи вчителів.

В результаті роботи було створено програмне забезпечення за рахунок сучасних технологій для легкого та комфортного перебігу опитування, в яке входять наступні процеси:

- а) створення тестів;
- б) редагування тестів;
- в) видалення тестів;
- г) проходження тестів;
- д) перегляд тестів;
- е) перегляд результатів проходження тестів.

Крім того, для такої функціональності було створено опції, пов'язані з профілем, а саме: реєстрація, вхід в аккаунт, відновлення паролю, редагування профілю.

Додаток доступний для Android платформи в безкоштовному варіанті. Для створення тестувань потрібно прикласти мінімум зусиль, які зможе зробити кожен користувач смартфоном. Цей додаток допоможе простому вчителю, студенту, учню та, в принципі, будь-кому створити швидко опитування для власних потреб. Для проходження потрібно ввести ідентифікатор тесту. Власник може бачити кількість людей, що зацікавились тестом та їх результати, зокрема помилки, які було допущено при проходженні. Прикладами програм схожих на мою тест-

програму були Google Forms, Quizizz, SurveyMonkey. Під час розробки додатку було використано такі засоби:

- а) Android Studio;
- б) Kotlin;
- в) Cloud Firestore;
- г) Firebase Authentication;
- д) Firebase Storage.

Однією з найголовніших цілей було ознайомитись з мобільною розробкою додатків для подальшого розвитку в сфері інформаційних технологій. Мобільна розробка була популярною ще до пандемії, але в теперешній час вона стала ще більш поширеною, адже людям необхідні мобільні засоби для щоденного користування, наприклад: замовити продукти з магазину, для того щоб зменшити ймовірність зараження, викликати таксі, купити одяг з тією ж метою.

Мобільні розробники зазвичай поділяються на дві групи: Android-розробники та iOS-розробники, в залежності від програмного забезпечення [14]. Для того, щоб бути ним необхідно мати ряд навичок: знати основи об'єктно-орієнтованого програмування, знати мову програмування (Java та Kotlin є найпоширенішими для розробки мобільних додатків), вміти користуватись середовищем Android Studio, вміти користуватись Git та Github, знати загальні принципи розробки - як оформляти код, правильно називати функції тощо, знати принципи роботи REST та HTTP, вміти працювати з бібліотеками готових рішень та фреймворками, знати принципи роботи операційної системи Android, файлової структуру та багато іншого.

Даний мобільний застосунок звичайно не є досконалим, оскільки це був для мене перший досвід. Але впродовж розробки я багато чого дізналась та багато чому навчилась.

Застосунок можна ще значно розширювати:

- а) зробити тести, які можуть поширюватися тільки на певний круг користувачів;
- б) отримувати доступ до тесту за допомогою посилання;

- в) ввести формат відповіді «сітка прапорців»;
- г) можливість вставляти в відповідь файли;
- д) зробити обов'язкові та необов'язкові питання;
- е) можливість додати відео до питання.

Оскільки мобільна розробка зацікавила мене, я на цьому не збираюся зупинятись та продовжу поглиблювати знання в цій сфері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Android Studio [Електронний ресурс] - Режим доступу до ресурсу:
<https://android-developers.googleblog.com/2014/12/android-studio-10.html>
2. Kotlin [Електронний ресурс] - Режим доступу до ресурсу:
<https://kotlinlang.org/docs/home.html>
3. Cloud Firestore [Електронний ресурс] - Режим доступу до ресурсу:
<https://firebase.google.com/docs/firestore>
4. Firebase [Електронний ресурс] - Режим доступу до ресурсу:
<https://firebase.google.com/docs>
5. ОС Windows 10 [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.microsoft.com/en-us/windows/features?activetab=NewPopular>
6. Google Forms [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.google.com/forms/about/>
7. Quizizz [Електронний ресурс] - Режим доступу до ресурсу:
<https://quizizz.com/join>
8. Surveymonkey [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.surveymonkey.com/>
9. IntelliJ IDEA [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.jetbrains.com/idea/>
10. JSON [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.json.org/json-en.html>
11. The 4 stages of app development [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.businessofapps.com/insights/stages-of-app-development/#:~:text=Pocket%20App%20breaks%20down%20the,project%20from%20beginning%20to%20end>
12. What Is The Mobile App Development Process Steps? [Електронний ресурс] - Режим доступу до ресурсу:

<https://clockwise.software/blog/mobile-app-development-process/>

13. Модель даних Cloud Firestore [Електронний ресурс] - Режим доступу до ресурсу:

<https://firebase.google.com/docs/firestore/data-model>

14. Розробка мобільних додатків від А до Я [Електронний ресурс] - Режим доступу до ресурсу:

<https://dan-it.com.ua/uk/rozrobka-mobilnih-dodatki-vid-a-do-ja-povnij-gajd/>

15. Дервянченко О.В., Плакидюк В.О. Система автоматичного тестування програм під ОС Android // Штучний Інтелект – № 3. – 2013. – С.442-453.

ДОДАТОК А

Лістинг коду деяких функцій в додатку

```
private fun validateRegisterDetails(): Boolean {  
  
    return when {  
        TextUtils.isEmpty(et_first_name.text.toString().trim { it <= ' ' }) -> {  
            showErrorSnackBar("Please enter first name.", errorMessage: true)  
            false  
        }  
  
        TextUtils.isEmpty(et_last_name.text.toString().trim { it <= ' ' }) -> {  
            showErrorSnackBar("Please enter last name.", errorMessage: true)  
            false  
        }  
  
        TextUtils.isEmpty(et_email.text.toString().trim { it <= ' ' }) -> {  
            showErrorSnackBar("Please enter email.", errorMessage: true)  
            false  
        }  
  
        TextUtils.isEmpty(et_password.text.toString().trim { it <= ' ' }) -> {  
            showErrorSnackBar("Please enter password.", errorMessage: true)  
            false  
        }  
  
        TextUtils.isEmpty(et_confirm_password.text.toString().trim { it <= ' ' }) -> {  
            showErrorSnackBar("Please enter confirm password.", errorMessage: true)  
            false  
        }  
  
        et_password.text.toString().trim { it <= ' ' } != et_confirm_password.text.toString()  
            .trim { it <= ' ' } -> {  
            showErrorSnackBar("Password and confirm password does not match.", errorMessage: true)  
            false  
        }  
  
        !cb_terms_and_condition.isChecked -> {  
            showErrorSnackBar("Please agree terms and condition.", errorMessage: true)  
            false  
        }  
  
        else -> {  
            true  
        }  
    }  
}
```

```

private fun registerUser() {
    // Check with validate function if the entries are valid or not.
    if (validateRegisterDetails()) {
        showProgressDialog("Please Wait...")
        val email: String = et_email.text.toString().trim { it <= ' ' }
        val password: String = et_password.text.toString().trim { it <= ' ' }
        // Create an instance and create a register a user with email and password.
        FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener {
                OnCompleteListener<AuthResult> { task ->
                    // If the registration is successfully done
                    if (task.isSuccessful) {

                        // Firebase registered user
                        val firebaseUser: FirebaseUser = task.result!!.user!!
                        val user = User(
                            firebaseUser.uid,
                            et_first_name.text.toString().trim { it <= ' ' },
                            et_last_name.text.toString().trim { it <= ' ' },
                            et_email.text.toString().trim { it <= ' ' }
                        )
                        FirestoreClass().registerUser( activity: this@RegisterActivity, user)

                    } else {
                        hideProgressDialog()
                        // If the registering is not successful then show error message.
                        showErrorSnackBar(task.exception!!.message.toString(), errorMessage: true)
                    }
                }
            }
    }
}

private fun loginRegisteredUser() {

    if(validateLoginDetails()){
        showProgressDialog("Please Wait...")
        val email: String = et_email.text.toString().trim { it <= ' ' }
        val password: String = et_password.text.toString().trim { it <= ' ' }

        FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(
                OnCompleteListener<AuthResult> { task ->

                    if (task.isSuccessful) {

                        FirestoreClass().getUserDetails( activity: this@LoginActivity)

                    } else {
                        hideProgressDialog()

                        // If the registering is not successful then show error message.
                        showErrorSnackBar(task.exception!!.message.toString(), errorMessage: true)
                    }
                }
            )
    }
}
}

```

```

private fun updateUserProfileDetails() {
    val userHashMap = HashMap<String, Any>()
    // Here the field which are not editable needs no update. So, we will update user Gender
    val firstName = et_first_name.text.toString().trim { it <= ' ' }
    if (firstName != mUserDetails.firstName) {
        userHashMap[Constants.FIRST_NAME] = firstName
    }
    val lastName = et_last_name.text.toString().trim { it <= ' ' }
    if (lastName != mUserDetails.lastName) {
        userHashMap[Constants.LAST_NAME] = lastName
    }
    val gender = if (rb_male.isChecked) {
        Constants.MALE
    } else {
        Constants.FEMALE
    }
    // Now update the profile image field if the image URL is not empty.
    if (mUserProfileImageURL.isNotEmpty()) {
        userHashMap[Constants.IMAGE] = mUserProfileImageURL
    }
    if (gender.isNotEmpty() && gender != mUserDetails.gender) {
        userHashMap[Constants.GENDER] = gender
    }
    userHashMap[Constants.COMPLETE_PROFILE] = 1
    // call the registerUser function of Firestore class to make an entry in the database.
    FirestoreClass().updateUserProfileData(
        activity: this@UserProfileActivity,
        userHashMap
    )
    btn_submit.setOnClickListener() { it: View!
        val email: String = et_email_forgot_pw.text.toString().trim { it <= ' ' }
        if(email.isEmpty()) {
            showErrorSnackBar("Please enter email.", errorMessage: true)
        }
        else {
            showProgressDialog("Please Wait...")
            FirebaseAuth.getInstance().sendPasswordResetEmail(email)
                .addOnCompleteListener {task ->
                    hideProgressDialog()
                    if(task.isSuccessful) {
                        Toast.makeText(
                            context: this@ForgotPasswordActivity,
                            "Email sent successfully to reset your password!",
                            Toast.LENGTH_SHORT).show()
                        finish()
                    } else{
                        showErrorSnackBar(task.exception!!.message.toString(), errorMessage: true)
                    }
                }
        }
    }
}

```

```

R.id.btn_submit -> {

    if (mSelectedOptionPosition == 0) {

        mCurrentPosition++

        when {

            mCurrentPosition <= mQuestionsList!!.size -> {

                setQuestion()
            }
            else -> {

                val intent =
                    Intent( packageContext: this@QuizQuestionsActivity, ResultActivity::class.java)
                intent.putExtra(Constants.USER_NAME, mUserName)
                intent.putExtra(Constants.CORRECT_ANSWERS, mCorrectAnswers)
                intent.putExtra(Constants.TOTAL_QUESTIONS, mQuestionsList!!.size)
                startActivity(intent)
                finish()
            }
        }
    } else {

        val question = mQuestionsList?.get(mCurrentPosition - 1)

        // This is to check if the answer is wrong
        if (question!!.correctAnswer != mSelectedOptionPosition) {
            answerView(mSelectedOptionPosition, R.drawable.wrong_option_border_bg)
        } else {
            mCorrectAnswers++
        }

        // This is for correct answer
        answerView(question.correctAnswer, R.drawable.correct_option_border_bg)

        if (mCurrentPosition == mQuestionsList!!.size) {
            btn_submit.text = "FINISH"
        } else {
            btn_submit.text = "GO TO NEXT QUESTION"
        }

        mSelectedOptionPosition = 0
    }
}
}

```