**MINI PROJECT ON**

**BrainiVerseQuizApp**

***Submitted by***

**Aditya Dadwal**

# UID- 24MCA20146

***Subject- Mobile Application Developmrnt***

**MASTER OF COMPUTER APPLICATIONS BONAFIDE CERTIFICATE**

Certified that this project report **“ BrainiVerseQuizApp ”** is the bonafide work of “ Aditya Dadwal **”** who carried out the project work under my/our supervision.

**SIGNATURE SIGNATURE**

**Dr. Abdullah Mrs Ruchika Devi**

**(HOD)**

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# INDEX

Introduction................................................................................................. 1

Objective ..................................................................................................... 1

Features ......................................................................................................1

Conclusion ................................................................................................. 2

## Project Title: BrainiVerseQuizApp

Mobile applications have become a vital part of our lives, transforming smartphones into powerful tools. One engaging use case is mobile-based quizzes, which offer users interactive learning and self-assessment experiences. This mini project demonstrates the development of a **Quiz App** using **Android Studio** with **Kotlin**, focusing on simplicity, interactivity, and user engagement.

Project Outline:

### Objective

The primary goal of this project is to:

* Understand the fundamentals of Android application development.
* Develop a simple, user-friendly quiz app.
* Demonstrate screen navigation, UI design, and data handling in Kotlin.
* Encourage learning through interactive quizzes.

### Features

1. The key features of the Quiz App include:

* User login and registration screen.
* Three quiz categories: **Reasoning**, **Web Development**, and **Computer Science**.
* Each quiz contains 10 multiple-choice questions.
* Auto-switching of questions with a timer (for one category).
* Display of total score at the end.
* Attractive UI with background images and intuitive navigation.

Project Details and Shell Script Implementation:

###  4.1 Tools and Technologies

* **IDE**: Android Studio
* **Language**: Kotlin
* **UI**: XML Layouts
* **Minimum SDK**: API Level 21 (Android 5.0)

## 4.2 App Architecture

* **MainActivity**: Launcher screen with navigation.
* **LoginActivity**: Login or Register.
* **QuizActivity**: Handles quiz questions and options.
* **ResultActivity**: Displays the final score.

## 4.3 Data Handling

* Questions are stored in Kotlin data classes or local arrays.
* No backend/database; the app works offline.

### Script Code Implementation

// LoginActivity.kt

class LoginActivity : AppCompatActivity() { override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_login)

val loginBtn: Button = findViewById(R.id.btnLogin) loginBtn.setOnClickListener {

val intent = Intent(this,

CategorySelectionActivity::class.java) startActivity(intent)

}

}

}

Category Selection

// Simple buttons to choose quiz category

// Each button starts QuizActivity with different question sets

5.3 Question Model data class Question( val questionText: String, val options: List<String>,

val correctAnswer: Int

)

Quiz Logic (QuizActivity.kt) class QuizActivity : AppCompatActivity() { private val questionList = listOf(

Question("What is Kotlin?", listOf("OS", "Language", "Game", "IDE"), 1), // more questions...

)

private var currentIndex = 0 private var score = 0

private fun loadQuestion() { // Load question and options to UI

}

private fun checkAnswer(selected: Int) {

if (selected == questionList[currentIndex].correctAnswer) score++ currentIndex++

if (currentIndex < questionList.size) loadQuestion() else showResult()

}

private fun showResult() {

val intent = Intent(this, ResultActivity::class.java)

intent.putExtra("SCORE", score)

startActivity(intent)

}

}

Result Screen (ResultActivity.kt) class ResultActivity : AppCompatActivity() { override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_result)

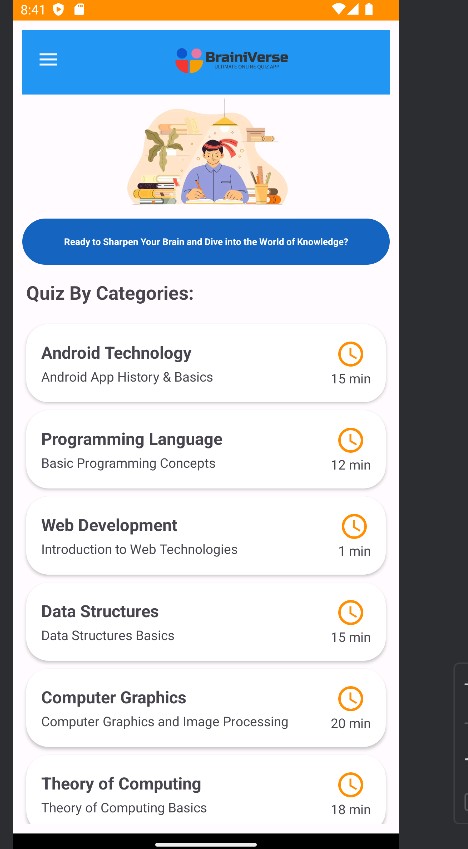
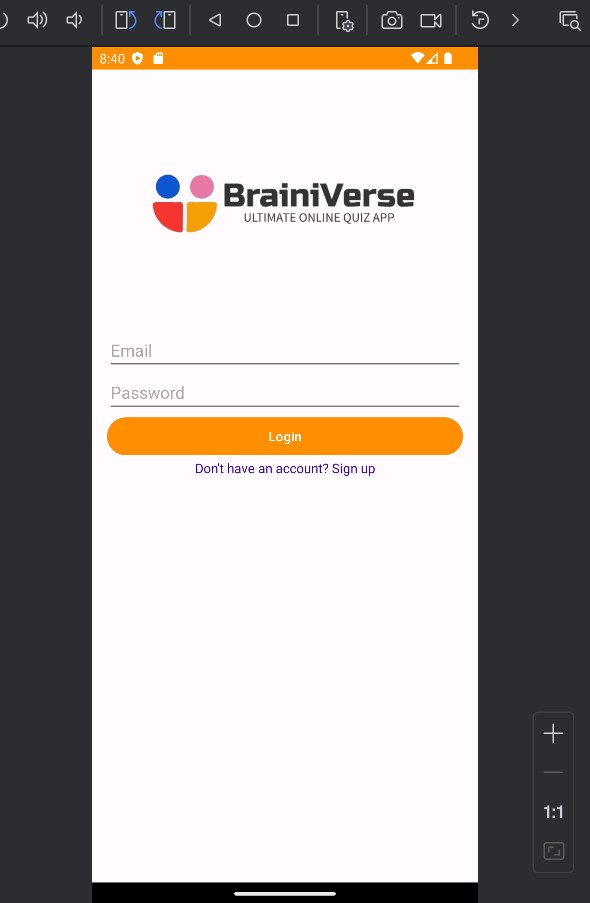
val score = intent.getIntExtra("SCORE", 0)

findViewById<TextView>(R.id.txtScore).text = "Your Score: $score/10"

}

}

# Output -



Conclusion

This mini project provided hands-on experience with mobile application development using Android Studio and Kotlin. The project encapsulated major concepts like screen navigation, data handling, event-driven programming, and UI design. Though simple, the app can be further enhanced with features like user authentication, database storage, high-score tracking, and an admin panel for adding questions.

## REFERENCES

Android Developer Docs: [https://developer.android.com](https://developer.android.com/)

* Kotlin Language Documentation: [https://kotlinlang.org](https://kotlinlang.org/)  Android Studio Guide: <https://developer.android.com/studio/intro>
* YouTube Tutorials:
* *FreeCodeCamp Android App Tutorial*
* *Philipp Lackner Kotlin Android Channel*