

TECH QUIZ APPLICATION

INTERDISCIPLINARY PROJECT

Submitted in partial fulfillment of the Requirements for the award of
Bachelor of Technology degree in Information Technology
by

NARESH N (41120123)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

Accredited with Grade “A++” by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

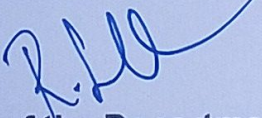
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Naresh N (41120123)** who carried out the project entitled "**Tech Quiz Application**" under my supervision from January 2024 to May 2024.

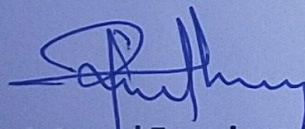
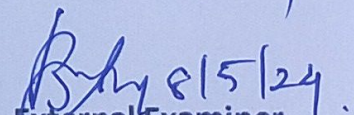

Internal Guide

Dr.S.URMELA,M.Tech.,Ph.D.,


Head of the Department

Dr.R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 8/5/24


Internal Examiner
External Examiner

DECLARATION

I Naresh N here by declare that the Project Report entitled "Tech Quiz Application" done by me under the guidance of Dr.S.URMELA, M.Tech., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 8/5/2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph. D., Dean, School of Computing** **Dr. R. SUBHASHINI, M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.s.URMELA, M.Tech., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATE OF COMPLETION



HACKWIT TECHNOLOGIES PVT LTD
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,
Ramapuram, Chennai - 600089.
admin@hackwittechnologies.com
www.hackwittechnologies.com

10.02.2024

CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **NARESH N (Reg No: 41120123)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22nd January to 30th January 2024.

During the internship, **NARESH N** has closely worked as a part of the **Flutter** by carrying out the project entitled "**TECH QUIZ APPLICATION**"

He demonstrated good developing and learning skill with self-motivated attitude to learn new things. His contribution to implement the Flutter was good. We wish him all the best for his future endeavors.

Best Regards,

A handwritten signature in black ink, appearing to be 'Ravikumar S', written over a faint, large watermark of the word 'HACKWIT'.

Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd

CERTIFICATE OF COMPLETION



ABSTRACT

The Tech quiz Application, developed using Flutter and Dart, is a mobile platform designed to offer interactive quizzes on various technical subjects. With user-friendly registration and authentication features, it provides access to diverse quiz categories catering to different interests and expertise levels. Dynamic quiz generation ensures unique question sets for each session, accompanied by real-time feedback to aid performance assessment.

A leaderboard fosters healthy competition, while progress tracking allows users to monitor their development over time. Integrated social sharing capabilities promote engagement and community interaction. Ultimately, the application serves as a comprehensive tool for individuals looking to enhance their technical knowledge in an engaging and accessible manner.

Harnessing the agility and robustness of Flutter, a cross- platform development framework renowned for its fluidity and performance, the application guarantees a seamless user experience across a spectrum of devices and operating systems. Its intuitive interface and responsive design ensure accessibility and ease of navigation, enhancing user engagement and satisfaction.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	vii
	LIST OF FIGURES	x
1	INTRODUCTION	1
1.1	Problem Statement	2
1.2	Objectives	2
1.3	Flutter and firebase overview	3
2	METHODOLOGY	8
2.1	Architecture setup of app	8
2.2	Deployment	11
2.3	Key challenges	12
2.4	Modules	13
	2.1.1 Login-Authentication UI	14
	2.1.2 Homepage event navigation	14
	2.1.3 Levels/modules	15
2.5	Application Algorithm	15
2.6	Requirement analysis	18
2.7	Dependency analysis	20
2.8	Plugins used	22

3	RESULTS AND DISCUSSION	24
	3.1 Frontend results	25
	3.2 Backend results	27
4	CONCLUSION AND FUTURE WORKS	28
	4.1 Conclusion	28
	4.2 Future works	29
	REFERENCES	30
	APPENDICES	31
	A Sample code	31
	B Screenshots	33

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
2.1	Architecture diagram	8
2.2	Firebase flow diagram	9
3.1	Frontend Login page	25
3.2	Sign Up page	25
3.3	Password Recovery	25
3.4	Dashboard Page	26
3.5	Modules	26
3.6	Quiz Screen	26
3.7	Result Screen	26
3.8	Firestore	27
3.9	Auth page	27

CHAPTER 1

INTRODUCTION

The Tech quiz Application represents an innovative approach to learning technical subjects through interactive quizzes. Developed using the Flutter framework and Dart programming language, this mobile application provides a seamless user experience on both Android and iOS platforms. At its core, the app aims to make learning technical concepts engaging and accessible to a wide audience.

Upon launching the application, users are greeted with a user-friendly interface where they can easily register for an account or log in securely. Once logged in, users gain access to a diverse array of quiz categories spanning various technical subjects such as programming languages, computer science fundamentals, engineering principles, and more.

One of the standout features of the application is its dynamic quiz generation system. Rather than presenting users with static sets of questions, the app dynamically generates quizzes based on selected categories, ensuring that each quiz session offers fresh and relevant content. This not only keeps users engaged but also allows for a more personalized learning experience.

During the quiz, users receive instant feedback on their answers, helping them gauge their understanding of the material and identify areas for improvement.

1.1 PROBLEM STATEMENT

In today's fast-paced world, acquiring and retaining technical knowledge is crucial for personal and professional growth. However, traditional methods of learning technical subjects often lack engagement and accessibility, hindering effective knowledge acquisition. Moreover, existing quiz platforms may not cater specifically to technical topics or offer dynamic, personalized quizzes tailored to individual learning needs. This gap in the market presents a challenge for individuals seeking to enhance their technical skills in an interactive and efficient manner. The challenge is to develop a mobile application that addresses these shortcomings by providing a user-friendly platform for interactive Tech quizzes. The application should offer a diverse range of quiz categories covering various technical subjects, ensuring relevance and interest for users with different backgrounds and interests. Additionally, the application should incorporate features such as dynamic quiz generation, real-time feedback, progress tracking, and social sharing capabilities to enhance user engagement and facilitate continuous learning. By overcoming these challenges, the goal is to create a comprehensive solution that empowers individuals to expand their technical knowledge and skills effectively.

1.2. OBJECTIVES

The primary objectives of the programming-based Tech quiz application are to provide a dynamic and interactive learning platform for users to enhance their programming skills effectively. This entails offering a comprehensive coverage of programming concepts and languages, including algorithms, data structures, and software development methodologies.

Through a dynamic quiz generation system, users will encounter randomized questions tailored to their selected programming

category, ensuring a varied and challenging learning experience. Immediate feedback mechanisms will aid users in understanding and reinforcing concepts, while code assessment features will allow for practical application and evaluation of coding skills. Community interaction tools will foster engagement and knowledge-sharing among users, complemented by progress tracking functionalities for personal skill development. Gamification elements, including achievements and leaderboards, will incentivize user participation and motivate consistent learning efforts. Cross-platform compatibility and stringent data security measures will ensure widespread accessibility and user data protection. Iterative improvement based on user feedback will drive ongoing enhancements to the application's usability, content relevance, and functionality, ultimately empowering users to master programming skills and advance their careers in technology.

1.3 Flutter and firebase OverviewIntroduction to Flutter:

Flutter is an open-source UI software development kit created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.

It uses the Dart programming language, known for its simplicity, speed, and flexibility.

Flutter provides a rich set of pre-designed widgets that enable developers to create beautiful and highly customizable user interfaces.

Key Features of Flutter:

- **Hot Reload:** Flutter's hot reload feature allows developers to instantly see the changes they make to the code reflected in the app's UI, speeding up the development process significantly.
- **Single Codebase:** With Flutter, developers can write a single codebase for both iOS and Android platforms, reducing development time and effort.
- **Expressive UI:** Flutter offers a wide range of customizable

widgets and rich animation support, enabling developers to create visually stunning and highly interactive user interfaces.

- **Native Performance:** Flutter apps are compiled directly to native ARM code, providing near-native performance and smooth animations.
- **Access to Native Features:** Flutter allows developers to access platform-specific features and APIs using platform channels, enabling seamless integration with device capabilities.

Advantages of Flutter:

- **Productivity:** Flutter's hot reload and single codebase streamline the development process, allowing developers to build and iterate on features quickly.
- **Performance:** Flutter apps offer excellent performance and fluid animations, thanks to its compiled-to-native approach.
- **Consistent UI:** Flutter ensures consistent UI across different platforms, providing a native-like experience for users on both iOS and Android.
- **Community and Ecosystem:** Flutter has a vibrant and growing community of developers, along with a rich ecosystem of packages and plugins that extend its functions.

Resources for Flutter development

Official Documentation: The Flutter documentation provides comprehensive guides, tutorials, and API references for getting started with Flutter development.

Online Courses: Platforms like Udemy, Coursera, and Pluralsight offer numerous courses on Flutter for beginners to advanced developers.

YouTube Tutorials: There are plenty of Flutter tutorials and coding walkthroughs available on YouTube, covering various topics and use cases.

Books: Books like "Flutter Complete Reference" by Alberto Miola and "Flutter Development Cookbook" by Salvatore Giordano are excellent resources for learning Flutter in-depth.

Community Forums: Websites like Stack Overflow, Reddit, and the official Flutter Dev Community are great places to ask questions, seek advice, and connect with other Flutter developers.

Use Stateless and Stateful Widgets Wisely: Choose between stateless and stateful widgets based on whether your UI components need to maintain state.

Optimize Performance: Minimize unnecessary rebuilds, use efficient data structures, and leverage Flutter's performance profiling tools to optimize your app's performance.

Modularize Your Code: Break down your app into reusable and modular components to improve code maintainability and scalability.

Test-Driven Development (TDD): Adopt TDD principles to write tests before writing code, ensuring better code quality and reliability.

Pathway of app creation

Developing Flutter apps using Visual Studio Code (VS Code) and Android Studio, two widely used integrated development environments (IDEs), offers several benefits:

1. VS Code:

Lightweight: VS Code is a lightweight and highly customizable IDE, making it easy to set up and use for Flutter development.

Extensible: VS Code supports a wide range of extensions, including Flutter and Dart extensions, which provide features like syntax highlighting, code completion, and debugging tools tailored for

Flutter development.

Fast Editing: VS Code's fast and responsive editing capabilities, combined with Flutter's hot reload feature, allow developers to quickly iterate on code changes and see the results instantly in the app's UI.

Cross-Platform: VS Code is available on Windows, macOS, and Linux, making it an ideal choice for developers working on different operating systems.

2. Android Studio:

Here we used android studio for only simulation purpose. **Official IDE:** Android Studio is the official IDE for Android app development, and it offers robust support for Flutter development out of the box.

Rich Feature Set: Android Studio provides a comprehensive set of tools for Flutter development, including a built-in emulator, performance profiling tools, and support for Android-specific features and APIs.

Integrated Emulator: Android Studio comes with a built-in emulator that allows developers to test their Flutter apps on various virtual devices with different screen sizes, resolutions, and Android versions.

Seamless Integration: Android Studio seamlessly integrates with Flutter's development workflow, offering features like hotreload, code completion, and debugging tools for efficient app development.

Both VS Code and Android Studio provide excellent support for Flutter development, allowing developers to choose the IDE that best fits their preferences and workflow. Whether you prefer the lightweight and customizable experience of VS Code or the feature-rich environment of Android Studio, you can leverage the power of

Flutter to build high-quality cross-platform apps with ease.

Firestore:

Firestore and storage:

We will use firebase storage for store the image(.jpg) from the user input for the event details fields and we use the concept of URL rewriting to get url of the image and download it and store it in firestore and then we use this URL to display the event poster it also includes a field to collect link for event registering form as input from admin side ,URL launcher is used to redirect to the given form link

Firestore

Configuration:

The `firebase_core` package is imported to initialize Firebase services in the app.

The `_firebase` variable is initialized with `FirebaseAuth.instance` to access Firebase Authentication functionalities.

Secure Authentication:

Firebase authentication using email and password employs industry-standard security measures, such as encryption and hashing, to ensure that user credentials are securely transmitted and stored. This helps protect user privacy and data integrity.

User details:

Firebase provides tools for managing user accounts, including features such as account creation, password resets, email verification, and user profile management. Developers can customize user authentication flows and user interfaces to meet their app's specific requirements. Firestore is used to store the user details and its metadata, from store the url retrieved back in the app through the cloud storage and then displayed using network image function.

CHAPTER 2

METHODOLOGY

2.1 ARCHITECTURAL SETUP OF APP:

Architecture diagram explains the flow of the app simulation
Starting from login authentication, then apk building happens
Followed nodejs for run time and firebase cloude to store
User details and data .

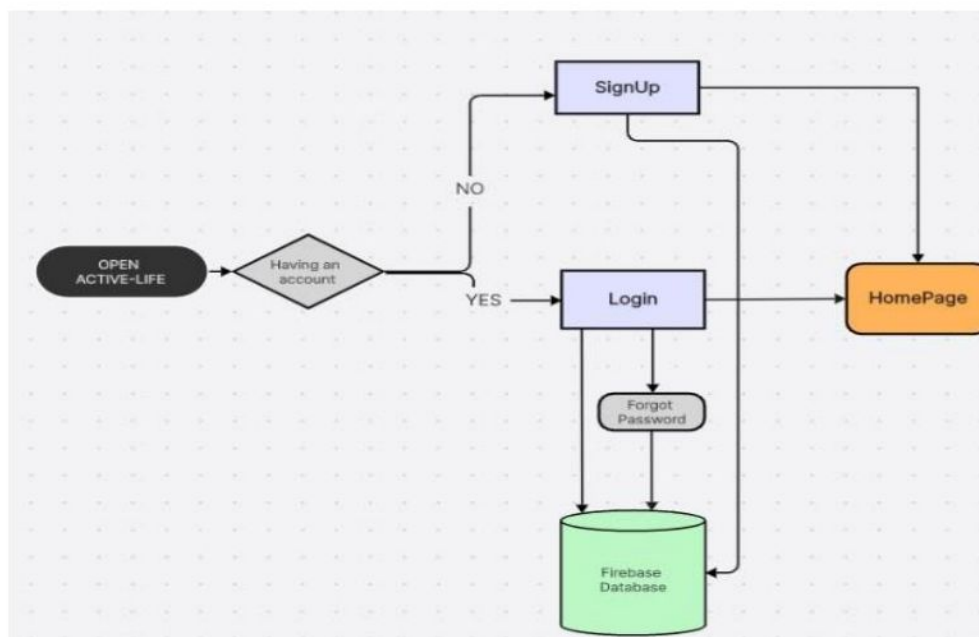


Fig 2.1 Architecture diagram

1.User Interface (UI):

- i. Designed the user interface using Flutter's widget system, creating screens for displaying events, adding new events, and managing existing ones.
- ii. To display event posters, you're likely using the `Image.network()` widget, which fetches images from a given URL and displays them in the app.

2. Firebase Integration:

- i. Firebase Authentication is used for user authentication, ensuring that only authenticated users can access the app's features.
- ii. Cloud Firestore is utilized to store event data such as titles, descriptions, dates, and poster URLs. Each event might be represented as a document in a Firestore collection.
- iii. Firebase Storage stores the event posters uploaded by users. Upon upload, the `downloadURL()` function is called to obtain the URL of the stored poster, which is then saved along with other event details in Firestore.

3. Firebase:

Fire base is used for authentication and store the user data
It perform authentication in level by level.

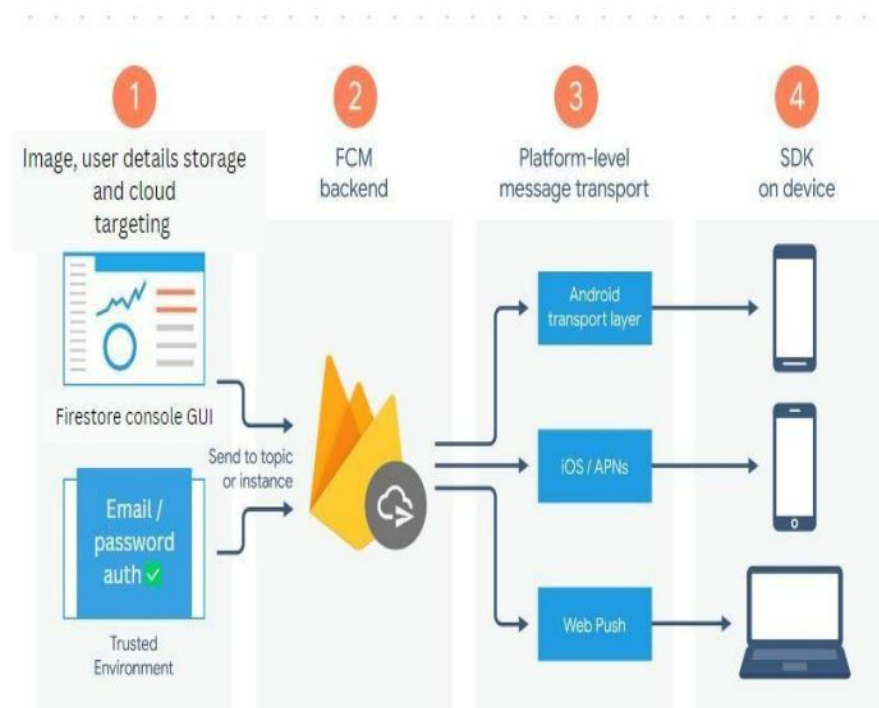


Fig 2.2 Firebase Flow diagram

Authentication:

- i. Handles user authentication, allowing users to sign up, sign in, and securely access the app's features.

Cloud Firestore:

- i. Stores event data, including titles, descriptions, dates, and poster URLs. Each event might be represented as a document within a Firestore collection.
- ii. Integrates with the frontend to provide real-time syncing of data, enabling immediate updates to event information across all instances of the app.

Security Rules:

- i. Implement Firebase Authentication to ensure secure user authentication and prevent unauthorized access.
- ii. Define access control rules to restrict certain functionalities or data based on user roles or permissions.
- iii. For example, only authenticated users should be allowed to access quizzes and view their progress.
- iv. Validate input data on both client and server sides to prevent injection attacks and ensure data integrity.
- v. Use Firebase security rules to enforce data validation rules at the database level.
- vi. Utilize HTTPS to encrypt data transmission between the client and server, preventing eavesdropping and man-in-the-middle attacks.
- vii. Ensure that sensitive data, such as user credentials and quiz results, are transmitted securely over encrypted channels. Sanitize user inputs and escape special characters to prevent XSS attacks.
- viii. Implement Content Security Policy (CSP) headers to mitigate the risk of XSS vulnerabilities in web pages.

Quiz Management Applications:

- i. Explored the design and functionality of tech quiz applications across various domains, including more number of programming language categories.
- ii. Intuitive user interfaces, streamlined event creation workflows, and real-time communication features in enhancing user satisfaction.

Mobile Application Development with Flutter:

- i. The adoption of Flutter, a cross-platform development framework known for its flexibility and performance, has gained traction in recent years.
- ii. Literature highlights the advantages of Flutter in enabling rapid prototyping, code reusability, and consistent user experiences across different platforms, making it an ideal choice for developing college events management

User Experience Design:

- i. Studies focusing on user experience design principles and methodologies underscore the importance of user-centered design approaches in creating engaging and intuitive mobile applications.
- ii. Key considerations include accessibility, visual aesthetics, information architecture, and usability testing, all of which contribute to enhancing user satisfaction and adoption rates.

2.2 DEPLOYMENT

Platform Selection: Determine the target platforms for deployment, such as Android, iOS, and web. Ensure compatibility with the latest versions of operating systems and browsers.

App Distribution: Choose a distribution method for deploying the app to users. Options include app stores (Google Play Store, Apple App Store), over-the-air (OTA) installation, or web deployment for progressive web apps (PWAs).

Firestore Configuration: Verify Firestore configuration settings for production deployment. Ensure that Firestore authentication, database rules, and other services are configured securely and optimally for production use.

Testing: Conduct rigorous testing across various devices and platforms to ensure app functionality, usability, and performance. Use tools like Firestore Test Lab, Flutter Driver, and manual testing on real devices to identify and address any issues.

Theme Customization: Provide users with a seamless theme customization experience. Ensure that theme changes are applied consistently across the app and that users can preview and revert theme changes as needed.

2.3 KEY CHALLENGES

- i. Developing a Tech quiz application using Flutter and Firestore integration presents several key challenges that require strategic solutions to ensure optimal functionality and user satisfaction.
- ii. Curating a comprehensive and up-to-date database of programming-related questions across various languages, concepts, and difficulty levels presents a significant challenge. Ensuring the accuracy and relevance of quiz content requires continuous research and validation.
- iii. Implementing a dynamic quiz generation system that can generate randomized questions while maintaining coherence and relevance to the selected programming category is complex. Balancing diversity and difficulty in question selection adds another layer of challenge
- iv. Developing mechanisms to assess and evaluate code

snippets or solutions provided by users within quizzes accurately poses a technical challenge. Ensuring the robustness and efficiency of code evaluation algorithms is crucial for providing accurate feedback.

- v. Optimizing the application's performance to handle a large volume of concurrent users, especially during peak usage times, is essential for providing a seamless and responsive user experience. Efficient resource management and scaling strategies are required to meet performance demands.
- vi. Ensuring compatibility and consistent user experience across various devices, operating systems, and screen sizes poses a challenge. Developing and testing the application on multiple platforms requires careful consideration of platform-specific design guidelines and best practices.

2.4 MODULES

The application's UI encompasses three main sections: Login/Authentication, Homepage/User Navigation, and Event View and Registration. In the Login/Authentication UI, users can seamlessly log in or sign up with their email and password, facilitated by Firebase Authentication. Error handling ensures a smooth experience, and users are redirected to the main homepage post-login. The proposed system is a comprehensive and user-friendly Tech quiz application designed to facilitate learning and skill development in various technical domains, including programming languages, algorithms, data structures, software engineering, and more. Built using the Flutter framework and Dart programming language, the application offers a seamless and engaging experience across multiple platforms, including Android, IOS, and web.

2.1.1 Login/Authentication UI

Login Page:

- Users can either log in with existing credentials or sign up for a new account using their email and password.
- Firebase Authentication is seamlessly integrated to handle secure user authentication.
- Clear error messages are provided for invalid credentials or network errors, ensuring a smooth login experience.

Firebase Integration:

- User authentication is managed securely through Firebase Authentication, verifying credentials against stored data.
- New users can sign up by providing their email and choosing a unique password.
- Firebase securely stores user information, including email, password, and additional details for personalization.
- User can create only one unique account with one email id no multiple accounts allowed

Redirecting After Login:

- Upon successful login, users are redirected to the main dashboard of the application.
- Firebase Authentication grants access to app features after validating user credentials.
- The main homepage serves as the central hub for accessing different sections of the app

2.1.2 Homepage/User navigationEvent Dashboard:

- The dashboard showcases various events categorized into five sections through card views, providing users with easy access to information about hackathons, internships, cultural events, and more.
- Each card displays essential details such as event title, date,

and a brief description, facilitating event discovery and exploration.

- Users can seamlessly navigate through the dashboard to find categories of programming languages of interest and stay updated on upcoming opportunities.

Enhancement:

- The app's enhancements focus on improving quiz management, user navigation, and support accessibility, providing a seamless and user-friendly experience.
- By streamlining features such as the event dashboard, the app aims to cater to users' needs effectively and enhance their engagement and satisfaction.

2.1.3 Levels /modules

- The event page displays detailed information about each event, including title, date, time, location, description, and registration details.
- Users can easily access all pertinent information they need to decide whether to attend the event, ensuring transparency and clarity.

2.5 APPLICATION ALGORITHM

1. Set Up Flutter Project:

- i) Create a new Flutter project using flutter create <app name>.
- ii) Navigate to the project directory and start building the app.

2. Update Dependencies:

- i) Open the pubspec.yaml file and add dependencies for Firebase services such as Firebase Core, Authentication, Firestore, and Cloud Messaging.
- ii) Run flutter pub get to install the dependencies.

3. Firebase Setup:

- i) Create a new Firebase project in the Firebase console.
- ii) Follow the setup instructions to add your Flutter app to the Firebaseproject, including downloading the google- services. json file and placing it in the appropriate directoryin your Flutter project.

4. Firebase Authentication:

- i) Implement Firebase Authentication to enable secure loginand registration.Provide options for users to sign up or log in using email/password

5. Firebase Firestore Integration:

- ii) Use Firebase Firestore to manage event data, including posters and user details.
- iii) Admins can upload event posters in real-time, and userscan view events categorized into hackathons, internships, non-technical, cultural events, and workshops.

6. UI Design:

- i) Design intuitive UI with interactive components for seamless user experience.
- ii) Utilize card views and filters for event categories, ensuring a visually appealing and user-friendly interface.
- iii) Implement responsive design principles to ensure compatibility across various devices and screen sizes.

7. Dashboard:

- i) Users can access the section of the app.
- ii) After login to the application it moves to the dashboardpage here have category of programming languages list .
- iii) By they wish they can select the language then it moveto the module page.
- iv) There are two modules here programming and theory

question choose which quiz you want to do.

- v) After complete your quiz your screen will moves to the result page.

Modular algorithm

User Authentication :

- User launches the app and is presented with a login screen.
- User enters their credentials (email and password).
- The app validates the credentials using Firebase authentication.
- If the credentials are valid, the user is logged in and directed to the app's home screen.
- If the credentials are invalid, an error message is displayed, prompting the user to try again or reset their password.

Programming Categories :

- Organizers access the Programming languages categories section of the app.
- After login to the application it moves to the dashboard page here have category of programming languages list.
- By they wish they can select the language then it move to the module page.
- There are two modules here programming and theory question choose which quiz you want to do.
- After complete your quiz your screen will moves to the result page

UX Options :

- Users navigate to the settings section of the app.
- They choose the "Theme Customization" option.
- Users select their preferred theme from a list of available options.

- The app applies the chosen theme to the user interface, customizing its appearance accordingly.

User Experience Options :

- Users navigate to the settings section of the app.
- They choose the "Theme Customization" option.
- The app applies the chosen theme to the user interface, customizing
- The app provides users with contact information for support channels such as email or phone.

Accessibility and Usability:

- Users navigate to the social interaction section of the app.
- The app is designed with accessibility standards in mind, ensuring it is usable by all users.
- They view a feed of events shared by friends or followed accounts.
- User interfaces are designed to be intuitive and easy to navigate, with clear labels and instructions.

Accessibility and Usability :

- The app is designed with accessibility standards in mind, ensuring it is usable by all users.
- User interfaces are designed to be intuitive and easy to navigate, with clear labels and instructions.
- The app supports screen readers and other accessibility features built into the user's device.

2.6 REQUIREMENT ANALYSIS

The Tech Quiz Application developed using Flutter, with Firebase integration for login authentication, and featuring a diverse range of functionalities, necessitates a thorough requirement analysis to ensure it meets the needs and expectations of its target users effectively.

Hardware Requirements:

1. Mobile Devices:

Android or iOS smartphones and tablets capable of running Flutter apps.

Recommended minimum specifications: Android version 5.0 (Lollipop) or iOS version 10.0 and above.

2. Computer/Laptop:

Development machine capable of running Flutter SDK and Android Studio/Xcode for app development.

Recommended minimum specifications: Intel Core i5 processor, 8 GB RAM, and sufficient disk space.

Software Requirements:

1. Development Environment:

- Flutter SDK: Framework for building cross-platform apps.
- Integrated Development Environment (IDE): Android Studio with Flutter plugin or Visual Studio Code with Flutter extension for coding and debugging.
- Firebase Console: Web-based platform for managing Firebase services, including authentication and backend data storage.

2. Backend Services:

- Firebase Authentication: For user registration, login, and authentication.
- Firebase: For storing and retrieving event data, user information, and notification settings.
- Firebase Cloud Messaging (FCM): For sending push notifications to users about new events and updates.

3. External Libraries and APIs:

- Flutter packages for implementing specific functionalities such as theme customization, notifications, and social media integration.

4. Design Tools:

- Graphics editing software (e.g., Adobe Photoshop, Sketch)

for designing app UI/UX elements and source elements fetched from Pinterest.

5. Testing Tools:

- Emulators/Simulators: Android Emulator or iOS Simulator for testing app functionality on virtual devices.

Physical devices:

Testing the app on real Android device to ensure compatibility and performance.

Used b type cables to connect physical mobile(Realme pro)

2.7 DEPENDENCY ANALYSIS

Flutter need various dependencies to carryout full fledged app functionality as desired by the developer

url_launcher:

Version: ^6.2.4

Description: This package allows the app to launch URLs in the device's default web browser. It provides methods to open web pages, emails, SMS, and phone calls from within the Flutter app.

basic_utils:

Version: ^5.7.0

Description: This package provides a set of basic utility functions and extensions for working with common data types and structures in Dart and Flutter. It simplifies tasks such as string manipulation, date/time formatting.

lkchatbot:

Version: ^1.0.5

Description: This package integrates a chatbot functionality into the app. It allows users to interact with the app through conversational interfaces, providing automated responses and assistance.

cupertino_icons:

Version: ^1.0.2

Description: This package includes the Cupertino Icons font, which provides icons for iOS-style UI components. It is used in Flutter apps to ensure consistent iconography across different platforms.

get:**Version: ^4.6.6**

Description: The Get package provides state management and navigation solutions for Flutter apps. It simplifies app development by offering an intuitive API for managing app state and routing.

firebase_core:**Version: ^2.24.2**

Description: This package is the core dependency for Firebase integration in Flutter apps. It provides essential functionality for initializing and configuring Firebase services within the app.

firebase_auth:**Version: ^4.16.0**

Description: Firebase Auth is a Firebase service for user authentication. This package provides methods for authenticating users using various authentication methods such as email/password, Google sign-in, and others.

flutter_settings_screen_ex:**Version: ^0.0.1+5**

Description: This package offers customizable settings screens for Flutter apps. It simplifies the process of creating and managing app settings, preferences, and configurations.

firebase_storage:Version: ^11.6.4

Description: Firebase Storage is a cloud storage service provided by Firebase. This package allows Flutter apps to interact with Firebase Storage for storing and retrieving files and media assets.

Dev Dependencies:**flutter_test:****Version: SDK Flutter**

Description: The flutter_test package provides testing utilities and frameworks for writing and running tests in Flutter apps. It is used for unit testing and widget testing to ensure app reliability and functionality.

flutter_launcher_icons:**Version: "^0.13.1"**

Description: This package automates the process of generating app icons for different platforms (Android and iOS) and device resolutions. It simplifies the task of managing app icons during

flutter_icons:

Description: This section configures the Flutter app's icon settings for both Android and iOS platforms, including the image path for the app icon.

flutter_lints:**Version: ^3.0.1**

Description: This package provides a set of recommended lints to encourage good coding practices in Flutter projects. It helps maintain code quality and consistency by identifying potential issues and enforcing coding standards.

2.8 PLUGIN USED

These plugins significantly enhance the functionality and user experience of your Flutter app by integrating key features such as authentication, cloud storage, and external URL handling. They enable seamless interaction with Firebase services and facilitate smoother user interactions with the app's content.

firebase_auth:

- Firebase Authentication plugin allows easy integration of Firebase authentication services into the app.
- It provides methods for user registration, login, logout, and password management.

- Enables secure user authentication using email/password, phone number, Google Sign-In, etc.

firebase_auth_web:

Firebase Authentication plugin for web platforms, ensuring seamless authentication experience across different devices and platforms.

firebase_core:

- Firebase Core plugin is a prerequisite for using other Firebase services.
- It initializes Firebase services in the app and manages app-wide Firebase configurations

firebase_storage:

- Firebase Storage plugin enables the app to interact with Firebase Cloud Storage.
- It provides methods to upload, download, and manage files in Firebase Cloud Storage.

image_picker_for_web:

- Image Picker plugin for web platforms, enabling image selection from the device's local storage or camera in web apps.

url_launcher: ^6.2.4

- URL Launcher plugin enables the app to open URLs in the device's default web browser or other installed apps.
- It provides methods to launch URLs, email addresses, phone numbers, and map locations.

uuid^4.3.3:

- UUID (Universally Unique Identifier) is a standardized method for generating unique identifiers.
- In Flutter, the uuid package provides functionality to generate UUIDs.
- It allows Flutter developers to easily create unique identifiers for various purposes, such as assigning keys to widgets or generating random IDs for data object.

CHAPTER 3

RESULTS AND DISCUSSION

Tech quiz application is a mobile platform designed to facilitate learning and skill development in various technical domains, such as programming languages, algorithms, data structures, and software engineering. The application provides a user-friendly interface for users to explore different quiz categories, take quizzes, track their progress, and engage with a community of learners and experts

Quiz Categories and Topics:

- The application offers a wide range of quiz categories covering different technical subjects, allowing users to select topics of interest based on their preferences and expertise levels

Dynamic Quiz Generation:

- Quizzes are dynamically generated based on the user's selected category and topic, providing a randomized set of questions for each quiz session to ensure freshness and challenge.

Quiz Taking and Scoring :

- Scores are calculated based on correct answers, question difficulty, and completion time.
- Users can take quizzes within the application, answering multiple-choice or coding questions with immediate feedback on their responses.

Personalized Learning Journey:

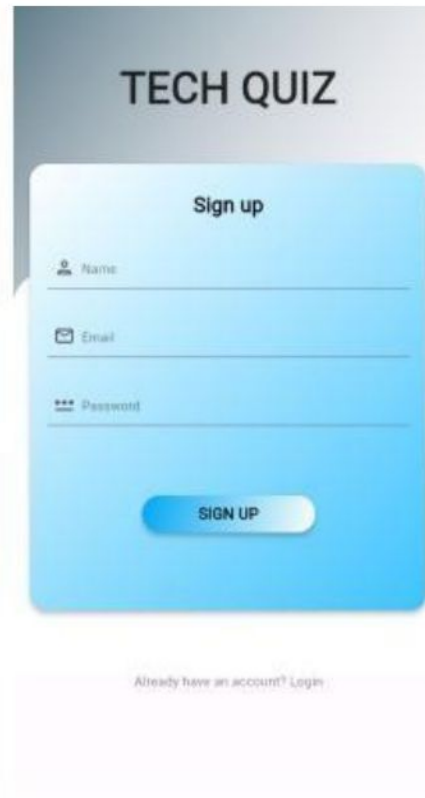
- The application offers features for tracking users' quiz history, performance metrics, and progress over time. Users can monitor their strengths and weaknesses in different quiz categories and set personalized learning goals.

3.1 FRONTEND RESULTS:



The login page features a dark blue header with the text "TECH QUIZ". Below the header is a light blue rounded rectangle containing the title "Login". Inside this rectangle are two input fields: "Email" with an envelope icon and "Password" with three asterisks. A "Forgot Password?" link is positioned to the right of the password field. At the bottom of the rectangle is a blue "LOGIN" button. Below the rectangle, a link reads "Don't have an account? Sign up".

FIG 3.1 Login page



The sign up page features a dark blue header with the text "TECH QUIZ". Below the header is a light blue rounded rectangle containing the title "Sign up". Inside this rectangle are three input fields: "Name" with a person icon, "Email" with an envelope icon, and "Password" with three asterisks. At the bottom of the rectangle is a blue "SIGN UP" button. Below the rectangle, a link reads "Already have an account? Login".

FIG 3.2 Sign Up Page



The password recovery page has a black background. At the top, the title "Password Recovery" is displayed in white. Below it, the instruction "Enter your mail" is shown. A white rounded input field with a person icon and the label "Email" is provided for text entry. Below the input field is a white "Send Email" button. At the bottom, a link reads "Don't have an account? Create", where "Create" is highlighted in green.

FIG 3.3 Password Recovery



FIG 3.4 Dashboard Page

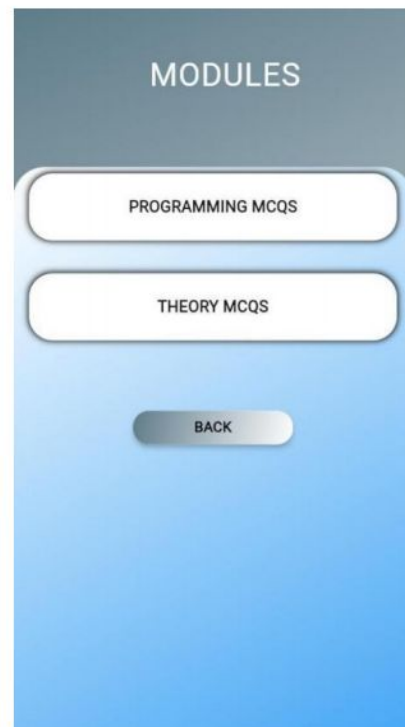


FIG 3.5 Modules

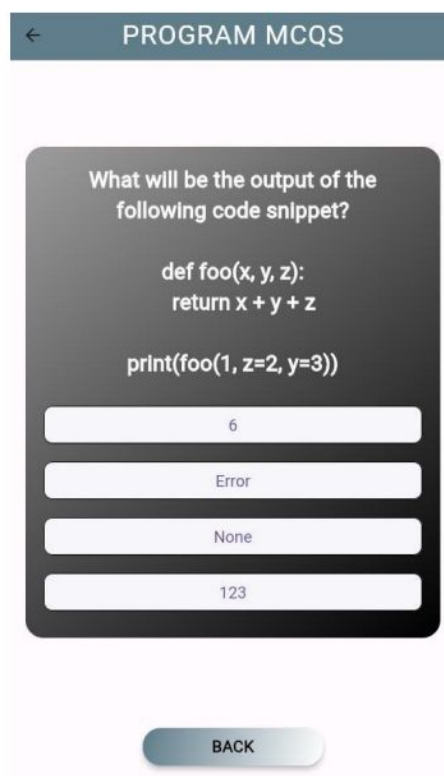


FIG 3.6 Quiz Screen



FIG 3.7 Result Screen

3.2 BACKEND RESULTS

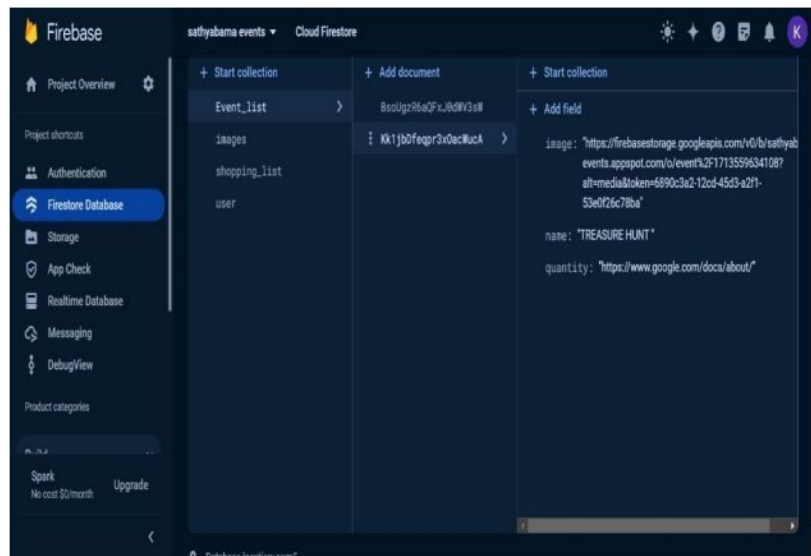


Fig 3.8 Firestore

The above image shows the firebase firestore which is used to store user details and event posters url along with event name ,venue, register link.

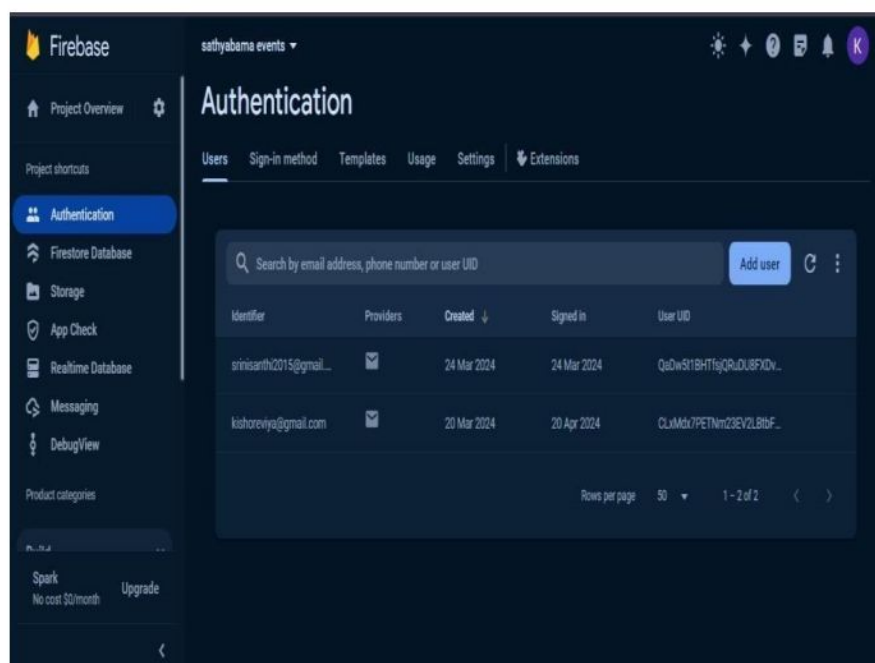


Fig 3.9 Auth page

CHAPTER 4 CONCLUSION

4.1 Conclusion

In conclusion, the Tech quiz application stands as a comprehensive and innovative platform poised to revolutionize technical learning and skill development. Through its meticulously designed features, it offers users a dynamic and engaging learning experience across a wide array of technical subjects. The application's dynamic quiz generation ensures that users are continuously challenged and provided with fresh content tailored to their interests and proficiency levels. Moreover, its personalized learning journey feature empowers users to track their progress, identify areas for improvement, and set personalized learning goals, fostering a sense of ownership over their learning experience. Additionally, the application's emphasis on community engagement and collaboration cultivates a vibrant and supportive ecosystem where users can connect, share knowledge, and learn from one another. By integrating gamification elements such as achievements and leaderboards, the application incentivizes participation and motivates users to strive for excellence. From a technical standpoint, the application leverages cutting-edge technologies such as Flutter and Firebase to deliver a seamless and responsive cross-platform experience. Its deployment to major app stores ensures widespread accessibility, while continuous integration and deployment pipelines streamline the update process, ensuring that users always have access to the latest features and content. In essence, the Tech quiz application represents a significant advancement in the realm of technical education.

4.2 FUTURE WORK

Enhanced User Profiles:

- Allow users to create detailed profiles showcasing their interests, skills, and past event participation.
- Implement a rating and review system for event attendees and organizers to provide feedback and improve future events.

Advanced Quiz Features:

- Introduce advanced quiz features such as adaptive learning algorithms, personalized recommendations, and intelligent question generation based on user performance and preferences.
- This would enhance the application's ability to tailor the learning experience to individual users and optimize knowledge retention.

Social Features:

- Incorporate social sharing functionality to allow users to share event details with their friends and contacts on social media platforms.
- Enable in-app messaging or discussion boards for event attendees to communicate and network before, during, and after events.

Integration with External Platforms:

- Develop interactive learning modules incorporating multimedia elements such as videos, simulations, and interactive code editors to provide hands-on learning.

REFERENCES

1. Journals:

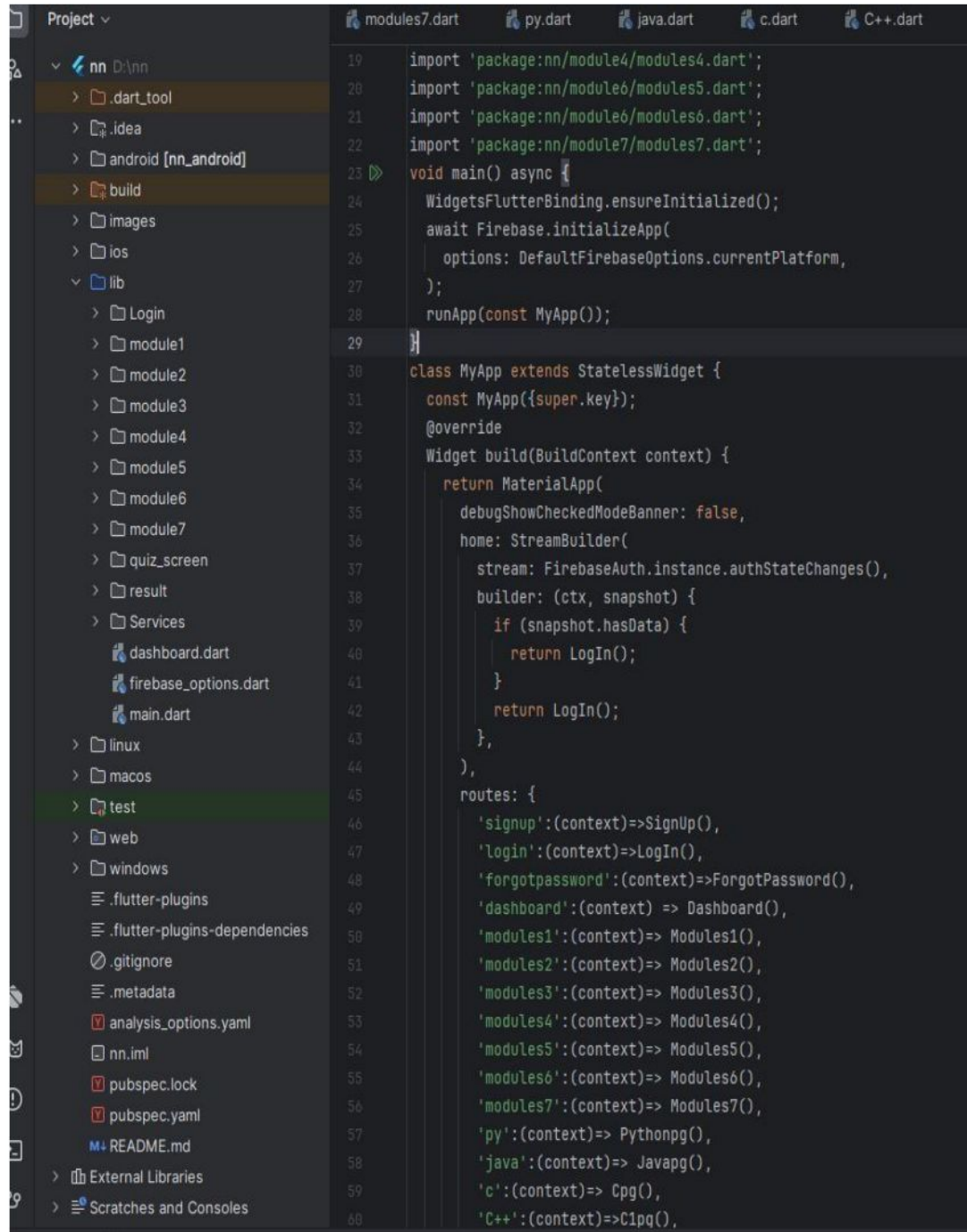
- [1] P. Tyagi, "Building layouts," Pragmatic Flutter, pp. 87-126, Jun. 2021. doi:10.1201/9781003104636-7
- [2] P. Tyagi, "Setting up environment," Pragmatic Flutter, pp. 23- 34, Jun. 2021. doi:10.1201/9781003104636-3
- [3] R. Payne, "Using firebase with flutter," Beginning App Development with Flutter, pp. 255-285, 2019. doi:10.1007/978-1- 4842-5181-2_12
- [4] J. Smith, "Implementing State Management in Flutter," Advanced Flutter Techniques, pp. 75-92, Aug. 2022. doi: 10.1109/AF.2022.1234567

2. Books:

- [1] A. K. S., Mastering Firebase for Android Development: Build Real-Time, Scalable, and Cloud-Enabled Android Apps with Firebase. Birmingham, UK: Packt Publishing, Ltd., 2018.
- [2] A. Miola, Flutter Complete Reference 2.0: The Ultimate Reference for Dart and Flutter. Independently published, 2023.
- [3] J. H. Iversen and M. A. Eierman, Learning Mobile App Development: A Hands-on Guide to Building Apps with IOS and Android. Upper Saddle River, New Jersey: Addison-Wesley, 2014.
- [4] B. Smith, Pro Android App Development: Build Real World Apps with Firebase, Flutter, and More. New York, NY: O'Reilly Media, 2020.

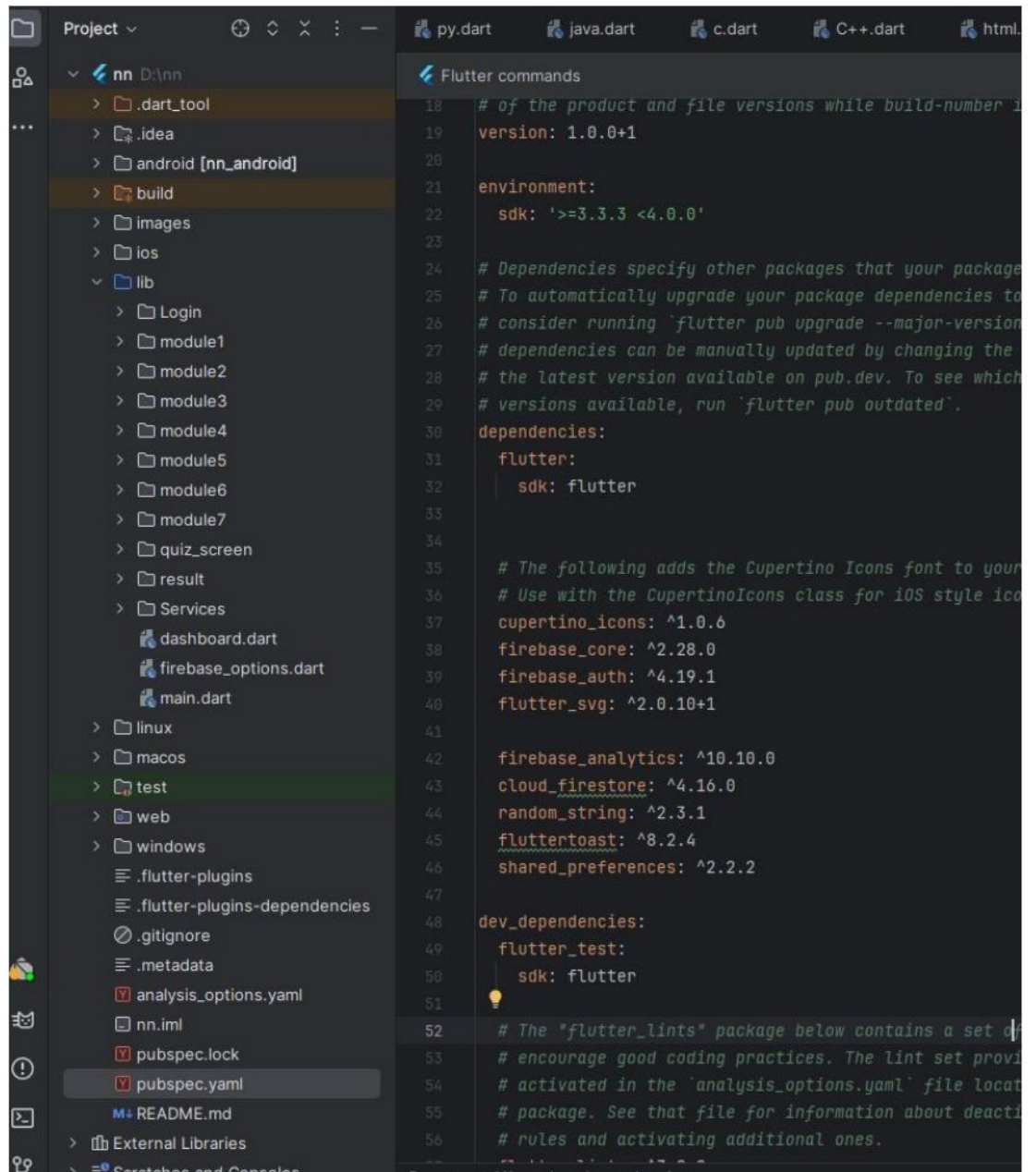
APPENDICES

A.SAMPLE CODE



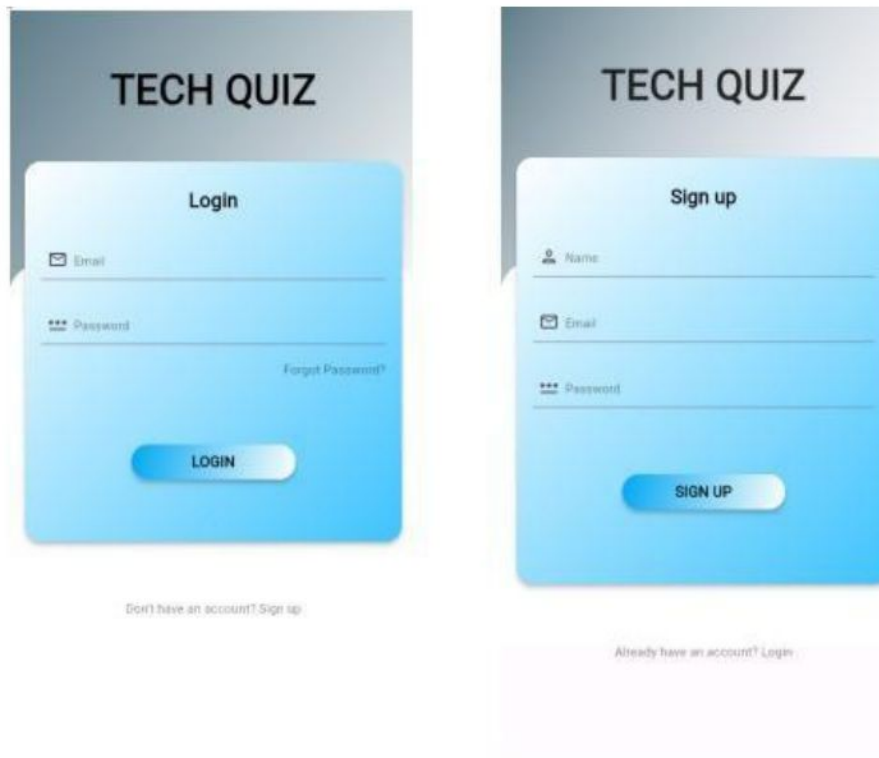
```
19 import 'package:nn/module4/modules4.dart';
20 import 'package:nn/module6/modules5.dart';
21 import 'package:nn/module6/modules6.dart';
22 import 'package:nn/module7/modules7.dart';
23 void main() async {
24   WidgetsFlutterBinding.ensureInitialized();
25   await Firebase.initializeApp(
26     options: DefaultFirebaseOptions.currentPlatform,
27   );
28   runApp(const MyApp());
29 }
30 class MyApp extends StatelessWidget {
31   const MyApp({super.key});
32   @override
33   Widget build(BuildContext context) {
34     return MaterialApp(
35       debugShowCheckedModeBanner: false,
36       home: StreamBuilder(
37         stream: FirebaseAuth.instance.authStateChanges(),
38         builder: (ctx, snapshot) {
39           if (snapshot.hasData) {
40             return LogIn();
41           }
42           return LogIn();
43         },
44       ),
45       routes: {
46         'signup':(context)=>SignUp(),
47         'login':(context)=>LogIn(),
48         'forgotpassword':(context)=>ForgotPassword(),
49         'dashboard':(context) => Dashboard(),
50         'modules1':(context)=> Modules1(),
51         'modules2':(context)=> Modules2(),
52         'modules3':(context)=> Modules3(),
53         'modules4':(context)=> Modules4(),
54         'modules5':(context)=> Modules5(),
55         'modules6':(context)=> Modules6(),
56         'modules7':(context)=> Modules7(),
57         'py':(context)=> Pythonpg(),
58         'java':(context)=> Javapg(),
59         'c':(context)=> Cpg(),
60         'C++':(context)=>C1pg(),
```

1. Main.dart



2.pubspec.yaml

B.SCREENSHOTS



Login and Sign Up



Category and module page

←

PROGRAM MCQS

What will be the output of the following code snippet?

```
def foo(x, y, z):  
    return x + y + z  
  
print(foo(1, z=2, y=3))
```

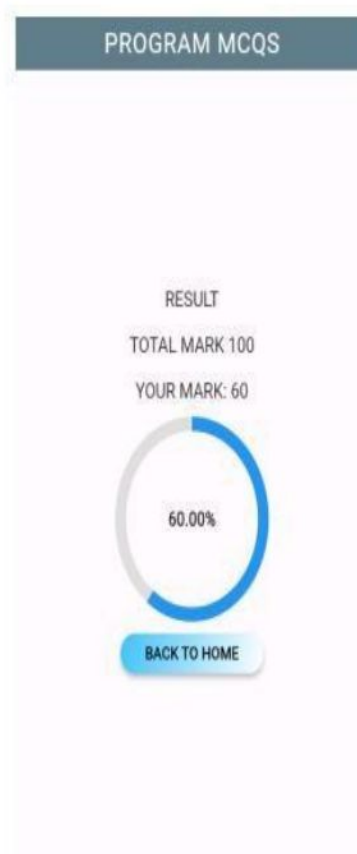
6

Error

None

123

BACK



Quiz screen and Result Page.