

My Notes Application

Name : Aniket Bourasee

Email : aniketbourasee007@gmail.com

Project Overview :

This project is a note-making application built for Android using Kotlin. It utilizes Firebase for user authentication and data storage, allowing users to create, manage, and organize their notes securely. The app features user registration with email verification and data synchronization with Firebase Realtime Database.

Features:

- **User Registration:** Allows users to create an account with email and password.
- **Email Verification:** Sends a verification email upon successful registration.
- **User Login:** Users can log in to their accounts after verification.
- **Notes Management:** Users can create, view, edit, and delete notes.
- **Data Storage:** All notes are stored in Firebase Realtime Database.
- **User-Friendly UI:** Clean and intuitive interface with Material Design components.

Technologies Used:

- **Programming Language:** Kotlin
- **Development Environment:** Android Studio
- **Database:** Firebase Realtime Database
- **Authentication:** Firebase Authentication

Project Structure:

- **Activities:**
 - **SplashActivity**: Displays the app name and checks user authentication status.
 - **CreateAccountActivity**: Handles user registration and email verification.
 - **LoginActivity**: Manages user login.
 - **MainActivity**: Displays the list of notes.
 - **NoteDetailsActivity**: Facilitates adding and editing notes.

- **XML Layouts:**
 - activity_splash.xml
 - activity_create_account.xml
 - activity_login.xml
 - activity_main.xml
 - activity_note_details.xml
 - recycler_note_item.xml

Steps to Create the Application:

Step 1: Set Up Your Development Environment

1. Create a new project using an Empty Activity template.
2. Configure Gradle with necessary Firebase dependencies.

Step 2: Set Up Firebase

1. Create a Firebase project and register your app.
2. Enable Firebase Authentication and Realtime Database.

Step 3: Design Layouts

1. Create the XML layouts.

Step 4: Implement Splash Screen

1. Create SplashActivity to check user authentication status.

Step 5: Implement User Registration

1. Create CreateAccountActivity for user registration and email verification.

Step 6: Implement User Login

1. Create LoginActivity to manage user login.

Step 7: Build Main Activity

1. Create MainActivity to display notes in a RecyclerView.

Step 8: Implement Note Management

1. Create NoteDetailsActivity for creating and editing notes.

Step 9: Handle Data in RecyclerView

1. Create a RecyclerView adapter to bind note data.

Step 10: Test the Application

1. Run the app to ensure all functionalities work as expected.

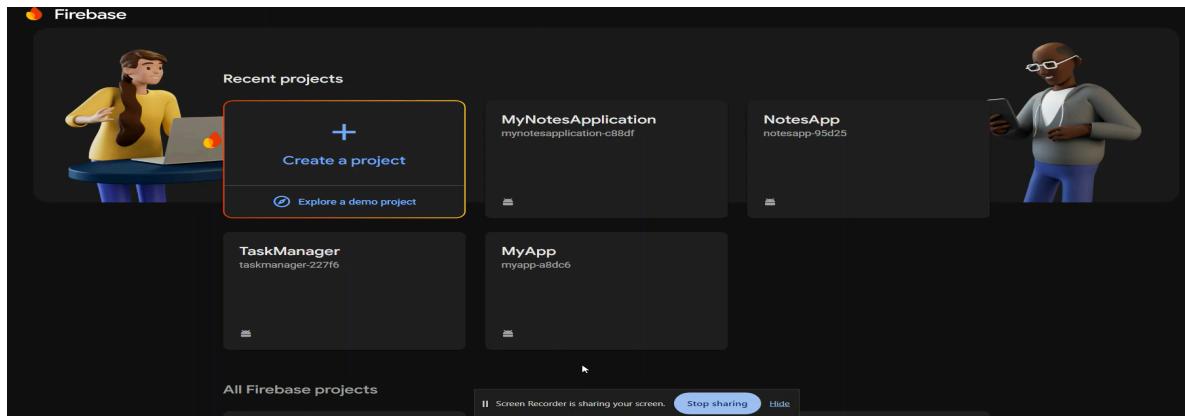
Conclusion:

This note-taking app provides a robust platform for users to manage their notes efficiently. By leveraging Firebase, it ensures secure user authentication and reliable data storage.

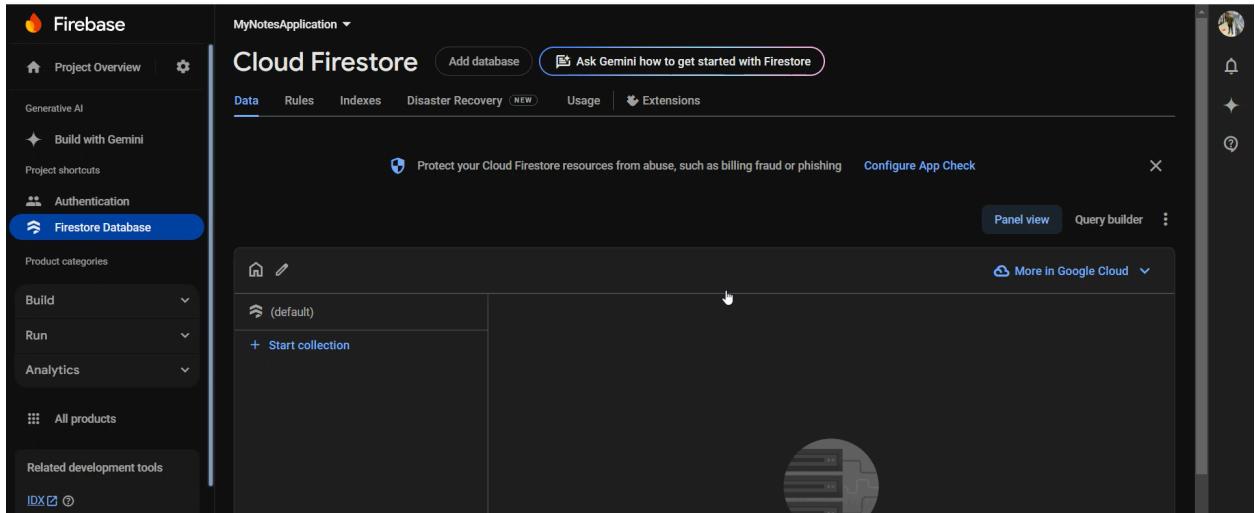
Project Demo

Firebase Connection:

- Our project, named “MyNotesApplication,” is successfully connected to the Firebase database.
- Currently, there are no users listed in Firebase Authentication, and the database is empty.

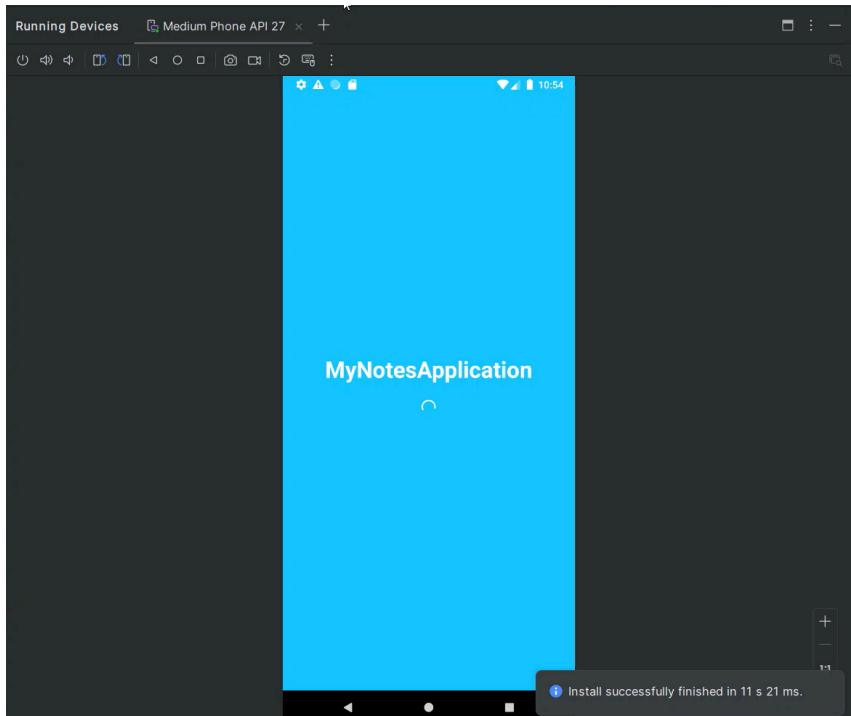


A screenshot of the 'Authentication' section within the Firebase console for the 'MyNotesApplication' project. The sidebar on the left shows 'Authentication' is selected. The main area is titled 'Authentication' and contains tabs for 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. A table below shows user data with columns: Identifier, Providers, Created, Signed in, and User UID. A message at the bottom states 'No users for this project yet.' There are buttons for 'Add user' and a refresh icon.



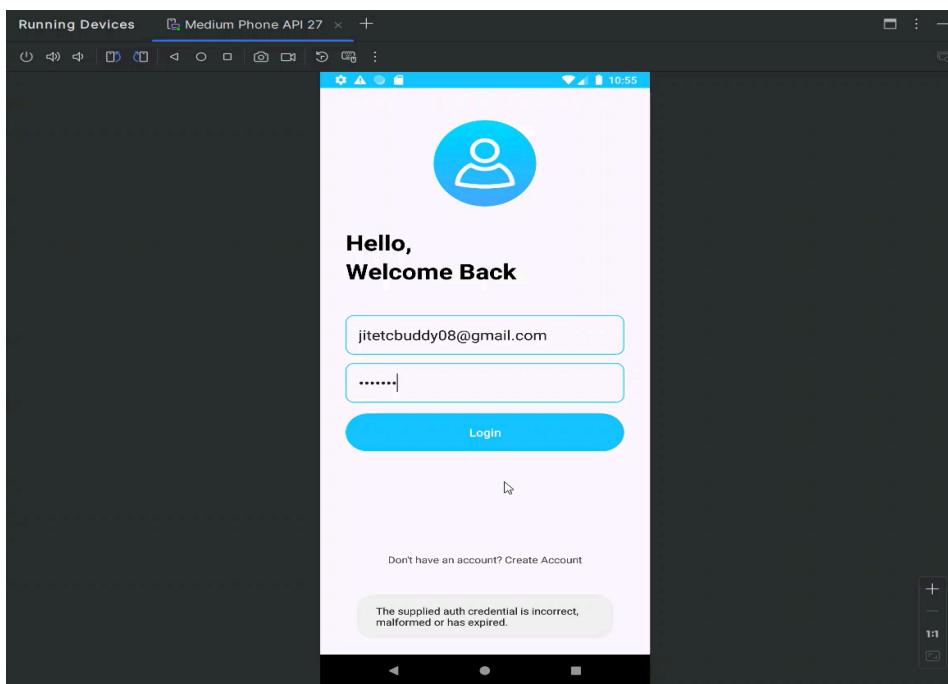
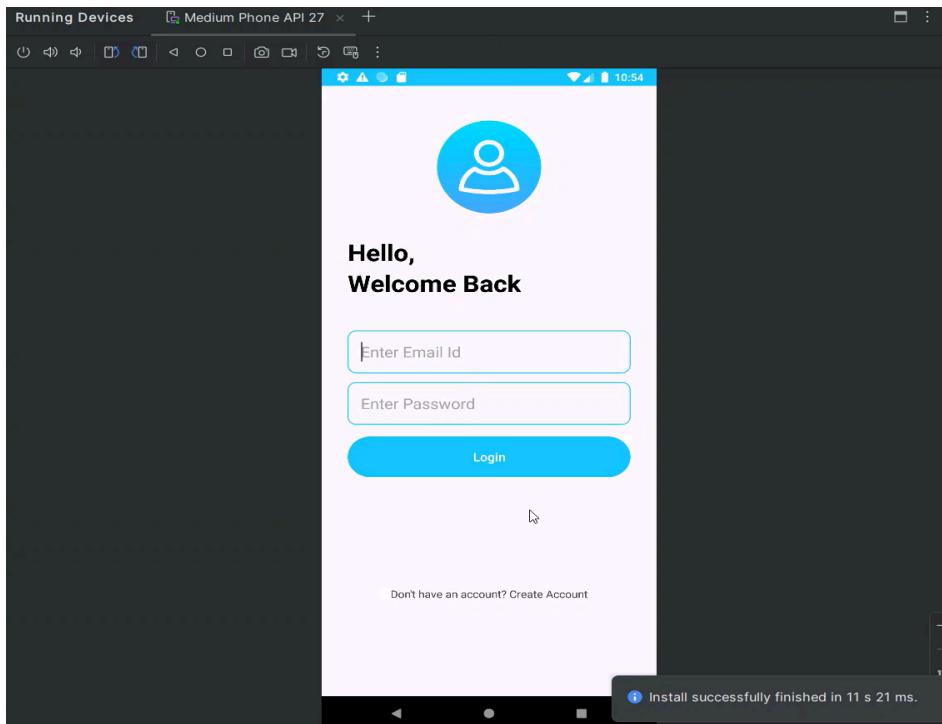
Launching the Application:

- Launch the app on an emulator.
- A **Splash Screen** is displayed with the app name and a loader animation.



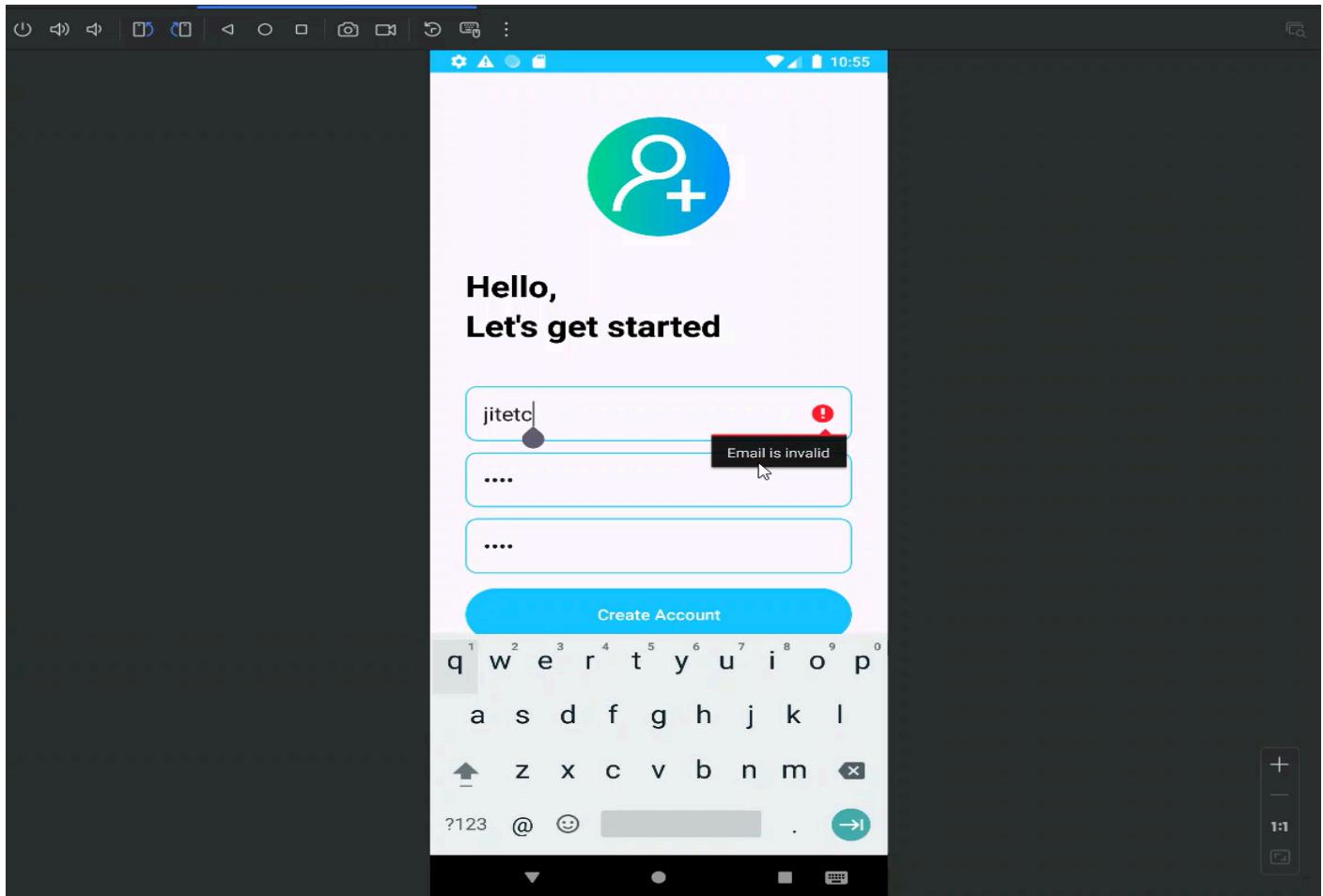
Login Page:

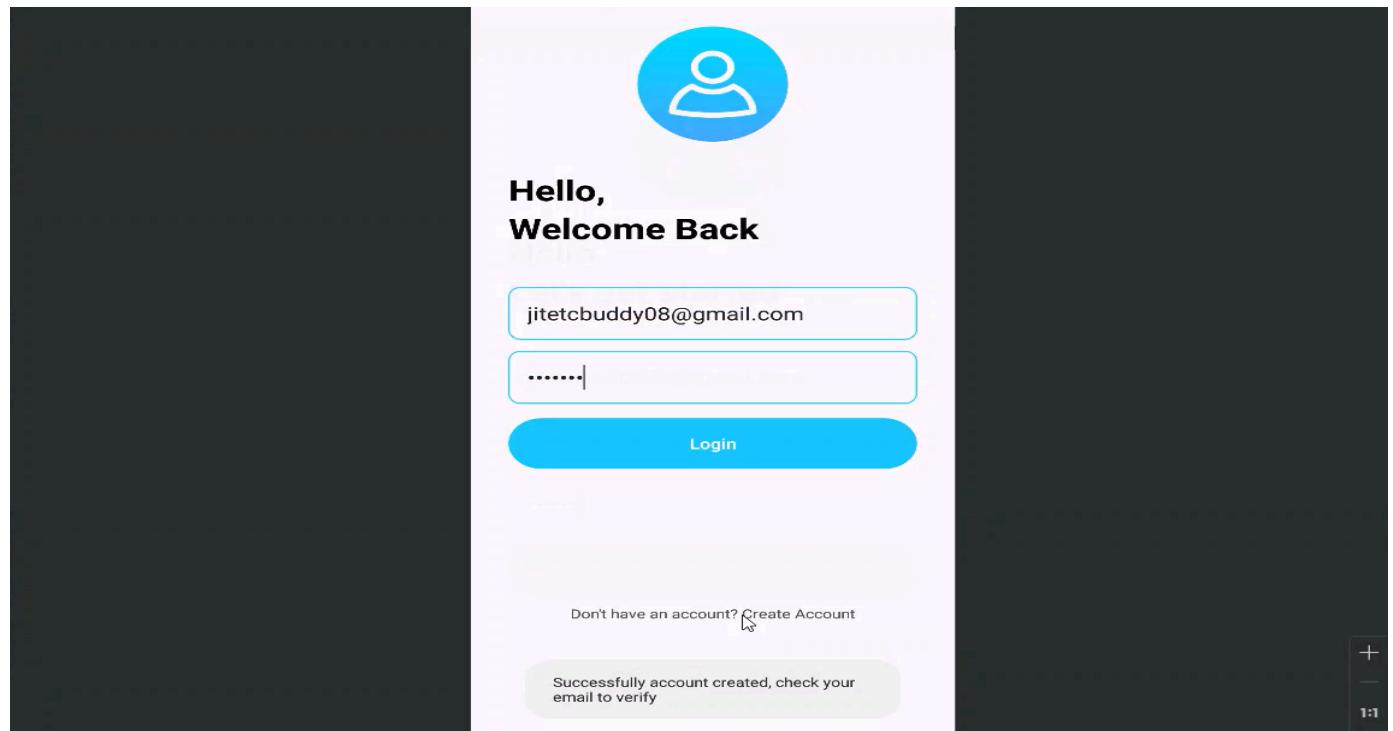
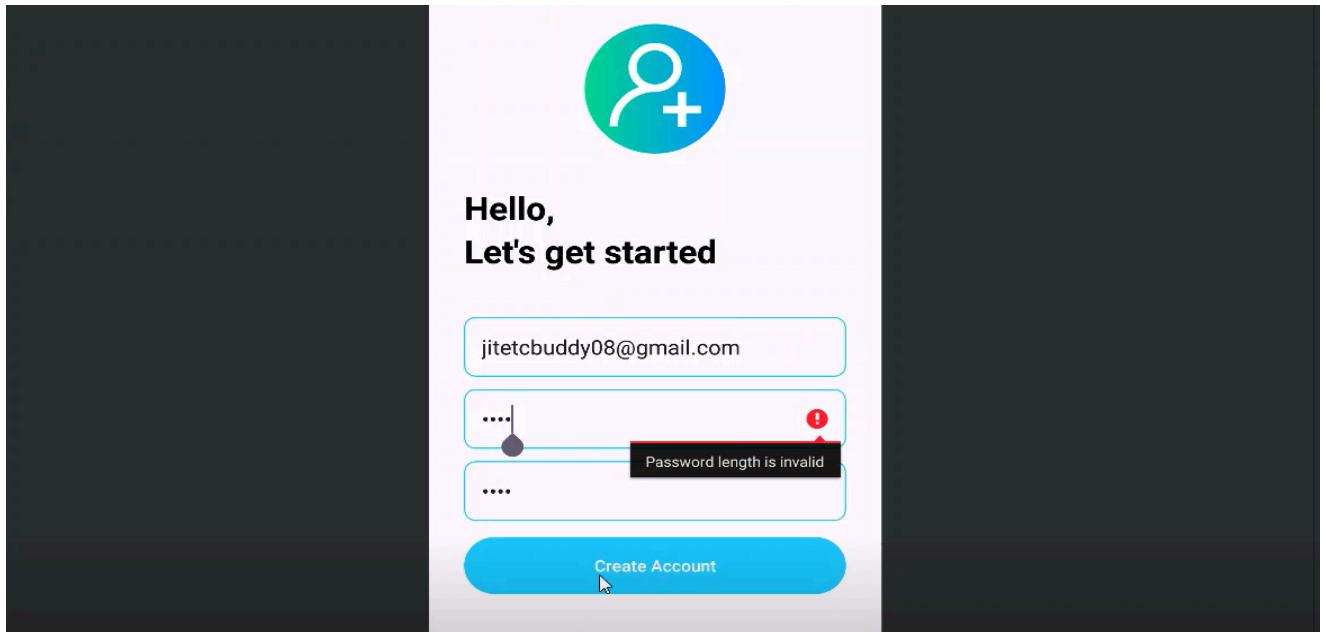
- After the splash screen, the **Login Page** appears, prompting the user to enter an **Email and Password**.
- If the user has not created an account yet, attempting to log in will result in an error.



Creating an Account:

- Since the account does not exist yet, we proceed to **create a new account**.
- Enter a valid email address; if the format is incorrect, an error message “**Email is invalid**” is displayed.
- Enter a password; if the password is less than 6 characters, an error message “**Password length is invalid**” is shown.
- Once a valid email and password are entered, a **Toast message** appears: “**Successfully created account, check your email to verify.**”





Email Verification:

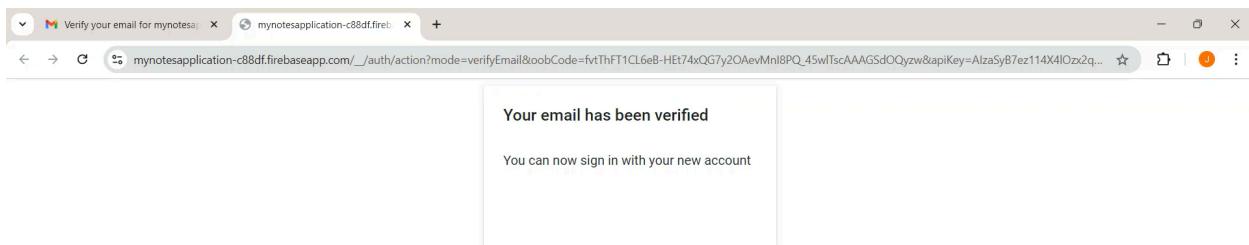
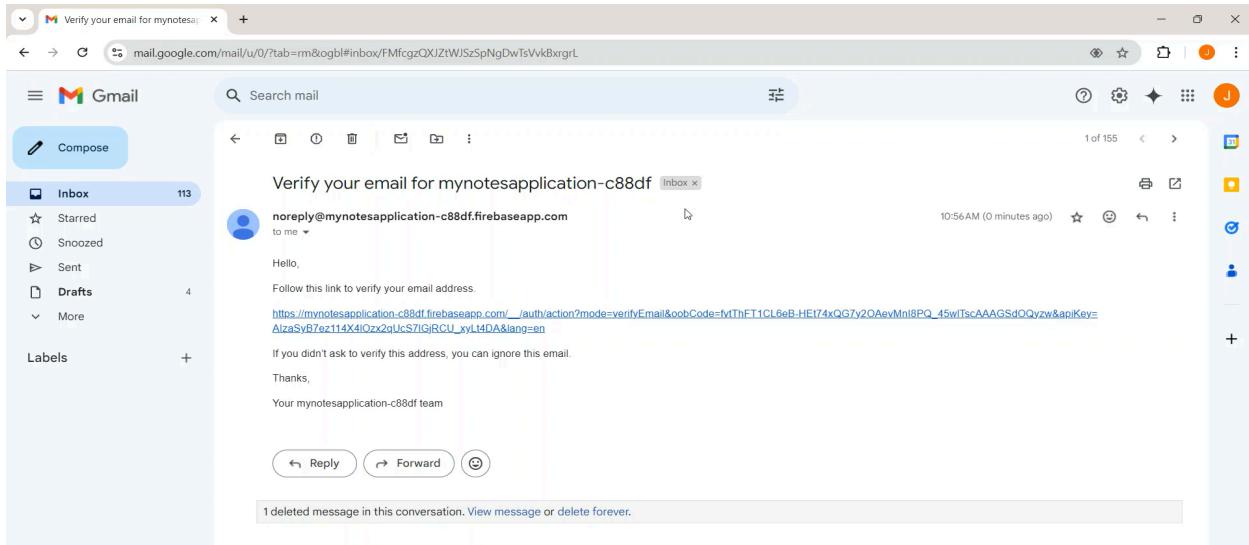
- Navigate to the **Firebase Authentication tab**, where the newly created user account details are now present.
- Check your email inbox, where you will receive a **verification link**.
- Click on the verification link, which opens a popup message: “**Your email has been verified.**”

The screenshot shows the Firebase console's Authentication section. On the left, there's a sidebar with Project Overview, Generative AI, Build with Gemini, Project shortcuts, Authentication (which is selected and highlighted in blue), Firestore Database, Product categories, Build, Run, Analytics, and All products. The main area is titled "Authentication" and has tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. Below the tabs is a search bar and a table displaying user information. One row is visible in the table:

Search by email address, phone number, or user UID	Providers	Created	Signed In	User UID
jitetcbuddy08@gmail.com		Oct 10, 2024	Oct 10, 2024	ESjNhDObXeqDtPu2cjMR...

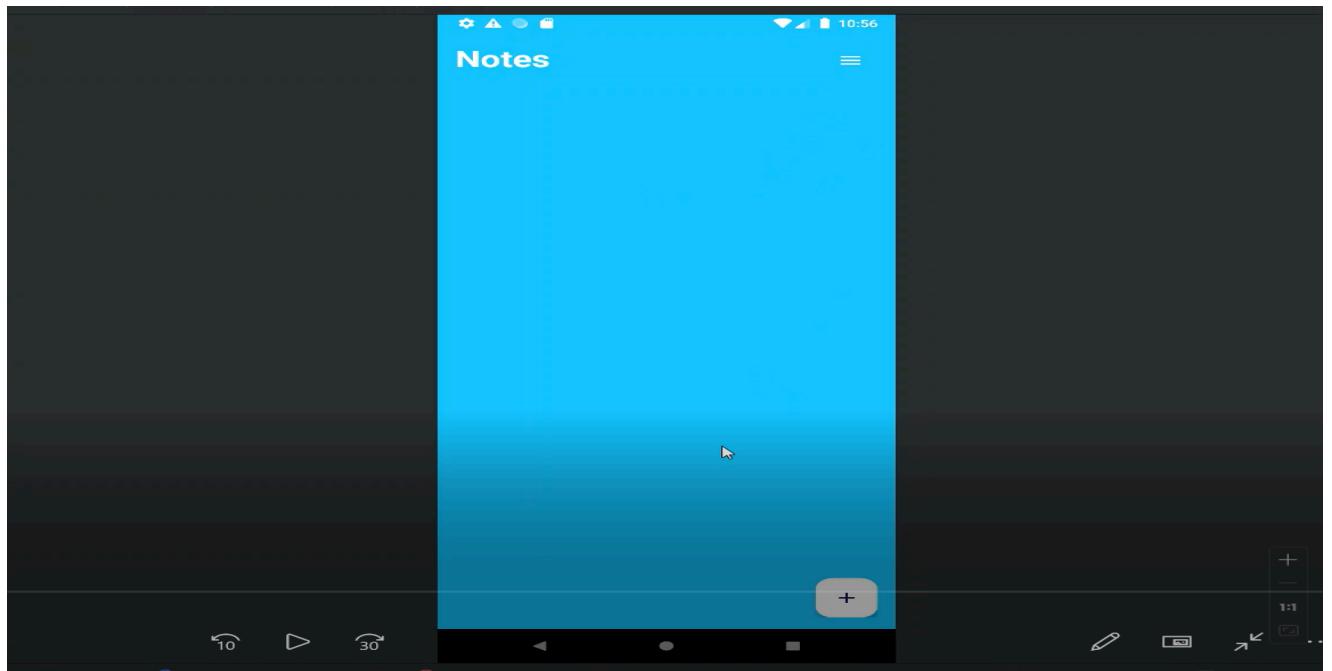
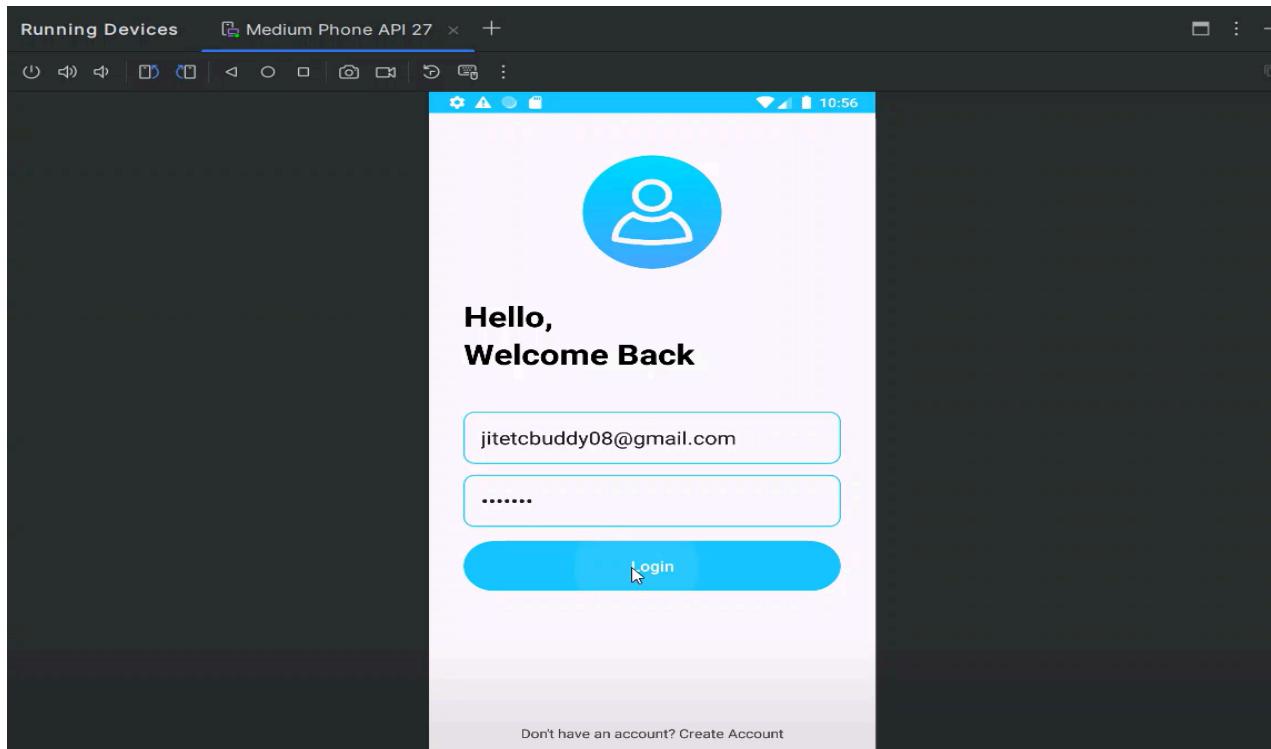
The screenshot shows the Gmail inbox. The sidebar on the left includes Compose, Inbox (114), Starred, Snoozed, Sent, Drafts (4), and More. Labels are listed below. The inbox lists several emails:

- A reply from "noreply" with the subject "Verify your email for mynotesapplication-c88df". It contains a verification link: <https://mynotesapplication-c88df.iam.gserviceaccount.com>. The date is Oct 1.
- An email from "Aniket Bourasee (vi.)" with the subject "Document shared with you: "1st october"" dated Oct 1.
- An email from "Naresh Technologies." with the subject "New announcement: "FREE ONLINE WORKSHOP Spring BOOT..."" dated Sep 28.
- A task reminder from "Aniket Bourasee" for "24th sept Tasks" dated Sep 24.
- An email from "Aniket, me 3" with the subject "Projects - Wipro_Android_Training.rar" dated Sep 21.
- An email from "Alison" with the subject "We can see JITETC's future. It's bright! - Discover a up and coming career for the next 10 years View in web brows..." dated Sep 20.



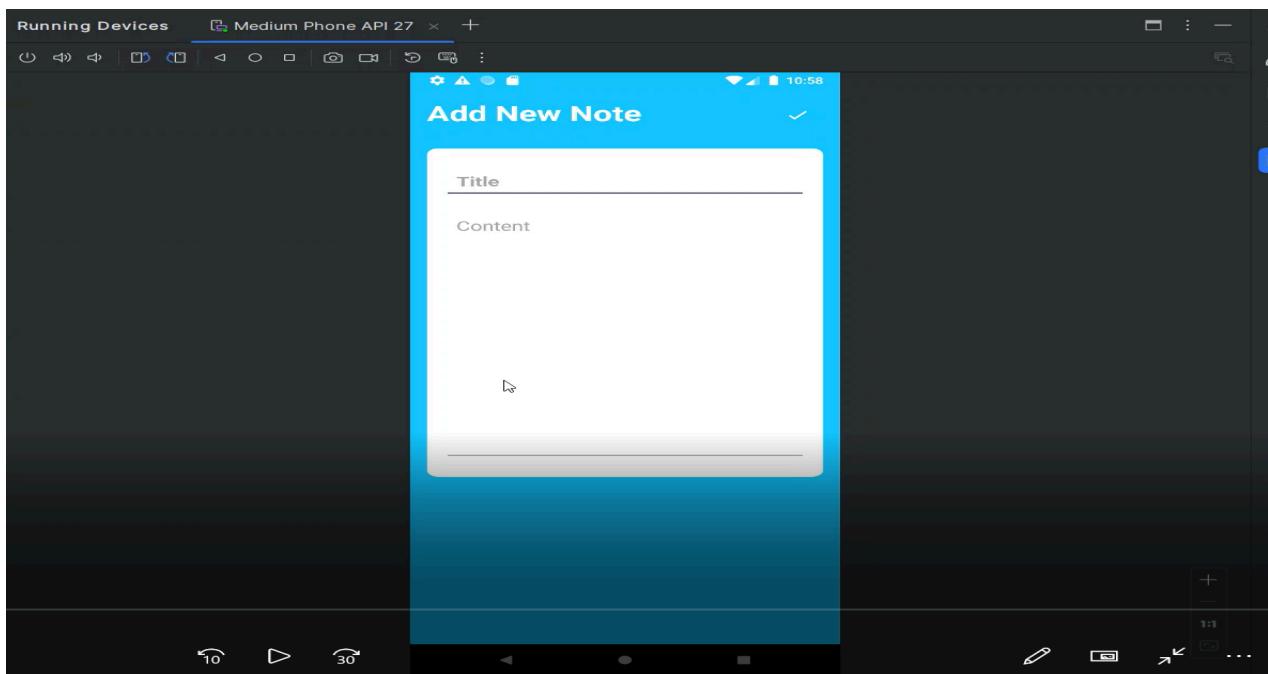
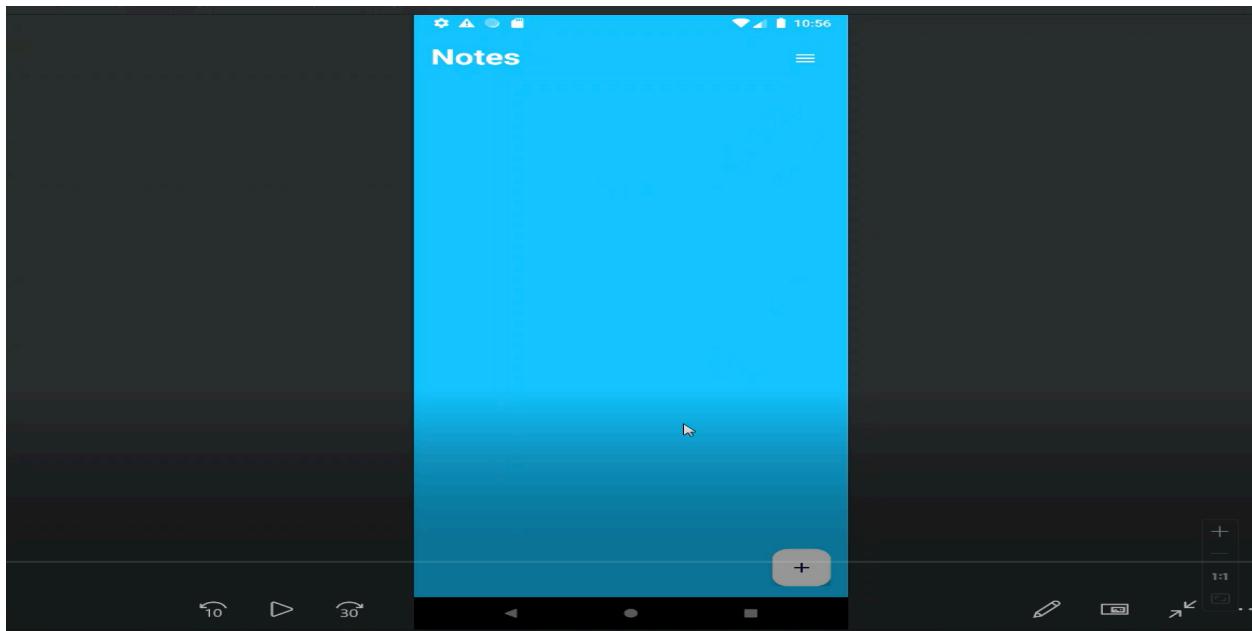
Logging In:

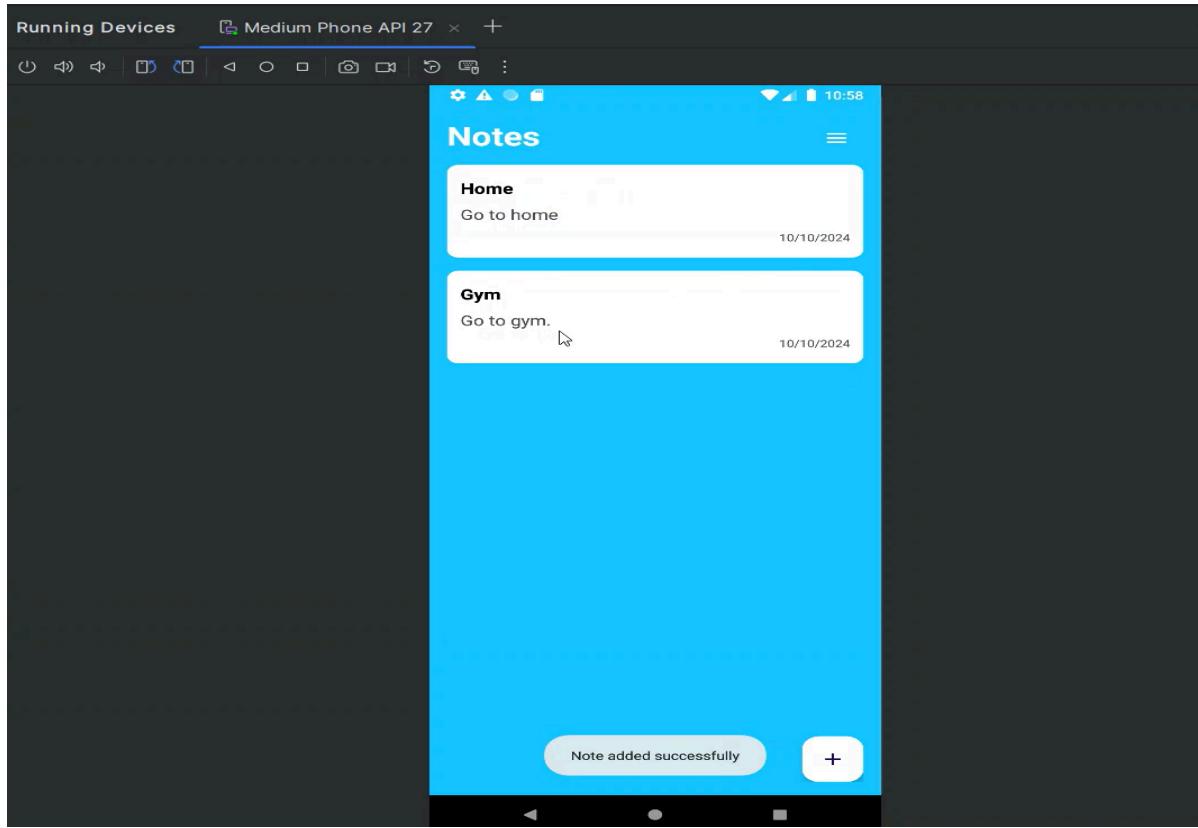
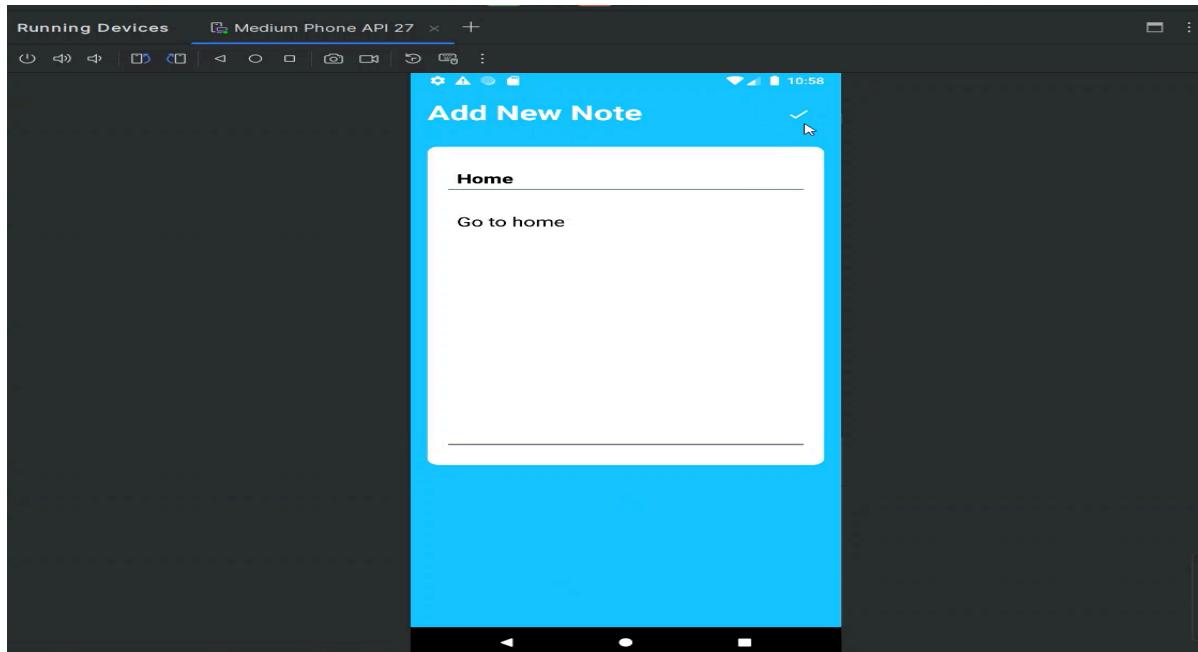
- Return to the app's **Login Page**. The email and password fields are already pre-filled.
- Simply click the **Login button** to log in.
- Now, the main screen of the app is opened.



Creating Notes:

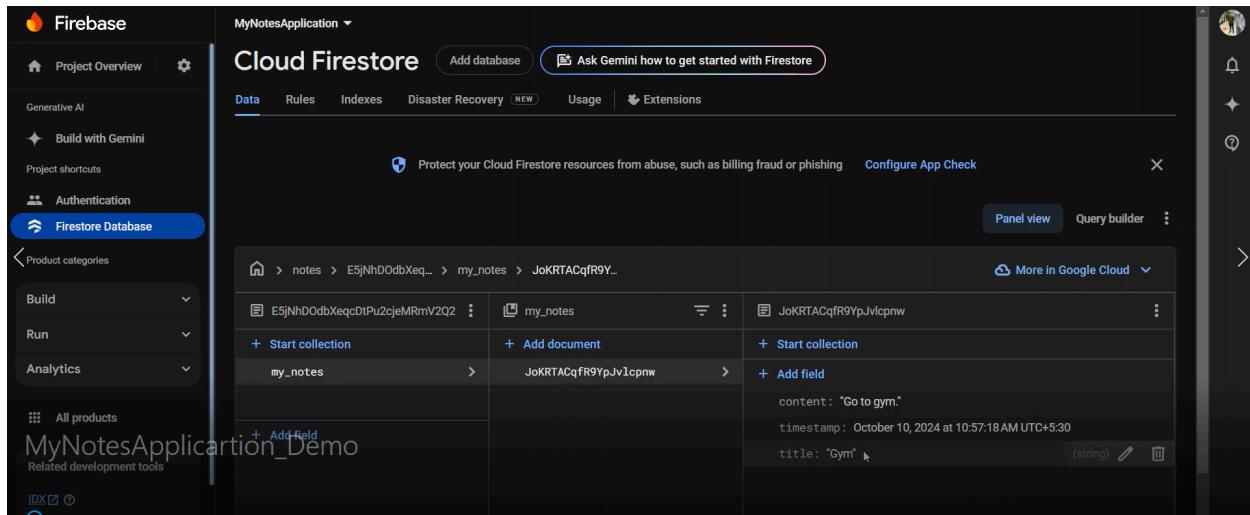
- On the main screen, there is a “+” icon at the bottom for creating notes.
- Click the “+” icon to navigate to the **Create Notes screen**.
- In this screen, enter the **title** and **content** of your note.
- After filling both fields, click the **done symbol** (checkmark) on the right side to save the note.
- A **Toast message** confirms the action: “**Note added successfully.**”



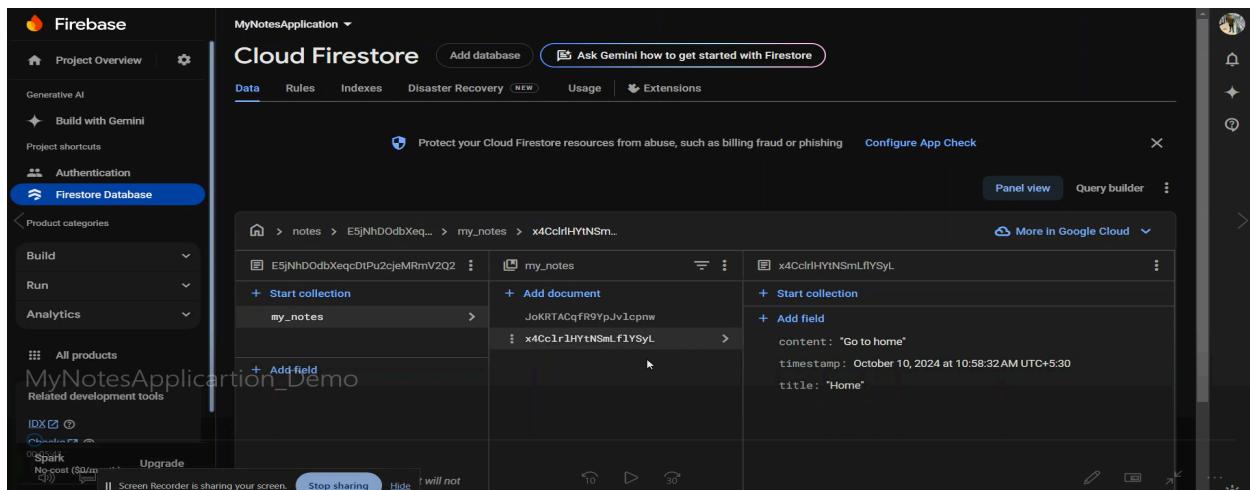


Checking the Database:

- Go to Firebase Firestore to check if the newly created note has been added.
- The note is now visible in the Firestore database.



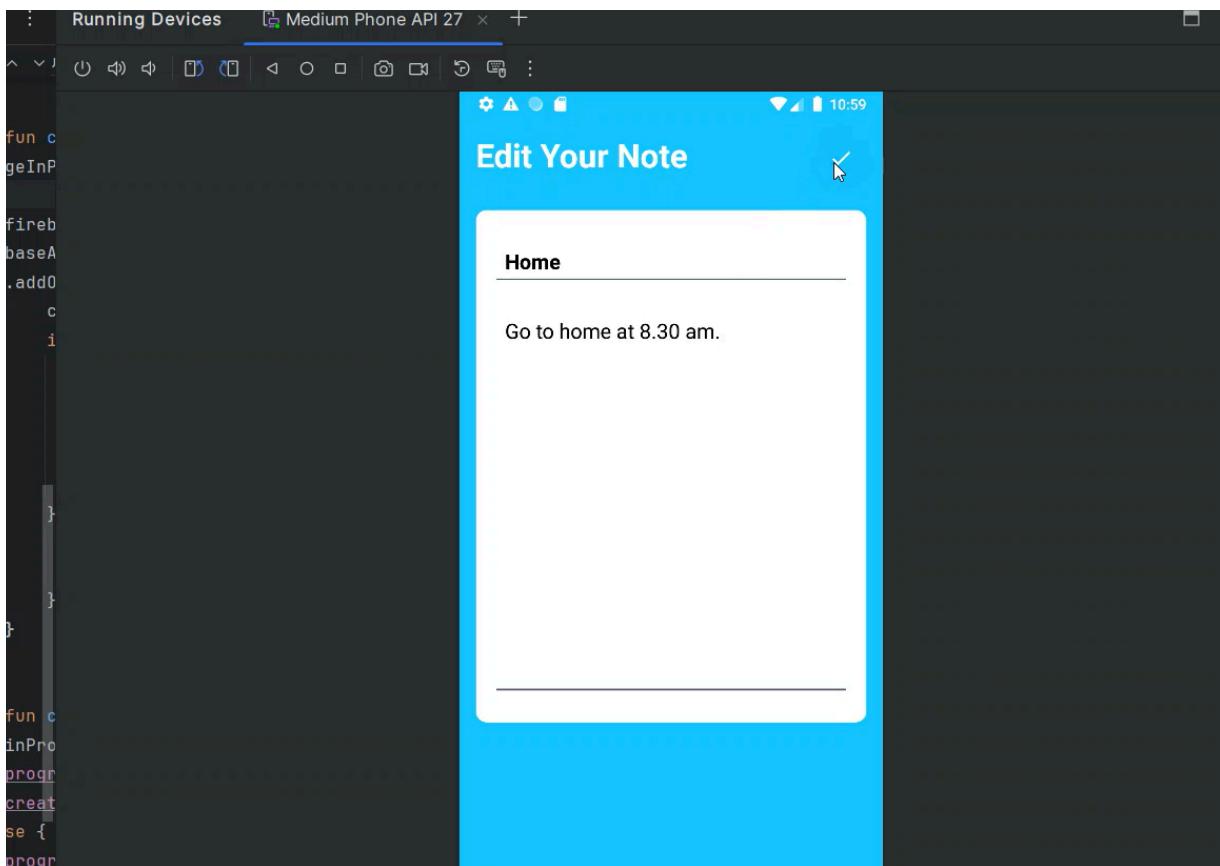
The screenshot shows the Firebase Cloud Firestore interface for a project named "MyNotesApplication_Demo". The left sidebar shows "Project Overview", "Generative AI", "Build with Gemini", "Project shortcuts", "Authentication", and "Firestore Database" (which is selected). The main area displays a document structure under the "notes" collection. The path is: notes > E5jNhD0dbXeqcDlPu2cjeMRmV2Q2 > my_notes > JoKRTACqfR9YpJv1cpnw. The document contains fields: content: "Go to gym.", timestamp: "October 10, 2024 at 10:57:18 AM UTC+5:30", and title: "Gym".

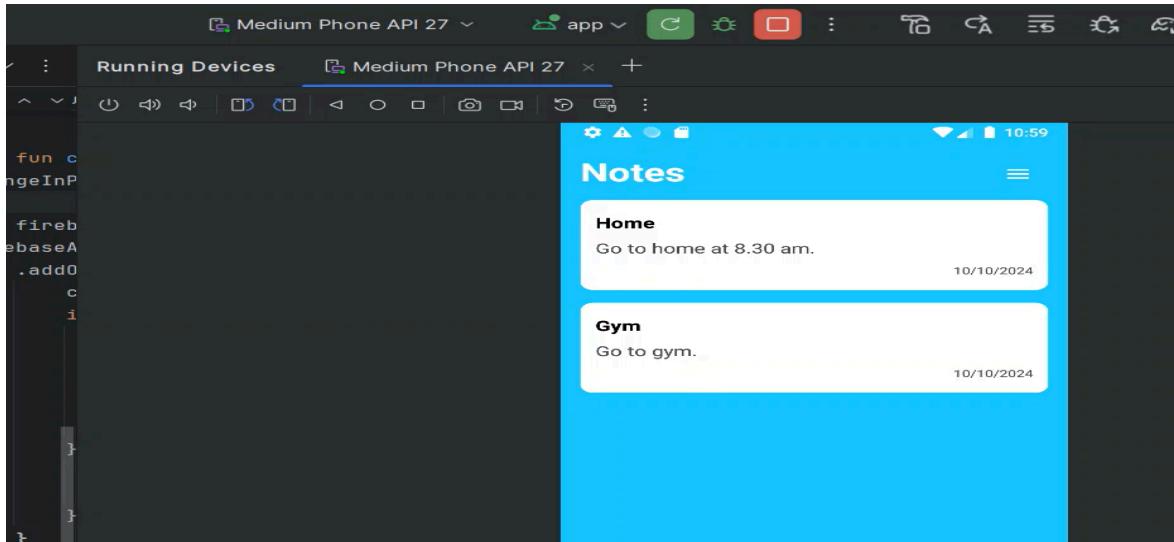


The screenshot shows the Firebase Cloud Firestore interface for the same project. The left sidebar shows "Project Overview", "Generative AI", "Build with Gemini", "Project shortcuts", "Authentication", and "Firestore Database" (selected). The main area displays a document structure under the "notes" collection. The path is: notes > E5jNhD0dbXeqcDlPu2cjeMRmV2Q2 > my_notes > x4CcIrlHytNSmLf1YSyL. The document contains fields: content: "Go to home.", timestamp: "October 10, 2024 at 10:58:32 AM UTC+5:30", and title: "Home".

Updating Notes:

- To update a note, select the note you wish to modify.
- Make the necessary changes, then click the **done symbol** to save the updated note.
- The updated note is reflected in the app, and the **Firebase database** is also updated.



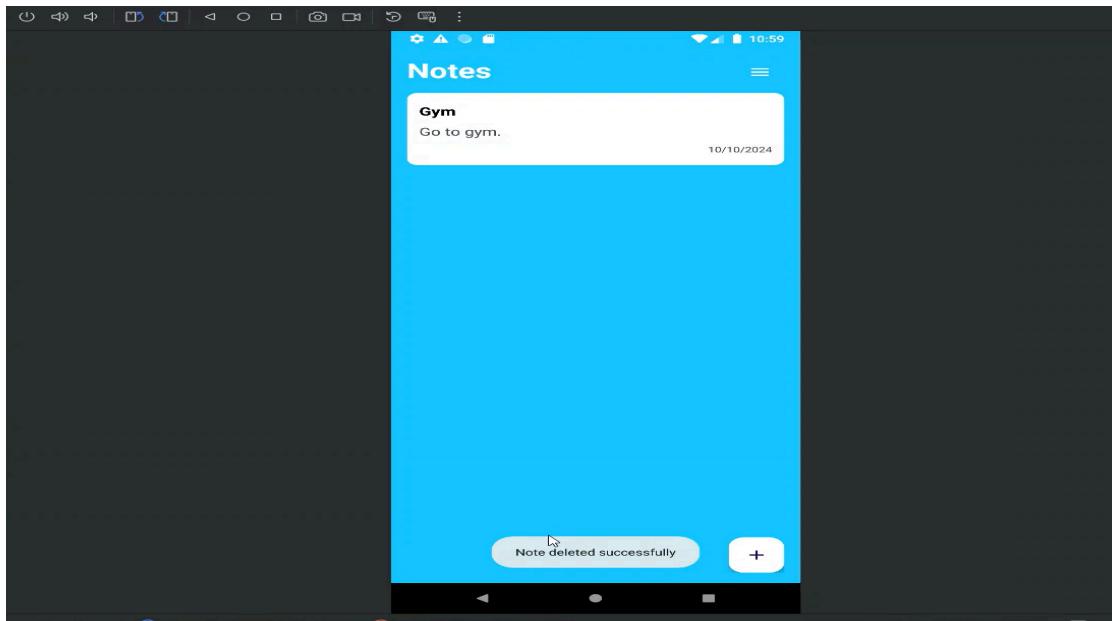
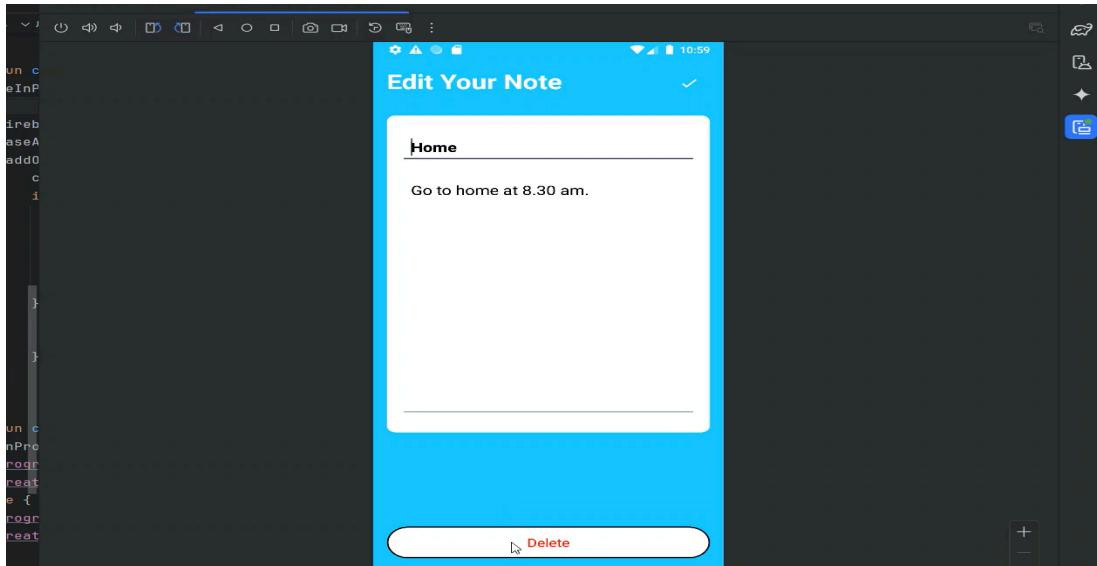


A screenshot of the Firebase Cloud Firestore console for the project "MyNotesApplication". The left sidebar shows the "Firestore Database" section selected. The main view shows a hierarchical list of documents under the "notes" collection. One specific document, "x4CcrlHYtNSmLfiYSyL", is expanded, showing its fields: "content" (string) with value "Go to home at 8.30 am.", "timestamp" (Timestamp) with value "October 10, 2024 at 10:59:09 AM UTC+5:30", and "title" (string) with value "Home". A "Configure App Check" button is visible above the document details.

Field	Type	Value
content	(string)	"Go to home at 8.30 am."
timestamp	(Timestamp)	October 10, 2024 at 10:59:09 AM UTC+5:30
title	(string)	"Home"

Deleting Notes:

- To delete a note, click the **delete** button.
- The note is removed from the app, and a **Toast message** confirms: **“Note deleted successfully.”**
- The note is also removed from the Firestore database.



The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation bar includes 'Project Overview', 'Generative AI', 'Build with Gemini', 'Project shortcuts', 'Authentication', and 'Firestore Database' (which is selected and highlighted in blue). Below these are sections for 'Build', 'Run', 'Analytics', 'All products', and 'Related development tools'. The main area displays a hierarchical view of a database collection: 'notes' > 'E5jNhD0dbXeqcDtPu2cjeMRmV2Q2' > 'my_notes' > 'JoKRTACqfR9YpJvlcpnw'. This document contains fields: 'content' (value: 'Go to gym.'), 'timestamp' (value: 'October 10, 2024 at 10:57:18 AM UTC+5:30'), and 'title' (value: 'Gym'). At the top of the page, there are tabs for 'Data', 'Rules', 'Indexes', 'Disaster Recovery', 'Usage', and 'Extensions'. A banner at the top right says 'Ask Gemini how to get started with Firestore'.

Logging Out:

- To log out of the app, click the **Logout** button.
- After logging out, the **Login Page** reappears, ready for the next session.

