



Kristu Jayanti College

AUTONOMOUS **Bengaluru**

Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

EMAIL ASSISTANT APP

Project Report

submitted in partial fulfillment of the requirements for
the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS (BCA)



Submitted By

Mohammed Yunus
22BCAE42

Under the guidance of

Prof. Sona K V

DEPARTMENT OF COMPUTER SCIENCE (UG)
BCA PROGRAMME
KRISTU JAYANTI COLLEGE (Autonomous)
K. Narayanapura, Kothanur P.O., Bengaluru – 560077



Kristu Jayanti College

AUTONOMOUS

Bengaluru

Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

DEPARTMENT OF COMPUTER SCIENCE (UG)

CERTIFICATE OF COMPLETION

This is to certify that the project entitled “**EMAIL ASSISTANT APP**” has been satisfactorily completed by **MOHAMMED YUNUS, 22BCAE42** in partial fulfillment of the award of the Bachelor of Computer Applications degree requirements prescribed by Kristu Jayanti College (Autonomous) Bengaluru (Affiliated to Bengaluru University) during the academic year 2024 - 25

Internal Guide

Head of the Department

Valued by Examiners

1: _____

Centre: Kristu Jayanti College

2: _____

Date:



Kristu Jayanti College

AUTONOMOUS Bengaluru

Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

DECLARATION

I, **MOHAMMED YUNUS (22BCAE42)** hereby declare that the project work entitled **EMAIL ASSISTANT APP** is an original project work carried out by me, under the guidance of **Prof. Sona K V.**

This project work has not been submitted earlier either to any University / Institution or any other body for the fulfillment of the requirement of a course of study.

Signature

MOHAMMED YUNUS

22BCAE42

Bengaluru

Date:

ACKNOWLEDGEMENT

“The success is the result of preparation, hard work, and learning from failure.”

It's my duty to acknowledge and thank the individuals who has contributed to the successful completion of the project.

I take this opportunity to express my profound and wholehearted thanks to **Rev. Fr. Dr. Augustine George, Principal** and **Rev. Fr. Dr. Lijo P Thomas, Vice Principal** and **Chief Financial Officer, Kristu Jayanti College, Bangalore.**

I am deeply grateful for invaluable support and encouragement given by **Dean Dr. Calistus Jude AL** throughout this journey. His steadfast guidance has been instrumental in shaping my academic path, and I am sincerely appreciative of his contributions to my development. I also extend my heartfelt thanks to our beloved **Head of the Department, Prof. Sevuga Pandian A, and Programme Coordinator Dr. Ranjitha M,** for their invaluable guidance and support. Additionally, I wish to express my deep gratitude to **Prof. Sona K V,** my mentor and guide, whose mentorship and profound insights have greatly contributed to my growth and learning journey. Your willingness to share expertise and offer constructive feedback has enriched my learning experience immensely.

It is my duty to express my thanks to all Teaching and Non-Teaching Staff members of Computer science department who offered me help directly or indirectly by their suggestions.

The successful completion of my project would not have been possible without my parent's Sacrifice, guidance, and prayers. I take this opportunity to thank everyone for their continuous Encouragement. I convey my thankfulness to all my friends who were with me to share my happiness and agony.

Last but not the least I thank almighty God for giving me strength and good health throughout my project and enabling me to complete it successfully.

TABLE OF CONTENT

1. Project Synopsis.....	1-2
1.1 Purpose	1
1.2 Goals.....	1-2
1.3 Intended Audience.....	2
2. Project Objectives	3
2.1 Modern User Interface and Robust Security	3
2.2 Reliable Email Connectivity and Intelligent Text Extraction.....	3
2.3 AI powered Summarization and Response Generation	3
3. Project Outline	4
3.1 User Registration Module.....	4
3.2 User Login Module	4
3.3 Gmail Connection Module	4
3.4 Welcome Page Module.....	4
3.5 Chat Module	4
3.6 Flow Diagram.....	5
4. Technologies Used	6
5. Screenshots.....	7-10
6. Project Features	11
6.1 Secure Authentication and User Management	11
6.2 Reliable Gmail Connectivity and Plain Text Email Retrieval	11
6.3 AI-Powered Summarization and Response Suggestions	11
6.4 ChatGPT-Style Conversational UI:	11
6.5 Real-Time Feedback and Error Handling	11
7. User Guide	12
8. Conclusion	13
9. References	14

1. PROJECT SYNOPSIS

1.1 Purpose:

The Email Assistant App is a modern Android application designed to help users efficiently manage their emails with the help of AI-powered features. The app provides a seamless experience for reading, summarizing, and responding to emails, integrating advanced AI models (now Gemini 2.0 Flash) for summarization and response suggestions. It is built with a focus on usability, security, and productivity for everyday email users. The main purpose of the Email Assistant App is to simplify email management by providing AI-generated summaries and response suggestions, reducing the time and effort required to process and reply to emails. By leveraging the latest advancements in AI, the app enables users to quickly understand the main points of their emails and respond with contextually appropriate replies, all within a secure and user-friendly environment. The app is designed to support daily email workflows with a modern, chat-like UI, real-time feedback, and robust error handling. It enforces strong security practices, including 2-Factor Authentication (2FA) and app password requirements for Gmail accounts, ensuring that user data remains protected. Whether for professionals, students, or anyone seeking to manage their inbox more efficiently, the Email Assistant App delivers a comprehensive solution that combines advanced technology, intuitive design, and practical features to enhance productivity and reduce email fatigue. With its seamless integration of AI, secure authentication, and responsive interface, the app stands out as a valuable tool for modern email management on Android devices.

1.2 Goals:

- Deliver a secure and user-friendly email management experience that prioritizes both privacy and ease of use. The app ensures robust authentication and intuitive navigation for all users.
- Integrate advanced AI (Gemini 2.0 Flash) to provide accurate, context-aware email summaries and response suggestions. This empowers users to process and reply to emails more efficiently.

- Support daily email workflows with a modern, chat-like UI that mimics popular messaging platforms. Real-time feedback and smooth transitions enhance productivity and user satisfaction.
- Maintain high reliability and availability by implementing fallback strategies and clear error handling. Users are always informed of connection issues or authentication problems, ensuring trust and transparency.
- Continuously improve the app's features and design based on user feedback and evolving technology standards. Regular updates ensure the app remains relevant, secure, and effective for all audiences.

1.3 Intended Audience:

- **Busy Professionals:** Individuals who receive a high volume of emails and need to quickly understand, organize, and respond to important messages with minimal effort.
- **Students and Academics:** Users who manage academic, project, or organizational emails and benefit from concise AI-generated summaries and suggested replies.
- **Remote Workers and Teams:** People collaborating across locations who require efficient email triage, clear communication, and streamlined response workflows.
- **Security-Conscious Users:** Individuals who prioritize privacy and data protection, seeking secure authentication, 2FA, and app password support for their email accounts.
- **Anyone Seeking Productivity:** Users who want to reduce email fatigue, save time, and leverage AI to optimize their daily email management on Android devices.

Overall, the Email Assistant App serves a diverse audience looking to enhance their email experience through advanced AI, secure connectivity, and a modern, chat-like interface. Its features are tailored to help users stay organized, respond efficiently, and maintain control over their inbox with confidence.

2. PROJECT OBJECTIVES

The Email Assistant App is designed with three core objectives, each focused on delivering a secure, user-friendly, and intelligent email management experience:

2.1 Modern User Interface and Robust Security

- Deliver a seamless, visually appealing user experience using Jetpack Compose and Material Design 3, with a chat-like interface for summaries and responses.
- Ensure the highest standards of security and privacy, including secure authentication (Firebase), 2FA, and app password enforcement for Gmail accounts.

2.2 Reliable Email Connectivity and Intelligent Text Extraction

- Provide robust and reliable connectivity to Gmail accounts using the IMAP protocol, supporting only secure app passwords and 2FA.
- Extract only the most relevant plain text content from emails, excluding attachments and HTML, using custom parsing and formatting logic.

2.3 AI-Powered Summarization and Response Generation

- Leverage Gemini 2.0 Flash for all email summarization and response generation, ensuring accurate, context-aware, and actionable AI assistance.
- Offer fallback and backup model support for legacy reliability, but prioritize Gemini for all user-facing features.
- Present AI-generated summaries and responses in a chat-like UI, with real-time feedback, copy-to-clipboard, and clear error handling to enhance productivity and reduce email fatigue.

3. PROJECT OUTLINE

3.1 Modules:

3.1 User Registration Module:

Handles secure user sign-up, collecting name, email, and password, and storing user profile data in Firebase Firestore. Ensures new users are properly authenticated and onboarded before accessing any app features.

3.2 User Login Module:

Manages user authentication using Firebase Authentication, supporting secure login and password reset. Only authenticated users can proceed to connect their Gmail account and use the app.

3.3 Gmail Connection Module:

Facilitates secure connection to the user's Gmail account using IMAP, enforcing 2FA and app password requirements. Fetches unread emails, extracts sender, subject, date, and plain text content, and manages connection errors and feedback.

3.4 Welcome Page Module:

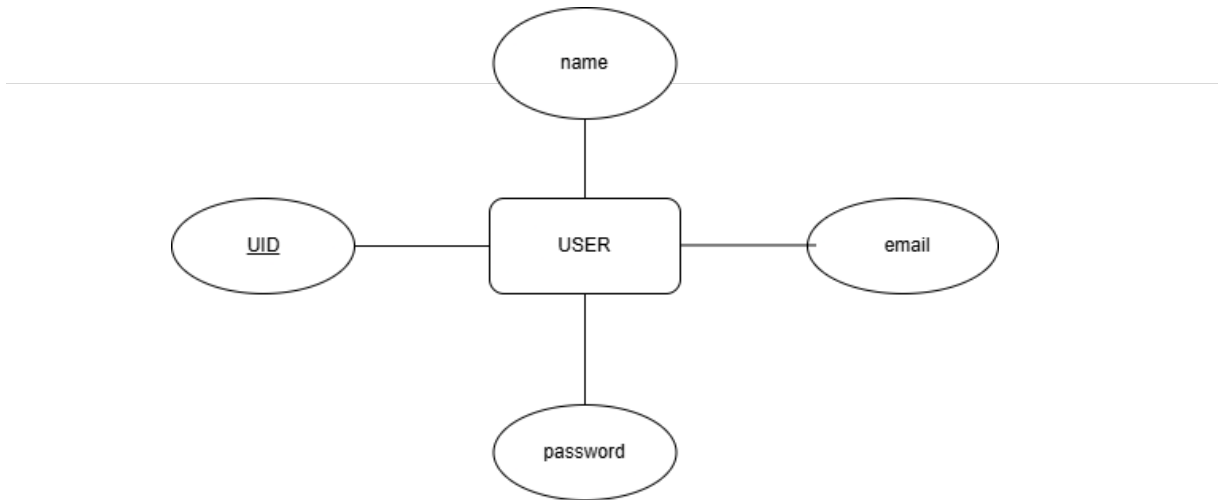
Acts as the main dashboard after login and Gmail connection, displaying an animated card with a user guide and a 'Fetch Daily Report' button. This button invokes the fetch email functionality, while the page also presents user profile info, navigation options, and the entry point to the chat experience.

3.5 Chat Module:

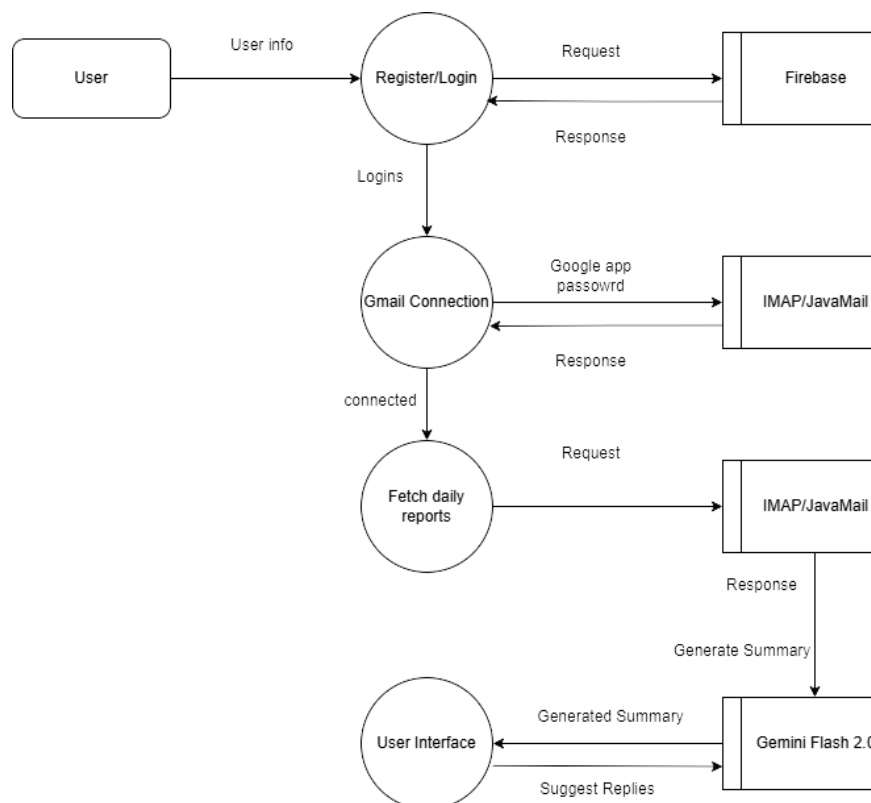
Implements the chat-like interface for AI-powered email summarization and response generation using Gemini 2.0 Flash. Displays email summaries and AI responses as chat bubbles, supports copy-to-clipboard, and provides real-time feedback and error handling.

3.6 Flow Diagram

ER Diagram:



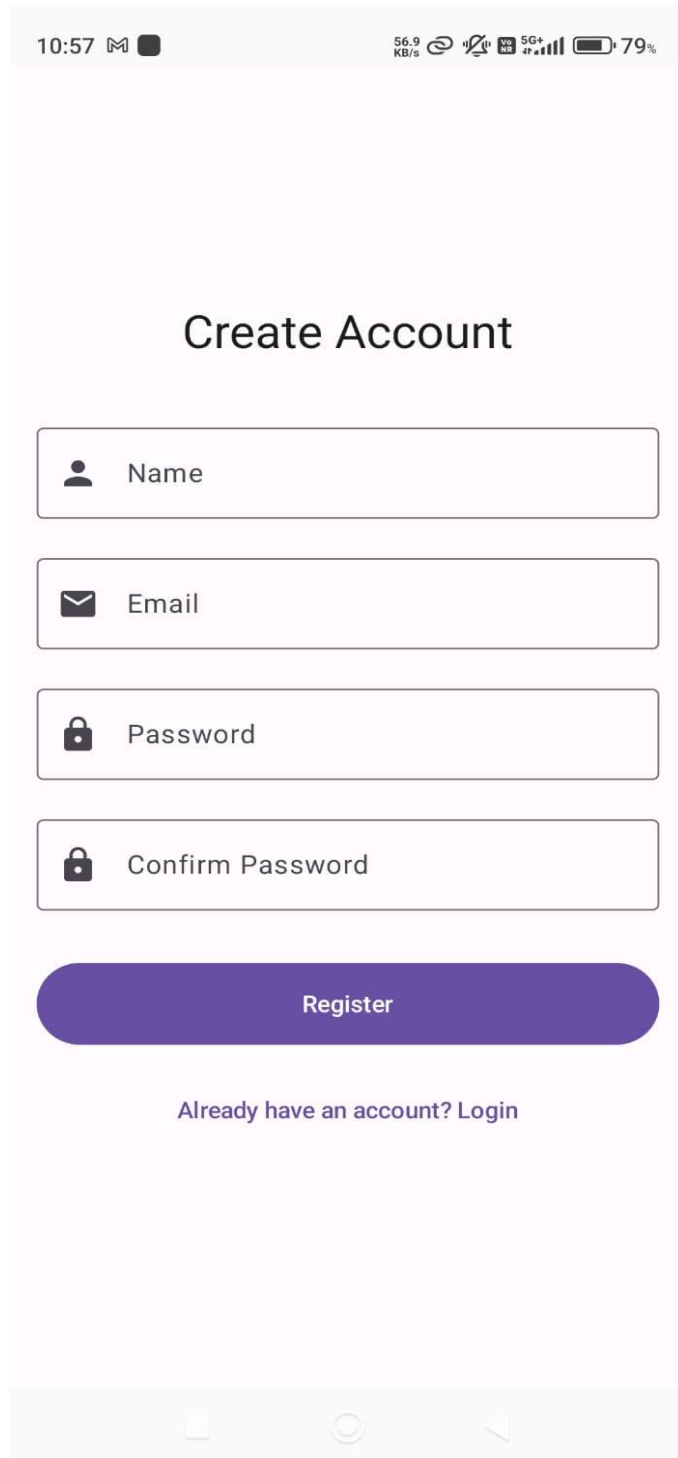
Data Flow Diagram (Level 1):



4 TECHNOLOGIES USED

Programming Language	Kotlin
Development Platform	Android Studio
Database	Firebase
Other Tools	Firebase Authentication, Gemini 2.0 Flash API, IMAP (JavaMail), OkHttp, Coroutines, Material Design 3

5 SCREENSHOTS



10:57 M

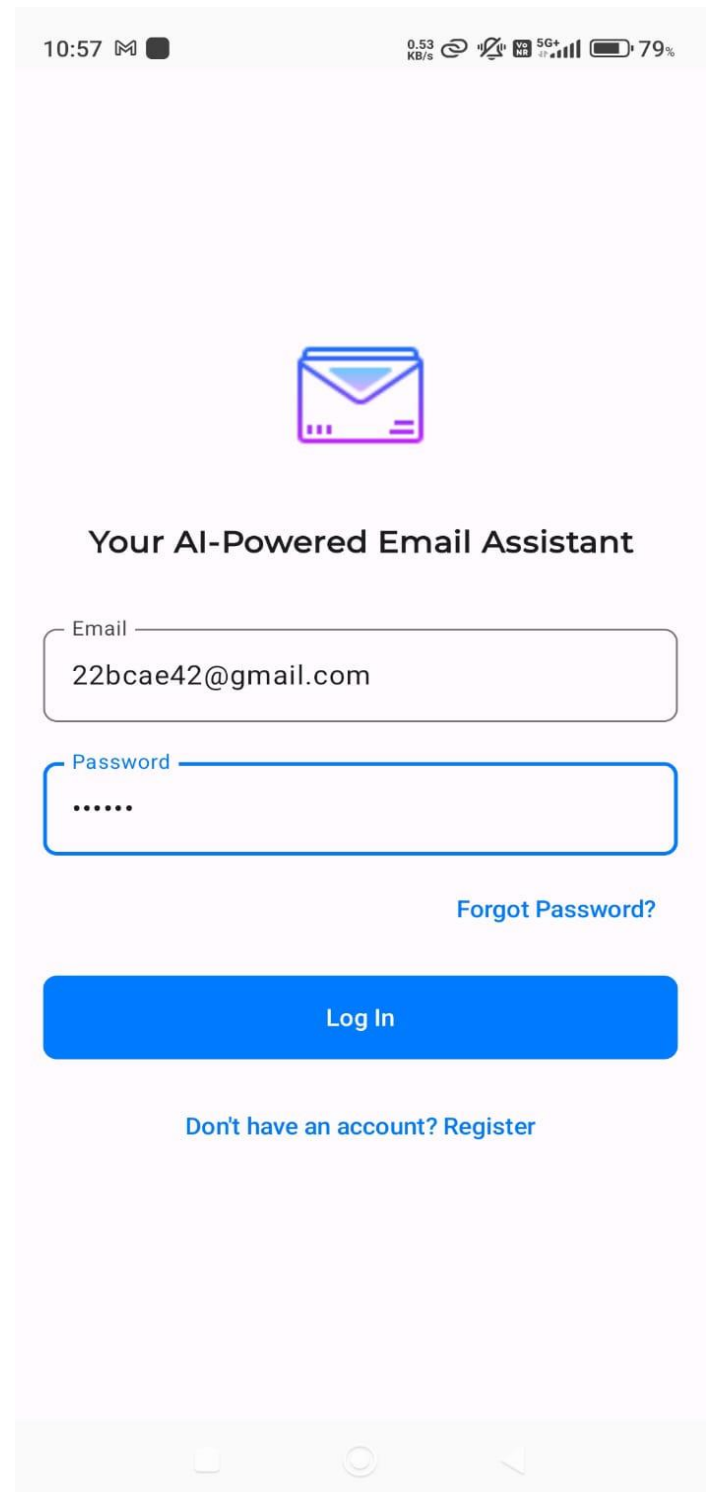
56.9 KB/s 5G+ 79%

Create Account

[Register](#)


[Already have an account? Login](#)

Figure 5.1 Register Page



10:57 M

0.53 KB/s 5G+ 79%



Your AI-Powered Email Assistant

22bcae42@gmail.com

.....

[Forgot Password?](#)

[Log In](#)

[Don't have an account? Register](#)

Figure 5.2 Login page

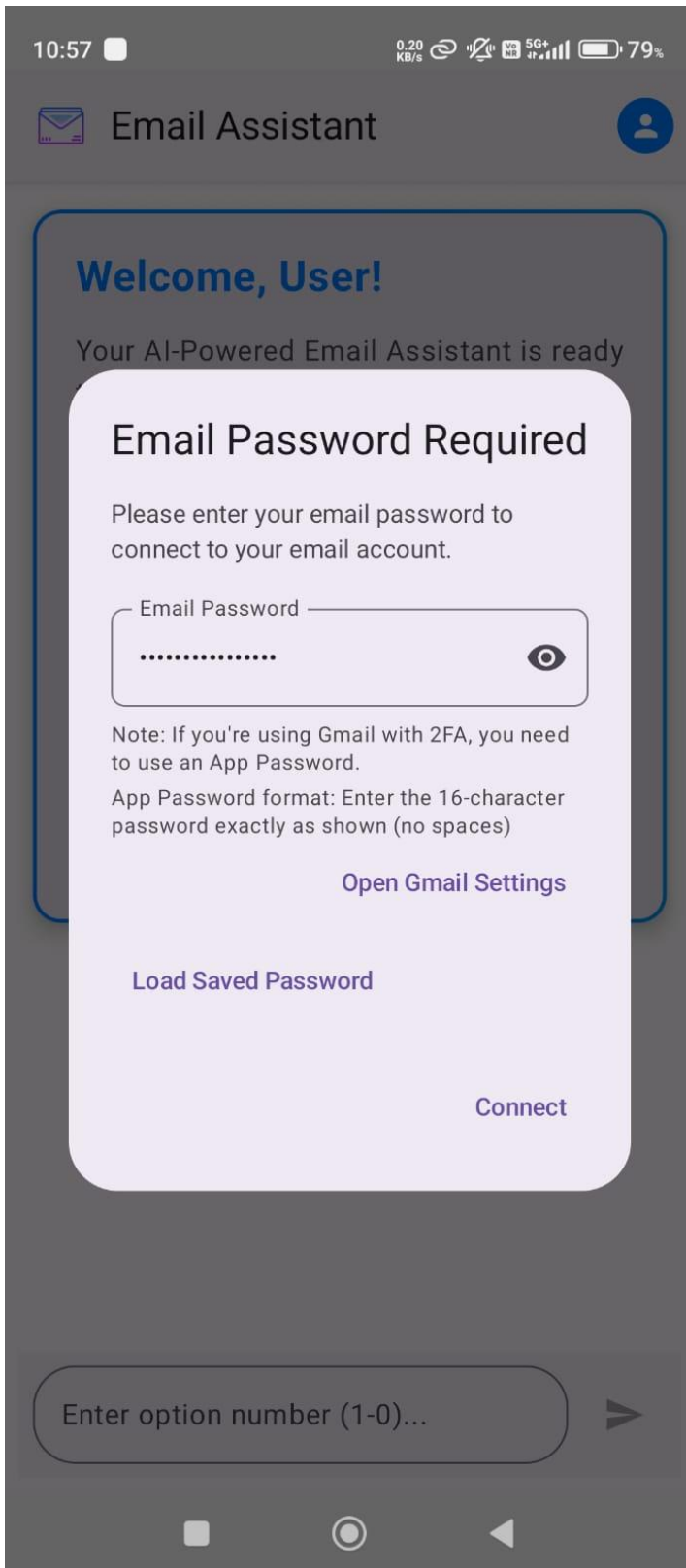


Figure 5.3 Gmail Connection

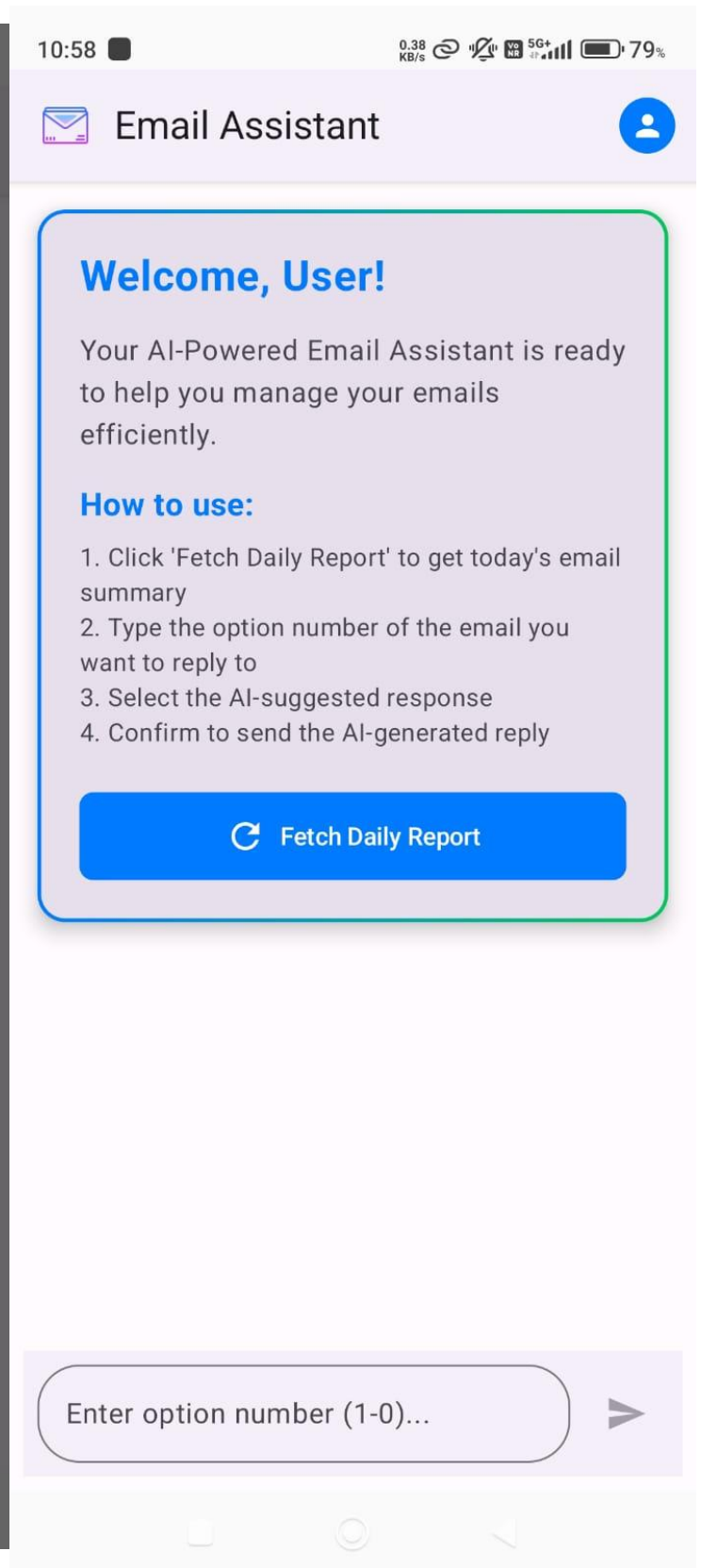


Figure 5.4 Welcome Page

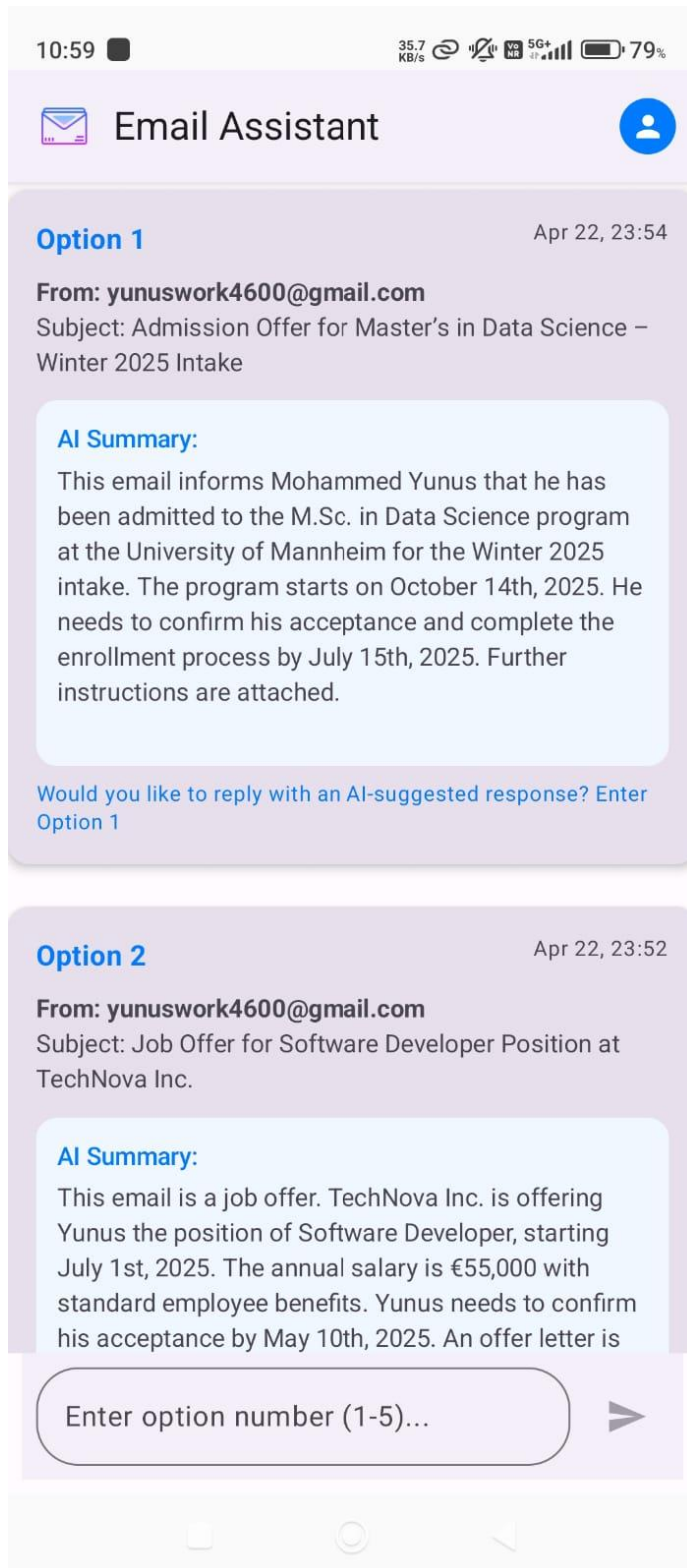


Figure 5.5 Email Summaries

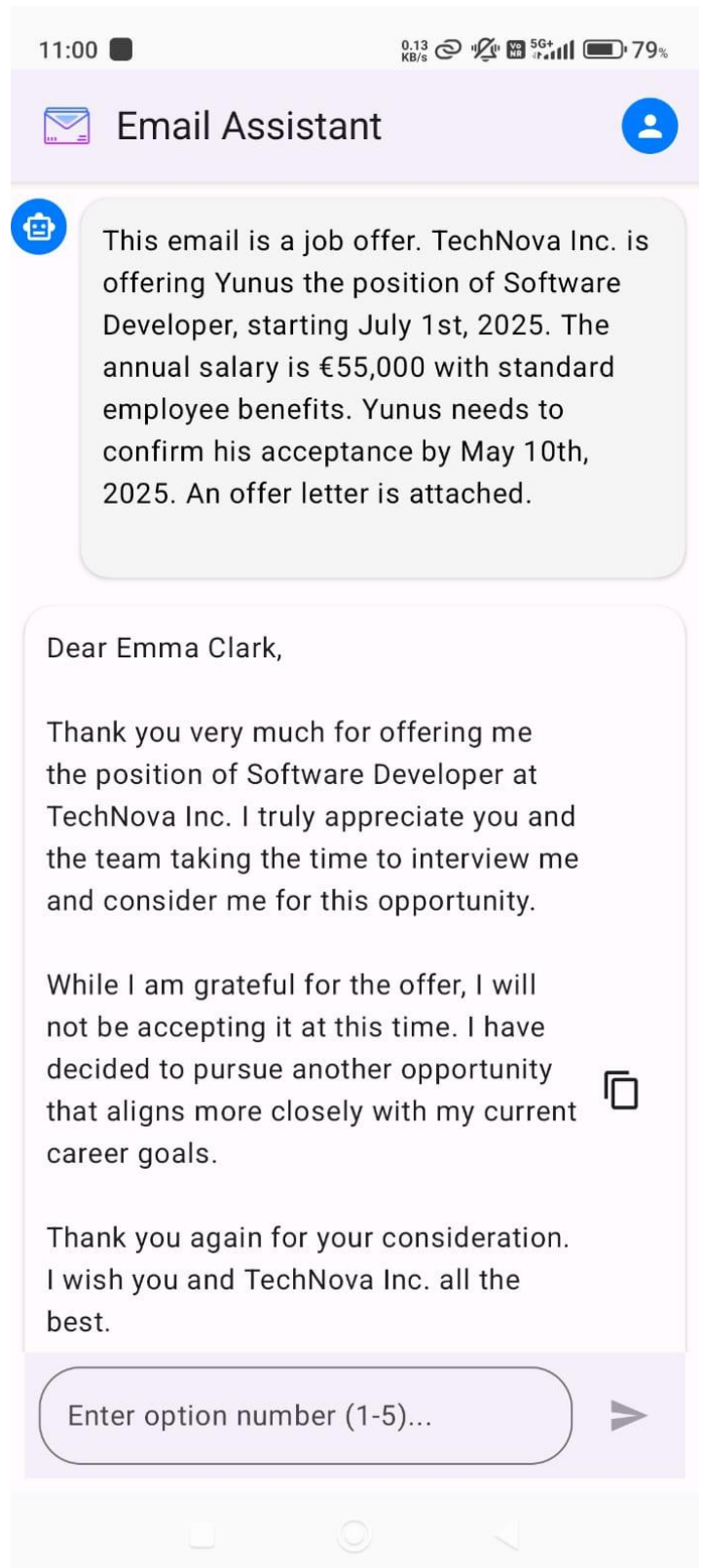


Figure 5.6 Suggested Responses

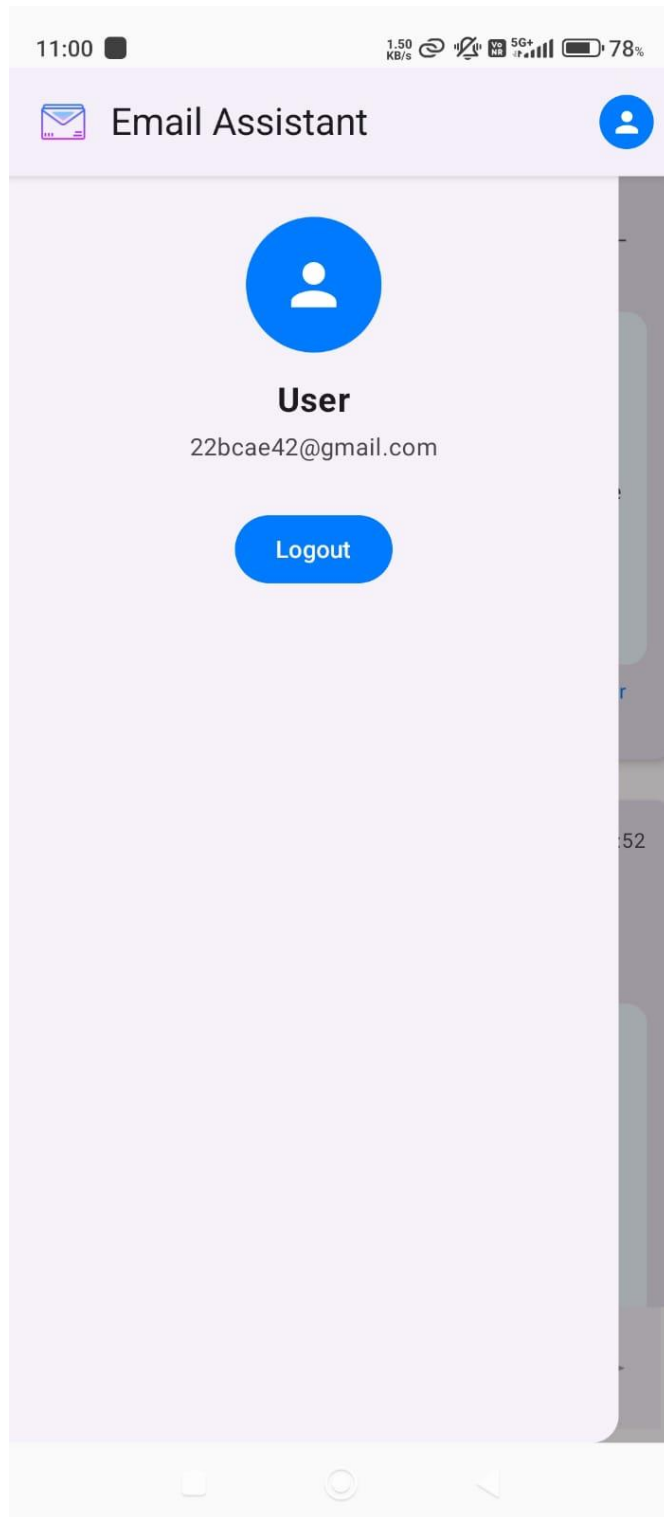


Figure 5.7 Side Bar

6 PROJECT FEATURES

6.1 Secure Authentication and User Management:

Provides robust user registration, login, and password reset using Firebase Authentication. Stores user profile data (name, email) securely in Firebase Firestore and ensures only authenticated users can access email features.

6.2 Reliable Gmail Connectivity and Plain Text Email Retrieval:

Connects to Gmail accounts using IMAP, requiring 2FA and app password for compliance with Google security policies. Fetches unread emails and extracts only the plain text content, ignoring attachments and HTML, and handles connection errors gracefully.

6.3 AI-Powered Summarization and Response Suggestions:

Uses Gemini 2.0 Flash API to generate concise, human-readable summaries for each email. Provides AI-generated response suggestions (accept/reject for offers) to streamline the reply process, with fallback and backup model support for legacy reliability.

6.4 ChatGPT-Style Conversational UI:

The main page features a chat-like interface, with summaries and AI responses as chat bubbles. The input textbox is fixed at the bottom, always accessible, and the chat area is scrollable; each AI response card includes a copy-to-clipboard icon for easy copying.

6.5 Real-Time Feedback and Error Handling:

Delivers real-time feedback for all user actions, including authentication, email fetching, and AI response generation. Utilizes toasts, error cards, and clear messaging to guide users and resolve issues promptly.

7 USER GUIDE

Installation Instructions:

Step 1: Clone the repository and open it in Android Studio.

Step 2: Add your Firebase configuration file (google-services.json) to the app directory.

Step 3: Add your Gemini 2.0 Flash API key to the local.properties file in the project root as GEMINI_API_KEY=your_api_key_here.

Step 4: Build and run the app on an emulator or Android device.

Step 5: Ensure your device has an active internet connection for authentication and email fetching.

Step 6: (For Gmail users) Enable 2-Factor Authentication (2FA) in your Google Account and generate an app password. Use this app password (not your regular Gmail password) to connect your email account in the app.

User Interface Guide:

- **Login/Register:** Start by creating a new account or logging in with your existing credentials. Registration requires your name, email, and password. Login uses Firebase Authentication for security.
- **Password Reset:** If you forget your password, use the 'Forgot Password' option on the login screen to receive a reset link via email.
- **Connect Email:** After logging in, you will be prompted to enter your Gmail app password. The app will not accept standard Gmail passwords due to Google security policies. Follow the in-app instructions to generate and use an app password if you have 2FA enabled.
- **Fetch Emails:** Tap the 'Fetch Daily Report' button on the welcome page to retrieve unread emails from your Gmail inbox. The app will connect via IMAP, fetch only the plain text content, and ignore attachments and HTML.
- **View Summaries:** Each fetched email is displayed as a card with sender, subject, date, and an AI-generated summary. This allows you to quickly review the main points of each email without reading the full content.
- **AI Responses:** To generate a reply, enter the option number of the email you wish to respond to. The app will provide AI-generated response suggestions (accept/reject for offers). Review and select a response to use as your reply.
- **Refresh Emails:** Use the refresh option in the email list screen to update your inbox and fetch new unread emails.
- **Logout:** Use the menu or profile drawer to securely log out of your account at any time.
- **Error Handling:** The app provides real-time feedback and clear error messages for connection issues, authentication failures, and invalid input. Follow the on-screen instructions to resolve any issues.

8 CONCLUSION

Developing and working on the Email Assistant App has been an immersive and rewarding journey, marked by both challenges and achievements. From the initial concept to the final implementation, every stage has reflected my dedication and passion for creating a comprehensive solution for email management on mobile devices.

The Email Assistant App represents the culmination of countless hours spent coding, testing, and refining to deliver a seamless and intuitive user experience. As a solo developer, I have approached each feature with the end user in mind, striving to anticipate their needs and simplify the complexities of email management through advanced AI and thoughtful design.

Throughout the development process, I have encountered a variety of challenges, from technical hurdles in integrating secure authentication and AI models, to design considerations for a modern, chat-like interface. Each obstacle has served as an opportunity for growth, strengthening my skills as a developer and problem solver.

As I prepare to share the Email Assistant App with users, I am excited about the positive impact it will have on professionals, students, and anyone seeking to manage their inbox more efficiently. With its intuitive interface, robust security, and powerful AI features, the app aims to empower users to handle their emails with ease and confidence.

In summary, the development of the Email Assistant App has been a solo journey defined by determination, innovation, and perseverance. As I embark on this new chapter of sharing my creation with the world, I am confident that the Email Assistant App will become an indispensable tool for modern email management.

9 REFERENCE

1. - Jetpack Compose Documentation: <https://developer.android.com/jetpack/compose>
2. - Firebase Authentication: <https://firebase.google.com/docs/auth>
3. - Firebase Firestore: <https://firebase.google.com/docs/firestore>
4. - Gemini 2.0 Flash API: <https://gemini.flash.api/docs>
5. - JavaMail API: <https://javaee.github.io/javamail/>
6. - Android IMAP Email Handling: <https://developer.android.com/training/basics/data-storage/databases>
7. - OkHttp: <https://square.github.io/okhttp/>
8. - Material Design 3: <https://m3.material.io/>
9. - Google App Passwords: <https://support.google.com/accounts/answer/185833>