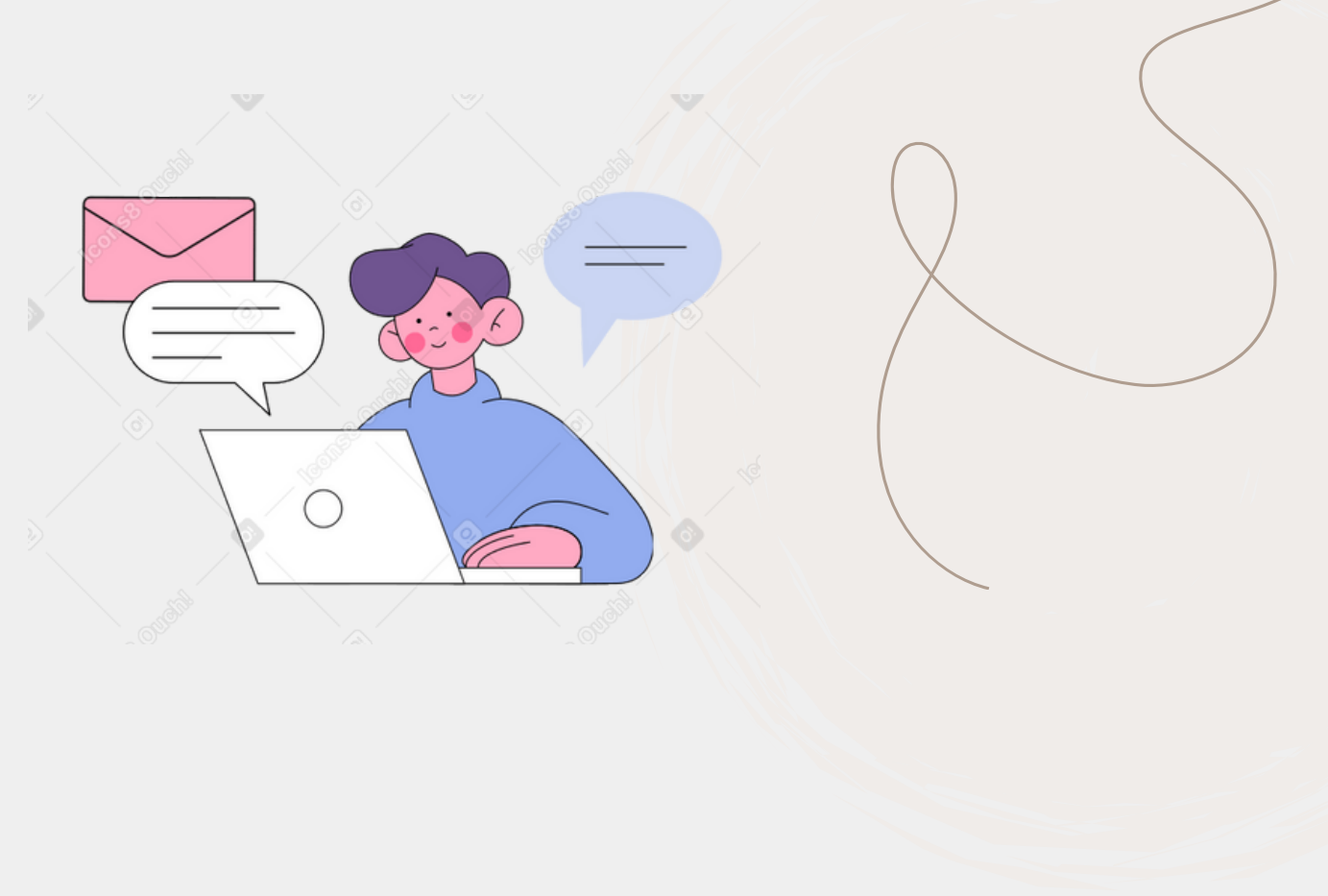


CHAT APPLICATION WE_CHAT



TEAM -D10

VADA GOURI HANSIKA REDDY CB.SC.U4AIE23304

MALAVIKA S PRASAD-CB.SC.U4AIE23315

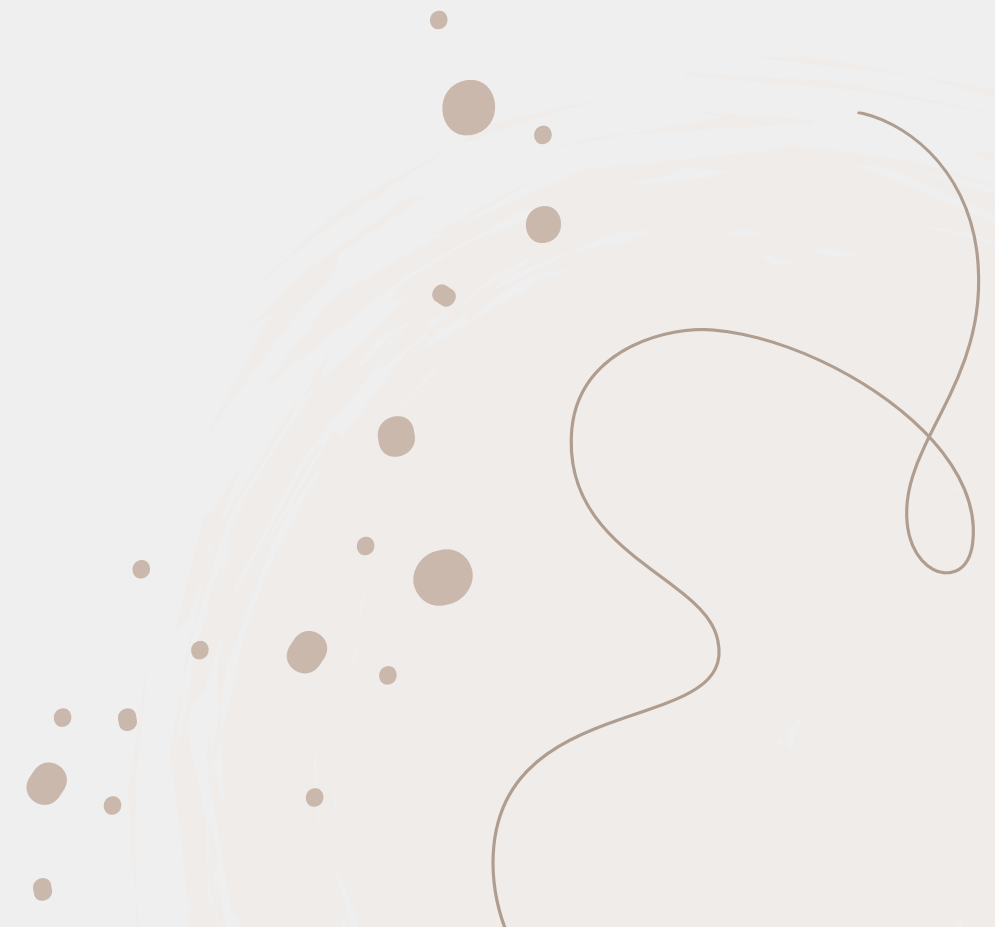
KATIKALA DEDEEPYA-CB.SC.U4AIE23349

GESHNA B-CB.SC.U4AIE23360

Overview



- Introduction
- Objective
- Relevance to computer network
- Software Used
- Methodology
- Plugin's Used
- Output



INTRODUCTION



- It is designed to be a seamless and efficient communication platform optimized for desktop environments with support for emulators and API-driven interactions, providing an engaging, real-time chat experience.
 - Built with a core emphasis on connectivity, functionality, and intuitive user experience, Conversify aims to bridge the gap between efficient communication and user-friendly design.
- ★ The AI Assistant interacts with users in a natural, conversational manner, using Natural Language Processing (NLP) to understand user queries and provide relevant responses. This allows users to get instant help and information without needing to navigate through complex menus.

OBJECTIVE

- **Deliver Reliable Real-Time Communication:** Facilitate seamless, instant messaging capabilities with real-time updates and notifications, ensuring a responsive and reliable communication experience.
- **Enable Secure Data Transmission:** Implement end-to-end encryption and secure protocols to protect user data, maintaining privacy and security throughout message exchanges.
- **Optimize Desktop Compatibility and API Integration:** Create a platform fully compatible with desktop environments using Pixel Ford API, supporting emulator-based testing and broad device adaptability.

- **Provide Scalable and Modular Architecture:** Design a flexible, modular structure to support future growth, allowing easy integration of additional features and scaling with user demand.
- **Focus on User-Friendly Interface Design:** Develop an intuitive and engaging user interface that prioritizes usability, making the chat experience efficient, enjoyable, and accessible.
- **Ensure High Performance Across Environments:** Utilize emulators to test and optimize the application for high performance, stability, and efficiency on various desktop systems.

HOW IS IT RELATED TO COMPUTER NETWORKS????



- **Network Communication Protocols:** This project leverages key computer networking protocols (such as HTTP, WebSocket, or TCP) to facilitate real-time data transmission, ensuring messages are sent and received instantly over a network.
- **Data Packet Transmission and Routing:** Messages and data are segmented into packets, routed through the network, and reassembled at the recipient's end, demonstrating core principles of packet-switched networks in computer networking.
- **Client-Server Architecture:** The application is based on a client-server model, where messages are sent from client devices to a centralized server and then directed to the intended recipient, illustrating one of the most common structures in computer networks.

- **Network Security and Encryption:** To protect user data, the project incorporates encryption techniques and secure protocols (e.g., TLS or SSL), applying network security practices essential to safeguarding data in transit over a network.
- **API Integration and Network Services:** Integrating Pixel Ford APIs and using emulators requires network connectivity and API-based communication, highlighting the role of network services and protocols in interfacing with external services.
- **Network Reliability and Latency Management:** By testing and optimizing the app in various network environments, the project addresses network reliability and latency issues, which are critical for maintaining seamless communication in any network-dependent application.

SOFTWARE USED

Android Studio

- Purpose: Primary IDE for Android app development, offering extensive debugging, emulation, and building tools.
- Features: Provides emulators for testing, integrated Firebase tools, and efficient debugging capabilities.

Firebase Authentication

- Purpose: Provides secure user authentication, crucial for maintaining privacy and security.
- Features: Supports multiple authentication methods (email, Google login, etc.) and generates unique tokens for session management.

Firebase Firestore

- Purpose: A NoSQL database solution designed for real-time data synchronization.
- Features: Enables real-time data sync, offline capabilities, and structured data storage, making it ideal for chat data management.

Firestore Cloud Messaging (FCM)

- Purpose: Handles push notifications to alert users of new messages or events in real-time.
- Features: Reliable delivery of messages across devices and integration with Firestore for seamless notifications.

Kotlin (or Java)

- Purpose: Core programming language for Android, used to write the app logic, UI, and interactions.
- Features: Modern language features like coroutines (in Kotlin) simplify asynchronous tasks like network requests and database operations.

Emulators and Physical Devices

- Purpose: Testing tools to verify app functionality, design, and responsiveness on various screen sizes and Android versions.
- Features: Emulators mimic different device configurations, ensuring the app functions well across diverse environments.

HOW THIS METHOD WORKS?



User Authentication

- **Sign-up/Sign-in:** When users sign up or log in, Firebase Authentication verifies credentials securely.
- **Token Generation:** Firebase generates a unique authentication token for each logged-in user, granting secure access to data and features.
- **Session Management:** The app maintains a session for authenticated users, enabling seamless access across sessions until they manually log out.

Real-Time Messaging

- **Message Sync:** Firebase Firestore's real-time capabilities allow instant storage and retrieval of messages across users.
- **Data Flow:** When a user sends a message, it is stored in Firestore and instantly synced across all active users in the chat.
- **Data Model:** The app stores messages with timestamps and sender details, allowing for chronological ordering and efficient retrieval.

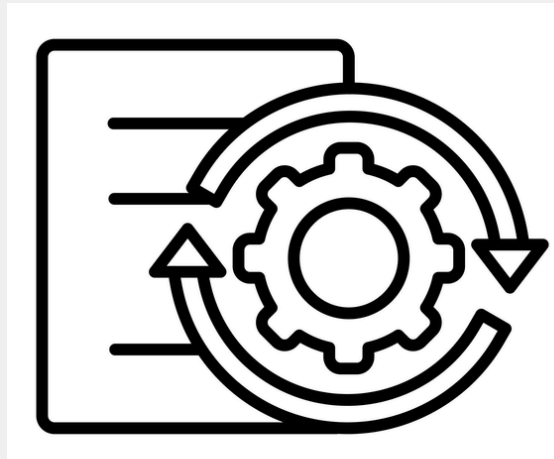
Push Notifications

- **FCM Setup:** Firebase Cloud Messaging (FCM) is configured to deliver notifications when new messages arrive.
- **Notification Triggers:** When a user sends a message, the server triggers FCM to notify the intended recipient(s).
- **Display and Response:** Notifications pop up even if the app is minimized, allowing users to respond directly or open the app to view the chat.

Database Management

- **Structured Data:** Firebase Firestore stores data in collections (e.g., "Users" and "Messages") to logically separate user data and conversations.
- **Data Access Rules:** Custom security rules ensure that only authenticated users can access or modify their messages and profiles.
- **Automatic Data Sync:** Firestore automatically synchronizes data across devices, ensuring consistency even with network fluctuations.

Deployment and Optimization



- **APK Creation:** The app is packaged as an APK using Android Studio, allowing for easy installation on Android devices.
- **Performance Optimizations:** Techniques like data caching, image compression, and minimal API calls reduce load times and conserve battery.
- **Testing and Debugging:** The app undergoes rigorous testing on various emulators and devices, ensuring compatibility and performance stability across platforms.

PLUGIN'S USED

Flutter Plugin

- Plugin ID: dev.flutter.flutter-plugin-loader
- Purpose: Facilitates Flutter development in Android Studio, streamlining build processes and adding Flutter support tools.
- Version: 1.0.0

Google Services (Firebase)

- Plugin ID: com.google.gms.google-services
- Purpose: Enables integration with Firebase services, such as Firestore, Authentication, and Cloud Messaging.
- Version: 4.3.15

File Selector Plugin

- Plugin ID: file_selector_linux/file_selector_plugin
- Purpose: Provides cross-platform file selection functionality within the app.
- Use: Essential if users need to upload files or share media within the chat app.

Kotlin Android Plugin

- Plugin ID: `org.jetbrains.kotlin.android`
- Purpose: Allows the use of Kotlin as the main programming language in the Android project.
- Version: 1.8.22

Android Application Plugin

- Plugin ID: `com.android.application`
- Purpose: Configures the project for Android app development, setting up build configurations, dependencies, and compiling settings.
- Version: 8.3.2

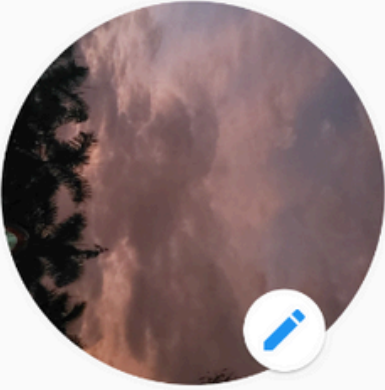
Gradle Plugins


- Plugin Management: Uses repositories like `google()`, `mavenCentral()`, and `gradlePluginPortal()` to handle dependencies for Android builds.
- Purpose: Essential for managing dependencies, plugins, and ensuring the project builds correctly.




←


Profile Screen







hansikareddy40@gmail.com










Name


Hansika Reddy

About

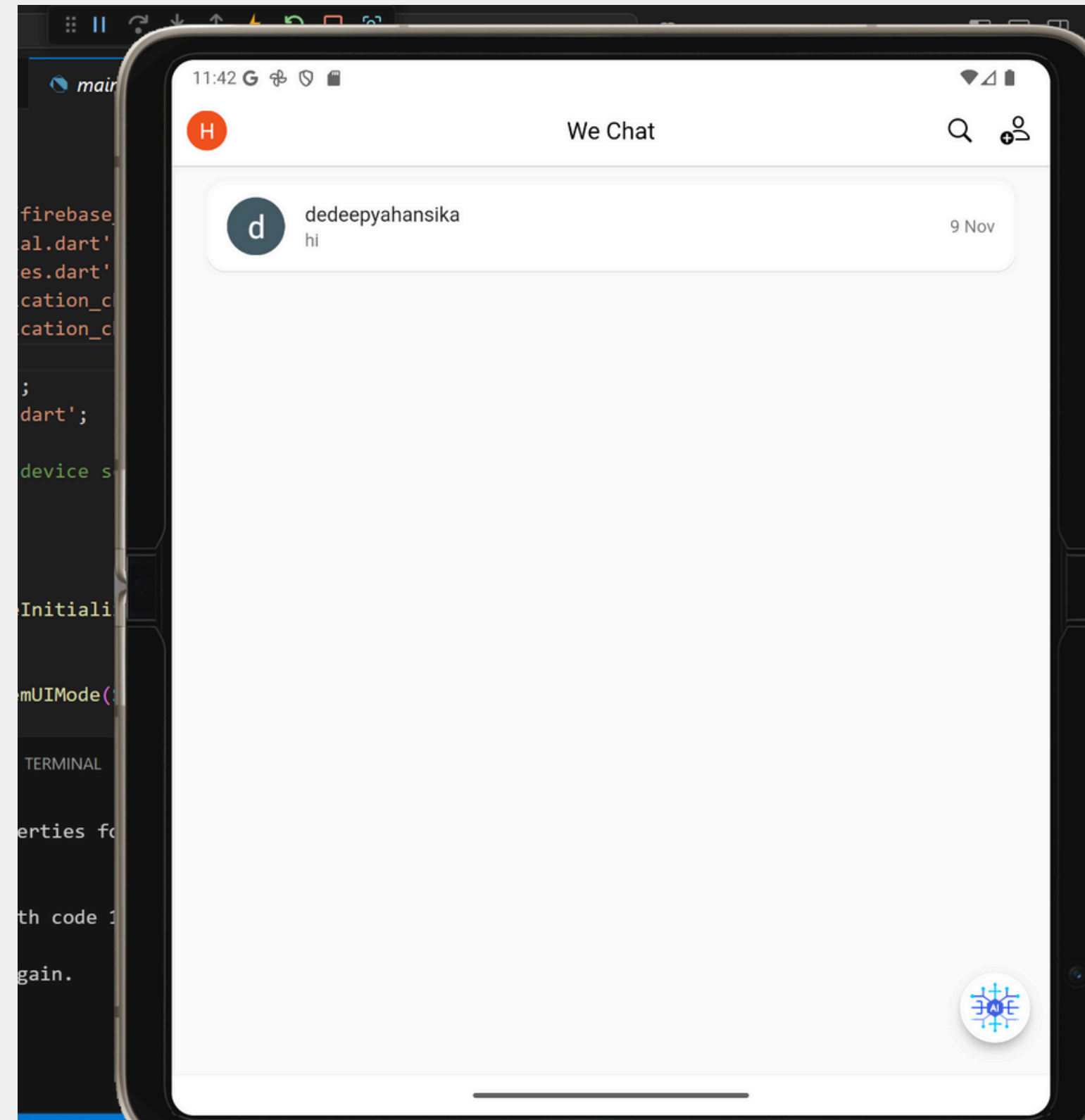
Hey, I'm using We Chat!

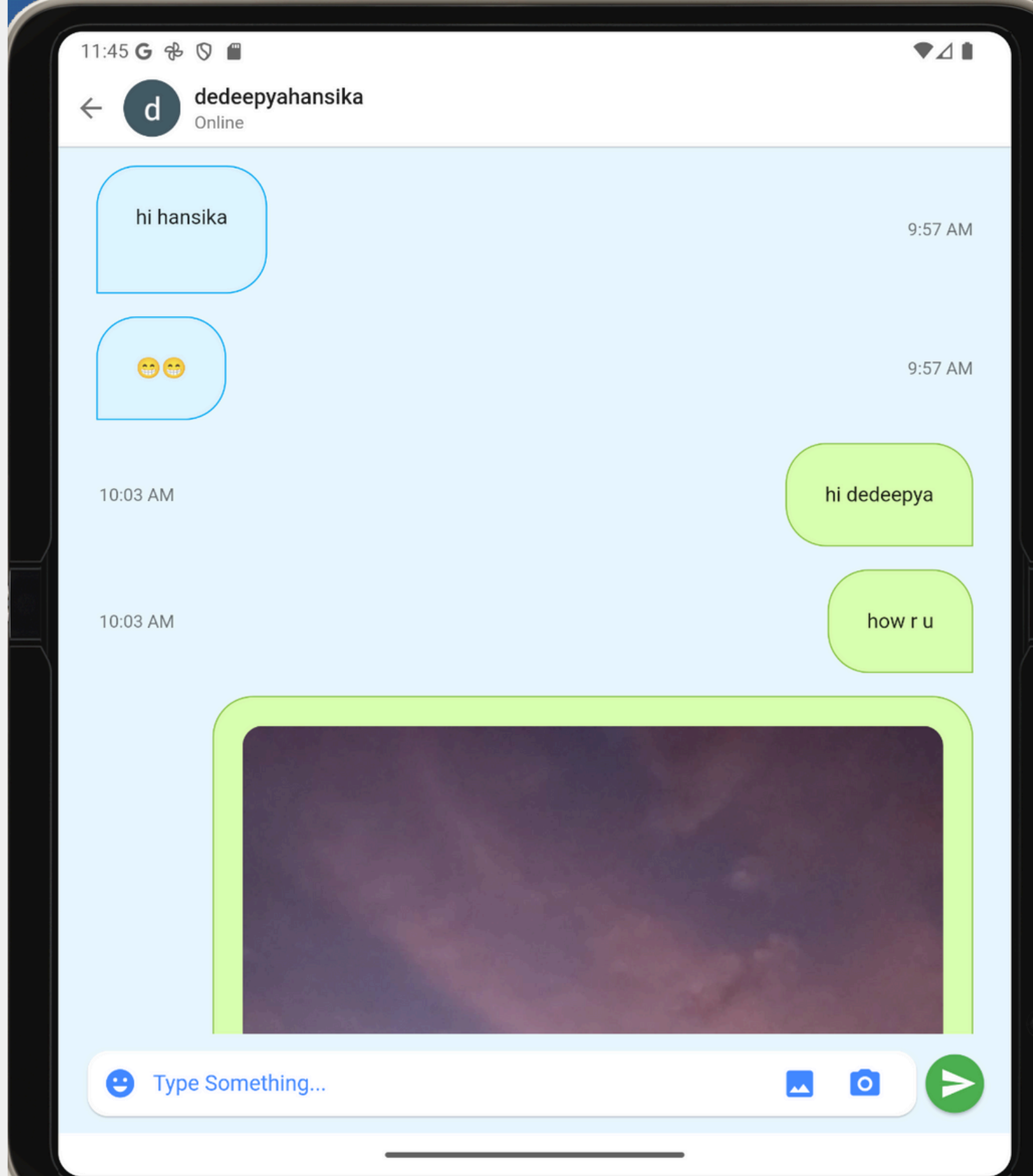
 UPDATE

Profile Updated Successfully!

 Logout

OUTPUTS





CHAT ASSIST

