

# **FLOOD RELIEF -FLOOD MANAGEMENT APP USING FLUTTER AND FIREBASE**

PROJECT REPORT

Submitted by

**NAME: SARA E JOHN**

**Reg No : TKM23MCA-2054**

To

TKM College of Engineering

*Affiliated to*

The APJ Abdul Kalam Technological University

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

MASTER OF COMPUTER APPLICATION



**Department of Computer Applications**  
**Thangal Kunju Musaliar College of Engineering**  
**Kerala**

(Government Aided and Autonomous)

NOVEMBER 2024

**DEPARTMENT OF COMPUTER APPLICATIONS**

**TKM COLLEGE OF ENGINEERING**

**(Government Aided and Autonomous)**

**KOLLAM – 691005**



**CERTIFICATE**

This is to certify that, the report entitled **FLOOD RELIEF- FLOOD MANAGEMENT APP** submitted by **SARA E JOHN (TKM23MCA-2054)** to the **APJ Abdul Kalam Technological University** in partial fulfillment of the requirements for the award of the Degree of **Master of Computer Application** is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor

Mini project Co-ordinator

## DECLARATION

I undersigned hereby declare that the project report **FLOOD RELIEF-FLOOD MANAGEMENT APP** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Prof. Sheera Shamsu**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

11/11/2024

---

Sara E John

## ACKNOWLEDGEMENT

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project. I am extremely grateful to **Prof. Natheera Beevi M**, Head of the Department, Dept of Computer Application, for providing us with the best facilities. I would like to place on record my sincere gratitude to my project guide **Prof. Sheera Shamsu**, Professor, Department of Computer Application for the guidance and mentorship throughout the course. Their contributions have played a crucial role in enhancing the overall learning experience. I also extend my heartfelt thanks to **Prof. Sheera Shamsu**, the mini project coordinator for their dedicated efforts and effective coordination. I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study. I owe thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

## **ABSTRACT**

This project proposes a comprehensive Flood Management System aimed at minimizing the impact of floods in Kerala's districts, with a particular focus on improving the frequency and intensity of flood responses through modern technology and efficient rescue operations. The primary objective of this system is to provide rapid rescue services and ensure the safety and security of flood victims. The system leverages technology to track and pinpoint areas most affected by floods in various Panchayats and Taluks within the district, allowing rescue teams to accurately locate and assist individuals in distress. A key feature of this system is its ability to manage critical lifeline support systems such as communication, power, and water supply during a flood. By integrating these services, the app ensures continuous support during and after a flood event. Additionally, it addresses issues related to the management of displaced individuals, facilitating the tracking of victims even if they are separated during the chaos, and enabling the identification of individuals in different rehabilitation centers. This integrated approach provides an effective solution to the challenges posed by floods, ensuring timely rescue, rehabilitation, and recovery for all affected communities.

# Table of Contents

Introduction .....	1
1.1 Existing System .....	2
1.2 Problem Statement .....	2
1.3 Proposed System .....	3
1.4 Objectives .....	4
Literature Survey .....	5
2.1 Purpose of Literature Survey.....	5
2.2 Related Works .....	5
2.2.1 Real Time Data Monitoring and Sharing .....	6
2.2.2 Resource Allocation and Logistic Management .....	6
2.2.3 Community and Rescue Team Coordination .....	6
Methodology .....	7
3.1 Architecture .....	7
3.1.1 System Design .....	8
3.2 Software Requirements and Specifications .....	9
3.2.1 Flutter.....	10
3.2.2 Firebase .....	10
3.2.3 Google Chrome.....	11
3.2.4 VS Code .....	11
3.2.5 Android Studio .....	12
3.2.6 Dart .....	13
3.3 Dependencies .....	13
3.3.1 Firebase Core .....	14
3.3.2 Firebase Auth .....	14
3.3.3 Firebase Storage .....	14
3.3.4 Cloud Firestore .....	14
3.3.5 Intl .....	15
3.4 Hardware and experimental environment .....	15
Result and Discussions .....	16
4.1 Registration and Login .....	17
4.1.1 Registration Page .....	18
4.1.2 Login Page .....	19

4.2 Admin Pages .....	20
4.2.1 Admin Home Page .....	20
4.2.2 Manage Camp .....	22
4.2.3 Manage Members .....	23
4.2.4 Manage QRT .....	24
4.2.5 View Details.....	25
4.2.6 Notifications .....	26
4.3 User Pages.....	27
4.4 Flood Management Team Page.....	29
4.5 Quick Response Team Page.....	31
4.6 Camp Page.....	33
4.7 Firebase Connection.....	35
4.8 Cloud Firestore.....	36
 Conclusion .....	 38
5.1 Future Enhancements .....	39
Reference .....	41

## List of Figures

3.1 Architecture	
3.1.1 System Design.....	9
4.1 Registration and Login	
4.1.1 Registration Page.....	18
4.1.2 Login page.....	19
4.2 Admin pages	
4.2.1 Admin Home Page.....	21
4.2.2 Manage Camp.....	22
4.2.3 Manage Members.....	23
4.2.4 Manage QRT.....	24
4.2.5 View Details.....	25
4.2.6 Notifications.....	26
4.3 User Pages	
4.3.1 User Home page.....	28
4.4 Flood Management Team Page	
4.4.1 FMT Page.....	30
4.5 Quick Response Team Page	
4.5.1 QRT Page.....	32
4.6 Camp Page	
4.6.1 Camp.....	34
4.7 Firebase Connection	
4.7.1 Firebase Console.....	35
4.8 Cloud Firestore	
4.8.1 Firestore Database.....	37



# Chapter 1

## Introduction

Flooding is a recurrent and severe challenge for districts across Kerala, with impacts that can disrupt lives, damage infrastructure, and overwhelm traditional emergency response systems. To address this, we propose a comprehensive flood management system, developed as a crossplatform Flutter application, which aims to reduce the impact of floods in terms of both frequency and intensity. This app focuses on providing rapid rescue services, efficient resource distribution, and streamlined rehabilitation processes, making use of cutting-edge technologies to support residents and responders during flood events.

This system operates by identifying and mapping flood-prone areas, specifically targeting affected panchayats and taluks within each district. Leveraging location data and real-time updates, the app enables swift localization of flood-stricken areas and delivers essential information to rescue teams. Residents in flood-impacted zones can use the app to register for rescue assistance, allowing them to be prioritized and promptly reached by rescue personnel. Upon being rescued, users receive automatic notifications assuring them of their safety, adding a layer of communication and comfort for both the rescued individuals and their loved ones.

To ensure the success of emergency responses, the app integrates and mobilizes critical resources, including communication channels, power supplies, and water access, all vital for managing flood scenarios. By deploying these resources effectively, the system addresses immediate needs and stabilizes life-support systems during crises. Additionally, the app's rehabilitation support feature ensures displaced individuals have access to secure locations post-rescue, streamlining their transfer to designated relief centers.

The app facilitates efficient coordination by providing rescue teams with precise locations, contact information, and requirements of those stranded, allowing them to prioritize responses. Furthermore, the app assists families by enabling them to locate relatives in rehabilitation centers, particularly in cases where family members have been separated during evacuation. This feature offers peace of mind and support in reuniting families during challenging times.

Cross-platform capabilities built in Flutter, this app can be readily deployed on both Android and iOS devices, reaching a wide user base across Kerala. The design is intuitive, ensuring

that even users with minimal technical skills can easily request help and stay informed. By fostering collaboration between residents and rescue teams, the app provides a unified and effective solution to flood management, ultimately reducing response times, improving safety outcomes, and enhancing disaster preparedness in Kerala's flood-prone regions.

## **1.1 Existing System**

The current flood management system in Kerala relies heavily on manual processes, lacking a centralized digital platform to efficiently coordinate rescue and relief efforts. Without a website or digital system, there is no way to input, store, or share real-time information on the location of victims, flood severity, or resource availability. This results in delayed decision-making, as rescuers and officials lack immediate access to critical data.

The absence of a structured command hierarchy further exacerbates the problem, with rescue teams operating without a clear chain of command, often leading to miscommunication and uncoordinated responses. Consequently, rescue teams may arrive at locations where victims have already been saved or, conversely, fail to reach areas where people are still trapped, wasting valuable time and resources. Rehabilitation centers, where rescued victims are relocated, frequently suffer from shortages in food, water, and medical supplies, as there is no organized system to monitor and manage resource distribution. This inadequacy creates additional hardship for displaced individuals, compromising their safety and well-being. Additionally, families separated during floods have no reliable way to reunite, as communication across rehabilitation centers is minimal.

The lack of equipment coordination further hinders rescue operations, delaying critical resources needed for effective flood response. Overall, these limitations lead to delayed, inefficient responses that put lives at risk and prolong the distress of flood victims. An integrated digital system is essential to address these challenges, enabling faster decisionmaking, better resource allocation, and more organized rescue and relief efforts across Kerala.

## **1.2 Problem Statement**

Floods in Kerala frequently disrupt lives, damage infrastructure, and pose severe challenges for disaster management. Currently, flood management processes are largely manual, lacking a centralized digital system to streamline rescue operations and efficiently allocate resources.

The absence of an organized command structure and a real-time tracking system delays response times, as rescue teams often lack critical information about victim locations and resource needs. This disorganization leads to inefficient rescue operations, with teams sometimes being deployed redundantly or unable to reach victims in isolated areas promptly. Additionally, rehabilitation centers suffer from inconsistent supply management, resulting in shortages of essential items like food, water, and medical supplies. Families separated during floods face further distress due to limited communication across centers, with no reliable way to locate or reunite with loved ones. The current system's inefficiencies and lack of digital infrastructure increase risks to human life, compromise the well-being of flood victims, and hinder timely and effective disaster response. Therefore, a comprehensive digital solution is essential to enhance flood management capabilities, improve resource coordination, and enable faster, more organized rescue and relief operations in Kerala.

### **1.3 Proposed System**

The proposed flood management system is a comprehensive digital platform designed to streamline rescue operations, resource allocation, and rehabilitation efforts during flood events. This system offers an online registration feature, allowing victims and common residents to easily request help and receive updates on rescue efforts.

Additionally, it enables real-time tracking and information sharing about people in rehabilitation centers, which helps both families and rescue teams stay informed. By establishing a clear line of command, the system improves coordination among rescue teams and other agencies, reducing response times and increasing operational efficiency.

The system is also designed to support precautionary measures, providing alerts and guidance to help people avoid high-risk zones before and during flood events. With a built-in capability to track the number of people across different camps, the system ensures that resources are distributed fairly and in alignment with the needs of each location.

The platform also facilitates sponsorships, allowing individuals and organizations to donate essential items directly to specific camps in need. Overall, this proposed system aims to improve the speed, accuracy, and effectiveness of flood response in Kerala, enhancing safety and resource management during critical times.

## 1.4 Objectives

- 2 **Provide an Online Registration System** : Enable victims and residents to register for rescue assistance and receive real-time updates on their safety status.
- 3 **Enhance Victim Tracking and Rescue Efficiency:** Allow rescue teams to quickly locate stranded individuals, reducing response time and improving the efficiency of rescue operations.
- 4 **Establish a Clear Command Structure:** Implement a line of command to ensure organized coordination among rescue teams, government agencies, and other stakeholders.
- 5 **Facilitate Resource Allocation and Management:** Track and manage the distribution of essential resources (food, water, medical supplies) across various rehabilitation centre's to ensure all centre's receive adequate supplies.
- 6 **Support Family Reunification** : Provide a system to help families locate and reunite with separated members who may be placed in different rehabilitation centers.
- 7 **Enable Precautionary Alerts and Safety Guidelines** : Issue timely alerts and precautionary guidance to help residents avoid high-risk areas and stay informed about flood situations.
- 8 **Track Population in Rehabilitation Camps:** Monitor and maintain records of the number of people in each camp to ensure accurate supply and aid distribution.
- 9 **Allow Sponsorship and Donations** : Facilitate sponsorship opportunities, enabling individuals and organizations to contribute resources to specific camps and support flood victims directly.
- 10 **Reduce Redundancy in Rescue Operations** : Prevent duplication of rescue efforts by providing real-time status updates on victim locations and rescue statuses to all teams involved.
- 11 **Improve Overall Disaster Preparedness and Response** : Create an organized, digital framework to enhance disaster response capabilities, ultimately minimizing the impact of floods on affected communities.

## **Chapter 2**

### **Literature Survey**

A literature survey, also known as a literature review, involves analyzing scholarly sources related to a particular subject. Examining the available literature, it provides a comprehensive overview of the state of the field, allowing you to identify relevant theories, approaches, and gaps in the existing body of knowledge. When conducting a literature review from an audit perspective, the main focus is on evaluating the relevant literature. This process covers information that has been published in a specific field of study and sometimes includes information published within a specific time frame.

#### **2.1 Purpose of Literature Survey**

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarising them into one complete report.
2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarise, evaluate, and compare original research in that specific area.
3. It ensures that researchers do not duplicate work that has already been done.
4. It can provide clues as to where future research is heading or recommend areas on which to focus.
5. It highlights the key findings.

#### **2.2 Related Works**

The study of prior research related to this project involves Three critical aspects:

1. Real-Time Data Monitoring and Sharing
2. Resource Allocation and Logistics Management
3. Community and Rescue Team Coordination

### **2.2.1 Real-Time Data Monitoring and Sharing**

Studies emphasize the importance of real-time data collection and sharing in enhancing flood response efforts. Monitoring systems using IoT sensors, GIS mapping, and satellite imaging have proven effective in tracking weather patterns, water levels, and flood-prone areas, enabling timely responses. Research by Thomas et al. (2019) on real-time flood mapping highlights how GIS-integrated platforms allow local authorities to monitor areas at risk continuously, improving preparedness and allowing for early warning systems to alert residents. Additionally, real-time data shared across agencies provides a common operational picture, ensuring that all parties—from government agencies to on-ground rescue teams—have access to the same critical information, leading to a more coordinated response.

### **2.2.2 Resource Allocation and Logistics Management**

Effective flood management systems must also focus on optimal resource allocation and logistics, especially during emergencies. Past studies on resource management have revealed that delays in distributing essential resources such as food, water, and medical supplies significantly impact victim survival and recovery. Research by Singh and Patel (2020) examined resource distribution models during disaster management and found that centralized resource tracking reduces redundancies and ensures resources reach the areas of greatest need. These studies underscore the importance of having a digital platform that not only tracks available resources but also optimizes distribution based on real-time demand, which is critical for ensuring rehabilitation centers remain adequately supplied.

### **2.2.3 Community and Rescue Team Coordination**

Lastly, research highlights the importance of efficient communication and coordination between the community, rescue teams, and local authorities during flood events. Studies like those by Zhang et al. (2021) discuss how a defined command structure and clear communication channels can prevent miscommunication and duplicative rescue efforts. The integration of mobile applications for community engagement has shown to be particularly effective in providing a means for residents to report emergencies, request help, and receive updates. Moreover, systems that allow for family reunification tracking and provide platforms for individual and organizational sponsorships have been found to improve the psychological and physical well-being of victims. Research indicates that platforms that integrate communication among these stakeholders significantly improve coordination, reduce response times, and promote a collaborative approach to disaster management.

# Chapter 3

## Methodology

The methodology for developing a flood management system follows a structured and systematic approach to ensure that the system addresses the core challenges faced during flood disasters and improves the efficiency of response operations. The first step, **requirement analysis**, involves gathering insights from key stakeholders such as local authorities, rescue teams, and the community to identify the specific needs and challenges in flood management.

Once the requirements are established, the next phase is **system design**, where a robust architecture is created to ensure scalability, reliability, and security. This includes designing a centralized database to store vital information such as victim details, resources available at rehabilitation centers along with designing user interfaces for both the public and rescue teams. The system's design also incorporates a clear line of command to facilitate coordination among stakeholders, ensuring that rescue teams and authorities work in sync.

In the **implementation phase**, the system is developed with a focus on both frontend and backend components. Frontend development involves creating the user interface, typically using a framework like Flutter to ensure cross-platform compatibility (Android and iOS).

Once developed, the system undergoes **testing** to ensure its functionality and reliability. This includes unit testing for individual modules, integration testing to ensure proper interaction between modules, and user acceptance testing (UAT) to verify that the system meets the practical needs of the users in real flood scenarios.

Finally, after the system passes all tests, it moves to the **deployment and evaluation phase**. Initially, the app will be deployed in a limited region for pilot testing, where real-time monitoring will help refine the system based on user feedback and operational observations.

### 3.1 Architecture

The architecture of the flood management system is designed to be modular, scalable, and efficient, ensuring that all stakeholders, including residents, rescue teams, and authorities, are able to interact effectively with the system. The architecture consists of three key layers: the

**Frontend (Client Side)**, the **Backend (Server Side)**, and the **Database Layer**. Each layer plays a crucial role in the overall performance and functionality of the system.

### 3.1.1 System Design

#### **Frontend (Client Side):**

The frontend layer of the system is responsible for the user interface (UI) and user experience (UX). It allows residents, rescue teams, and authorities to interact with the system via mobile devices (Android/iOS) and web browsers. The frontend will be built using **Flutter**, a crossplatform framework that allows the app to be deployed across multiple platforms seamlessly.

#### **User Registration and Authentication:**

- Users (victims, rescue teams, authorities) can sign up and log in to the app via Firebase Authentication (email/password, phone number, or Google sign-in).
- Victims can submit requests for rescue, track the status of their rescue operations, and receive notifications.

#### **Rescue Request and Status Tracking:**

- Users can request rescue services by providing details such as location (using GPS) and emergency type (flood, medical, etc.).
- Real-time status updates are provided on the progress of the rescue operation.

#### **Real-Time Location Tracking:**

- The app can access the GPS location of users, allowing rescue teams to track victims in real-time. This data is sent and updated in **Cloud Firestore**.

rescue requests, sending notifications, or handling complex business rules that need to run on the server side. For example, Cloud Functions may be triggered when a rescue request is made to notify rescue teams in real-time.



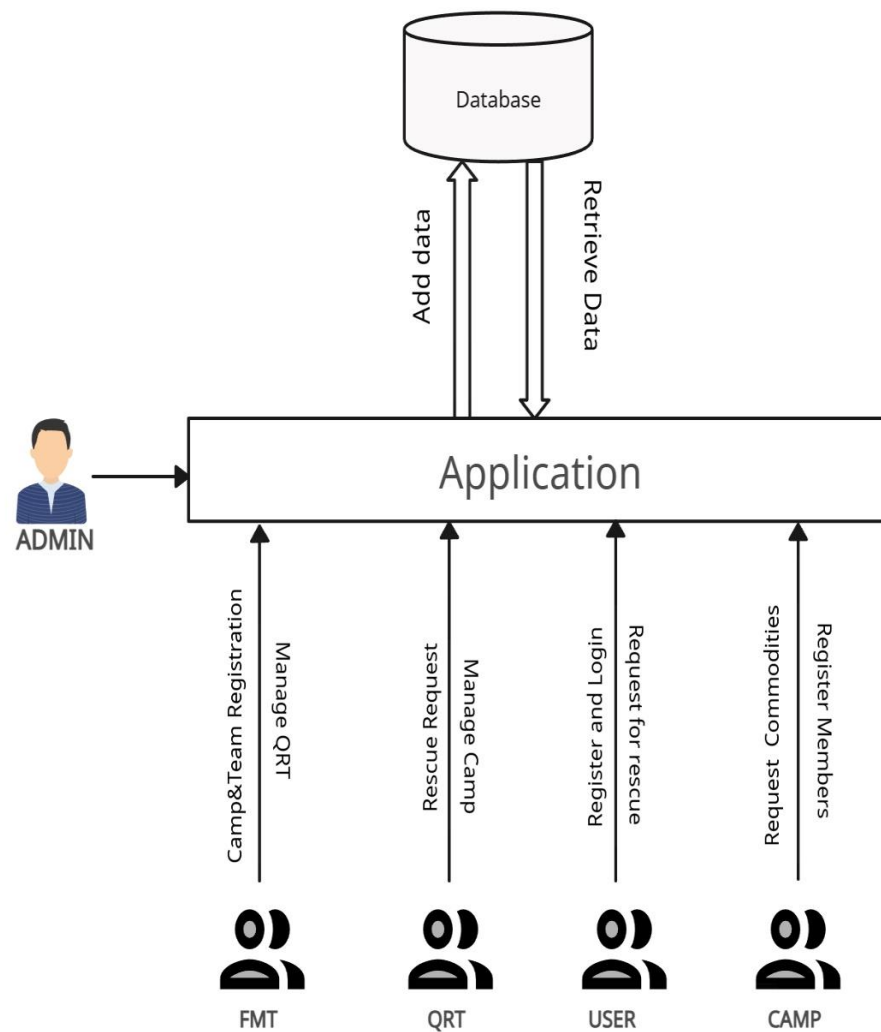


Figure 3.1 system design

### 3.2 Software Requirements and Specifications

The software requirements for this project includes:

1. Flutter
2. Firebase
3. Google Chrome
4. VS Code
5. Android Studio
6. Dart

### **3.2.1 Flutter**

Flutter is a versatile framework created by Google for building cross-platform applications from a single codebase, covering mobile (Android and iOS), web, and desktop. Written in the Dart programming language, Flutter offers native-like performance by compiling directly to machine code. Its “hot reload” feature stands out, allowing developers to instantly view changes in the app without restarting it, thus enhancing productivity. Flutter's rich library of customizable widgets simplifies the development of aesthetically consistent UIs across platforms, with dedicated widgets for both Android (Material Design) and iOS (Cupertino).

Another key feature is Flutter’s layered architecture, which offers full control over UI rendering and customization. This flexibility makes it easy to create complex animations and tailor visual elements precisely. Moreover, Flutter supports plugins and integrations with services like Firebase, Google Maps, and device APIs, making it suitable for feature-rich applications. The framework is supported by a large community, extensive documentation, and continuous updates from Google, helping developers stay up-to-date and troubleshoot effectively. With its balance of performance, flexibility, and ease of use, Flutter is an increasingly popular choice for businesses and developers aiming for consistency and efficiency in cross-platform app development.

### **3.2.2 Firebase**

Firebase is a robust backend-as-a-service platform provided by Google that supports the development of web and mobile applications. It offers a suite of tools and services that streamline backend functionality, allowing developers to focus more on building engaging user experiences. A key feature of Firebase is its Authentication service, which simplifies user login with support for email and social logins like Google and Facebook. Firebase also provides powerful, cloud-hosted databases: Firestore for structured data and Realtime Database for applications that need data updates in real-time across multiple users.

Beyond data storage, Firebase includes Cloud Functions, which enable server-side code execution in response to app events, reducing the need for traditional backend infrastructure. Cloud Storage in Firebase is ideal for handling and serving large files such as images or videos, and Firebase Hosting offers fast, secure deployment for web applications and static content. Additionally, Firebase Analytics allows developers to monitor app performance, user

engagement, and events, while Crashlytics offers detailed reports to address app crashes and improve stability. Finally, Firebase's Machine Learning Kit makes it easy to incorporate AI features like image recognition or text analysis with minimal setup.

Together, these features make Firebase highly scalable and flexible, catering to both small projects and large-scale enterprise applications. Its seamless integration with Google's ecosystem also allows developers to leverage other services like Google Cloud, making Firebase an all-in-one platform that is well-suited to modern app development needs.

### **3.2.3 Google Chrome**

Google Chrome is a fast, secure, and widely-used web browser developed by Google, designed to provide an efficient and smooth browsing experience. Known for its speed and minimalist design, Chrome was built with the WebKit rendering engine and later switched to Google's Blink engine, which enhanced its performance and loading speeds. Chrome supports crossplatform compatibility, available on Windows, macOS, Linux, Android, and iOS.

Its core features include automatic updates, incognito mode for private browsing, tabbed browsing with individual process isolation for each tab, and a robust security system that includes phishing and malware protection. Chrome also seamlessly integrates with Google services, such as Google Drive, Gmail, and Google Docs, offering a unified experience for users in the Google ecosystem.

The browser supports a wide range of extensions and add-ons through the Chrome Web Store, allowing users to customize functionality, from productivity tools to ad blockers. Chrome's sync feature enables users to save bookmarks, passwords, and history across multiple devices when logged into their Google account.

### **3.2.4 VS Code**

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft that offers a range of features for developers. Known for its speed and versatility, VS Code supports a wide variety of programming languages through built-in features and an extensive marketplace of extensions. The editor includes essential tools like syntax highlighting, code navigation, intelligent code completion, and error detection, making it ideal for beginners and advanced developers alike.

One of VS Code's standout features is IntelliSense, an advanced code completion tool that provides context-aware suggestions, enhancing productivity and reducing errors. Additionally, the integrated terminal allows developers to run scripts and commands directly within the editor, streamlining workflows. VS Code also supports version control with Git integration, letting users manage repositories, track changes, and commit code without leaving the editor.

Its extension ecosystem is another major advantage, with plugins for everything from language support (e.g., Python, Java, Go) to frameworks (e.g., React, Angular) and tools (e.g., Docker, Kubernetes). Extensions also enable custom themes, keybindings, and language-specific linting, allowing developers to tailor VS Code to their specific needs. This flexibility, along with its cross-platform compatibility (available on Windows, macOS, and Linux), has made VS Code one of the most popular code editors in the developer community.

### **3.2.5 Android Studio**

Android Studio is an integrated development environment (IDE) developed by Google for building Android applications. Based on IntelliJ IDEA, it provides a suite of tools tailored for Android development, including a powerful code editor, a real-time preview of user interface changes, and extensive debugging options. Android Studio supports features like Gradle-based build system for managing dependencies, emulators for testing on various devices, and code templates for quick setup of standard Android components.

One of its standout features is Android Jetpack integration, which includes libraries, tools, and guidance to help developers reduce boilerplate code and focus on creating robust applications. Android Studio also provides profiling tools for performance optimization, allowing developers to monitor CPU usage, memory, and network activity. Furthermore, it supports Kotlin and Java for Android app development and integrates seamlessly with Firebase for backend services like authentication and real-time database.

With continuous updates from Google, Android Studio remains the official and most comprehensive IDE for Android development, providing the resources to build, test, and optimize applications on a variety of Android devices.

### **3.2.6 Dart**

Dart is an open-source, general-purpose programming language developed by Google. It's designed to be optimized for building fast, scalable, and high-performance applications, particularly for front-end development on both web and mobile platforms. Dart powers Flutter, Google's popular UI toolkit, making it especially favoured for cross-platform app development. With a syntax similar to languages like JavaScript and Java, Dart is easy for developers to learn and adopt.

Key features include AOT (Ahead-of-Time) and JIT (Just-in-Time) compilation, which boosts application startup speed and enables hot-reloading, allowing developers to see code changes in real-time during development. Additionally, Dart's asynchronous programming support simplifies handling of operations such as network requests and file I/O, which are common in mobile and web applications.

Dart has a rich standard library that supports tasks like data manipulation, networking, and file handling, and the language emphasizes strong typing and sound null safety to reduce runtime errors. Its combination of ease of use, efficient performance, and cross-platform capabilities makes Dart a powerful choice for modern app development.

## **3.3 Dependencies**

The Dependencies for this project includes:

1. Firebase Core
2. Firebase Auth
3. Firebase Storage
4. Cloud Firestore
5. Intl

### **3.3.1 Firebase Core**

Firebase Core is a foundational module within Firebase that connects your application to Firebase's backend services. It provides essential functions like app initialization, configuration, and connection to Firebase services. By integrating Firebase Core, developers gain access to tools like analytics, cloud messaging, authentication, and storage, enabling them to build feature-rich, scalable applications quickly. Firebase Core is compatible with various platforms, including Android, iOS, and web applications, and is a critical entry point for leveraging the robust ecosystem of Firebase services.

### **3.3.2 Firebase Auth**

Firebase Auth is a powerful authentication tool within Firebase that simplifies user sign-in and authentication processes in mobile and web applications. It supports various authentication methods, including email and password, phone numbers, and third-party providers like Google, Facebook, and Apple. Firebase Auth handles complex backend processes, such as token management and user session persistence, making it easier for developers to secure user data while providing a seamless login experience. Additionally, Firebase Auth integrates with Firebase's other services, allowing for a cohesive development experience with robust security and usability.

### **3.3.3 Firebase Storage**

Firebase Storage is a cloud storage solution that allows developers to store and serve user-generated content such as images, videos, and other files. It offers robust, secure, and scalable storage backed by Google Cloud Storage. With Firebase Storage, developers can upload and download files directly from their apps, using a simple API that integrates seamlessly with Firebase Authentication for secure access control. It also supports features like resumable uploads, metadata management, and access rules to ensure files are stored and shared according to the app's requirements.

### **3.3.4 Cloud Firestore**

Cloud Firestore is a flexible, scalable NoSQL cloud database designed for mobile, web, and server development. It allows developers to store, sync, and query data for their applications in real time. With features like automatic scaling, offline support, and real-time updates, Cloud Firestore simplifies data management for apps. Its powerful querying capabilities and structured data organization help developers create responsive user experiences while ensuring

data consistency across different platforms. Integration with other Firebase services further enhances its functionality, making it a robust choice for modern applications.

### **3.3.5 Intl**

Intl is a package in Dart that provides internationalization and localization support for Flutter applications. It allows developers to format dates, numbers, and currencies based on the user's locale. By using the Intl library, developers can easily create applications that cater to diverse language and regional preferences, making it easier to reach a global audience. The package supports pluralization, gender-based translations, and message formatting, enabling a more tailored user experience. Integrating Intl into a Flutter app enhances its accessibility and usability for users worldwide.

## **3.4 Hardware and experimental environment**

This project was built and tested on a consumer laptop with an Amd Ryzen 5 3500H processor with 4 cores and 8 threads, and 16 GB RAM, running on Windows 11 64-bit operating system. The laptop also acted as the web server for serving the project files and the platform for executing the application.

The experimental environment was prepared using Flutter and Firebase on Visual Studio Code. The application was tested on multiple web browsers, namely Google Chrome, Brave Browser, and Mozilla Firefox.

# Chapter 4

## Result and Discussions

The **Flood Management App** has significantly improved the overall flood response process in Kerala by addressing several key challenges in flood disaster management. One of the major improvements is the **reduction in response time** for rescue operations. The app enables **realtime tracking** of both victims and rescue teams, allowing for accurate location sharing and faster responses. Victims can now directly register for rescue services, providing detailed information and location data, which is then sent to rescue teams for immediate action. This has helped in minimizing delays and avoiding overlapping rescues, ensuring that no one is left behind.

Additionally, the app has optimized **resource management** in rehabilitation centers by providing real-time tracking of **resources** such as food, water, and medical supplies. Users can also **donate** or sponsor specific camps, ensuring that resources are directed where they are most needed.

The app's **data accuracy** and **real-time synchronization** have also been crucial. Built on **Cloud Firestore**, the system ensures that all users access the most current data, even in areas with limited connectivity, thanks to the app's offline capabilities. This feature has ensured that rescue teams and victims can still input and retrieve critical data when needed. Additionally, the app's use of **Flutter** has made it **scalable** and **cross-platform compatible**, ensuring that it is accessible to a wide user base across both iOS and Android devices. The app's simplicity and ease of use have contributed to its widespread adoption, with positive feedback from both victims and rescue teams.



## 4.1 Registration and Login

The **Registration Screen** of the Flood Management App is designed to allow users to create a new account with the app, providing necessary details such as their name, email, password, role, and for specific roles (like 'User'), additional information like their situation and location. The screen is structured to offer a smooth and intuitive registration process, guiding users through various form fields and validation checks.

Figure Form includes essential fields such as **Name**, **Email**, and **Password** inputs, along with a dropdown menu for **Role Selection**. The role selection dynamically updates the UI and ensures appropriate privileges based on the chosen role. For users registering as a **User**, additional fields for **Situation** and **Location** are provided. These fields are critical for users who need help and help direct rescue efforts. The form's design is intuitive, with clear labels and centered text inputs, ensuring a smooth registration process.

The **Login Screen** of the Flood Management App serves as the entry point for users to authenticate and access various roles within the app. This screen uses Firebase Authentication to handle user login and role-based navigation. The layout consists of several key components, making it both functional and user-friendly.

When the user enters their email, password, and selects a role, pressing the Login button triggers the `\_handleLogin` function. This function first authenticates the user using Firebase Authentication's `signInWithEmailAndPassword` method. After successful authentication, it retrieves the user's role from Firestore and checks if it matches the selected role.

If the roles match, the user is directed to the appropriate screen based on their role (Admin, FMT, QRT, User, or Camp). If authentication or role validation fails, the app displays error messages through Snackbar widgets, notifying the user of issues like "user not found," "wrong password," or "role mismatch."

### 4.1.1 Registration Page

100

Name

Email

Role

Password

Confirm Password

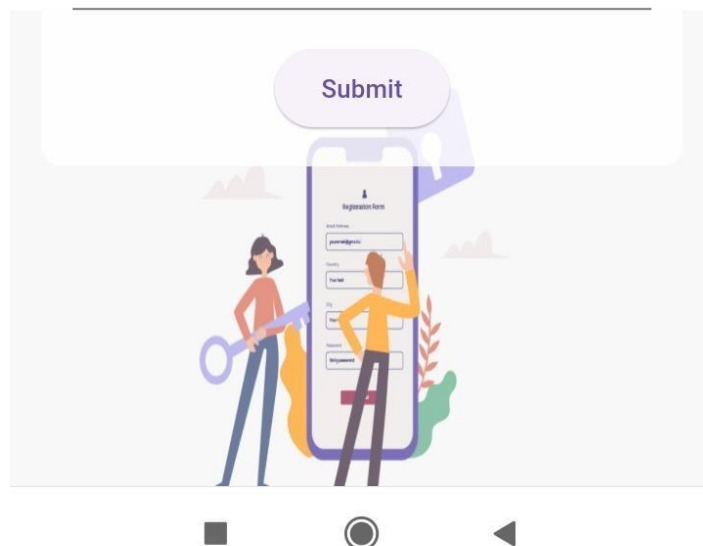


Figure 4.1.1 Registration Page

### 4.1.2 Login Page

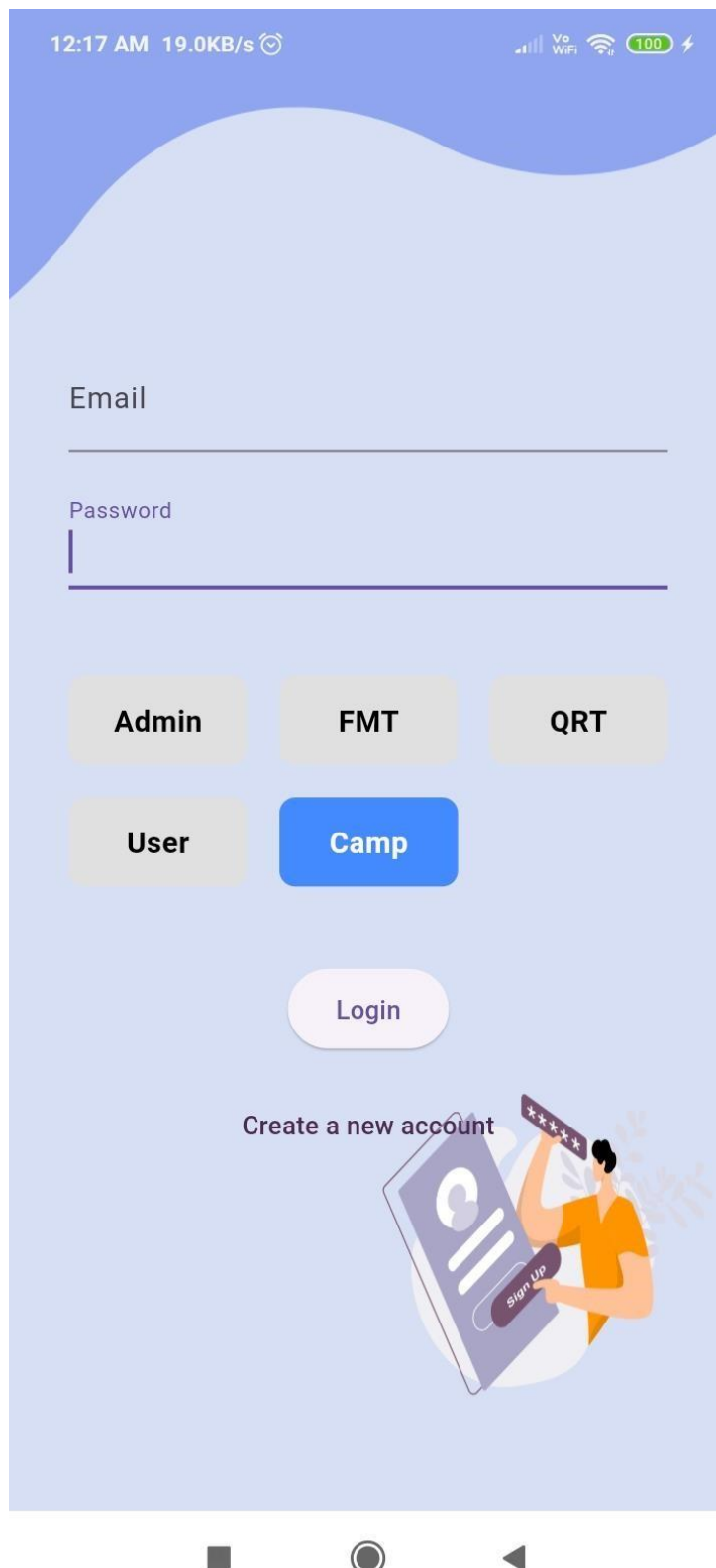


Figure 4.1.2 login Page

## 4.2 Admin Pages

The Admin module plays a central and crucial role in the flood management app, serving as the backbone for managing the entire system. The admin has full access to register and log in, with authentication typically handled via Firebase. Once logged in, the admin gains control over various critical aspects of the application.

Managing Camps involves overseeing the registration, update, and allocation of camps, ensuring that resources and shelter are appropriately distributed across the flood-affected areas. The Quick Response Team (QRT) Management feature allows the admin to assign tasks, track progress, and ensure that the rescue efforts are properly coordinated.

Additionally, the admin is responsible for Managing Members, which entails adding, removing, or updating the details of users and staff in the system, as well as overseeing their roles and responsibilities. The admin also has the capability to Display Notifications to inform teams and users about important updates, emergency alerts, or new directives.

Finally, the View Details functionality lets the admin access all the necessary data regarding the camps, QRT status, and user reports, helping in decision-making and efficient resource management. Overall, the admin module ensures smooth operation and oversight, allowing for quick, coordinated responses in times of crisis.

### 4.2.1 Admin Home Page

The **Admin Home** screen is a well-organized, user-friendly dashboard where the admin can navigate between different administrative tasks. The background image creates an aesthetically pleasing environment, and the transparent app bar ensures the content remains the focal point. The clickable list items make it easy for the admin to manage various aspects of the flood management system. This approach ensures the admin can efficiently handle their responsibilities within the app, such as managing camps, teams, and users, while being able to view essential system details and notifications.

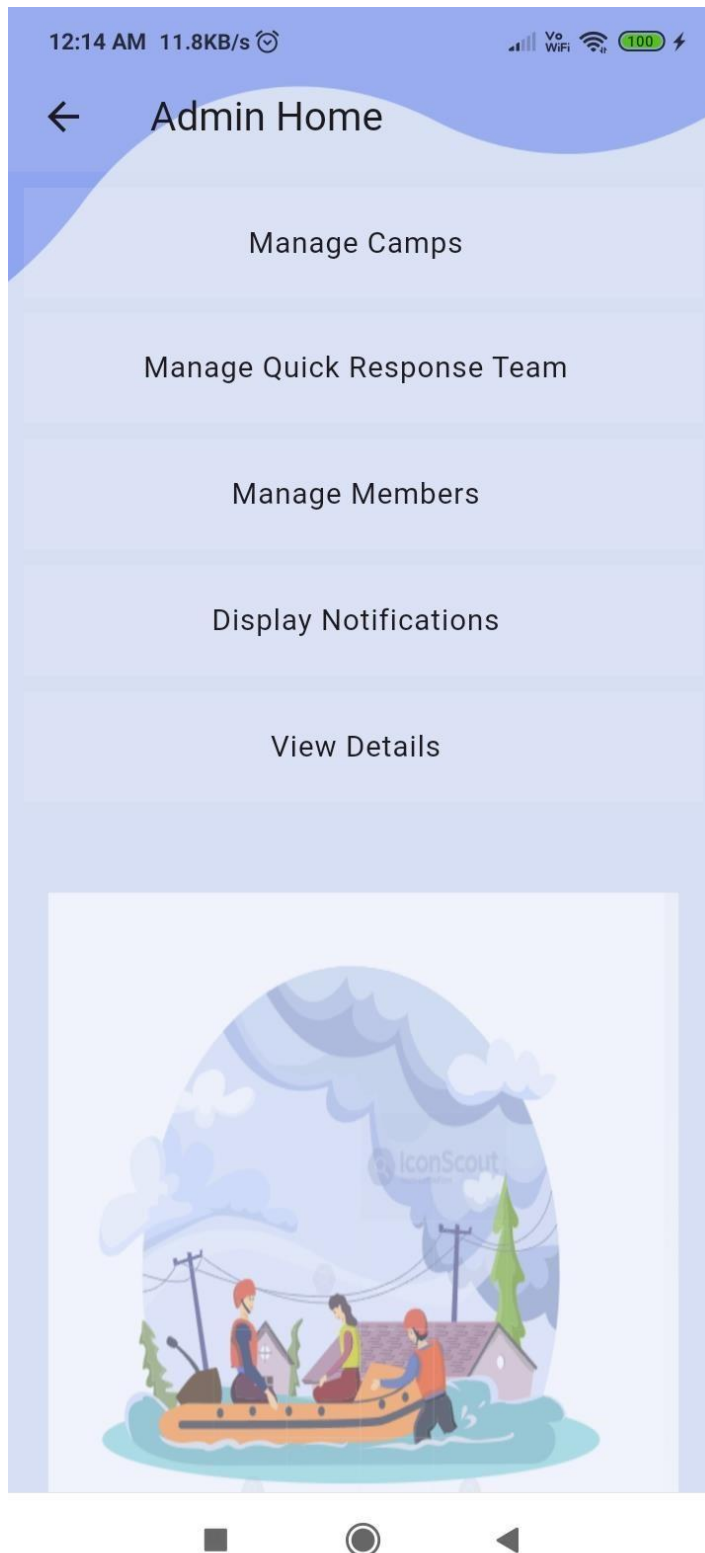


Figure 4.2.1 Admin Home page

#### 4.2.2 Manage camp

The "Manage Camp" feature in the admin module of the flood management app enables administrators to oversee camp setup, resources, and staff assignments efficiently. Admins can register camps with location and capacity details, track essential supplies, and make requests when resources are low. Staff members can be assigned specific roles to ensure smooth operations, and admins receive real-time updates on camp status, occupancy, and urgent needs. Additionally, the feature provides reports and analytics, helping admins make data-driven decisions for better resource allocation and preparedness.

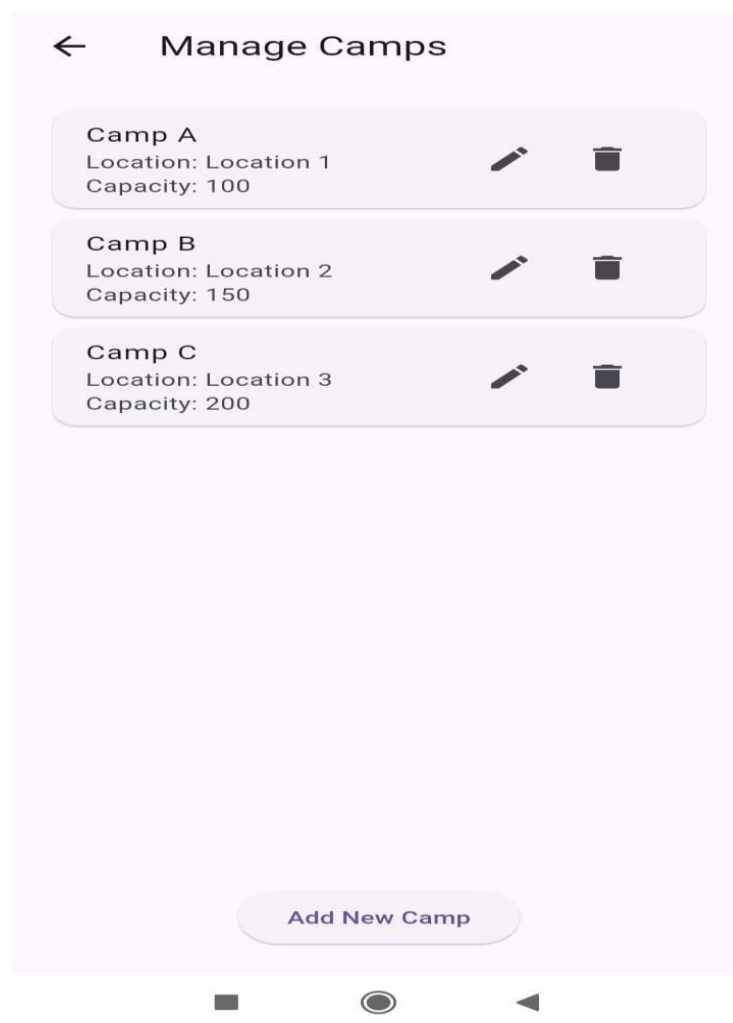


Figure 4.2.2 Manage camp

### 4.2.3 Manage Members

The "Manage Members" feature in the admin module allows administrators to oversee and organize all personnel involved in flood management efforts. Admins can add, update, and view member details, assign roles, and track their activity within the system. This ensures that the right personnel, like quick response teams or camp staff, are assigned effectively, enhancing coordination and response efficiency across all operations.

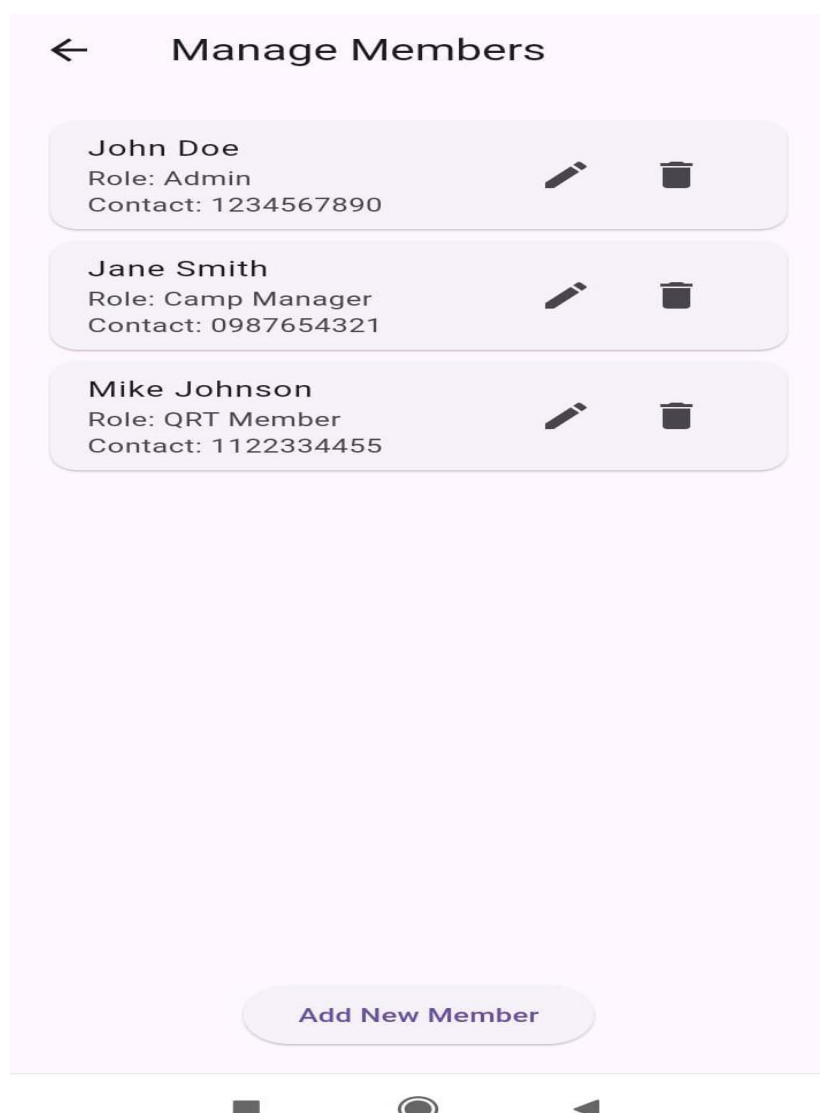


Figure 4.2.3 Manage Members

#### 4.2.4 Manage QRT

The "Manage QRT" feature in the admin module allows administrators to coordinate Quick Response Teams (QRTs) for effective flood management. Admins can register and assign QRT members, track team locations, and allocate them to high-need areas based on real-time data. This feature enables quick deployment, helps monitor team activities, and ensures each team has the necessary resources and instructions to assist affected communities. By centralizing QRT management, the feature supports fast, organized rescue operations and resource distribution during flood events.

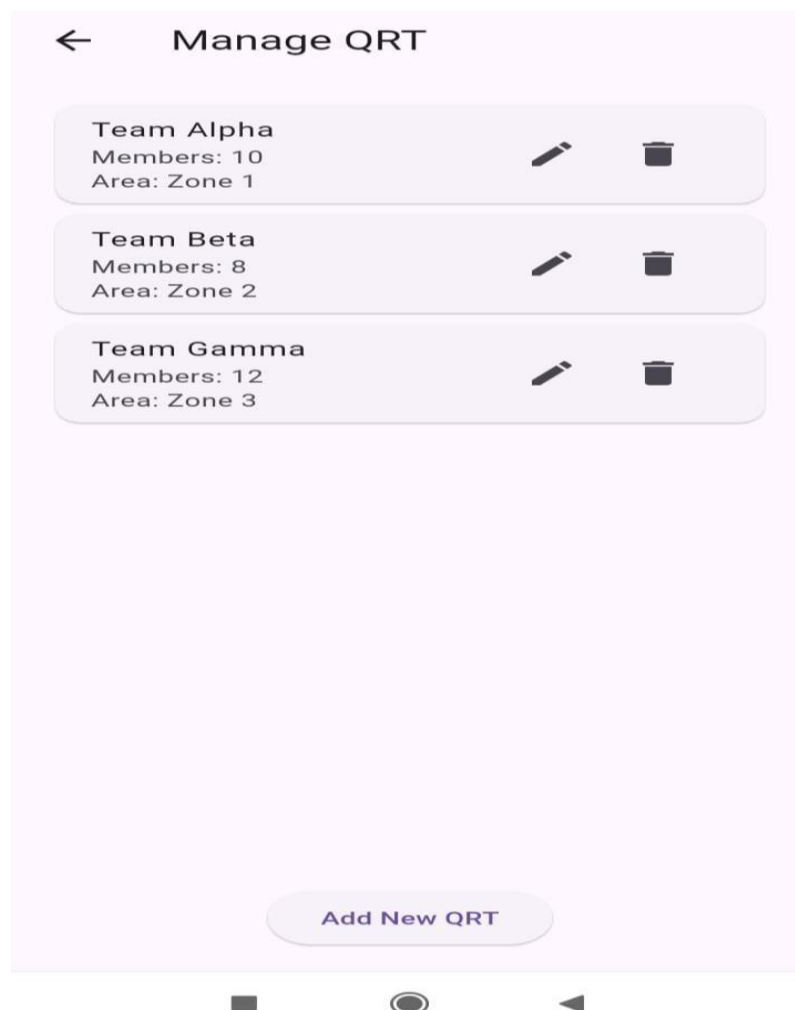


Figure 4.2.4 Manage QRT



#### 4.2.5 View Details

The "View Details" feature in the admin page allows administrators to quickly access comprehensive information about all active camps, rescue teams, and registered members. This includes real-time updates on each camp's status, available resources, and occupancy, as well as details about team assignments and user profiles. By centralizing this data, the feature enables admins to monitor operations closely, make informed decisions, and respond swiftly to any emerging needs or shortages across camps.

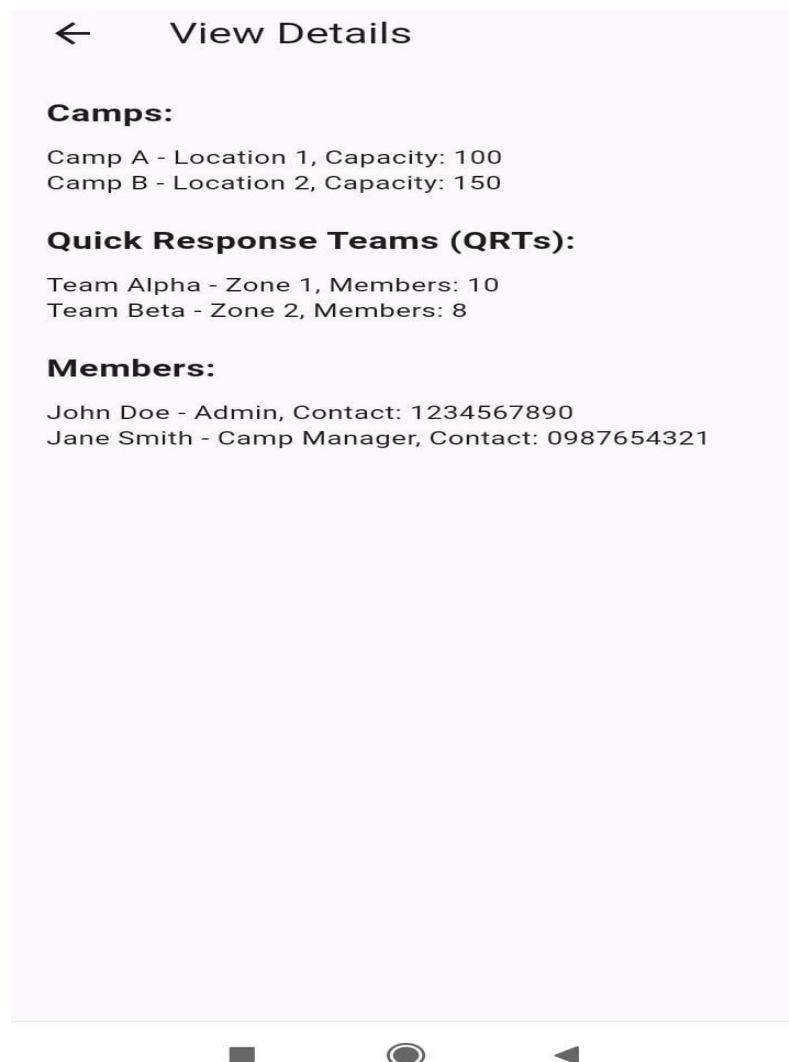


Figure 4.2.5 view details

#### 4.2.6 Notifications

The "Notifications" feature on the admin page enables administrators to send and view important alerts and updates for users within the system. Admins can quickly notify relevant teams, camps, or users about urgent information, such as flood warnings, rescue updates, or supply needs. This helps ensure timely communication across all roles, enhancing coordination and response efforts during flood events.

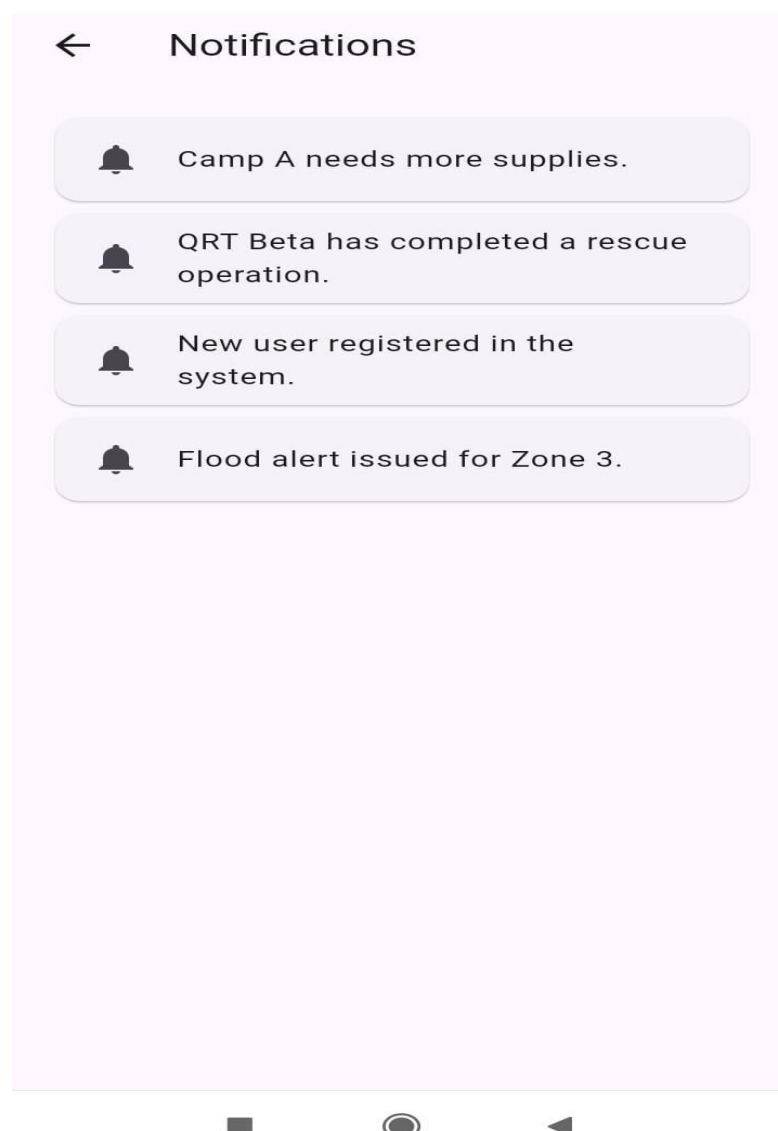


Figure 4.2.6 Notifications

### 4.3 User Page

The **User page** is designed to provide a seamless and efficient experience for individuals seeking assistance or support within the system. It caters to users who are typically not involved in the administrative or management tasks but require access to essential features like registering, requesting rescue, and updating their profile. Here's a detailed breakdown of the key functionalities within the **User page**:

The **User page** starts with the process of registering and logging into the system. Users are required to create an account by providing essential details such as their name, email, password, and selecting a role, which will categorize them within the system. Once registered, users can log in to access the system's features. The login process uses email and password authentication to ensure the security and privacy of user data. If they are already registered, they can simply log in using their credentials.

One of the primary functionalities of the **User page** is the ability to **request rescue**. In emergency situations such as floods, users can quickly submit a rescue request for themselves or others. The request form captures critical information such as the user's **location** and **situation**, allowing emergency teams or administrators to assess the severity of the request and respond accordingly. The user can also optionally add additional context to help responders locate and assist them faster.

The **User page** also allows users to update their profiles. This feature provides users the flexibility to maintain accurate and up-to-date information about themselves. Users can edit details such as their name, email, and other personal information. This can be particularly important if a user moves to a new location or changes their contact details. By keeping the profile current, users ensure that the system has the necessary information for future interactions, such as sending notifications, rescue requests, or any other communication related to the app's services.

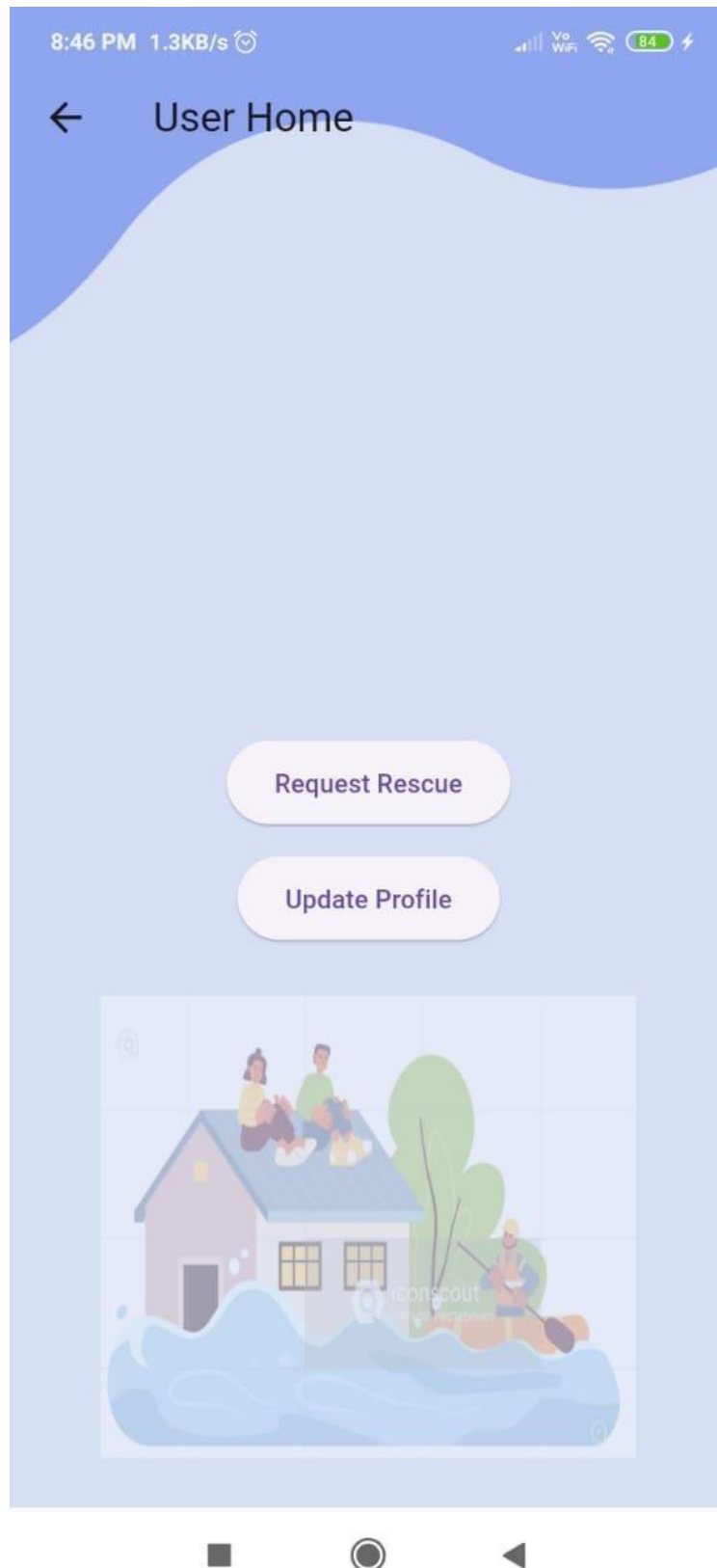


Figure 4.3.1 User Page

## 4.4 Flood Management Team Page

The **FMT (Flood Management Team)** module plays a crucial role in coordinating and responding to emergencies, such as floods, by managing camps, coordinating with Quick Response Teams (QRT), viewing details about ongoing events, and assigning rescue requests to the QRT. Here's a detailed breakdown of the functionalities within the **FMT module**:

The **Flood Management Team (FMT)** members must first create an account by registering with their personal details, including name, email, password, and role selection. Once registered, the members can log in using their credentials (email and password) to gain access to the features specific to the FMT role. The login process ensures that only authorized personnel can access the system's management features for flood control and response.

The **FMT module** allows users to **register camps** and **quick response teams (QRT)**. The FMT personnel can initiate the registration process by entering details about temporary shelters or **camps** being set up for displaced individuals during a flood or natural disaster. These details may include the camp's location, capacity, available resources, and current population.

The **View Details** functionality provides the FMT with an overview of important information related to flood management efforts. This feature could include real-time data about the current flood situation, including water levels, affected areas, ongoing rescue operations, and the number of people needing assistance. It may also allow the FMT to access reports about registered camps, QRT statuses, and other key metrics.

One of the most important functions of the **FMT module** is the ability to **assign rescue requests to Quick Response Teams (QRT)**. When a rescue request comes in from the **User** module (such as a request for evacuation or medical aid), the FMT is responsible for evaluating the request and dispatching the appropriate QRT to handle it.

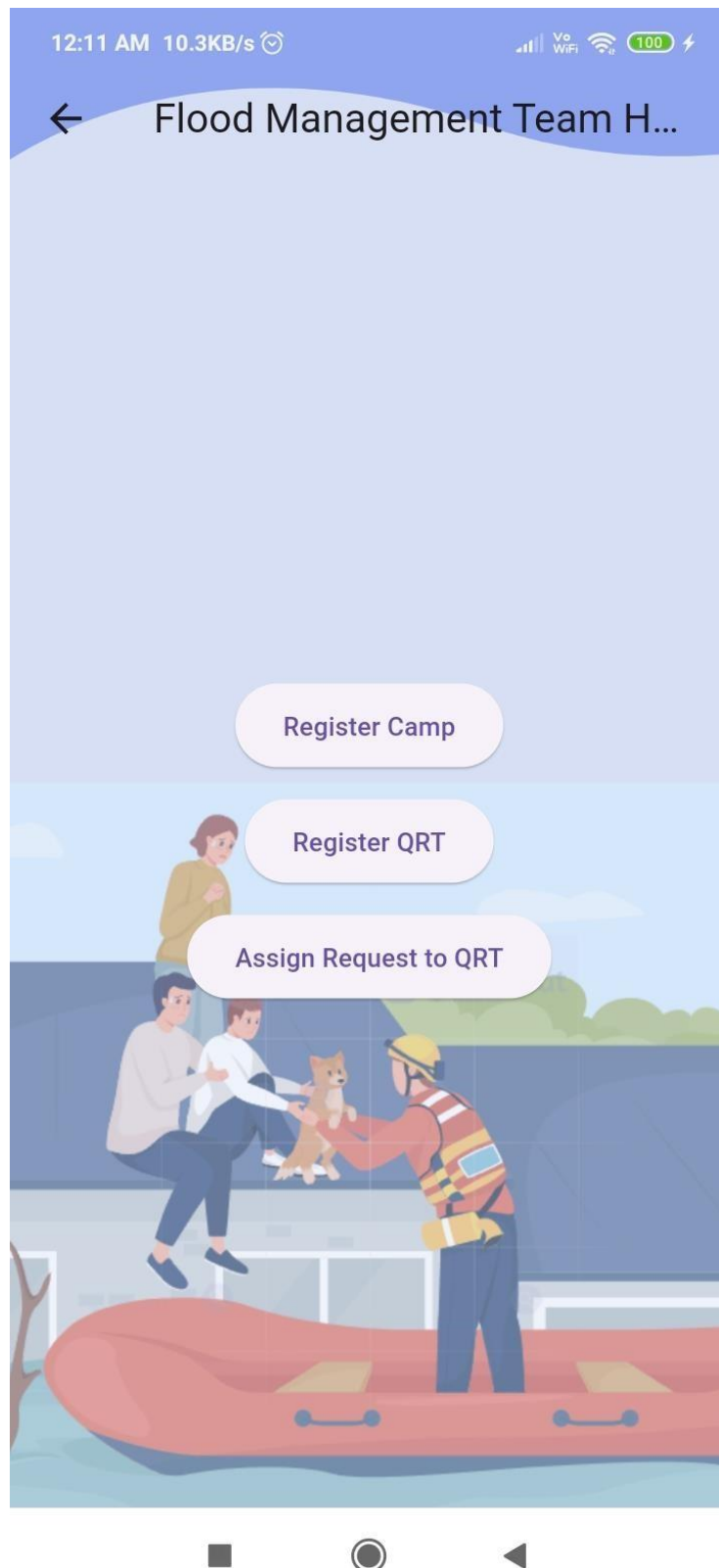


Figure 4.4.1 FMT Page

## 4.5 Quick Response Team Page

The **Quick Response Team (QRT)** module is designed to facilitate the team's critical role in providing immediate assistance during emergencies like floods, rescues, and other urgent disaster responses. The QRT's duties are pivotal for ensuring the safety and well-being of affected individuals, and the module enables them to perform essential tasks effectively. Below is a detailed explanation of the functions within the **QRT page**:

Similar to other modules, the **QRT page** begins with the **register and login** process. This ensures that only authorized members of the Quick Response Team can access the system. During the **registration** process, the team members provide their personal details, role-specific information, and credentials. After successfully registering, the QRT members can **log in** using their credentials (email and password) to access the system.

The **View Request from Users** functionality allows QRT members to access and review **rescue requests** submitted by users. When a **User** registers a request for help (such as a rescue operation, evacuation, or medical assistance), this request is sent to the **QRT page**. The QRT can then view and assess these requests to prioritize and assign resources for rescue efforts.

The **requests** may include essential details such as the user's location, type of emergency (e.g., flood, medical emergency, injury, or missing persons), and any other relevant information that will help the team prepare for the operation. This screen could feature filters or sorting options to help QRT members sort through different requests, based on urgency, proximity, or type of need.

After successfully completing a rescue mission, QRT members need to **register the rescued persons' details** into the system. This is an essential function for tracking the individuals rescued, ensuring that records are updated, and that necessary follow-up actions are taken (e.g., medical aid, transport to a camp, etc.).

The **View Camp Details** functionality allows the QRT to access information about **camps** set up for displaced people. Once a rescue operation is completed, rescued individuals may need to be sent to a camp for temporary shelter, food, or medical care.

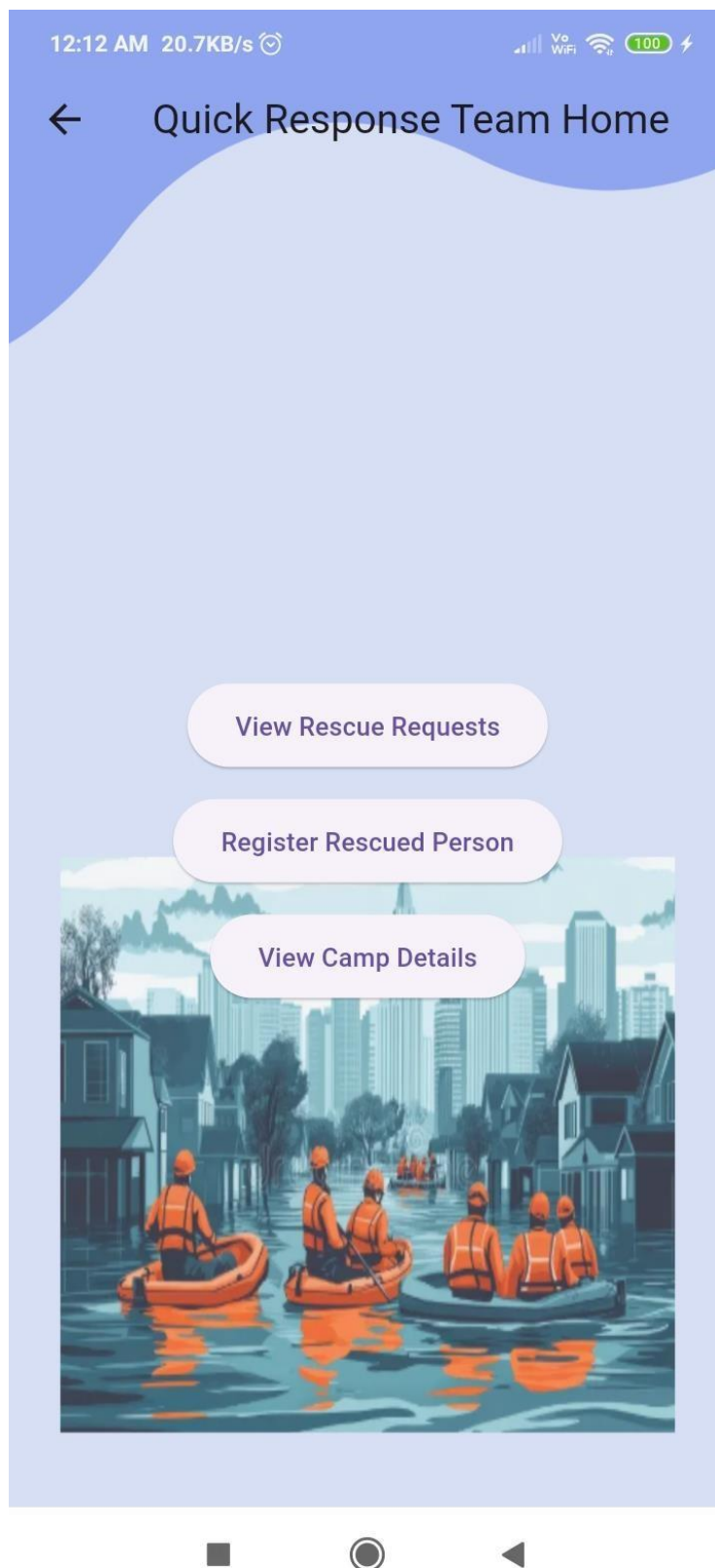


Figure4.5.1 QRT Page



## 4.6 Camp Page

The **Camp Module** is a crucial component of the system, particularly in disaster management scenarios such as floods, where displaced individuals need shelter, food, and medical care. The camp module serves multiple purposes, from registering the camp and its members to managing supplies and viewing essential details. Below is an expanded explanation of the **Camp Module** functionalities:

The Register and Login feature ensures that only authorized personnel, such as camp managers and staff, can access the system and manage camp-related activities. During registration, camp administrators can create an account by providing essential details like name, email, password, and role, ensuring that only individuals with valid credentials can interact with the camp management features. Once registered, camp administrators can log in using their credentials (email and password). This login authentication process ensures secure access, restricting unauthorized users from viewing or modifying camp data, thereby safeguarding sensitive information and maintaining control over the camp's operations.

The Requested for Commodities feature is designed to manage the supplies and resources needed by the camp, which is crucial in disaster situations where various resources like food, clothing, medical supplies, water, and tents are required. Camp administrators can use this function to request specific items from external suppliers or government agencies, with the system tracking these requests and their fulfilment status. Additionally, This ensures that camp administrators can efficiently allocate resources, avoid shortages of essential supplies, and meet the needs of displaced individuals within the camp.

The Member Registration feature allows camp administrators to register displaced individuals arriving at the camp, capturing essential information to facilitate effective camp management and service provision. The registration form includes fields for key details such as the individual's name, age, gender, health status (e.g., medical conditions or injuries), family details (for reunification or special accommodations), and the type of assistance required (e.g., medical care, food, water, or shelter). This data is stored in the camp's database, ensuring that it is readily accessible for reporting, decision-making, and coordination with other services, such as medical aid or food distribution

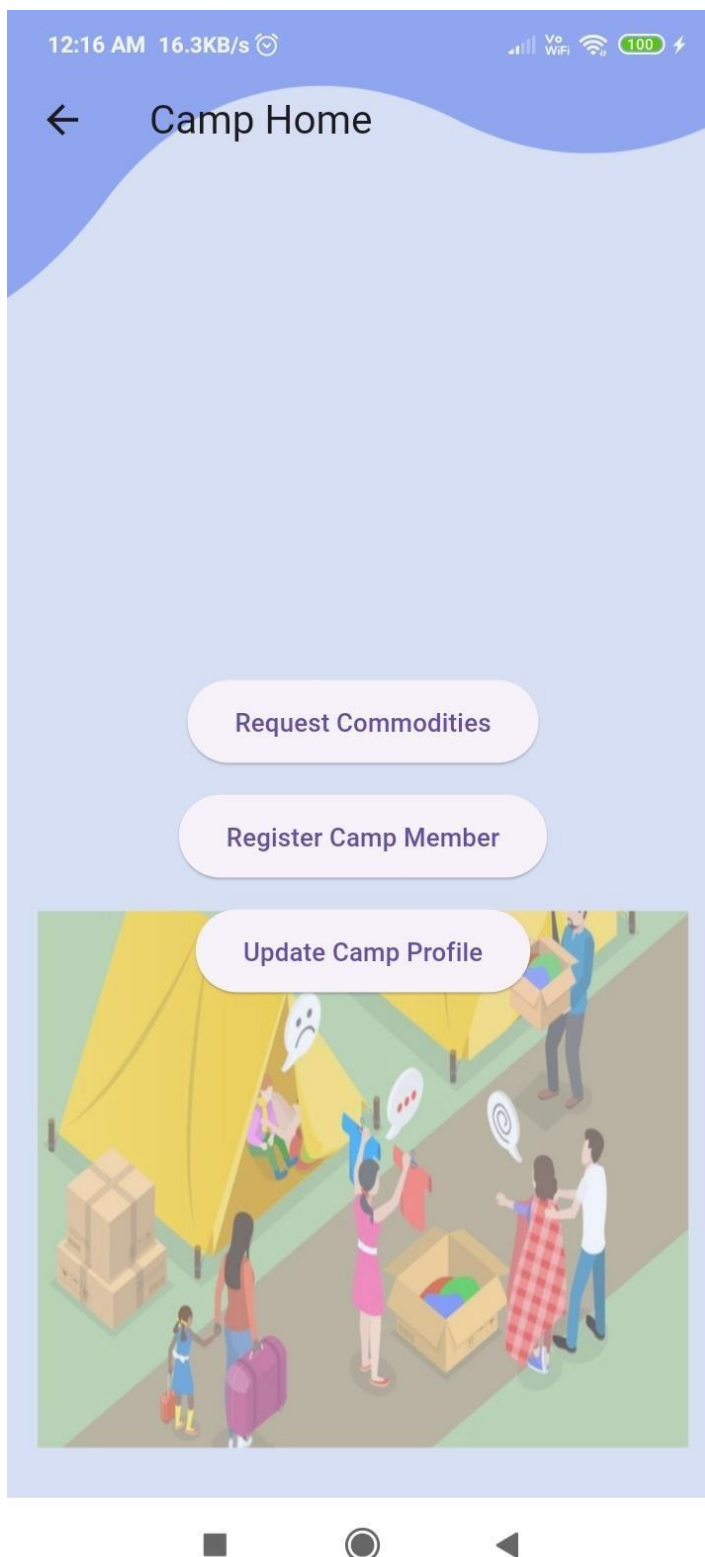


Figure 4.6.1 Camp

## 4.7 Firebase Connection& Authentication

The flood management app integrates Firebase to provide a robust, scalable backend solution that supports real-time data storage, authentication, and notifications, which are essential for an efficient and responsive disaster management system. Figure 4.7.1 Firebase Authentication is implemented to enable secure login and registration processes for different user roles, such as administrators, camp managers, rescue teams, and citizens. This ensures that only verified personnel and registered individuals have access to sensitive features like rescue requests, camp management, and commodity distribution.

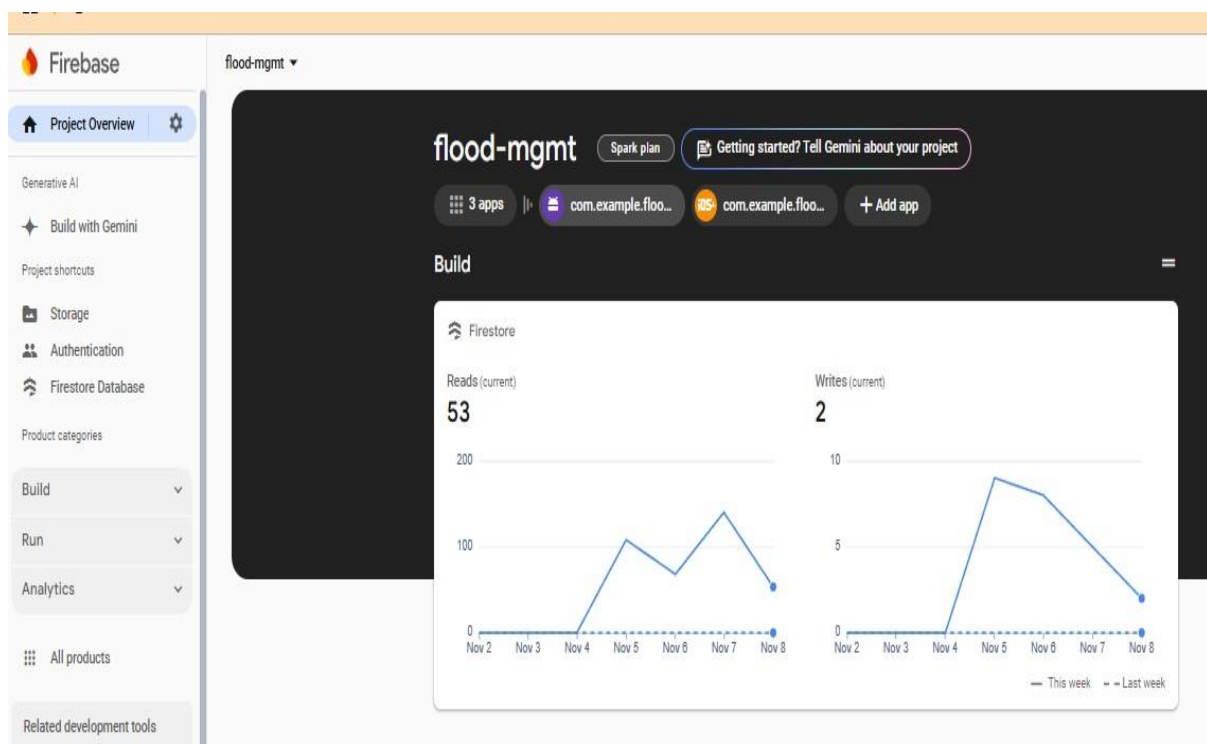


Figure 4.7.1 Firbase console

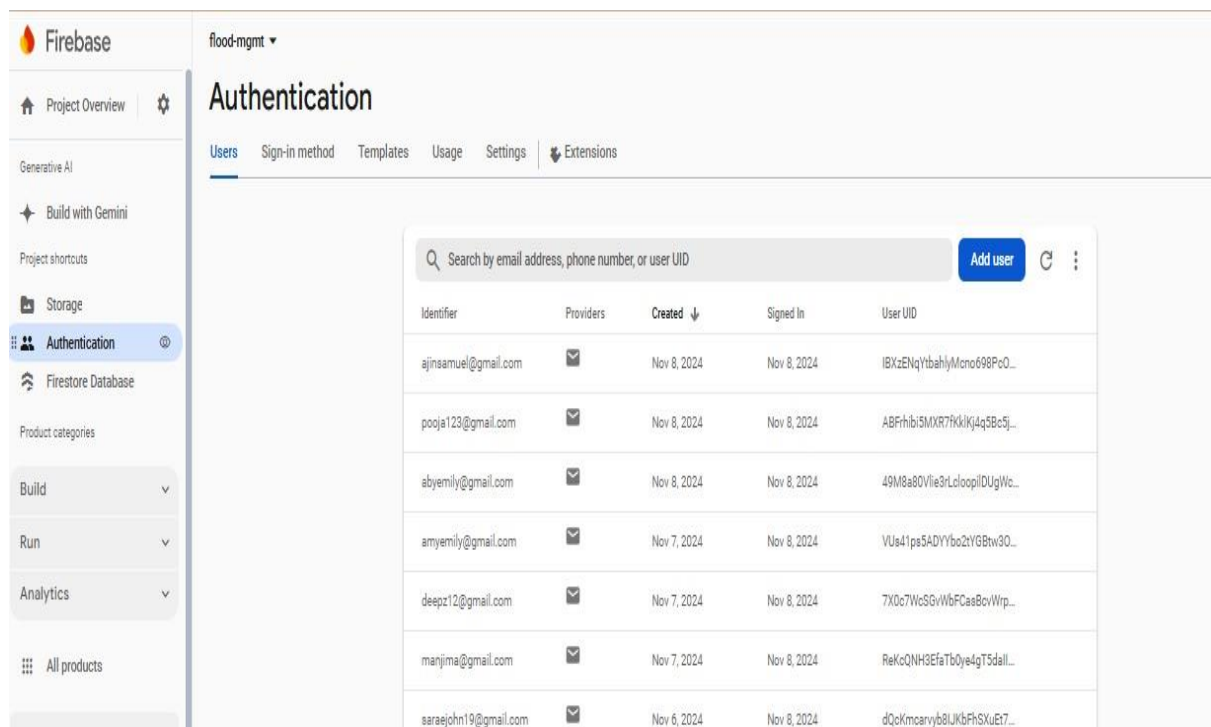


Figure 4.7.2 Authentication

4.8 Cloud Firestore

Firebase Firestore, a real-time NoSQL cloud database, is used to store and manage various types of data, such as user profiles, rescue requests, camp details, inventory status, and member information. This allows data to be accessible to users in real time, facilitating quick response and coordination among teams. For instance, when a citizen places a rescue request, it is instantly available to the Quick Response Team (QRT) members, who can act swiftly to provide assistance.

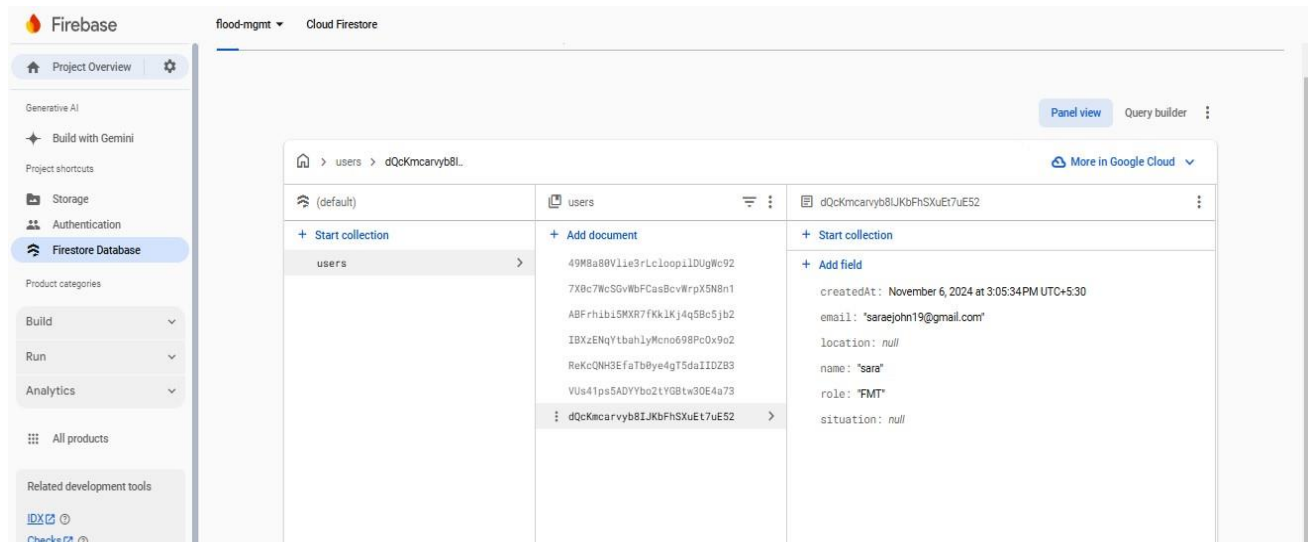


Figure 4.8.1 Firestore Database

## Chapter 5

### Conclusion

The Flood Management App serves as an integrated solution to manage and respond effectively to flood-related crises. Through its modular design, the application enables efficient communication and coordination between various stakeholders, including administrators, camp managers, quick response teams (QRTs), and users affected by floods. Each module is tailored to the specific needs of its users, ensuring that critical tasks such as registration, rescue operations, inventory management, and member registration are streamlined and efficiently executed.

For administrators, the app provides comprehensive tools for managing camps, quick response teams, members, and notifications, ensuring that resources are allocated effectively and that the needs of affected populations are met promptly. The FMT (Flood Management Team) module enhances the ability to request necessary commodities, view camp details, and assign tasks to QRTs, ensuring a smooth flow of resources and responsibilities. The QRT module allows for efficient response to user requests, registering rescued persons, and viewing critical camp details, which ensures that rescue operations are well-coordinated and tracked. Lastly, the Camp module supports camp managers in registering displaced individuals, requesting commodities, and updating member details to provide adequate shelter and services to the displaced populations.

This system not only addresses the immediate logistical needs of managing flood relief efforts but also ensures that data is captured accurately for long-term planning, resource allocation, and disaster recovery efforts. With the seamless integration of Firebase for authentication and Firestore for data storage, the app ensures real-time data access, enhancing decision-making and operational efficiency.

In conclusion, the Flood Management App is a robust tool designed to empower all involved parties to manage flood relief efforts in a more organized, timely, and efficient manner, ultimately improving the effectiveness of disaster response and minimizing the impact of floods on vulnerable populations.

## 5.1 Future Enhancements

While the current version of the Flood Management App provides a comprehensive set of features to manage flood relief operations effectively, there are several areas where future enhancements could further improve the app's functionality, scalability, and user experience.

Below are some proposed enhancements for the app:

### Offline Functionality

- **Offline Data Entry:** In areas with poor internet connectivity, it would be useful to allow camp administrators and rescue teams to enter data offline. Once a stable internet connection is available, the app could sync the data to the cloud, ensuring that all critical information is captured even in remote locations.

### Push Notifications and SMS Integration

- **Push Notifications for Urgent Alerts:** Implement push notifications to alert users about emergencies, such as rising flood levels, sudden evacuations, or important announcements from the authorities. This feature could be particularly helpful in ensuring users are kept informed in real time, even when the app isn't actively being used.

### Crowdsourced Data Collection

- **User-Generated Updates:** Allow users to report flood conditions, roadblocks, or other emergency situations through the app. These reports could be crowdsourced to provide real-time updates on flood impact and help teams direct their resources to areas with the most urgent needs.

### Automated Task Assignment for QRTs and Admins

- **Task Management for QRTs:** Introduce a feature that automatically assigns tasks to Quick Response Teams based on the severity of user requests and available resources. This would help streamline operations, reduce response time, and ensure efficient task distribution.

### **User Feedback and Improvement Loop**

- **User Feedback System:** Add a feature that allows users (e.g., camp residents, QRT members, or admin staff) to provide feedback on the app's performance. This feedback could then be used to continually improve the app's functionality and user experience.



## References

- 1    **Prof.Gayathri Ranjit** Application of Information and Communication Technologies in Natural Disasters: The case of Kerala Floods -Journal of Emerging Technologies and Innovative Research 2020
- 2    **Y. Rahimi, A. Bouguettaya, and E. Yao**, "Emergency Mobile Application for Disaster Response," in *Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Kassel, Germany, 2021, pp. 346-350.
- 3    **J. D. Franco, C. González, and P. Mendoza**, "Mobile Applications for Natural Disaster Management," in *Proceedings of the 2019 IEEE International Conference on Mobile Data Management*, Hong Kong, China, 2019, pp. 229-232.