**ActNow - COMMUNITY SERVICE CONNECTOR APP**

**CS19611 – Mobile Application Development Laboratory**

***Submitted by***

**SREENIDHI K (2116220701285)**

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

**BACHELOR OF ENGINEERING**

***in***

# COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE ANNA UNIVERSITY, CHENNAI**

# MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

**BONAFIDE CERTIFICATE**

Certified that this Project titled **“ActNow - Community Service Connector App”** is the bonafide work of **“ SREENIDHI K (2116220701285)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

# SIGNATURE

## Dr. Duraimurugan N., M.Tech., Ph.D.,

## SUPERVISOR

Professor

Department of Computer Science and Engineering,

Rajalakshmi Engineering College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on

**Internal Examiner External Examiner**

# ABSTRACT

In today's rapidly evolving society, a significant number of individuals express a desire to contribute to social causes but struggle to find opportunities that align with their skills, interests, and availability. Simultaneously, many non-profit organizations, NGOs, and local initiatives face challenges in reaching and mobilizing potential volunteers effectively. To address this gap, our project titled **"ActNow - Community Service Connector App"** presents a mobile-based solution aimed at seamlessly connecting volunteers with relevant service opportunities in their area.ActNow serves as a centralized platform where users can sign up either as Volunteers or Organizations. Volunteers can explore and register for community service events based on location, category, and skill preference, while organizations can post events and manage participant engagement. The application facilitates easy communication, provides real-time updates, and helps volunteers maintain a digital log of their contributions. The user-friendly interface ensures quick access to essential functionalities and encourages participation in social good.Developed using Android Studio with Kotlin and integrated with Firebase for data management, ActNow is lightweight, responsive, and designed for real-world deployment. The app aims to promote civic engagement, enhance the reach of charitable initiatives, and streamline the coordination between those who want to help and those in need of help. This project not only showcases our technical skills but also reflects our commitment to social responsibility.

# 

# 

# 

# ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our Project Coordinator**, Mr. N. DURAI MURUGAN, M.Tech., Ph.D.,** Professor Department of Computer Science and Engineering for his useful tips during our review to build our project and for his valuable guidance throughout the course of the project.

**SREENIDHI K** **2116220701285**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

|  |  |
| --- | --- |
| **ABSTRACT**  **ACKNOWLEDGMENT** | **iii**  **iv** |
| **LIST OF TABLES** | **viii** |
| **LIST OF FIGURES** | **ix** |
| **LIST OF ABBREVIATIONS** | **x** |
| **1. INTRODUCTION** | **11** |
| 1.1 GENERAL | 11 |
| 1.2 OBJECTIVES | 11 |
| 1.3 EXISTING SYSTEM | 12 |
| **2. LITERATURE SURVEY** | **12** |
| **3. PROPOSED SYSTEM** | **14** |
| 3.1 GENERAL | 14 |
| * 1. SYSTEM ARCHITECTURE DIAGRAM | 14 |
| 3.3 KEY MODULES | 15 |
| 3.4 DEVELOPMENT ENVIRONMENT | 16 |
| 3.4.1 HARDWARE REQUIREMENTS | 16 |
| 3.4.2 SOFTWARE REQUIREMENTS | 16 |
| 3.5 DESIGN THE ENTIRE SYSTEM | 17 |
| 3.5.1 ACTIVITY DIAGRAM | 17 |
| 3.4.2 DATA FLOW DIAGRAM | 18 |
| **4. MODULE DESCRIPTION** | **18** |

* 1. [MAIN MODULES20](#_TOC_250016)

4.1.1 [MAIN ACTIVITY 20](#_TOC_250015)

4.1.2 VOLUNTEER HOMEACTIVITY [20](#_TOC_250014)

4.1.3 ORGANIZATION HOMEACTIVITY 21

4.1.4 EVENT REPOSITORY 22

4.1.5 EVENT DATA MODEL 22

4.2 MODULE INTERACTION 23

1. IMPLEMENTATIONS AND RESULTS 24
   1. [IMPLEMENTATION 24](#_TOC_250004)
   2. DEVELOPMENT ENVIRONMENT 24
   3. FUNCTIONALITY IMPLEMENTED 24
   4. [OUTPUT SCREENSHOTS 25](#_TOC_250003)
2. CONCLUSION AND FUTURE ENHANCEMENT 28
   1. [CONCLUSION 28](#_TOC_250002)
   2. [FUTURE ENHANCEMENT 29](#_TOC_250001)

## LIST OF TABLES

**TABLE NO TITLE PAGE NO**

3.1 HARDWARE REQUIREMENTS 15

3.2 SOFTWARE REQUIREMENTS 15

## LIST OF FIGURES

**FIGURE NO TITLE** **PAGE NO**

3.1 SYSTEM ARCHITECTURE 14

3.2 ACTIVITY DIAGRAM 17

3.3 DFD DIAGRAM 18

5.1 LOGIN SCREEN WITH ROLE SELECTION 24

5.2 VOLUNTEER HOME SHOWING EVENT CARDS 25

5.3 REGISTER EVENT BUTTON AND CONFIRMATION TOAST 25

5.4 ORGANIZATION HOME WITH EVENT INPUT FORM 25

5.5 ADD EVENT SUCCESS POPUP 25

5.6 EVENTS SECTION WITH DELETE POPUP 26

5.7 VOLUNTEER HOME SHOWING THE NEW ADDED EVENT 26

## 

## LIST OF ABBREVIATIONS

| **Abbreviation** | **Full Form** |
| --- | --- |
| UI | User Interface |
| UX | User Experience |
| NGO | Non-Governmental Organization |
| API | Application Programming Interface |
| IDE | Integrated Development Environment |
| DB | Database |
| CRUD | Create, Read, Update, Delete |
| OTP | One-Time Password |
| FCM | Firebase Cloud Messaging |
| DFD | Data Flow Diagram |
| ERD | Entity Relationship Diagram |
| XML | Extensible Markup Language |
| JVM | Java Virtual Machine |
| SDK | Software Development Kit |
| MVVM | Model-View-ViewModel (optional future architecture) |
| UI/UX | User Interface / User Experience |
| PDF | Portable Document Format |

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

Volunteering is a vital component of social development and community growth. Despite the willingness of many individuals to contribute to social causes, there exists a lack of proper channels to connect them with organizations in need of assistance. This disconnect leads to underutilized human potential and inefficiencies in organizing community-based activities.**ActNow - Community Service Connector App** is designed to bridge this gap by providing a mobile platform that facilitates collaboration between volunteers and organizations. The app simplifies the process of discovering, joining, and managing social service events. Volunteers can sign up, browse events by interest or location, and register with a single click. Organizations, in turn, can post events, manage volunteer participation, and track event effectiveness.This mobile application aims to enhance civic engagement, ensure smoother operations for social organizations, and promote transparency and accountability in volunteer efforts. By digitizing the volunteer management process, ActNow aspires to become a valuable tool for social impact.

# OBJECTIVE

The primary objective of this project is to design and implement a mobile application that:

* Provides a centralized platform for volunteers and organizations.
* Allows volunteers to discover and join social events based on location, date, and interest.
* Enables organizations to post, edit, and manage volunteer events.
* Tracks volunteer participation and service hours.
* Encourages consistent civic engagement through timely notifications and an intuitive interface.

# 

# EXISTING SYSTEM

In the current landscape, volunteer coordination is often conducted through manual means such as WhatsApp groups, social media pages, and spreadsheets. These methods are disorganized, lack personalization, and are difficult to scale. Additionally, volunteers often miss opportunities due to lack of timely information, while organizations struggle to reach the right audience efficiently.Some existing platforms offer event listings or social cause networking, but they are either overly complex, not tailored for volunteer-based community service, or lack proper integration with mobile devices. This creates a need for a purpose-built app that focuses solely on connecting service-minded individuals with credible, real-time opportunities.

# CHAPTER 2 LITERATURE SURVEY

# A literature survey is essential in understanding the scope of existing technologies and methodologies that address the problem of volunteer management and social engagement. The research reveals that although several applications and platforms facilitate community interaction, very few are focused specifically on structured, localized volunteer coordination and event participation.

# 

# 2.1 Existing Applications

# FacebookGroups/WhatsAppGroups Many volunteer-based organizations currently use informal platforms such as Facebook groups or WhatsApp to manage events. These methods are manual, lack structure, and do not provide analytics, notifications, or participation tracking.

# GiveGab/VolunteerMatch These platforms offer web-based volunteering networks. However, they are primarily U.S.-based, often too complex for local NGOs, and not mobile-first. Moreover, they lack real-time interaction and event management features suited for smaller communities.

# GoogleForms+Sheets A commonly used alternative is collecting registrations through Google Forms and managing lists in Google Sheets. This requires manual follow-up and is not integrated with a notification or reminder system, leading to participant drop-off and inefficiencies.

# 2.2 Limitations of Existing Systems

# No personalized matching of volunteer interests and skills with events.

# Lack of real-time updates or reminders about event changes.

# Absence of a user-friendly mobile interface for on-the-go volunteering.

# No centralized database for tracking volunteer hours or generating participation records.

# Difficulty for NGOs to manage recurring volunteers or edit/update events dynamically.

# 2.3 Need for the Proposed System

# The proposed mobile application ActNow fills this gap by providing:

# A simplified, intuitive interface for both volunteers and organizers.

# Real-time event listing, registration, and notifications.

# Role-based login for separate functionalities.

# Local storage-based participation logs with the potential for future integration with Firebase for cloud syncing.

# An inclusive and community-focused digital platform to streamline social contribution.

**CHAPTER 3**

**PROPOSED SYSTEM**

* 1. **GENERAL**

The ActNow - Community Service Connector App is a mobile application designed to bridge the gap between volunteers and organizations. It simplifies the process of finding and managing social service activities through an intuitive, mobile-first interface. The proposed system aims to address the limitations of existing solutions by providing a centralized, easy-to-use, and real-time platform for both event organizers and volunteers.

# SYSTEM ARCHITECTURE DIAGRAM

# The application follows a modular, role-based structure:

# MainActivity: A login and role-selection screen where users enter their name, email, and select their role as either "Volunteer" or "Organization".

# VolunteerHomeActivity: Displays a list of community service events. Volunteers can browse, register, and view their registered events.

# OrganizationHomeActivity: Allows organizations to add, manage, and delete events.

# EventRepository: A central data store that holds shared events between volunteers and organizations.

# The application is developed using Kotlin and Jetpack Compose within Android Studio, following clean UI practices and structured navigation.

# KEY MODULES

 **Role-based Login**  
Users select whether they are a Volunteer or an Organization. Based on the role, they are redirected to the appropriate home screen.

**Event Creation and Management**  
Organizations can add events by entering the event name, description, and location. Events are dynamically stored and can be deleted if needed.

 **Volunteer Event Registration**  
Volunteers can view all events and register with a single click. Registered events are stored locally and can be reviewed via a dedicated "Registered Events" section.

 **UI & Styling Enhancements**  
The app includes colorful themes, rounded card corners, material components, scrollable views, and toast messages for interactivity.

# 

**Fig 3.1: System Architecture**

# DEVELOPMENTAL ENVIRONMENT

* + 1. **HARDWARE REQUIREMENTS**

The hardware specifications serve as the foundation for implementing and running the mobile application smoothly. A clear understanding of hardware requirements is crucial to ensure compatibility, responsiveness, and stability during both development and usage. These specifications help guide design decisions and ensure reliable performance on real Android devices and emulators.

**Table 3.1 Hardware Requirements**

|  |  |
| --- | --- |
| **COMPONENTS** | **SPECIFICATION** |
| PROCESSOR | Intel i5 or above (recommended) |
| RAM | 8 GB RAM OR Higher |
| HARD DISK | Minimum 4 GB free space |
| DISPLAY | 1280 x 720 resolution or higher |
| SMARTPHONE | Android 7.0 (API 24) and above |

# SOFTWARE REQUIREMENTS

# The software requirements define the essential components required for the design, development, testing, and deployment of the ActNow - Community Service Connector App. These specifications ensure that developers have the necessary tools and environments for efficient development. They also help in planning, cost estimation, task allocation, and version control throughout the development lifecycle.

**Table 3.2 Software Requirements**

| SOFTWARE COMPONENTS | DESCRIPTION |
| --- | --- |
| Operating System | Windows 10 / macOS / Linux |
| IDE | Android Studio (Electric Eel / later) |
| Programming Language | Kotlin (with Jetpack Compose) |
| Design Tool | XML for UI (Jetpack Compose UI Toolkit) |
| Emulator / Device | Android Emulator or physical Android phone |
| Database (Optional) | Firebase (can be added in future) |

# DESIGN OF THE ENTIRE SYSTEM

# 

# ACTIVITY DIAGRAM

The Activity Diagram models the logical flow of the application from launch to interaction. It shows how a user begins by opening the app, selects their role, and then proceeds to either volunteer-related or organization-related activities. This diagram helps visualize the user's journey and how different paths diverge based on role.

**Key Activities:**

* Launch the app.
* Select role: Volunteer or Organization.
* Volunteer:
  + View list of events.
  + Register for events.
* Organization:
  + Add new event.
  + Delete existing events.

The flow ends once the user completes their interaction or exits the app.

# 

**Fig 3.2: Activity Diagram**

# 3.4.2 DATA FLOW DIAGRAM

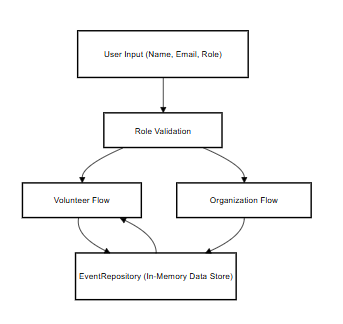
The Data Flow Diagram (DFD) describes how data moves within the system at a high level. It identifies the main processes involved in handling user input and shows the interaction with the data store (in this case, the EventRepository).

**Components:**

* **User Input**: The user provides their name, email, and selects their role.
* **Role Check**: Determines whether the user is a volunteer or organization.
* **Volunteer Process**:
  + Fetches and displays events.
  + Sends registration info to the data store.
* **Organization Process**:
  + Allows event creation or deletion.
  + Sends updates to the shared event store.
* **EventRepository**:
  + Acts as a shared in-memory storage that both user types interact with.

This flow ensures data consistency and coordination between volunteers and organizations, even without a backend database.

.



**Fig 3.3:Data Flow Diagram**

# CHAPTER 4 MODULE DESCRIPTION

The **ActNow** application is divided into well-defined modules, each responsible for specific functionality based on the user's role: **Volunteer** or **Organization**. These modules ensure a clean separation of concerns, ease of maintenance, and better user experience.

**4.1 MAIN MODULES**

**4.1.1 MainActivity (Login & Role Selection)**

This is the first screen the user interacts with. It allows users to:

* Enter their name and email.
* Select their role (Volunteer or Organization).
* Redirects users to the appropriate screen based on the selected role.

**Features:**

* Basic input validation.
* Radio buttons for role selection.
* Intent-based navigation to respective home activities.

**4.1.2 VolunteerHomeActivity**

This screen is designed for users who register as volunteers.

**Features:**

* Displays a scrollable list of upcoming and default events.
* Allows volunteers to register for events.
* Registered events are stored locally and viewable separately.
* Includes a Back button to return to the main screen.
* Each event opens a dialog with details (description & location).

**UI Components Used:**

* Card layouts
* Buttons
* Toasts
* AlertDialog
* ScrollView (vertical)

**4.1.3 OrganizationHomeActivity**

This module serves the event organizers.

**Features:**

* Allows organizations to create new events by entering name, description, and location.
* Events are added dynamically to a shared list (EventRepository).
* Each added event appears in a scrollable list with a Delete button.
* UI is styled with rounded corners and shadows for professional look.
* Includes Back to Main functionality.

UI Components Used:

* OutlinedTextField
* Material Buttons
* Toast messages
* MutableStateList (for live event updates)

**4.1.4 EventRepository (Shared Event Data Storage)**

This is a singleton Kotlin object that acts as a central event list shared between volunteers and organizations.

**Purpose:**

* Store default events and dynamically added events.
* Ensure both roles view the same updated event list without using a real-time database.

**Data Structure:**

**object EventRepository {**

**val eventList = mutableListOf<Event>()**

**}**

**4.1.5 Event Data Model**

This is a simple Kotlin data class used to represent each event.

**data class Event(**

**val name: String,**

**val description: String,**

**val location: String**

**)**

It ensures structured and consistent handling of event-related data throughout the app.

**4.2 MODULE INTERACTION**

* MainActivity handles login and redirects.
* VolunteerHomeActivity and OrganizationHomeActivity both access and update the same EventRepository.
* UI is rendered using Jetpack Compose, and logic is handled through Kotlin's remember and mutableStateOf for ractive updates.

# CHAPTER 5 IMPLEMENTATION AND RESULTS

# IMPLEMENTATION

# This chapter explains how the ****ActNow - Community Service Connector App**** was implemented using Android Studio and Kotlin, and presents the results observed during development and testing. The app was built using ****Jetpack Compose**** for modern UI development, ensuring responsive design, component reusability, and clean architecture.

# 5.2 DEVELOPMENT ENVIRONMENT

| Component | Description |
| --- | --- |
| IDE | Android Studio (Hedgehog / Electric Eel) |
| Language | Kotlin |
| UI Framework | Jetpack Compose |
| OS for Development | Windows 10 / macOS |
| Emulator | Android Emulator (API 30+) or real device |

# FUNCTIONALITY IMPLEMENTED:

# The following features were successfully implemented and tested:

# Login & Role Selection

# Users input name and email, and select a role: Volunteer or Organization.

# Role-based redirection is handled using Kotlin's Intent.

# Volunteer Functionality

# Scrollable list of community service events.

# Each event has a Register button.

# Clicked events are saved to a Registered Events list and shown in a dialog.

# Event details shown in popup dialog on click.

# Back button allows return to login screen.

# Organization Functionality

# Can add events with name, description, and location.

# Events are stored in a shared repository (EventRepository) and reflected in volunteer screen.

# Added events are listed with a Delete button.

# Scrollable UI for managing long lists of events.

# Form fields auto-reset after successful event submission.

# Shared Repository

# A singleton object (EventRepository) holds the current list of events.

# Ensures both Volunteer and Organization roles interact with the same live data.

# UI Styling

# Modern UI using Material Design 3.

# Colorful themes, button highlights, and shadowed cards for aesthetics.

# Toasts used to provide user feedback for actions.

# OUTPUT SCREENSHOTS

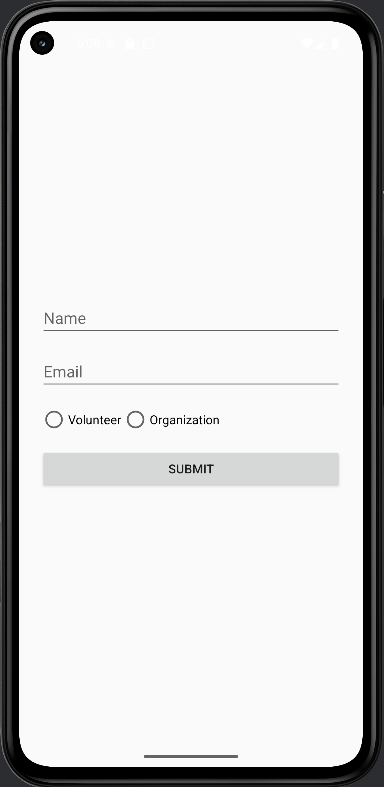


Fig 5.1 Login Screen with role selection

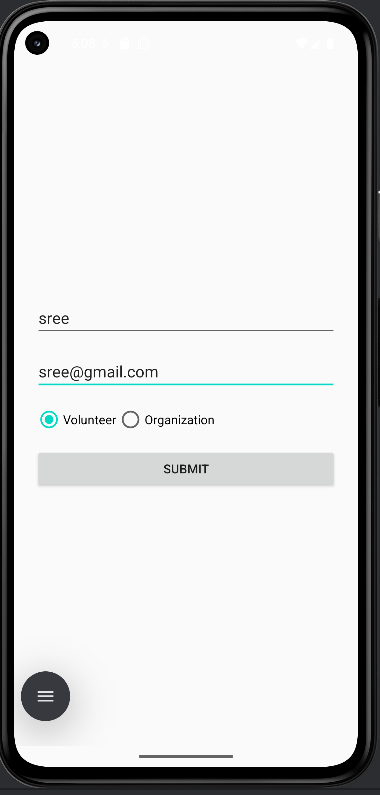
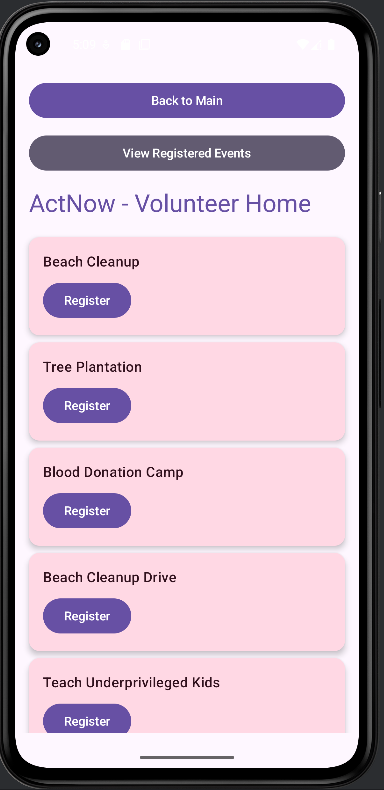
 

Fig 5.2 Volunteer home showing event cards

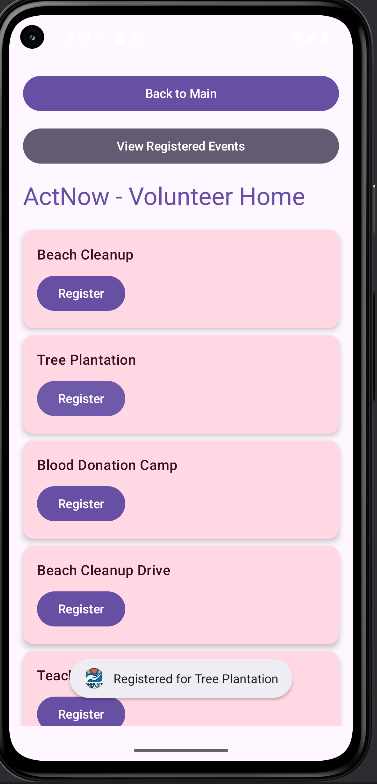
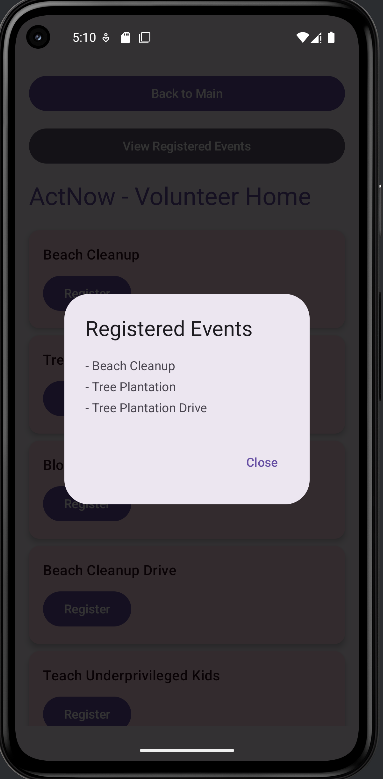
 

Fig 5.3 Register event button and confirmation toast

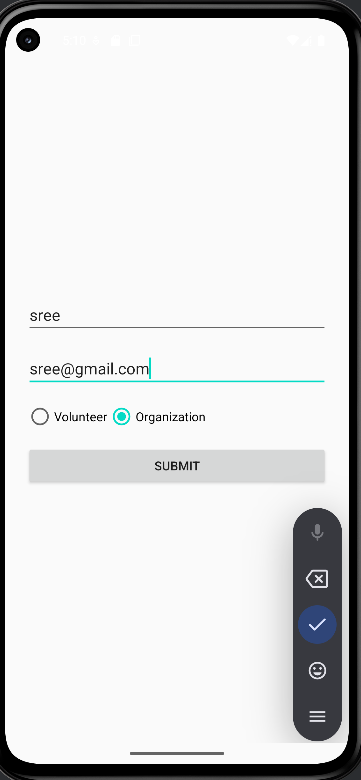
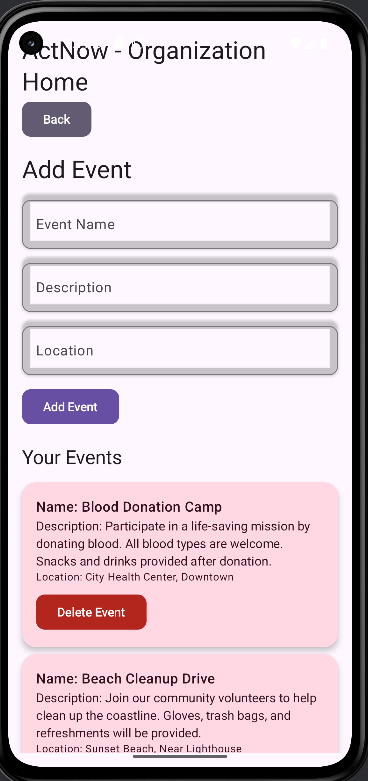
** **

Fig 5.4 Organization home with event input form

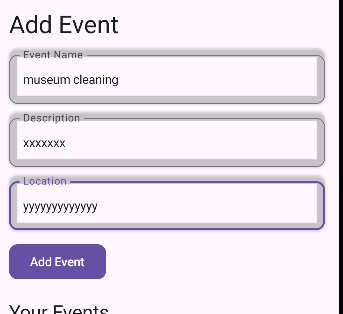
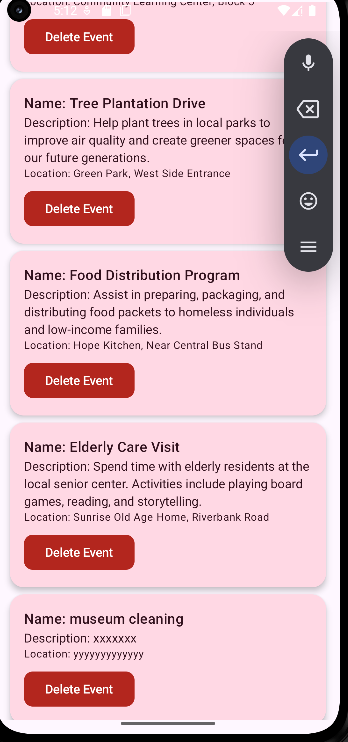
** **

Fig 5.5 Add event success popup

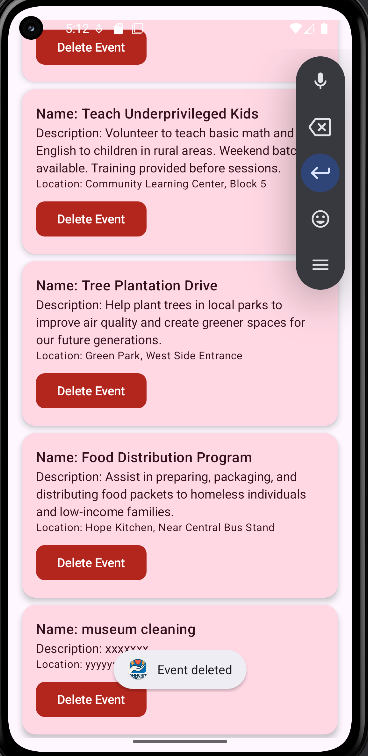
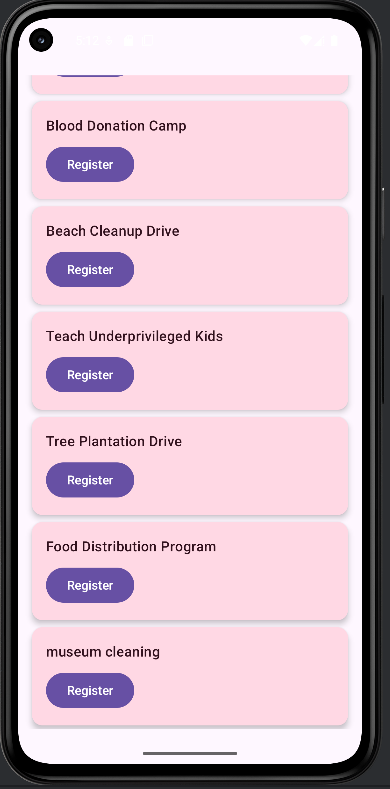
** **

Fig 5.6 Events section with delete popup

Fig 5.7 Volunteer Home showing the new added event

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

# CONCLUSION

# The ActNow - Community Service Connector App was developed to address a real-world problem: the lack of an efficient, accessible platform to connect volunteers with social service opportunities. Through this mobile application, we have successfully created a system where:

# Volunteers can easily explore and register for events relevant to their interests and location.

# Organizations can post, manage, and track volunteer participation.

# A shared repository ensures consistent, real-time event visibility between both user roles.

# A clean and modern UI improves usability, engagement, and accessibility.

# The app’s architecture promotes modularity, scalability, and maintainability, making it easy to add new features in the future. The use of Jetpack Compose enabled quick and elegant UI development while Kotlin provided the flexibility and safety required for robust mobile applications. This project not only allowed us to apply practical software development skills but also encouraged us to contribute toward solving a meaningful community challenge.

# FUTURE ENHANCEMENT

While the current version of ActNow meets the essential goals, several features are planned to make the app more powerful and scalable:

* **Firebase Integration:**Store user data and event lists in real-time databases to enable persistent storage and cross-device access.
* **Authentication System:**Implement secure login using email/password, Google Sign-In, or OTP-based login for identity verification.
* **Digital Certificates:**Automatically generate and issue digital certificates or badges to volunteers for participation.
* **Map Integration:**Use Google Maps API to show event locations and directions for better navigation.
* **Notification System:**Add Firebase Cloud Messaging to send real-time updates and event reminders.
* **Chat & Community Forum:**Enable communication between volunteers and organizers to discuss event logistics.
* **Admin Panel:**For verifying NGOs, moderating content, and tracking participation analytics.
* **LinkedIn / Resume Export:**Allow users to export their volunteering history as a certificate or add it directly to their professional profiles.

With these future enhancements, ActNow has the potential to evolve into a full-fledged platform that truly bridges the gap between willing volunteers and impactful social causes.