# Project Report: Volunteer Management System (Volunty)

## Problem Statement

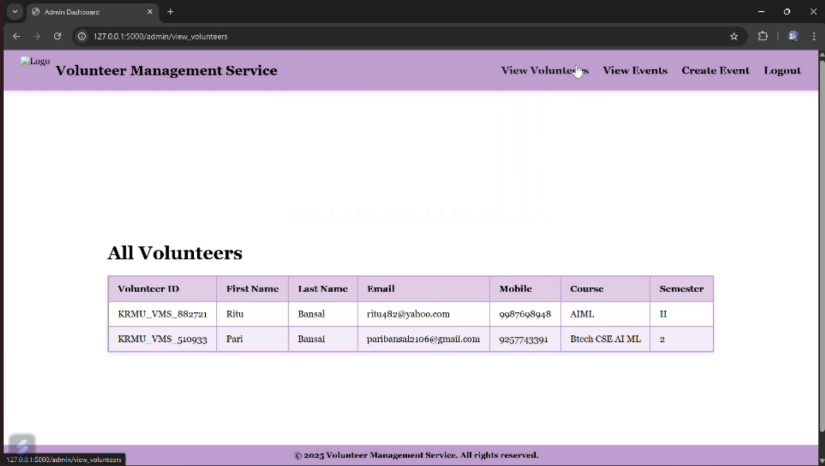
Campus events often require volunteer support, yet organizing and managing volunteers manually is inefficient, error-prone, and time-consuming. Many institutions lack a centralized platform that allows event organizers to easily connect with willing student volunteers, track participation, and manage event logistics in one place.

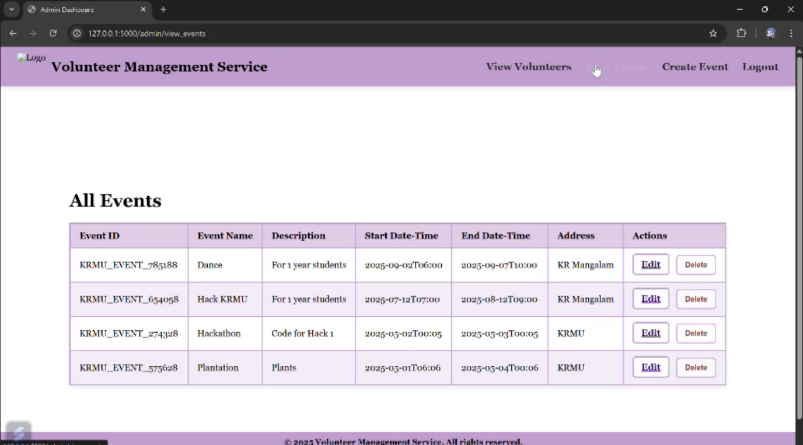
## Project Objectives

- Develop a centralized web-based platform for organizers and volunteers.  
- Allow event creation, editing, deletion, and poster uploading by organizers.  
- Enable volunteers to sign up, complete a profile, and register for events.  
- Provide multiple sign-in methods (Google, Facebook, Phone).  
- Ensure responsive and user-friendly UI for both organizers and volunteers.  
- Enable theme switching (light/dark) for accessibility and user preference.

## Frontend

Technologies Used:  
- HTML5 & CSS3: Structure and styling  
- JavaScript: Client-side logic and dynamic interactions  
- Responsive Design: Ensures usability across devices  
- Theme Toggle: Implemented via JS & CSS (dark/light mode toggle with icons)  
  
Features Implemented:  
- Modal-based login with role selection (Organizer/Participant).  
- Form-based volunteer registration with fields like Name, Age, Course, etc.  
- Role-specific dashboards:  
 - Organizer Dashboard: Create/post events, upload posters, delete events.  
 - Volunteer Dashboard: View upcoming events, sign up, and see status.





## Backend

Note: The provided implementation appears frontend-focused, but here's the scalable plan for full backend integration.  
  
Planned Technologies:  
- Node.js with Express.js or Python Flask: REST API services  
- Authentication: Firebase Auth, OAuth2.0 (Google/Facebook), Twilio (for OTP)  
- Form Data Handling: Server receives, validates, and stores user input.

## 

## Database Design

Tools: Firebase Firestore / MongoDB / MySQL (based on tech stack)  
  
Tables / Collections:  
- Users: Stores user profiles including role, personal data, and login method.  
- Events: Stores event name, description, date, time, poster image URL, createdBy.  
- Volunteers: Tracks event sign-ups per user with timestamp and status.  
  
Key Queries:  
- Fetch events by organizer.  
- Fetch available events by date or popularity.  
- Join volunteers with events to track participation.

## Algorithms / Logic Implemented

- Dynamic Form Rendering: Based on login method selected.  
- Conditional Role Routing: Determines interface based on Organizer/Volunteer.  
- Event Deletion Logic: Event DOM element is removed on button click.  
- Poster Upload Handling: Reads image using FileReader() and displays preview.  
- Dark Mode Toggle: Uses a JS toggle to switch CSS variables/theme classes.  
- Validation Logic: Ensures form fields are filled before event creation/sign-in.

## Security (Planned)

- Role-based access control for organizers and volunteers.  
- Input validation and sanitization to avoid XSS or data corruption.  
- User authentication and session handling using Firebase/Auth0.

## Future Goals

- Backend Integration: Full support for login, session tracking, and database storage.  
- Event Analytics: Dashboard for organizers to view volunteer count, feedback.  
- Volunteer Badging System: Recognition for top contributors.  
- Email/Push Notifications: Event reminders, confirmations, and updates.  
- Admin Panel: Oversee platform activity, moderate content, and manage users.  
- Mobile App Version: React Native-based version for mobile accessibility.

## Conclusion

The Volunteer Management System serves as a modern, scalable solution for bridging the gap between event organizers and volunteers on campus. It simplifies communication, improves event planning efficiency, and creates a transparent system for participation tracking. With further development and integration, this platform can serve as a full-fledged event-volunteer ecosystem for educational institutions.