



SHAHZAIB IQBAL

**01-134192-106**

MALIK ZOHAIB MUSTAFA

**01-134192-030**

# A Collaborative Platform for Community Services

**Bachelor of Science in Computer Science**

Supervisor: Mrs. Saima Jawad

Department of Computer Science  
Bahria University, Islamabad

May 2023



## **Certificate**

We accept the work contained in the report titled “A Collaborative Platform for Community Support”, written by Mr.Shahzaib Iqbal AND Mr. Malik Zohaib Mustafa as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by ....:

Supervisor: Mrs. Saima Jawad

---

Internal Examiner: Dr Hafiz Ishfaq Ahmed

---

External Examiner: Dr. Labiba Gillani Fahad

---

Project Coordinator: Ms. Maryam Khalid Multani

---

Head of the Department: Dr. Arif ur Rahman

---

1<sup>th</sup>, May 2023

# **Abstract**

The Community Support Portal is a non-profit web-based platform designed to address the challenges faced by small-scale charitable organizations. These organizations often struggle with a lack of funding, volunteers, and digital presence, which can hinder their ability to provide effective assistance to those in need. Additionally, the lack of collaboration among NGOs can make it difficult to coordinate efforts during disasters, leading to some victims being left unaided. The main objective of this project is to facilitate collaboration between NGOs and volunteers for effective community services. Through the use of this platform, NGOs can easily communicate and share resources with each other to better serve the community. Small non-profit organizations can also quickly find volunteers for community work. By bringing together different NGOs and volunteers, the platform aims to create a more cohesive and effective community service network. To achieve this goal, we will follow a complete software development plan and apply a variety of techniques, as well as a number of different technologies. The development process will be carried out in a structured and organized manner, ensuring that the platform is efficient, user-friendly, and meets the needs of its intended users. After deploying this project, we hope to create a more collaborative and efficient community service network, where NGOs and volunteers can work together to provide timely and appropriate aid to those in need.

# **Acknowledgments**

I would like to express my heartfelt gratitude to all those who supported me in the completion of this project. First and foremost, I extend my sincere thanks to my project supervisor, Mrs. Saima Jawad for their guidance and support throughout the project. Their insightful feedback, timely suggestions, and constant encouragement have been invaluable in shaping this project. I am also deeply grateful to the staff of various non-profit organizations and community members who provided valuable feedback and insights into the project's requirements. Their support and enthusiasm for this project were a significant motivating factor throughout the development process.

Furthermore, I would like to thank each and every teacher, friend, and family for their unwavering support and encouragement throughout this project. Their moral support and motivation provided me with the strength to overcome any obstacles and complete this project successfully. Finally, I extend my gratitude to all those who have contributed in some way to this project. Your contributions and support have been essential in the development of this collaborative platform for community services.

**SHAHZAIB IQBAL, MALIK ZOHAIB MUSTAFA**  
Bahria University Islamabad, Pakistan

May 2023

*“We think someone else, someone smarter than us,  
someone more capable, someone with more resources will solve that problem.  
But there isn’t anyone else.”*

Regina Dugan

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Background . . . . .	1
1.2	Problem Description . . . . .	3
1.3	Project Objectives . . . . .	4
1.4	Methodology . . . . .	4
1.5	Project Scope . . . . .	6
1.5.1	Limitations . . . . .	7
1.5.2	Features . . . . .	7
<b>2</b>	<b>Literature Review</b>	<b>11</b>
2.1	Community Support Systems in Pakistan . . . . .	11
2.1.1	The Citizens Foundation (TCF) . . . . .	11
2.1.2	Edhi Foundation . . . . .	12
2.1.3	Pakistan Poverty Alleviation Fund (PPAF ) . . . . .	13
2.1.4	Shaukat Khanum Memorial Cancer Hospital and Research Centre . . . . .	13
2.1.5	Alkhidmat Foundation . . . . .	14
2.2	Community Support systems Overseas . . . . .	15
2.2.1	Charity Navigator . . . . .	15
2.2.2	Givewell . . . . .	15
2.2.3	GlobalGiving . . . . .	16
2.3	Comparison with the Community Support Platform . . . . .	17
2.4	Conclusion . . . . .	17
2.5	Comparison Table . . . . .	17
<b>3</b>	<b>Requirement Specifications</b>	<b>19</b>
3.1	Existing systems . . . . .	19
3.1.1	Limitations . . . . .	20
3.1.2	Drawbacks . . . . .	20
3.2	Proposed System . . . . .	21
3.3	Functional Requirements . . . . .	21
3.3.1	User Registrations and Login . . . . .	22
3.3.2	Service Category Management . . . . .	22
3.3.3	User and Project Management . . . . .	22
3.3.4	Community Service Hours . . . . .	22
3.3.5	E-certificate Issuance . . . . .	22
3.3.6	Collaboration Tools . . . . .	22
3.3.7	Communication tools . . . . .	22

3.3.8	Support options . . . . .	23
3.4	Non-Functional Requirements . . . . .	23
3.4.1	Performance . . . . .	23
3.4.2	Security . . . . .	23
3.4.3	Usability . . . . .	23
3.4.4	Compatibility . . . . .	23
3.4.5	Maintenance . . . . .	23
3.4.6	Scalability . . . . .	23
3.4.7	High availability . . . . .	24
3.4.8	Reliability . . . . .	24
3.4.9	Data management and security features . . . . .	24
3.5	Use Cases . . . . .	24
3.5.1	Login/Register Use case . . . . .	25
3.5.2	Specific Volunteer Use Case: . . . . .	25
3.5.3	Specific NGO Use Case: . . . . .	26
3.5.4	Specific Admin Use Case: . . . . .	27
3.6	Tabular Descriptions of Use Cases . . . . .	28
3.6.1	Use Tabular Case Description for Login: . . . . .	28
3.6.2	Use Case Tabular Description for Requesting E-Certificate . . . . .	28
3.6.3	Use Case Tabular Description for Giving Reward Points . . . . .	29
3.6.4	Use Case Tabular Description for Approve NGO Requests . . . . .	30
<b>4</b>	<b>Design</b>	<b>31</b>
4.1	System Architecture . . . . .	31
4.2	Design Constraints . . . . .	32
4.2.1	Programming Language: . . . . .	32
4.2.2	Design Standard . . . . .	32
4.2.3	Assumptions and Dependencies . . . . .	32
4.3	Sequence Diagrams . . . . .	33
4.3.1	Overall System . . . . .	33
4.3.2	Login/Register . . . . .	34
4.3.3	Project Sharing . . . . .	34
4.3.4	Project Approval . . . . .	35
4.3.5	Assigning Projects . . . . .	35
4.3.6	Search . . . . .	36
4.3.7	Admin can edit NGOs/Volunteers . . . . .	36
4.3.8	Admin can delete NGOs/Volunteers . . . . .	37
4.3.9	Logout . . . . .	37
4.4	GUI Design . . . . .	38
4.4.1	Homepage . . . . .	38
4.4.2	Services . . . . .	39
4.4.3	Active Projects . . . . .	39
4.4.4	Partner NGO's . . . . .	40
4.4.5	Footer . . . . .	40
4.4.6	NGO Profile . . . . .	41
4.4.7	Admin Dashboard . . . . .	42
4.4.8	Customization Page . . . . .	42

4.4.9	Projects Page . . . . .	43
4.4.10	View Projects . . . . .	44
4.4.11	Volunteers . . . . .	44
4.4.12	Manage Requests . . . . .	45
4.4.13	Registered NGO's . . . . .	45
4.4.14	Volunteer Dashboard . . . . .	46
4.4.15	Volunteer Profile Information . . . . .	46
4.4.16	Projects . . . . .	47
4.4.17	View all Projects . . . . .	48
4.4.18	Donations . . . . .	48
4.4.19	Rewards . . . . .	49
4.4.20	NGO Dashboard . . . . .	49
4.4.21	NGO Profile . . . . .	50
4.4.22	Projects . . . . .	51
4.4.23	Volunteers page . . . . .	51
4.4.24	Awards Page . . . . .	52
4.5	External Interfaces . . . . .	52
<b>5</b>	<b>System Implementation</b>	<b>53</b>
5.1	System Architecture . . . . .	53
5.1.1	Client Layer . . . . .	53
5.1.2	Application Layer . . . . .	53
5.1.3	Database Layer . . . . .	54
5.2	Internal System Components . . . . .	54
5.2.1	User Management . . . . .	54
5.2.2	Project Management . . . . .	55
5.2.3	Communication and Collaboration . . . . .	55
5.2.4	Feedback and Reviews . . . . .	56
5.2.5	Volunteer Skill Matching . . . . .	56
5.2.6	Knowledge Sharing and Best Practices . . . . .	56
5.2.7	Scheduling and Availability Management . . . . .	56
5.3	Tools and Technology . . . . .	57
5.4	System Methodology . . . . .	57
<b>6</b>	<b>System Testing and Evaluation</b>	<b>59</b>
6.1	Introduction . . . . .	59
6.2	Graphical User Interface Testing . . . . .	59
6.2.1	Navigation Testing . . . . .	59
6.2.2	Form Testing . . . . .	60
6.3	Usability Testing . . . . .	60
6.3.1	User Scenarios . . . . .	60
6.3.2	User Feedback . . . . .	60
6.4	Integration Testing . . . . .	61
6.5	Exception Testing . . . . .	61
6.5.1	Input Validation . . . . .	61
6.5.2	Error Handling . . . . .	61
6.6	Performance Testing . . . . .	62

6.6.1	Load Testing . . . . .	62
6.6.2	Stress Testing . . . . .	62
6.7	Installation/Deployment Testing . . . . .	62
6.8	Test Cases . . . . .	62
<b>7</b>	<b>Conclusions</b>	<b>69</b>
7.1	Introduction . . . . .	69
7.2	Problems Faced and Lessons Learned . . . . .	69
7.3	Project Summary . . . . .	70
7.4	Future Improvements . . . . .	71
<b>A</b>	<b>User Manual</b>	<b>72</b>
	<b>References</b>	<b>73</b>

# List of Figures

1.1	Overview . . . . .	6
2.1	About Page Of TCF . . . . .	12
2.2	Homepage of Edhi . . . . .	12
2.3	About Page of PPAF . . . . .	13
2.4	Homepage of SKMT . . . . .	14
2.5	About Page of Alkhidmat . . . . .	14
2.6	Homepage of Charity Navigator . . . . .	15
2.7	About Page of Givewell . . . . .	16
2.8	Homepage of Global Giving . . . . .	16
3.1	Login/Register Use case . . . . .	25
3.2	Specific Volunteer Use Case . . . . .	26
3.3	Specific NGO Use Case . . . . .	27
3.4	Specific Admin Use Case . . . . .	27
4.1	System Architecture . . . . .	32
4.2	Sequence Diagram for Overall System . . . . .	33
4.3	Sequence Diagram for Login/Register . . . . .	34
4.4	Sequence Diagram for Project Sharing . . . . .	34
4.5	Sequence Diagram for Project Approval . . . . .	35
4.6	Sequence Diagram for Assigning Project . . . . .	35
4.7	Sequence Diagram for Search . . . . .	36
4.8	Sequence Diagram for Admin Edit Data . . . . .	36
4.9	Sequence Diagram for Admin Delete Data . . . . .	37
4.10	Sequence Diagram for Logout . . . . .	37
4.11	Homepage of Proposed Community Service Platform . . . . .	38
4.12	Offered Services Design . . . . .	39
4.13	Active Projects Design . . . . .	39
4.14	Partner NGO's Design . . . . .	40
4.15	Footer Design . . . . .	40
4.16	NGO Details Page Design . . . . .	41
4.17	Admin Dashboard Design . . . . .	42
4.18	Customization Page Design . . . . .	43
4.19	Add Project Design . . . . .	43
4.20	View All Projects Design . . . . .	44
4.21	Volunteers Page Design . . . . .	44
4.22	Manage Requests Design . . . . .	45

4.23 Registered NGOs Page Design . . . . .	45
4.24 Volunteer Dashboard Design . . . . .	46
4.25 Volunteer Profile Design . . . . .	47
4.26 Volunteer Projects Design . . . . .	47
4.27 Volunteer All Projects Design . . . . .	48
4.28 Donations Page Design . . . . .	48
4.29 Reward Page Design . . . . .	49
4.30 NGO Dashboard Design . . . . .	50
4.31 NGO Profile Information . . . . .	50
4.32 NGO Add Project . . . . .	51
4.33 Volunteer Page Design . . . . .	51
4.34 Awards Page Design . . . . .	52

# List of Tables

2.1 Comparison Table between the features of some Community Service Platforms . . . . .	18
3.1 Tabular Description for Login . . . . .	28
3.2 Tabular Description for Requesting E-Certificates . . . . .	29
3.3 Tabular Description for Giving Reward Points . . . . .	29
3.4 Tabular Description for Approving NGO Requests . . . . .	30
6.1 User Registration Test Case . . . . .	63
6.2 User Login Test Case . . . . .	63
6.3 User Profile Update Test Case . . . . .	63
6.4 Create new Project Test Case . . . . .	64
6.5 Edit Existing Project Test Case . . . . .	64
6.6 Delete Existing Project Test Case . . . . .	64
6.7 Complete a Project Test Case . . . . .	65
6.8 Send Rewards Request Test Case . . . . .	65
6.9 Accept Rewards Test Case . . . . .	66
6.10 Provide Ratings to NGO Test Case . . . . .	66
6.11 Search for a Project Test Case . . . . .	66
6.12 User view profile Test Case . . . . .	67
6.13 View Project Details Test Case . . . . .	67
6.14 Leave a project Test Case . . . . .	67
6.15 Volunteer for a project Test Case . . . . .	68
6.16 View Volunteers of Project Test Case . . . . .	68

# **Acronyms and Abbreviations**

NGO	Non-Profit Organization
CSP	Community Service Points
MERN	MongoDB, Express, React, Node
TDD	Test-Driven Development
HTML	HyperText Markup Language
JS	JavaScript
CSS	Cascading Style Sheets
SQL	Structured Query Language
JSON	JavaScript Object Notation

# Chapter 1

## Introduction

### 1.1 Project Background

The concept of community support has been a vital aspect of human society for centuries. However, with the advent of technology, there has been an increasing need for digital solutions to support community engagement and collaboration. Traditional forms of community support, such as volunteering and donating, have been hindered by a lack of communication and cooperation. It can significantly slow down the effectiveness of community support systems, as listed below:

- **Inefficient use of resources:** Different organizations and individuals may be duplicating efforts without adequate communication and collaboration. Using resources inefficiently can result in a lack of coordination and lead to limited impact.
- **Limited reach:** Community support systems may struggle to reach those in need without collaboration and communication. For example, a lack of communication between different organizations can result in gaps in service delivery, leaving some individuals or groups without access to support.
- **Reduced effectiveness:** When communication and collaboration are limited, community support systems may be less effective in addressing community issues. For example, without cooperation and communication, organizations may be unable to share best practices or learn from one another, resulting in less effective service delivery.

- **Duplication of efforts:** Without proper communication and collaboration, different organizations may be working on similar or identical projects, resulting in a waste of resources and a lack of progress.
- **Lack of trust and cooperation:** In cases of poor communication and lack of collaboration, it's hard for the organizations and individuals to establish trust, teamwork, and mutual understanding, which could make it harder for them to work effectively together.
- **Lack of Transparency:** Communication and collaboration are essential for ensuring transparency and accountability within community support systems. In their absence, tracking the use of resources and the impact of different initiatives can be difficult, leading to a lack of trust and confidence in the support systems.

Lack of collaboration and communication can cause community support systems to be less efficient, less effective, less transparent, and less able to reach those in need. Community support systems can more successfully address the community's needs by fostering collaboration and communication.

The rapid development of technology and the internet has led to increased use of online platforms for communication and collaboration. This has created new opportunities and challenges for communities, as people can now share information in impossible ways. One area that has seen significant growth in recent years is the use of online platforms for community support, which are used for social and economic issues. Social media has been disseminated to a broader audience, including its severity and impact. This has helped to raise awareness of the disaster and mobilize support from local and international communities. It has also given flood victims a way to interact with one another and with aid agencies. People can now donate as much as they can, thanks to it. It has been employed to raise money for relief efforts. Many people and organizations have utilized social media platforms to start fundraising campaigns and collect donations.

With the increasing complexity of modern societies, communities face various challenges, including poverty, inequality, and social isolation. Additionally, many traditional forms of community support, such as volunteering and charitable giving, have become less effective in addressing these issues. This is partly due to a need for more connectivity and communication among individuals and organizations working to support their communities. The project addresses this problem by creating a digital platform that provides a one-stop shop for

community support, including volunteer opportunities, donation tracking, and community events.

The Community Support Portal is based on developing a nonprofit collaborative platform for Non-Governmental Organizations (NGOs)/charity organizations. It brings together these organizations and volunteers for effective community service. It provides a platform to share ideas, collaborate and contribute to community service initiatives. It enables users to be aware of charity work being done by charity organizations and helps them get involved in community service initiatives.

## 1.2 Problem Description

The main problem is “lack of collaboration” among charity organizations that can lead to several issues, including inefficient resource allocation, limited reach and impact, lack of specialization, limited fundraising opportunities, limited communication and sharing of best practices, difficulties in evaluating the collective impact and losing opportunities to access government funding. Without effective collaboration, charity organizations may duplicate efforts and waste resources, as each organization may be unaware of the work done by others in the same field. They may also miss opportunities to reach a larger audience and make a more significant impact. Collaboration allows organizations to pool resources and knowledge to serve more people. Additionally, organizations that do not collaborate may be forced to spread themselves thin trying to address a wide range of issues; instead of focusing on their areas of expertise. Collaboration can open up new fundraising opportunities, as organizations can pool their resources and leverage the combined power of their networks to raise funds. [1]

Additionally, charity organizations can miss important information and best practices that could improve their work without collaboration. Collaboration can enable organizations to learn from one another, share successes and failures, and adapt their approaches accordingly. With the lack of collaboration, it can be challenging to evaluate the collective impact and measure the outcome of the community projects. The lack of collaboration between organizations can lead to a lack of data and statistics to measure the overall effect of their work.

This project is to develop “A Collaborative Platform for Community Support” to address the challenges NGOs face in providing community support. The project seeks to create a web-based platform for NGOs, charity organizations, association corporations, and CSP clubs to collaborate and share resources to

serve their communities more effectively. The platform is designed to facilitate information sharing, and resource management between NGOs and their target communities. Additionally, the platform aims to empower communities by providing them with access to resources and information, and opportunities for participation in community-building activities.

Many small charitable organizations operate in various cities in Pakistan but need more resources like funding, volunteers, and a solid online presence. In addition, none of these NGOs have a platform for collaboration and cooperation, which is a problem since, in the event of a disaster, many NGOs are helping the victims. Still, there are no means for them to connect with one another. As a result, some victims are unable to receive relief because every NGO only operates irregularly in different places, which is why some victims are unable to receive support. If all NGOs are connected and deliver aid to victims in the proper manner, then all victims experience relief quickly. Charity groups can efficiently participate, exchange ideas, and assist the community through this platform.

### **1.3 Project Objectives**

The main objective of this project is to address the challenges faced by NGOs/charity organizations in providing community support. The primary aim is to develop a web-based platform to facilitate collaboration between NGOs/charity organizations and volunteers that enables them to serve their communities more effectively. The platform is designed to facilitate communication, information sharing, and resource management between NGOs and the communities they do. The secondary aim is to empower small local communities by providing them with access to resources and information, as well as opportunities for participation in community-building activities. By achieving these objectives, the project will provide a comprehensive solution to the challenges faced by NGOs in offering community support and empowering communities by giving them access to resources and opportunities for participation in community-building activities. [2]

### **1.4 Methodology**

**Research and Planning:** In the initial stage, thorough research was conducted to understand the current challenges faced by NGOs in collaboration and

community support. The scope and requirements of the project were defined, and a project plan was created.

### **Design and Development:**

The web-based platform was created with a user-friendly interface, with features like login/registration, after any volunteer registers, a volunteer dashboard appears in front of him, and he can be able to see his:

- **Profile:** where they can update their profile,
- **Projects:** where they can view their projects,
- **Donations:** where they can view their donations,
- **Rewards:** where they can request their rewards like e-certificates etc., and also view CSP hours.

If any NGO wishes to register, the administrator must first approve the request before the NGO is permitted to log in to the system. The platform was developed using modern web development technologies such as HTML, CSS, JavaScript, and a library such as React.

### **Testing:**

The platform is thoroughly tested for functionality and usability to ensure it meets the requirements defined in the planning stage.

### **Deployment:**

Once the platform is seemed ready for launch, it is deployed to a web server and made available to the public.

### **Maintenance and Support:**

The platform was continuously monitored for performance and security, and any issues were promptly addressed. Ongoing support and maintenance were provided to ensure the platform remains functional and valuable for the NGOs, volunteers, and admin.



Figure 1.1: Overview

## 1.5 Project Scope

The project aims to develop a web-based platform that facilitates collaboration between NGOs and volunteers to better support local communities' needs. This project's scope includes the design, development, and deployment of a platform that allows NGOs to communicate and share resources with each other to serve the community better, and small non-profit organizations can quickly find volunteers for community work. After providing services to society, volunteers can also get incentives in the form of e-certificates and can launch charity campaigns in recognition of their contribution.

The platform enables NGOs to create and manage their own profiles, share information about their activities and events, and connect with other NGOs working in the same or related fields. Additionally, the platform includes features to enable NGOs to share resources such as funding opportunities, volunteer information, and other materials that can be useful for community support efforts. The project also includes a search feature to allow NGOs and community members to find and connect with organizations that meet their needs quickly. A user management system to ensure secure access to the platform and data. The project includes testing and deployment of the platform and comprehensive documentation of the software, data management, and user guides.

### 1.5.1 Limitations

Initially, the scope of this platform was limited to Rawalpindi/Islamabad-based small NGOs/charity organizations. In contrast, for volunteers, it was limited to Bahria University Students. However, with the help of marketing skills and digital campaigns, the project expanded to other universities and NGOs.

#### **Data quality:**

The platform relies on NGOs to provide accurate and up-to-date information about their activities and resources. If NGOs do not keep their profiles updated, the forum may not be as valuable for community members and other NGOs looking for information.

#### **User adoption:**

The platform's success depends on the willingness of NGOs to adopt and use the forum. If NGOs do not see the value in using the platform, it may not gain widespread adoption.

#### **Technology limitation:**

The project is based on web technologies and requires a stable internet connection for its total usage. With internet access, use would be unlimited.

#### **Maintenance and sustainability:**

The development and deployment of the platform is one-time project, but the platform need regular maintenance and support. This can be challenging in terms of funding and resources, and it might take much work to sustain the platform in the long run.

#### **Security:**

As the platform handles sensitive information of NGOs and community, it requires a robust security system to protect the data. Any security breach could severely damage the reputation of the platform and the participating NGOs.

### 1.5.2 Features

There were three primary sorts of users for this application, and each has unique capabilities based on their roles, which are described below:

#### 1. **Admin**

- Objectives/Motivation**

- Admin can Update the Homepage. Like Uploading/Deleting slider Images.

- Admin can Update About Us pages (Objective/Motivations).

- **Add/Remove/Update a Service category**

- Admin can Add a service category.
- Admin can Delete a service category.
- Admin can View a service category

- **Control User Features (User can be an NGO Or Volunteer)**

- Admin can Approve Profile of a User.
- Admin can Disable/Enable a User (This will not delete the User).
- Admin can Remove a User.
- Admin can View all Users.
- Admin can provide Incentives for the Volunteer (This includes the CSP Hours.)
  - \* After the Volunteer has Completed the project. He will get a button to request for CSP Hours. This will send a request to Admin and Admin will upload the certificate.
- Admin can provide badges (Gold, Silver, Bronze) (Will be shown on their profile) to the NGO based on the ratings they get by the Volunteer for the projects.

- **Control Project Features**

- Admin can Approve a Project Uploaded by an NGO.
  - Admin can Disable/Enable a Project (This will not delete the Project).
  - Admin can Add a Project. (Project can be uploaded with reference to an Admin. Like In NGO it gets uploaded with respect to NGO.).
  - Admin can Update a Project.
  - Admin can Remove a Project.
  - Admin can View all Projects of an NGO.
  - Admin can View all volunteers who are currently working on a project.
- Admin can view all projects that a specific volunteer enrolled in.

## II.      **NGO/Charity Organization**

- **Profile Management**

- NGO can create their profile on Join button and give following details (Registration Number(**optional**), NGO Name, Location (Address), Email, Password, Contact Details, Registration Certificate(**optional**)).
- After Join Approval from Admin, NGO can Update its profile with details (Registration Number, NGO Name, Location (Address), Email, Password, Contact Details, and Registration Certificate).
- NGOs can view their Profile

- **Banking Management**

- After profile creation, NGOs can Add their Banking Details.
- NGOs can Update their Banking Details
- These banking details will show to guests or volunteers on the donation button

- **NGOs can control project features.**

- NGO can Upload new Projects. (Admin will approve the project, then it will be shown to volunteers and public pages).
  - \* Project Name, Location, Description, Start Date, End Date, Service Category (**Added by Admin**), Project Cover (Cover photo of Project).
- NGOs can view their Uploaded Projects.
- NGO can change status of A project (**On Going, Completed, Deactivated**)
  - \* Deactivate will not **delete** the project from Database. Deactivated projects will be in DB and shown to NGO only but won't be public. Deletion can be done by Admin.

- **Manage Volunteers**

- View Volunteers Only those Volunteers who enrolled for the project uploaded by that NGO will be shown here.
- Issue an e-certificate to a volunteer who volunteered for a project.
  - \* After the Volunteer has Completed the project. He will get a button to request for Certificate. This will send a request to NGO and NGO will upload the certificate.

- **Support**

- Chat with Admin and Volunteer.

### III.      **Volunteer**

- **Create profile**
  - Volunteers can create their profile on Join button and give following details (Volunteer Name, Email, Phone(optional), CNIC, Institution(optional), Enrollment(optional), Address).
- **Update profile.**
  - After Approval from Admin, NGO can Update their profile with details (Volunteer Name, Email, Phone, CNIC, Address, Profile Picture).
- **View All project.**
  - This shows all active projects.
- **Volunteer for a project.**
  - Volunteers can enroll in a project.
- **Request for Incentives.**
  - After completing the enrolled project, the Volunteer can request for an e-certificate from the NGO.
  - The Volunteer can also request for CSP Points from Admin. (Based on the project)
- **Query for Info (Using support).**
  - Can chat to Admin/Volunteer or see FAQs.
- **Can Do Donations.**
  - This will show the donations page of all the NGOs.
- **Support**
  - Chat with Admin

# **Chapter 2**

## **Literature Review**

The literature review aims to explore existing community service systems and evaluate their effectiveness in facilitating collaboration among NGOs and volunteers. The provision of community services is a vital aspect of society, with various NGOs and charity organizations working towards improving the lives of individuals and communities. However, the coordination and collaboration among these organizations can often be challenging, leading to inefficiencies and a lack of impact. This is a problem because in the event of a disaster, there were numerous NGOs aiding victims, but there was no way for them to communicate with one another. As a result, some victims are unable to receive assistance, because every NGO works randomly on various locations and some victims may fail to receive aid. All victims receive relief if all NGOs are connected and appropriately providing assistance to victims. Charity organizations can also share ideas, and effectively contribute and help the community. The proposed platform aims to provide an online community support portal for NGOs and charity organizations to collaborate effectively and provide aid to those in need, particularly in the event of a disaster.

### **2.1 Community Support Systems in Pakistan**

#### **2.1.1 The Citizens Foundation (TCF)**

The Citizens Foundation (TCF) is a non-profit organization in Pakistan that aims to provide quality education to underprivileged children. TCF operates a network of over 1,500 schools across the country, serving over 250,000 students. The organization also provides teacher training and community engagement programs to support its mission. [3] Figure Below shows the About Page of TCF:



Figure 2.1: About Page Of TCF

### 2.1.2 Edhi Foundation

The Edhi Foundation is a non-profit organization in Pakistan that provides a wide range of community services, including emergency services, orphanages, and rehabilitation centers. The organization operates a fleet of ambulances and emergency response teams, as well as various welfare centers across the country. [4]

Figure Below shows the Homepage of Edhi Foundation:



Figure 2.2: Homepage of Edhi

### 2.1.3 Pakistan Poverty Alleviation Fund (PPAF)

Pakistan Poverty Alleviation Fund (PPAF) is a non-profit organization that aims to alleviate poverty in Pakistan through community-based development programs. PPAF works with a network of local organizations to implement its programs, which focus on areas such as education, health, and livelihoods. [5]

Figure Below shows the About Page of PPAF:



Figure 2.3: About Page of PPAF

### 2.1.4 Shaukat Khanum Memorial Cancer Hospital and Research Centre

Shaukat Khanum Memorial Cancer Hospital and Research Centre is a non-profit organization in Pakistan that provides cancer treatment and research. The organization operates two cancer hospitals in Lahore and Peshawar and also conducts research to improve the treatment of cancer in Pakistan. [6]

Figure Below shows the Homepage of SKMT:



Figure 2.4: Homepage of SKMT

### 2.1.5 Alkhidmat Foundation

Alkhidmat Foundation is a non-profit organization in Pakistan that provides a wide range of community services, including emergency services, education, health, and livelihoods. The organization operates a fleet of ambulances and emergency response teams, as well as various welfare centers across the country. Alkhidmat Foundation also runs a number of schools, hospitals, and other welfare centers and projects. [7]

Figure Below shows the Introduction Page of Alkhidmat Foundation:

Alkhidmat Foundation Pakistan is one of the leading, non-profit organization, fully dedicated to humanitarian services since 1990. Alkhidmat's workers and volunteers continue to work tirelessly for the relief of affected people across Pakistan and worldwide. Our dedicated services include disaster management, health services, education, orphan care, clean water, Mawakhat (interest-free loan) and other community services.

**Our Vision**

*"Service to Humanity with Integrity"*

**Our Mission**

Figure 2.5: About Page of Alkhidmat

## 2.2 Community Support systems Overseas

### 2.2.1 Charity Navigator

Charity Navigator helps donors make informed decisions about which charities to support. It provides information on the financial health, accountability, and transparency of over 8,000 charities. The platform uses a rating system to evaluate charities based on financial performance, accountability, and transparency. This system allows donors to easily compare charities and make informed decisions about where to direct their donations. [8]

Figure Below shows the About Page of Charity Navigator:

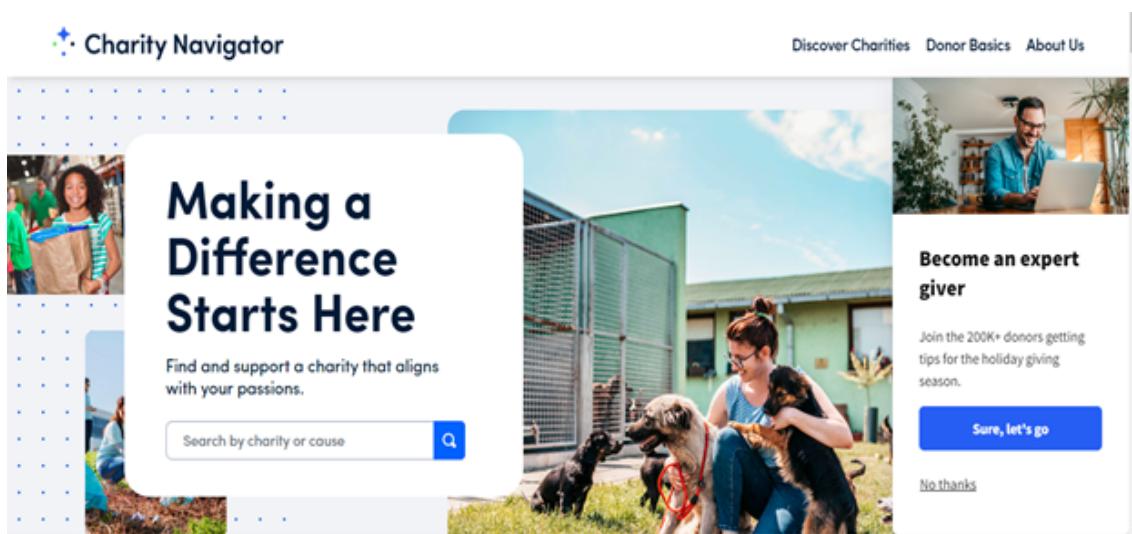


Figure 2.6: Homepage of Charity Navigator

### 2.2.2 Givewell

Givewell is a nonprofit organization that evaluates charities based on their effectiveness in improving the lives of people in need. The organization conducts in-depth research on various charities and provides information on their impact, cost-effectiveness, and overall sustainability. Givewell also publishes a list of top-rated charities based on their research, which can be used by donors to make informed decisions about which charities to support. [9]

Figure Below shows the About Page of Givewell:

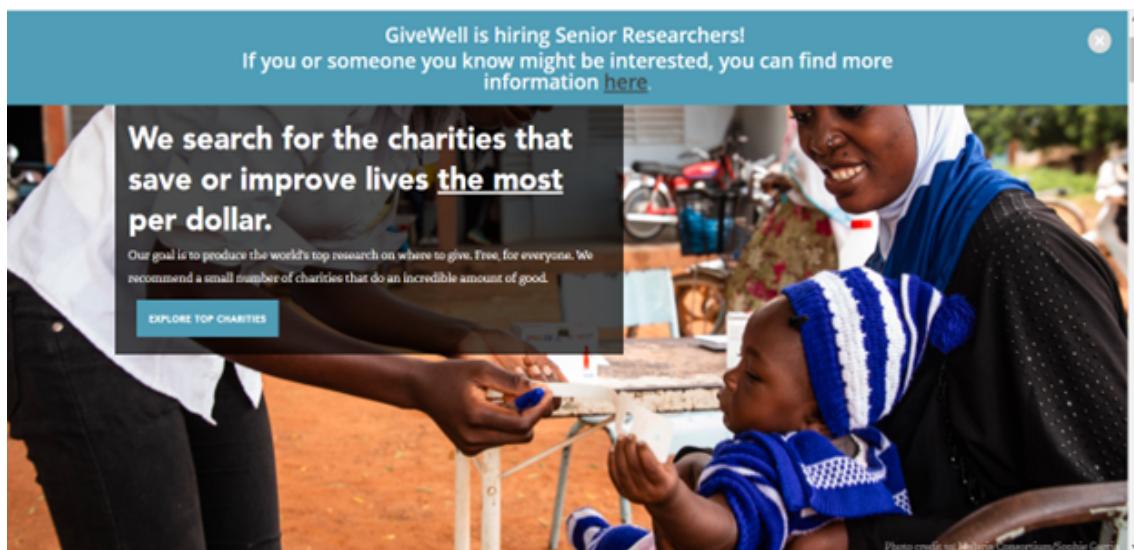


Figure 2.7: About Page of Givewell

### 2.2.3 GlobalGiving

GlobalGiving is a crowdfunding platform for nonprofits, connecting individual donors with grassroots projects around the world. The platform allows donors to search for and support specific projects and provides information on the nonprofit organizations behind the projects. GlobalGiving also includes training and support to nonprofits to help them run successful crowdfunding campaigns. [10]

Figure Below shows the About Page of Global Giving:

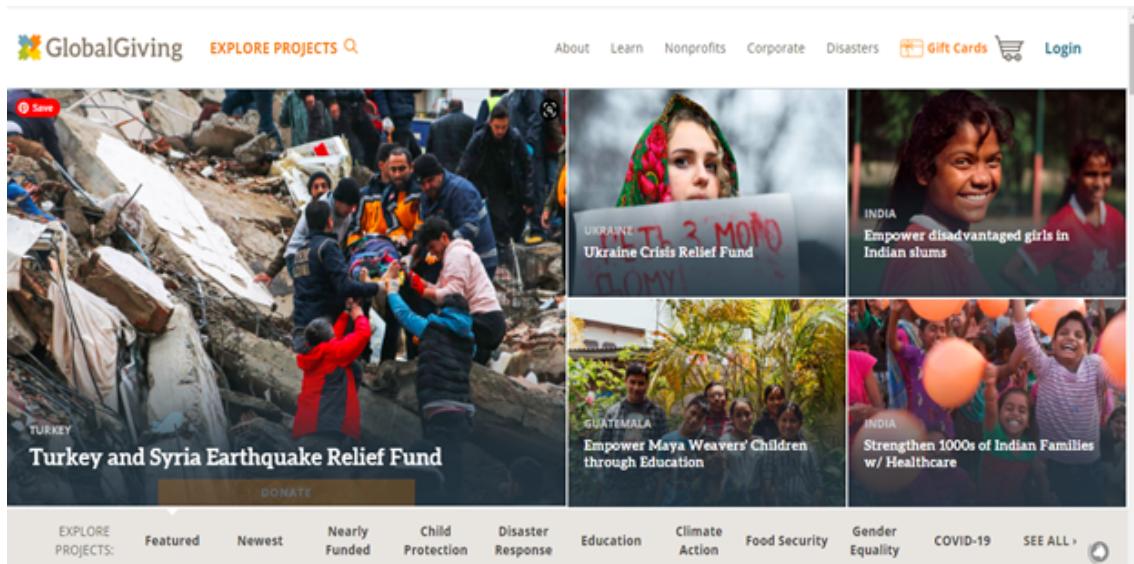


Figure 2.8: Homepage of Global Giving

### 2.3 Comparison with the Community Support Platform

While existing systems such as The Citizens Foundation, Edhi Foundation, Pakistan Poverty Alleviation Fund and Shaukat Khanum Memorial Cancer Hospital and Research Centre and Alkhidmat Foundation provide provides a wide range of essential community services in Pakistan Additionally, these existing organizations also run a number of schools, hospitals, and other welfare centers and projects, but they do not have a feature to track impact, measure the success of volunteer efforts and provide incentives for volunteer involvement like Community Support Platform does. Similarly, Overseas Existing Systems Like the Charity Navigator, Givewell, and GlobalGiving provide information and resources to help donors make informed decisions about which charities to support but they need to focus on facilitating collaboration among NGOs and volunteers for community services. Community Support Platform aims to fill this gap in collaboration and volunteer involvement in the community services sector in Pakistan and provide a space for NGOs to connect, share ideas, and coordinate efforts for the betterment of community services in Pakistan. Additionally, the Community Support Portal aims to provide incentives for volunteer involvement, such as service points and e-certificates, which is absent in any of the above-mentioned systems. Therefore, the proposed platform is unique in its approach and was beneficial for both NGOs and volunteers.

### 2.4 Conclusion

The literature review has highlighted the need for a platform to facilitate collaboration among NGOs and volunteers for community services. The study has highlighted various existing systems that provide essential community services in Pakistan, but each of them are in their own pace and no collaboration is among them. The proposed platform for the final year project aims to fill this gap by providing a centralized platform for NGOs and volunteers to connect, collaborate, and coordinate their efforts. Additionally, the proposed platform would also provide a way for volunteers to earn service points and incentives in recognition of their contributions.

### 2.5 Comparison Table

A comparison table that compares the characteristics offered by the above-mentioned Community Service Platforms.

Table 2.1: Comparison Table between the features of some Community Service Platforms

Characteristics	Community Support Program	The Citizen Foundation	Edhi Foundation	Pakistan Poverty Alliviation Fund	Shaukat Khanam	Alkhidmat Foundation
Colaboration Between NGOs and Local Volunteers.	√					
Involvement of Volunteers	√	√	√	√	√	√
Incentives for Volunteer Involvement	√					
Connecting Volunteers with non-profit organization	√					
Utilization Of Technology and open data for disaster response effort	√					√
Funding Projects				√	√	√
Providing Health Services	√		√		√	√
Providing Education	√	√				√

# **Chapter 3**

## **Requirement Specifications**

Requirement specifications outlines the specific requirements for a project. This includes the functional and non-functional requirements for the project, the features, and capabilities that the platform should have, as well as any constraints or limitations.

### **3.1 Existing systems**

The project aims to develop a platform to facilitate community support. There are already several existing systems that have been designed to address similar challenges, and it is essential to understand these existing systems to identify the areas where they are lagging and where the proposed platform can improve upon.

Many websites and portals exist that provide information about community resources and support services. These portals typically include a directory of services, as well as information about their mission, goals, and activities. However, most of these portals could be more user-friendly and it can be difficult for community members to find the information they need.

Many existing communities support platforms need to be designed to facilitate collaboration between community members, organizations, and service providers. They may provide information about available resources, but they do not have tools to allow community members to work together to address specific challenges. Therefore, the primary issue with the existing systems is need for collaboration.

Moreover, existing community support platforms need to have effective communication channels for community members, organizations, and service providers to connect and share information. They may have a directory of resources, but they need to provide tools for community members to communicate with each other or with service providers.

### **3.1.1 Limitations**

The current systems might be limited in several ways, for instance:

- **Collaboration and communication:**

Many community support systems lack efficient communication channels that would help community members and organizations collaborate and communicate. They do offer information on resources that are accessible, but they lack the tools that would enable members of the community to collaborate to solve problems or to connect and share information effectively.

- **Limited Support:**

Existing systems may need more support, such as customer service or technical support, which can make it difficult for community members to get help when they need it.

- **Social Presence:**

Many small existing community support platforms need a significant social presence, which can make it difficult for community members to discover the platform and for the platform to reach a broad audience. They may need a significant following on social media, a lack of reviews and testimonials and limited web presence in terms of SEO and web traffic.

- **Limited Reach:**

Small community support platforms may only serve a specific geographical area or target a particular community, limiting their reach and accessibility to a broader audience. This can make it difficult for community members outside of the specific area or neighborhood to find and use the platform.

- **Limited Engagement:**

Small community support platforms may need more engagement or active users, which can make it difficult for community members to find and connect with other members, organizations, and service providers.

### **3.1.2 Drawbacks**

Many existing systems need to be designed to facilitate collaboration between NGOs and other organizations. They do not have mechanism to allow NGOs

to work together to address specific challenges. This can lead to a lack of coordination and inefficiency in providing aid to the community.

Many existing systems need effective communication channels for NGOs to connect and share information. They do not have system for NGOs to communicate with each other or with other organizations. This can lead to a lack of information sharing and duplication of efforts.

Existing systems may need proper transparency in place to ensure that the aid is reaching the right people and that the aid distribution process is fair.

## 3.2 Proposed System

The project aims to develop a platform to facilitate community support, specifically focusing on improving collaboration, communication and social presence. The proposed system would include the following features:

The platform would provide tools for community members, NGOs, and service providers to work together to address specific challenges. This could include a forum for community members to discuss challenges, a project management system for coordinating efforts, and a calendar for scheduling events and meetings. It could also include functionalities like creating and joining projects, creating and updating profiles, launch own campaign only for eligible volunteers. [11]

The platform would provide effective communication channels for community members and NGOs to connect and share information. This could include a chat function for direct communication, a bulletin board for posting announcements from admin and NGOs, and a directory of resources. It could also have functionalities like sending messages and sending notifications.

The platform would have a solid social presence, which would help to increase its reach and engagement with the community. This could include a strong presence on social media, a blog, and a review/testimonial system to help increase the platform's visibility and credibility. It could also include functionalities like social media integration and a referral system.

## 3.3 Functional Requirements

Functional Requirements are the specific features and capabilities that a system should have to meet the needs of its users. Some of the functional requirements for this project could include the following:

### **3.3.1 User Registrations and Login**

Users should be able to create an account and log in to the platform to access its features and resources. This may include options for different types of users, such as community members (Volunteers), NGOs, and admin.

### **3.3.2 Service Category Management**

The admin should have the ability to update, add, or remove a service category on the platform. This could include the ability to create new categories, edit existing categories, and delete categories that are no longer needed.

### **3.3.3 User and Project Management**

The admin should have the ability to add, remove, edit, and view users and projects on the platform. This could include the ability to create new projects, delete user accounts, and view a list of all users on the platform.

### **3.3.4 Community Service Hours**

NGOs should be able to assign community service hours to a volunteer and track the volunteer's progress towards completing their community service hours.

### **3.3.5 E-certificate Issuance**

NGOs should have the ability to issue an e-certificate to a volunteer upon completion of the required community service hours. This could include options for customizing the certificate, such as adding the NGO's logo, and the volunteer's name and details.

### **3.3.6 Collaboration Tools**

The platform offers resources that enable service providers, NGOs, and community members to collaborate on specific problems. This could contain a discussion board for issues facing the community, a method for coordinating efforts, and a calendar for gatherings and activities.

### **3.3.7 Communication tools**

The platform ought to offer efficient ways for service providers, NGOs, and community members to connect and share information. This incorporates a chat feature for one-on-one discussion, a notice board for posting updates, and a resource directory.

### 3.3.8 Support options

The platform should provide adequate support, such as customer service or technical support.

## 3.4 Non-Functional Requirements

Non-Functional Requirements are the specific characteristics and constraints of a system that are not related to the particular features or capabilities of the system but still must be met for the system to be successful. Some examples of non-functional requirements for this project could include the following:

### 3.4.1 Performance

The platform should be able to handle many users' and NGO requests and should have fast response times. This could include requirements for system scalability, and response time under peak loads.

### 3.4.2 Security

The platform should have robust security measures in place to protect users, and NGO data, such as encryption, secure login, and monitoring of user activity.

### 3.4.3 Usability

The platform should have a user-friendly interface and should be accessible to users with disabilities.

### 3.4.4 Compatibility

The platform should be compatible with different browsers and mobile devices and should work well on different devices and platforms.

### 3.4.5 Maintenance

The platform should be easy to maintain and should have an update and bug-fixing schedule in place.

### 3.4.6 Scalability

The platform should be able to handle a large number of users and should be able to scale up or down as needed.

### **3.4.7 High availability**

The primary objective of designing the platform is to reduce the occurrence of periods when the system is unavailable, commonly known as downtime, and to guarantee a consistently high level of accessibility. This is achieved by implementing a comprehensive disaster recovery plan and establishing a resilient infrastructure. In practical terms, it entails building a platform that can effectively handle unforeseen failures or disruptions and has the capability to swiftly recover from them. By minimizing downtime, we aim to ensure that the platform remains operational for extended durations.

### **3.4.8 Reliability**

The ultimate goal of the system is to execute the assigned tasks consistently and flawlessly, without encountering any failures or errors. This means that whenever a task is requested, the system should reliably and dependably carry it out, ensuring the desired outcomes are achieved. To ensure such consistent performance, various factors come into play. Firstly, the system needs to be designed with robustness in mind. This involves building a solid foundation and implementing reliable mechanisms that enable the system to handle the assigned tasks effectively. It entails incorporating well-tested algorithms, error-handling routines, and efficient data structures to ensure the system's stability and resilience.

### **3.4.9 Data management and security features**

The platform must prioritize the implementation of robust security measures to safeguard users' data and ensure their privacy and confidentiality. This entails adopting multiple layers of protection, such as encryption and secure login mechanisms, to fortify the platform's defense against potential threats and unauthorized access attempts. Encryption plays a fundamental role in preserving the integrity and confidentiality of sensitive information

## **3.5 Use Cases**

A use case is a visual representation of the interactions between actors and use case in a system and how a system was used to achieve a specific goal or set of plans. It describes the steps that a user or system takes to complete a specific task or accomplish a particular objective.

### 3.5.1 Login/Register Use case

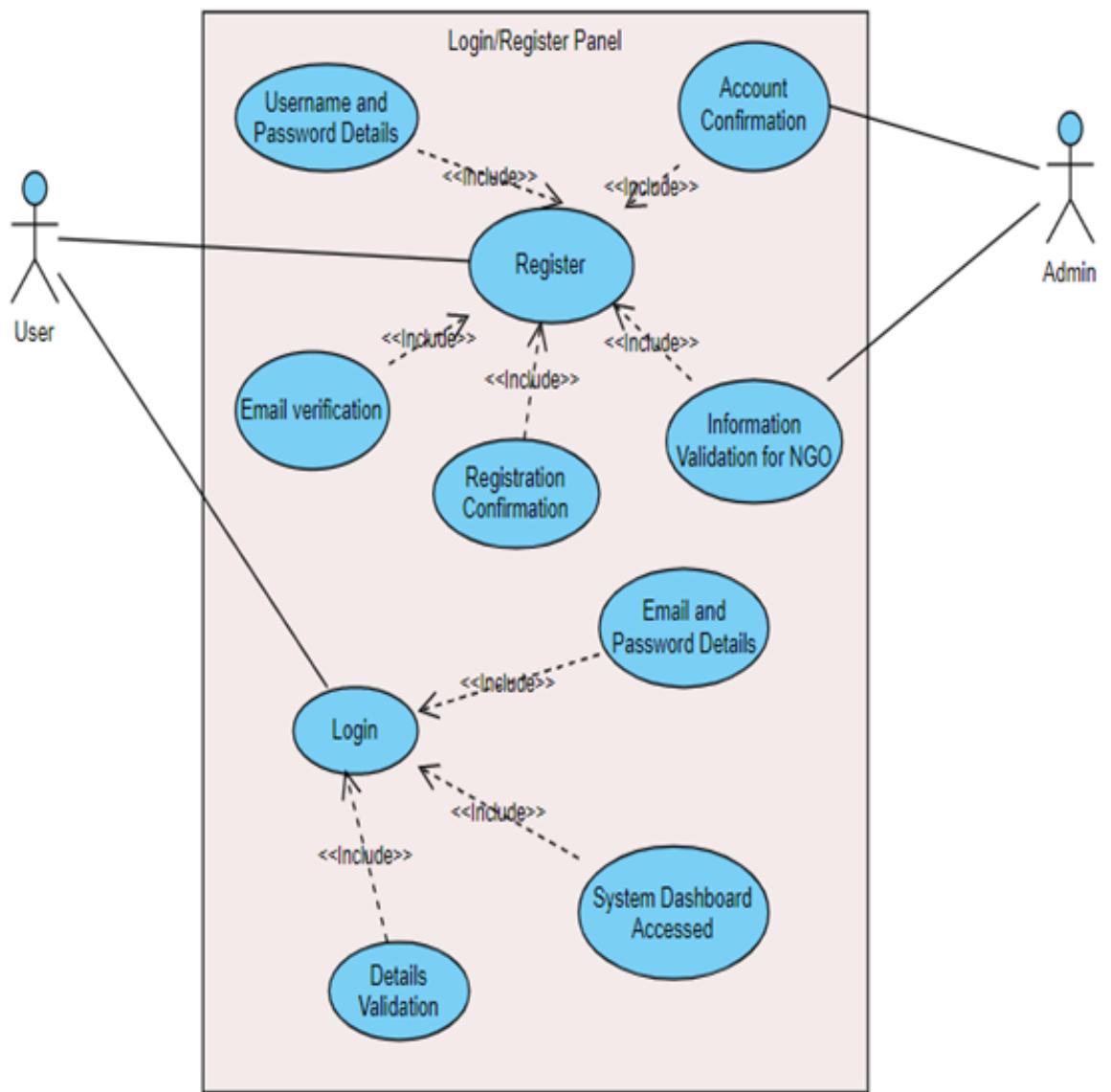


Figure 3.1: Login/Register Use case

### 3.5.2 Specific Volunteer Use Case:

A use case for a specific volunteer on this project could be:

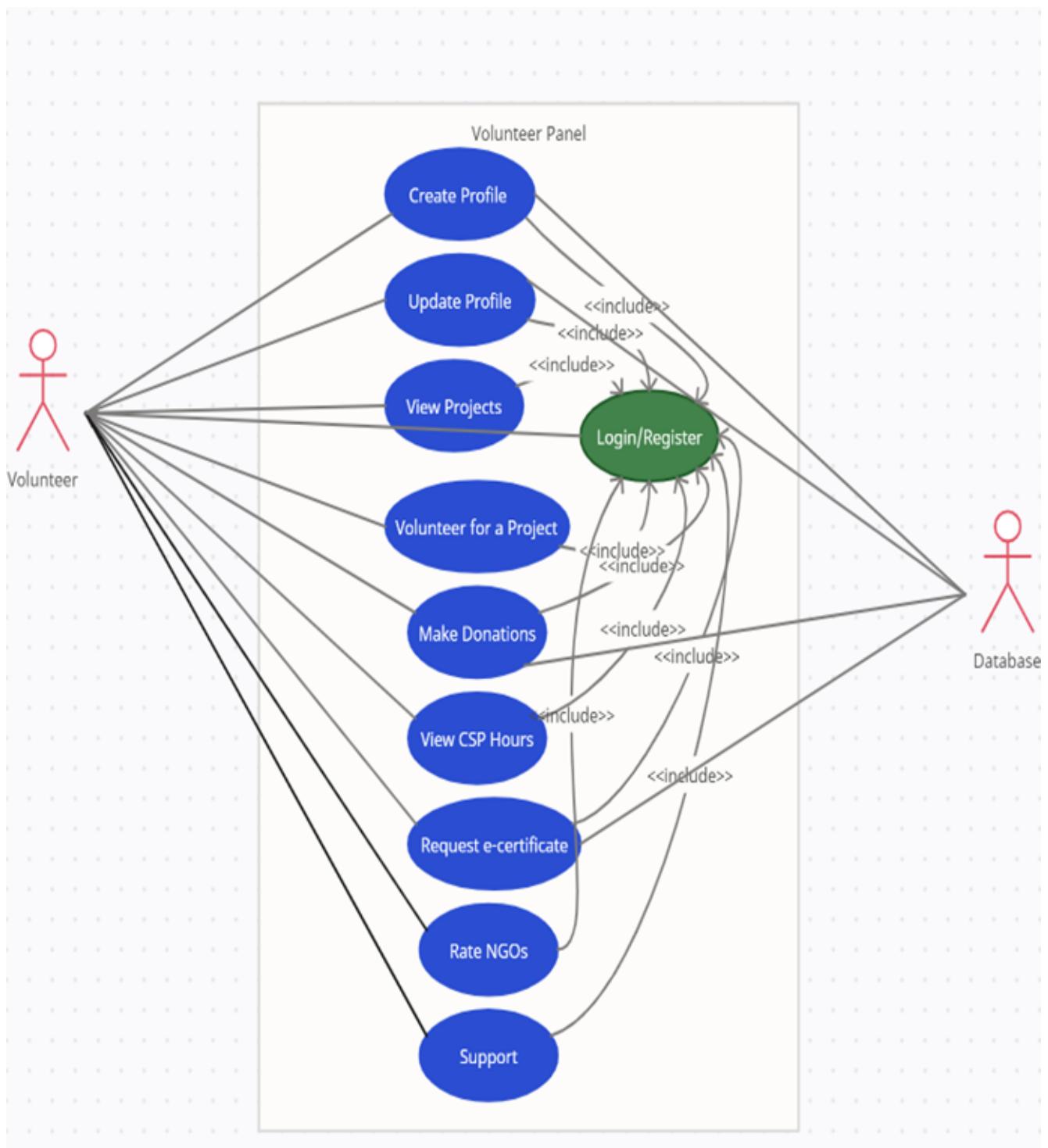


Figure 3.2: Specific Volunteer Use Case

### 3.5.3 Specific NGO Use Case:

A use case for a specific NGO on this project could be:

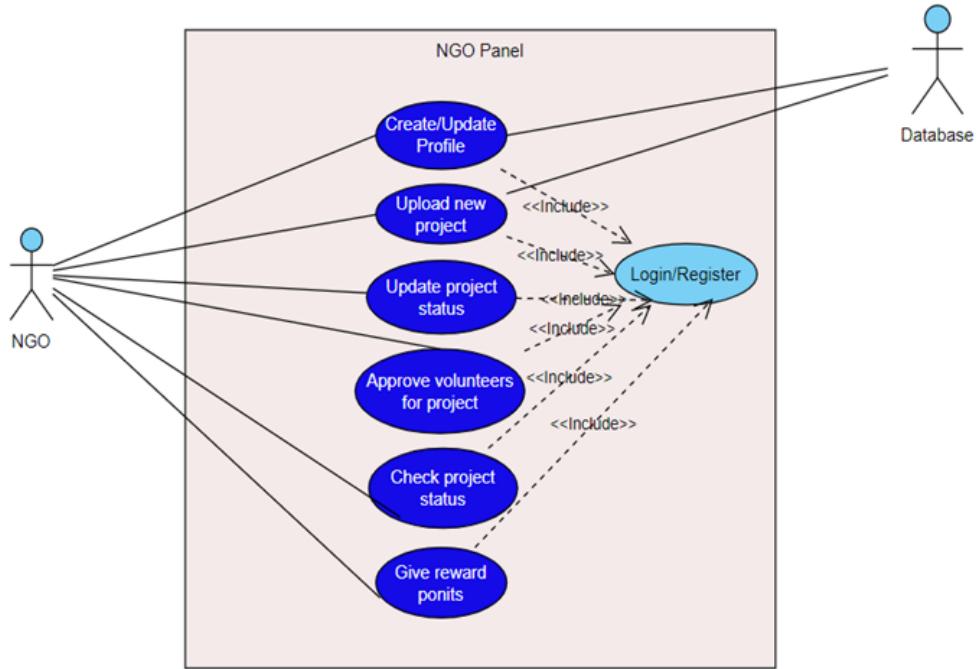


Figure 3.3: Specific NGO Use Case

### 3.5.4 Specific Admin Use Case:

A use case for a specific Admin on this project could be:

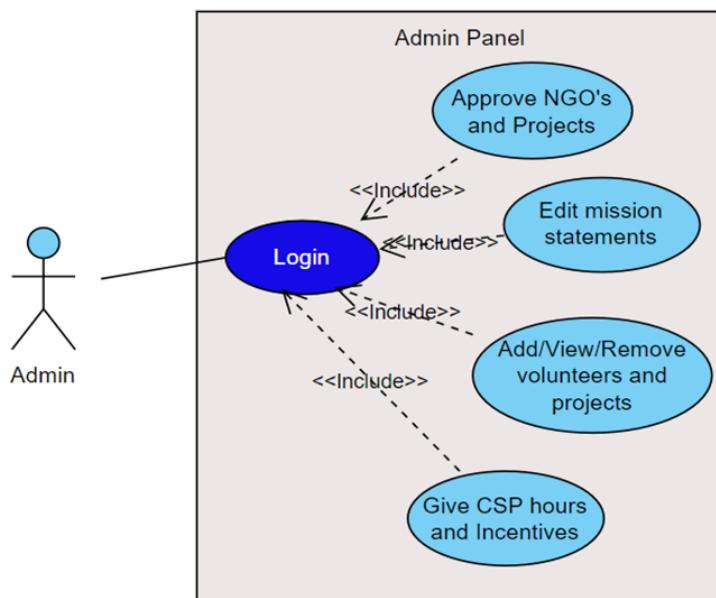


Figure 3.4: Specific Admin Use Case

## 3.6 Tabular Descriptions of Use Cases

This section provides descriptive details of the use case in tabular form.

### 3.6.1 Use Tabular Case Description for Login:

A use case tabular description for login on this project could be:

<b>Use Case Id</b>	UC-01		
<b>Use Case Name</b>	Login		
<b>Actor(s)</b>	Admin, NGO, Volunteer		
<b>Data</b>	Email, Password		
<b>Trigger/Stimulus</b>	The actor opens the portal.		
<b>Pre-condition</b>	The user should already be registered / signed up.		
<b>Assumption</b>	User has network access to the system.		
<b>Description/Steps</b>	<b>Normal Flow of Events</b> Actor enters email and password (credentials) in the fields provided and clicks the Login button. Credentials was verified by comparing with credentials already stored in database. Upon successful matching, user was redirected to the next page and a successful login message was displayed.	<b>Alternate flow of events</b> If credentials do not match, error message was displayed. If email does not exist, actor was shown a message asking them to register with the system.	
<b>Main Success Scenario</b>	Login is successful (Email and password match).		
<b>Post Conditions</b>	Actor has access to the system, successfully logged in page is displayed.		
<b>Comments</b>	—		

Table 3.1: Tabular Description for Login

### 3.6.2 Use Case Tabular Description for Requesting E-Certificate

A use case tabular description for “Requesting E-Certificate” on this project could be:

<b>Use Case Id</b>	UC-02	
<b>Use Case Name</b>	Request e-certificate	
<b>Actor(s)</b>	Volunteer	
<b>Data</b>	Email, Password	
<b>Trigger/Stimulus</b>	Actor opens portal.	
<b>Pre-condition</b>	Volunteer must have to complete required CSP hours or points.	
<b>Assumption</b>	User has network access to the system.	
<b>Description/Steps</b>	<b>Normal Flow of Events</b> Actor enters email and password (credentials) in the fields provided and clicks the Login button, to log into system then click on rewards button and then Redeem his certificate.	<b>Alternate flow of events</b> Volunteer click on the “Rewards” button and there is not “Redeem Certificate” button is showing.
<b>Main Success Scenario</b>	Successfully redeem the certificate in first attempt.	
<b>Post Conditions</b>	Actor has access to the system..	
<b>Comments</b>	—	

Table 3.2: Tabular Description for Requesting E-Certificates

### 3.6.3 Use Case Tabular Description for Giving Reward Points

A use case tabular description for “Giving Reward Points” on this project could be:

<b>Use Case Id</b>	UC-03	
<b>Use Case Name</b>	Give Rewards Points	
<b>Actor(s)</b>	NGO	
<b>Data</b>	Email, password, must be registered NGO.	
<b>Trigger/Stimulus</b>	Actor opens his portal.	
<b>Pre-condition</b>	Volunteer must have to complete required points.	
<b>Assumption</b>	User has network access to the system.	
<b>Description/Steps</b>	<b>Normal Flow of Events</b> Actor enters email and password (credentials) in the fields provided and clicks the Login button, to log into system then check the points of volunteers, those who completed the required points (based on services), issue certificates to those volunteers.	<b>Alternate flow of events</b> 1. When NGO click on the reward button, few volunteers doesn't complete the required points, so they don't issue certificate to them. 2. When NGO click on reward button, no volunteer record is showing.
<b>Main Success Scenario</b>	Successfully issue certificates to volunteers in first attempt.	
<b>Post Conditions</b>	Actor has access to the system..	
<b>Comments</b>	—	

Table 3.3: Tabular Description for Giving Reward Points

### 3.6.4 Use Case Tabular Description for Approve NGO Requests

A use case tabular description for “Approve Requests of NGOs” on this project could be:

<b>Use Case Id</b>	UC-04	
<b>Use Case Name</b>	Approve Requests of NGOs	
<b>Actor(s)</b>	Admin	
<b>Data</b>	Email, password, Registration certificate.	
<b>Trigger/Stimulus</b>	Actor opens his portal.	
<b>Pre-condition</b>	NGO must have to registered from “Social Welfare Department” - Gov. of Punjab.	
<b>Assumption</b>	User has network access to the system.	
<b>Description/Steps</b>	<b>Normal Flow of Events</b> Actor enters email and password (credentials) in the fields provided and clicks the Login button, to logged into system then check the dashboard for request or click on “Manage Requests” section and then verify the NGOs registration certificate, if NGO is genuinely registered from “Social welfare Department of Pakistan”, then we approve the request of NGO.	<b>Alternate flow of events</b> 1. When admin click on the “Manage Requests” button, then check the NGOs details and if NGO registration certificate is fake, then reject the NGO request. 2. When admin click on the “Manage Requests” section, there is no requests available.
<b>Main Success Scenario</b>	Successfully verified the NGOs requests based on genuine registration certificate.	
<b>Post Conditions</b>	Actor has access to the system..	
<b>Comments</b>	—	

Table 3.4: Tabular Description for Approving NGO Requests

# **Chapter 4**

## **Design**

This chapter is going to elaborate the architecture and design of the system. There are three basic modules in the system which are given below:

- Admin
- NGOs
- Volunteers

### **4.1 System Architecture**

System architecture represents the system application as it's a high-level logical representation of the system. It shows the components and their relationship. In figure 4.1 the system architecture of our application is shown. System architecture includes:

- Presentation Layer
- Business Layer
- Database Layer

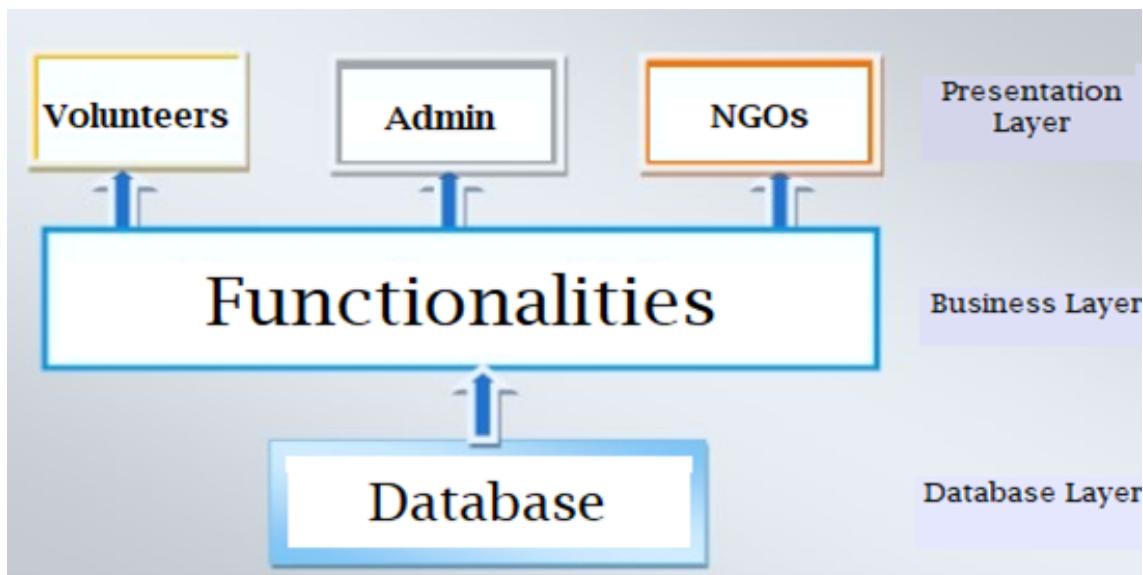


Figure 4.1: System Architecture

## 4.2 Design Constraints

### 4.2.1 Programming Language:

The primary programming language used in this project are MERN Stack. It has 4 components i.e., MongoDB, Express.JS, React.JS and Node.js. React.JS includes the HTML5, CSS and JavaScript. Another main library is Material UI which help add new feature. Node.js and Express.Js are being used for Backend and to handle the Database, Mongo DB is being used.

### 4.2.2 Design Standard

The software project must conform to the UML standards and must be documented at every step. An object-oriented approach should be adopted for the development of the application for reusability and to reduce complexity. Three-tier architectures was of vital importance to the correct implementation of the system and therefore require further analysis.

### 4.2.3 Assumptions and Dependencies

- It is independent software, and the system is robust and is available to user 24/7. The source code for this software is not shared, and the database is updated with the last information.
- All the software required are open source.

- The Internet should always be accessible and should be able to respond to many requests.
- Users must have sufficient knowledge of Web-Based applications.

## 4.3 Sequence Diagrams

Sequence diagram shows the interaction between the objects of the system application. The tools used for Sequence Diagram is listed below:

- **LucidChart:** <https://www.lucidchart.com/pages/>  
It is an online tool used to draw Sequence Diagrams.
- **Edraw-Max:** <https://www.edrawsoft.com>  
It is a Software helpful in making Sequence Diagrams

### 4.3.1 Overall System

The Sequence Diagram below shows the working of overall system.

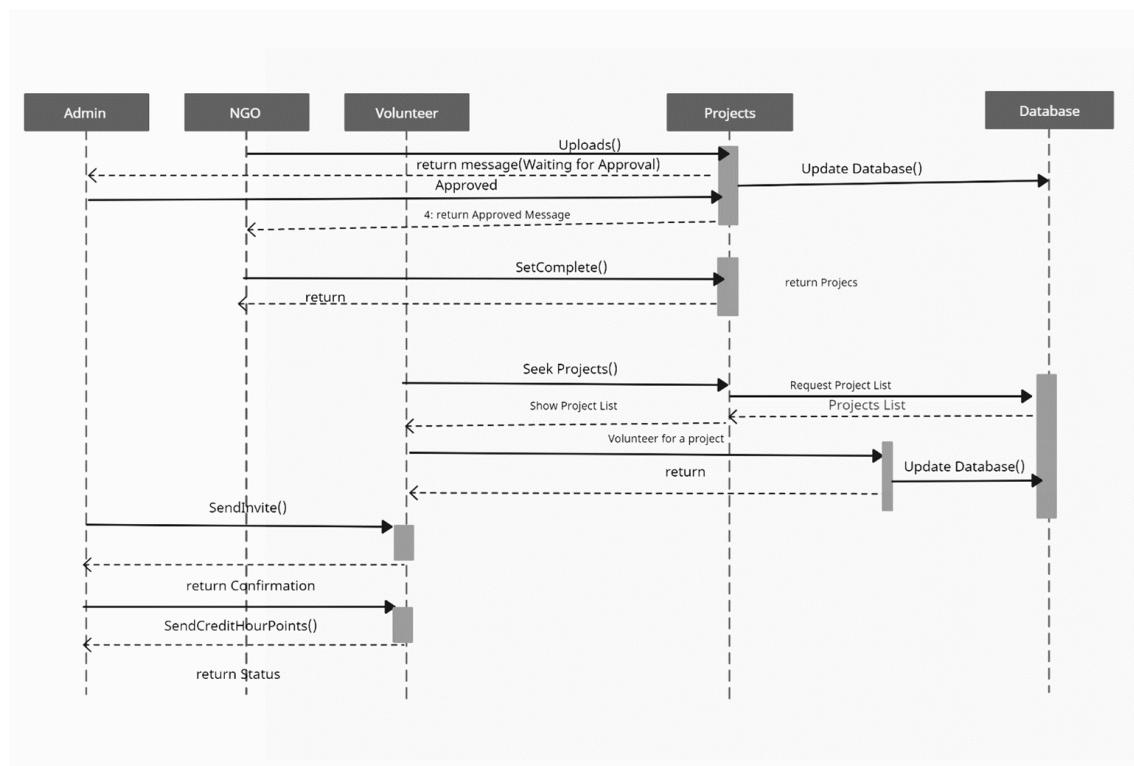


Figure 4.2: Sequence Diagram for Overall System

### 4.3.2 Login/Register

Sequence Diagram below shows the working of Login/Register.

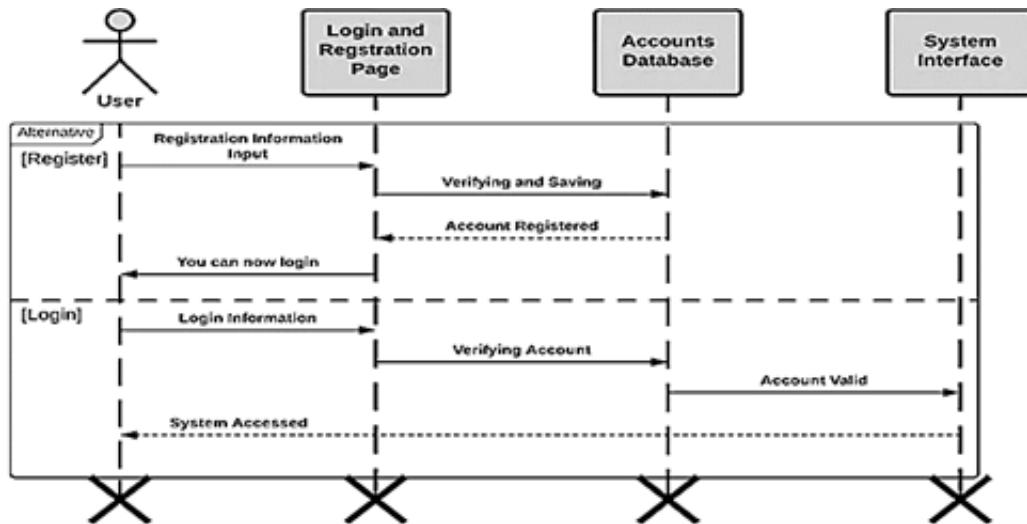


Figure 4.3: Sequence Diagram for Login/Register

### 4.3.3 Project Sharing

Sequence Diagram below shows the working of Project Sharing.

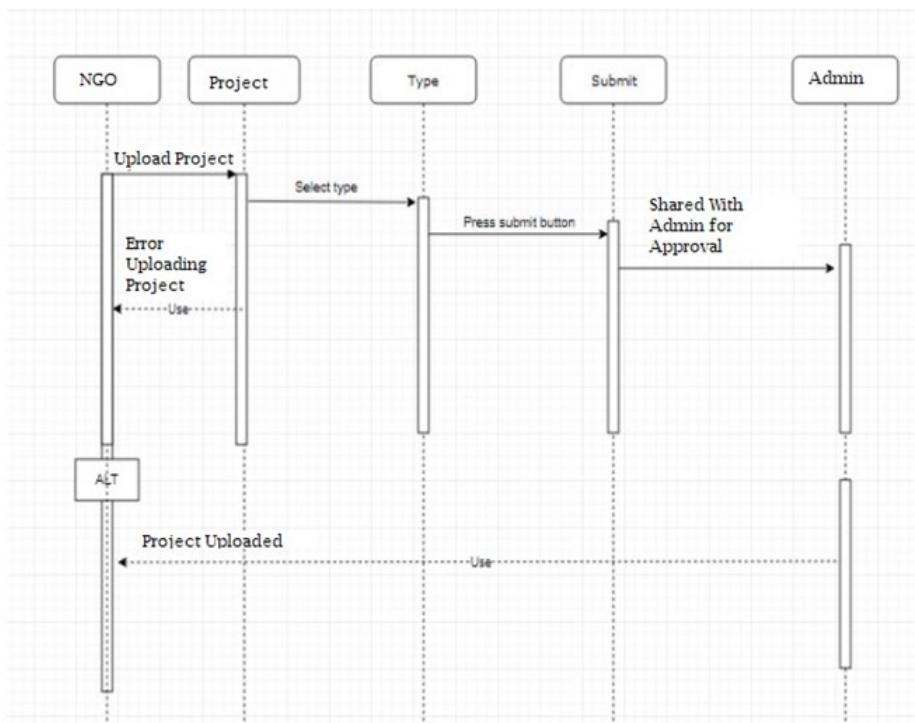


Figure 4.4: Sequence Diagram for Project Sharing

#### 4.3.4 Project Approval

Sequence Diagram below shows the working of Project Approval.

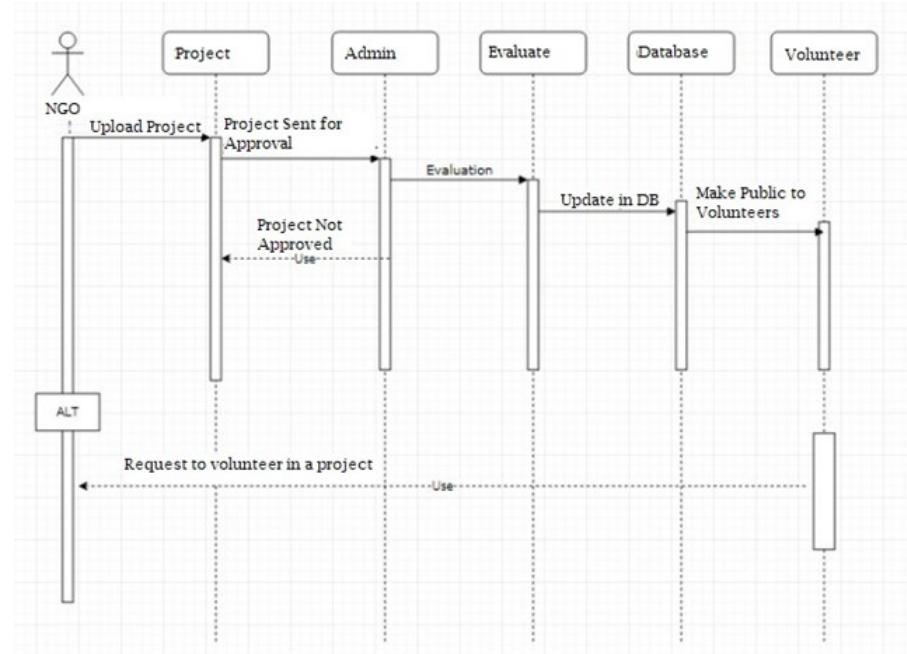


Figure 4.5: Sequence Diagram for Project Approval

#### 4.3.5 Assigning Projects

Sequence Diagram below shows the working of Assigning Projects.

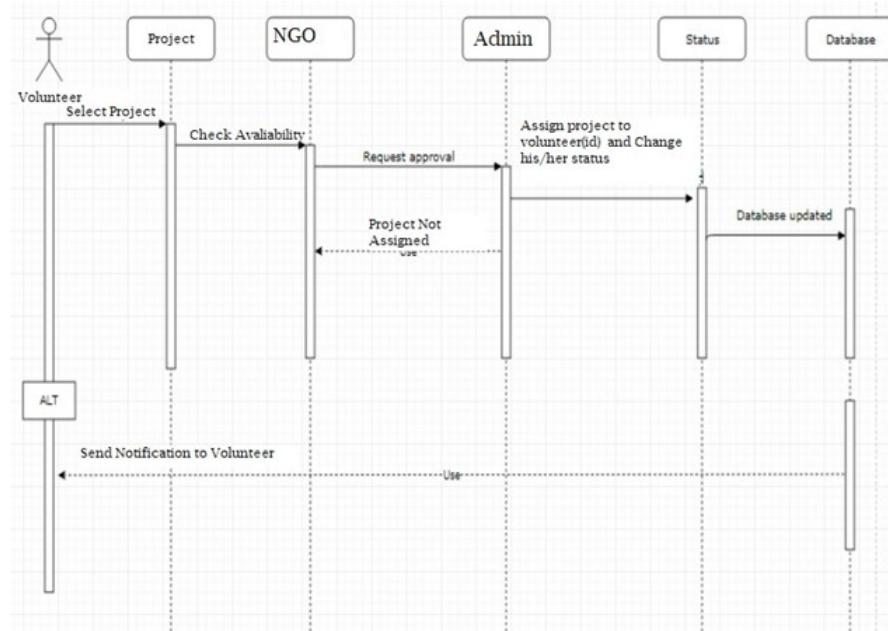


Figure 4.6: Sequence Diagram for Assigning Project

#### 4.3.6 Search

Sequence Diagram below shows the working of Searching Feature.

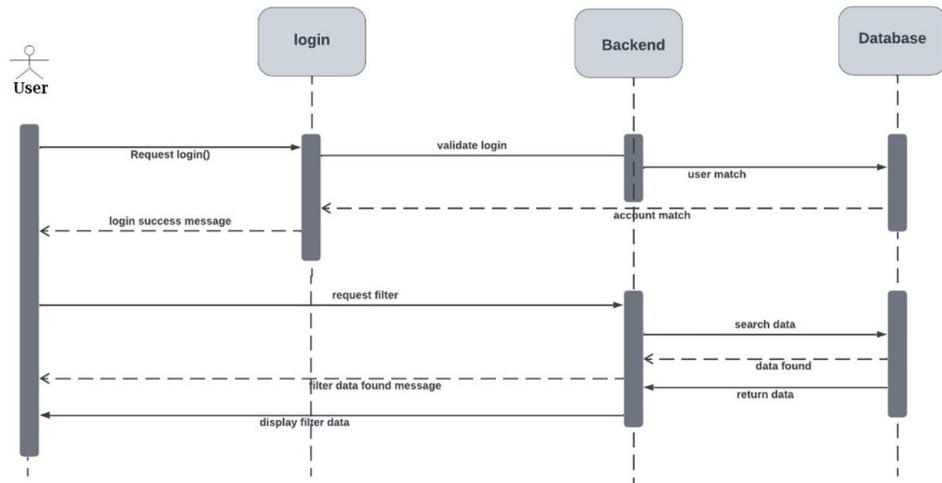


Figure 4.7: Sequence Diagram for Search

#### 4.3.7 Admin can edit NGOs/Volunteers

Sequence Diagram below shows the working of Editing NGO and Volunteers.

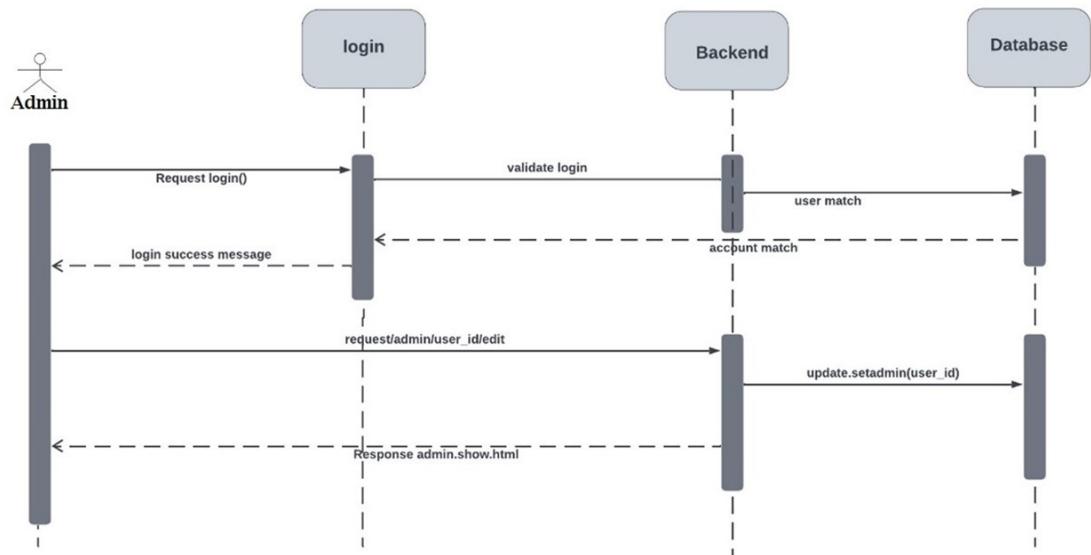


Figure 4.8: Sequence Diagram for Admin Edit Data

#### 4.3.8 Admin can delete NGOs/Volunteers

Sequence Diagram below shows the working of Deleting NGO and Volunteer.

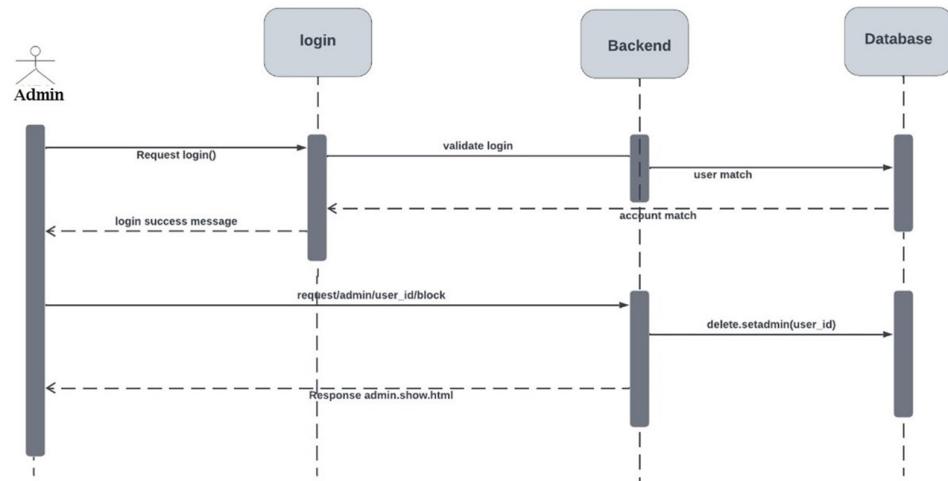


Figure 4.9: Sequence Diagram for Admin Delete Data

#### 4.3.9 Logout

Sequence Diagram below shows the working of Logout Function

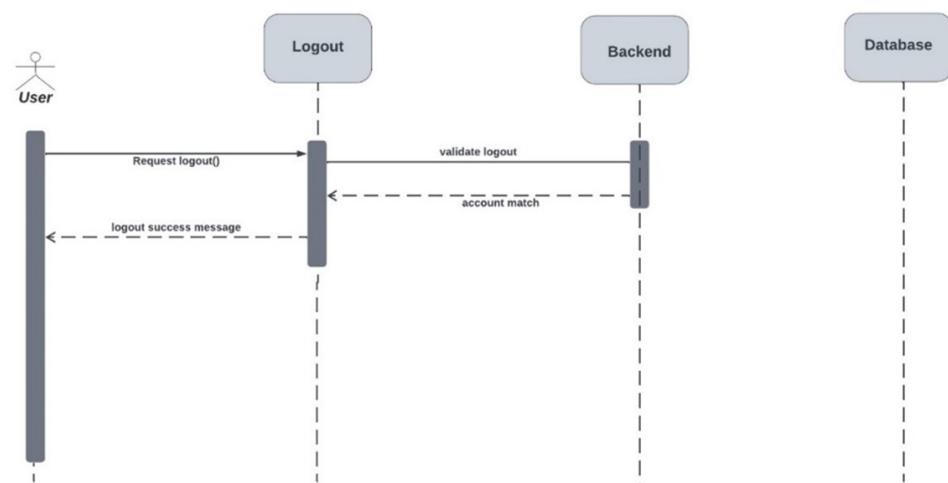


Figure 4.10: Sequence Diagram for Logout

## 4.4 GUI Design

In order to provide a comprehensive overview of our community service project, we are pleased to present all of the graphical user interface (GUI) designs that have been developed. These designs have been thoughtfully created to address the specific needs and requirements of our project, ensuring a user-friendly and intuitive interface for all stakeholders involved.

### 4.4.1 Homepage

Below Is the Design of Homepage. It gives the overview of Community Support Portal.

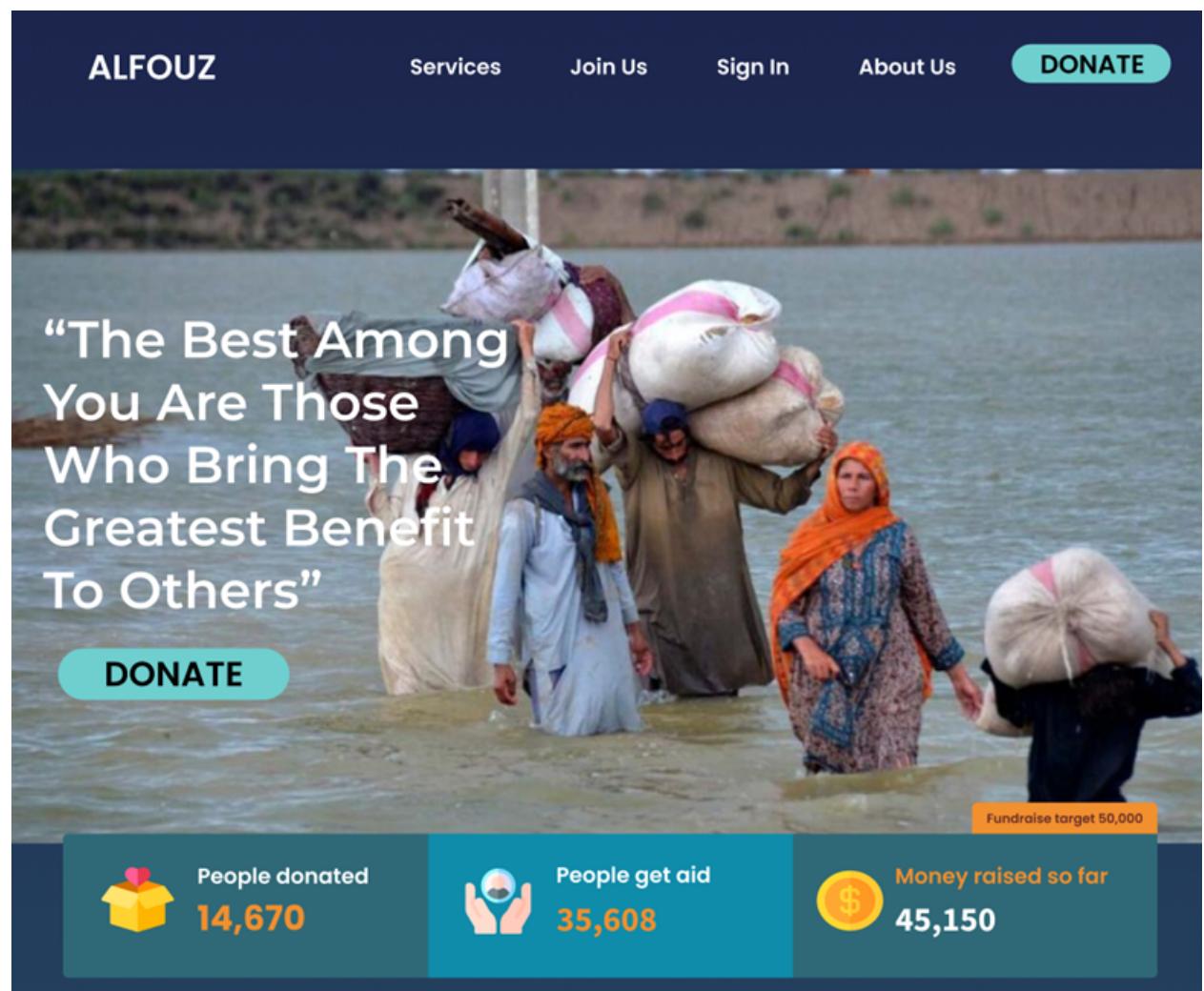


Figure 4.11: Homepage of Proposed Community Service Platform

#### 4.4.2 Services

Below Is the Design of Services Section on our homepage. It gives an idea of the types of services that the Community Service Portal offers.



Figure 4.12: Offered Services Design

#### 4.4.3 Active Projects

Below Is the Design of Active Projects Section on our homepage. It gives the idea of currently active projects on Community Service Portal.

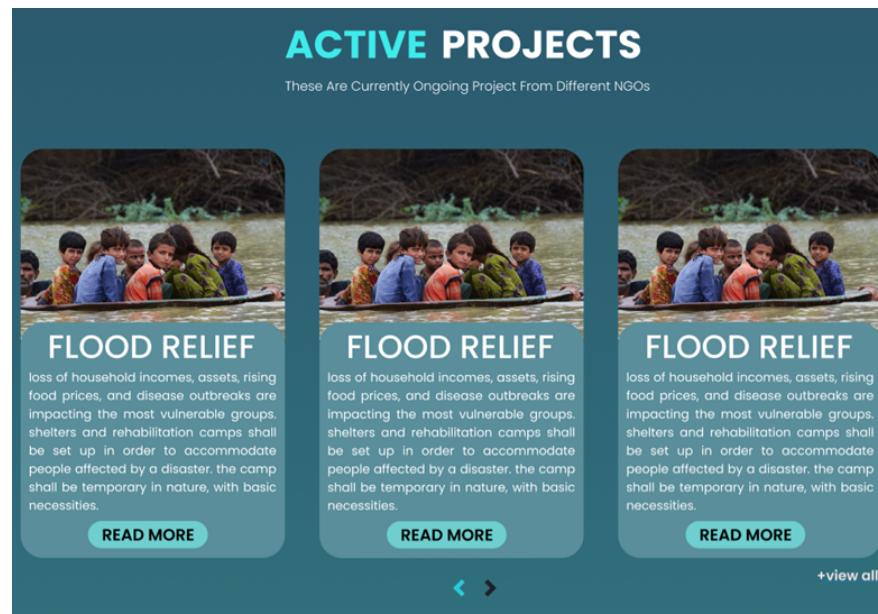


Figure 4.13: Active Projects Design

#### 4.4.4 Partner NGO's

Below Diagram gives the information about our partner NGOs.

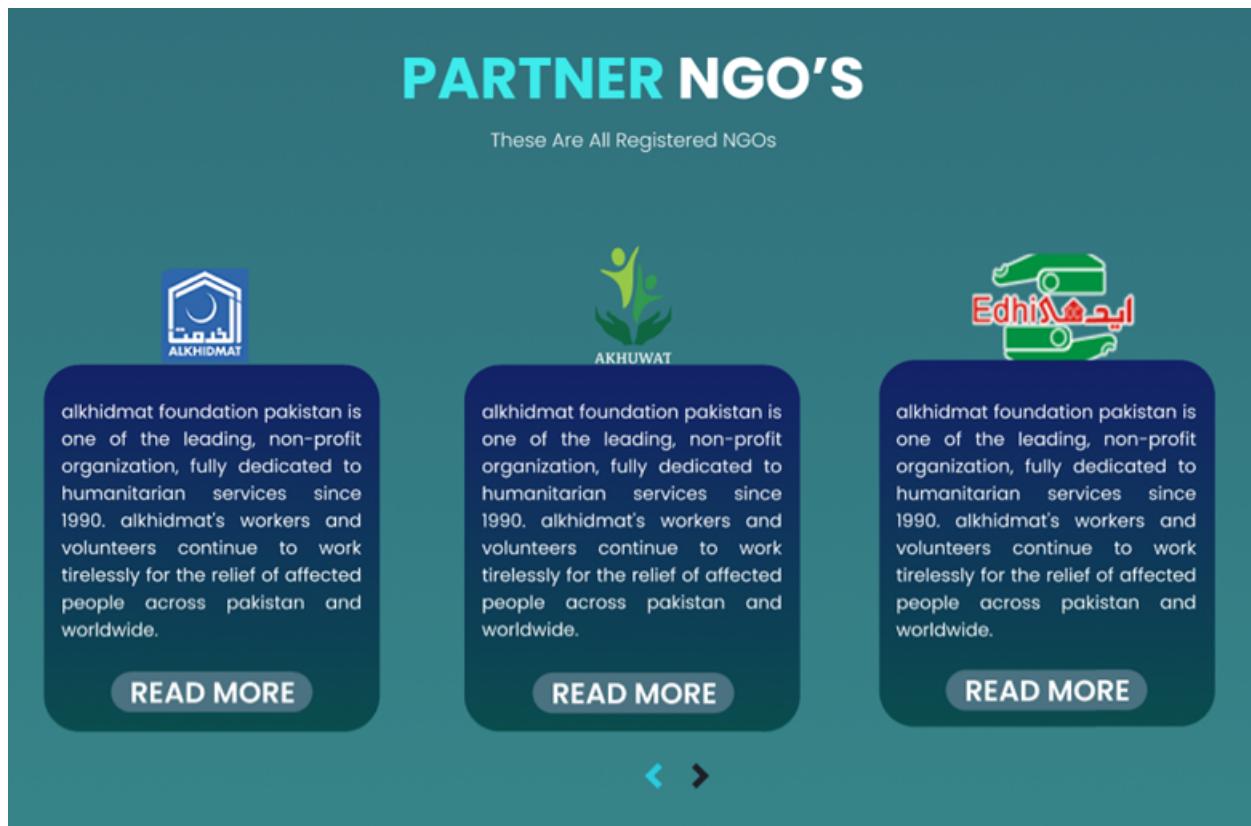


Figure 4.14: Partner NGO's Design

#### 4.4.5 Footer

Below Is the Design of Footer Section on our homepage.

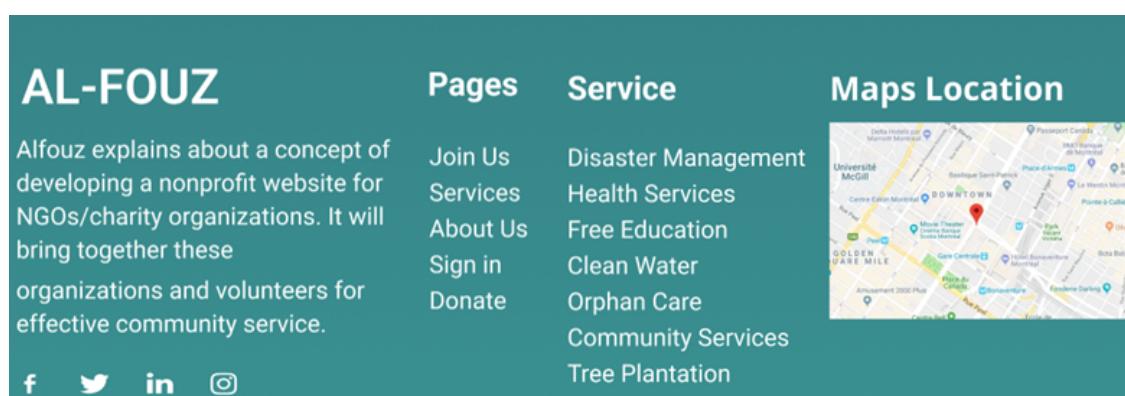


Figure 4.15: Footer Design

#### 4.4.6 NGO Profile

Below Is the Design of NGO Details Page. It provides details about the vision of the NGO.

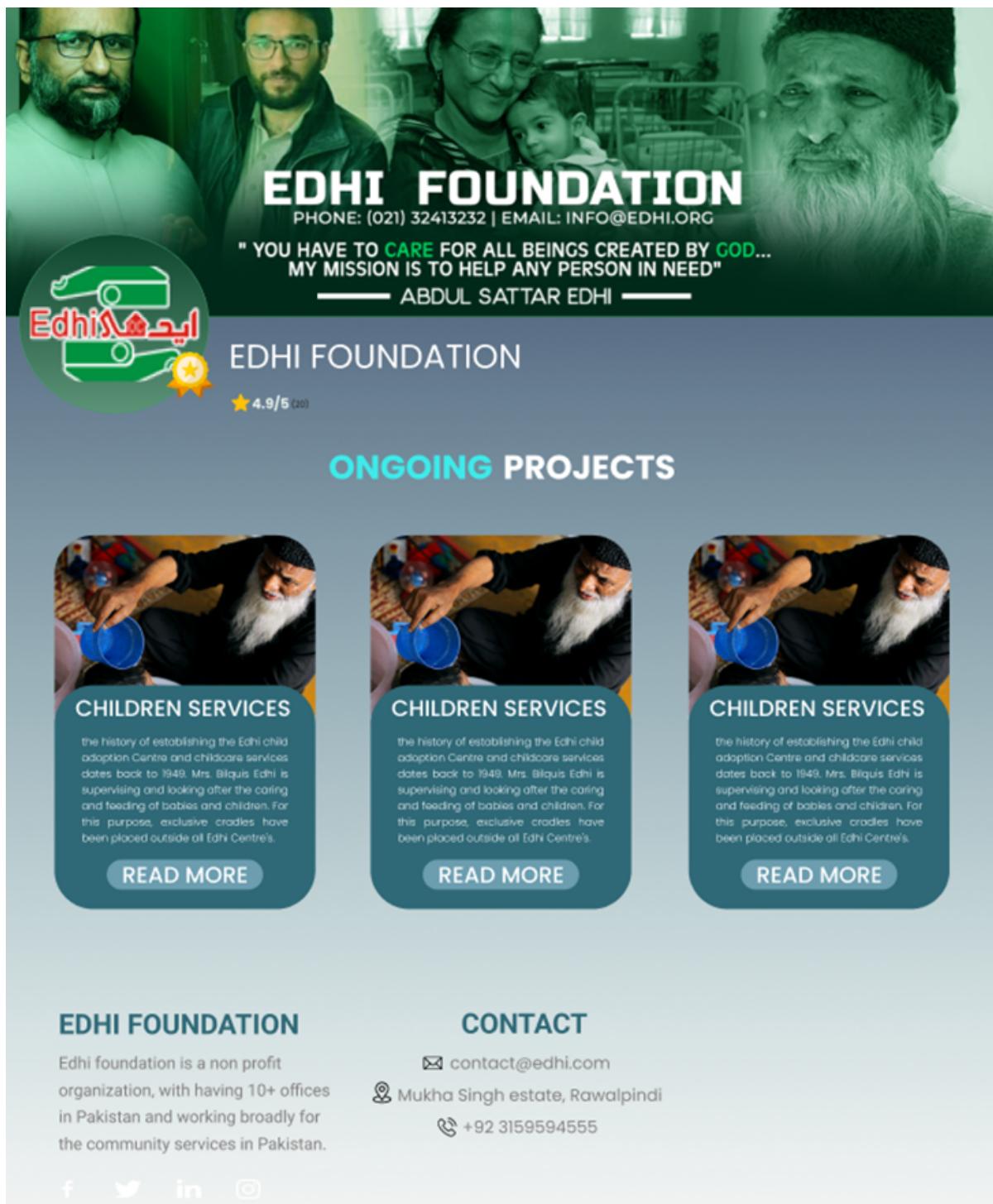


Figure 4.16: NGO Details Page Design

#### 4.4.7 Admin Dashboard

Below Is the Design of Admin Dashboard. It provides accessibility to different functionality that the Community Support Platform has to offer to an Admin.

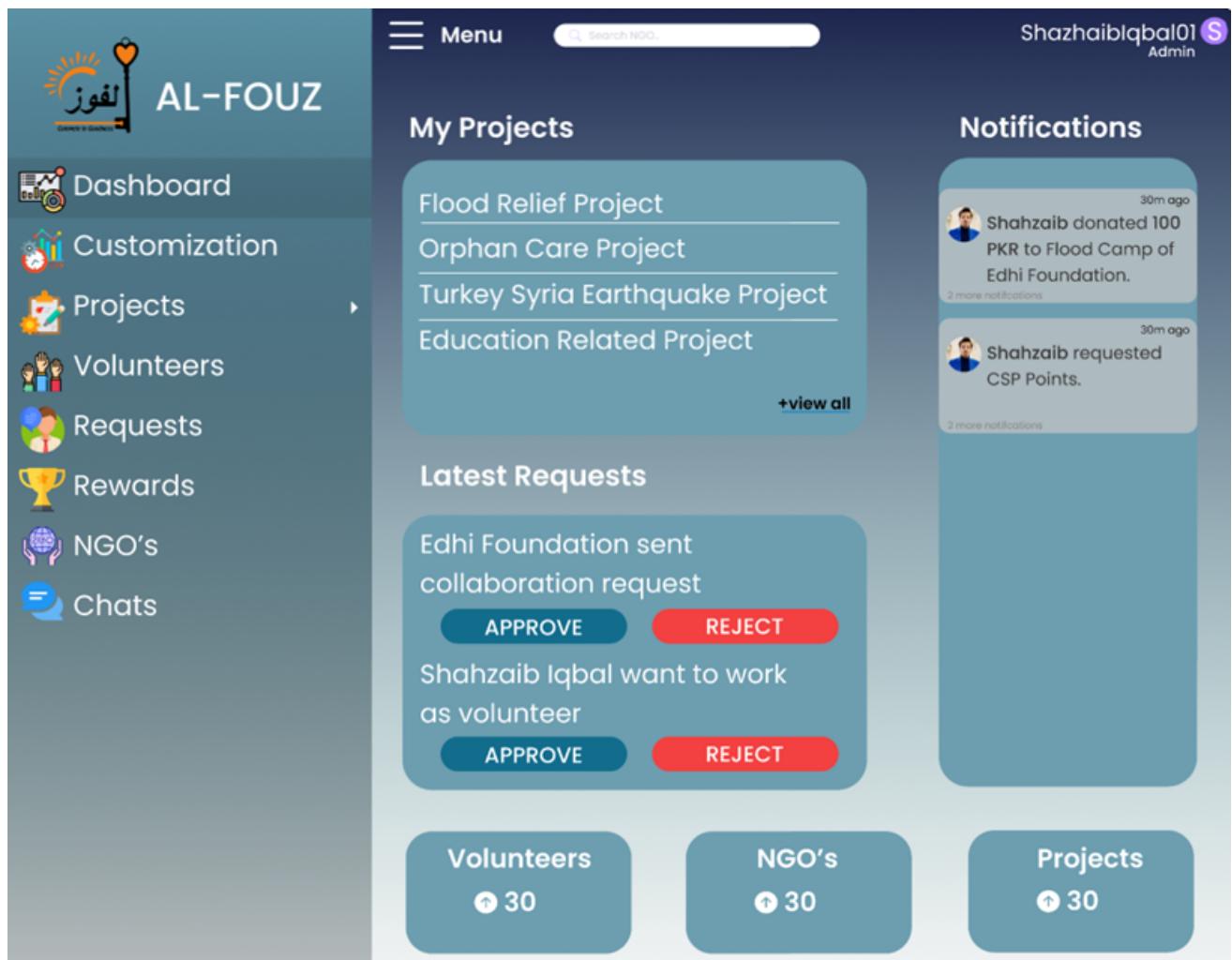


Figure 4.17: Admin Dashboard Design

#### 4.4.8 Customization Page

Admin can manage web slider and add, delete service categories and also add some basic information as well.

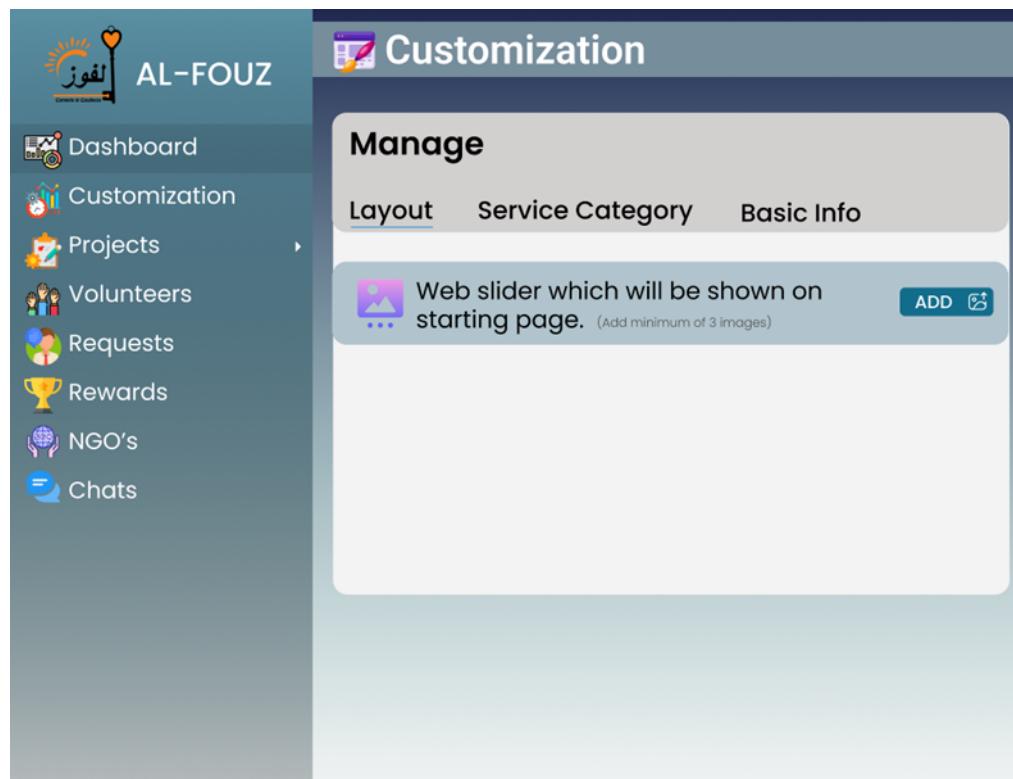


Figure 4.18: Customization Page Design

#### 4.4.9 Projects Page

Admin can add, view, and perform different operations on projects.

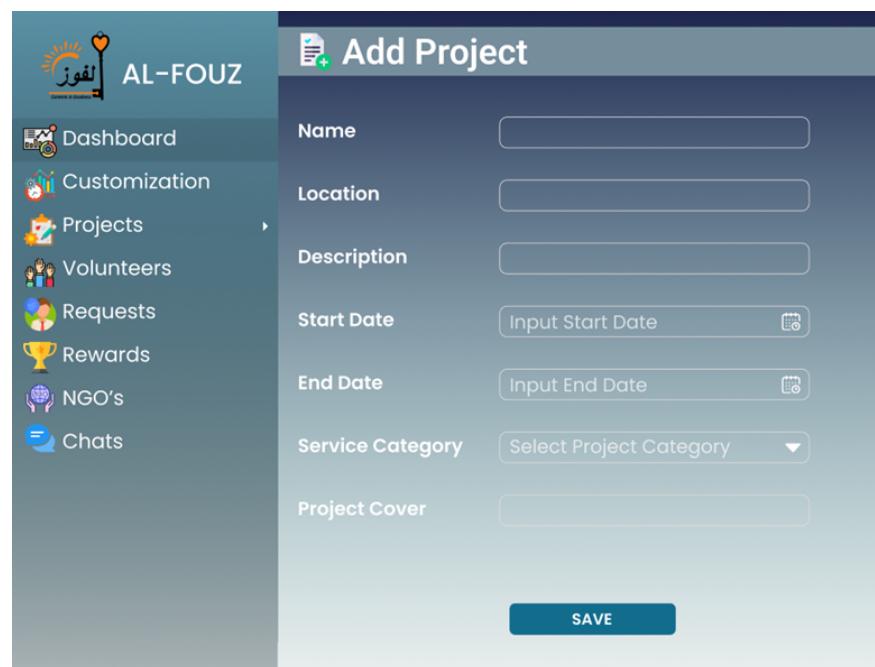


Figure 4.19: Add Project Design

#### 4.4.10 View Projects

Admin can view the status of the project and see all the volunteers enrolled in particular project.

#	Name	Category	Start Date	End Date	Status	Volunteers
1.	Flood	Disaster	01/02/2023	15/02/2023	Done	<a href="#">SEE ALL</a>
2.	Orphan	Community	01/02/2023	15/02/2023	Done	<a href="#">SEE ALL</a>
3.	Winter	Disaster	01/02/2023	15/02/2023	Pending	<a href="#">SEE ALL</a>
4.	Iftar	Community	01/02/2023	15/02/2023	Done	<a href="#">SEE ALL</a>
5.	Sehri	Community	01/02/2023	15/02/2023	Pending	<a href="#">SEE ALL</a>

Figure 4.20: View All Projects Design

#### 4.4.11 Volunteers

All volunteers they are currently enrolled in any of project.

#	Name	Contact No.	Address	Projects	User Status
1.	Shahzaib	0315-095456	Street # 3	<a href="#">SEE ALL</a>	ENABLED-
2.	Sheeza	0315-095456	Street # 3	<a href="#">SEE ALL</a>	ENABLED-
3.	Khizer	0315-095456	Street # 3	<a href="#">SEE ALL</a>	ENABLED-
4.	Junaid	0315-095456	Street # 3	<a href="#">SEE ALL</a>	ENABLED-

Figure 4.21: Volunteers Page Design

#### 4.4.12 Manage Requests

Admin has the ability to accept the requests of new NGO's, volunteers, project launch, and CSP points.



Figure 4.22: Manage Requests Design

#### 4.4.13 Registered NGO's

Admin can view the information about registered NGO's.



Figure 4.23: Registered NGOs Page Design

#### 4.4.14 Volunteer Dashboard

Below Is the Design of volunteer Dashboard. It provides accessibility to different functionalities that the Community Support Platform has to offer to volunteers.

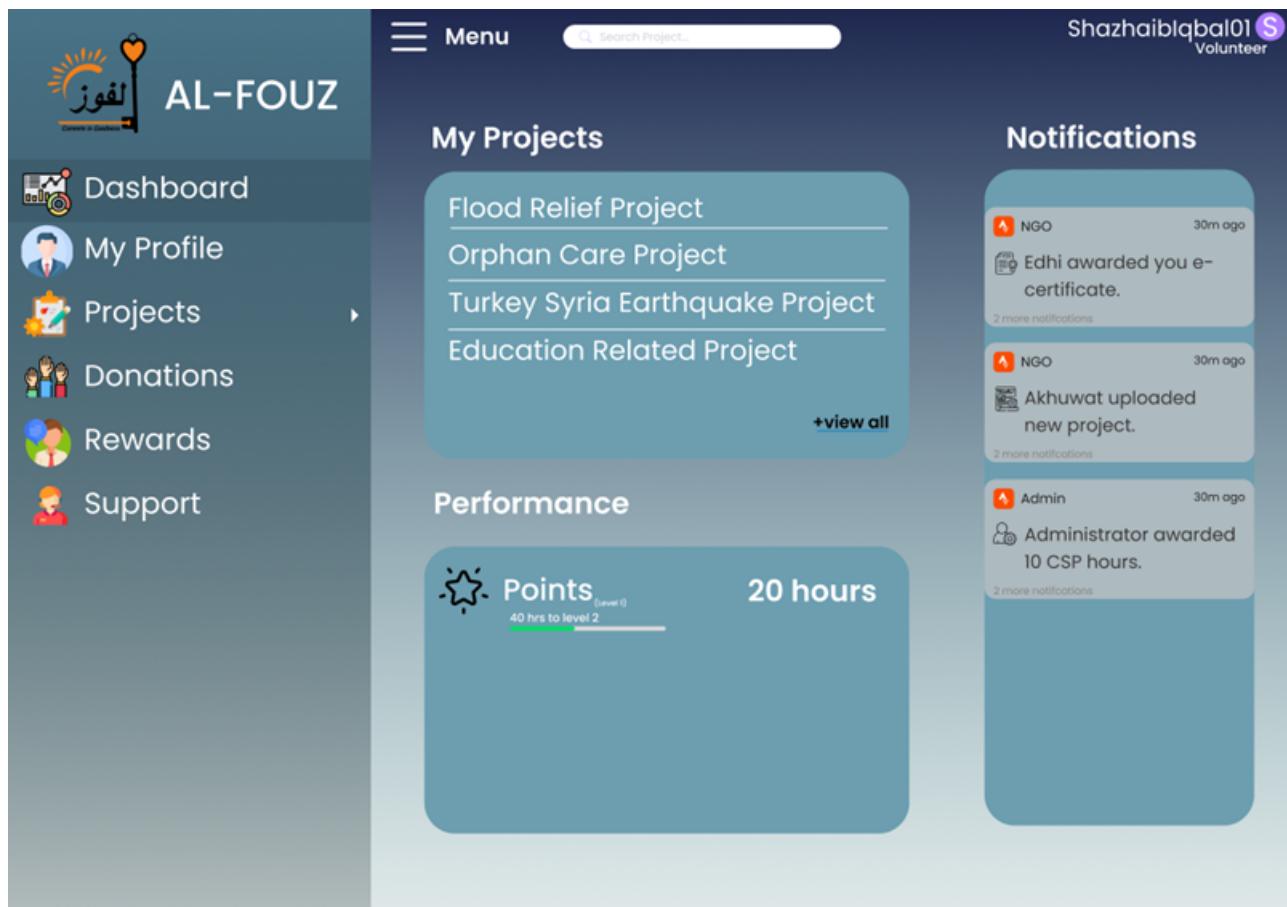


Figure 4.24: Volunteer Dashboard Design

The volunteer dashboard serves as a centralized hub for volunteers, providing them with a comprehensive overview of their involvement and contribution to our community service initiatives. Designed with their needs in mind, the dashboard offers essential features and functionalities that empower volunteers to navigate their engagement effectively.

#### 4.4.15 Volunteer Profile Information

Volunteer can edit and see their profile information.

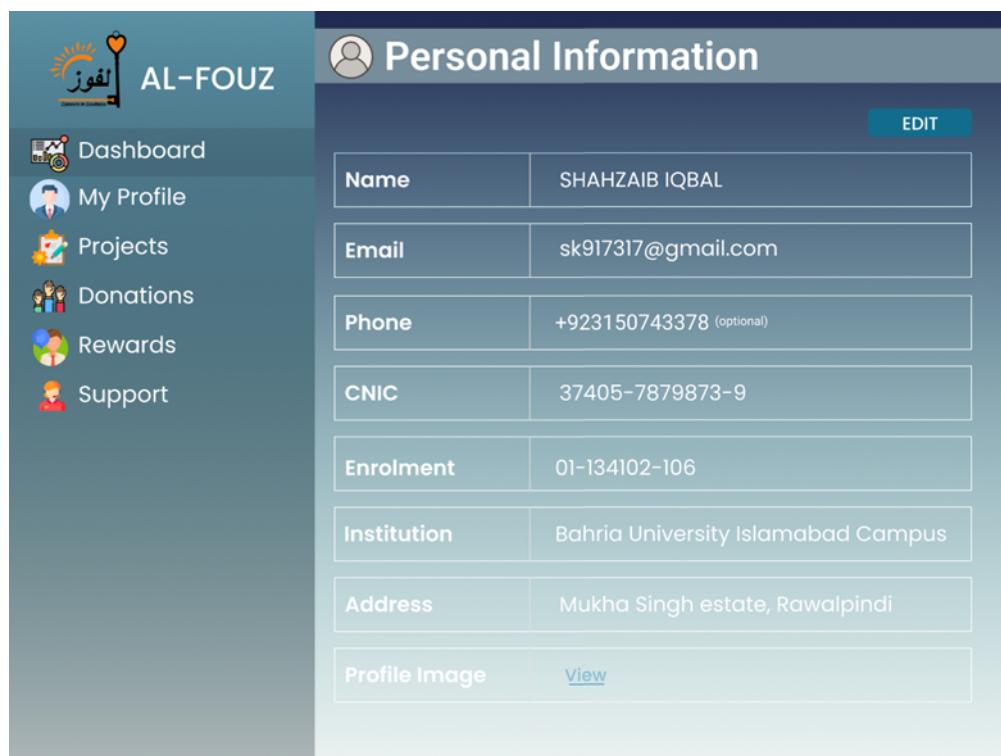


Figure 4.25: Volunteer Profile Design

#### 4.4.16 Projects

Volunteer can see all projects in which they are enrolled.

The screenshot shows the 'My Projects' section of the AL-FOUZ volunteer profile. The left sidebar has icons for Dashboard, My Profile, Projects, Donations, Rewards, and Support. The main area has a title 'My Projects' and a table listing five projects:

#	Name	Category	Start Date	End Date	Status
1.	Flood	Disaster	01/02/2023	15/02/2023	Done ▾
2.	Orphan	Community	01/02/2023	15/02/2023	Done ▾
3.	Winter	Disaster	01/02/2023	15/02/2023	Done ▾
4.	Iftar	Community	01/02/2023	15/02/2023	Pending ▾
5.	Sehri	Community	01/02/2023	15/02/2023	Pending ▾

Figure 4.26: Volunteer Projects Design

#### 4.4.17 View all Projects

Volunteer can view all active projects and have ability to enroll in any project.



Figure 4.27: Volunteer All Projects Design

#### 4.4.18 Donations

Volunteer can donate to any of registered NGO.

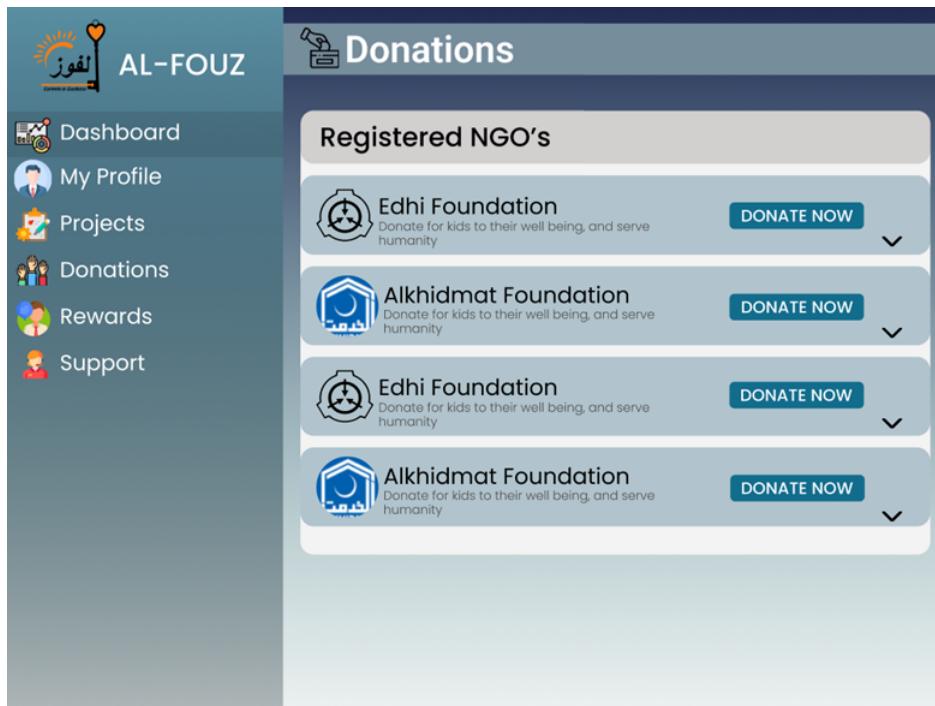


Figure 4.28: Donations Page Design

#### 4.4.19 Rewards

As a token of appreciation for their valuable contributions and dedicated efforts, volunteers have the opportunity to receive rewards through our community service project. These rewards are designed to recognize and celebrate the commitment of volunteers, motivating them to continue making a positive impact in our community. One form of reward available to volunteers is an e-certificate. In addition to e-certificates, volunteers can also accumulate CSP (Community Service Points) points as a reward for their active participation.

#	Name	Start Date	End Date	Status	Action
1.	Flood	01/02/2023	15/02/2023	Done	<a href="#">GET REWARD ▾</a>
2.	Orphan	01/02/2023	15/02/2023	Done	<a href="#">GET REWARD ▾</a>
3.	Winter	01/02/2023	15/02/2023	Done	<a href="#">GET REWARD ▾</a>
4.	Iftar	01/02/2023	15/02/2023	Done	<a href="#">GET REWARD ▾</a>
5.	Sehri	01/02/2023	15/02/2023	Done	<a href="#">GET REWARD ▾</a>

Figure 4.29: Reward Page Design

#### 4.4.20 NGO Dashboard

Below Is the Design of volunteer Dashboard. It provides accessibility to different functionalities that the Community Support Platform has to offer to volunteers.

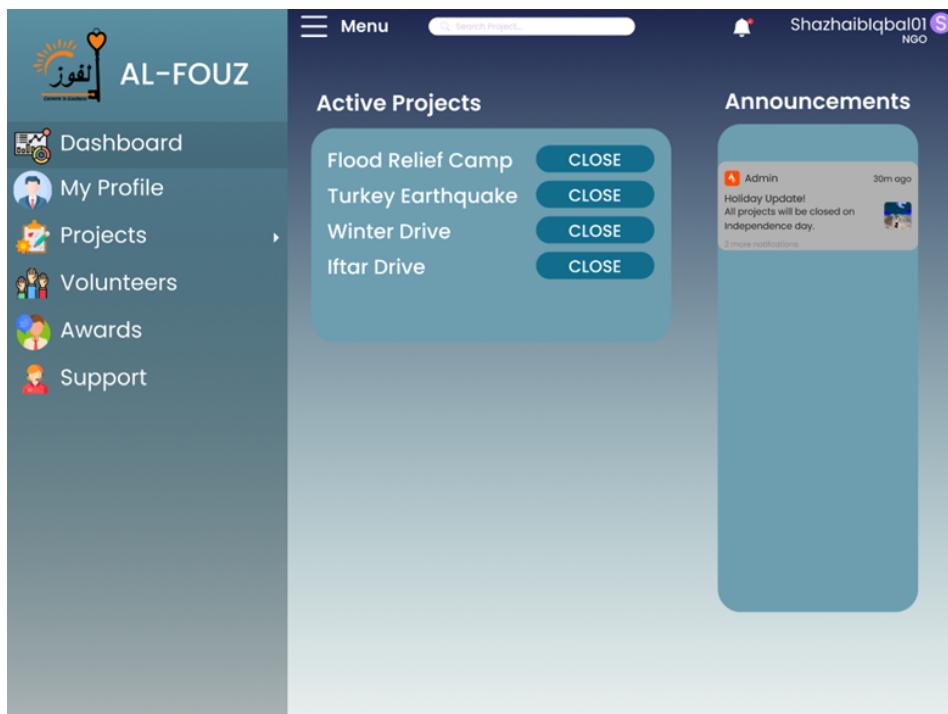


Figure 4.30: NGO Dashboard Design

#### 4.4.21 NGO Profile

NGO can see and edit their profile.

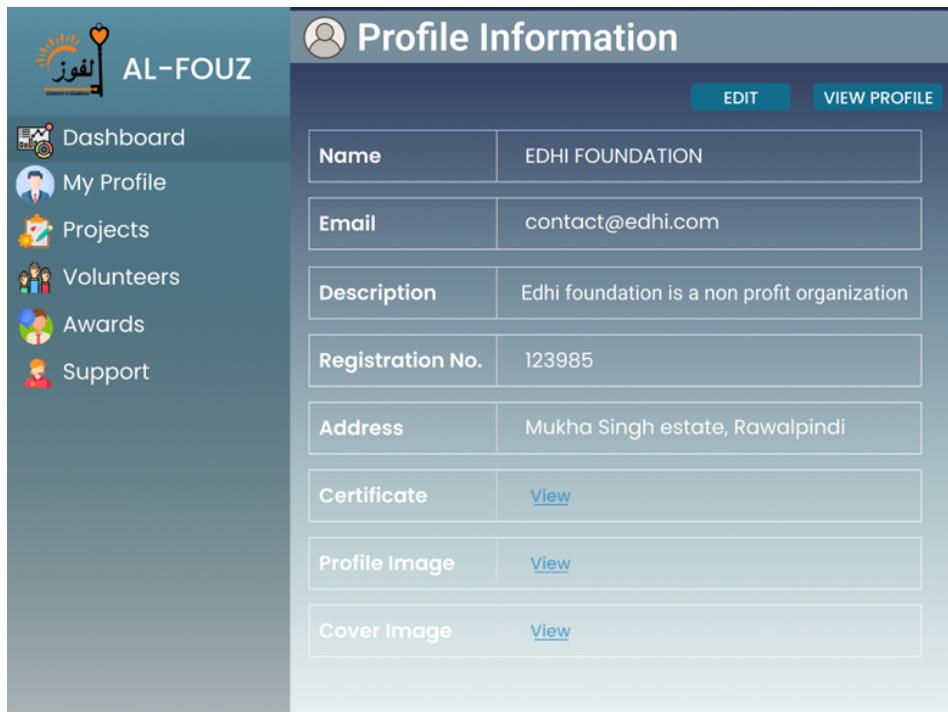


Figure 4.31: NGO Profile Information

#### 4.4.22 Projects

Ngo can add and view all the projects.

The screenshot shows the 'Add Project' form. The sidebar on the left includes the 'Projects' icon, which is highlighted. The main form fields are:

- Name: [Input field]
- Location: [Input field]
- Description: [Input field]
- Start Date: [Input field] with a calendar icon
- End Date: [Input field] with a calendar icon
- Service Category: [Select dropdown] labeled 'Select Project Category'
- Project Cover: [Input field]

A blue 'SAVE' button is located at the bottom right of the form.

Figure 4.32: NGO Add Project

#### 4.4.23 Volunteers page

NGO can view all the enrolled volunteers.

The screenshot shows the 'Volunteers' page. The sidebar on the left includes the 'Volunteers' icon, which is highlighted. The main area displays a table of volunteers:

#	Name	Contact No.	Address
1.	Shahzaib	0315-095456	Street # 3, Rawalpindi
2.	Sheeza	0315-095456	Street # 3, Rawalpindi
3.	Khizer	0315-095456	Street # 3, Rawalpindi
4.	Junaid	0315-095456	Street # 3, Rawalpindi

Figure 4.33: Volunteer Page Design

#### 4.4.24 Awards Page

NGO can give the incentives in the form of e-certificate to volunteers.

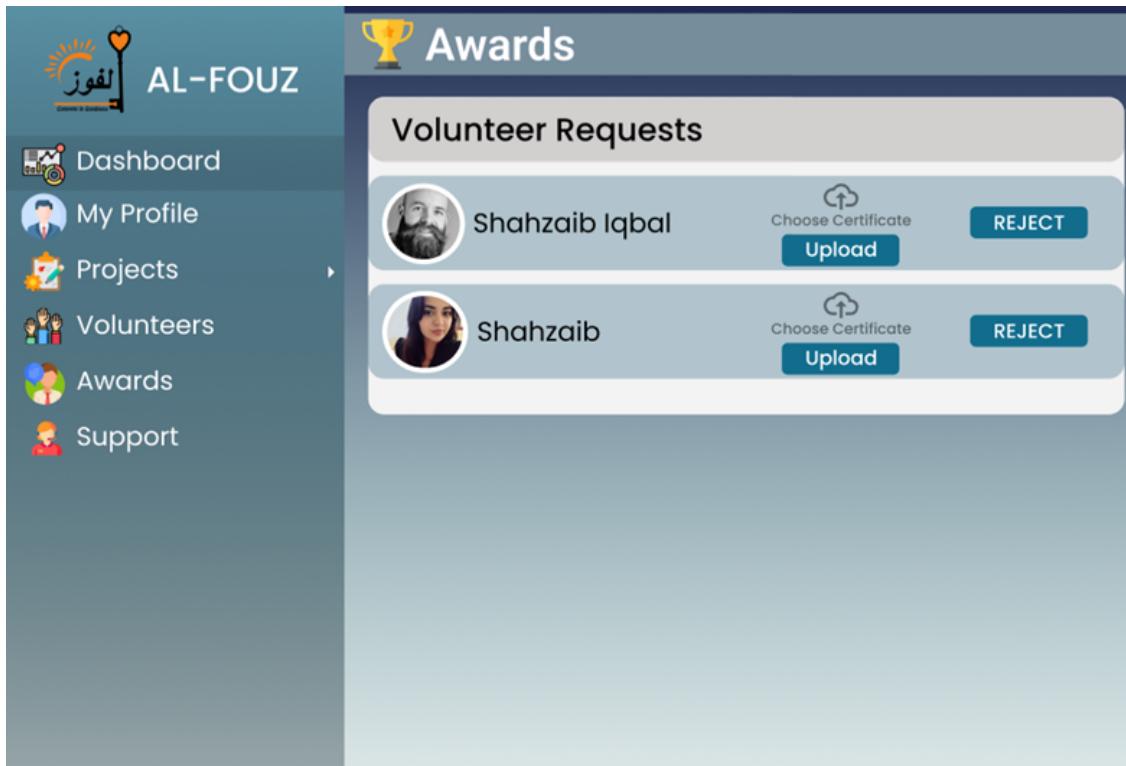


Figure 4.34: Awards Page Design

## 4.5 External Interfaces

The platform uses GUI based external interface. It doesn't require any command line interface. But we also want that the web application handles server requests in more efficient manner. So, the external interface requirements were.

- Windows Server/10
- MS SQL Server
- IIS Web Server

# **Chapter 5**

# **System Implementation**

## **5.1 System Architecture**

The Collaborative Platform for Community Services is designed using a three-tier architecture, which is a popular architecture pattern in modern software development. This architecture pattern ensures separation of concerns, maintainability, and scalability. The three tiers are as follows:

### **5.1.1 Client Layer**

The client layer represents the front-end user interface of the platform. This layer is responsible for presenting the platform's data and functionality to the users in an intuitive and visually appealing manner. The client layer has been developed using React, a popular JavaScript library for building user interfaces, and Material-UI, a comprehensive UI framework that provides a wide range of pre-built components and design elements based on Google's Material Design guidelines.

The client layer consists of several views and components that are organized into a modular structure. Each view corresponds to a specific functionality or section of the platform, such as user authentication, project browsing, event creation, and communication. The components are reusable pieces of UI that can be easily combined and customized to create various user interface elements.

### **5.1.2 Application Layer**

The application layer, developed using Node.js and Express, is responsible for handling user requests, processing data, and managing business logic. This layer communicates with the client and database layers i.e., it serves as the middleware between the client and

database layers. Developed using Node.js and Express, a lightweight web application framework, this layer is responsible for handling user requests, processing data, and managing business logic. The application layer ensures that the client layer's requests are appropriately processed and forwarded to the database layer and that the data retrieved from the database layer is formatted and delivered back to the client layer.

In addition to handling user requests, the application layer is also responsible for implementing various security measures, such as user authentication and authorization. JSON Web Tokens (JWT) are used for securely transmitting user information between the client and application layers, ensuring that only authorized users can access certain functionalities and data.

### 5.1.3 Database Layer

The database layer is responsible for storing and managing the platform's data. MongoDB, a NoSQL database system, was chosen for this purpose due to its flexible data model, efficient querying capabilities, and horizontal scalability. The data stored in the database layer includes user profiles, community service projects, events, and communication records.

The database layer's schema is designed in a way that allows for easy data retrieval and modification. Collections are used to organize data into logical groups, and indexes are applied to optimize query performance. The application layer communicates with the database layer through a well-defined API, which ensures that data is consistently accessed and manipulated.

## 5.2 Internal System Components

The internal components of the Collaborative Platform for Community Services are organized into several modules to streamline the system's functionality and improve maintainability. Each module is responsible for handling a specific set of functionalities, as described below:

### 5.2.1 User Management

The User Management module is responsible for handling all aspects of user registration, authentication, and profile management. This includes creating and managing user accounts, handling user authentication, and allowing users to update their profile information, change passwords, and set preferences. This module uses JWT-based authentication to ensure that only authorized users can access specific functionalities and data. It also incorporates various security measures, such as password hashing and input validation, to protect user data and prevent unauthorized access.

Additionally, the User Management module facilitates role-based access control, ensuring that users are granted access only to the features and data appropriate for their role (e.g., volunteer, organization, or administrator). By providing role-based access control, the system maintains a clear separation of responsibilities and streamlines user experiences by only presenting relevant functionalities.

This module uses JWT-based authentication to ensure that only authorized users can access specific functionalities and data. It also incorporates various security measures, such as password hashing and input validation, to protect user data and prevent unauthorized access.

The module also includes a comprehensive search feature that allows users to find projects based on various criteria, such as interests, skills, or location. This functionality enables users to form connections with others who share their interests or work on similar community projects, fostering a sense of community and collaboration within the platform.

### **5.2.2 Project Management**

The Project Management module enables NGOs to create, manage, and track community service projects. NGOs can create new projects, define project requirements, set deadlines, and assign tasks to volunteers. Volunteers can browse available projects, apply to participate, and track their progress. Administrators can approve or reject project proposals and monitor ongoing projects.

This module incorporates various features to streamline project management, such as automatic notifications, task prioritization, and progress tracking. It also provides tools for collaboration and communication among project participants, ensuring that everyone stays informed and engaged. Furthermore, the Project Management module offers advanced filtering and sorting options to help users find projects that align with their interests and skills, making it easier for volunteers to find suitable opportunities and organizations to find exemplary volunteers.

### **5.2.3 Communication and Collaboration**

The Communication and Collaboration module is responsible for facilitating communication and collaboration among platform users. This includes features like direct messaging, discussion forums, and file sharing. Users can communicate with the admin in real time, exchange ideas and resources, and collaborate on projects. This module also includes a notification system that alerts users to essential updates, such as new project assignments and messages from other users. The notification system is designed to be customizable, allowing users to choose which types of notifications they want to receive and how they wish to receive them (e.g., email, push notification, or in-app alert).

To support diverse communication needs, the module incorporates various communication channels, such as instant messaging and FAQs. These channels provide users with a wide range of options to collaborate effectively and stay connected, regardless of their location or preferred communication method.

#### **5.2.4 Feedback and Reviews**

The Feedback and Reviews module facilitates collecting and managing user feedback and reviews for projects, events, and organizations. Volunteers can rate and review their experiences, while organizations can gather valuable feedback to improve future projects and events. This module enables the platform to maintain a transparent and accountable environment that encourages continuous improvement and high-quality community service opportunities.

The module also includes tools for managing and analyzing feedback data, such as trend identification. These insights can help NGOs and administrators identify areas for improvement and inform future decision-making processes.

#### **5.2.5 Volunteer Skill Matching**

The Volunteer Skill Matching module ensures that volunteers are matched with projects that align with their skills, interests, and availability. This module allows volunteers to create profiles detailing their skills, experience, and preferences, while organizations can specify the required skills and qualifications for each project.

The module uses a simple matching technique to suggest suitable projects to volunteers, maximizing the likelihood of successful and fulfilling project experiences. This system helps organizations find the most suitable volunteers for their projects and encourages volunteers to develop their skills and engage in meaningful community service opportunities. [12]

#### **5.2.6 Knowledge Sharing and Best Practices**

The Knowledge Sharing and Best Practices module fosters a collaborative environment where NGOs and volunteers can exchange knowledge, experiences, and best practices related to community service projects.

#### **5.2.7 Scheduling and Availability Management**

The Scheduling and Availability Management module allows volunteers and organizations to manage their schedules and availability for community service projects. Volunteers can indicate their availability and preferences, while NGOs can specify project timelines and deadlines.

This module provides tools for automatic scheduling and conflict resolution, ensuring that volunteers are not double-booked and that project timelines are met. It also offers calendar integration, allowing users to synchronize their project schedules with their calendars for seamless time management. This module enhances the overall organization and efficiency of the platform, ensuring that projects run smoothly, and volunteers are utilized effectively. [13]

### 5.3 Tools and Technology

The Collaborative Platform for Community Services was developed using various modern tools and technologies, which were chosen for their suitability to the project requirements, ease of use, and widespread adoption in the software development community. Some of the key tools and technologies used in the project include:

- **Front-end:** React, Material-UI, Redux, Axios
- **Back-end:** Node.js, Express
- **Database:** MongoDB, Mongoose
- **Authentication:** JWT
- **Version Control:** Git, GitHub
- **Project Management:** Bitrix24

### 5.4 System Methodology

The project followed an Agile methodology, emphasizing iterative development, collaboration, and continuous improvement. The Agile approach allowed the development team to respond quickly to changing requirements, incorporate user feedback, and deliver a high-quality product in a timely manner. By adopting Agile, the team could prioritize features and improvements based on their importance and value to the end-users, ensuring that the most critical functionalities were implemented first.

The project was divided into several sprints, lasting two to four weeks. During each sprint, the team focused on implementing specific features or improvements, which were identified and prioritized based on user feedback and project goals. At the beginning of each sprint, the team held a sprint planning meeting to discuss the objectives, estimate the effort required, and assign tasks to individual team members. Throughout the sprint, the group maintained constant communication, sharing progress updates and collaborating on problem-solving.

In addition to following the Agile methodology, the project also incorporated best practices from other software development methodologies, such as Test-Driven Development (TDD) and Continuous Integration (CI). Test-Driven Development involves writing tests for new features or improvements before implementation, ensuring that the code meets the desired functionality and is less prone to errors. This approach helped maintain the codebase's quality and facilitated a smoother integration of new components into the existing system.

Conversely, continuous integration is a practice where developers frequently merge their code changes into a shared repository, ideally several times a day. This process helps to identify and fix integration issues early, reducing the risk of conflicts and ensuring a smoother development process. By implementing CI, the team was able to keep the platform's codebase up-to-date, reducing the likelihood of unexpected issues during deployment and facilitating a faster release cycle.

The combination of Agile, TDD, and CI practices allowed the team to work efficiently, adapt to changes, and maintain a high level of quality throughout the development process. As a result, the Collaborative Platform for Community Services was able to meet the needs of its users, providing a robust and reliable solution for NGOs and volunteers to collaborate on community service projects.

# **Chapter 6**

# **System Testing and Evaluation**

## **6.1 Introduction**

System testing and evaluation are essential components of the software development process, ensuring that the collaborative platform for community services is reliable, efficient, and user-friendly. This chapter aims to outline the various testing techniques employed to evaluate the system thoroughly, identify any bugs or issues, and improve the overall quality of the platform. The testing process includes Graphical User Interface, Usability, Integration, Exception, Performance, and Installation/Deployment Testing. Moreover, specific test cases are provided in a tabular format to demonstrate the testing process in action.

## **6.2 Graphical User Interface Testing**

Graphical User Interface (GUI) testing is crucial to ensure the platform's visual elements are functioning correctly and providing an intuitive user experience. It involves testing the user interface components, such as buttons, forms, menus, and navigation elements, for correct appearance, behavior, and responsiveness. During this testing phase, testers focus on design consistency, color schemes, fonts, and layout to ensure the application's interface is visually appealing and easy to use.

### **6.2.1 Navigation Testing**

Navigation testing verifies the proper functioning of navigational elements, such as menus, buttons, and links. This testing phase aims to ensure that users can easily navigate the platform and access different features without any confusion or difficulty. It includes

testing the main menu, dropdown menus, sidebars, and any other navigational components that guide users through the application.

### **6.2.2 Form Testing**

Form testing focuses on validating the proper functioning of various forms used throughout the platform. This includes testing input fields for accepting the correct data types and lengths, validating form submissions, checking error messages, and ensuring that required fields are correctly marked. Form testing is essential for ensuring that users can easily submit information and interact with the platform.

## **6.3 Usability Testing**

Usability testing evaluates the application's overall user experience, ensuring the platform is intuitive, easy to use, and meets user expectations. It typically involves real users interacting with the platform and gathering feedback regarding their experience. This feedback is invaluable in identifying areas for improvement and ensuring that the platform is genuinely user-friendly.

### **6.3.1 User Scenarios**

User scenarios are developed to guide the usability testing process, reflecting the tasks and goals typical users would want to achieve while using the platform. These scenarios help testers simulate real-world usage and evaluate the application's ability to meet user needs effectively. Examples of user scenarios for the collaborative platform for community services may include:

- Signing up as a new user
- Searching for community service opportunities
- Creating and managing a community service project
- Collaborating with other users on a project
- Providing feedback on completed projects

### **6.3.2 User Feedback**

After users have completed the usability testing process, their feedback is collected and analyzed to identify areas for improvement. This feedback may include comments on the platform's overall design, specific features, navigation, or any other aspect of the user

experience. The development team can then use this feedback to refine the platform and enhance its usability.

## 6.4 Integration Testing

Integration testing is a crucial phase in the testing process, which focuses on verifying the proper functioning of individual components when integrated into the complete system. This testing phase aims to identify any issues that may arise when different modules of the collaborative platform for community services interact. Test cases are developed to simulate various integration scenarios, such as communication between the front end and back end, database interactions, and third-party API integrations. [14]

## 6.5 Exception Testing

Exception testing evaluates the platform's ability to handle unexpected situations or erroneous inputs. This testing phase aims to identify any issues that may arise when the system encounters unpredictable conditions, such as incorrect user input, server errors, or connection issues. Exception testing ensures the platform can handle these situations gracefully, displaying appropriate error messages and guiding the user toward the correct course of action.

### 6.5.1 Input Validation

Input validation testing is an essential part of exception testing that focuses on verifying the system's ability to handle various incorrect or malicious user input types. Test cases are designed to input unexpected values or data types into the platform's forms and fields to check if the system can identify and reject the invalid input. This process helps ensure the platform remains secure and robust despite potential security threats or user errors.

### 6.5.2 Error Handling

Error handling testing evaluates the platform's ability to handle different types of errors, such as server errors, network issues, or database errors. This testing phase aims to ensure that the system can effectively detect, log, and communicate these errors to the user, providing them with clear instructions on how to resolve the issue. Proper error handling is crucial for maintaining a positive user experience and minimizing frustration.

## **6.6 Performance Testing**

Performance testing is conducted to evaluate the platform's responsiveness, stability, and scalability under various workload conditions. This testing phase helps identify potential bottlenecks or performance issues that may impact the user experience or the platform's ability to handle many users simultaneously.

### **6.6.1 Load Testing**

Load testing is a type of performance testing that aims to evaluate the platform's behavior under a high volume of concurrent users or requests. Test cases are designed to simulate different levels of user load, from a small number of users to a large number of users, to assess the system's ability to handle the increased workload without sacrificing performance or stability.

### **6.6.2 Stress Testing**

Stress testing is another form of performance testing, focusing on the platform's behavior under extreme or abnormal conditions. This testing phase aims to push the system to its limits by subjecting it to an excessive workload or resource constraints, allowing the development team to identify any potential weaknesses or vulnerabilities in the platform.

## **6.7 Installation/Deployment Testing**

Installation and deployment testing is carried out to ensure that the collaborative platform for community services can be successfully installed, configured, and deployed on various target environments. This testing phase focuses on verifying the platform's compatibility with different operating systems, browsers, and devices. It also checks that the installation process is smooth, well-documented, and free from any errors or issues.

## **6.8 Test Cases**

The following are sample test cases for the collaborative platform for community services:

Table 6.1: User Registration Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>User Registration</b>	The user should be able to register	<b>1</b>	Open Registration Form	Registration form is displayed	Pass
		<b>2</b>	Enter required information	Users can input all required fields	Pass
		<b>3</b>	Submit Registration Form	A user account is created, and a confirmation message displayed	Pass

Table 6.2: User Login Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>User Login</b>	The user should be able to log in	<b>1</b>	Open Login Form	The login form is displayed	Pass
		<b>2</b>	Enter your email and password	Users can input email and password	Pass
		<b>3</b>	Click Login Button	The user is successfully logged in and redirected to the dashboard	Pass

Table 6.3: User Profile Update Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>User Profile Update</b>	Users should be able to update their profile	<b>1</b>	Open User Profile	The user profile page is displayed	Pass
		<b>2</b>	Update user information	Users can edit and update their information	Pass
		<b>3</b>	Save changes	Changes are saved, and a confirmation message displayed	Pass

Table 6.4: Create new Project Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>Create New Project</b>	The user should be able to create a new project	<b>1</b>	Open Project Creation Form	The project creation form is displayed	Pass
		<b>2</b>	Enter project details	Users can input project details	Pass
		<b>3</b>	Submit project form	The project is created, and a confirmation message displayed	Pass

Table 6.5: Edit Existing Project Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>Edit Existing Project</b>	The user should be able to edit a project	<b>1</b>	Open Project Edit Form	The project edit form is displayed	Pass
		<b>2</b>	Update project details	Users can update project details	Pass
		<b>3</b>	Save changes	Changes are saved, and a confirmation message displayed	Pass

Table 6.6: Delete Existing Project Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>Delete Existing Project</b>	The user should be able to delete a project	<b>1</b>	Open Project Delete Confirmation	Delete confirmation prompt is displayed	Pass
		<b>2</b>	Confirm deletion	The user can confirm the deletion	Pass
		<b>3</b>	Project deletion	The project is deleted, and a confirmation message displayed	Pass

Table 6.7: Complete a Project Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>Complete a Project</b>	The user should be able to complete a project	<b>1</b>	Open Project Details Page	Project details are displayed	Pass
		<b>2</b>	Click the complete Project Button	Users can click the complete project button	Pass
		<b>3</b>	Confirm completion project	The user completes the project and receives a confirmation message	Pass

Table 6.8: Send Rewards Request Test Case

Test Case Description	Functional Requirement	Steps	Tasks	Expected Outcome	Result
<b>Send Rewards Request</b>	The user should be able to send Rewards request on project completion	<b>1</b>	Open Project Details Page	Project details are displayed	Pass
		<b>2</b>	Click the complete Project Button	Users can click the Complete Project button	Pass
		<b>3</b>	Click on sending CSP/Certificate request button	Users can click the Request button	Pass
		<b>4</b>	Display Success Message	The request is sent, and a confirmation message is displayed	Pass

Table 6.9: Accept Rewards Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>Accept Rewards</b>	User should be able to accept rewards	<b>1</b>	Open Rewards Page	Rewards for Projects are Displayed	Pass
		<b>2</b>	Click the Get Reward Button	Users can click Get Reward button	Pass
		<b>3</b>	Confirm accepting Rewards	Reward (CSP/Certificate) is accepted and a confirmation message is displayed.	Pass

Table 6.10: Provide Ratings to NGO Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>Provide Ratings to NGO</b>	User should be able to provide ratings on a project	<b>1</b>	Display Rating Form after Sending Reward Request	The rating form is displayed	Pass
		<b>2</b>	Click Rating Stars	Users can click rating stars	Pass
		<b>3</b>	Submit Rating form	Ratings are submitted, and a confirmation message is displayed	Pass

Table 6.11: Search for a Project Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>Search for a Project</b>	The user should be able to search for a project	<b>1</b>	Open Search Projects Page	The search Projects page is displayed	Pass
		<b>2</b>	Enter search keywords	Users can input search keywords	Pass
		<b>3</b>	Click Search Button	Search results matching keywords are displayed	Pass

Table 6.12: User view profile Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>View User Profile</b>	Users should be able to view their profile	<b>1</b>	Open User Profile Page	The user Profile page is displayed	Pass
		<b>2</b>	View profile details	User can view their profile details	Pass

Table 6.13: View Project Details Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>View Project Details</b>	The user should be able to view project details	<b>1</b>	Open Project Details Page	The project Details page is displayed	Pass
		<b>2</b>	View project information	Users can view project information	Pass

Table 6.14: Leave a project Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>Leave a Project</b>	The user should be able to leave a project	<b>1</b>	Open Project Details Page	The project Details page is displayed	Pass
		<b>2</b>	Click the Leave Project Button	Users can click leave project button	Pass
		<b>3</b>	Confirm leaving project	User leaves the project, and a confirmation message is displayed	Pass

Table 6.15: Volunteer for a project Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>Volunteer for a Project</b>	The user should be able to volunteer for a project	<b>1</b>	Open Project Details Page	The project Details page is displayed	Pass
		<b>2</b>	Click Volunteer Button	A volunteer form is displayed	Pass
		<b>3</b>	Fill out the form and submit	The user is added as a volunteer, and a confirmation message is displayed	Pass

Table 6.16: View Volunteers of Project Test Case

<b>Test Case Description</b>	<b>Functional Requirement</b>	<b>Steps</b>	<b>Tasks</b>	<b>Expected Outcome</b>	<b>Result</b>
<b>View Volunteers of a Project</b>	User should be able to view volunteers of a project	<b>1</b>	Open Project Details Page	The project Details page is displayed	Pass
		<b>2</b>	Click the View Volunteers Button	A list of volunteers is displayed	Pass

# **Chapter 7**

## **Conclusions**

### **7.1 Introduction**

The collaborative platform for community services project aimed to develop a web-based solution to facilitate the management and organization of community service initiatives. This collaborative platform for community services is designed to enable users to manage, share, and participate in various community service projects. The primary goal was to create an easy-to-use platform that connects volunteers, organizations, and community members, streamlining communication and collaboration. Several challenges and obstacles emerged as the project progressed, requiring adaptability and resilience. This chapter summarizes the project, discusses problems faced and lessons learned, and outlines potential future improvements.

### **7.2 Problems Faced and Lessons Learned**

Throughout the project's development, numerous challenges arose, including technical issues, time constraints, and the need for effective collaboration among team members. The technical issues were resolved by researching and implementing the appropriate technologies and tools, such as integrating APIs and selecting the platform's most suitable programming languages and frameworks. Time management was critical to the project's success, and the team learned the importance of setting clear goals, prioritizing tasks, and maintaining open communication channels to ensure timely completion.

The team also experienced the need for effective collaboration and communication among team members. Regular meetings, progress reports, and ongoing discussions proved essential in keeping the team focused and on track. This experience highlighted

the importance of teamwork and effective communication in a project of this scale and complexity.

These challenges include:

- **Requirement Gathering:** Initially, gathering the requirements and understanding the expectations of various stakeholders was daunting. Several meetings and discussions were held to ensure that the final product catered to the needs of all involved parties.
- **Integration of Different Modules:** The platform required the integration of multiple modules, such as project management, user management, and communication features. Ensuring seamless integration between these modules and maintaining data consistency proved challenging.
- **Time Constraints:** The project had to be completed within a tight deadline, necessitating efficient time management and prioritization of tasks.

Despite the challenges, valuable lessons were learned during the project. These lessons include:

- **Effective Communication:** Clear communication with stakeholders and team members was reinforced throughout the project. Regular meetings and updates helped keep the project on track and ensured all parties were on the same page.
- **Project Management:** The experience of managing different aspects of a large-scale project, from planning and design to implementation and testing, was invaluable. This project helped me to develop skills in task prioritization, time management, and resource allocation.
- **Technical Skills:** The platform's development involved working with various web development tools and technologies, which contributed to enhancing technical skills and knowledge.

### 7.3 Project Summary

The collaborative platform for community services is a comprehensive web-based solution designed to streamline community service initiatives' organization and management. The platform features a user-friendly interface, allowing volunteers, organizations, and community members to register, create, and manage projects efficiently. The platform also includes essential functionalities like project tracking, volunteer management, event scheduling, and resource allocation. The project successfully achieved its objectives and demonstrated the potential of technology to improve collaboration and efficiency in community service initiatives.

## 7.4 Future Improvements

There is potential for future improvements and enhancements to the platform. These could include adding a mobile application to complement the web-based platform, enabling users to access the platform on the go. Furthermore, incorporating advanced analytics and reporting features would provide organizations valuable insights into community service initiatives' performance and impact. Integration with popular social media platforms could also facilitate the promotion and visibility of community service projects, attracting more volunteers and participants. Lastly, expanding the platform to cater to different languages and regions would increase its reach and make it a more inclusive and accessible tool for diverse communities.

Several potential improvements can be made to the collaborative platform for community services:

- **Mobile Application:** Developing a mobile application for the platform would make it more accessible and convenient for users, allowing them to access the platform and receive real-time updates.
- **Advanced Analytics:** Integrating advanced analytics and reporting tools would provide valuable insights to project organizers, allowing them to make data-driven decisions to improve the effectiveness of their projects.
- **Gamification:** Incorporating gamification elements, such as badges and rewards, could encourage user engagement and motivate volunteers to contribute more actively to community service projects.
- **Integration with social media:** Allowing users to share project updates and milestones on social media platforms could increase the visibility and reach of community service projects, ultimately attracting more volunteers and resources.

In conclusion, the collaborative platform for community services was successfully developed, addressing the challenges faced throughout the project and resulting in a comprehensive web-based solution for managing community service initiatives. The lessons learned during the development process have contributed to personal and professional growth, and the platform has the potential for further improvements and enhancements. This project exemplifies how technology can be harnessed to facilitate collaboration and streamline processes in the community service domain, ultimately leading to more effective and impactful projects that benefit society.

## **Appendix A**

## **User Manual**

- The administrator of the community service platform will verify NGOs based on their social welfare registration number, which is provided by the Government of Punjab.
- To ensure the authenticity of volunteers, the administrator of the community service platform will verify the information provided by them during the registration process. This verification process is crucial to establish trust and credibility within the community. The administrator will carefully review the information provided by the volunteers, including their name, contact information, and other relevant details. If the information appears genuine, the administrator will approve the volunteer's registration.
- However, if the information provided by the volunteer appears to be fake or spam, the administrator will reject the volunteer's registration. This process will prevent any mischievous or fraudulent activities on the platform and provide a secure environment for volunteers to participate in community service activities.

# References

- [1] Yi Lu and Jiuping Xu. Ngo collaboration in community post-disaster reconstruction: field research following the 2008 wenchuan earthquake in china. *Disasters*, 39(2):258–278, 2015. Cited on p. 3.
- [2] Karen Healy and Gabrielle Meagher. The reprofessionalization of social work: Collaborative approaches for achieving professional recognition. *British Journal of Social Work*, 34(2):243–260, 2004. Cited on p. 4.
- [3] Marie Lall. Creating agents of positive change—the citizens foundation in pakistan. *The Citizens Foundation, Karachi, enero de*, 2009. Cited on p. 11.
- [4] Abid Iqbal, Gulzar Ahmad, and Muhammad Akbar Malik. Edhi foundation: A struggle for humanitarian survival under the global welfare vision. 2021. Cited on p. 12.
- [5] Nasim Shah Shirazi and Aman Ullah Khan. Role of pakistan poverty alleviation fund’s micro credit in poverty alleviation: A case of pakistan. *Pakistan Economic and Social Review*, pages 215–228, 2009. Cited on p. 13.
- [6] Farhana Badar, Shahid Mahmood, Raqib Faraz, Aneel Yousaf, A Ul Quader, Hina Asif, et al. Epidemiology of breast cancer at the shaukat khanum memorial cancer hospital and research center, lahore, pakistan. *J Coll Physicians Surg Pak*, 25(10):738–742, 2015. Cited on p. 13.
- [7] Suhail Ahmad. How islamic social finance contribute to the sustainable development goals: An impactful story of al-khidmat foundation pakistan. *Islamic Wealth and the SDGs: Global Strategies for Socio-economic Impact*, pages 197–216, 2021. Cited on p. 14.
- [8] Susan B Anders. Charity navigator and guidestar. *The CPA Journal*, 85(6):72, 2015. Cited on p. 15.
- [9] Doug Johnson. Is givewell right that health interventions should prioritized over education interventions? Cited on p. 15.
- [10] Lalin Anik, Thomas J Steenburgh, and Laura Steenburgh. Globalgiving. Cited on p. 16.
- [11] Mauro Falasca, Christopher Zobel, and Cliff Ragsdale. Helping a small development organization manage volunteers more efficiently. *Interfaces*, 41(3):254–262, 2011. Cited on p. 21.

- [12] Jose Ramon Saura, Pedro Palos-Sanchez, and Felix Velicia-Martin. What drives volunteers to accept a digital platform that supports ngo projects? *Frontiers in psychology*, 11:429, 2020. Cited on p. [56](#).
- [13] R Dale Safrit, Ryan J Schmiesing, Joseph A Gliem, and Rosemary R Gliem. Competencies for contemporary volunteer administration: An empirical model bridging theory with professional best practice. *Journal of Volunteer Administration*, 23(3):5, 2005. Cited on p. [57](#).
- [14] Yuri Cartier, Caroline Fichtenberg, and Laura M Gottlieb. Implementing community resource referral technology: Facilitators and barriers described by early adopters: A review of new technology platforms to facilitate referrals from health care organizations to social service organizations. *Health Affairs*, 39(4):662–669, 2020. Cited on p. [61](#).