WebCharity

ASWIN SURESH	**Reg.No: 32021131017**
BINEESH BABU	**Reg.No: 32021131018**
PRANAV P	**Reg.No: 32021131033**
VISWAV S	**Reg.No: 3202131044**

SREE NARAYANA COLLEGE CHERTHALA

DEPARTMENT OF COMPUTER SCIENCE

SREE NARAYANA COLLEGE

CHERTHALA

2023

(Affiliated to University of Kerala)

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF BSC COMPUTER SCIENCE DEGREE OF UNIVERSITY OF KERALA

2023

SREE NARAYANA COLLEGE CHERTHALA

(Affiliated to the University of Kerala)

BSc. Computer Science

2021-2024

CERTIFICATE

This is to certify that the project work entitled **" WebCharity "** is bonafide report of the work done **ASWIN SURESH, BINEESH BABU, PRANAV P, VISWAV S** in partial fulfilment of the requirements for the award of the Bachelor's degree in BSc. Computer Science of the institution during the year 2021-2024.

Staff in charge Head of Department Principal

Presented for the viva – voice examination conducted by the University of Kerala held on …..... /. ...../ 2023 at Sree Narayana College, Cherthala & verified by :

Date: Internal Examiner External Examiner

"Remembering and thanking all the individuals who aided and guided for the completion of this project..... "

DECLARATION

We here by declare that the project report entitled **"WebCharity"** is a record of project work done under the guidance of the department of Computer Science, Sree Narayana College, Cherthala.

We also declare that this report has not been submitted to any other University or Institute for the award of any fellowship or Degree or Diploma.

Place: Cherthala

Date:

Names: **ASWIN SURESH**
BINEESH BABU
PRANAV P
VISWAV S

ACKNOWLEDGMENT

This is the first time we got involved in a project for creating a Web application. We believe that the resultant software had very much matched our objective. On the course of this project, we received great aid and support from our friends, family, faculties etc. and we express our gratitude towards them.

We are esteemed to thank **Dr. T.P BINDU**, The Principal Incharge, Sree Narayana College, Cherthala for granting us permission to do the project.

We convey our sincere thanks to **Dr BINDU N (Head of the Department Computer Science** for their valuable instructions and guidance to the project and their apt encouragement towards the successful completion of our project.

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

ABSTRACT

The **WebCharity** project is a comprehensive charity management system consisting of three key modules: Admin, Charity, and Donor. This system is designed to streamline and enhance the process of charitable donations. The Admin module serves as the backbone, allowing administrators to oversee and manage the entire platform efficiently. They can verify and approve charities, monitor donations, and ensure the system's integrity. The Charity module empowers charitable organizations to create profiles, share their mission, and raise funds for their causes. It provides them with a platform to connect with potential donors and showcase their impactful work. The Donor module enables individuals to contribute to their chosen charities securely and conveniently. Donors can browse verified charities, make donations, and receive tax receipts, all within the system. WebCharity aims to facilitate the spirit of giving by providing a user-friendly, transparent, and efficient platform. Through this system, donors can make a difference in the world, contributing to a better, more compassionate society.

CONTENTS

List of Figures

12.1 Gantt Chart

List of Tables

1. INTRODUCTION

1.1 EXISTING SYSTEM AND DISADVANTAGE

The existing system of the WebCharity project is often characterized by manual, time-consuming processes, and a lack of centralized management. Charitable organizations struggle to reach donors effectively, relying on traditional fundraising methods like events and direct mail campaigns.

1.2 PROPOSED SYSTEM AND ADVANTAGES

The proposed system provides a website where all types of users like Admin, Charity and Donors can use the system to satisfy their requirements.

Efficiency: The system streamlines the donation process, reducing paperwork and processing time.

Transparency: Donors can see the impact of their contributions in real-time, fostering trust.

Convenience: Donors can make contributions from anywhere, at any time, through a user-friendly

Increased Outreach: Charities can reach a broader donor base, expanding their resources strategies.

Data Insights: The system collects valuable data for administrators and charities to improve their

Security: Robust security measures protect sensitive donor and charity information.

Community Building: Donors can connect with like-minded individuals

1.3 PROJECT PROFILE

Title : WebCharity.

Type : Python ,Django.

Objective :Develop a Charity System.

Duration : 3 months.

Internal Guide : BINDU N.

Project Team : ASWIN SURESH

BINEESH BABU.

PRANAV P.

VISWAV S.

1.4 ORGANIZATION PROFILE

Sree Narayana College, Cherthala is affiliated to University of Kerala and approved by NAAC and UGC. Sree Narayana College, Cherthala was established By Sree Narayana Trusts, Kollam. The aim of college at assessing and accrediting institution of higher learning with and objective of improving quality of education especially of the backward communities of the locality.

1.5 PROJECT OVERVIEW

WebCharity is an innovative online platform designed to revolutionize charitable giving. It offers a secure and user-friendly environment where charitable organizations can connect with donors. The system includes an Admin module for centralized control, a Charity module for organizations to showcase their mission, and a Donor module for easy, secure contributions. WebCharity enhances transparency, streamlines the donation process, and fosters a sense of community among donors.

With robust security and real-time reporting, it empowers both charities and donors to make a meaningful impact in the world of philanthropy

2. REQUIREMENT ENGINEERING

Requirement engineering provides the appropriate mechanism for understanding what the customer needs, analyzing type of workers, assessing the feasibility, negotiating reasonable solutions unambiguously validating the specification and managing the requirements as they are transformed into an operational system.

2.1 REQUIREMENT ELICITATION

Requirement's elicitation or requirements gathering focuses on the objectives of the systems

2.1.1 Feasibility Study

The main objective of this study is to determine whether the proposed system is feasible or not. System analysis is most important phase in the lifecycle of the system development. The investigation points to question whether the project is feasible. A feasibility is conducted to identify the best system that meets the all requirements. This includes an identification description, a valuation of the proposed system and selection of the best system for the job. The requirement of the system is specified with a set of constrains such as system objective and the description of the output. It is then duty of the analyst to evaluate the feasibility of the proposed system to generate the above result. Three key factors are to be considered during the feasibility study.

2.1.1.1 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised

2.1.1.2 Feasibility Study

The main considerations are given to the study of available resources of the organisation where the software is to be implemented. Here the system analyst evaluates the technical merits of the system given emphasis on the performance reliability, maintainability and productivity. By taking the consideration before developing the proposed system, the resource availability of the organisation was studied. The organisation has immense computer facilities equipped with sophisticated machine and software hence this is technically feasible.

2.1.1.3 Economical feasibility

Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. Cost benefit analysis is usually performed for his purpose. It is a comparative study of the cost verses the benefit and savings that are expected from the proposed system. Since the organisation is well equipped with the required hardware, the project was found to be economically feasible.

2.1.2 Requirement Definition

WebCharity is an innovative online platform designed to revolutionize charitable giving. It offers a secure and user-friendly environment where charitable organizations can connect with donors. The system includes an Admin module for centralized control, a Charity module for organizations to showcase their mission, and a Donor module for easy, secure contributions. WebCharity enhances transparency, streamlines the donation process, and fosters a sense of community among donors.

With robust security and real-time reporting, it empowers both charities and donors to make a meaningful impact in the world of philanthropy

2.1.2.1 Functional Requirements

Functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output datasets

- Admin
    1. Login
    2. Verification of Charity
    3. Verification of Campaign
    4. View Donations
    5. View Users
- Charity
    1. Register
    2. Login
    3. Request for campaign
    4. View Donations
- Donor
    1. Register
    2. Login
    3. View Campaigns
    4. Give Donations

2.1.2.2 Non-Functional Requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within. The non-functional requirements include performance requirements. There are two types of performance requirements- static and dynamic. Static requirement includes a number of terminals supported, number of simultaneous users to be supported, number of file system has to process and their sizes etc. Dynamic requirements include execution time behaviour of the system such as throughput, response time, expected time for completion of operation etc. The non-functional requirements include design constraints, logical database requirements, and standard compliance and so on.

The non-functional requirements which we look forwarded are: -

a) **Maintainability:** Maintainability is the important part of the non-functional requirements. Our project is easily maintainable and there is no extra cost to maintain it.

b) **Portability:** The size of the project is very small, so it's very portable and can be easily taken from one place to another place even in small storage devices.

3. DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

- INPUT DEVICE : MOUSE, KEYBOARD
- OUTPUT DEVICE : MONITOR
- MEMORY : 4 GB RAM (MINIMUM)
- PROCESSOR : 11^{th} Gen Intel(R) Core(TM) i5-1155G7

3.2 SOFTWARE REQUIREMENTS

- OPERATING SYSTEM : WINDOWS 11 Home Single Language
- FRONT END : Django
- WEB : HTML
- BACK END : MySql Server
- BROWSER : Google Chrome
- WEBSERVER : Django Server
- CODE EDITOR : Visual Studio Code

3.2.1 About Windows

Microsoft Windows (or simply **Windows**) is a met family of graphical operating systems developed, marketed, and sold by Microsoft. It consists of several families of operating systems, each of which cater to a certain sector of the computing industry.

Microsoft introduced an operating environment named *Windows* on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). soft Windows came to dominate the world's personal computer market with over 90% market share, overtaking Mac OS.

As of September 2022, the most recent version of Windows is Windows 11for consumer PCs and tablets, Windows 11 Enterprise for corporations, and Windows Server 2022 for servers. The most recent versions for server computers and embedded are respectively Windows Server 2012 R2 and Windows Embedded 8. A specialized version of Windows runs on the Xbox One game console.

All Windows versions from Windows NT 3 have been based on a file system permission system referred to as AGLP (Accounts, Global, Local, Permissions) AGDLP which in essence where file permissions are applied to the file/folder in the form of a 'local group' which then has other 'global groups' as members. These global groups then hold other groups or users depending on different Windows versions used

3.2.2 MySQL

MySQL, the most popular open-source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and ―pointers‖ between different tables. The database enforces these rules, so that with a well - designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of ―MySQL‖ stands for ―Structured Query Language‖. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

MySQL software is Open Source.

Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application.

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked.

MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backend, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

A large amount of contributed MySQL software is available.

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favourite application or language supports the MySQL Database Server.

3.2.3 DJANGO

Django is a high-level, open-source web framework for building web applications using Python. It is known for its simplicity, flexibility, and emphasis on rapid development. Some of the key features and characteristics of Django include

Advantages of Django

1. MVC Architecture (Model-View-Controller).

Django follows the MVC architectural pattern, where Models define the data structure, Views handle the presentation logic, and Controllers (handled by Django itself) manage the communication between Models and Views.

2. Extremely Flexible

Django is highly flexible whether it is during an ongoing project or after completing the project. Flexibility in a scripting language is very crucial, as functionality can change anytime during the course of a project. The best part about Django is the ability to make changes even after starting the project and this saves valuable time. A developer does not have to write fresh codes or command functions, as changes to the existing code and functions can be done and used.

3. Easy Integration and Compatibility

Django is compatible with a large majority of operating systems. It can easily run on different platforms, including UNIX, Solaris, and Linux. As it can be integrated without effort with

Other technologies, such as Java, existing software does not require re-development. This Saves time and money.

4. Efficient Performance

Depending on how the web developer codes, Django has the potential to turn in an efficient language. It is scalable when used for writing codes and can also be used for creating a large number of applications. It is the programming language of choice when a website has several web pages.

5. Cost-Efficient

Django is an open-source web language, hence is completely free. There is no expense involved in purchasing expensive licenses or software. It can work efficiently with different databases, such as MySQL, Apache, and PostgreSQL. The cost of developing a website using Django is minimal.

6. Gives Web Developer More Control

Compared to other programming languages, Django allows the website developer to have more control. Other programming languages are bogged down by long, complicated scripts, but this isn't true for Django. A few simple lines of code are sufficient. Furthermore, Django allows tags, and hence, website developers can add and/or mix HTML tags, making the content extremely dynamic.

Developers don't have to worry about placing codes in the right place when using Django, as it is written between tags. Hence, functions and codes do not have to be written in any specific order, as long as they are within the tags.

3.2.4 Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected.It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980's, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.1. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward- compatible and much Python 2 code does not run unmodified on Python 3. Python consistently ranks as one of the most popular programming languages.

Advantages of Python

Following is the list of other advantages of using Python:

- **Easy to Read, Learn and Write:** Python is a high-level programming language that has English-like syntax. This makes it easier to read and understand the code. Python is really easy to pick up and learn, that is why a lot of people recommend Python to beginners. You need less lines of code to perform the same task as compared to other major languages like C/C++ and Java.
- **Improved Productivity:** Python is a very productive language**.** Due to the simplicity of Python, developers can focus on solving the problem. They don’t need to spend too much time in understanding the syntax or behaviour of the programming language. You write less code and get more things done.
- **Interpreted Language:** Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred. Python shows only one error even if the program has multiple errors. This makes debugging easier.
- **Dynamically Typed:** Python doesn’t know the type of variable until we run the code. It

automatically assigns the data type during execution. The programmer doesn't need to worry about declaring variables and their data types.

- **Free and Open-Source:** Python comes under the OSI approved open-source license. This makes it free to use and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behaviour and use their version for development.
- **Vast Libraries Support:** The standard library of Python is huge; you can find almost all the functions needed for your task. So, you don't have to depend on external libraries. But even if you do, a Python package manager (pip) makes things easier to import other great packages from the Python package index (PyPi). It consists of over 200,000 packages.
- **Portability:** In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

3.2.5 VS Code

VS code is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License. Based on the powerful Scintilla editing component, VS code is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness.

The following is a list of features of Notepad++:

- Syntax highlighting and Syntax Folding
- PCRE(Perl Compatible Regular Expression) Search/Replace
- GUI entirely customizable
- Auto completion: Word completion, Function completion and Function parameters Hint
- Multi-document(Tab interface)
- Macro Recording and playback
- Multilanguage Environment Supported

VS code was first released on Source Forge on 25 November 2003, as a Windows-only application. It is based on the Scintilla editor component, and is written in C++ with only Win32 API calls using only the STL to increase performance and reduce program size.

3.2.6 Web Server

In the context of web development with Python, Django includes a built-in development web server for local testing and debugging. To initiate this server, developers use the "python manage.py runserver " command within their Django project directory. This convenient tool is invaluable during the development phase, providing a quick and straightforward way to preview the web application on a local machine. However, it's crucial to note that the Django development server is not suitable for production deployments due to its limited scalability and security features. For production, deploying Django applications typically involves using production-ready web servers like Apache, Nginx, or Gunicorn.

4. SOFTWARE DESIGN

4.1 SYSTEM DESIGN

System design is the process of defining the architecture, components, interface and data for a system to satisfy specified requirements. Object-oriented analysis and design methods are the most widely used methods for computer system design. It translates the system requirement into ways of making them operational. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. Design goes through logical and physical stages of development. The characters of well-defined system are:

1. Security
2. Practicality
3. Efficiency
4. Acceptability
5. Flexibility
6. Economy
7. Reliability
8. Simplicity

4.1.1 Data Flow Diagram (DFD)

The data flow diagram is a way of expressing system requirements in a graphical form. This led to the modular design. A data flow diagram also known as a "Bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become program in system design.

A Data Flow Diagram (DFD) or a Bubble Chart describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by using a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output data-flows which go to other processes or external entities or files. Data in files many also flow to processes as inputs.

DFD's can be hierarchically organized, which help in partitioning and analyzing large systems. As a first step, one Data Flow Diagram can depict an entire system which gives the system overview. It is called Context Diagram of level 0 DFD.

The Context Diagram can be further expanded. The successive expansion of a DFD from the context diagram to those giving more details is known as levelling of DFD. Thus, a top-down approach is used, starting with an overview and then working out the details

The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformation of data is done, what files are used, and where the results flow.

Basic DFD Symbols:

Data flow is a route, which enables data to travel from one point to another. Data may flow from a source to a data store or process. An arrow line depicts the flow, with arrowhead pointing in the direction of the flow.

A process represents transformation where incoming data flows are changed into outgoing data flows. A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational File, they should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double - headed arrow is used.

Symbols used in DFD

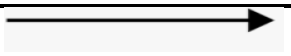
	*External Entity*
	*Process*
	*Data Store*
	*Data Flow*
	*Output*

Table 4.1.1.1

A source or sink is a person or part of an organization, which enters or receives information from the system, considered to be outside the contest of data flow model.

Level-0 Context Diagram

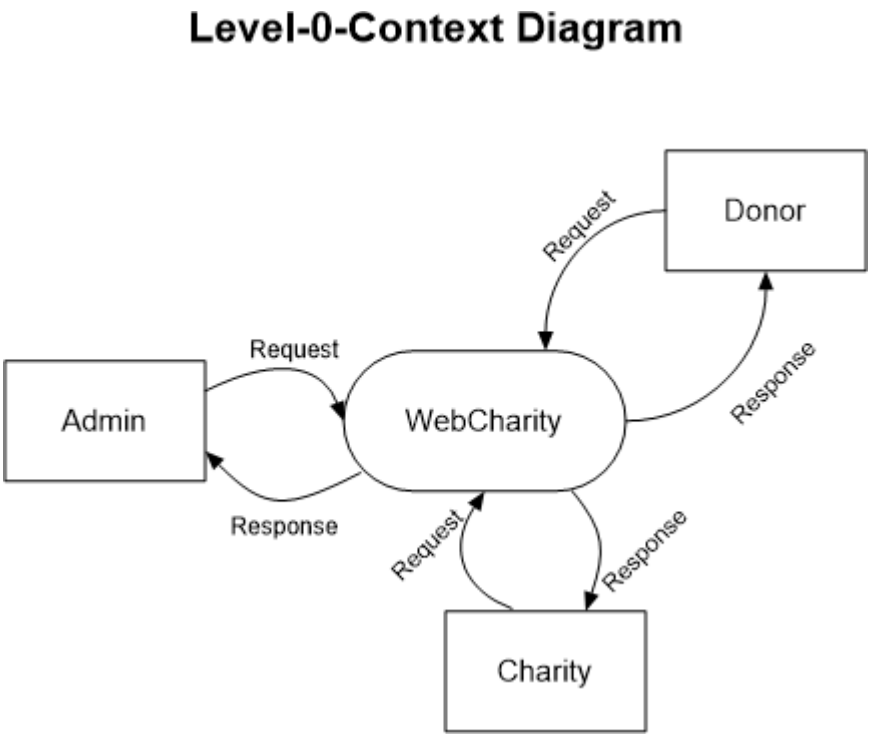


Figure 4.1.1.1

Level-1 Admin

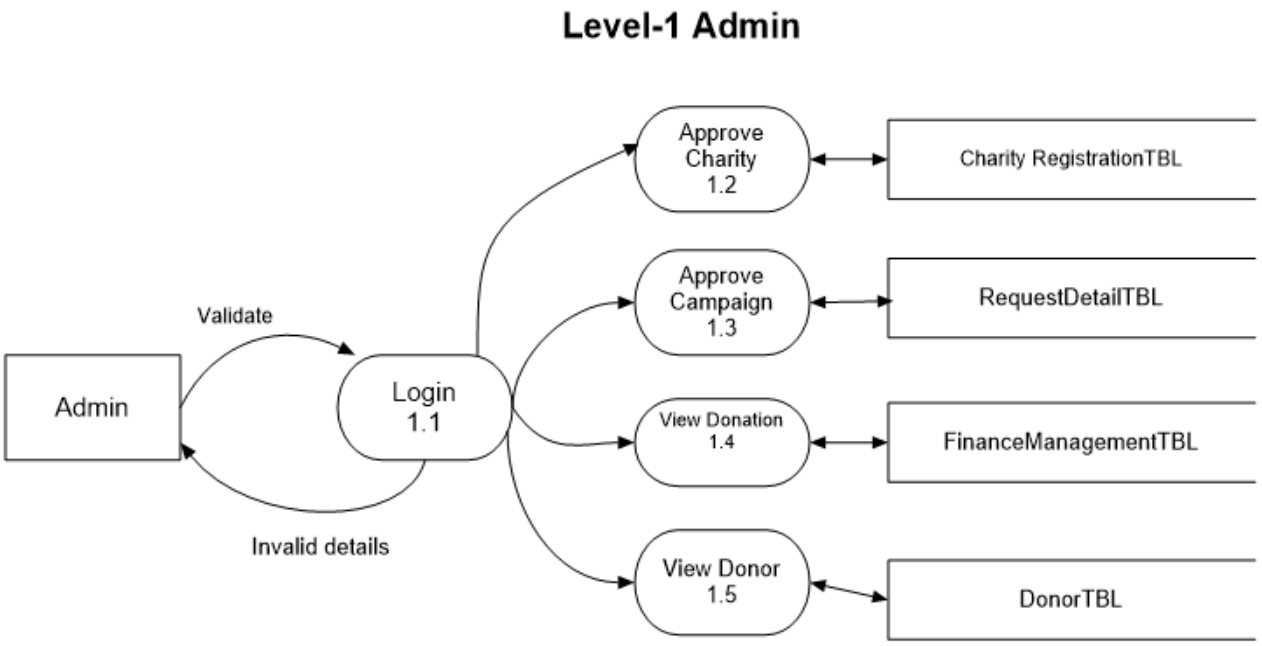


Figure 4.1.1.2

Level-1.1 Charity

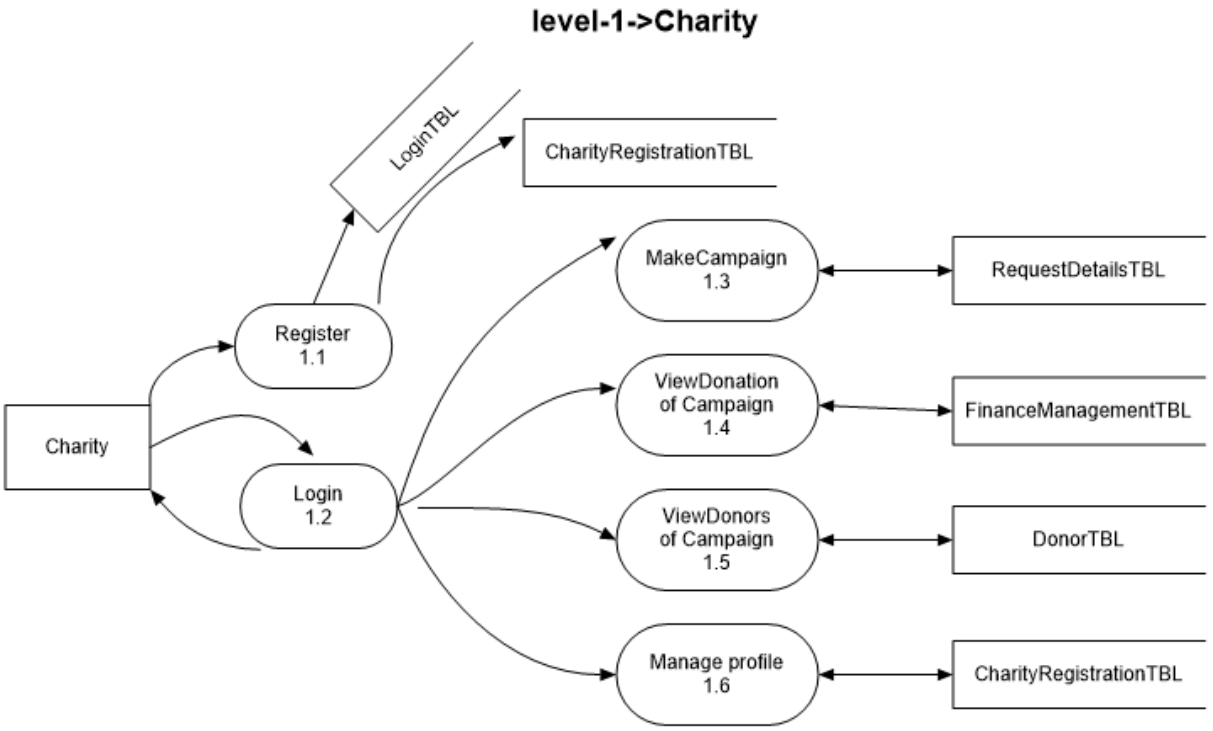


Figure 4.1.1.3

Level-1.2 Donor

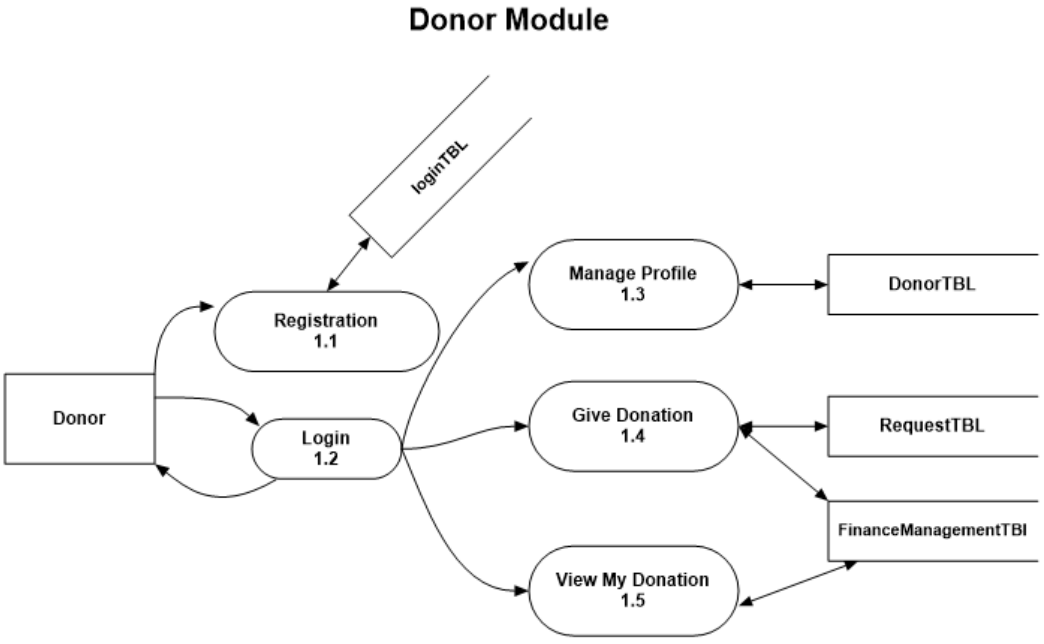


Figure 4.1.1.4

4.1.2 Entity Relationship Diagram (E-R Diagram)

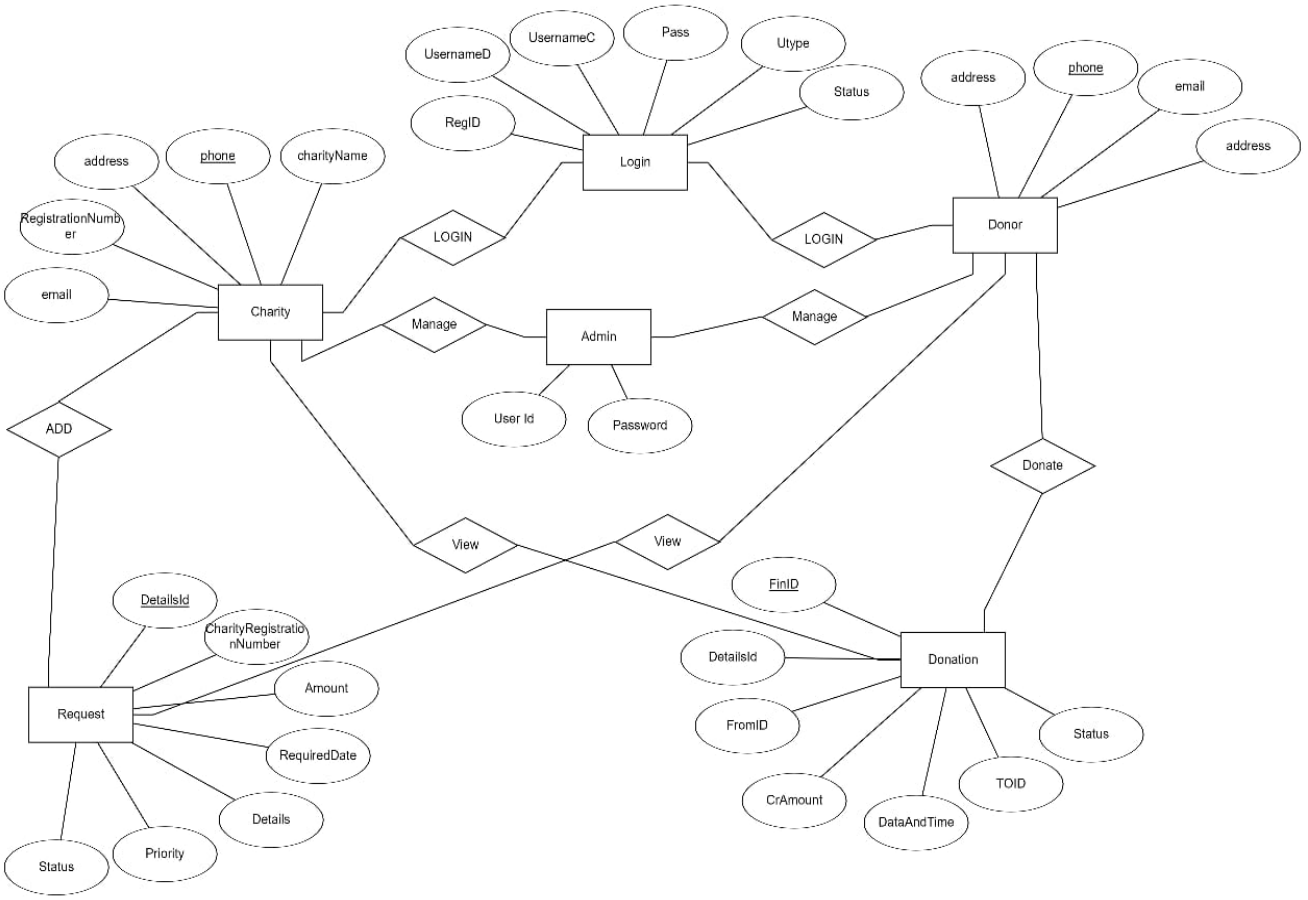


Figure 4.1.2

4.1.3 Usecase Diagram

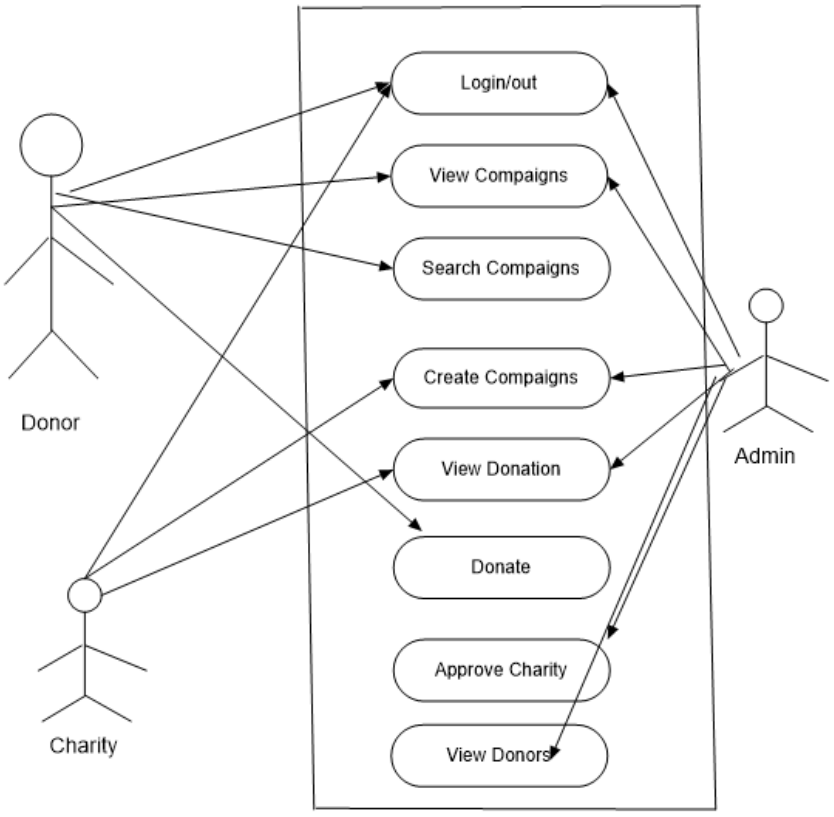


Figure 4.1.3

4.1.4 Activity Diagram

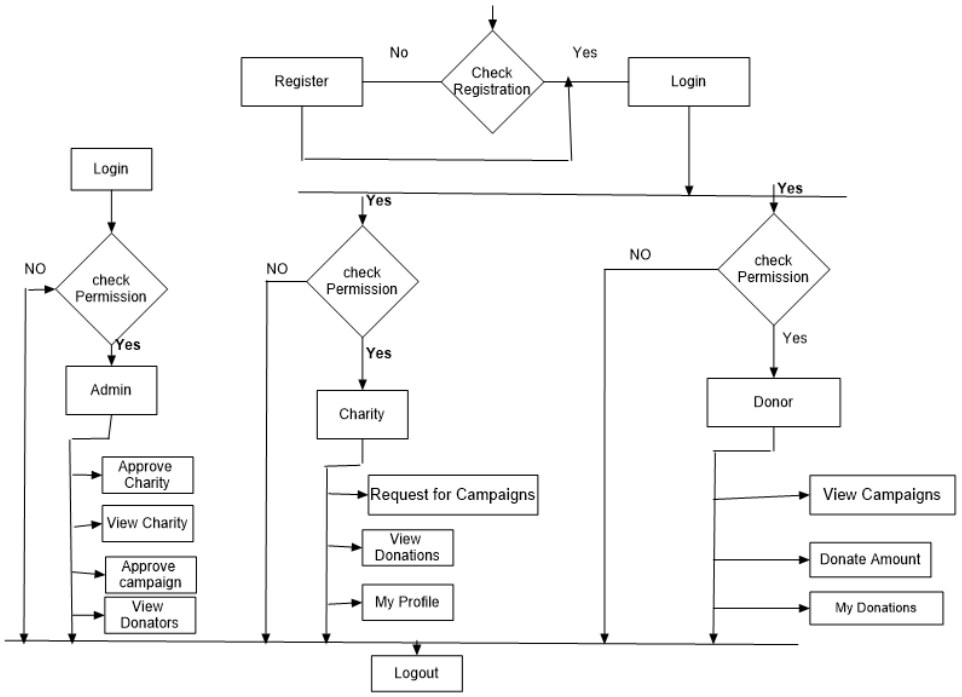


Figure 4.1.4

4.1.5 Sequence Diagram

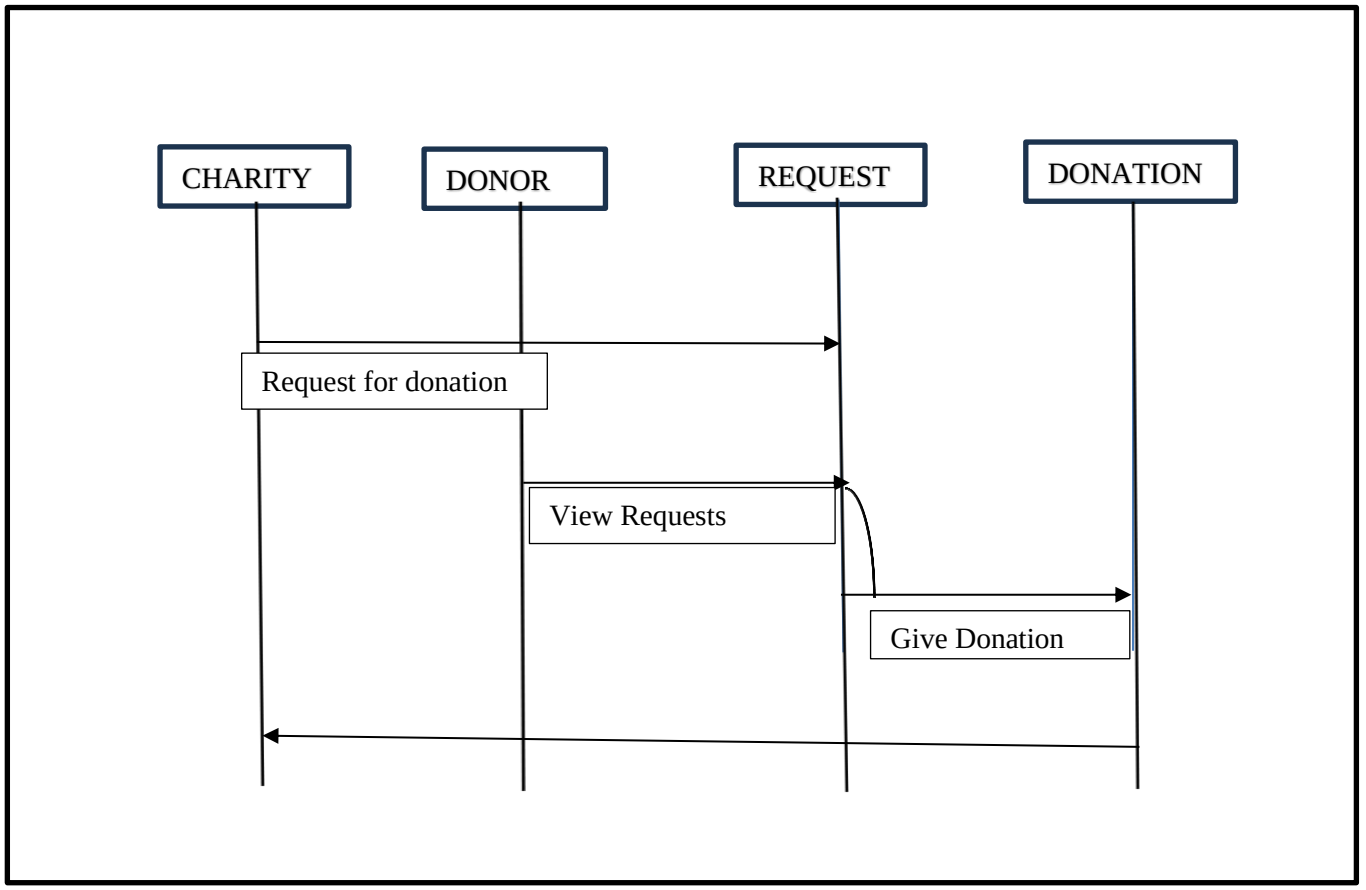


Figure 4.1.5

Table Name :- CharityRegistrationTBL
P.K :- phone

Name	Type	Size	Constraints	Description
phone	**Varchar**	**13**	**Primary Key**	**Phone Number**
charityName	**Varchar**	**25**	**Not null**	**Charity Name**
email	**Varchar**	**35**	**Not null**	**E-mail**
RegistrationNumber	**Varchar**	**8**	**Allow Null**	**Registration Number**
address	**Varchar**	**50**	**Not null**	**Address**

Table Name :- LoginTBL
P.K :- RegID

Name	Type	Size	Constraints	Description
RegID	**Int**	**5**	**Priamry Key**	**Registration ID**
UsernameC	**Varchar**	**13**	**Foreign Key, Allow null**	**Phone Number Charity**
UsernameD	**Varchar**	**13**	**Foreign Key, Allow null**	**Phone Number Donor**
Pass	**Varchar**	**16**	**Not null**	**Password**
Utype	**Int**	**1**	**Not null**	**User Type**
Status	**Int**	**1**	**Not null**	**Status**

Table 4.2

Name :- DonorTBL
P.K :- phone

Name	Type	Size	Constraints	Description
phone	**Varchar**	**13**	**Primary Key**	**Phone Number**
Name	**Varchar**	**25**	**Not null**	**Name**
email	**Varchar**	**35**	**Not null**	**E-mail**
address	**Varchar**	**50**	**Not null**	**Address**

Table Name :- RequestDetailsTBL
P.K :- DetailsId

Name	Type	Size	Constraints	Description
DetailsId	**Int**	**5**	**Primary Key**	**Request ID**
CharityRegistrationNumber	**Varchar**	**13**	**Foreign Key**	**Charity phone number**
Amount	**Money**		**Not null**	**Amount**
RequiredDate	**Date**		**Not null**	**Date**
Details	**Varchar**	**100**	**Not null**	**Description**
Priority	**Int**	**1**	**Not null**	**Priority**
Status	**Int**	**1**	**Not null**	**Status**

Table Name :- FinanceManagementTBL

P.K :- FinID

Name	Type	Size	Constraints	Description
FinID	**Int**	**5**	**Primary Key**	**Transation ID**
DetailsId	**Int**	**1**	**Foreign Key**	**Request ID**
FromID	**Varchar**	**13**	**Foreign Key**	**Donor Phone**
ToID	**Varchar**	**13**	**Foreign Key**	**Reciver Phone**
CrAmount	**Money**		**Not null**	**Amount**
DataAndTime	**Date**		**Not null**	**Date**
Status	**Int**	**1**	**Not null**	**Status**

4.2 INPUT DESIGN

As the person login as the admin with his username and password, he can verify the registered farmers, researchers, horticulture units and the shops. Without his approval the users can't perform any functions in the website. Farmers can login to system by entering their respective username and password and also a new farmer can be registered by entering the respective details mentioned. After logged in, the farmers can view faq, ask queries to researchers, view available horticulture units list, view available shop lists, view answers given by the researchers and find the most suitable crop for their cultivation. Researchers can login in to the system by entering username and password and can view the queries of farmers and reply to them and they can be registered by entering the respective details mentioned. Horticulture units can login in to the system by entering username and password and can edit available seeds, crops, fertilizers in their store and they can be registered into the system by entering the respective details. Also, the shops can login into system by entering username and password and can edit the available equipments in their store and can be registered into the system by entering the required details.

4.3 OUTPUT DESIGN

The outputs received by the modules are:

For the farmers, researchers, horticulture units, shops, as they registers to the site, they will get a username and password. The username and password are generated as they register in the website. Using this when a farmer login, he is directed to farmer logged in page where he can ask questions to researchers, find the most suitable crop for their land, view registered horticulture units and shops. If a Researcher is login in then he will be directed to the researchers' home page where he can view and reply to the queries put forward by the farmers. If a horticulture unit is logged in, then they will be directed to the horticulture logged in home page where they can edit the availability of crops, seeds, fertilizers they sell. If a shop owner is logged in then he will be directed to the shop logged in home page where he acan edit the availability of equipment's in his store. If the admin is the person who is signing in, he will have the user id and password which is built in. when he enters the username and password, he will enter into the administrator navigation page. Here he can view and verify the newly registered farmers, researchers, horticulture units as well as shops.

4.4 INTERFACE DESIGN

User interface design or user interface engineering is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centred design. Where good graphic/industrial design is bold and eye catching, good user interface design is to facilitate finishing the task at hand over drawing attention to it. Graphic design may be utilized to apply a theme or style to the interface without compromising its usability. The design process of an interface must balance the meaning of its visual elements that confirm the mental model of operation, and the functionality from a technical engineering perspective, in order to create a system that is both usable and easy to adapt to the changing user needs. In our system we provide effective and user-friendly interfaces.

5. CODING

Views.py

```
from sqlite3 import IntegrityError
from django.shortcuts import get_object_or_404, render
from django.shortcuts import render,redirect
from django.http import HttpResponse,JsonResponse, response
from .models import *
from django.views.decorators.csrf import csrf_exempt
import os
from django.conf import settings
from django.http import FileResponse



######## HOME ########################################



@csrf_exempt

def Index(request):
    request.session.flush()
    return render(request,'index.html')


@csrf_exempt

def forgotPassword2(request):
    uname=request.GET.get("uname")
    pswrd=request.GET.get("pswrd")
    Utype=request.GET.get("Utype")
    uid=request.GET.get("uid")
    try:
        if int(Utype)==2:
            ob=CharityRegistrationTBL.objects.get(phone=uid,charityName=uname)
            oj=LoginTBL.objects.get(RegID=ob.phone,Username=ob.phone,Utype=2)
            oj.Pass=pswrd
            ob.save()
            data={"status":1}
            return JsonResponse(data,safe=False)
        elif int(Utype==3):
            ob=Donor.objects.get(phone=uid,Name=uname)
            oj=LoginTBL.objects.get(RegID=ob.phone,Username=ob.phone,Utype=3)
            oj.Pass=pswrd
```

```
            ob.save()
            data={"status":1}
            return JsonResponse(data,safe=False)
    except Exception as e:
        data={"status":0}
        return JsonResponse(data,safe=False)

@csrf_exempt

def forgotPassword(request):
    return render(request,'forgotPassword.html')


@csrf_exempt

def about(request):
    return render(request,'about.html')



@csrf_exempt

def CharityReg(request):
    return render(request,'CharityReg.html',{})

@csrf_exempt

def DonorReg(request):
    return render(request,'DonorReg.html',{})

@csrf_exempt

def Login(request):
    return render(request,'login.html')
@csrf_exempt

def adminhome(request):
    return render(request,'admin/adminhome.html',{})

@csrf_exempt

def charityhome(request):
    ph=request.session['UserName']
    ob=CharityRegistrationTBL.objects.get(phone=ph)
    return render(request,'charity/charityhome.html',{"Name":ob.charityName})

@csrf_exempt

def donorhome(request):
    raised=0
    num=0
```

```
    selected_rows = FinanceManagement.objects.filter(FromID=request.session['UserName'])
    for i in selected_rows:
        raised=raised+int(i.CrAmount)
        num=num+1
    return render(request,'donor/donorhome.html',{"part":num,"amt":raised})


@csrf_exempt

def CheckLogin(request):
    uname=request.GET.get("uname")
    pswrd=request.GET.get("pswrd")
    Utype=request.GET.get("Utype")
    print(uname,pswrd)
    if(uname=="admin" and pswrd=="admin" and Utype=="1"):
        data={"status":1}
        return JsonResponse(data,safe=False)
    else:
        try:
            ob=LoginTBL.objects.get(Username=uname,Pass=pswrd,Utype=Utype)
            request.session['UserName'] =ob.RegID
            if ob.Utype==2:
                data={"status":2}
                return JsonResponse(data,safe=False)
            if ob.Utype==3:
                data={"status":3}
                return JsonResponse(data,safe=False)
        except Exception as e:
           data={"status":0}
           return JsonResponse(data,safe=False)


@csrf_exempt

def RegCharity(request):
    name=request.POST.get("name")
    phone=request.POST.get("phone")
    registration=request.POST.get("registration1")
    email=request.POST.get("email")
    address=request.POST.get("address")
    password=request.POST.get("password")
    data={}
    print(registration)
    if registration==None:
        registration="NON CHARITY"
    print(registration)
    try:
        CharityRegistrationTBL.objects.create(phone=phone,charityName=name,email=email,Regi
strationNumber=registration,address=address)
        LoginTBL.objects.create(RegID=phone,Username=phone,Pass=password,Utype=2,Status=0)
        data={"msg":"Successfully Registread"}
```

```
    except IntegrityError:

        data={"msg":"User Already Exist"}

        print(data)
        return render(request,'CharityReg.html',data)
    except Exception as e:
        data={"msg":"Submission failed"}
        print(e)
        return render(request,'CharityReg.html',data)

    return render(request,'CharityReg.html',data)



@csrf_exempt

def RegDonor(request):
    Name=request.POST.get("name")
    phone=request.POST.get("phone")
    email=request.POST.get("email")
    address=request.POST.get("address")
    password=request.POST.get("password")
    data={}
    try:
        Donor.objects.create(Name=Name,phone=phone,email=email,address=address)
        LoginTBL.objects.create(RegID=phone,Username=phone,Pass=password,Utype=3,Status=1)
        data={"msg":"Successfully Registread"}
    except IntegrityError:

        data={"msg":"User Already Exist"}
        return render(request,'DonorReg.html',data)
    except Exception as e:
        data={"msg":"Submission failed"}
        print(e)
        return render(request,'DonorReg.html',data)

    return render(request,'DonorReg.html',data)



########################################  ADMIN
#############################################################################################
########################



@csrf_exempt

def ViewUser(request):
```

```
    data={}
    ob= Donor.objects.all()
    data1={}
    datalist=[]
    for i in ob:
        user_data = {
            "name": i.Name,
            "phone": i.phone
        }
        datalist.append(user_data)
    data["data"] = datalist
    print(data)
    return render(request,'admin/ViewDonor.html',data)



@csrf_exempt

def ViewUser2(request):
    phone_number = request.GET.get('phone')
    ob=Donor.objects.get(phone=phone_number)
    user_data = {
        "name": ob.Name,
        "phone": ob.phone,
        "email": ob.email,
        "address":ob.address,
        }
    data={}
    obj= FinanceManagement.objects.filter(FromID=ob.phone)
    data1={}
    datalist=[]
    for i in obj:
        n=CharityRegistrationTBL.objects.get(phone=i.ToID)
        user_data1 = {
            "name": n.charityName,
            "date": i.DataAndTime,
            "amt": i.CrAmount,
            "TID": i.FinID,
            "PDF": "INVOICE"+str(i.FinID),
        }
        datalist.append(user_data1)
    user_data["user"]=datalist
    return render(request,'admin/ViewDonor2.html',user_data)



@csrf_exempt

def ApproveCharity(request):
    try:
        data={}
```

```
        ob= CharityRegistrationTBL.objects.all()
        data1={}
        datalist=[]
        for i in ob:
            if LoginTBL.objects.filter(RegID=i.phone,Status='0',Utype='2'):
                user_data = {
                    "name": i.charityName,
                    "Reg": i.phone
                }
                datalist.append(user_data)
        data["data"] = datalist
        print(data)
        return render(request,'admin/ApproveCharity.html',data)
    except Exception as e:
        print(e)
        data={}
        return render(request,'admin/ApproveCharity.html',data)

@csrf_exempt

def ApproveCharity2(request):
    phone_number = request.GET.get('phone')
    user_data={}
    ob=CharityRegistrationTBL.objects.get(phone=phone_number)

    type=""
    if ob.RegistrationNumber=="NON CHARITY":type="For Personal Purpose"
    else: type="Charity"
    user_data = {
        "name": ob.charityName,
        "phone": ob.phone,
        "email": ob.email,
        "address":ob.address,
        "RegNumber":ob.RegistrationNumber,
        "Type": type,
        }

    return render(request,'admin/ApproveCharity2.html',user_data)


@csrf_exempt

def ApproveCharity3(request):
    P=request.POST.get('myInput')

    print(P)
    ob=LoginTBL.objects.get(Utype=2,RegID=P)
    ob.Status="1"
    ob.save()
    return render(request,'admin/ApproveCharity.html',{"msg":" APPROVED SUCESSFULLY "})
```

```
@csrf_exempt

def ApproveDonation(request):
    try:
        data={}
        ob= RequestDetailsTBL.objects.all()
        datalist=[]
        for i in ob:
            if i.Status==0:
                o=CharityRegistrationTBL.objects.get(phone=i.CharityRegistrationNumber)
                user_data = {
                    "name": o.charityName,
                    "Reg": i.DetailsId
                }
                datalist.append(user_data)
                data["data"] = datalist
        print(data)
        return render(request,'admin/ApproveDonation.html',data)
    except Exception as e:
        data={}
        return render(request,'admin/ApproveDonation.html',data)




@csrf_exempt

def ApproveDonation2(request):
    RequestID = request.GET.get('phone')
    user_data={}
    ob=RequestDetailsTBL.objects.get(DetailsId=RequestID)
    o=CharityRegistrationTBL.objects.get(phone=ob.CharityRegistrationNumber)
    type=""
    if o.RegistrationNumber=="NON CHARITY":type="For Personal Purpose"
    else: type="Charity"



    pdf_file_path1=RequestID
    file_path = str(pdf_file_path1) + '.pdf'
    pdf_file_path = os.path.join(settings.BASE_DIR, file_path)

    if os.path.exists(pdf_file_path):
        print("SSSSSSSSSS")
    print(file_path)
```

```
    if 'Pdf' in request.session:
        del request.session['Pdf']

    request.session['Pdf'] =pdf_file_path
    user_data = {
        "name": o.charityName,
        "phone": o.phone,
        "DetailsId": ob.DetailsId,
        "Amount":ob.Amount,
        "email":o.email,
        "RequiredDate":ob.RequiredDate,
        "Details":ob.Details,
        "address":o.address,
        "type":type,
        "pdf_file_path": file_path,
        }

    return render(request,'admin/ApproveDonation2.html',user_data)




@csrf_exempt

def ApproveDonation3(request):
    DetailID=request.POST.get('myInput')
    Amount=request.POST.get('amount')
    Priority=request.POST.get('Priority')
    if 'Approve' in request.POST.get('action'):
        ob=RequestDetailsTBL.objects.get(DetailsId=DetailID)
        ob.Priority=Priority
        ob.Amount=Amount
        ob.Status="1"
        ob.image='img'+str(DetailID)+'.jpg'
        ob.save()
        return render(request,'admin/ApproveDonation.html',{"msg":" APPROVED SUCESSFULLY
"})
    elif 'Cancel' in request.POST.get('action'):
        ob=RequestDetailsTBL.objects.get(DetailsId=DetailID)
        ob.Status="2"   # 2 denots the request is blocked permentently
        ob.save()

        return render(request,'admin/ApproveDonation.html',{"msg":" REMOVED SUCESSFULLY "})

    else:
        return render(request,'admin/ApproveDonation.html',{"msg":" Failed "})



@csrf_exempt
```

```
def pdfDonationViewAdmin(request):
    FIN=request.GET.get('FID')
    FIN="INVOICE"+str(FIN)+".pdf"
    print(FIN)
    return render(request,'admin/pdfDonationViewAdmin.html',{"PDF":FIN})
```

```
@csrf_exempt

def viewcharities(request):
    try:
        data={}
        ob= CharityRegistrationTBL.objects.all()
        data1={}
        datalist=[]
        for i in ob:
            if LoginTBL.objects.filter(RegID=i.phone,Status='1',Utype='2'):
                user_data = {
                    "name": i.charityName,
                    "Reg": i.phone
                }
                datalist.append(user_data)
        data["data"] = datalist
        print(data)
        return render(request,'admin/viewcharities.html',data)
    except Exception as e:
        data={}
        print(e)
        return render(request,'admin/viewcharities.html',data)
```

```
from django.db.models import Sum
@csrf_exempt

def viewcharities2(request):
    phone_number = request.GET.get('Reg')
    ob=CharityRegistrationTBL.objects.get(phone=phone_number)
    user_data = {
        "name": ob.charityName,
        "phone": ob.phone,
        "email": ob.email,
        "address":ob.address,
        "RegistrationNumber":ob.RegistrationNumber,
```

```
        }

    priority_mapping = {
    1: "High",
    2: "Medium",
    3: "Low"
    }

    data={}
    obj= RequestDetailsTBL.objects.filter(CharityRegistrationNumber=ob.phone)
    data1={}
    datalist=[]
    for i in obj:
        total_cramount =
FinanceManagement.objects.filter(DetailsId=i.DetailsId).aggregate(sum_cramount=Sum('CrAmoun
t'))['sum_cramount']
        if total_cramount is None:
            total_cramount = 0.0

        user_data1 = {
            "DetailsId": i.DetailsId,
            "Amount": i.Amount,
            "RequiredDate": i.RequiredDate,
            "Priority": priority_mapping.get(i.Priority, "Unknown"),
            "Status": "Active" if int(i.Status) == 1 else "Inactive",
            "raised":total_cramount

        }
        datalist.append(user_data1)
    user_data["user"]=datalist
    return render(request,'admin/viewcharities2.html',user_data)



@csrf_exempt

def DonationStatusAdmin(request):
    try:
        data={}
        ob= RequestDetailsTBL.objects.filter(Status=1)
        data1={}
        datalist=[]
        for i in ob:
            n=CharityRegistrationTBL.objects.get(phone=i.CharityRegistrationNumber)
            user_data = {
                    "name": n.charityName,
                    "Reg": i.DetailsId,
                    "Details":i.Details
            }
            datalist.append(user_data)
        data["data"] = datalist
```

```
        print(data)
        return render(request,'admin/DonationStatusAdmin.html',data)
    except Exception as e:
        data={}
        print(e)
        return render(request,'admin/DonationStatusAdmin.html',data)

@csrf_exempt

def DonationStatusAdmin2(request):
    DetailsId = request.GET.get('ReqID')
    priority_mapping = {
    1: "High",
    2: "Medium",
    3: "Low"
    }
    total_cramount =
FinanceManagement.objects.filter(DetailsId=DetailsId).aggregate(sum_cramount=Sum('CrAmount'
))['sum_cramount']
    if total_cramount is None:
            total_cramount = 0.0
    ob=RequestDetailsTBL.objects.get(DetailsId=DetailsId)
    o=CharityRegistrationTBL.objects.get(phone=ob.CharityRegistrationNumber)
    user_data ={
        "DetailsId": ob.DetailsId,
        "CharityRegistrationNumber": ob.CharityRegistrationNumber,
        "CharityName":o.charityName,
        "Amount": ob.Amount,
        "RequiredDate":ob.RequiredDate,
        "Details":ob.Details,
        "Priority":priority_mapping.get(ob.Priority, "Unknown"),
        "Status":"Active" if int(ob.Status) == 1 else "Inactive",
        "TotalC":total_cramount,
        "Shortage":int(ob.Amount)-int(total_cramount)

        }

    data={}
    obj= FinanceManagement.objects.filter(DetailsId=ob.DetailsId)
    data1={}
    datalist=[]
    for i in obj:

        n=Donor.objects.get(phone=i.FromID)
        user_data1 = {
            "FinID": i.FinID,
            "FromName": n.Name,
            "CrAmount": i.CrAmount,
            "DataAndTime": i.DataAndTime,

        }
```

```
        datalist.append(user_data1)
    user_data["user"]=datalist
    print(user_data)
    return render(request,'admin/DonationStatusAdmin2.html',user_data)
```

```
#####################################  CHARITY
#############################################################################################
#########################
```

6. SOFTWARE TESTING

Software testing is a critical element of software assurance and represents the ultimate review of specification, design and coding. Testing includes test case design and their execution. Testing demonstrates the software functions according to the specification.

Many types of testing are done in various stages of development to detect errors in the software and corrected. In testing the internal logic of the program, the inputs and the outputs are tested. Each test has different purpose, all works to verify that the system elements have been properly integrated and perform allocated functions.

TESTING OBJECTIVES

- Testing is the process of executing a program with intension of an error
- A good test case is one that has high probability of finding a yet undiscovered error
- Purpose of testing
- To affirm the quality of a product
- To find and eliminate any residual error
- To demonstrate the presence of all specified functions in the product
- To validate the software as a solution to the original problem.

In our system we performed the following testing:

6.1 TESTING STRATEGY

6.1.1 Unit Testing

The modules are tested separately. This is done during programming stage itself. Any logical errors found are corrected. Finally ensure that each module is working as expected. In our system there are four modules which we have successfully tested separately.

6.1.2 Integration Testing

After unit testing all modules are integrated and flow of information between the modules has been tested. This testing is done with sample data. There by find out the overall performance of the system. In our case study we have dependent modules which we have performed integration testing successfully and ensured the flow of information between modules as expected.

6.1.3 Data Validation Testing

After recovering all logical and interface errors, the software is completely assembled as a package. Then perform validation testing by inputting dummy data and ensure that it satisfies all the requirements of the user. It helps performance requirements and us to ensure that the software meets all functional behaviours.

6.1.4 System Testing

The chapter testing describes various testing methodologies which are adapted and detailed view of test data within each database. During testing of a program to be tested is executed with a set of test data and the output of the program for test data is evaluated.

In a software development project, errors can be introduced at any stage during development. The errors are detected after each phase by techniques like inspections, but some errors remain undetected. Ultimately, these remaining errors will be reflected in the code. Hence the final code is likely to have some requirement errors or design errors, in addition to errors introduced during the coding activity. Testing is the activity where the errors remaining from all the previous phases must be detected.

During testing, the software to be tested is executed with a set of test cases, and the behaviours of the system for the test cases is evaluated to determine if the system is performing as expected.

6.2 TEST CASES

Test Case Id	Test Id	Expected Output	Actual Output
1	Admin enters using username and password.	Get access to admin home page.	Login Successfully. Home page display.
2	Invalid username and password.	Redirect to confirm page and print error message	Redirect to confirm page and print error message
3	View and verify charity	View registered charity list and perform verification	Successfully done.
4	View and verify requests	View registered request list and perform verification	Successfully done.
5	View donors	View registered donors	Successfully done.
6	View donations	View all donations given by donors	Successfully done.
7	Click on logout button	Gets out of admin page and display admin logout page	Page displayed successfully
8	Charity enter username and password	Gets access to charity home page	Login successfully. Page displayed
9	If invalid username and password	Redirected to confirm page and an error message is displayed	Display error message referring to invalid entry.
10	Request for campaings	requested	Successfully displayed
11	View status of campaings	View status of campaings page will show	Successfully displayed
12	View all donations	Viewing all donations	Successfully done.
13	My profile section	Display the my profile page	Successfully done.

14	Click logout button	Gets out of the Charity home page and display farmer logout page	Page displayed successfully
15	Donor enter username and password	Get access to Donor home page	Logged in successfully. Page displayed.
16	If invalid username and password	Redirect to confirm page and display error message	Redirected to confirm page and message displayed successfully
17	View campaings	Display all donation requests	Successfully displayed
18	Donate button	View textbox for entering donation	Successfully done
19	Payment page	Payment page will load successfully	Successfully done
20	Click logout button	Gets out of the donor home page and display farmer logout page	Successfully done

7. IMPLEMENTATION

Implementation is a process of converting a new system into an operational one. The designed system is converted into an operational one using a suitable programming language. Implementation includes all those activities that take place to convert an old system into new one. Proper implementation is necessary to provide a reliable system to meet organizational requirement.

The implementation phase involves dedicated booking portal and constraints, to ensure a user-friendly platform for all the users to manage and to ensure the secure collection and dumping of waste without a third-party concern, admin has the super power to access all the functions and activities in this system. In this system, here user gets an instant status of waste collecting vehicle which they are looking for, by tracking vehicle. Through this system user can book and pay amount by online process.

8. SCREENSHOTS

Home Page

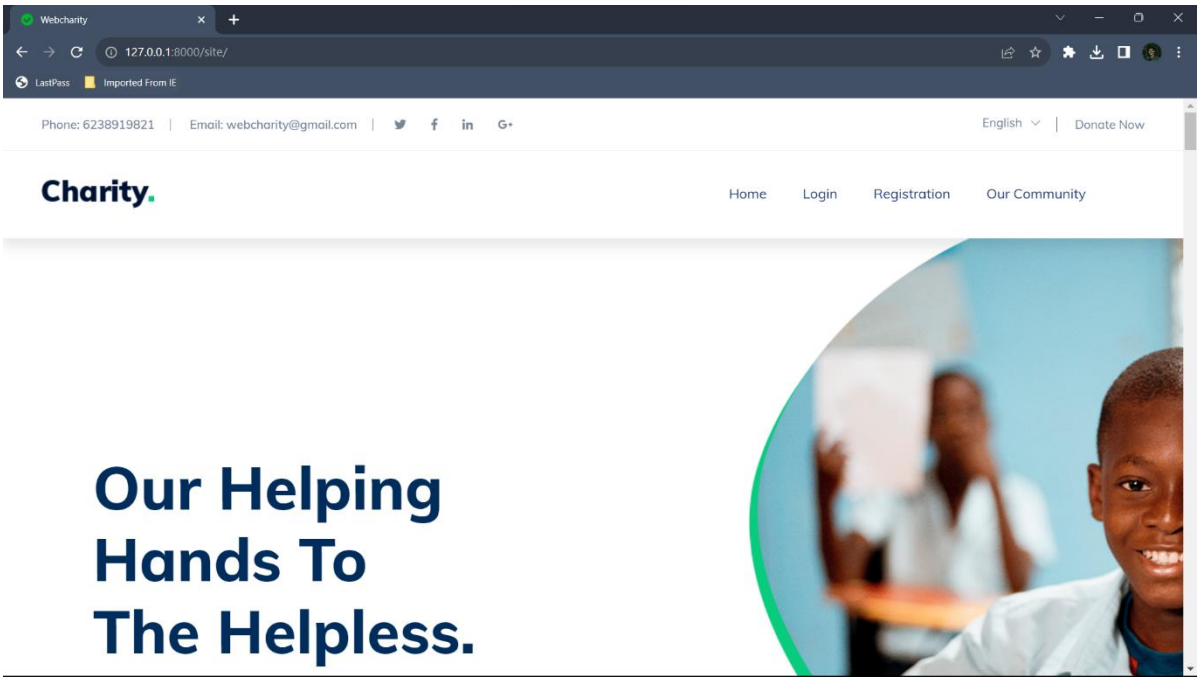


Figure 8.1

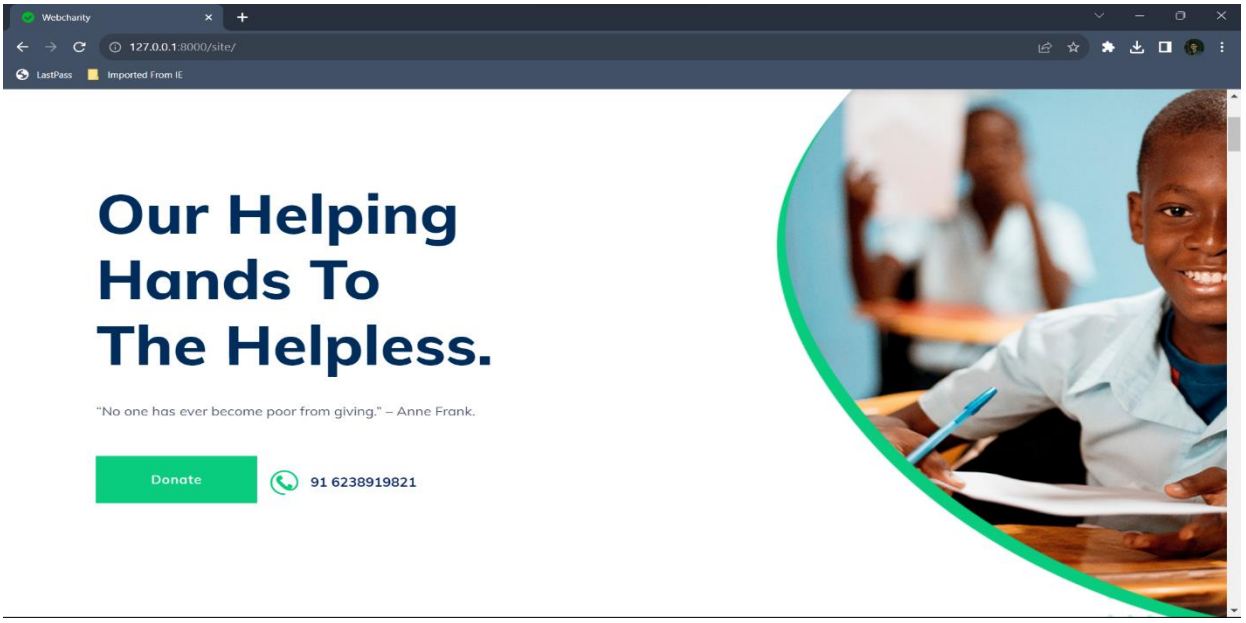


Figure 8.1.1

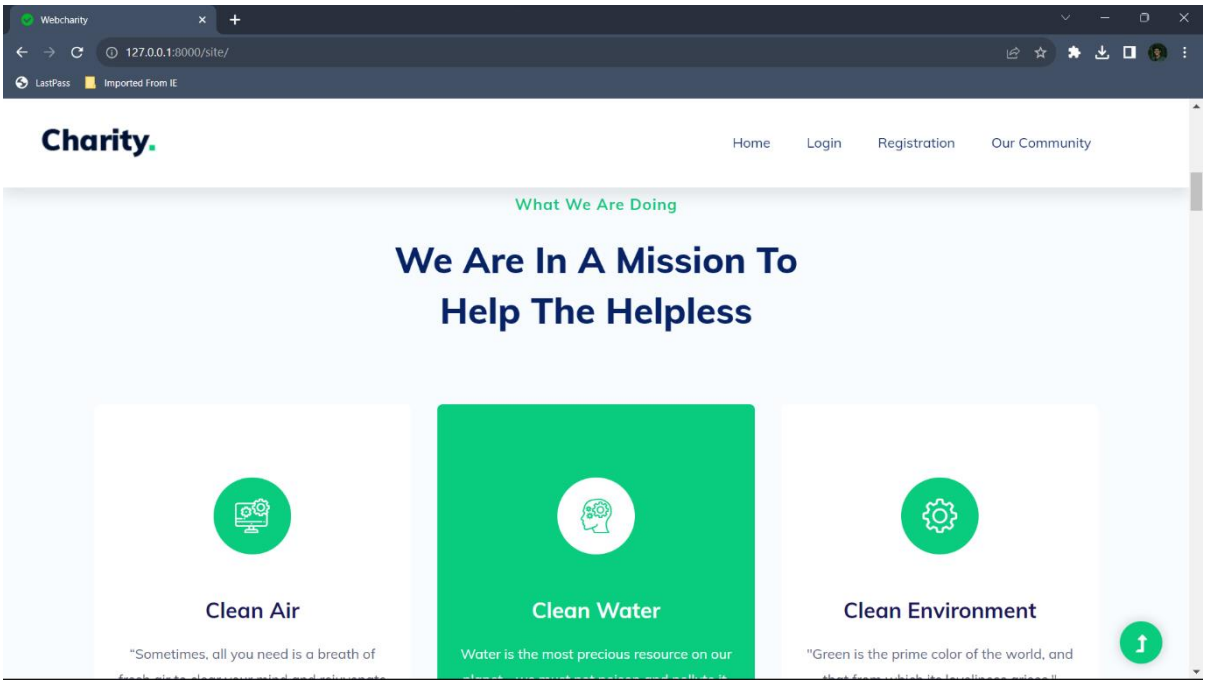


Figure 8.1.2

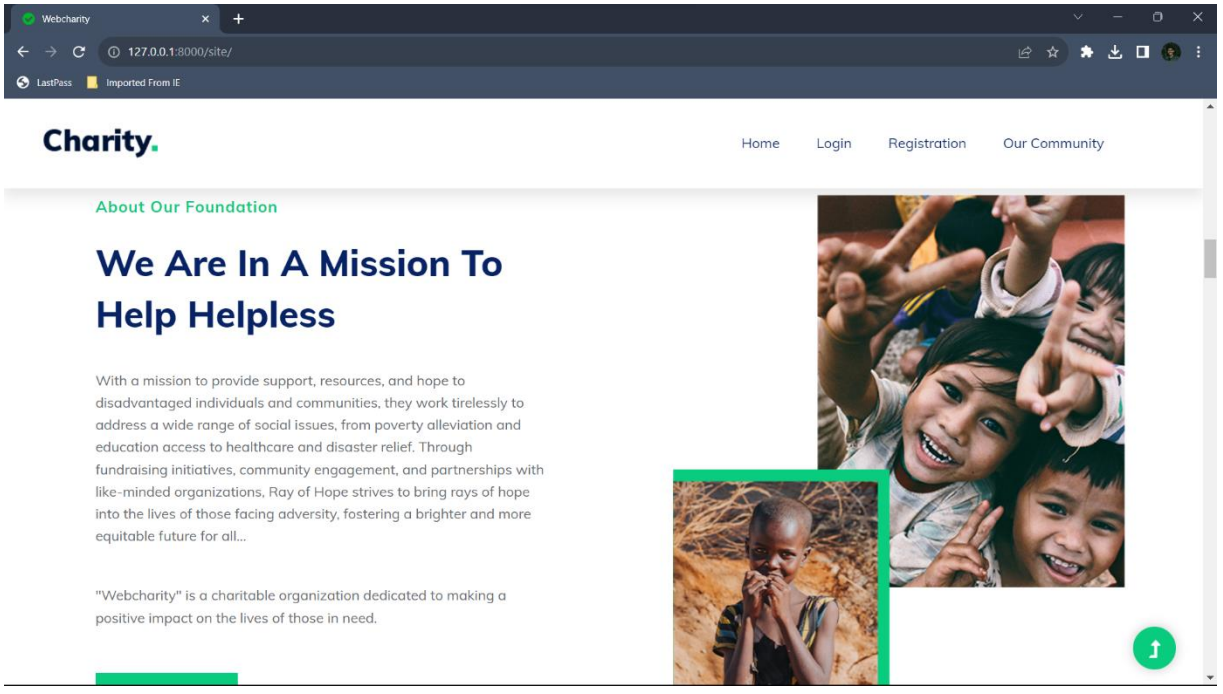


Figure 8.1.3

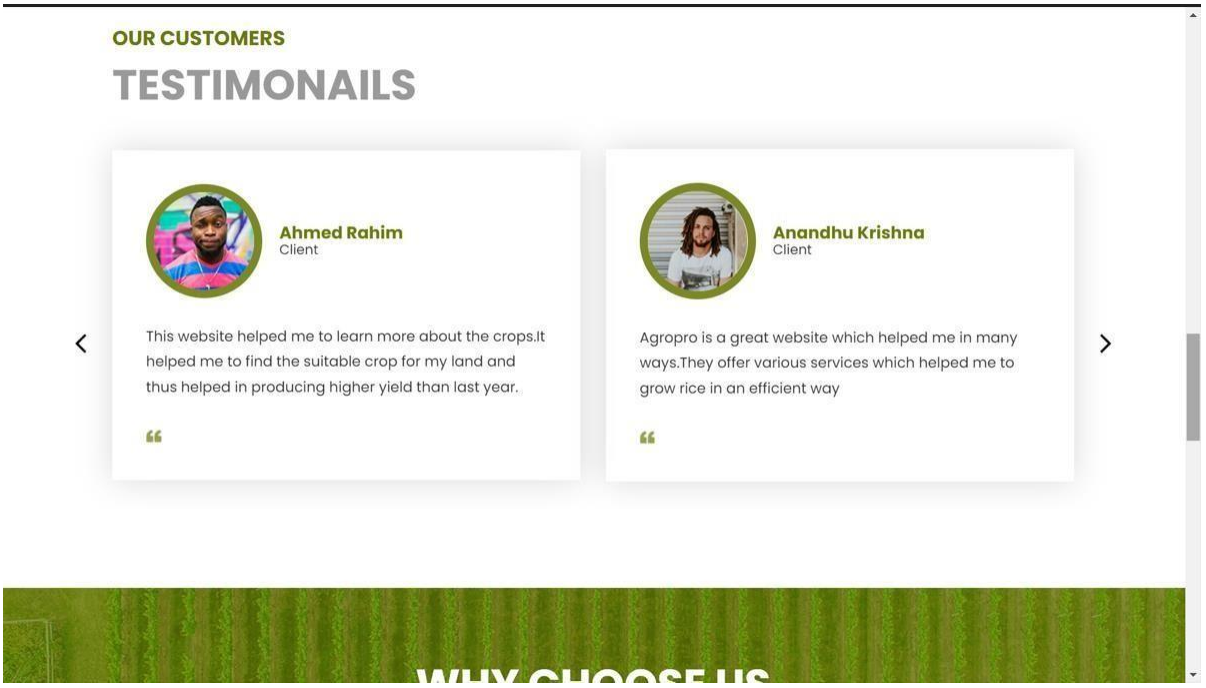


Figure 8.1.4

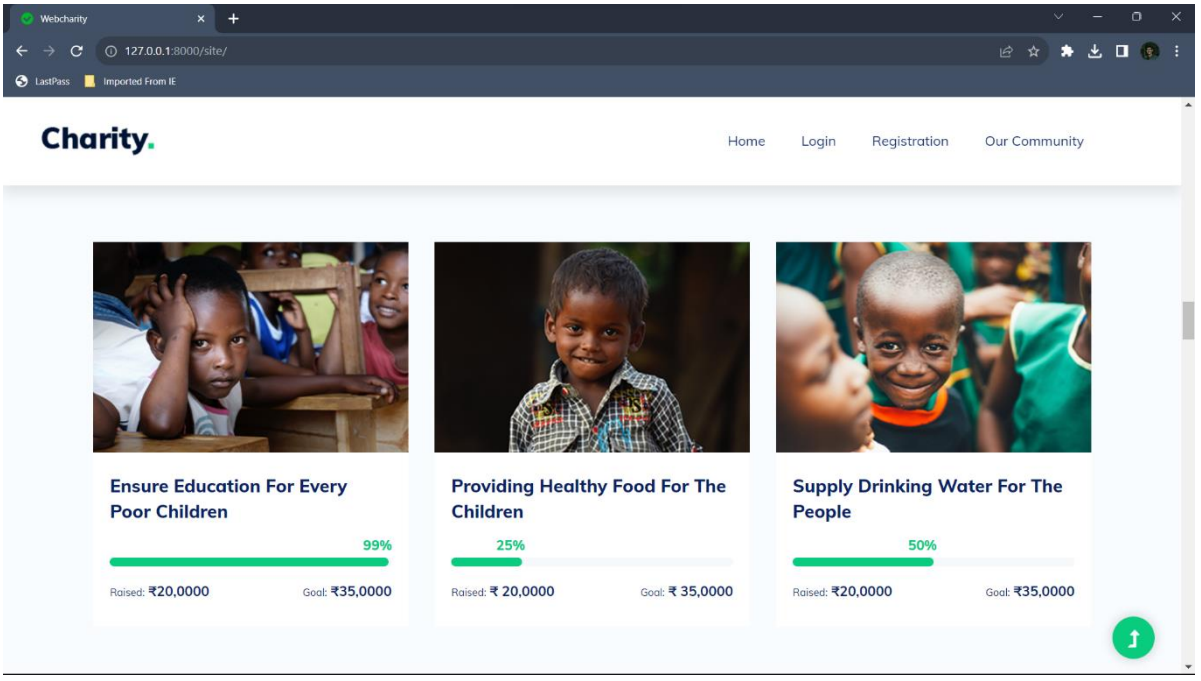


Figure 8.1.5

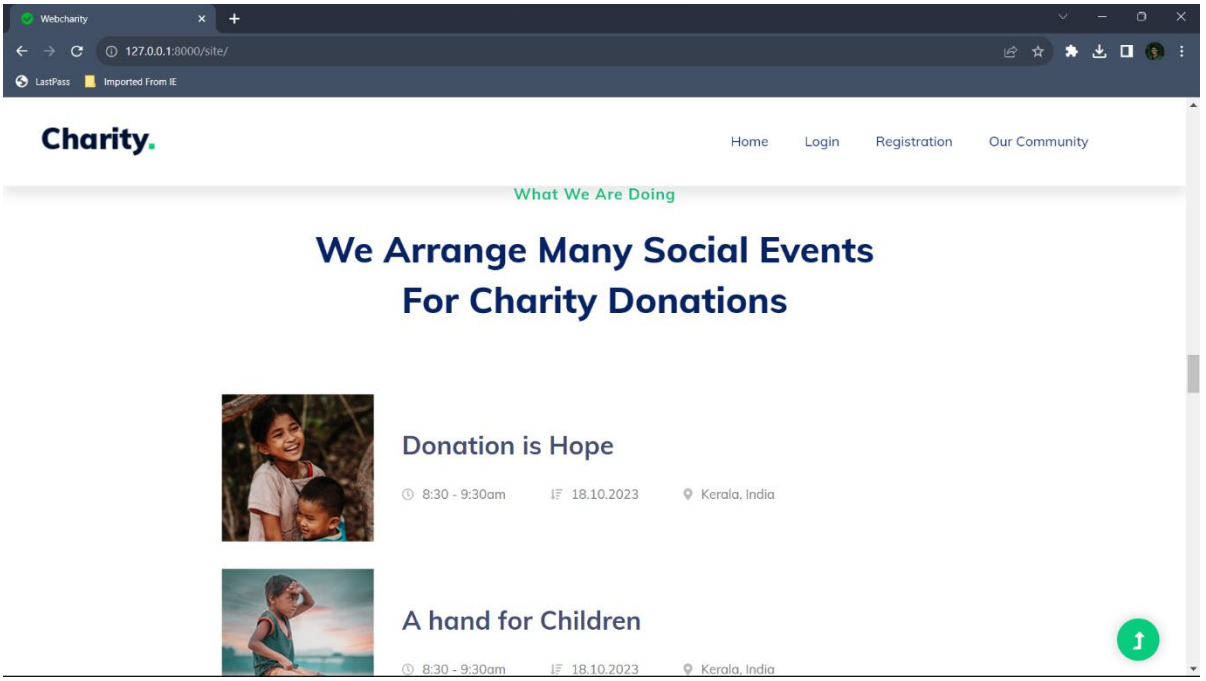


Figure 8.1.6

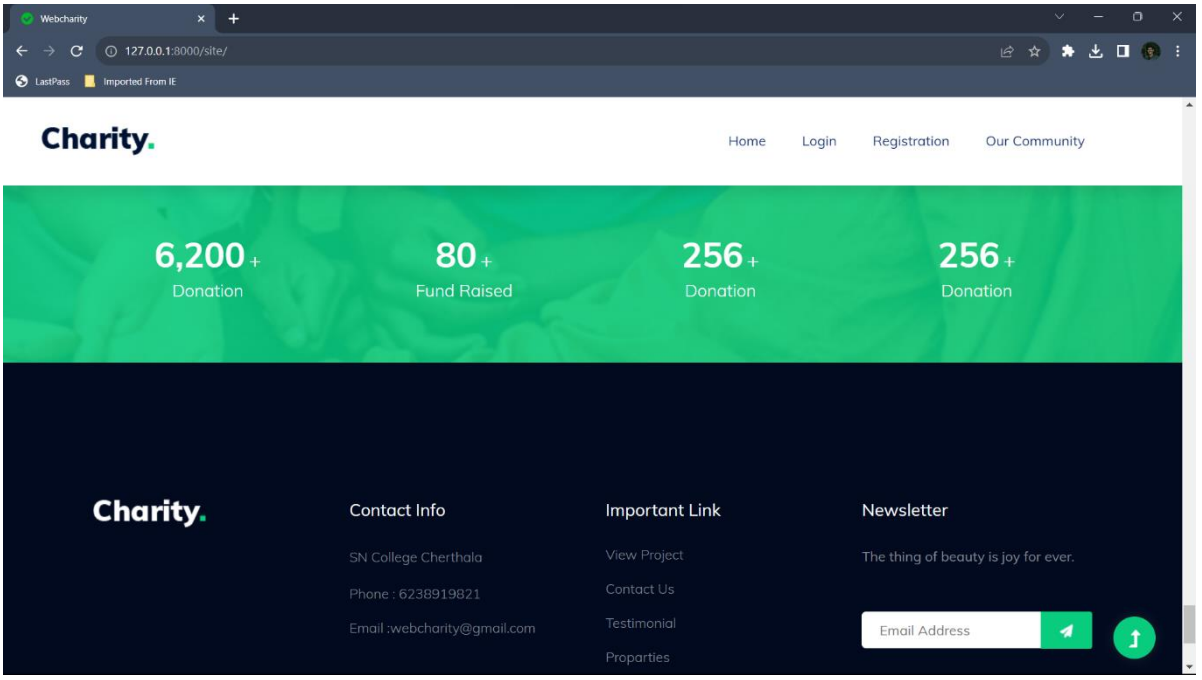


Figure 8.1.7

About us

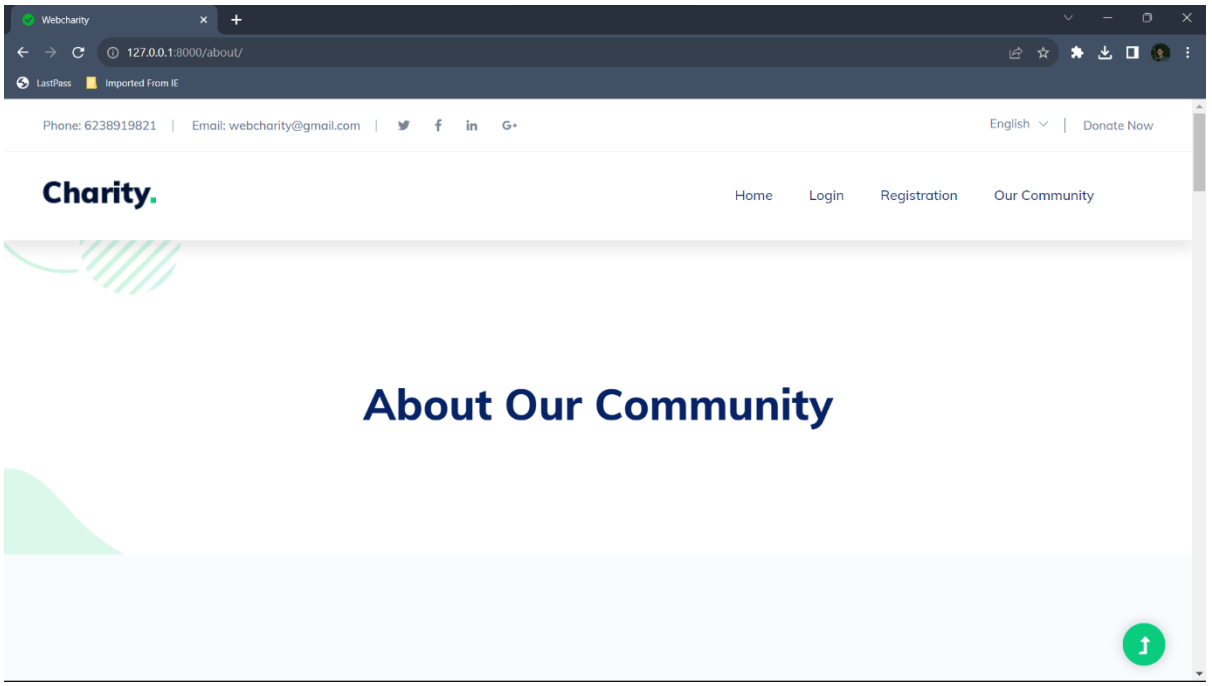


Figure 8.2

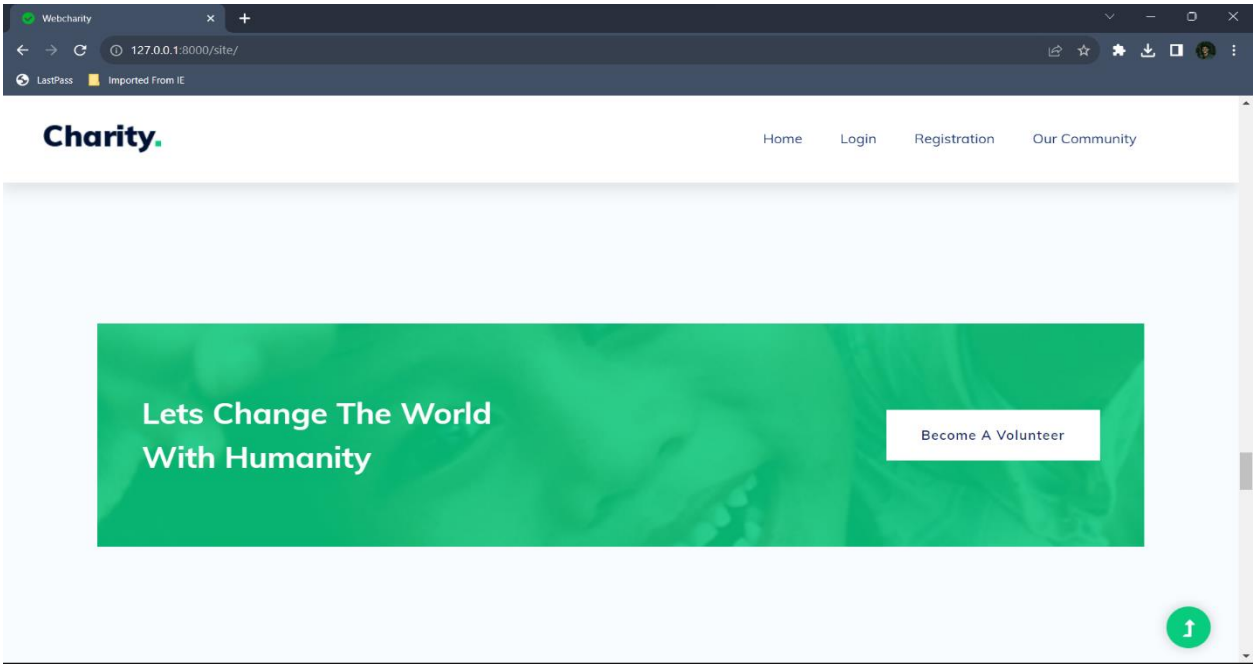


Figure 8.2.1

Service Page

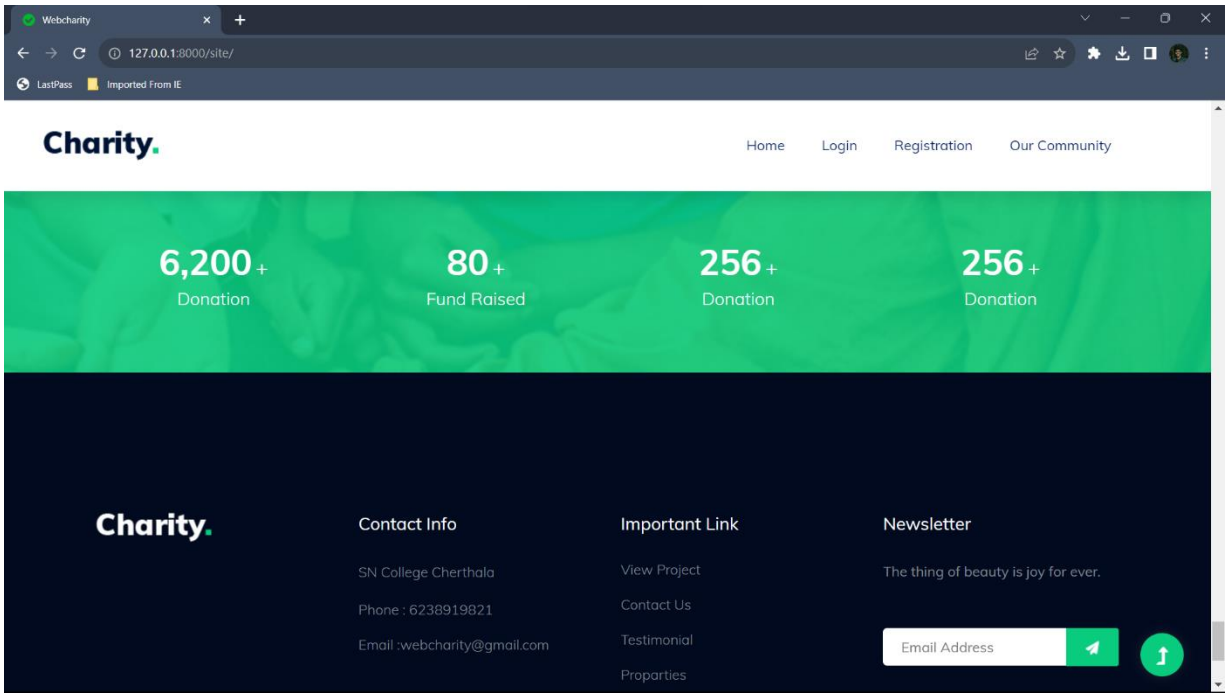


Figure 8.3

Registration Page

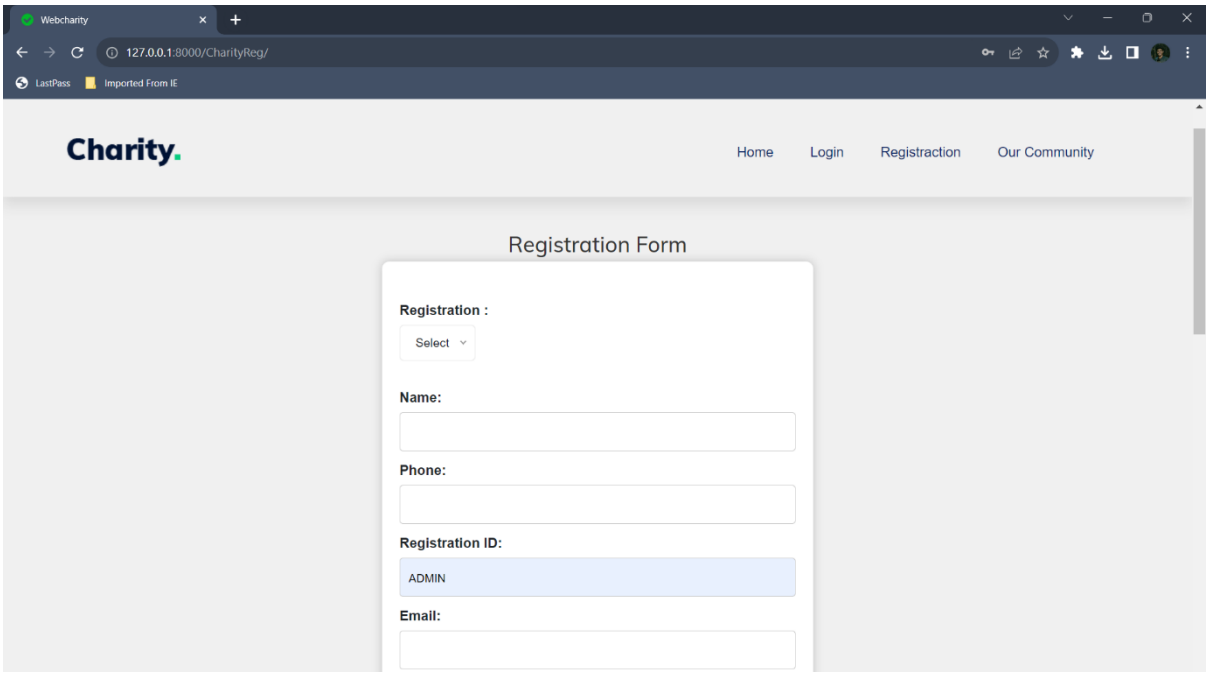


Figure 8.4

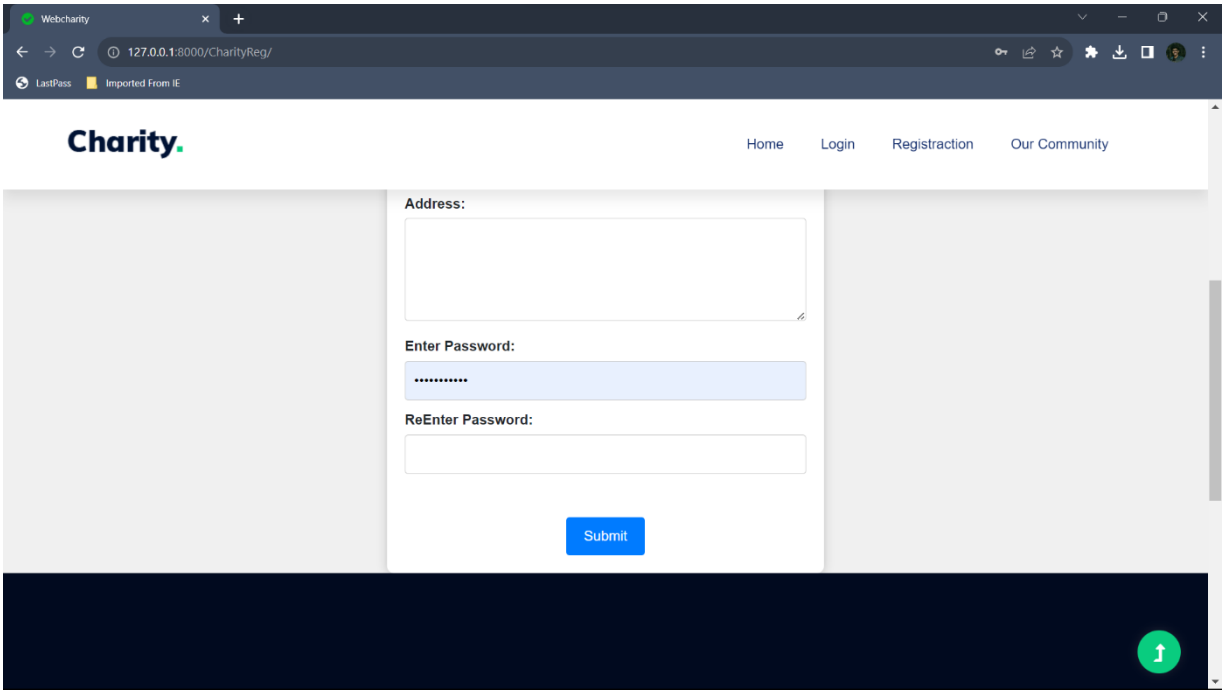


Figure 8.4.1

Donor Registration form

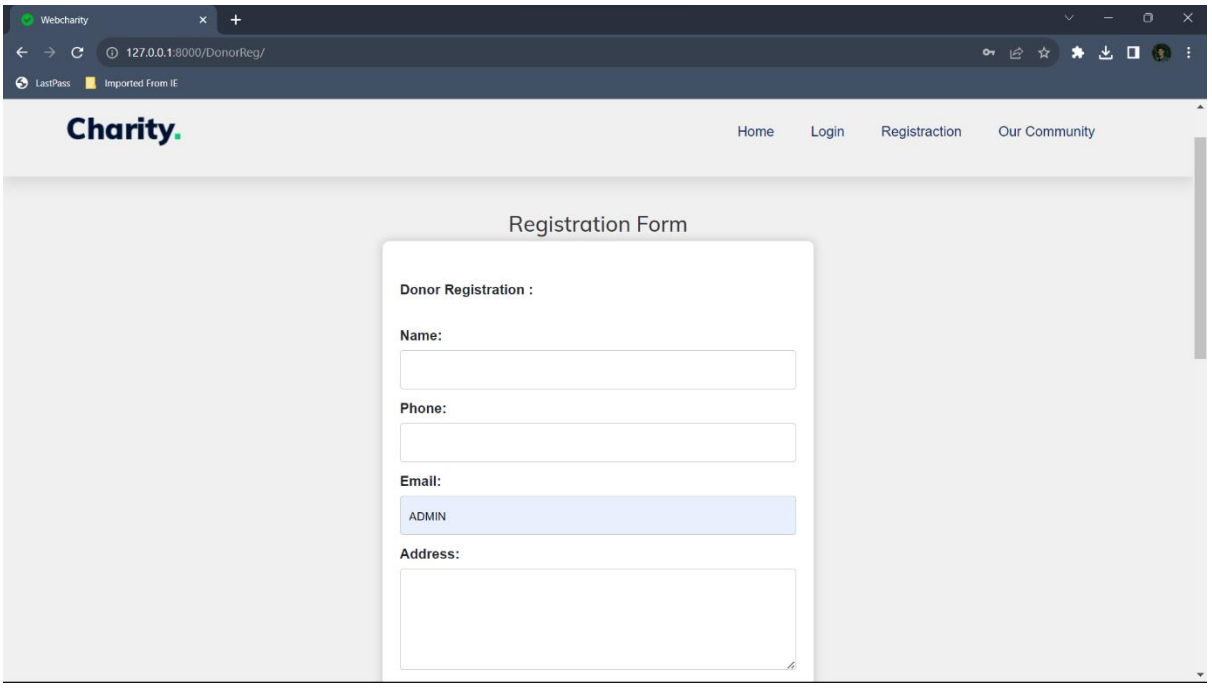


Figure 8.5

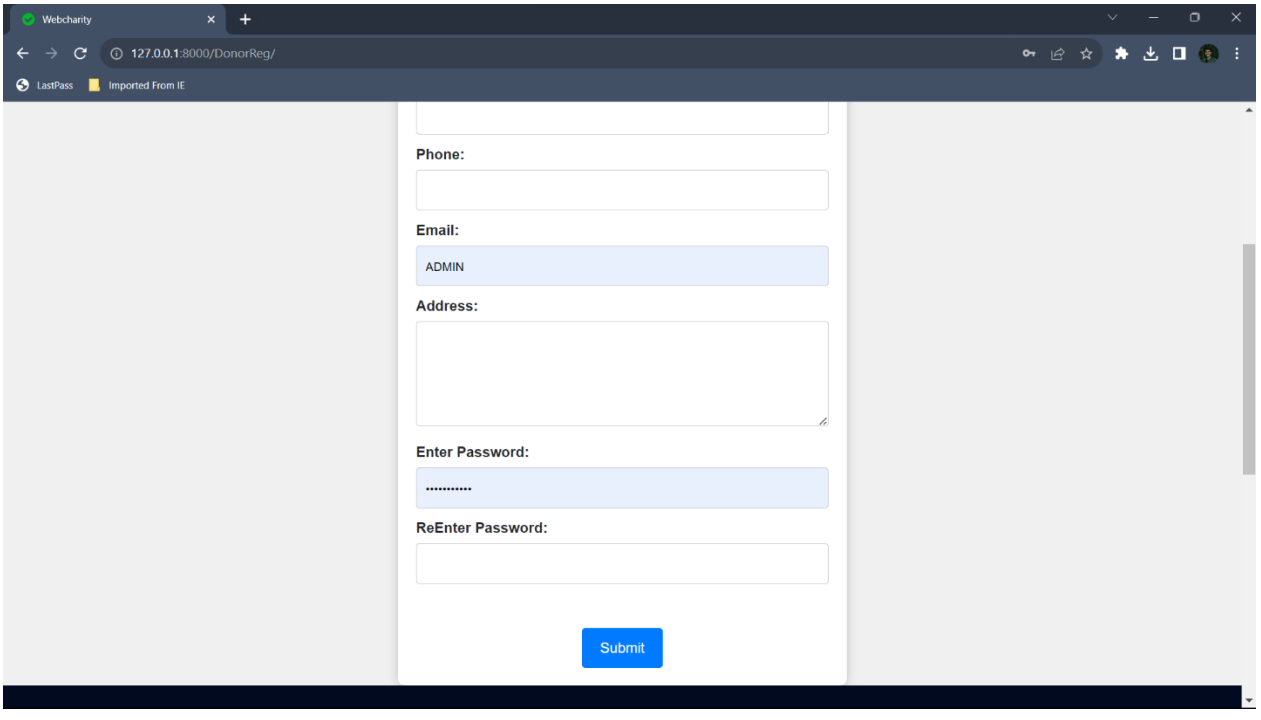


Figure 8.5.1

Charity Registration Form

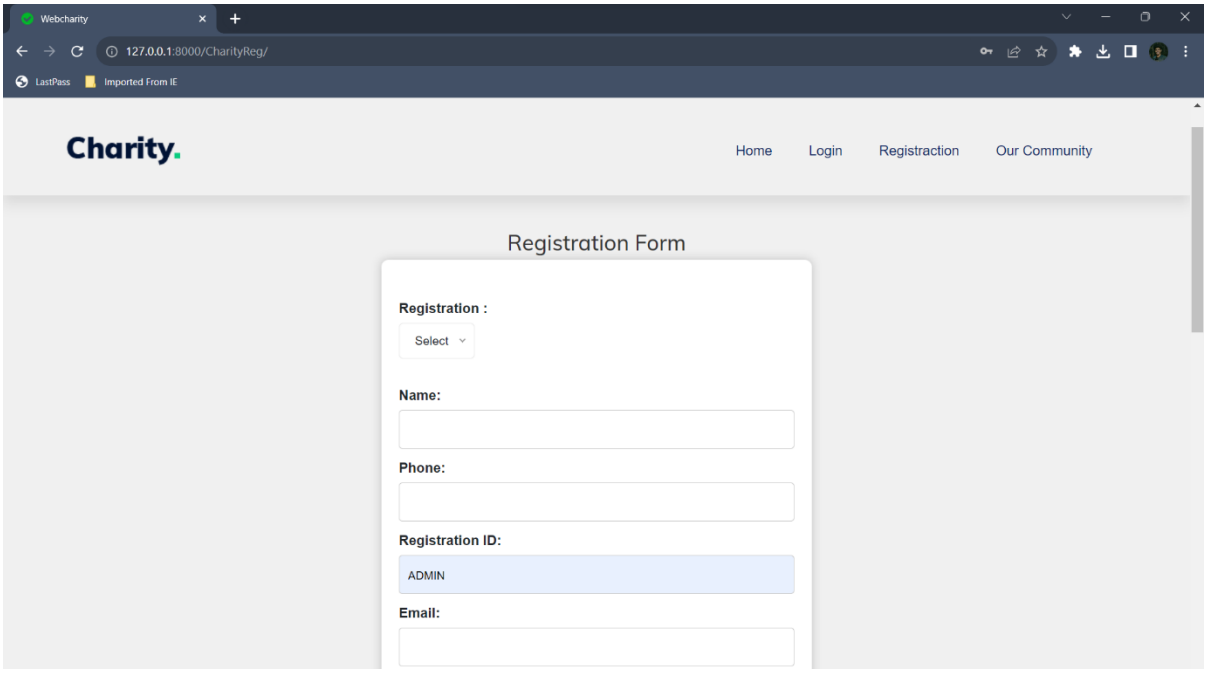


Figure 8.6

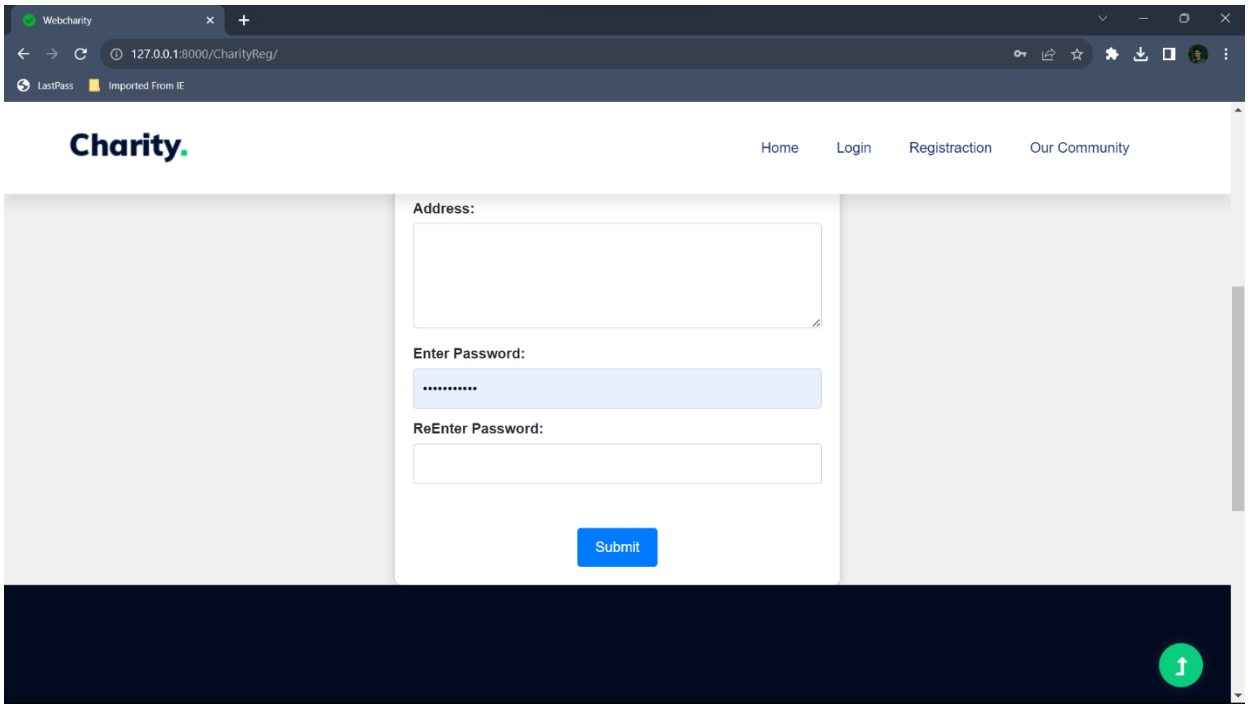


Figure 8.6.1

Login Pages

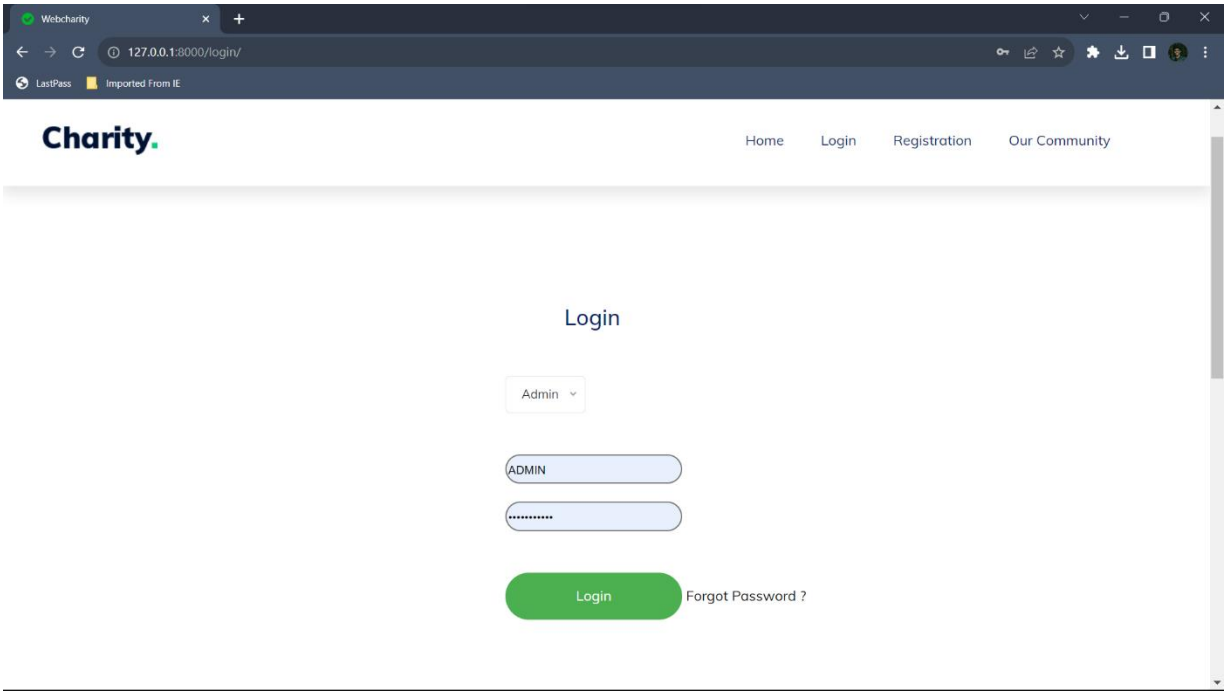


Figure 8.7

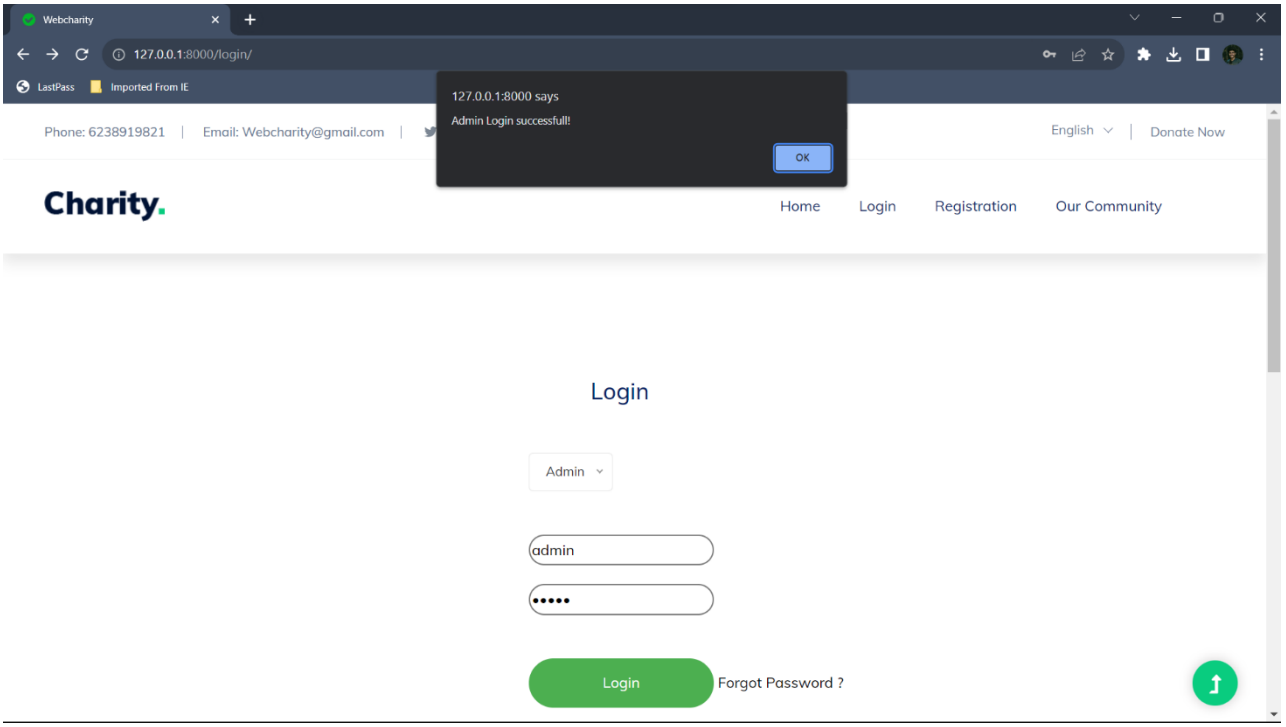


Figure 8.7.1

Admin Pages

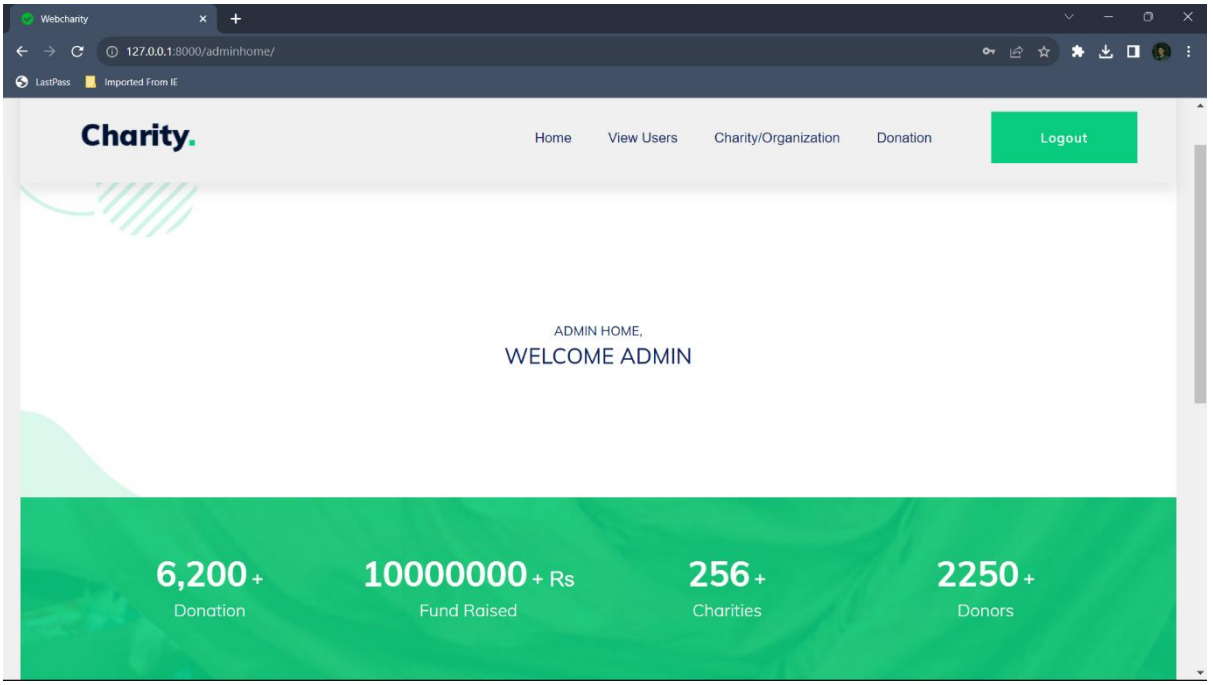


Figure 8.8

View Donors

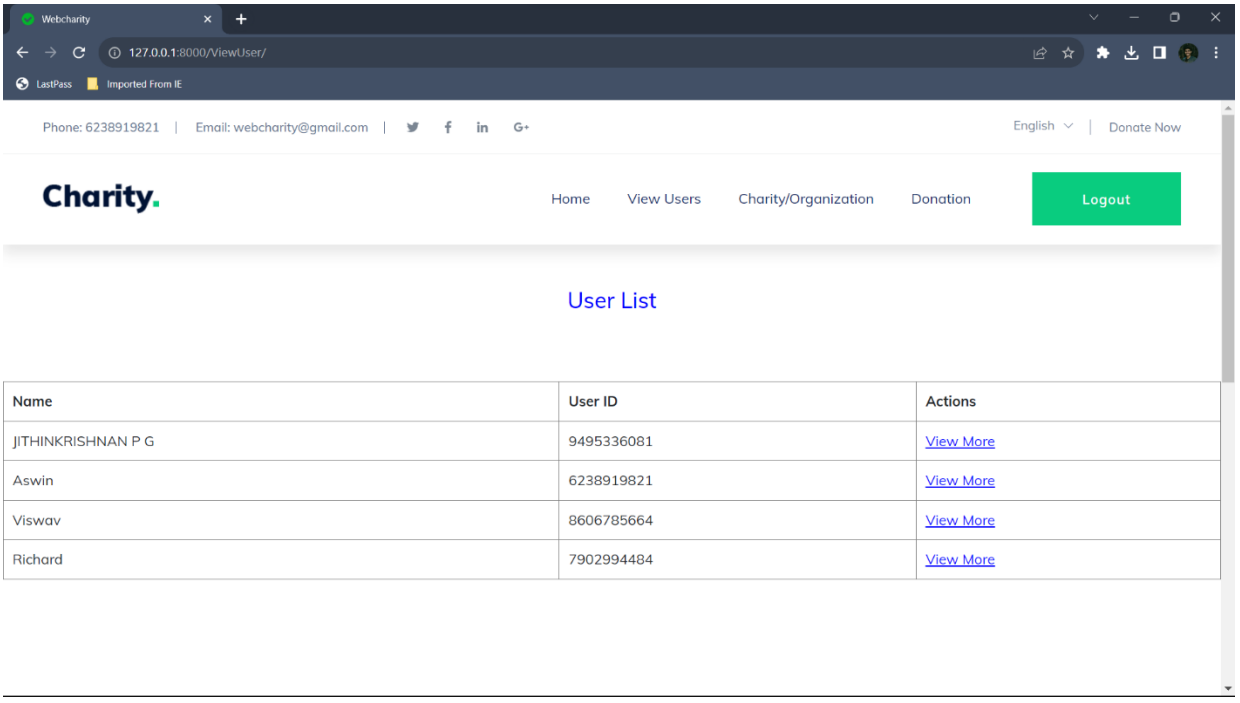


Figure 8.9

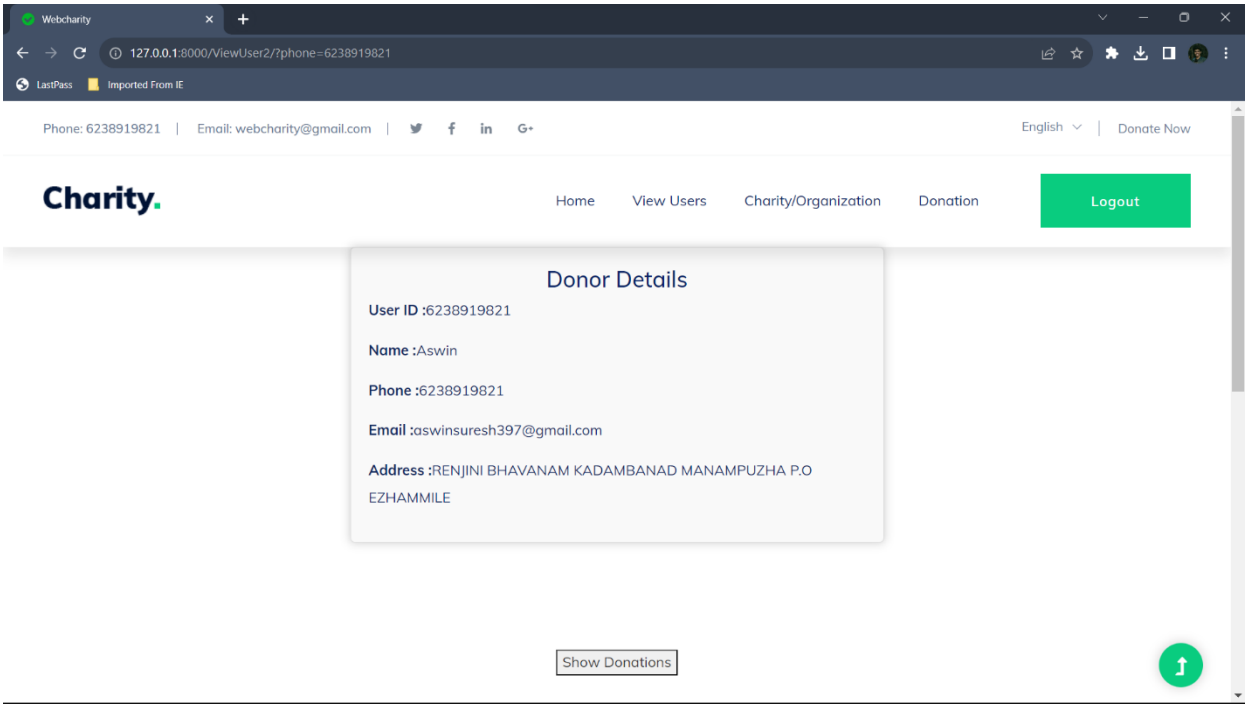
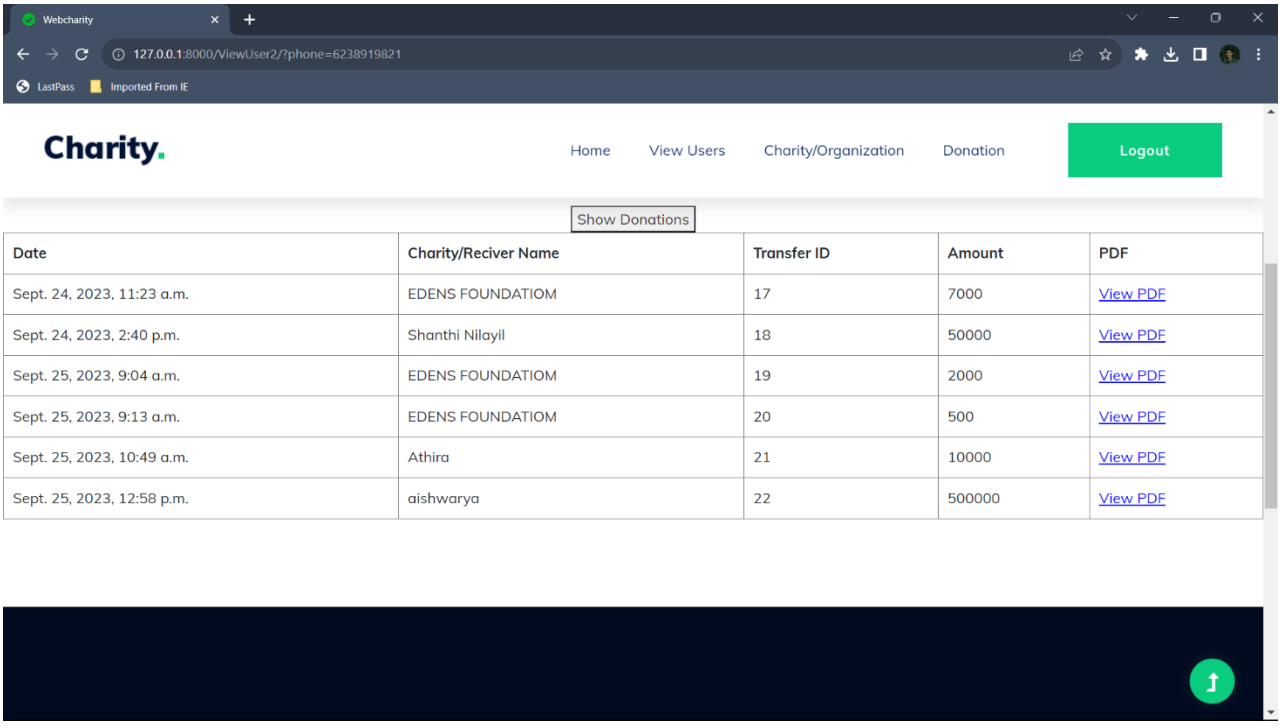


Figure 8.9.1



Date	Charity/Reciver Name	Transfer ID	Amount	PDF
Sept. 24, 2023, 11:23 a.m.	EDENS FOUNDATIOM	17	7000	View PDF
Sept. 24, 2023, 2:40 p.m.	Shanthi Nilayil	18	50000	View PDF
Sept. 25, 2023, 9:04 a.m.	EDENS FOUNDATIOM	19	2000	View PDF
Sept. 25, 2023, 9:13 a.m.	EDENS FOUNDATIOM	20	500	View PDF
Sept. 25, 2023, 10:49 a.m.	Athira	21	10000	View PDF
Sept. 25, 2023, 12:58 p.m.	aishwarya	22	500000	View PDF

Figure 8.9.2

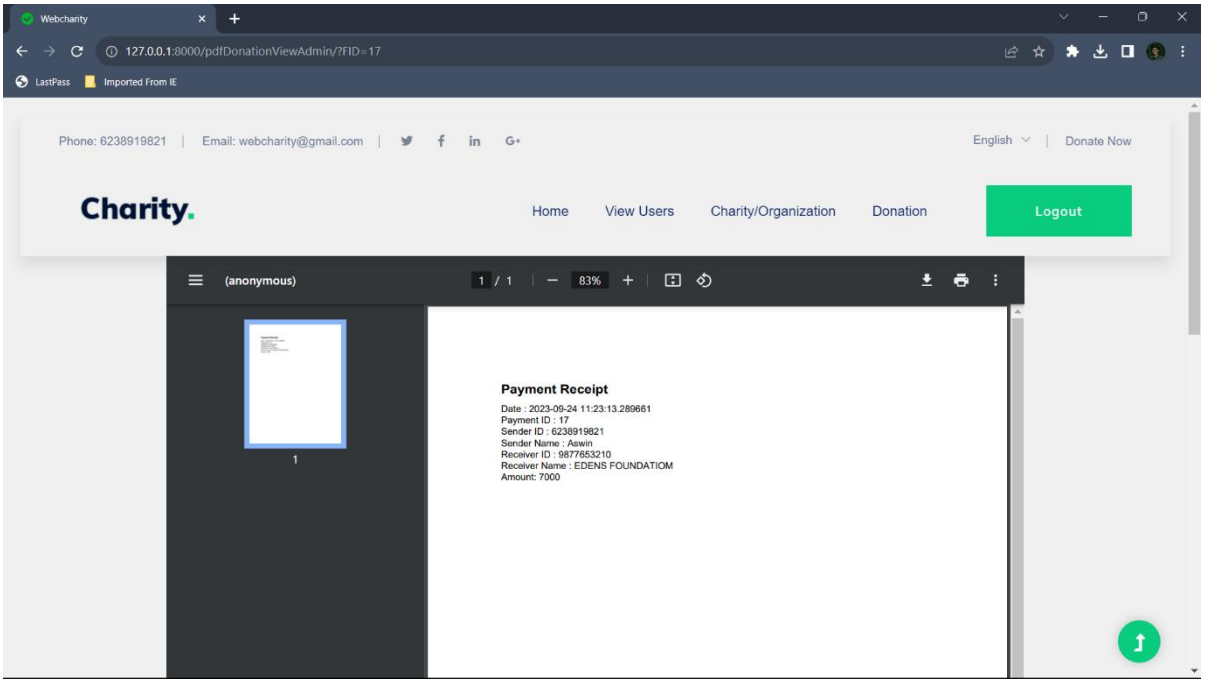


Figure 8.9.3

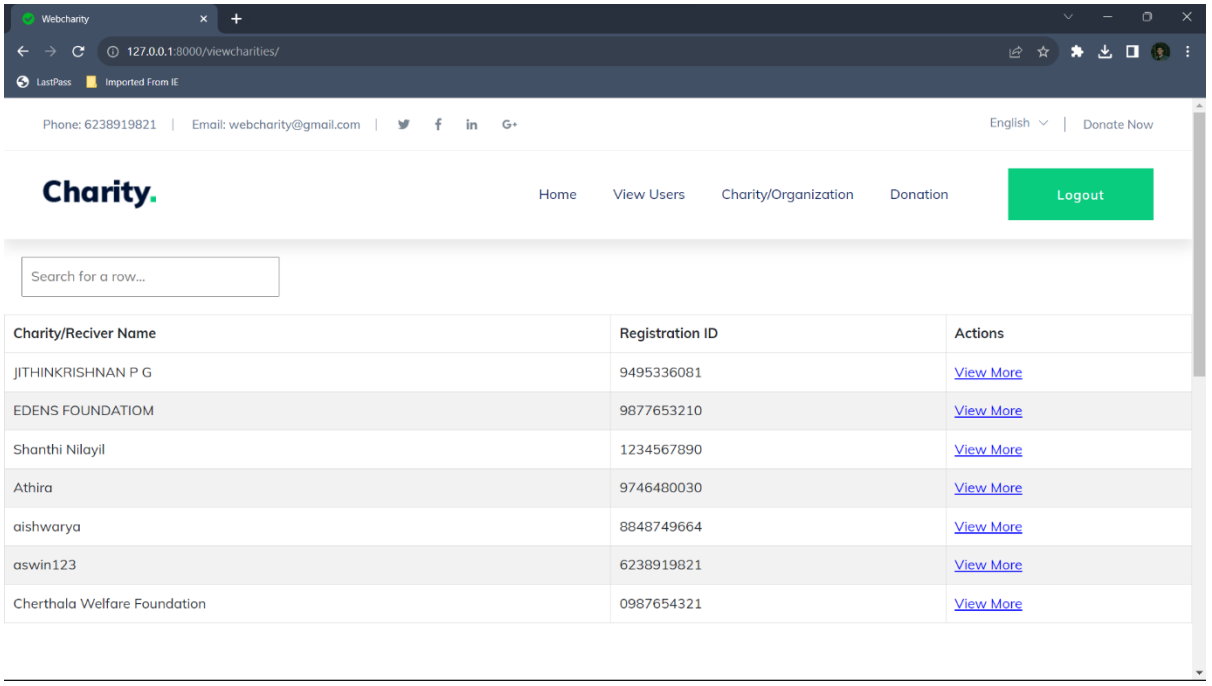


Figure 8.9.4

Charity Details

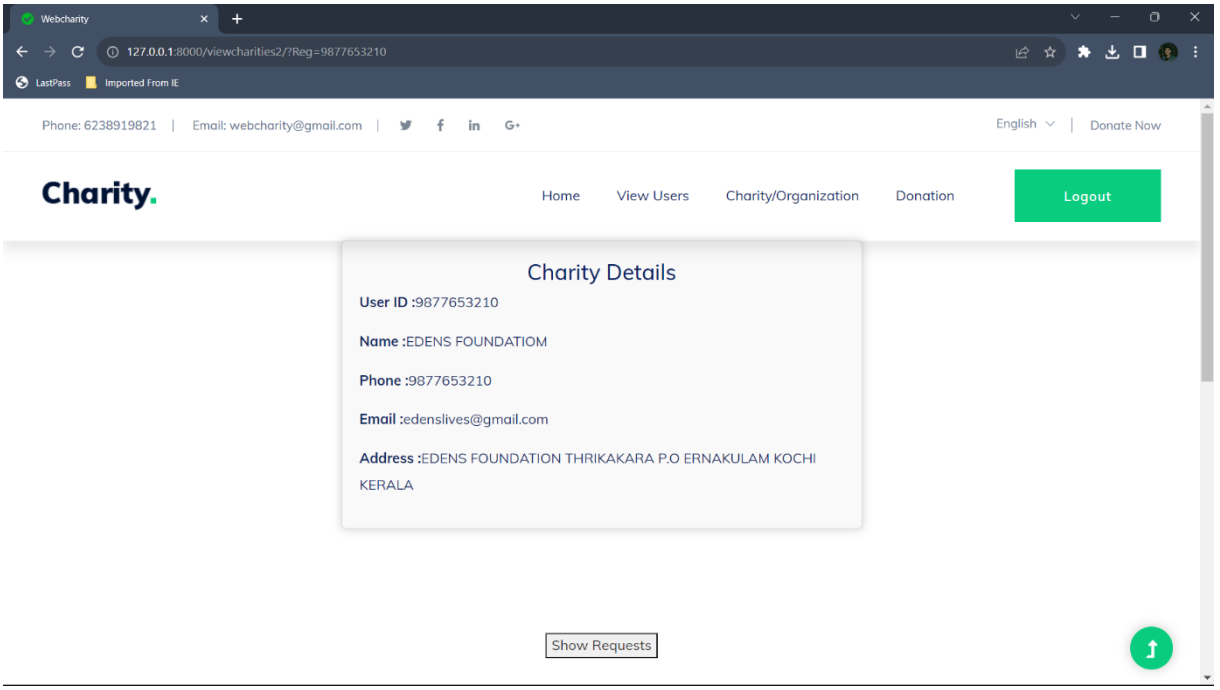


Figure 8.10

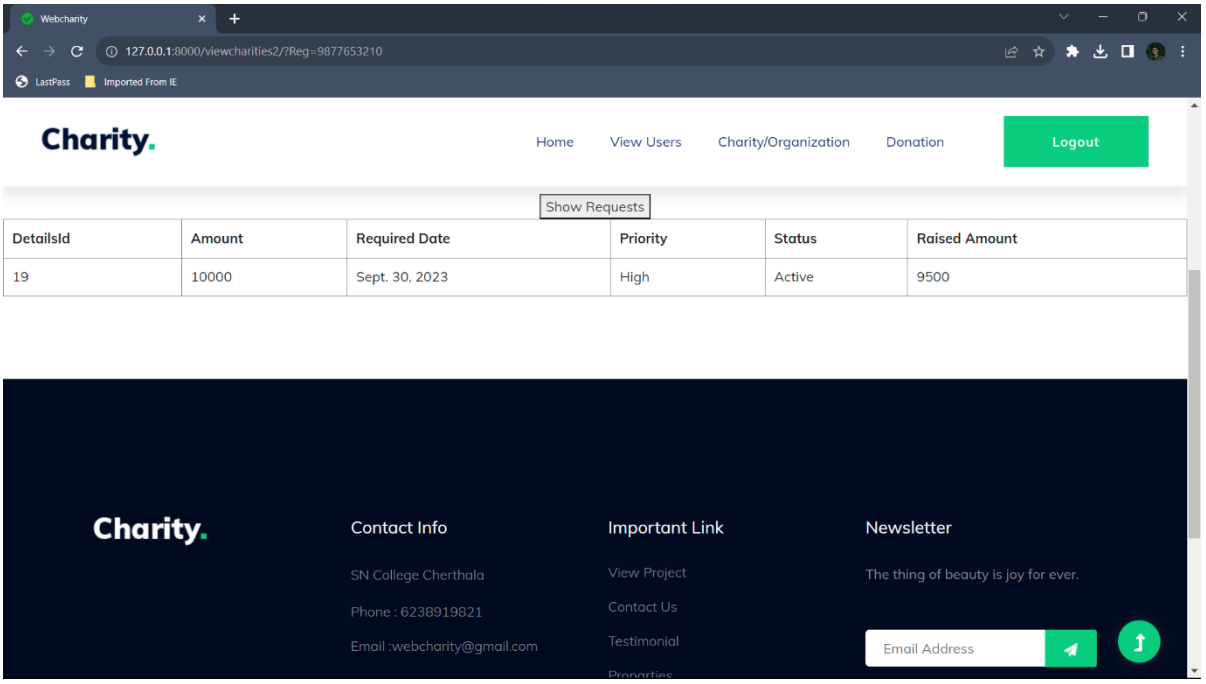


Figure 8.10.1

Approve Charity

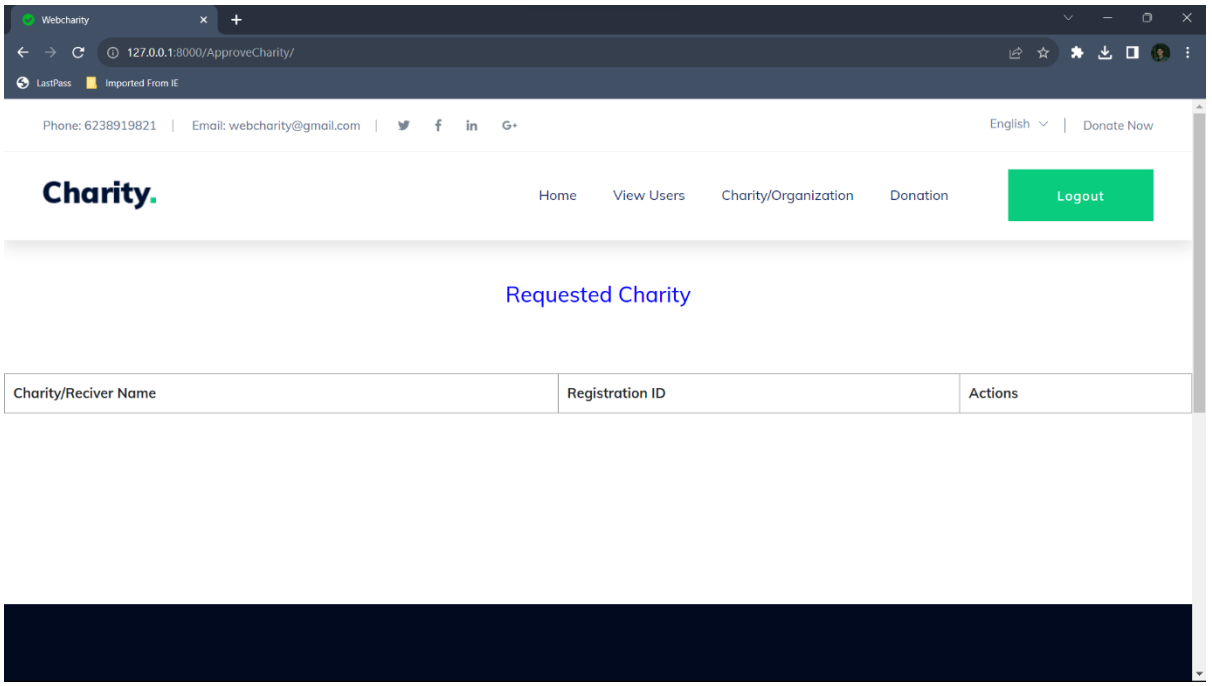


Figure 8.11

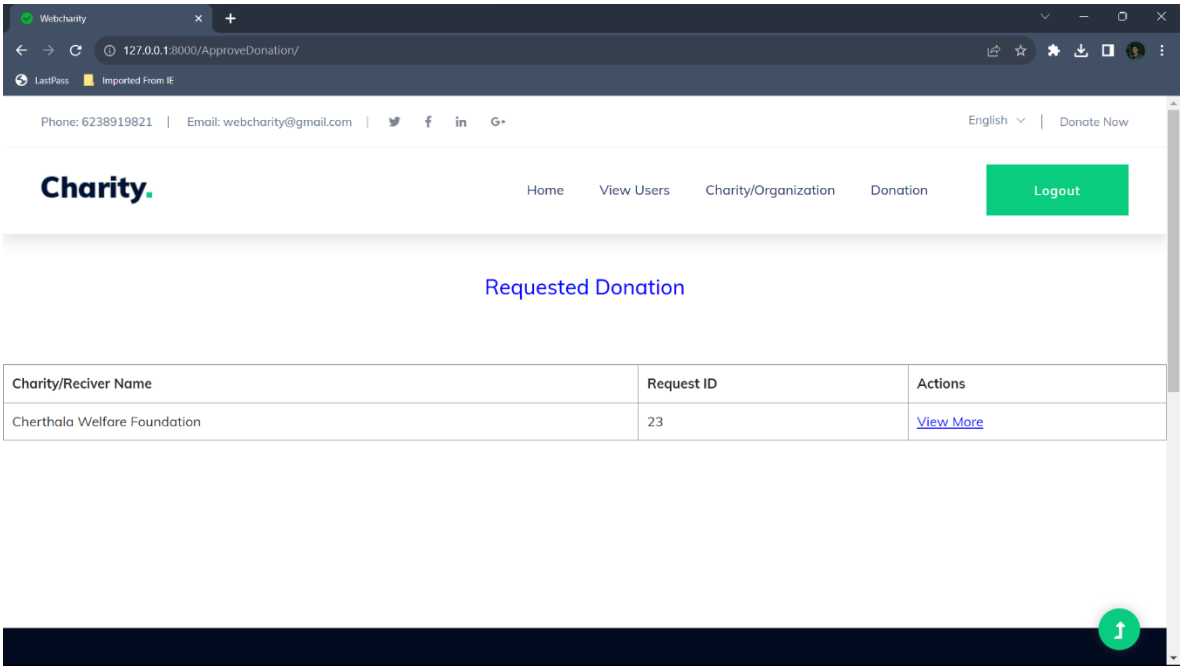


Figure 8.11.1

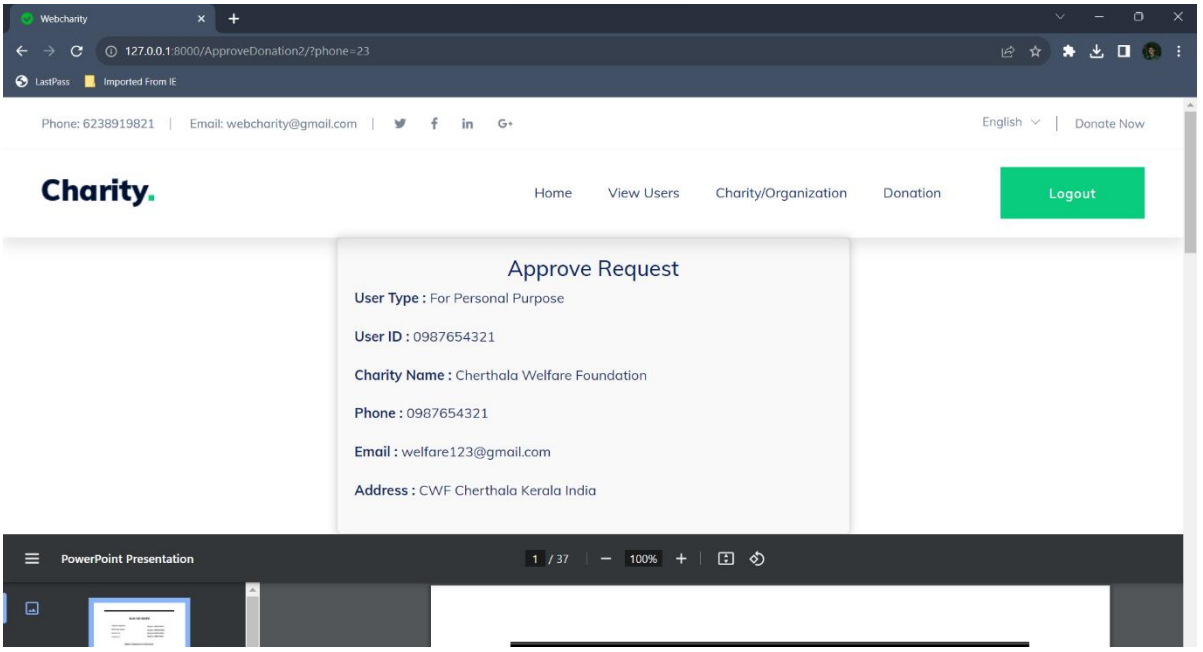


Figure 8.11.2

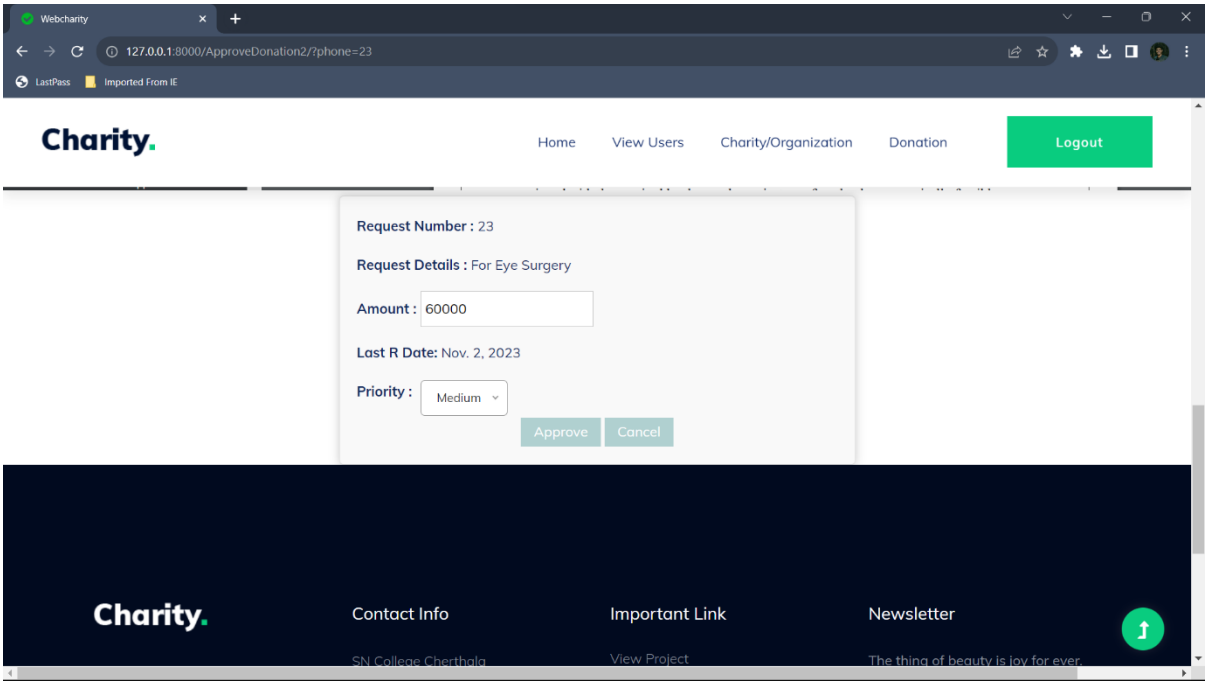


Figure 8.11.3

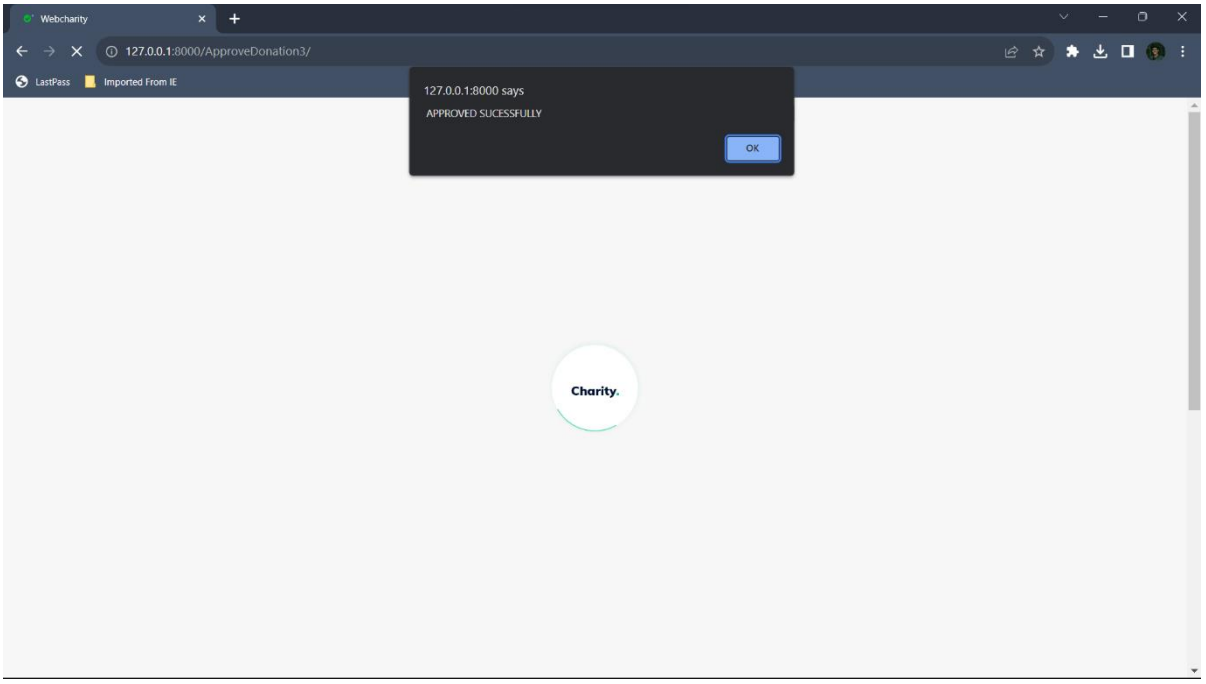


Figure 8.11.4

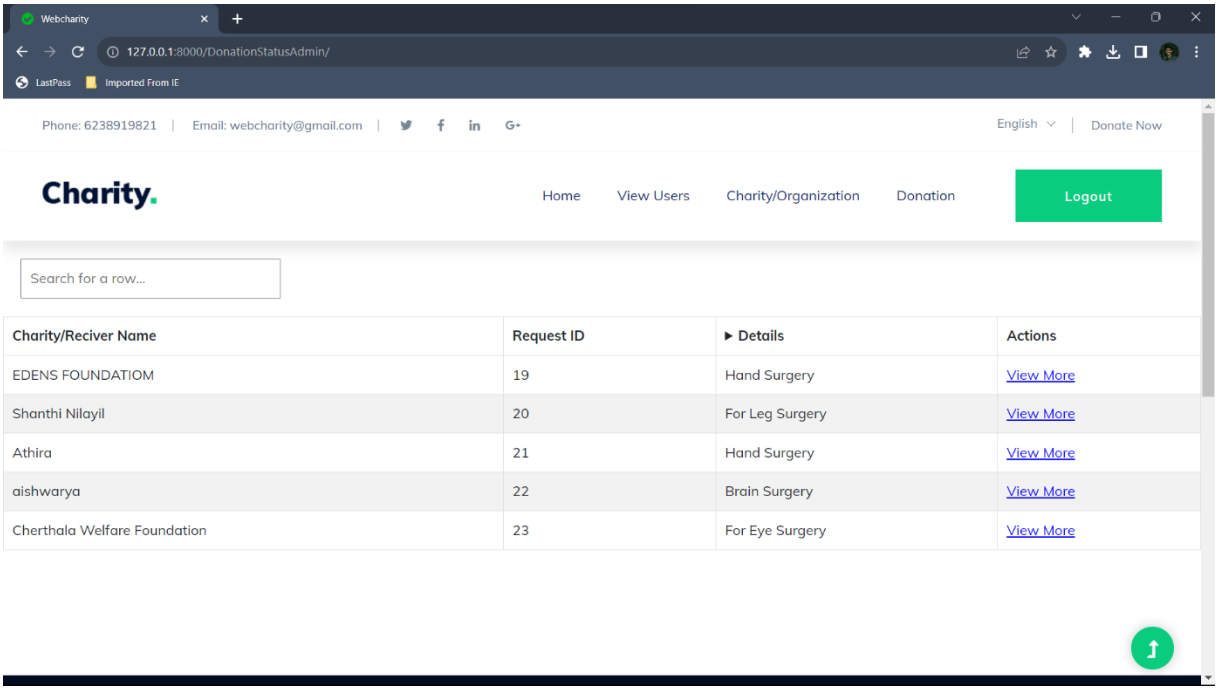


Figure 8.11.5

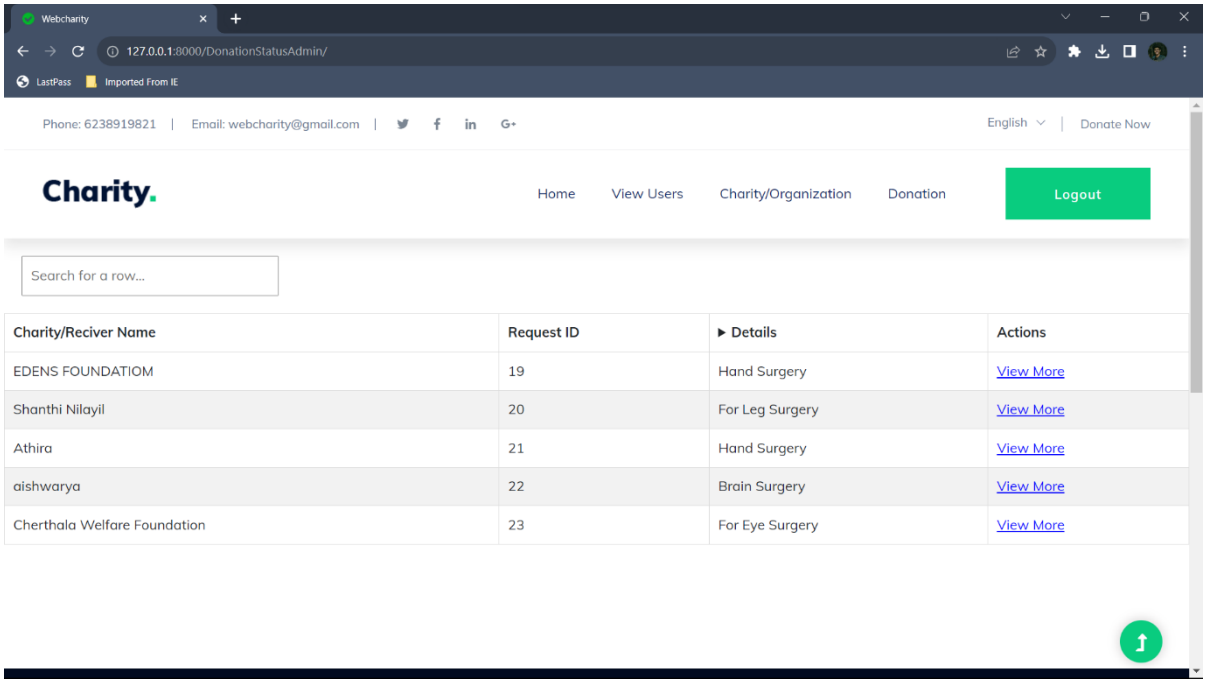

Webcharity
127.0.0.1:8000/DonationStatusAdmin/
LastPass
Imported From IE
Phone: 6238919821 | Email: webcharity@gmail.com |
English | Donate Now
Charity.
Home
View Users
Charity/Organization
Donation
Logout
Search for a row...
Charity/Reciver Name
Request ID
▸ Details
Actions
EDENS FOUNDATIOM
19
Hand Surgery
View More
Shanthi Nilayil
20
For Leg Surgery
View More
Athira
21
Hand Surgery
View More
aishwarya
22
Brain Surgery
View More
Cherthala Welfare Foundation
23
For Eye Surgery
View More

Figure 8.11.6

CHARITY

Charity Login Page

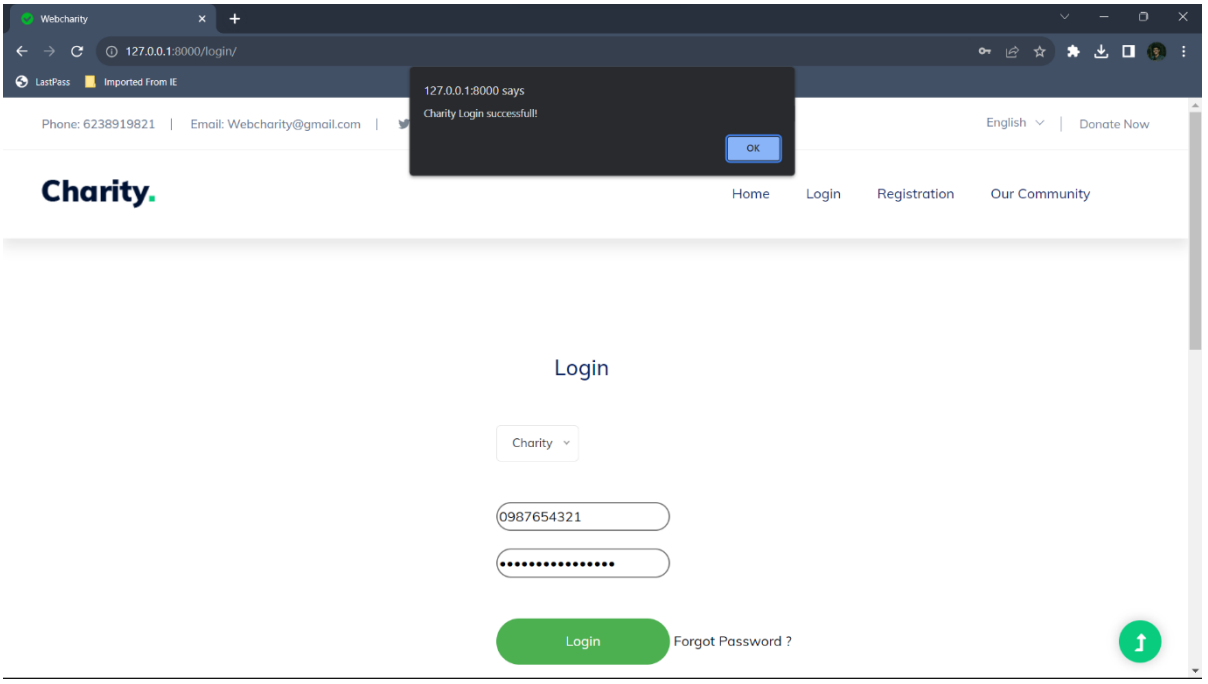


Figure 8.12

Charity Home Page

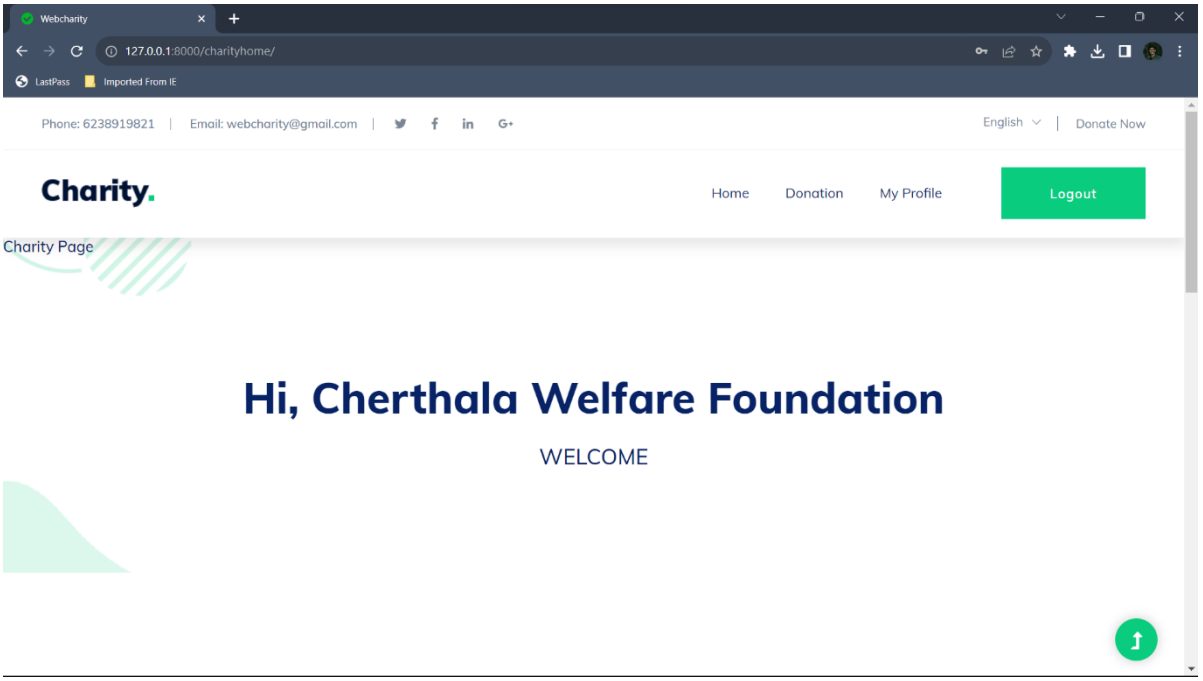


Figure 8.13

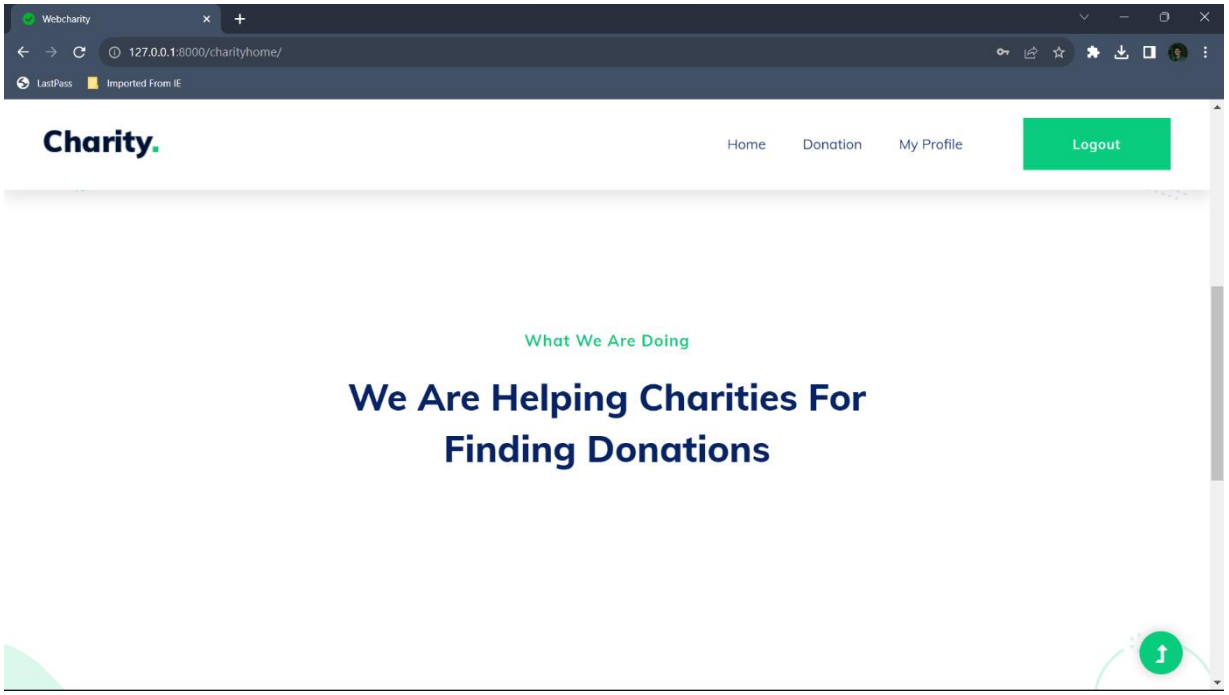


Figure 8.13.1

View Donation

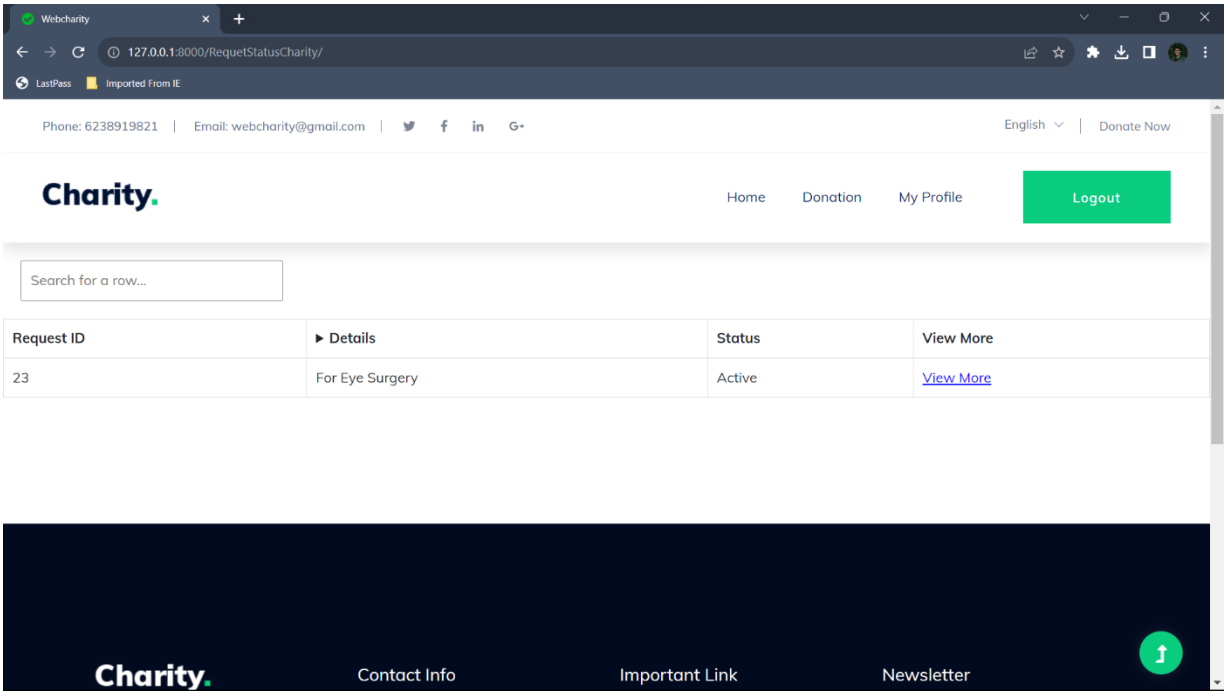


Figure 8.14

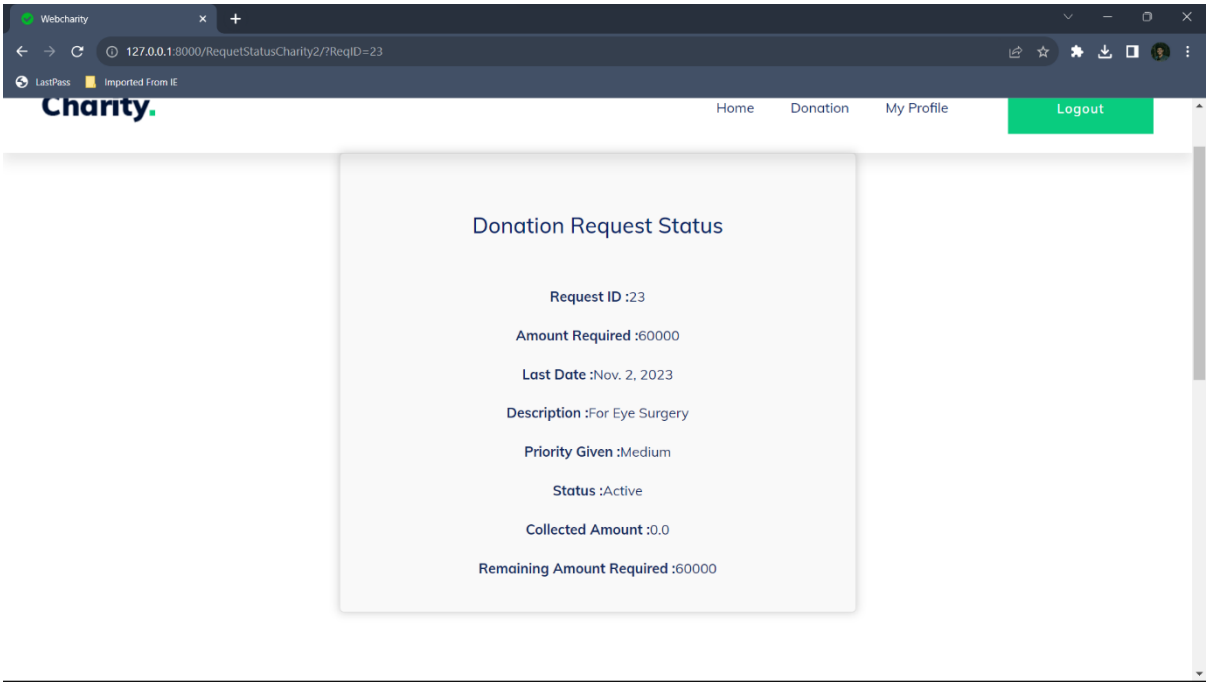


Figure 8.14.1

Request Donation

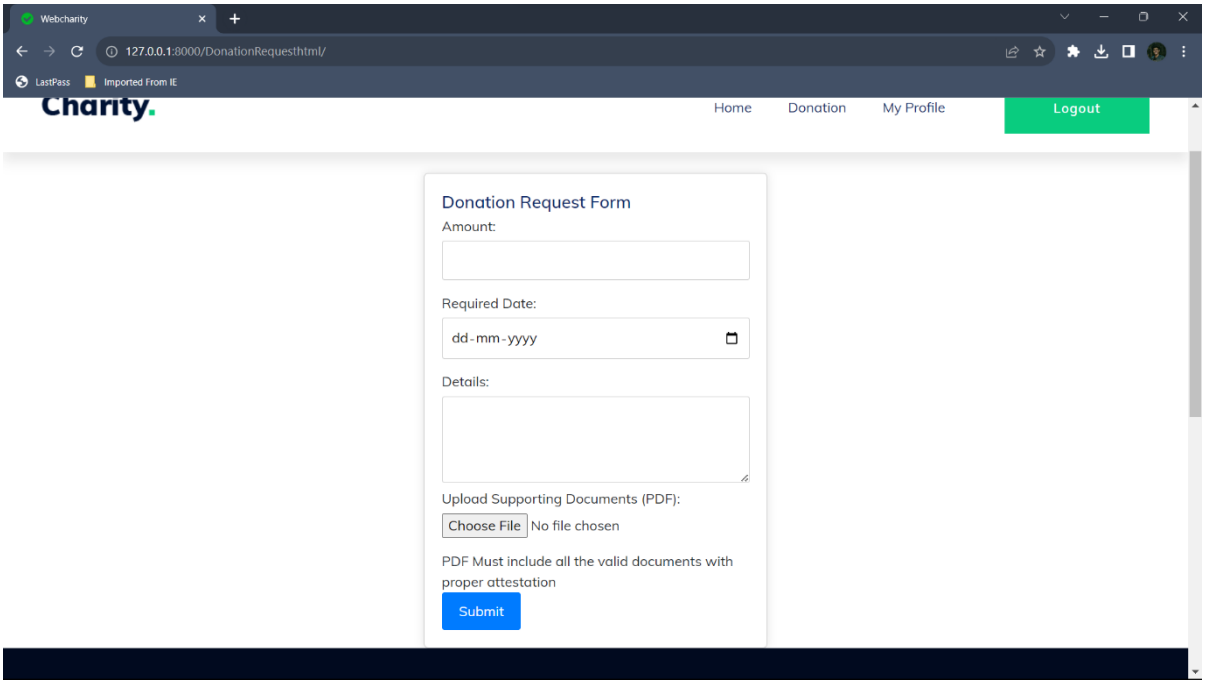


Figure 8.15

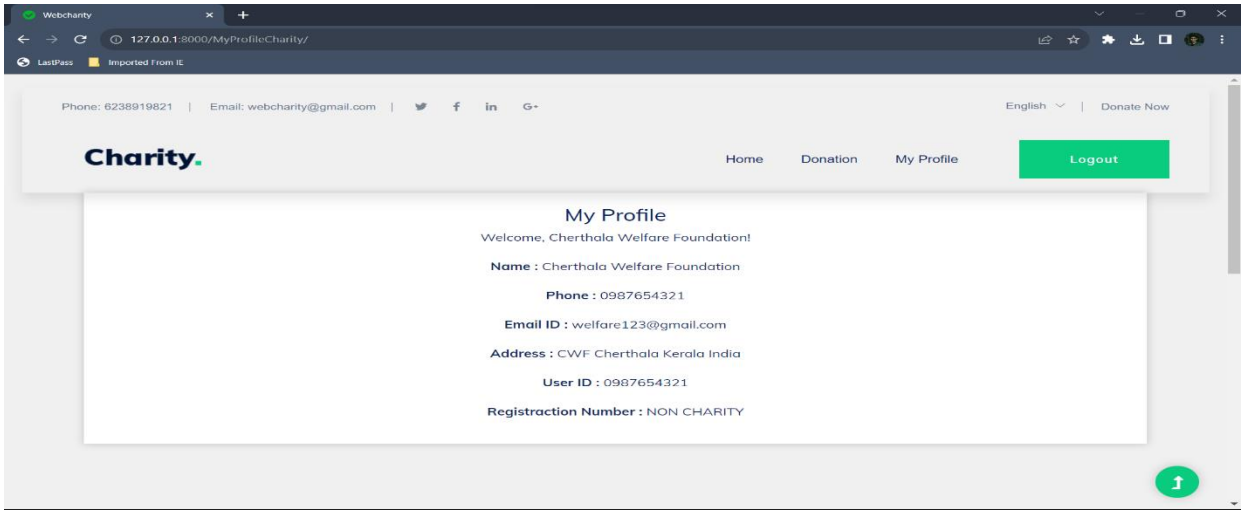

Webcharity
127.0.0.1:8000/MyProfileCharity/
LastPass
Imported From IE
Phone: 6238919821 | Email: webcharity@gmail.com |
English | Donate Now
Charity.
Home
Donation
My Profile
Logout
My Profile
Welcome, Cherthala Welfare Foundation!
Name : Cherthala Welfare Foundation
Phone : 0987654321
Email ID : welfare123@gmail.com
Address : CWF Cherthala Kerala India
User ID : 0987654321
Registraction Number : NON CHARITY

Figure 8.15.1

Donor Login Page

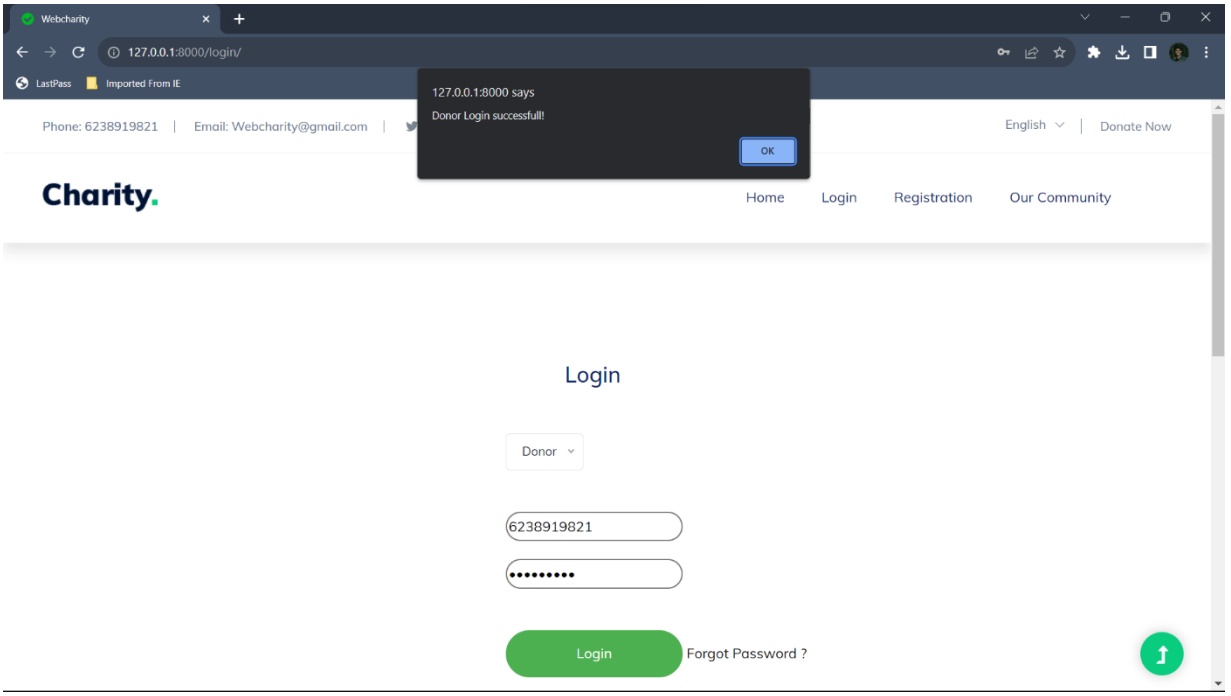


Figure 8.16

Donor Home Page

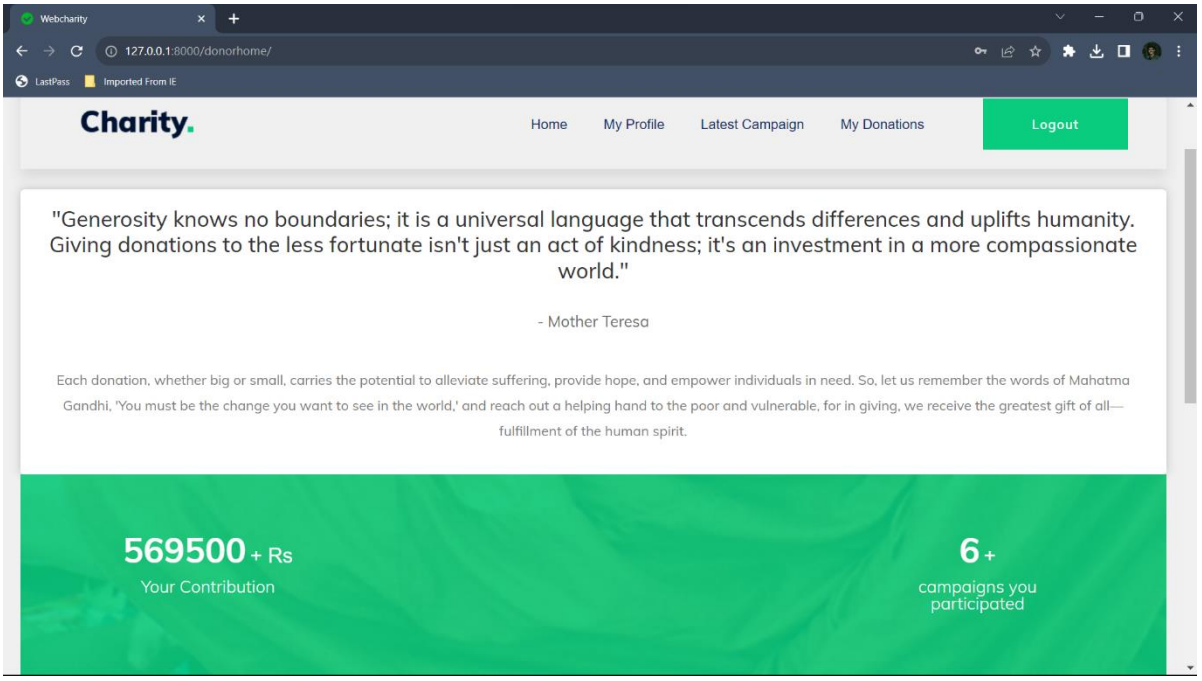


Figure 8.17

Donor Profile

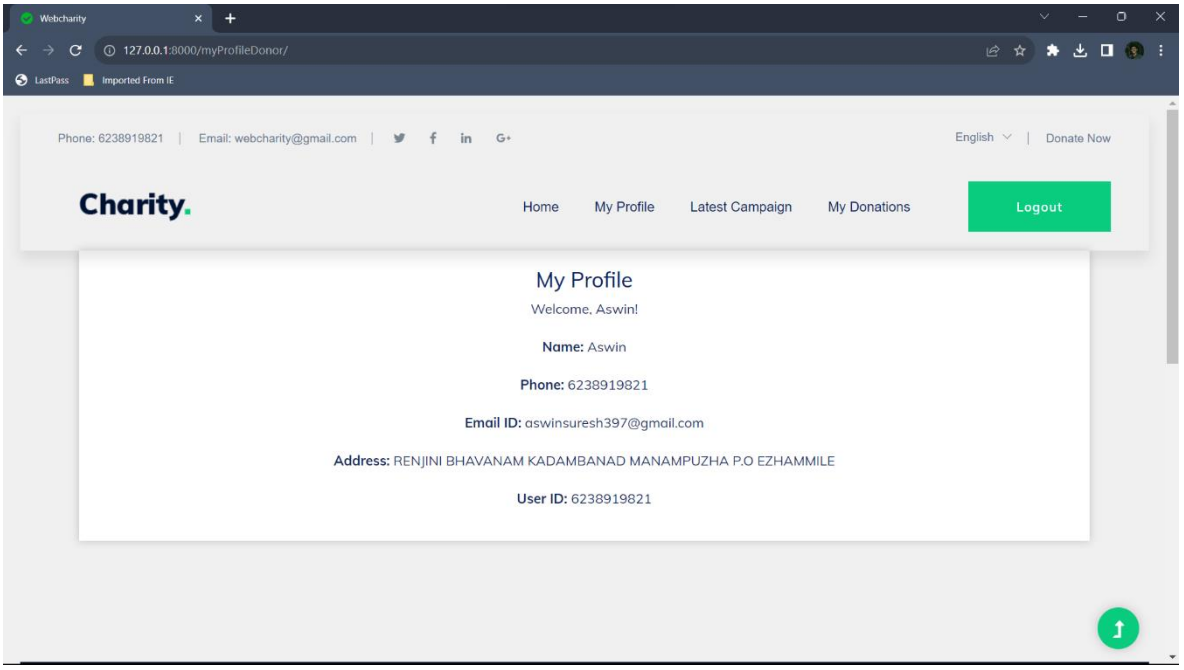


Figure 8.18

Fund Raising Campaigns

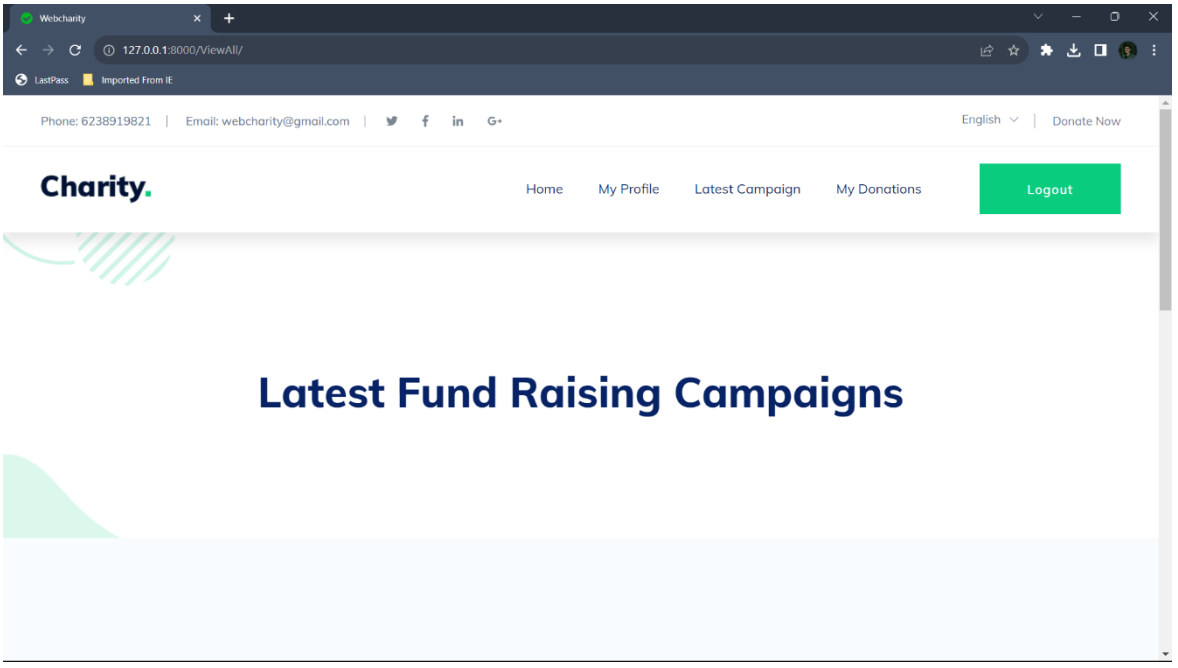


Figure 8.19

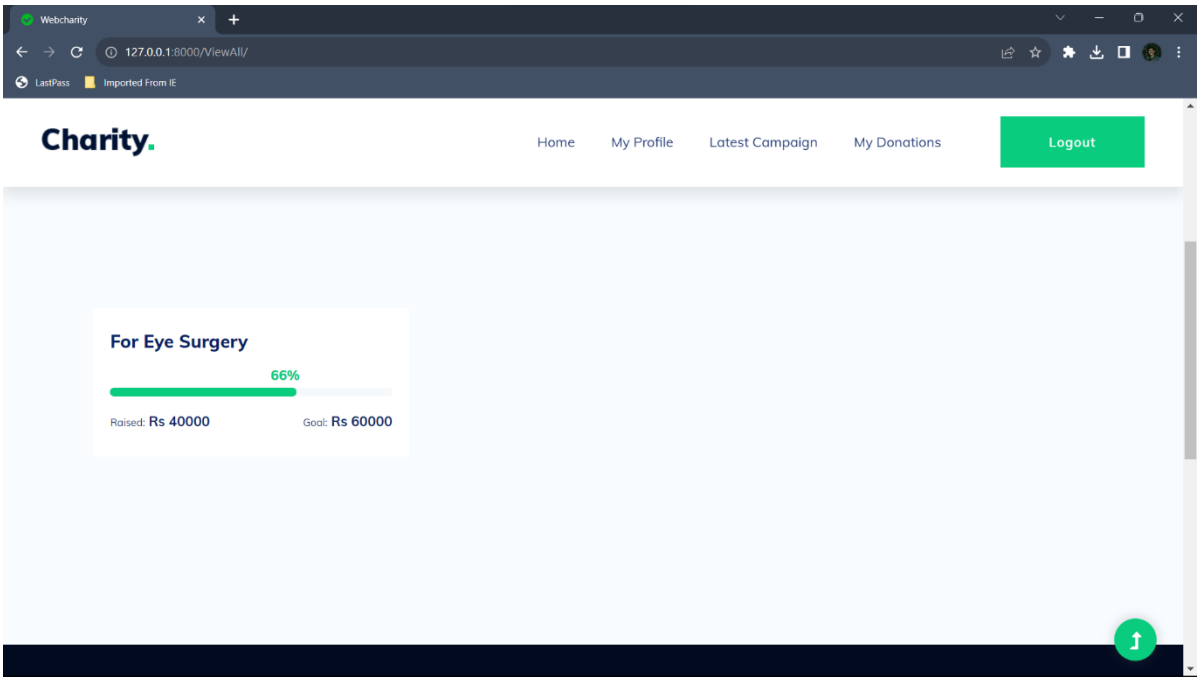


Figure 8.19.1

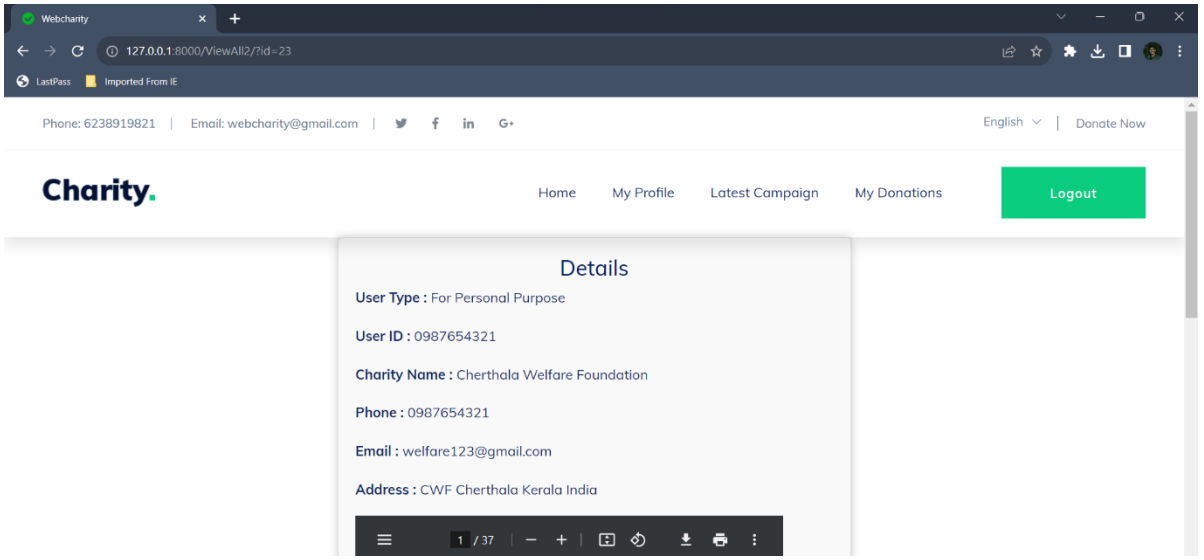


Figure 8.19.2

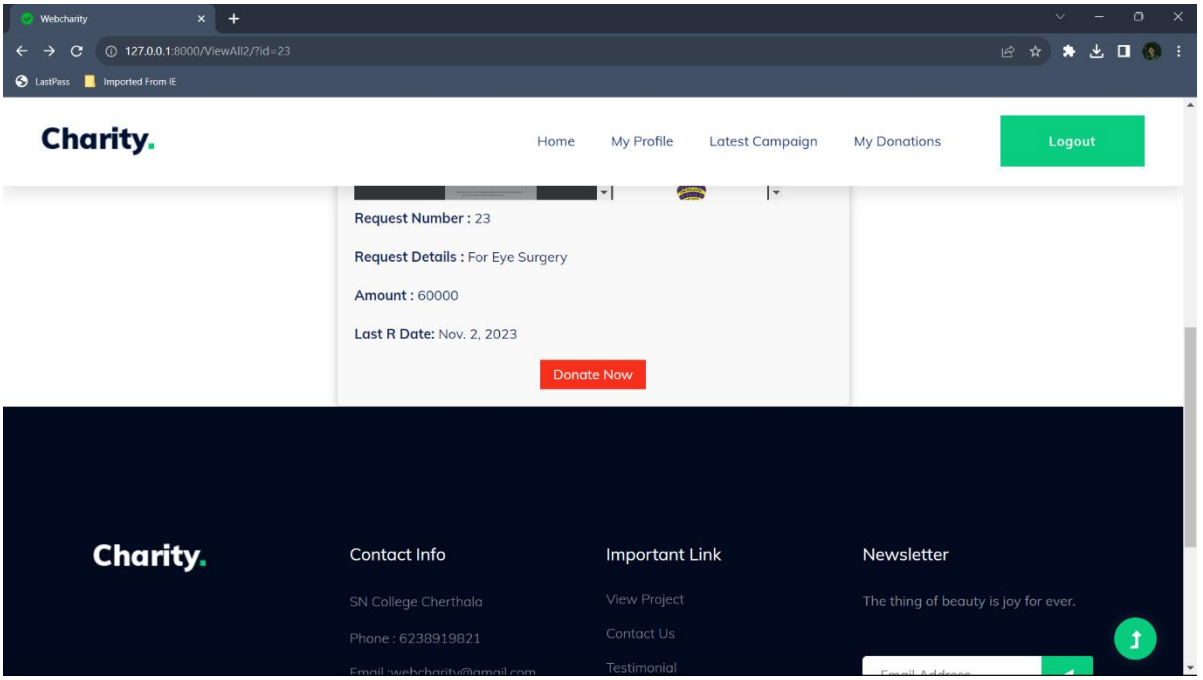


Figure 8.19.3

Payment Form

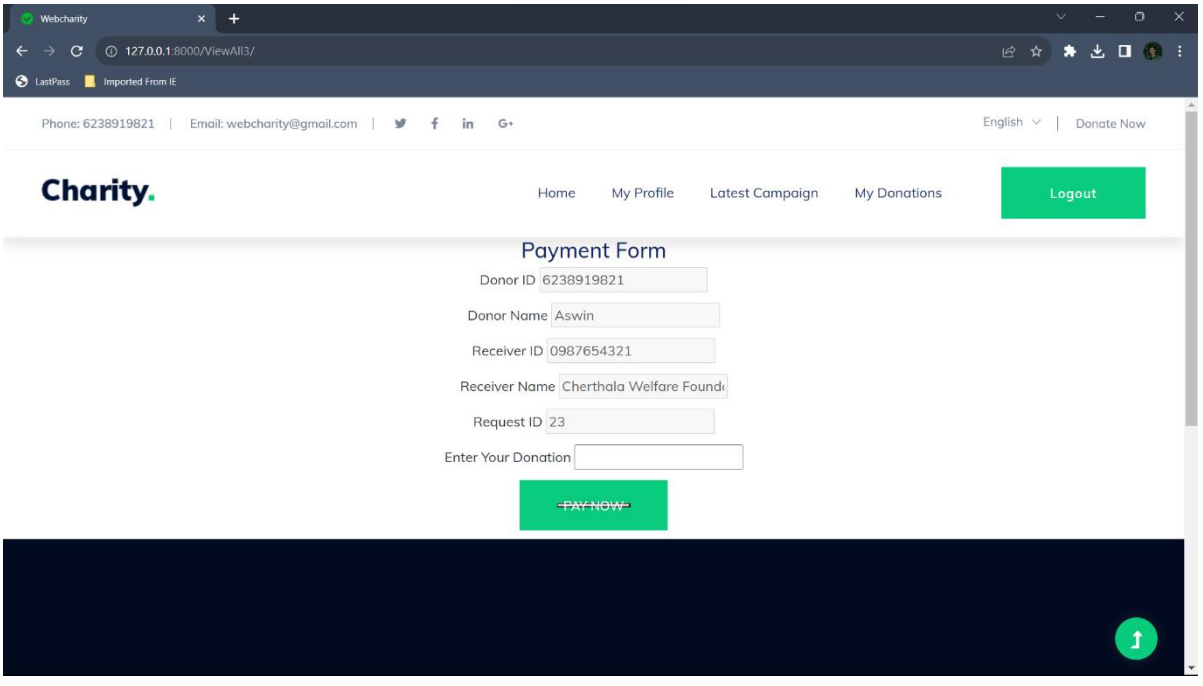


Figure 8.20

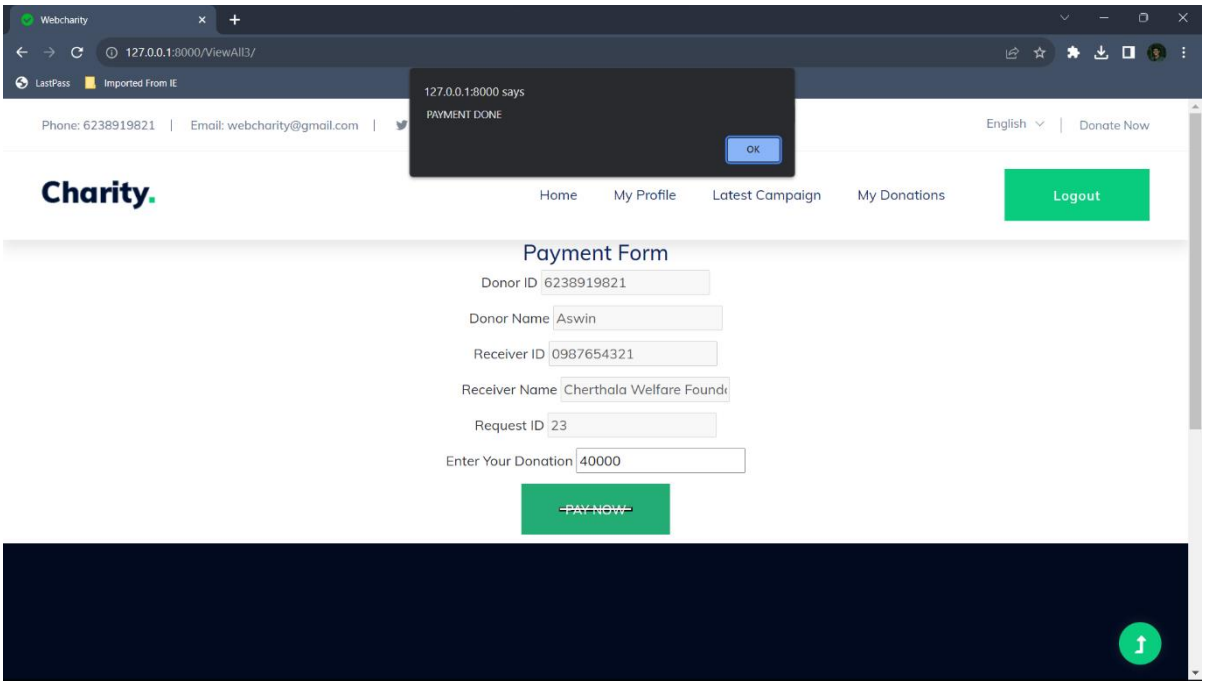


Figure 8.20.1

Donation Details

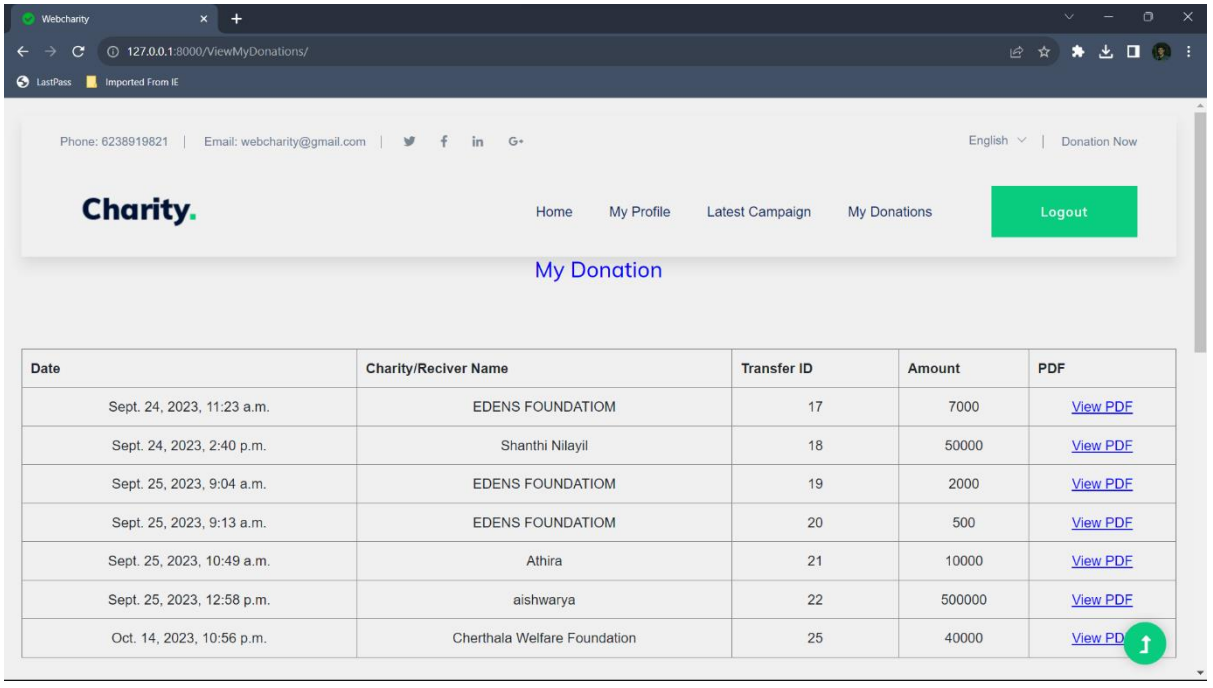


Figure 8.20.2

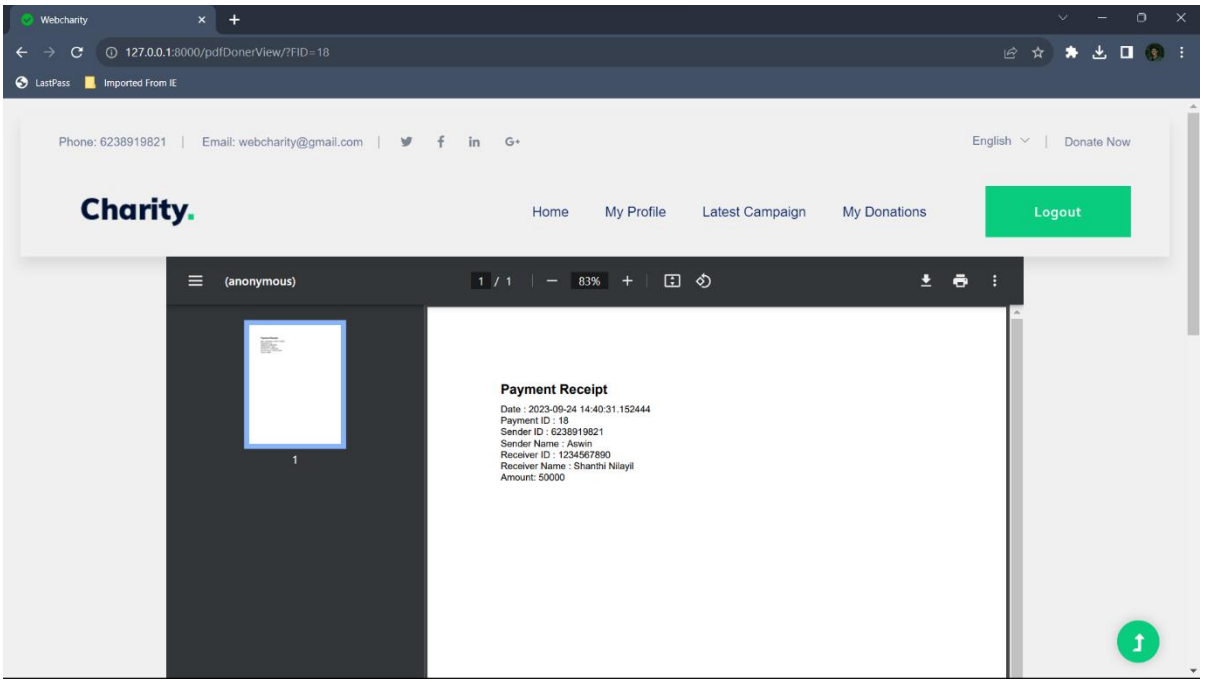


Figure 8.21

Forgot Password

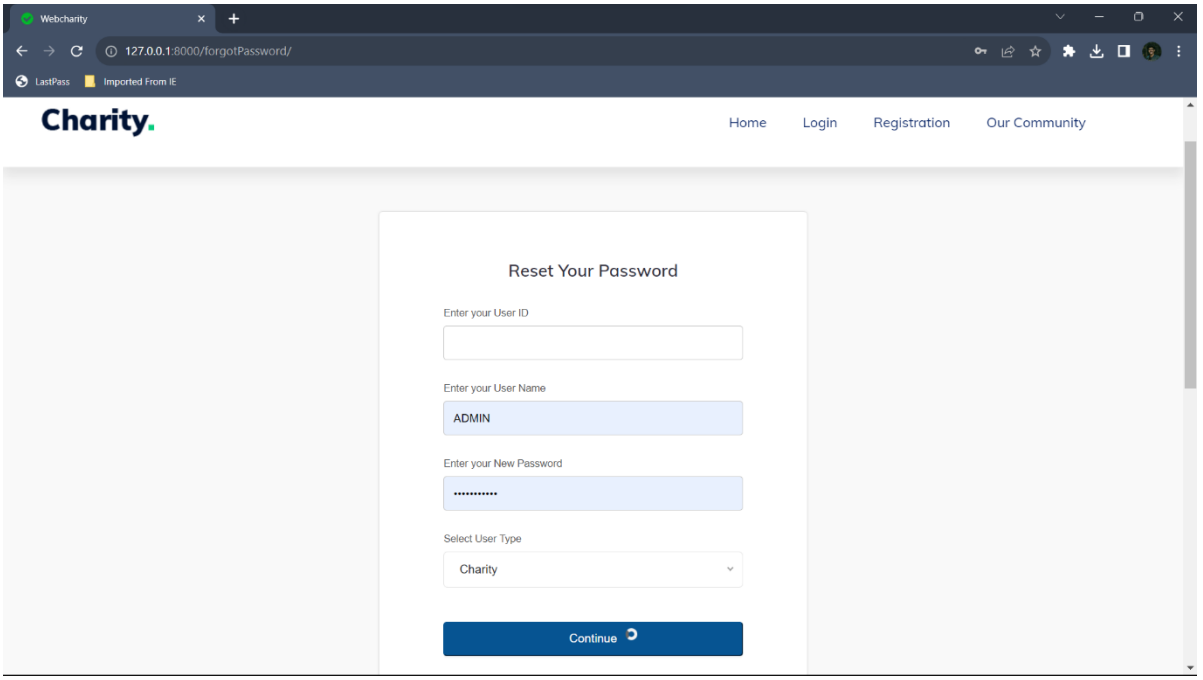


Figure 8.22

9. CONCLUSION

In conclusion, the Charity Management System project represents a significant step forward in enhancing the efficiency, transparency, and impact of charitable organizations. Through the development and implementation of this system, we have successfully addressed several critical challenges faced by charitable organizations, such as donor management, fund allocation, volunteer coordination, and impact assessment.

This project has provided a comprehensive and user-friendly solution for charity management, enabling organizations to streamline their operations, make data-driven decisions, and maximize their ability to fulfill their missions. By centralizing data and automating various processes, the system has not only saved valuable time and resources but has also improved accountability and reduced the likelihood of errors.

Furthermore, the project's adaptability to changing needs and technologies ensures its sustainability in an ever-evolving landscape. It is worth noting that continuous support, updates, and community involvement will be essential in maintaining and expanding the system's capabilities.

As we reflect on this project's journey, it's essential to recognize the collaborative efforts and commitment of the project team, stakeholders, and volunteers who have played a vital role in its success. Their dedication to the cause of improving charitable operations will undoubtedly make a positive impact on the lives of countless individuals and communities.

In closing, the Charity Management System represents a significant milestone in the charitable sector, and its ongoing success will depend on the commitment of all those involved. It is a testament to the potential of technology to empower organizations, increase their efficiency, and ultimately, make a more substantial difference in the world of philanthropy.

10. FUTURE ENHANCEMENT

Our proposed enhancement for the Charity Management System involves leveraging Artificial Intelligence (AI) to gain invaluable donor insights. AI will enable predictive analytics for understanding donor behavior, segmentation for personalized interactions, sentiment analysis for feedback, and recommendations for giving opportunities. Furthermore, AI can predict donor churn, optimize fund allocation, fine-tune fundraising campaigns in real-time, detect fraudulent activities, and provide real-time impact assessments. AI-powered chatbots will offer immediate support and engagement. This advancement will deepen donor relationships, enhance fundraising effectiveness, and bring a new level of data-driven donor-centricity to charitable organizations, maximizing their positive impact.

11. BIBLIOGRAPHY

TEXT REFERENCES

1. Rajib Mall, ‐Fundamentals of software engineering‖, Prentice, Hall of India private Limited, 7 Edition, 2000.
2. Roger S Pressman, ― Software Engineering‖, Pr Roger S Pressman, ‐Software Engineering‖, Practitioners Approach Publishing, Fifth Edition, 2005.
3. Korth F Henry,Silberschatz, Abraham, ‐Database System Concepts‖, McGraw Hill.
4. S A Kelkar, ‐Software Project Management‖, Prentice Hall of India.
5. Silberschatz, Korth, Sudarshan, ‐Database System Concepts‖ ,McGrawHill.
6. ‐Taming PYTHON by Programming‖, Jeeva Jose Khanna Publications.
7. V.K.Jain, Advanced Programming in Web Design, Cyber Tech Publications.
8. Julie C.Meloni, PHP, MySQL and Apache, Pearson Education.
9. Ivan Byross, PHP for Beginners.

12. APPENDIX

- **GANTT CHART**

A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elementsof a project. Terminal elements and summary elements comprise the work break down structureof the project. Some Gantt charts also show the dependency (i.e., precedence network) relationships between activities

Meeting Minute

Date : 04 July 2023

Time : 9:45 AM – 10:15 AM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

04 July 2023

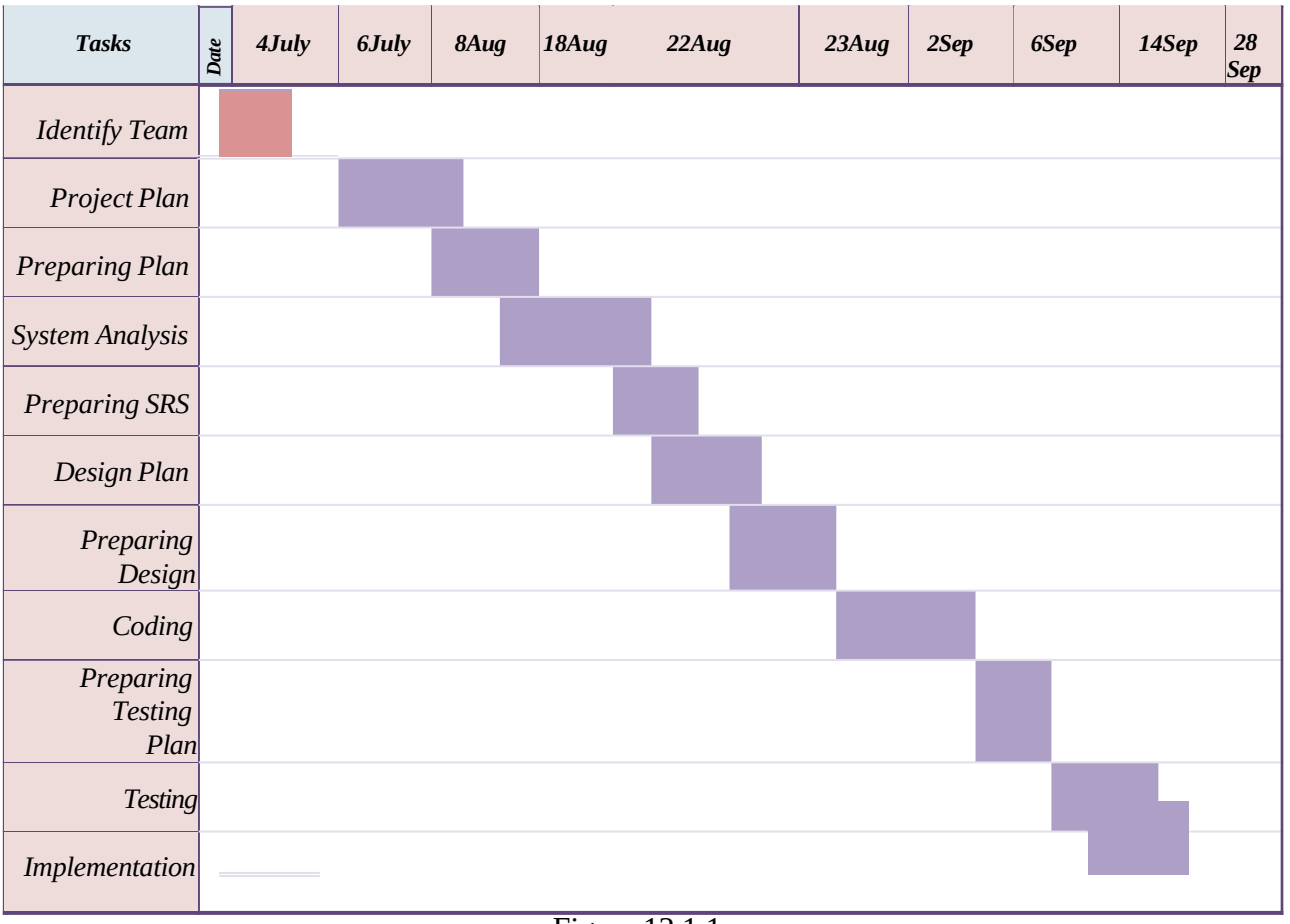


Figure 12.1.1

Meeting Minute

Date : 06 July 2023

Time : 10:30 AM – 11:00 AM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

06 July ***2023***

Figure 12.1.2

Meeting Minute

Date : 08 August 2023

Time : 9:45 AM – 10:15 AM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

08 August 2023

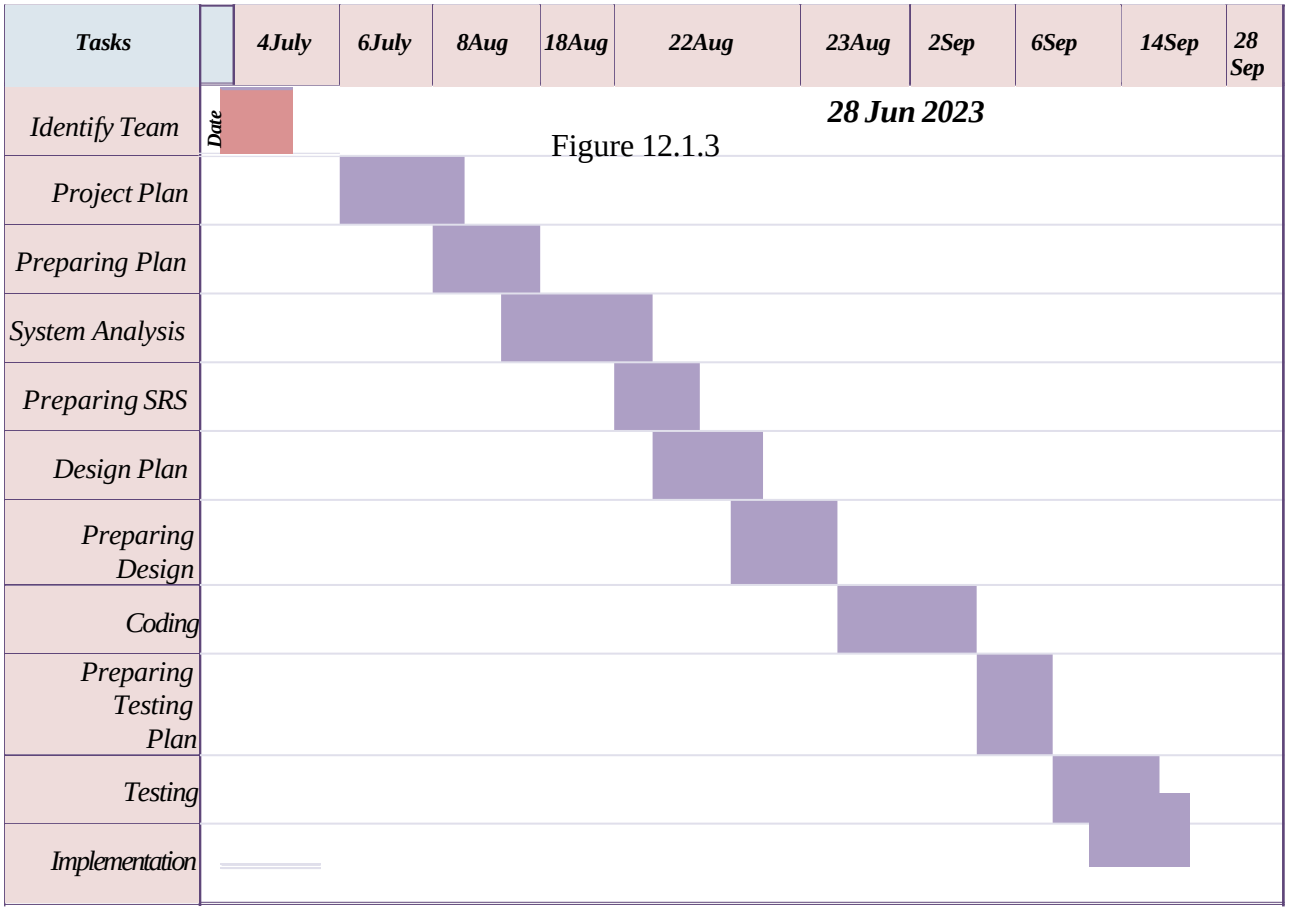


Figure 12.1.3

Figure 12.1.3

Meeting Minute

Date : 18 August 2023

Time : 02:00 AM – 02:30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

18 August 2023

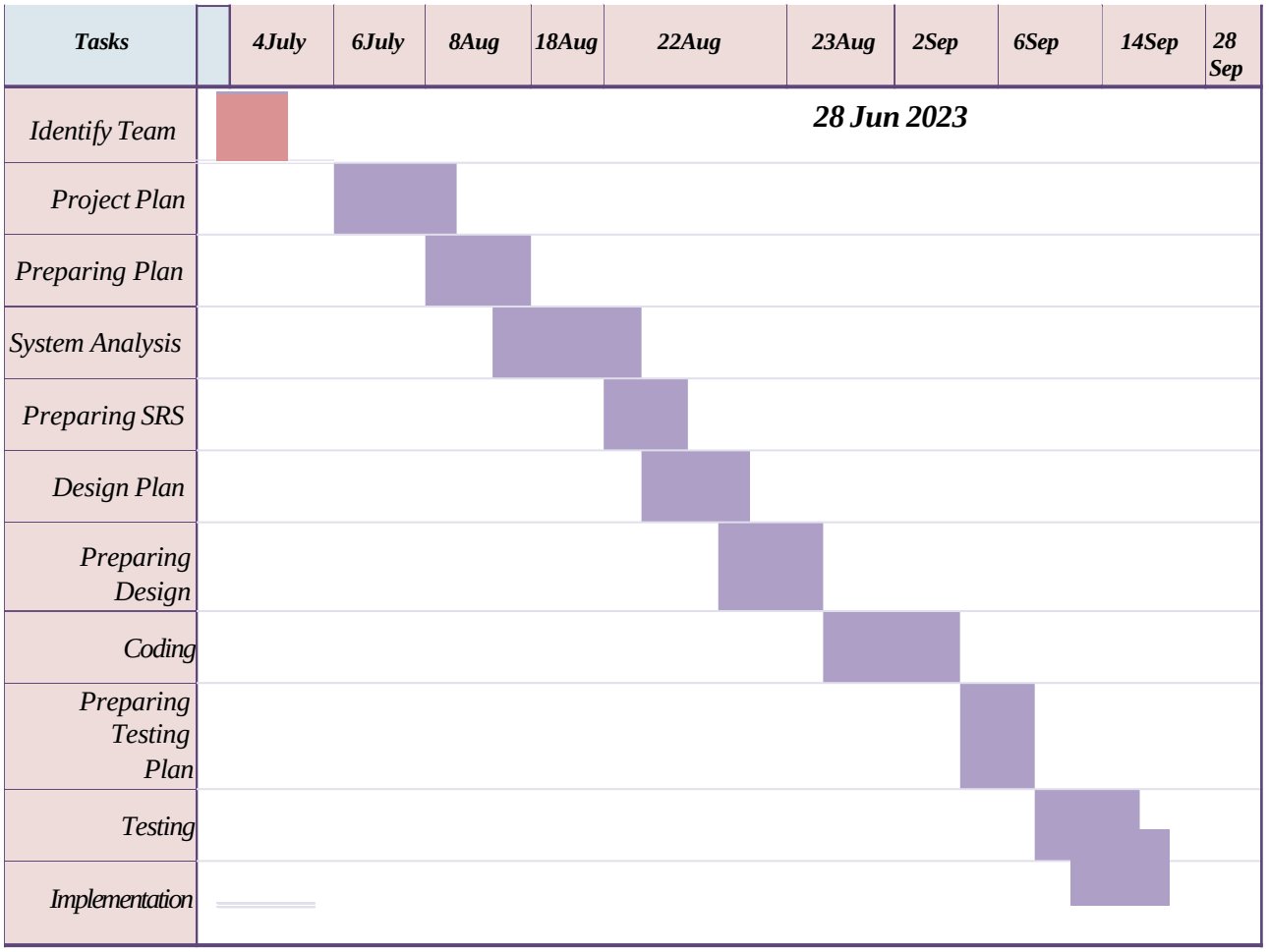


Figure 12.1.4

Meeting Minute

Date : 22 August 2023

Time : 02:00 AM – 02:30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

22 August 2023

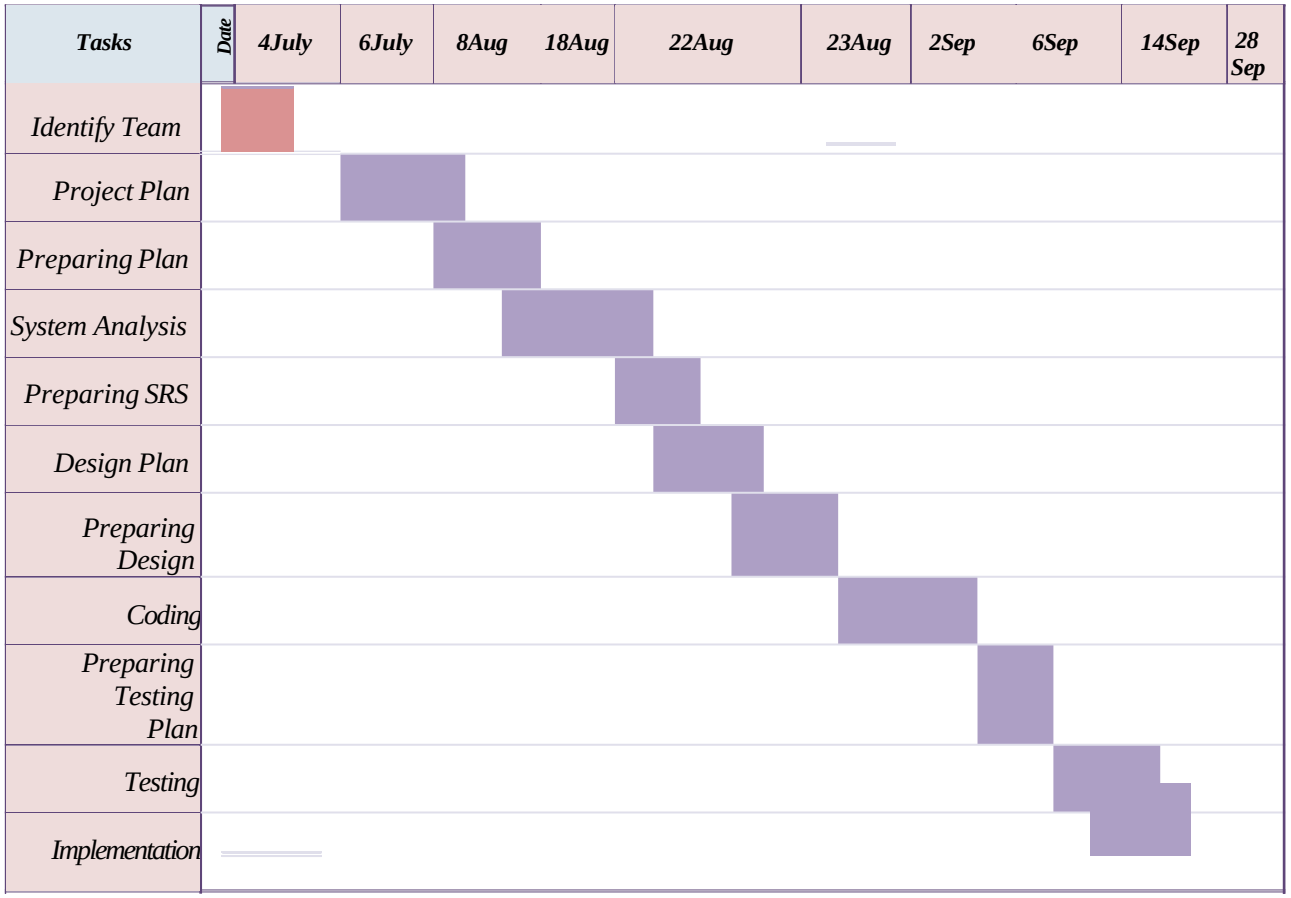


Figure 12.1.5

Meeting Minute

Date : 23 August 2023

Time : 11:00 AM – 11::30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

23 August 2023

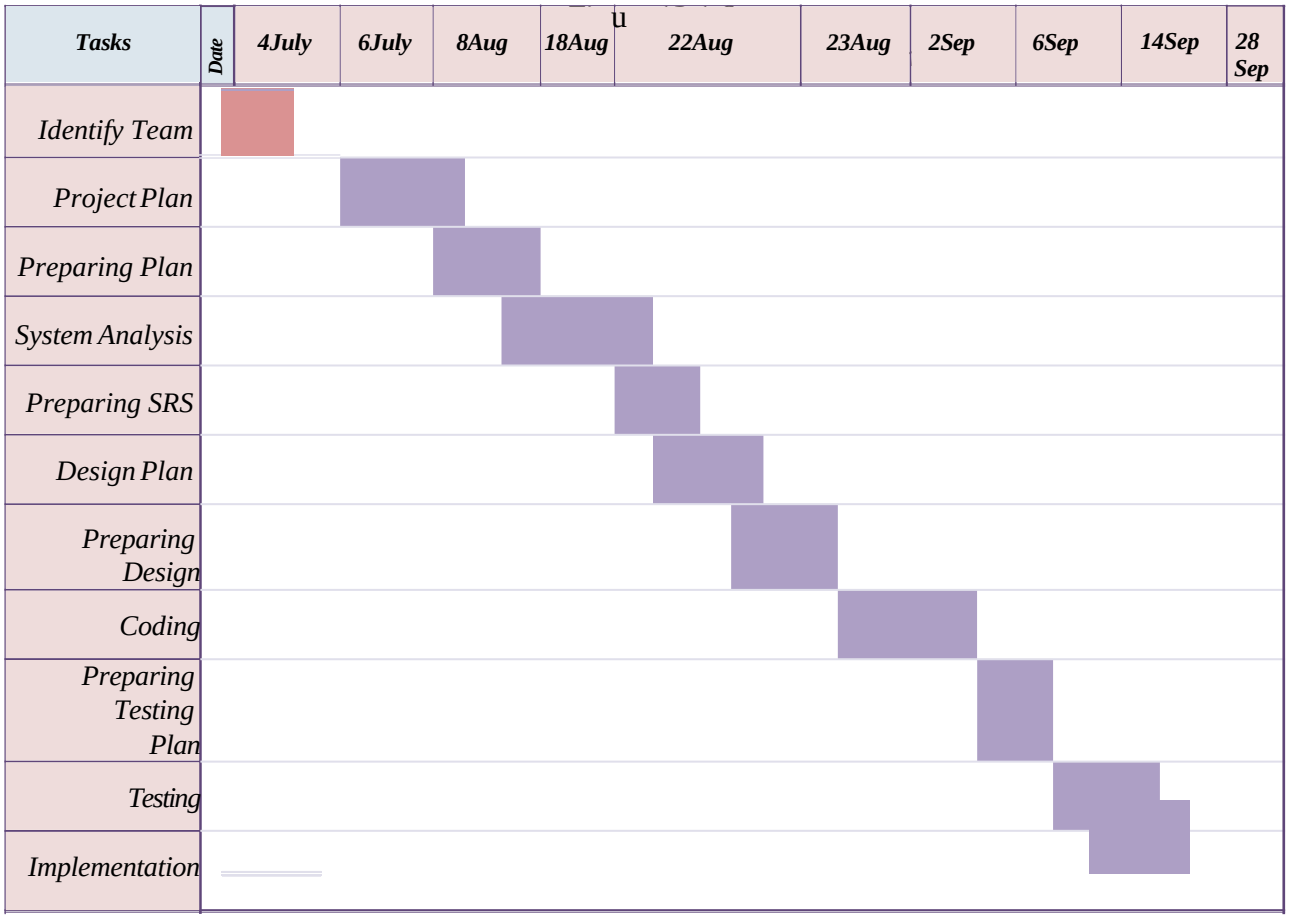


Figure 12.1.6

Meeting Minute

Date : 02 September 2023

Time : 11:00 AM – 11::30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

02 September 2023

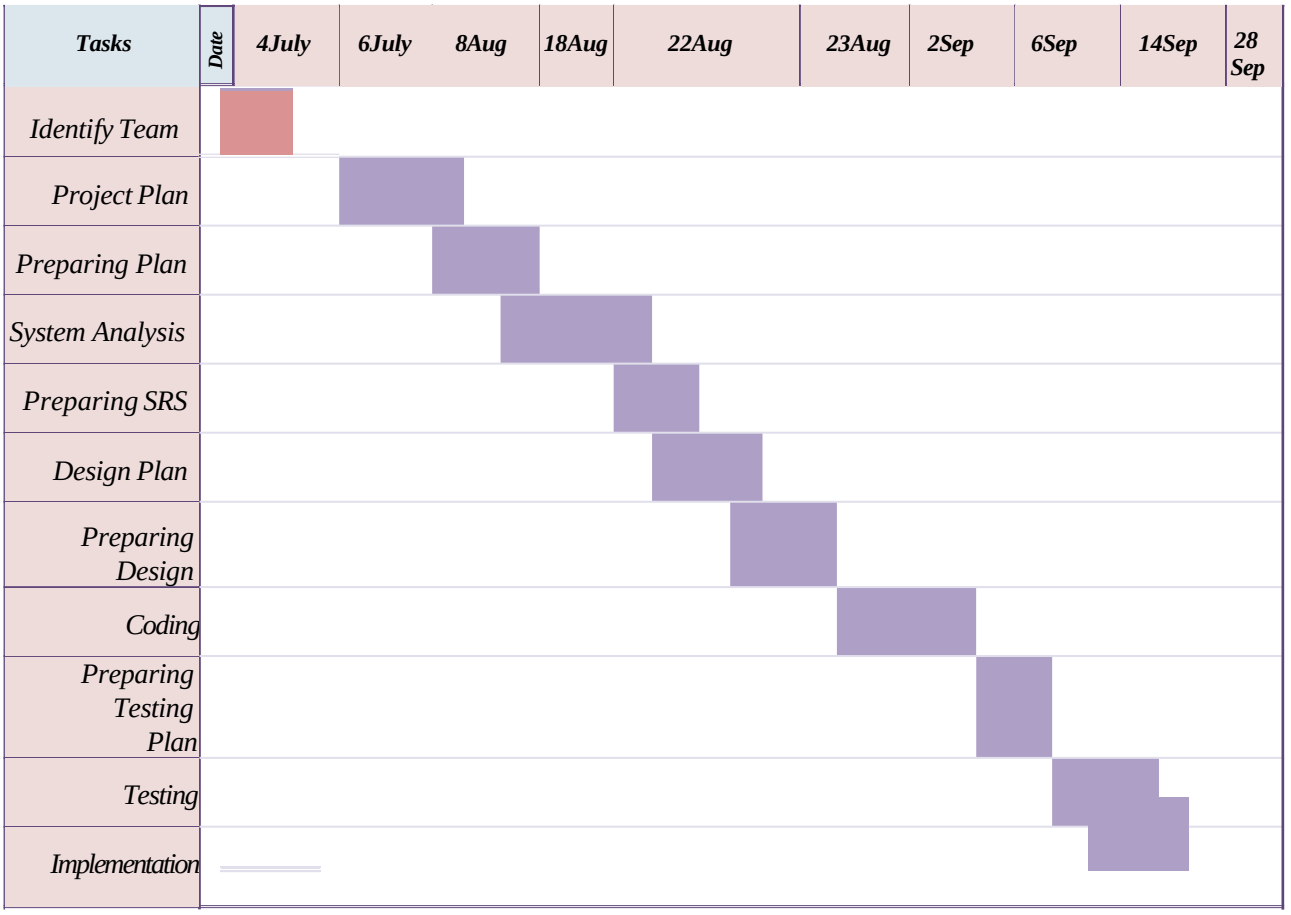


Figure 12.1.7

Meeting Minute

Date : 06 September 2023

Time : 10:00 AM – 10::30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

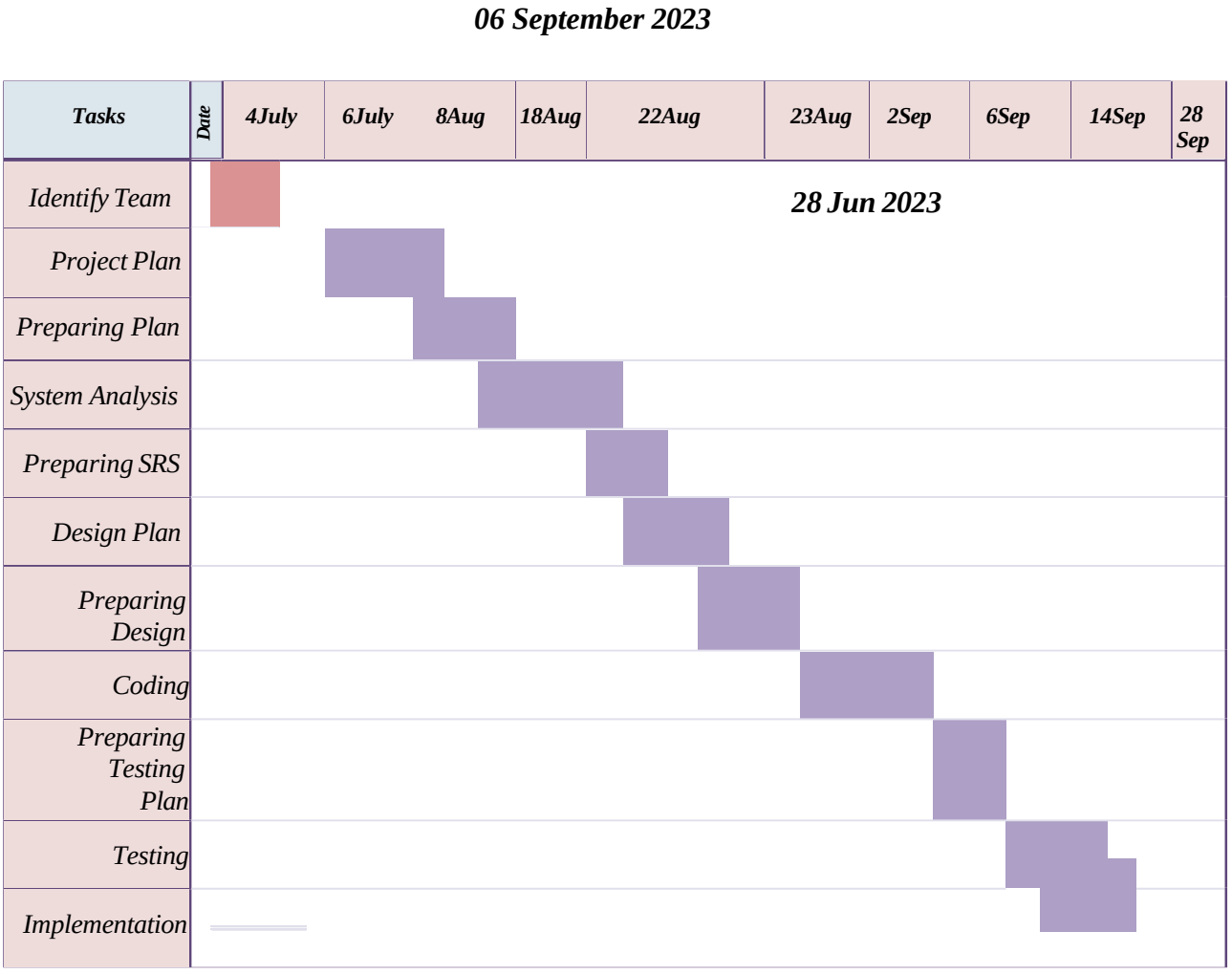

06 September 2023
Tasks
Date
4July
6July
8Aug
18Aug
22Aug
23Aug
2Sep
6Sep
14Sep
28 Sep
28 Jun 2023
Identify Team
Project Plan
Preparing Plan
System Analysis
Preparing SRS
Design Plan
Preparing Design
Coding
Preparing Testing Plan
Testing
Implementation

Figure 12.1.8

Meeting Minute

Date : 14 September 2023

Time : 10:00 AM – 10::30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

14 September 2023

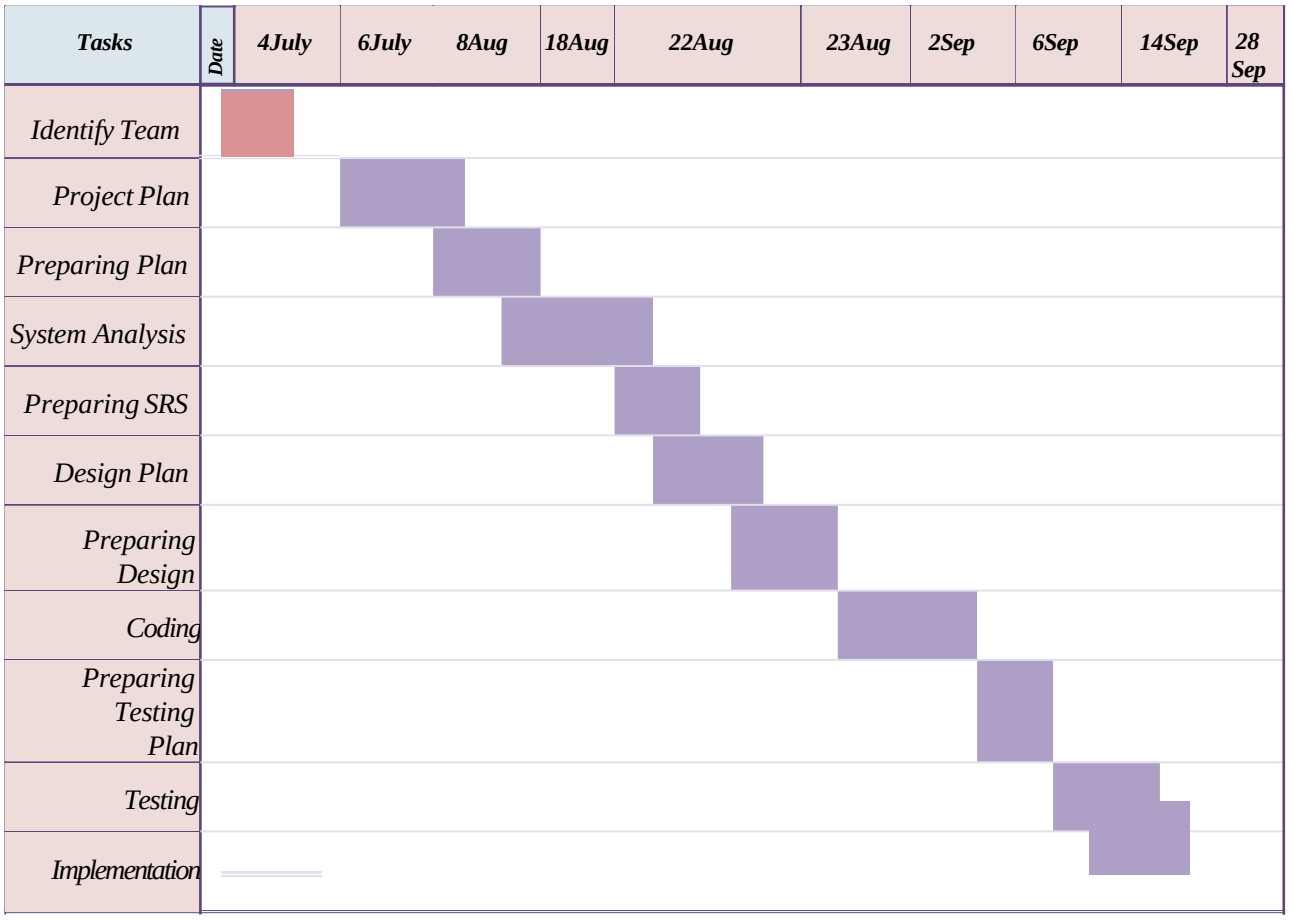


Figure 12.1.9

Meeting Minute

Date : 28 September2023

Time : 10:00 AM – 10::30 PM

Location : Sree Narayana College, Cherthala

Topic : WebCharity

Participants

Guide Name : BINDU N Signature :

Students Name : Signature :

ASWIN SURESH

BINEESH BABU

PRANAV P

VISWAV S

Completion of assigned task

Submitted to Dept. of Computer Science

28 September 2023

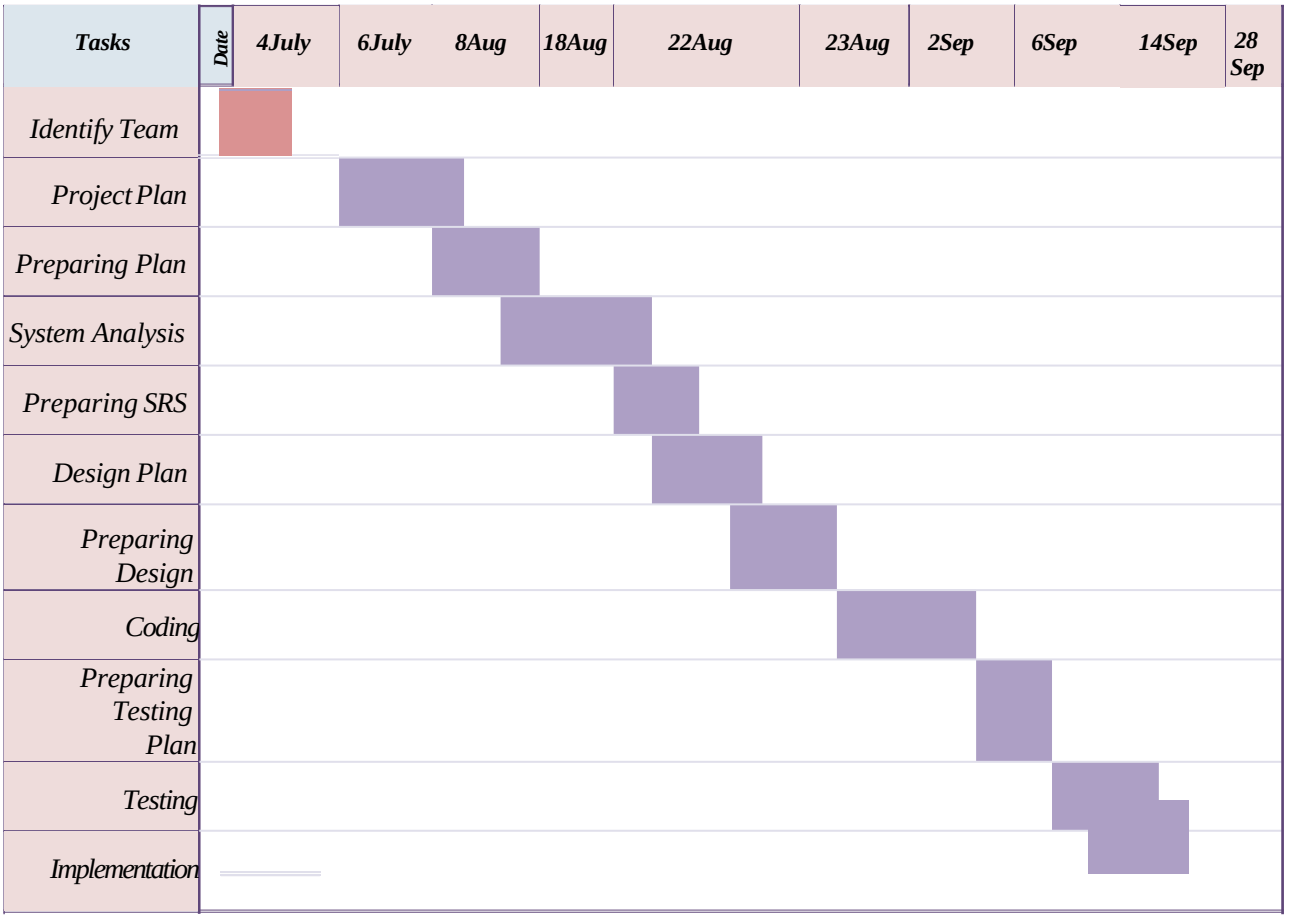


Figure 12.1.10