

ONLINE CHARITY MANAGEMENT SYSTEM



By

Tehreem Khan

Institute of Information Technology

Quaid-i-Azam University

Islamabad Pakistan

April 2021

ONLINE CHARITY MANAGEMENT SYSTEM



By

Tehreem Khan

A report submitted to

The department Of Information Technology, Quaid-i-Azam University

As a partial fulfillment of the requirement

For the award of degree of M.Sc. in Information Technology

April 2021

DEDICATION

This project is dedicated to my Parents who have never failed to give financial and moral support, for giving all need during the time I developed my system and for teaching me that even the largest task can be accomplished if it is done one step at a time. Throughout the preparation and production of this thesis, many people became involved.

Project Brief:

Title: Online Charity Management System

Under Taken By: Tehreem Khan

Internal Supervisor: Ms.Robina Rashid

Starting Date: 21st September, 2020

Completion Date: 29th April, 2021

Development Tools: MS Visual Studio 2019, Microsoft SQL Server
Management Studio 2018, Asp.net

Operating System: Windows 10 pro

System Used: 64 bit operating system, x64-based processor

ABSTRACT

Charity is an act of kindness, where a person who has financially more than enough of what he or she needs to contribute a part of his or her surplus income for the fulfillment of needs of those who are less capable.

In this project, this system has three modules Admin, Agent and People's. Admin adds Agent and then Agent collect data (e.g. total family members, number of male and female, monthly income etc.) who are poor and needy and want charity. There are different types of charity.

- There are some people who access charity only for single time, one who is jobless due to pandemic.
- Second who needs charity every month as there is no earning source.

Agent collects information about the people who wants to donate. Then the merit will be calculated according to the number of people and donation amount.

ACKNOWLEDGEMENT

It was very exciting for me to work on the project of ONLINE CHARITY MANAGEMENT SYSTEM. During this work I have gained both practical as well as theoretical knowledge of great significance. I express my gratitude and thanks to Ms. Robina Rashid for being a source of strength, inspiration, and helped me to surmount the obstacles in my way to the successful completion of my challenging work. I thank to my parents and few people indirectly who helped me for this project work and my friends for their enthusiasm and timely help during this academic year 2019-2021.

Preface

This report contains detailed description, analysis, design and implementation of system along testing. Though the iterative and linear strategy is followed first to last step, each stage is represented in the form of chapter in this report.

Chapter 1 contains the introduction, background which describes what exactly the online charity management system is about. It also contains problem statement, scope and objectives. Then come Chapter 2 which includes system overview and requirement specification, a review to the current system, requirements of a new system. Requirements are further classified into user requirements. Chapter 3 includes system design in which activity diagram and sequence diagram represents how the system is working. Chapter 4 covers all the aspects of implementation, selection of language, and overview to ASP.NET as well. Chapter 5 contains interfaces of the new system which has been developed. Chapter 6 includes system testing in which black box technique is used by making test cases of developed system and chapter 7 includes the conclusion and future work.

Table of Contents:

CHAPTER 1: Introduction	Page no
1.1 Introduction	10
1.2 Problem Definition	10
1.3 Motivation	10
1.4 Problem Solution	10
1.5 Goals and Objectives	11
1.6 Scope	11
1.7 Constraints	11
1.8 Summary	12
CHAPTER 2: System Requirements and Analysis	
2.1 Introduction	13
2.2 User Requirements	13
2.2.1 Functional Requirements	13
2.2.2 Non Functional Requirements	15
2.3 Appendices	17
2.3.1 Introduction	17
2.3.2 Purpose	17
2.3.3 Advantages	17
CHAPTER 3: System Design	
3.1 Introduction	18
3.2 Use case Diagram	19
3.2.1 Use cases	20
3.3 Activity Diagram	27
3.3.1 Activity Diagram for Admin Login	28
3.3.2 Activity Diagram for Agent Login	29
3.4 Sequence Diagram	30
3.4.1 Sequence Diagram for Admin Login	31
3.4.2 Sequence Diagram for Agent Login	32
3.4.3 Sequence Diagram for Entire process	33
3.5 Entity Relationship Diagram (E.R.D)	34

3.6 Class Diagram	36
-------------------	----

CHAPTER 4: Tools and Technology

4.1 Introduction	38
4.2 Tool and Language Selection	38
4.2.1 Visual Studio	38
4.2.2 ASP.NET	39
4.2.3 Html	39
4.2.4 CSS	39
4.2.5 Bootstrap	39
4.2.6 SQL	40
4.2.7 Microsoft Visio	40

CHAPTER 5: Interface

5.1 Introduction	41
5.2 Admin Login Page	42
5.3 Admin Panel Dashboard	43
5.4 Add Admin/Agent Section in Admin Panel	44
5.5 Data of Admin/Agent in Admin Panel	45
5.6 Add Donor in Admin Panel	46
5.7 Displaying Data of Donor in Admin Panel	47
5.8 Add Permanent Recipient in Admin Panel	48
5.9 Displaying Data of Permanent Recipient in Admin Panel	49
5.9 Add Single Recipient in Admin Panel	50
5.10 Displaying Data of Single Recipient in Admin Panel	51
5.11 Agent Login Page	52
5.12 Add Donor in Agent Panel	53
5.13 Displaying Data of Donor in Agent Panel	54
5.14 Add Permanent Recipient in Agent Panel	55
5.15 Displaying Data of Permanent Recipient in Agent Panel	56
5.16 Add Single Recipient in Agent Panel	57
5.17 Displaying Data of Single Recipient in Agent Panel	58
5.18 Agent Logout	59

CHAPTER 6: System Testing

6.1 Introduction	60
6.2 Software Testing Types	60
6.3 Testing Approaches	60
6.4 Test Cases	61
6.1 Test Case of Admin Login	61
6.2 Test Case of Dashboard of Admin Panel	61
6.3 Test Case of Add Agent in Admin Panel	62
6.4 Test Case of Add Permanent Recipient in Admin Panel	62
6.5 Test Case of Add Single Recipient in Admin Panel	63
6.6 Test Case of Add Donor in Admin Panel	63
6.7 Test Case of Agent Login	63
6.8 Test Case of Add Permanent Recipient in Agent Panel	64
6.9 Test Case of Add Single Recipient in Agent Panel	64
6.10 Test Case of Add Donor in Agent Panel	65

CHAPTER 7: Conclusion and Future Work

7.1 Future Work	66
7.2 Conclusion	67
7.3 References	68

Chapter 1: Introduction

1.1 Introduction

The core purpose of the project is to introduce a site for needy people. This site is basically designed for charity purpose that will be greatly beneficial for poor people So that they can easily fulfill their needs. This kind of charity website provides a good platform for both, the donors and needy people.

1.2 Problem Definition

The main purpose of giving charity is to fulfill the needs of poor people who are the victims of war, natural disaster, hunger and poverty but some people in our society request for charity that are not eligible for it and show them poor. These people deprive the rights of poor people. Secondly, the people who donate for charity are afraid that their charity amount reaches the needy people or not.

1.3 Motivation

This system will be able to handle many services to take care of all people records in a quick manner. This system will save time and paper work and will make the record more systematic and disciplined. In this system the poor people will get the charity amount after the completion of process. Similarly, the people who are willing to donate for charity can donate the desired amount online.

1.4 Problem Solution

To find the solution of defined problems in the above section we have a special customer care service which is available 24/7. This group ensures the safety and security of the people and their amount that it reaches the desired people. The solution to the second problem is that the people who donate the amount will be informed through text messaging and people can track it online as well.

1.5 Goals and Objectives

- Corporate between the data stored in the server of department and online system to deal with online system in an easy way
- Create strong and secret data base that allow for any concern in a secret way, to prevent any outside or inside attack.
- Specify a privilege for each person that allows each person use this system to create his own account and have a complete record of his charity amount.
- Allow each person to create more than one account so that if more than one family member wants to donate and wants his record, he can do it as well.

1.6 Scope

- Online Charity Management System is designed for needy peoples who cannot fulfill their needs
- This site will help all people as the people who want to donate for charity can donate easily without facing any kind of hurdles and the recipients can easily get the charity.
- The charity donated to poor people will help in reducing poverty from the country.

1.7 Constraints

Constraints are the terms and conditions to the system that must be kept in mind when using the system. In this system there are also some constraints that must be kept in mind before using this project.

- Agent can create account
- During creation agent has to pick a password
- Data saved on the server
- Access of data is through the login ID of the agent

1.8 Summary

The main reason of introducing this website is to allow the people who want to donate for charity can donate easily without facing any kind of hurdles. This site will help all people. It will give a better way to donate charity, understanding to resolve the issues like administrative and departmental. Every person will get his/her Personal ID number to log in to the account where every solution will be present.

Chapter 2: System Requirement & Analysis

2.1 Introduction

Requirements and analysis of the requirement for this website are discussed in this chapter. The main purpose of requirement gathering and specification is to specify the requirements of the system.

The scope and limitation of this system is:

- The online charity system design to welfare institutions
- Hold all operations and generate reports to admin
- Allow the people to see their mark
- Verify security, authority and safety

2.2 User Requirements

The user requirements for this system are to make the system flat, flexible, less prone to error, reduce expenses and save time.

- Time can be saved by online transfer that can be performed by click of button
- A facility to generate a result chart as pre required without manual interface

2.2.1 Functional Requirements

Functional requirements are those functionalities that software/project should perform. Functional requirements specify the work of system that it performs. These requirements are necessary to run a system smoothly and make them functional.

This section gives a functional requirement that is applicable to Online Charity Management System. There are for sub modules in this place.

- Admin module
- Agent module
- Donor module
- Recipient module

MODULES FUNCTIONALITY

Admin module: The admin module allows managing members and services.

Agent module: The agent module is a person hired legally on behalf of OCMS. He collects information from the donor and recipient.

Donor module: The donor module is the one who donates the money for charity.

Recipient module: The recipient module is the one who is going to receive the money of charity.

Admin Feature

Admin has full access over the OCMS.

Admin can create/delete an account.

Admin can change the password.

Admin view dashboard that has complete details about agents, donors and recipients etc.

Admin can add accounts of agents and make them active or inactive.

Admin also has a record of the amount donated by every donor.

Admin can remove any recipient from any base.

Admin can see the detail of the recipient which is approved.

Admin can see the detail of the amount every month has been deposited, how much amount has been paid to the recipient and the remaining amount as well.

Agent Features

Agent can view its profile and modify it.

Agent can modify the profile of donors and recipients.

Agent can remove any donor from any base.

Agent can remove any recipient from any base.

Agent can access to the details of charity

Agent can see the detail of the amount every month has been deposited, how much amount has been paid to the recipient and the remaining amount as well.

Donor Features

Donor donates the money for charity.

Recipient Features

Recipient receives the money of charity.

2.2. Nonfunctional Requirements

Performance Requirements

Some Performance requirements identified are listed below:

- The database shall be able to accommodate a minimum of 10,000 records of People
- The software shall support use of multiple users at a time

Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

Software Quality Attributes

The Quality of the System is maintained in such a way so that it can be very user friendly to all the users.

The software quality attributes are assumed as under:

- Accurate and hence reliable
- Secured

- Fast speed
- Compatibility

System Interfaces:

This section describes how the software interfaces with other software products or users for input or output.

User Interface

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with Microsoft Internet Explorer for version 6 and above. No user would be able to access any part of the application without logging on to the system.

Hardware Interfaces

Server Side:

- Operating System: Windows 9x/xp, Windows ME
- Processor: Pentium 3.0 GHz or higher
- RAM: 256 Mb or more
- Hard Drive: 10 GB or more

Client side:

- Operating System: Windows 9x or above, MAC or UNIX.
- Processor: Pentium III or 2.0 GHz or higher.
- RAM: 256 Mb or more

Software Interfaces

- **Client Side:** .HTML, Web Browser, Windows XP/2000/Vista
- **Web Server:** .HTML, Windows XP/2000/Vista

Communications Interfaces

The Customer must connect to the Internet to access the Website:

- Dialup Modem of 52 kbps
- Broadband Internet
- Dialup or Broadband Connection with Internet Provider

Appendices:

Definition of Online Charity Management System

Introduction:

Online Charity Management System is a software application which allows a particular institution to arrange, conduct and donate charity online.

Purpose:

The purpose of this system is to conduct various types of certificate/non-certificate charity programs at different centers across any country via online.

Advantages:

- Today, most of the companies are shifting to digital world to be a part of this fastest growing world
- Online Charity Management System covers almost all types of problems faced
- The charity donated to the poor people will help in reducing poverty from the country

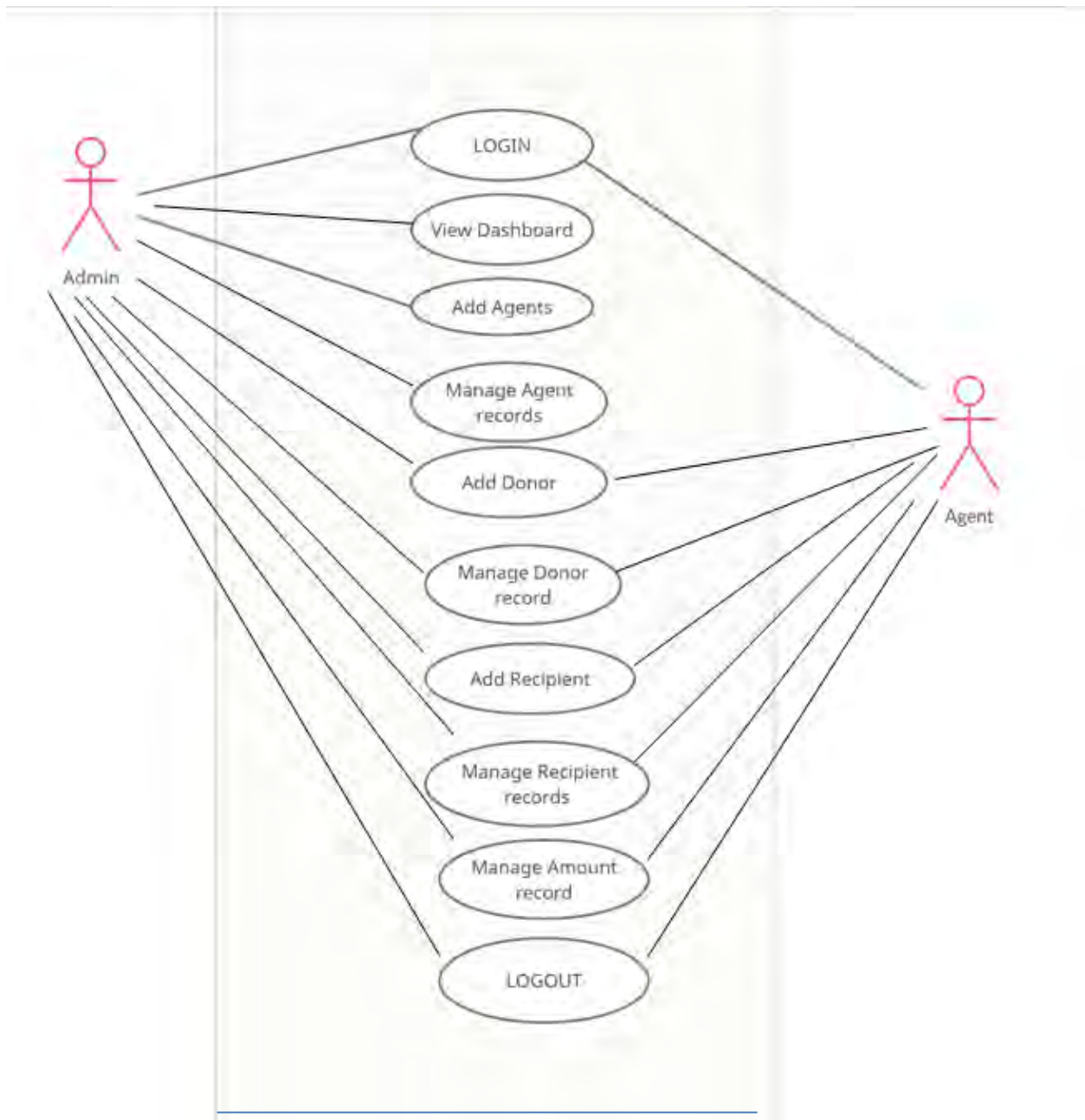
Chapter 3: System Design

3.1 Introduction

To understand any system in a better way we need the help of different diagrams of the system. With these diagrams, we can understand the system more quickly than reading all the documents about the system. In this chapter, some of these diagrams are designed which includes use-case diagram, Activity Diagram, Entity relation diagram (ERD), Class diagram.

3.2 Use-case Diagram

In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system. Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering.



Use Case Diagram for Admin and Agent

Table 3.1: Use case of Login

ID	Use case 1
Primary Actors	Admin/Agent.
Input	Login id and Password.
Pre-Condition	Admin/Agent is registered and information is stored in a database.
Post-Condition	Logged in successfully.
Main success Scenario	System display log in screen having two input fields for Registration Login id and Password. <ul style="list-style-type: none">• Admin/Agent Enter Login id and Password.• Admin/Agent clicks on Submit button.• System Validate Login id and Password.• Logged in successfully and Home screen is displayed.
Alternative flow or Extensions	Incorrect ID and Password. <ul style="list-style-type: none">• Login id is valid but information not stored in database.• System shows error message that invalid ID or password.

Table 3.2: Use case of Active Status

ID	Use case 2
Primary Actor	Agent
Input	None
Pre-Condition	1-Logged in the system. 2-Enable the internet connection. 3-Active on Home screen.
Post-Condition	• (Able to active/inactive) Status of agent.
Alternative flow or Extensions	•Unable to change status.

Table 3.3: Use case of Manage Agent Records

ID	Use case 3
Primary Actor	Admin
Input	Agent record include <ul style="list-style-type: none">•ID•Name•Contact•Email•Address•Password•Role id•Active status
Pre-Condition	<ul style="list-style-type: none">•Actor(Admin/Agent)•Logged in the system.•Enable the internet connection.•Active on Home screen.
Post-Condition	Able to edit, update and delete Agent Record.
Main Success Scenario	System display the input fields for option" Manage Agent Record" and Actor (Admin) can <ul style="list-style-type: none">• View the Record of all the agents.• Able to update the record of agents.• Able to add new record of agents.
Alternative flows	Actor (Admin) updated /deletes the record but not change status on database.

Table 3.4: Use case of Single Recipient Records

ID	Use case 4
Primary Actors	Admin/Agent
Input	Single Recipient records include <ul style="list-style-type: none">• ID•Name•Phone•Address•City•Monthly income•Emergency description•Active status
Pre-Condition	<ul style="list-style-type: none">•Actor(Admin/Agent)•Logged in the system.•Enable the internet connection.•Active on Home screen.
Post-Condition	Able to edit, update and delete Single recipient Record.
Main Success Scenario	System display the input fields for option” Manage Single Recipient Record” and Actor (Admin/Agent) can <ul style="list-style-type: none">• View the Record of the entire single recipient.• Able to update the record of single recipient.• Able to add new record of single recipient.
Alternative flows	Actor (Admin/Agent) updated /deletes the record but not change status on database.

Table 3.5: Use case of Permanent Recipient Records

ID	Use case 5
Primary Actor	Admin/Agent
Input	Permanent Recipient record includes <ul style="list-style-type: none"> • ID •Name •Phone •Address •City •Family Members •Monthly Income •Health expenditure •Food expenditure •Study expenditure •Total expenditure •Total income •Active status
Pre-Condition	<ul style="list-style-type: none"> •Actor(Admin/Agent) •Logged in the system. •Enable the internet connection. •Active on Home screen.
Post-Condition	Able to edit, update and delete Permanent recipient Record.
Main success Scenario	System display the input fields for option” Manage Permanent Recipient Record” and Actor (Admin/Agent) can <ul style="list-style-type: none"> • View the Record of the entire permanent recipient. • Able to update the record of permanent recipient. • Able to add new record of permanent recipient.
Alternative flows	Actor (Admin/Agent) updated /deletes the record but not change status on database.

Table 3.6: Use case of Donor Records

ID	Use case 6
Primary Actor	Admin/Agent
Input	Donor record includes <ul style="list-style-type: none">• ID• Name• Contact• Address• City• Gender• Occupation• Amount• Donation date• Active status
Pre-Condition	<ul style="list-style-type: none">• Actor(Admin/Agent)• Logged in the system.• Enable the internet connection.• Active on Home screen.
Post-Condition	Able to edit, update and delete Donor Record.
Main success Scenario	System display the input fields for option” Manage Donor Record” and Actor (Admin/Agent) can <ul style="list-style-type: none">• View the Record of all the donors.• Able to update the record of donors.• Able to add new record of donors.
Alternative flow	Actor (Admin/Agent) updated /deletes the record but not change status on database.

Table 3.7 Use Case of Add/Delete Recipient Record

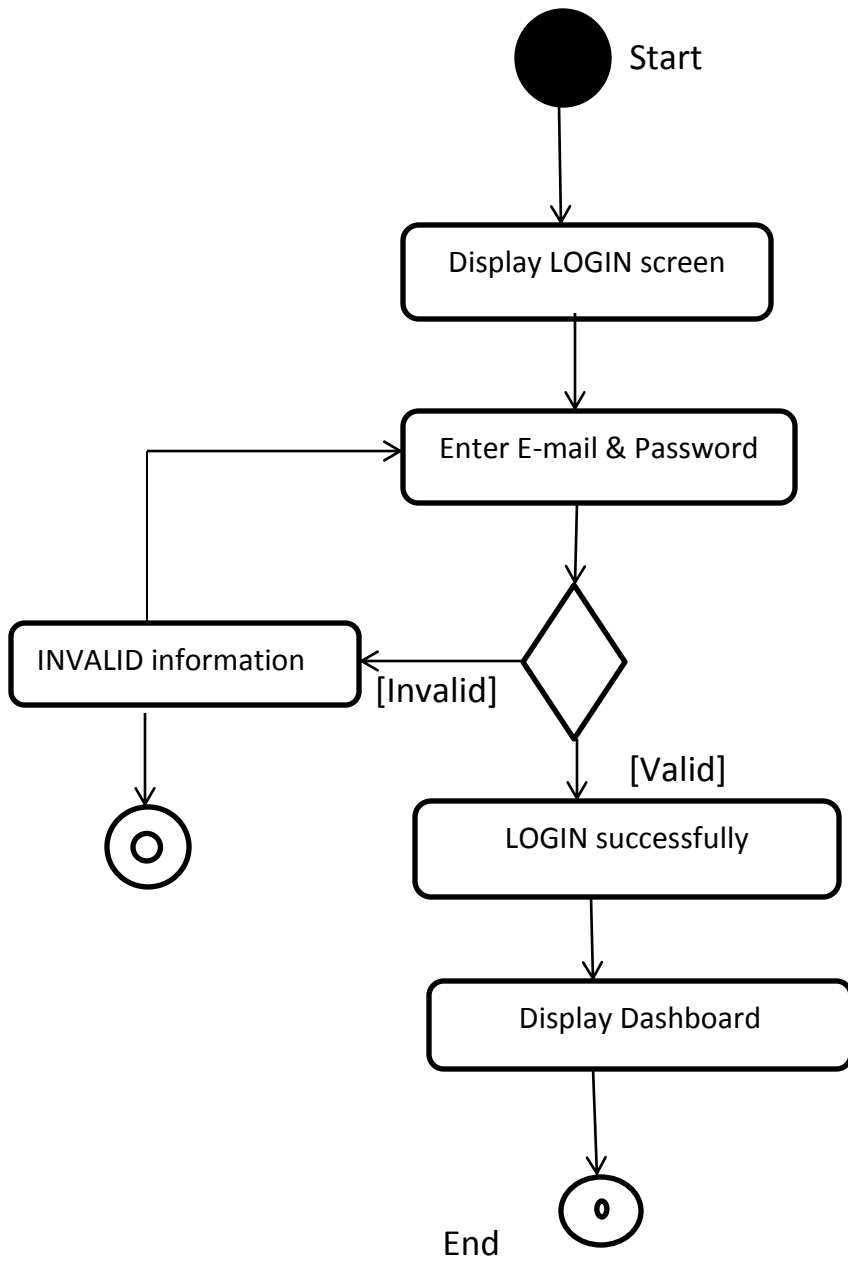
ID	Use case 7
Primary Actor	Admin/Agent
Input	None
Pre-Condition	<ul style="list-style-type: none">•Logged in the system.•Enable the internet connection.•Active on Home screen.
Post-Condition	1-Able to check recipient record. 2-Add any recipient. 3-Delete any recipient
Alternative flow or Extensions	Unable to view, add or delete recipient record.

Table 3.8 Use Case of Logout

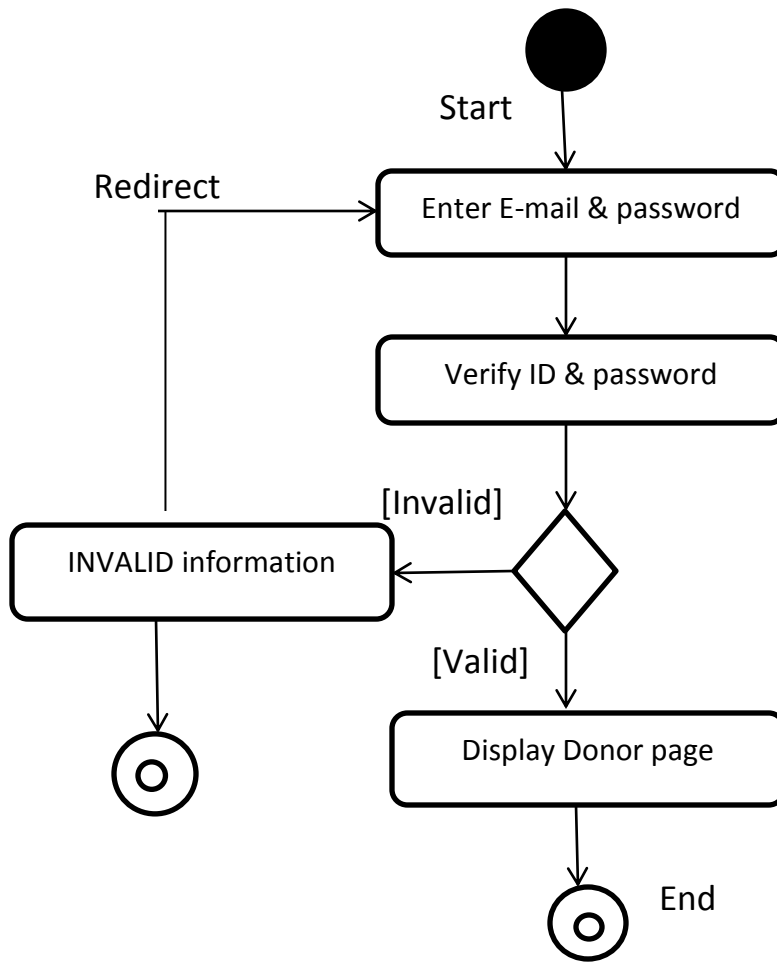
ID	Use case 8
Primary Actors	Admin/Agent
Input	None
Pre-Condition	User must have login to system.
Post-Condition	When user click on logout button then destroy the session and user logout from their account.
Alternative flows or Extensions	Actor click on logout button but unable to logout from account.

3.3 Activity Diagram

Activity diagram is another significant diagram in UML to define the vibrant features of the system. Activity diagram is mainly a flowchart to symbolize the flow from one activity to another activity. Activity by itself can be termed as an operation of the system. The control flow is drawn from one operation to another. This flow can be progressive, diverged, or parallel. Activity diagrams deal with all type of flow control by using diverse elements such as fork, join, etc.



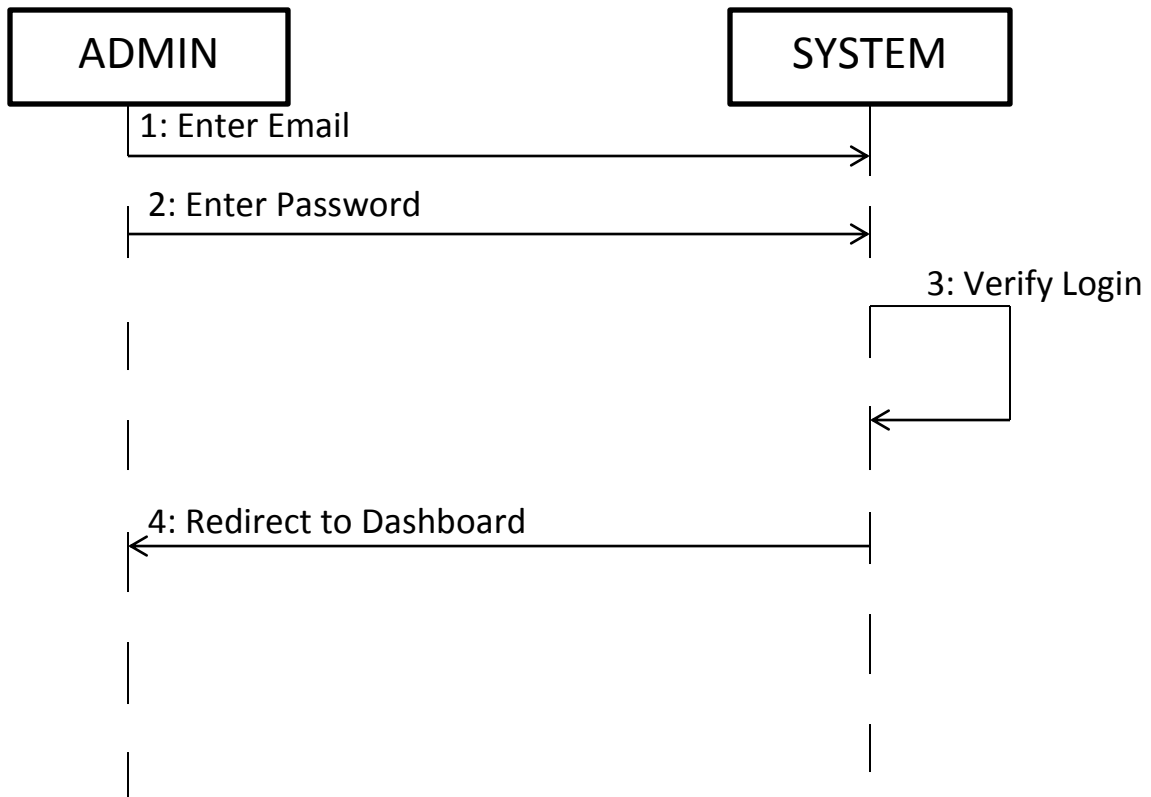
Activity Diagram for ADMIN login



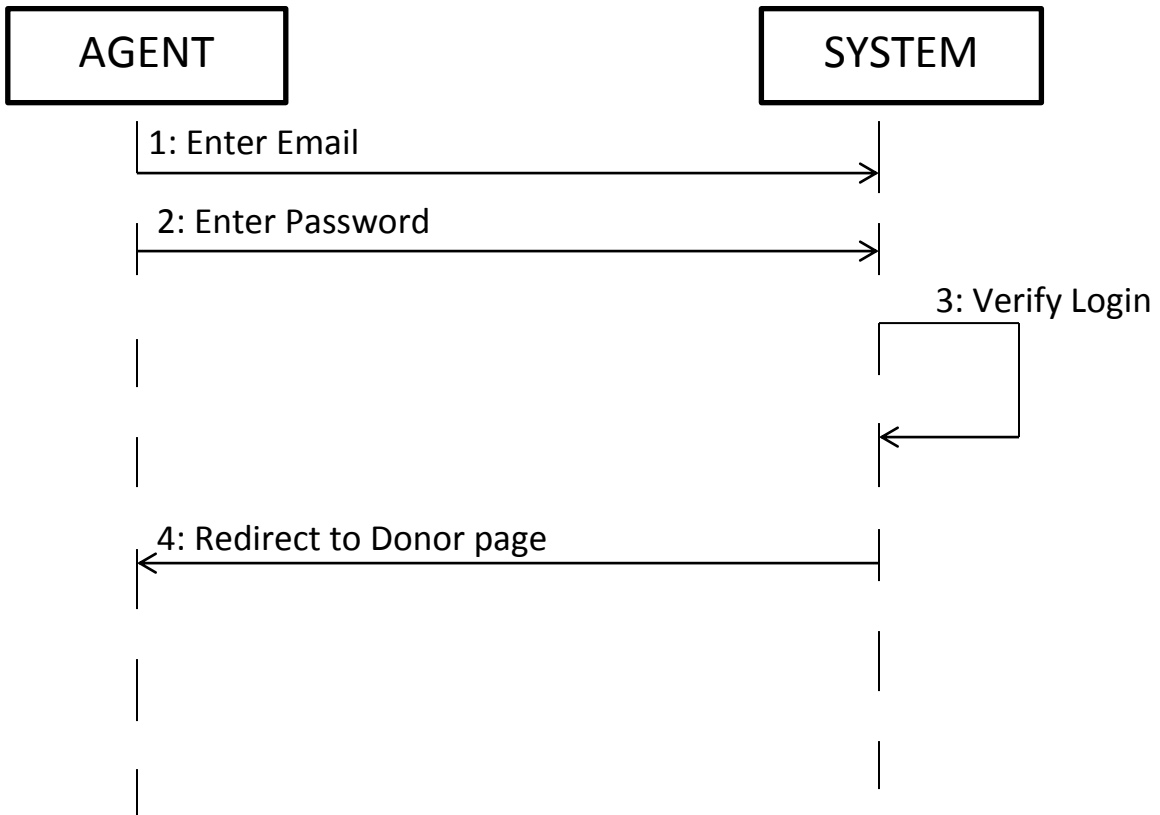
Activity Diagram for AGENT Login

3.4 Sequence Diagram

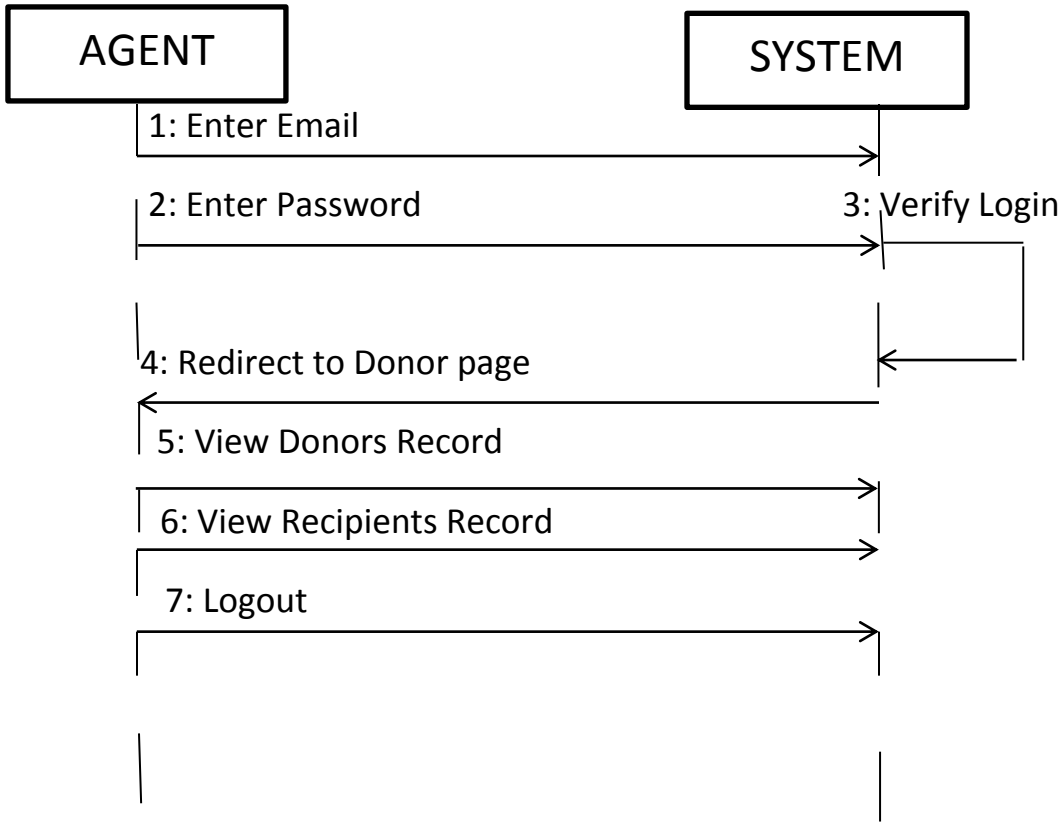
A sequence diagram simply depicts interactions between objects a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function



Sequence Diagram for Admin Login



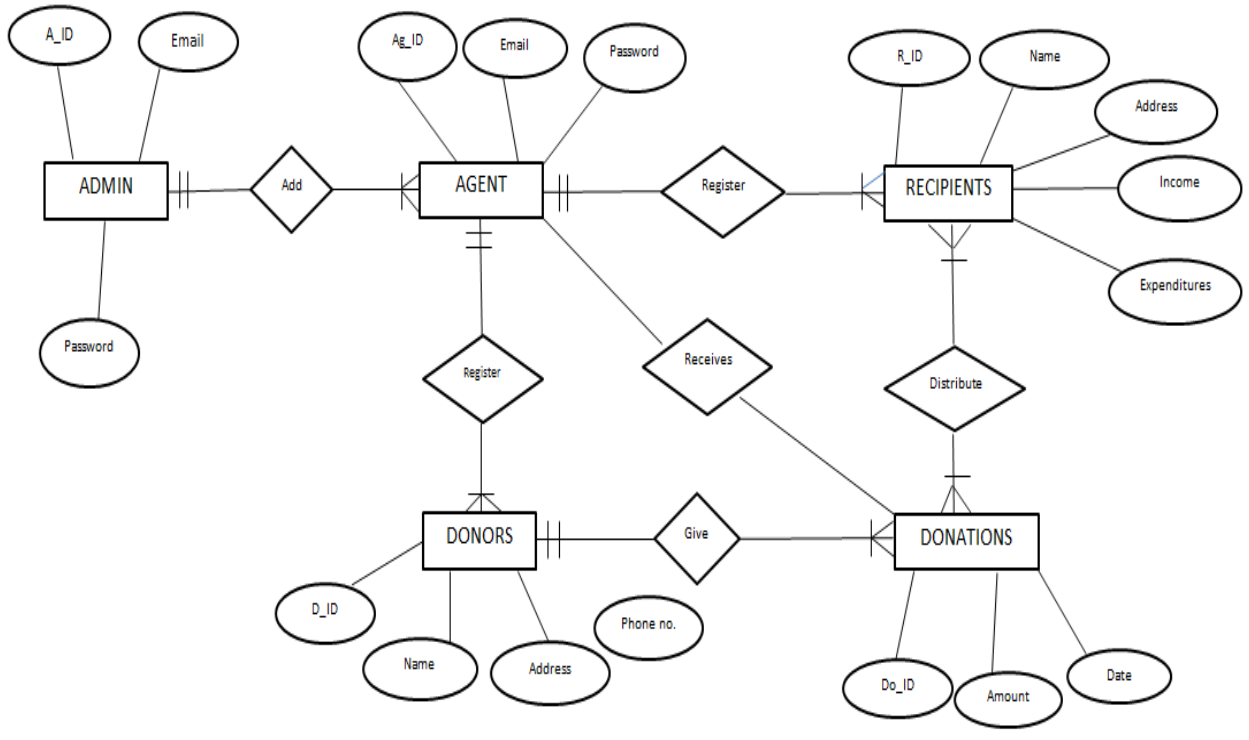
Sequence Diagram for Agent Login



Sequence Diagram for Entire Process

3.5 Entity Relationship Diagram (E.R.D)

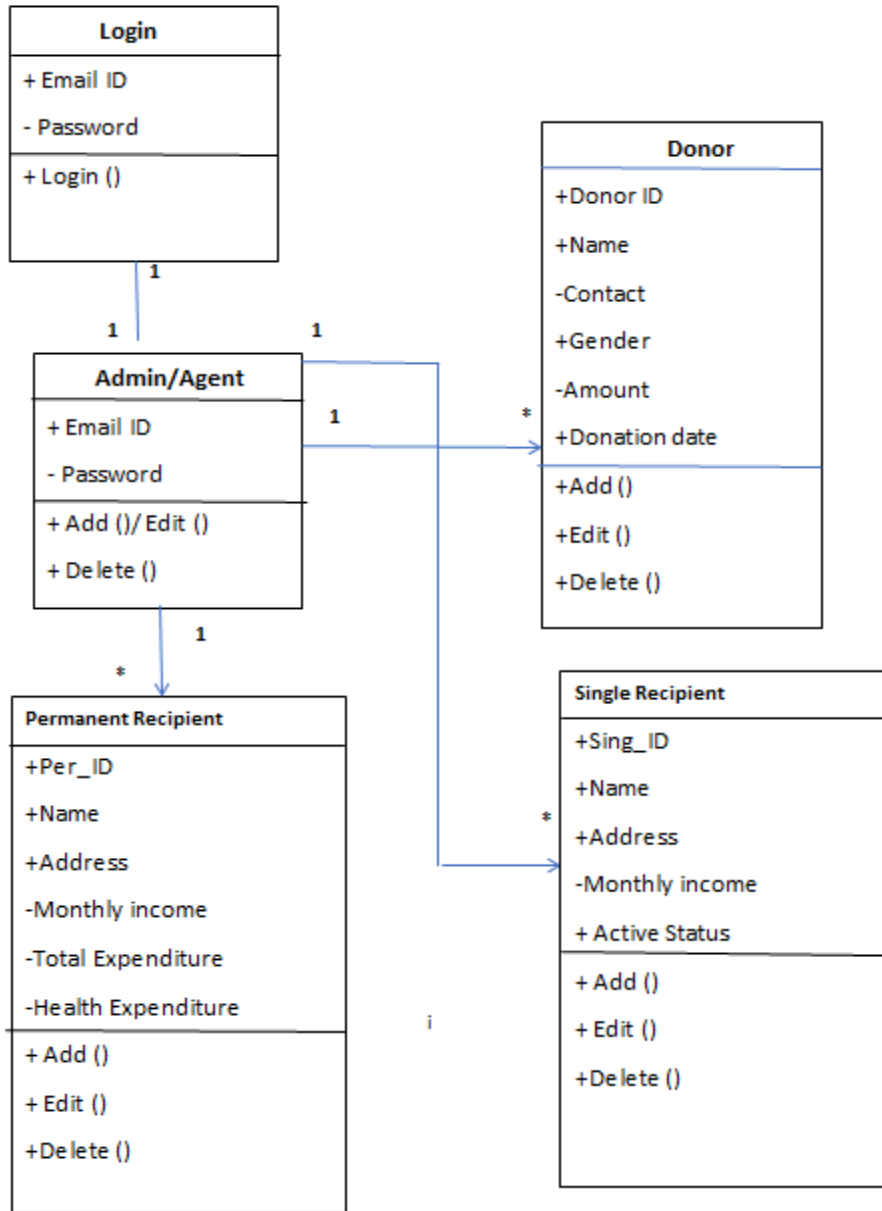
An ER diagram displays the affiliation amid entity sets. An entity set is a collection of alike entities and these entities can have attributes. In term of DBMS, an entity is a table or attribute of a table in database, so by displaying relationship between tables and their attributes, ER diagram displays comprehensive logical structure of a database.



ERD for Entire Process

3.6 CLASS DIAGRAM:

The **class diagram** is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code, can also be used for data modeling.



Class Diagram for OCMS

CHAPTER 4: TOOLS AND TECHNOLOGY

4.1 Introduction

Subsequently completing the design stage, we step forward to implementation phase to transform our design into an executable and working product. Implementation is basically grasping the technical requirements of the system and then deploying them using various tools and technologies available at large extent in the world of internet. As we know that this system is web based, so it is needed to decide which tools and technologies we have to opt for system development. In this chapter we will discuss about all possible options necessary for development and deployment of the system and which programming languages and tools we have used throughout the implementation phase.

4.2 Tool and Language Selection

It may not be a tough task for an experienced developer to choose from a never-ending list of tools and technologies, but for a beginner having no experience in the field; surely, it becomes a grim task.

Following are the tool and languages, which are being used for development of the system.

- asp.net
- Visual Studio
- Bootstrap
- HTML
- CSS
- SQL

4.2.1 Visual Studio

Visual Studio is an **Integrated Development Environment (IDE)** developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE

as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages.

4.2.2 ASP.NET

ASP.NET MVC Model-View-Controller (MVC) is a framework for developing applications which have become so popular these days. Gone are the days when developers would go for ASP.NET to create applications from scratch. MVC is a Model Controller View Pattern; the basic need of introducing MVC was to make complex application development easy. MVC is a lightweight highly testable framework as compared to traditional ASP.NET Web Forms. MVC focuses on Separation of Concerns. The purpose of MVC is to separate the content from the presentation and data processing from content.

4.2.3 Html

HTML was developed by WHATWG in 1993. It is the standard markup language for making Web pages. It stands for Hyper Text Markup Language and styles the structure of a Web page, which consists of a series of HTML elements. These elements tell the browser how to display the content and are represented by different tags. These tags label pieces of content such as "heading", "paragraph", and "table. Browsers use these tags to render the content of the page.

4.2.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the appearance of a document written in a markup language like HTML. CSS is a bedrock technology of the World Wide Web, alongside HTML and JavaScript. World Wide Web in Consortium developed it on December 17, 1996. CSS is designed to enable the parting of appearance and content, including layout, colors and font.

4.2.5 Bootstrap

Bootstrap is a free front-end framework for faster and less difficult web development, which includes HTML, and CSS based design templates for design, forms, buttons, tables, navigation, modals, image containers and many other, as well as optional JavaScript plugins. Bootstrap additionally offers the ability to easily create responsive designs which is; creating those web sites that automatically adjust themselves to look good on all devices, from small phones to large desktop.

4.2.6 SQL

SQL stands for Structured Query Language. SQL is used to converse with database. According to ANSI, it is the standard language for relational database management systems. SQL statements are used to carry out tasks such as update data on a database or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. The standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" are mostly used to achieve nearly all tasks that one may need to do with a database.

4.2.7 Microsoft Visio

Microsoft Visio is a part of Microsoft family and formerly named as Microsoft Office Visio. Microsoft Visio is software designed to translate complex information from text and tables into diagrams. Visio diagrams facilitate communication by breaking down information and display it to be understood at glance.

CHAPTER 5: INTERFACE

5.1 Introduction

User interface design is the design of user interfaces for different software or machines with a clear purpose: to make a better experience for users when navigating through your platform. The main focus is to clarify the usability of various tools and also create a great aspect. Your job is to make things as simple as possible, with clear and well-defined elements, such as buttons, icons, typography, and colors. The design process must be a perfect combination of technical functionality and aesthetic look. Because of that, it is very important not to draw unnecessary attention to elements that can be a distraction from the essential functions of the interface.

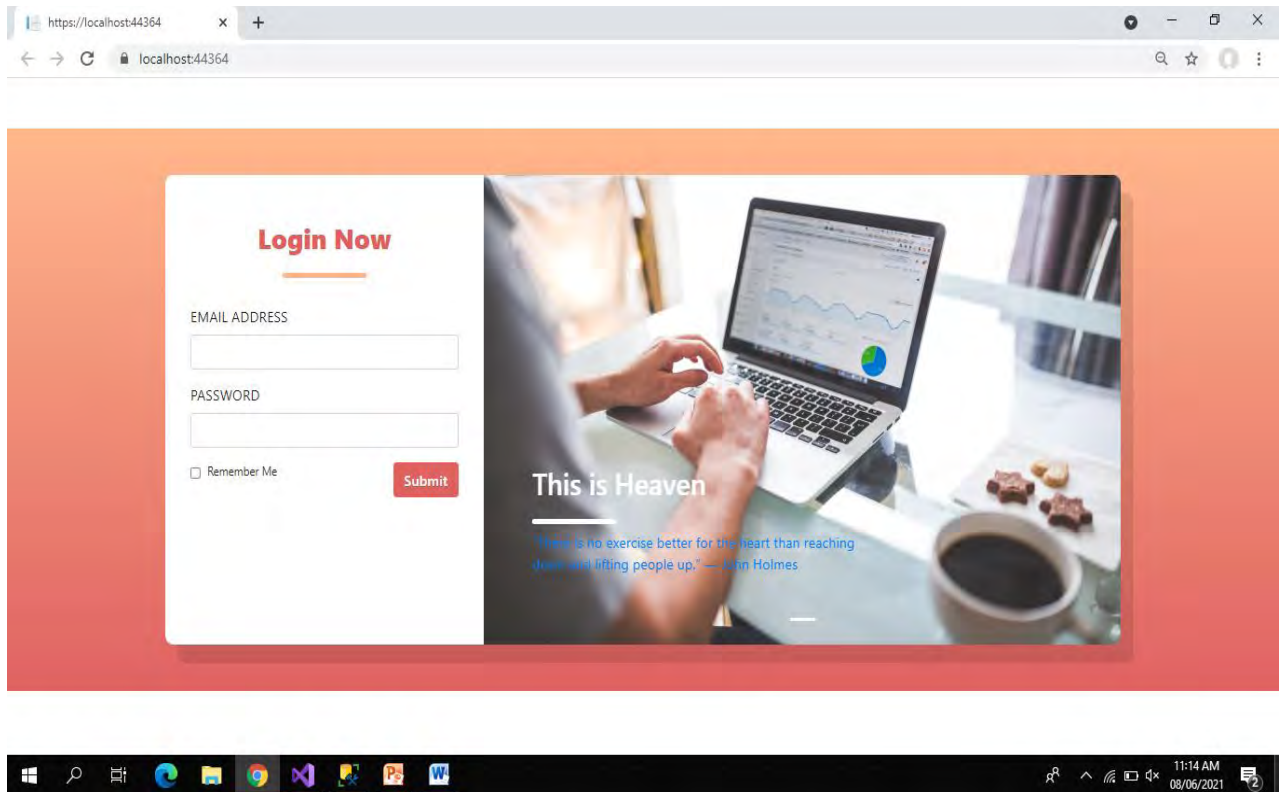


Figure 5.1: Admin Login Page

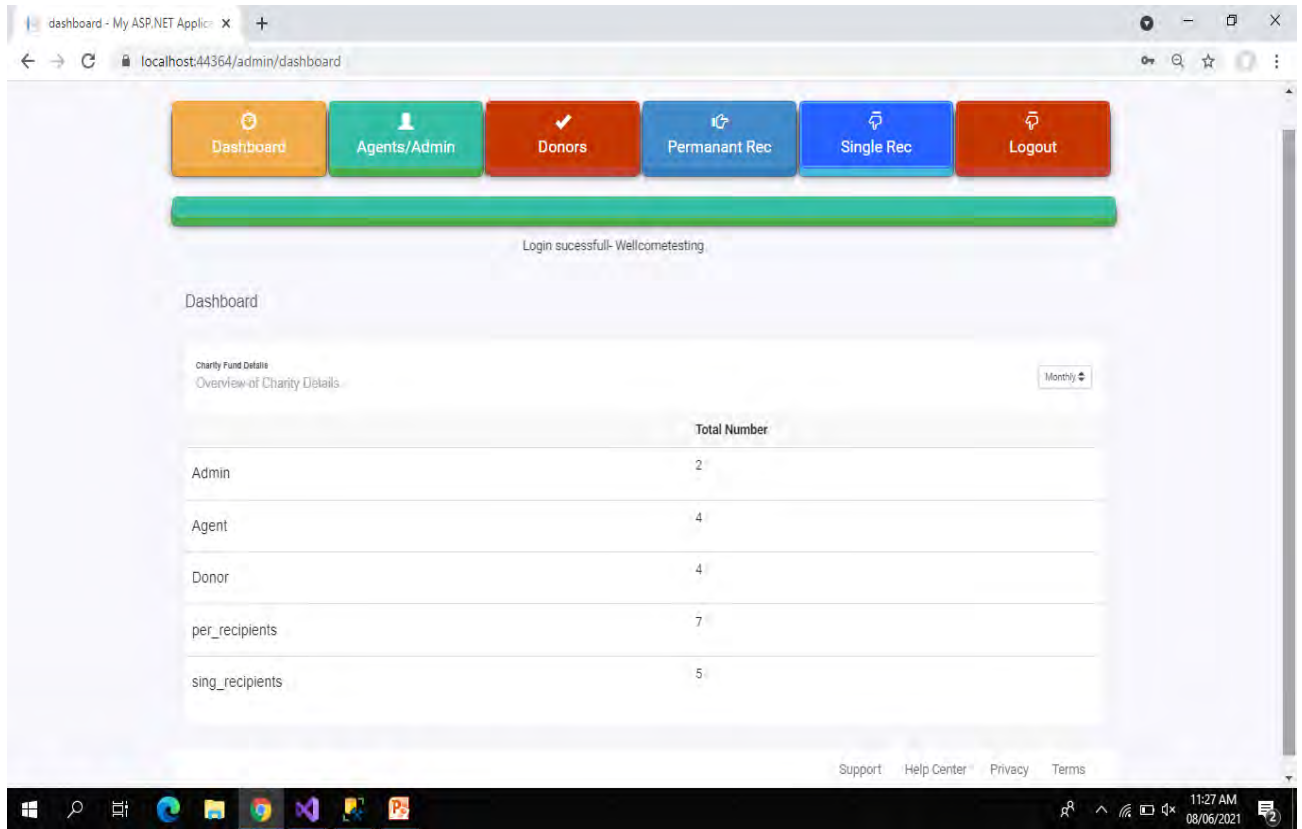


Figure 5.2: Admin Panel Dashboard

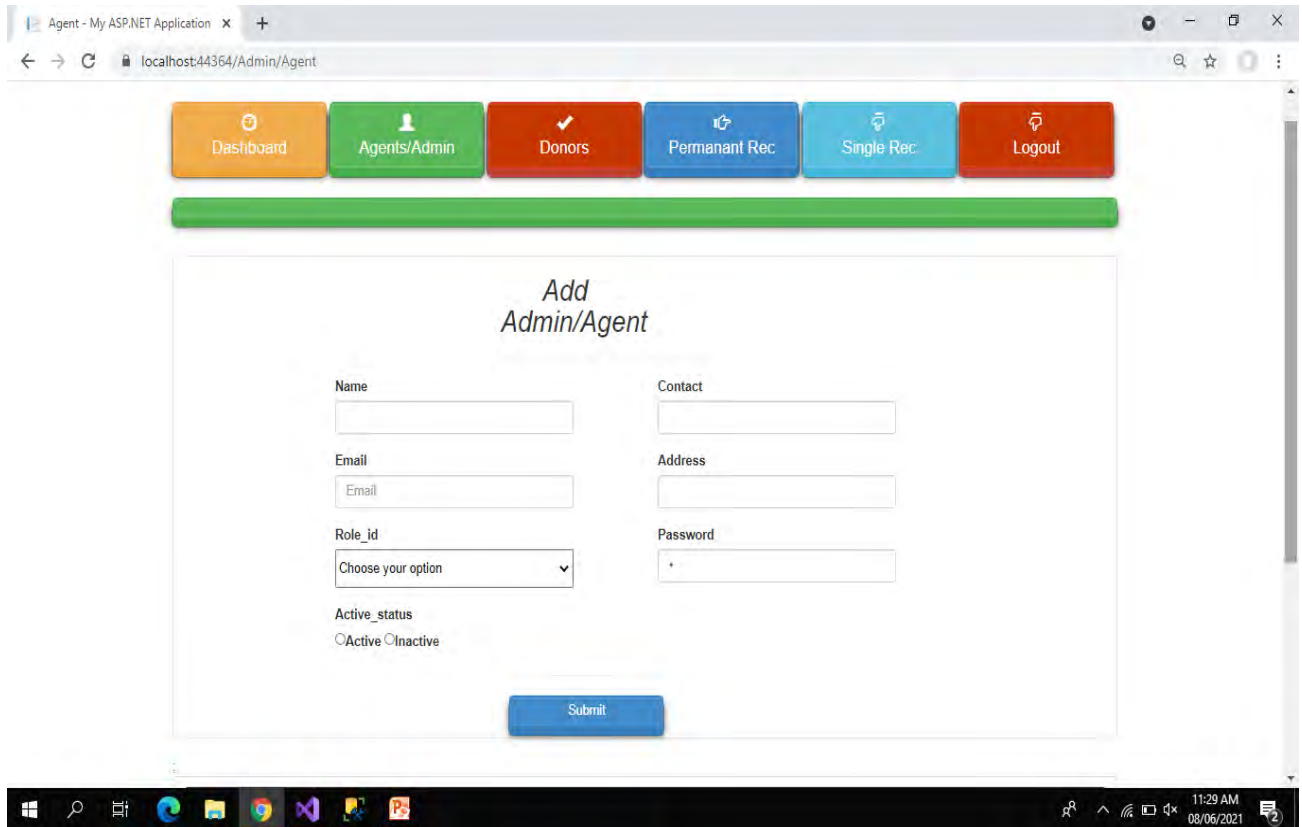


Figure 5.3: Add Admin/Agent Section in Admin Panel











#	Name	Contact	Email	Address	Role_id	Active_Status	Action
1	testing	784963	test@gmail.com	Houseabc	Admin	Inactive	 
2	testing 1234	1234	testingg@testing.com	testing testing	Agent	Active	 
3	Rabia	03002365914	rabia@gmail.com	abc12	Agent	Active	 
4	test	03002563248	abcd@gmail.com	abc13	Agent	Active	 
5	testing201	03335239642	testing201@gmail.com	test22	Admin	Active	 
#	Name	Contact	Email	Address	Role_id	Active_Status	Action

Figure 5.4: Data of Admin/Agent in Admin Panel

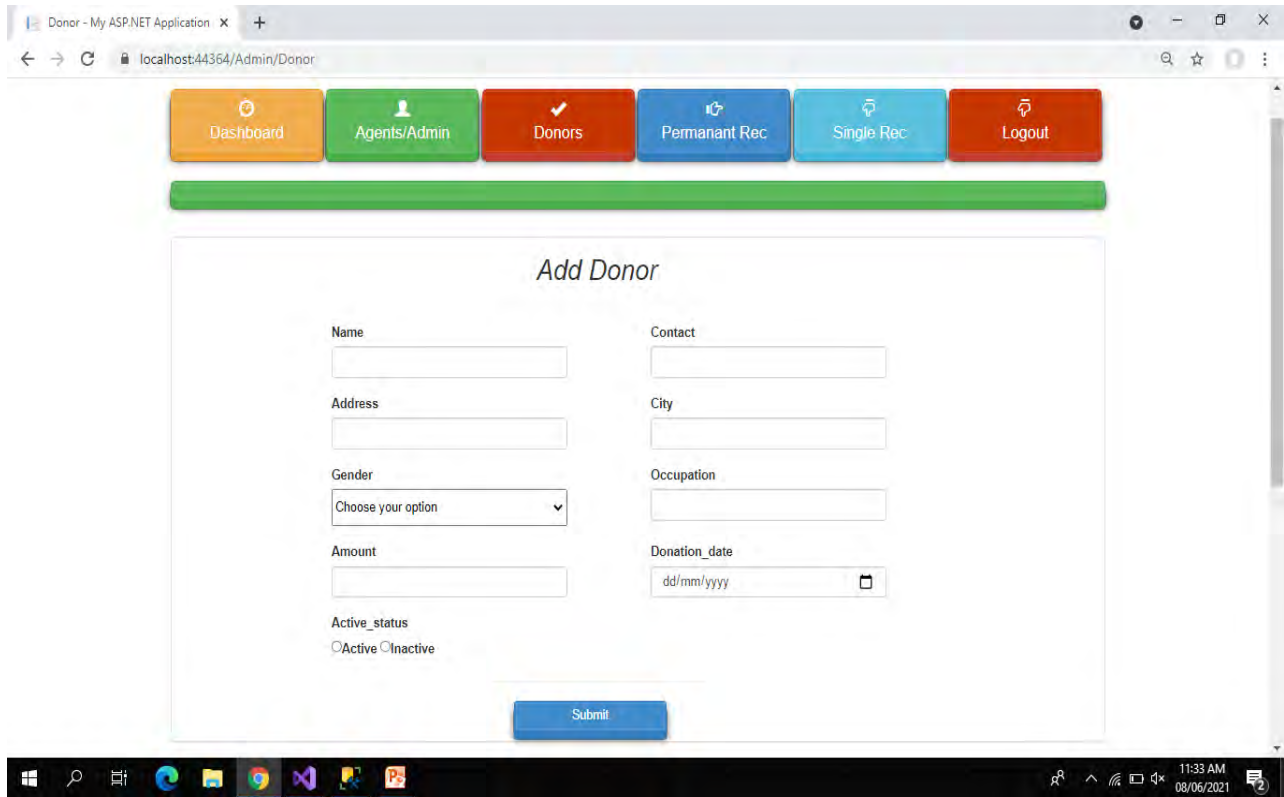


Figure 5.5: Add Donor Section in Admin Panel













#	Name	Contact	Address	City	Gender	Occupation	Amount	Donation_date	Active_Status	Action
1	testing	03335268974	testing testing	Karachi	1	testing 123	60000	2020-12-02	Active	 
2	Aqsa	033658975	testing	Islamabad	2	Teacher	8000	2021-01-04	Active	 
3	Hadia	0345269875	House 1512	Islamabad	2	abc	5000	2021-04-19	Active	 
4	Asia	0311526789	xyz	Islamabad	2	Teacher	5000	2021-06-07	Inactive	 
5	Ashraf	25635489	testing	Lahore	1	xyz	2000	2021-06-03	Inactive	 
6	Sana	03335237605	Hou	Islamabad	2	Teacher	5000	2021-06-18	Active	 

Figure 5.6: Display Data of Donor in Admin Panel

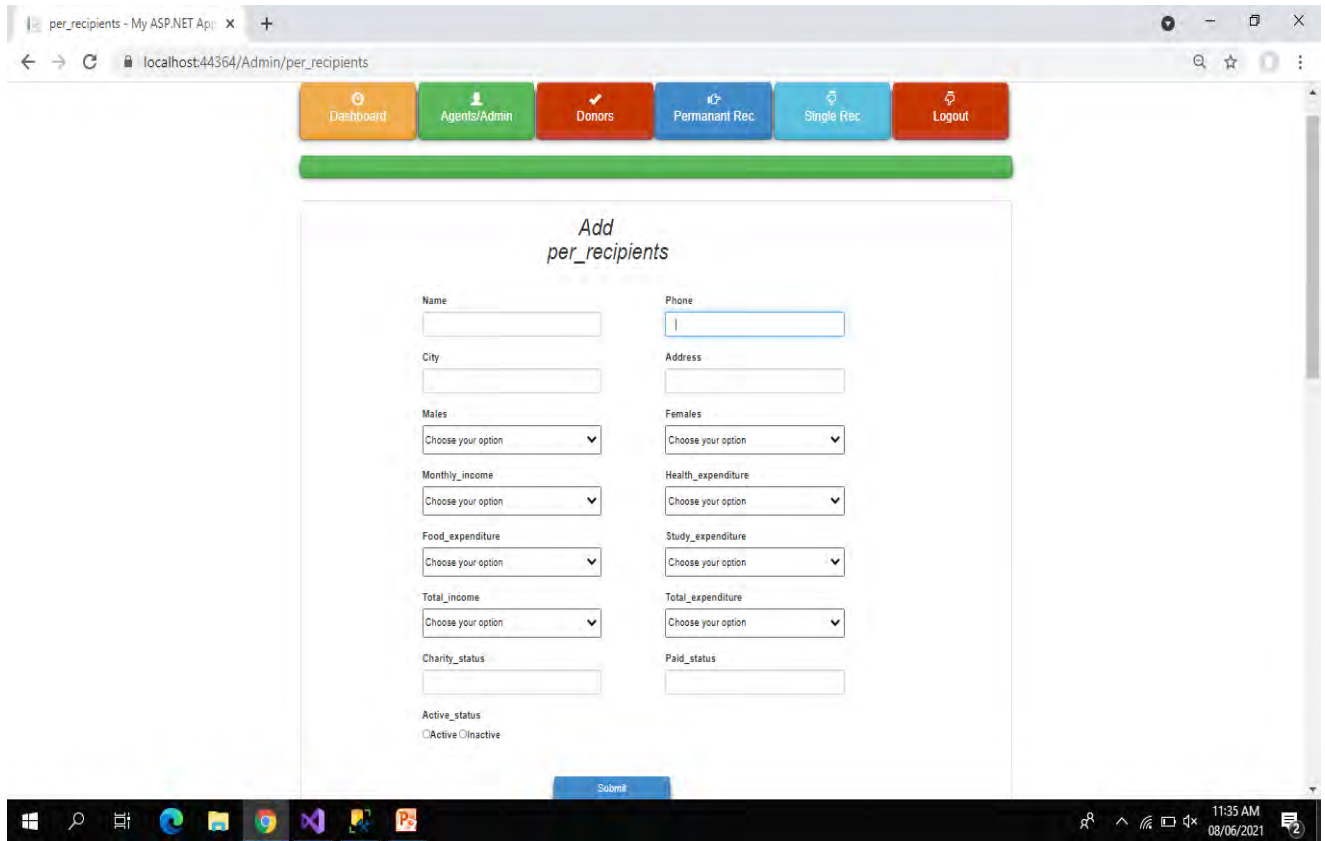


Figure 5.7: Add Permanent Recipient in Admin Panel



















#	Name	Phone	Address	Monthly income	Total income	Total expenditure	Percentage saving amount	Active_Status	Charity_Status	Paid_Status	Action
1	Anum	424849	opoi	25000-30000	32500	27500	118	Active	xyz	ghghg	 
2	Donia	789456	abcd	25000-30000	42500	32500	130	Active	xyz	ghghg	 
3	Ebaad	0336589	house 45	20000-25000	37500	32500	115	Active	abcd	afdfs	 
4	lfra	036589666	House 529	20000-25000	27500	27500	100	Active	abcd	afdfs	 
5	Ahsan	01218596	abcd	15000-20000	22500	22500	100	InActive	abcd	afdfs	 
6	Asim	325698742	abcde	25000-30000	37500	32500	115	InActive	xyz	asd	 
7	test	033356897	abcd	20000-25000	27500	27500	100	InActive	xyz	abc	 
8	Sabeen	0333569875	abc4555	20000-25000	27500	22500	122	InActive	charity	not yet	 
9	Shakeel	0324589756	testing	15000-20000	22500	22500	100	Active	2	2	 
10	testing	3478923	dasjnk	15000-20000	12500	12500	100	Active	active	active	 
11	Laiba	0321569786	testing456	15000-20000	27500	27500	100	Active	active	active	 
12	Noor	0235685666	test	25000-30000	32500	37500	86	InActive	active	active	 

Figure 5.8: Display Data of Permanent Recipient in Admin Panel

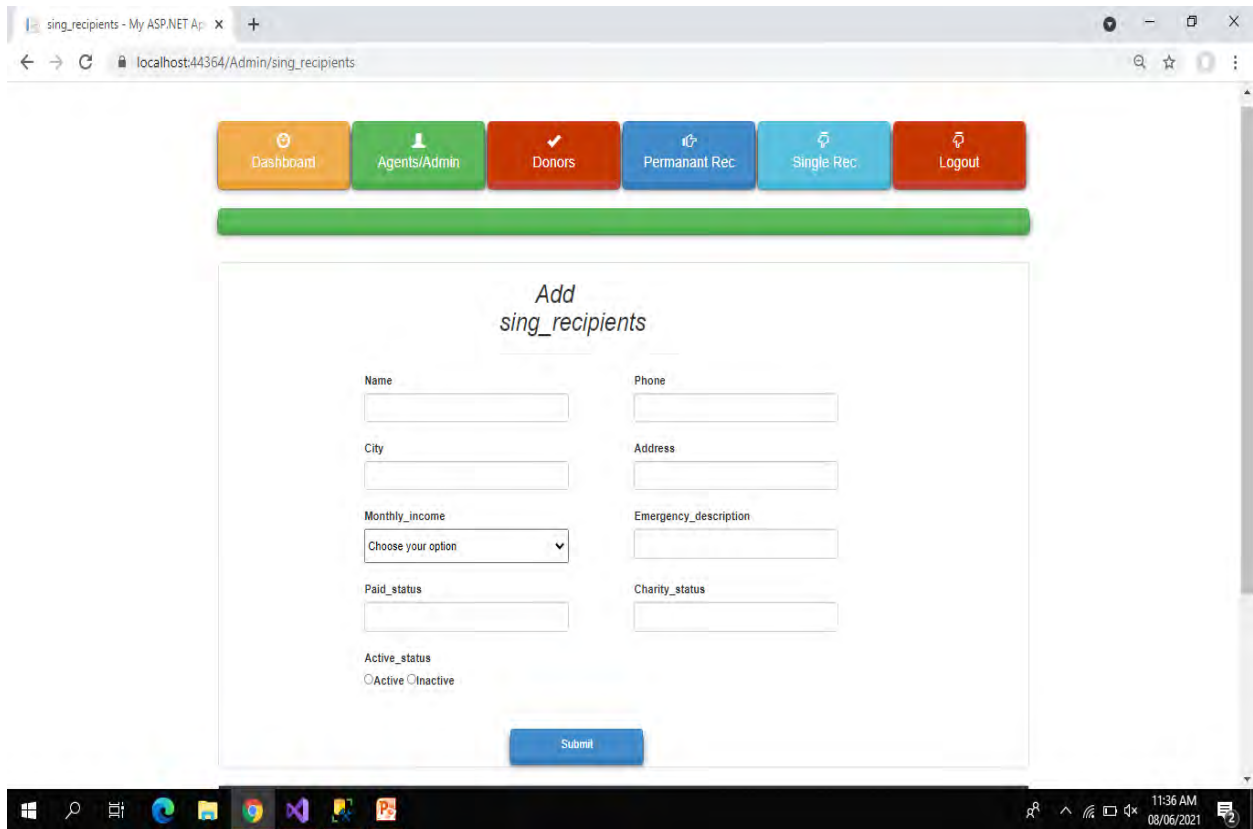


Figure 5.9: Add Single Recipient in Admin Panel











#	Name	Phone	Address	Monthly income	Emergency_description	Active_Status	Charity_Status	Paid_Status	Action
1	Amna Abrar	325978	abcd	30000-35000	abc	Active	0	not yet	 
2	Sana	4569875236	abcv896	25000-30000	xyz	Active	0	not yet	 
3	Maria khan	0235689726	testing 123		abc	Active	0	0	 
4	Fizza	0321567566	House4562	25000-30000	abcd	InActive	0	0	 
5	Amara	0335072037	House456	25000-30000	operation	Active	0	not yet	 
#	Name	Phone	Address	Monthly income	Emergency_description	Active_Status	Charity_Status	Paid_Status	Action

Figure 5.10: Display Data of Single Recipient in Admin Panel

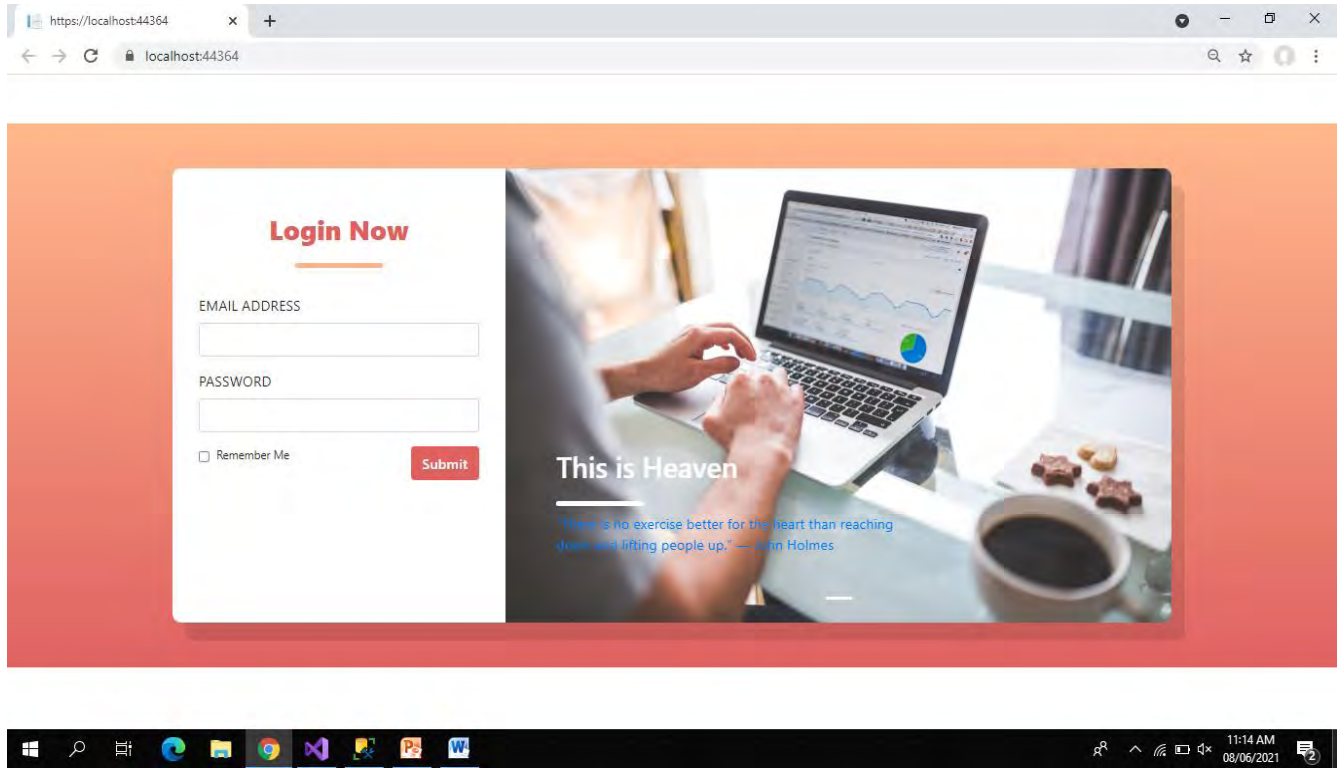


Figure 5.11: Agent Login Page

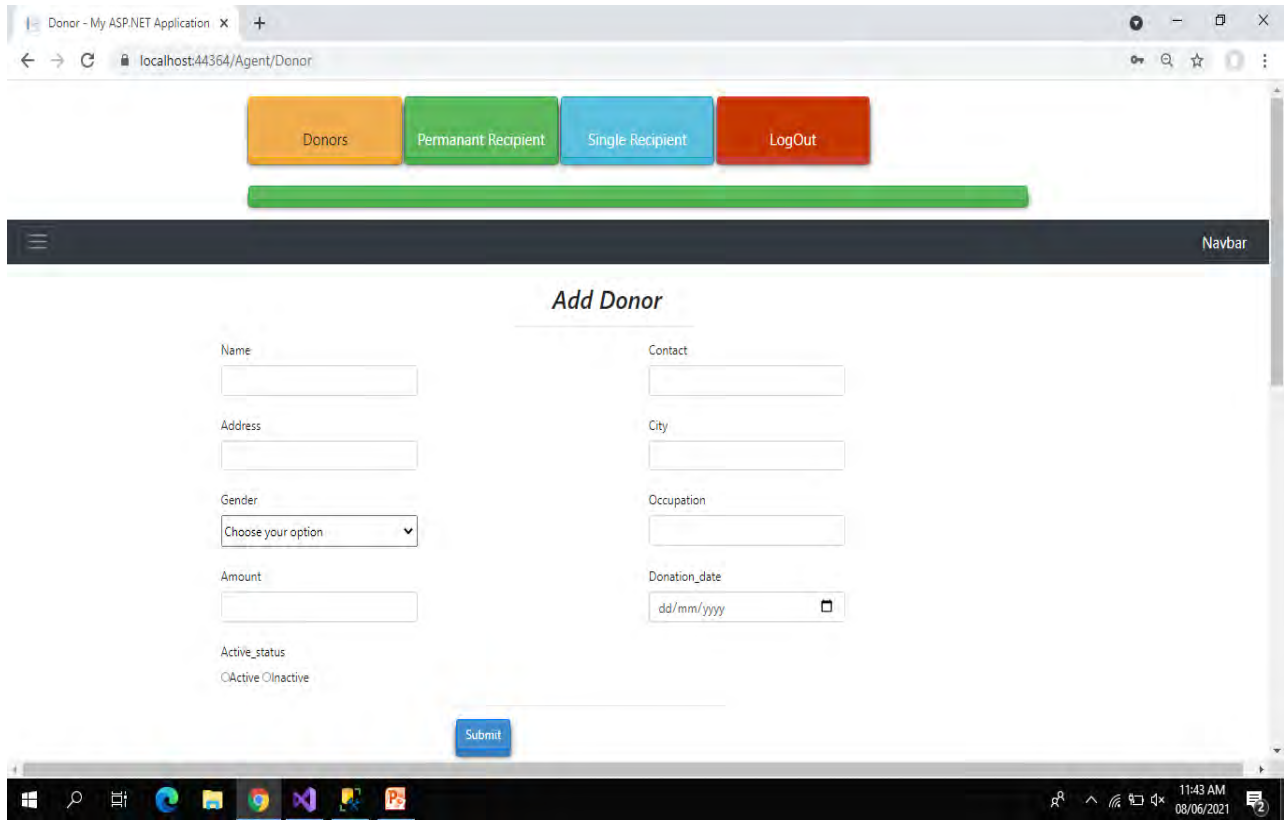


Figure 5.12: Add Donor in Agent Panel

#	Name	Contact	Address	City	Gender	Occupation	Amount	Donation_date	Active_Status	Action
1	testing	03335268974	testing testing	Karachi	1	testing 123	60000	2020-12-02	Active	Edit Delete
2	Aqsa	033658975	testing	Islamabad	2	Teacher	8000	2021-01-04	Active	Edit Delete
3	Hadia	0345269875	House 1512	Islamabad	2	abc	5000	2021-04-19	Active	Edit Delete
4	Asia	0311526789	xyz	Islamabad	2	Teacher	5000	2021-06-07	Inactive	Edit Delete
5	Ashraf	25635489	testing	Lahore	1	xyz	2000	2021-06-03	Inactive	Edit Delete
6	Sana	03335237605	Hou	Islamabad	2	Teacher	5000	2021-06-18	Active	Edit Delete

Figure 5.13: Displaying Data of Donor in Agent Panel

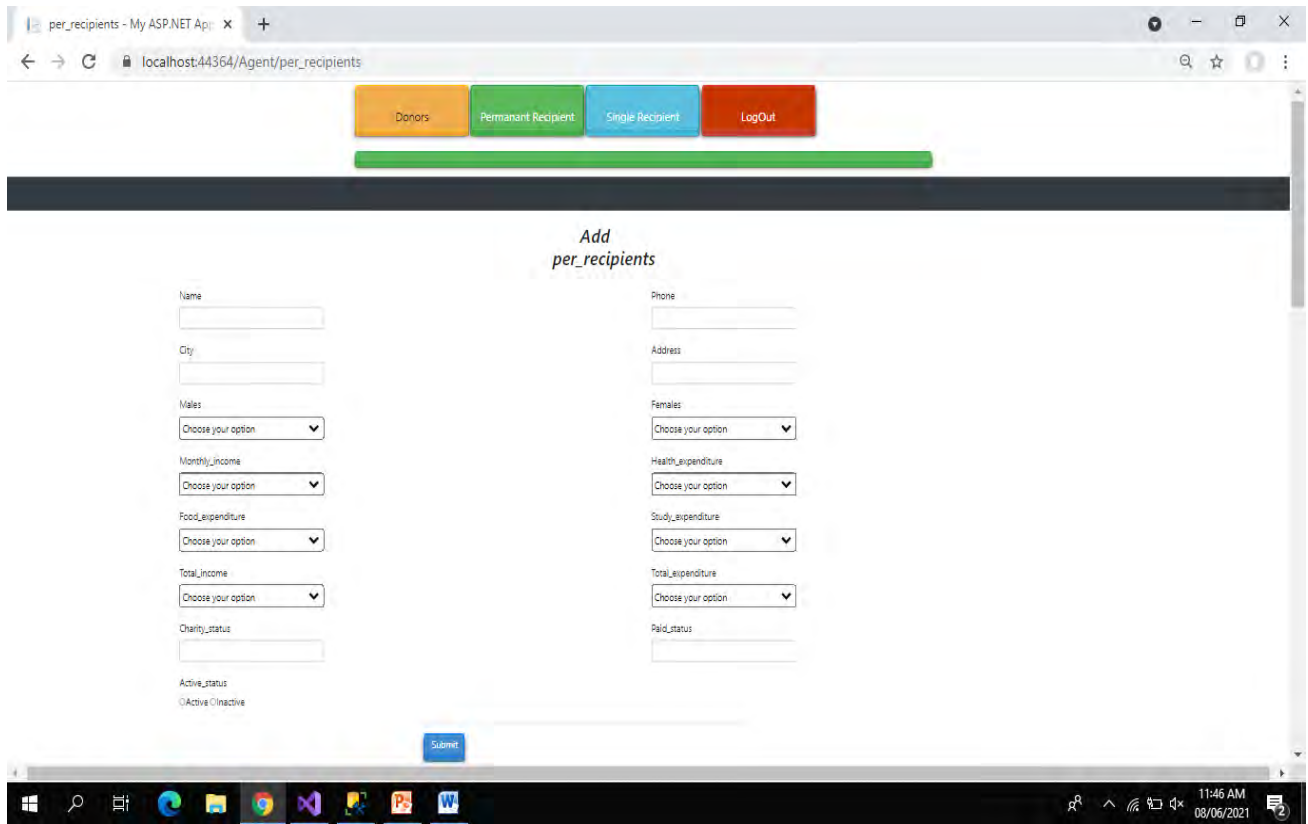


Figure 5.14: Add Permanent Recipient in Agent Panel

#	Name	Phone	Address	Monthly income	Total income	Total expenditure	Active_Status	Charity_Status	Paid_Status	Action
1	Anum	424849	opoi	25000-30000	32500	27500	Active	xyz	ghghg	Edit Delete
2	lfra	036589666	House 529	20000-25000	27500	27500	Active	abcd	afdfs	Edit Delete
3	Ahsan	01218596	abcd	15000-20000	22500	22500	InActive	abcd	afdfs	Edit Delete
4	test	033356897	abcd	20000-25000	27500	27500	InActive	xyz	abc	Edit Delete
5	Shakeel	0324589756	testing	15000-20000	22500	22500	Active	2	2	Edit Delete
6	testing	3478923	dasjkh	15000-20000	12500	12500	Active	active	active	Edit Delete
7	Noor	0235685666	test	25000-30000	32500	37500	InActive	active	active	Edit Delete

Figure 5.15: Displaying Data of Permanent Recipients in Agent Panel

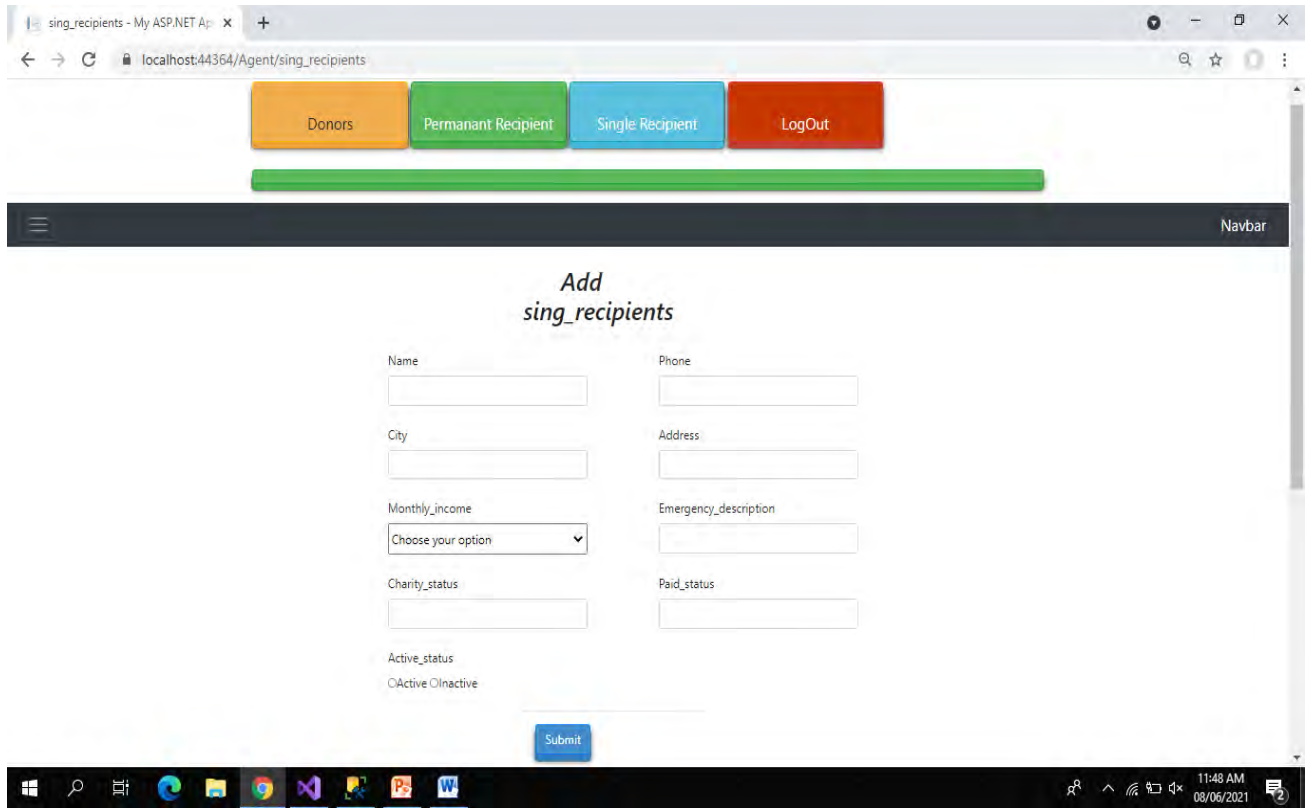


Figure 5.16: Add Single Recipient in Agent Panel

#	Name	Phone	Address	Monthly income	Emergency description	Active_Status	Charity_Status	Paid_Status	Action
1	Amna Abrar	325978	abcd	30000-35000	abc	Active	0	not yet	Edit Delete
2	Sana	4569875236	abcv896	25000-30000	xyz	Active	0	not yet	Edit Delete
3	Maria khan	0235689726	testing 123		abc	Active	0	0	Edit Delete
4	Fizza	0321567566	House4562	25000-30000	abcd	InActive	0	0	Edit Delete
5	Amara	0335072037	House456	25000-30000	operation	Active	0	not yet	Edit Delete

Figure 5.17: Displaying Data of Single Recipients in Agent Panel

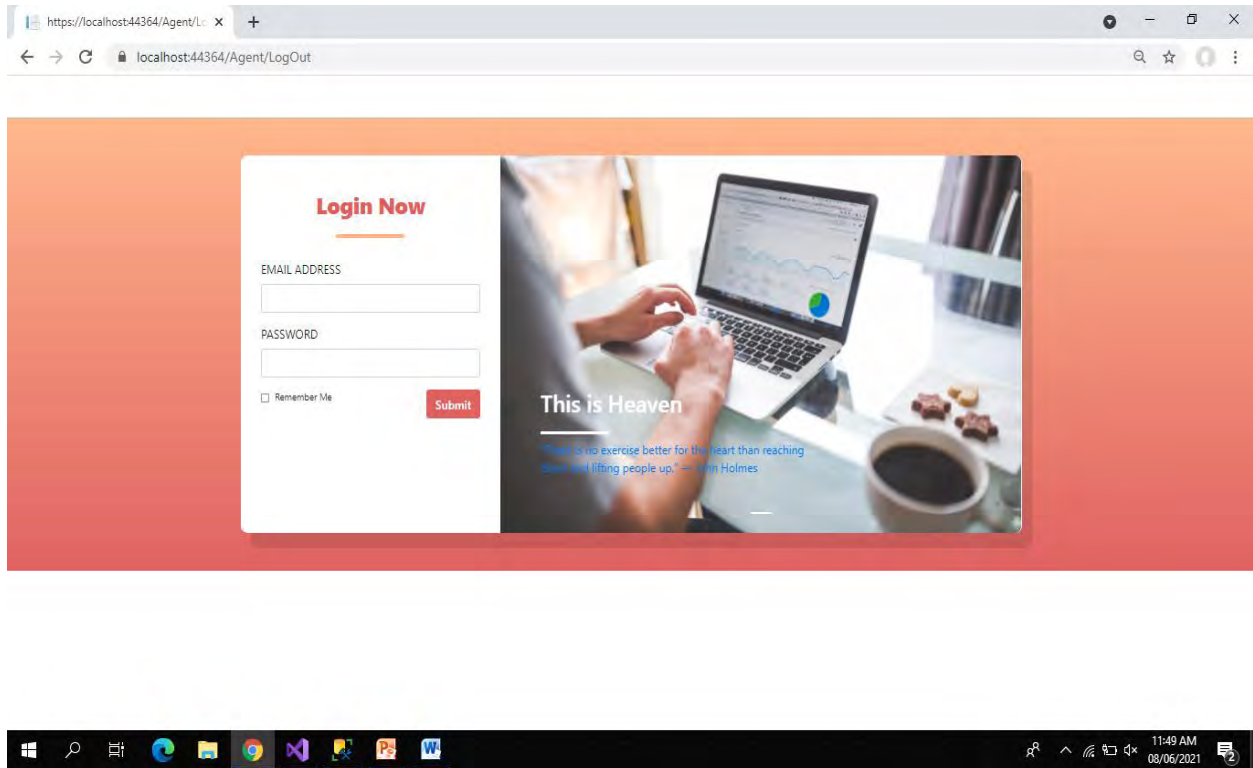


Figure 5.18: Agent Logout

CHAPTER 6: SYSTEM TESTING

6.1 Introduction

Software testing is a procedure to assess the functionality of a software application intending to discover whether the developed software met specified requirements or not and to detect the faults to confirm that the product is fault free in order to produce the quality product.

6.2 Software Testing Types

6.2.1 Manual Testing

Manual testing is the procedure of testing the software by hand to find the faults. Tester should have knowledge of the viewpoint of end-users to confirm that all features are working as stated in the software requirement document. In this course of testing, tester executes the test cases and generates the reports by hand deprived of using any automatic tools.

6.2.2 Automation Testing

Automation testing is the procedure of testing software using an automatic tool to discover faults. In this procedure, testers execute the test scripts and generate the test outcomes automatically by using various tools. Some of the well-known automation testing tools for functional testing are QTP/UFT and Selenium.

6.3 Testing Approaches

- White Box Testing
- Black Box Testing

6.3.1 White Box Testing

It is also termed as Glass Box, Clear Box, and Structural Testing. This testing is grounded on applications' internal code structure. In white-box testing, core viewpoints of the system as well as programming skills are used to design test cases. This testing is generally done at the unit level.

6.3.2 Black Box Testing

It is also termed as Behavioral, Specification-Based or Input-Output Testing. In this testing process, testers assess the functionality of the software under tests deprived of watching at the internal code structure of the software.

6.4 Test Cases

Some important functionality can be tested by the following test cases:

Table 6.1: Test Case of Admin Login

Description	In this test case, Log in functionality of admin is tested. It shows that Log in, as admin is only possible if ID and password are correct.
Setup	Register as admin with following credentials ID: test@gmail.com Password: test
Instructions	Enter ID: test@gmail.com Password: test Click on Submit button.
Results	Admin ID as "test@gmail.com" with password "test" log in Successfully. Admin ID as "test12@gmail.com" with password "test" log in failed

Table 6.2: Test Case of Dashboard of Admin Panel

Description	In this test case "Dashboard in Admin Panel" functionality has been checked.
Setup	Log in as Admin by entering login ID and password
Instructions	Enter admin ID: test@gmail.com Password: **** Enter Log in Click on "Dashboard"
Results	Total number of Agents, Permanent and Single recipients will appear.

Table 6.3: Test Case of Add Agent in Admin Panel

Description	In this test “Agents” functionality are checked.
Setup	<ul style="list-style-type: none">• Log in as admin by enter ID and password• Click on “Agent” button.• Add agent details.
Instructions	<ul style="list-style-type: none">• Enter admin ID: test@gmail.com• User Password: ****• Enter Log in
Results	A page will appear Add Agent(To enroll the agent it will ask You for details like Name, Contact, Email, Password, Role id and Active status.)

Table 6.4: Test Case of Add Permanent Recipient in Admin Panel

Description	In this test “Permanent Recipient” functionality are checked.
Setup	<ul style="list-style-type: none">• Log in as admin by enter ID and password• Click on “Permanent Recipient” button.• Add permanent Recipient details.
Instructions	<ul style="list-style-type: none">• Enter admin ID: test@gmail.com• Password: ****• Enter Log in
Results	A page will appear Add per recipient (To enroll the permanent recipient it will ask You for details like Name, Phone, Address, Monthly income, Charity status and Active status.)

Table 6.5: Test Case of Add Single Recipient in Admin Panel

Description	In this test "Single Recipient" functionality are checked.
Setup	<ul style="list-style-type: none"> • Log in as admin by enter ID and password • Click on "Single Recipient" button. • Add Single Recipient details.
Instructions	<ul style="list-style-type: none"> • Enter admin ID: test@gmail.com • Password: **** • Enter Log in
Results	A page will appear Add sing recipient (To enroll the single recipient it will ask You for details like Name, Phone, Address, Monthly income, Charity status and Active status.)

Table 6.6: Test Case of Add Donor in Admin Panel

Description	In this test "Donor" functionality are checked.
Setup	<ul style="list-style-type: none"> • Log in as admin by enter ID and password • Click on "Donor" button. • Add Donor details.
Instructions	<ul style="list-style-type: none"> • Enter admin ID: test@gmail.com • Password: **** • Enter Log in
Results	A page will appear Add donor (To enroll the donor it will ask You for details like Name, Phone, Address, Occupation, Amount and Active status.)

Table 6.7: Test Case of Agent Login

Description	In this test "Agent login" functionality is checked.
Setup	Log in as Agent by enter ID and password.
Instructions	<p>Enter agent ID : abcd@gmail.com Password: abc225 Click on the Submit button.</p>
Results	<p>Agent ID as " abcd@gmail.com" with password " abc225" log in Successfully. Agent ID as " abcd@gmail.com" with password "abc2" log in failed</p>

Table 6.8: Test Case of Add Permanent Recipient in Agent Panel

Description	In this test "Permanent Recipient" functionality are checked.
Setup	<ul style="list-style-type: none">• Log in as agent by enter ID and password• Click on "Permanent Recipient" button.• Add permanent Recipient details.
Instructions	<ul style="list-style-type: none">• Enter agent ID : abcd@gmail.com• Password: *****• Enter Log in
Results	A page will appear Add per recipient (To enroll the permanent recipient it will ask You for details like Name, Phone, Address, Monthly income, Charity status and Active status.)

Table 6.9: Test Case of Add Single Recipient in Agent Panel

Description	In this test "Single Recipient" functionality are checked.
Setup	<ul style="list-style-type: none">• Log in as agent by enter ID and password• Click on "Single Recipient" button.• Add Single Recipient details.
Instructions	<ul style="list-style-type: none">• Enter agent ID : abcd@gmail.com• Password: *****• Enter Log in
Results	A page will appear Add sing recipient (To enroll the single recipient it will ask You for details like Name, Phone, Address, Monthly income, Charity status and Active status.)

Table 6.10: Test Case of Add Donor in Agent Panel

Description	In this test “Donor” functionality are checked.
Setup	<ul style="list-style-type: none">• Log in as agent by enter ID and password• Click on “Donor” button.• Add Donor details.
Instructions	<ul style="list-style-type: none">• Enter agent ID : abcd@gmail.com• Password: *****• Enter Log in
Results	A page will appear Add donor (To enroll the donor it will ask You for details like Name, Phone, Address, Occupation, Amount and Active status.)

CHAPTER 7: CONCLUSION & FUTURE WORK

7.1 FUTURE WORK:

We can improve the system by adding more functionality like the eligibility of charity will be confirmed through text message.

We will make online bank fund system for donors so that they can donate easily. Donors should be given access all around the world so that donation can be increased and more people can be helped through it. In Ramadan, food packets can also be given.

7.2 CONCLUSION:

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of Programming language e.g. asp.net & also provide opportunity to learn develop project. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

References

List of useful websites

[1] Website: www.w3schools.com

[2] Website: www.fontawesome.com

[3] Website: www.mdbootstrap.com

[4] Website: www.wikipedia.com