# CHAPTER 1

# INTRODUCTION

### INTRODUCTION

In a world where innovation drives progress and education shapes futures, ProjectXpert emerges as a transformative force in academic project management. At the heart of our vision lies a belief in the power of structured guidance, transparent evaluation, and collaborative growth—a belief that fuels our commitment to bridge the gap between students, guides, and institutions. Our mission is to streamline the journey from project ideation to completion, empowering academic excellence and fostering a culture of innovation and accountability.

In today’s fast-paced educational environment, we recognize that managing multiple project submissions, evaluations, and feedback loops can be complex and overwhelming. Yet, the desire to nurture talent, encourage creativity, and maintain fairness remains constant within academic communities. This is where ProjectXpert steps in, serving as a dynamic platform designed to simplify, organize, and elevate the project management process.

For students aspiring to bring their ideas to life, ProjectXpert provides an intuitive and supportive space to submit project abstracts, receive timely feedback, and track their progress seamlessly. Our user-friendly interface ensures that the process of turning innovative ideas into impactful projects is smooth and empowering. Project guides, who play a pivotal role in shaping young minds, find in ProjectXpert a reliable companion that aids in fair evaluation, systematic ranking based on feasibility, innovation, and clarity, and efficient feedback management—all designed to support and strengthen mentorship

Institution heads and administrators, entrusted with maintaining quality and transparency, benefit from an insightful dashboard that offers real-time oversight, comprehensive reporting, and efficient management of academic project activities. With automated email notifications and integrated feedback tracking, ProjectXpert ensures that no communication is missed, no progress goes unnoticed, and every student receives the guidance they deserve.

As you embark on this journey with ProjectXpert, know that every project you nurture, every idea you support, and every piece of feedback you offer carries the potential to inspire innovation and build a brighter academic future. Welcome to ProjectXpert—where every project is a step toward excellence, and every evaluation is a milestone in the journey of learning.

### SCOPE

The scope of this project is to create an advanced web-based platform that can be used by educational institutions to streamline the submission, evaluation, and monitoring of student projects. The platform will allow student groups to submit multiple project abstracts, which will be organized and ranked automatically based on feasibility, innovation, and clarity. The system will have three main types of users: admin, project guide, and student, each with their own specific features and capabilities. Students can submit and track their project progress, guides can review, rank, approve, or reject submissions, provide feedback, and schedule reviews, while admins can oversee all activities and generate institutional reports. Automated notifications and feedback tracking will enhance communication and ensure transparency. The goal of the project is to create a structured, efficient, and supportive environment that fosters innovation and improves project management within educational settings.

### OVERVIEW OF THE PROJECT

This project aims to create a web-based platform that streamlines the submission, evaluation, and management of student projects in educational institutions. The platform will allow student groups to submit multiple project abstracts, which will be automatically ranked based on feasibility, innovation, and clarity. There will be three main types of users: admin, project guide, and student, each with its own set of features and capabilities. The platform will allow users to interact in a way that best suits their roles, whether it be submitting projects, evaluating and giving feedback, or monitoring overall progress. The goal of the project is to simplify project handling, improve communication, and ensure a transparent evaluation process. Overall, this project has the potential to greatly enhance the academic project experience for both students and faculty.

**Module Details**

**Admin**

1. Add and manage project guides (create accounts for guides and set login credentials).
2. View a list of registered project guides.
3. Add and manage student accounts (assign students under specific guides).
4. View a list of registered student groups and their project submissions.
5. Review and approve/reject submitted project abstracts.
6. View the ranking list of submitted project abstracts based on feasibility, innovation, and clarity.
7. Track the status of approved and rejected projects.
8. Monitor feedback provided by guides to student groups.
9. Schedule and manage project review sessions.
10. Generate and view reports of project activities and student progress.
11. Change their own password.
12. View a list of all project-related notifications and contact inquiries.

**Student Module**

1. Register and create student accounts (set login credentials).
2. Submit multiple project abstracts for evaluation under assigned project guides.
3. View the status of submitted projects (approved, rejected, or under review).
4. Receive feedback from project guides on project abstracts.
5. Track progress based on feedback and resubmit updates.
6. View a list of all ongoing review sessions and their schedules.
7. Receive automated email notifications about project approval, feedback, and deadlines.
8. View and attend scheduled project review sessions with the guide.
9. Access and update project details, including title, abstract, and description.
10. Change their own password for secure access to the platform

**Project Guide Module**

1. Register and create guide accounts (set login credentials).
2. View a list of assigned student groups and their submitted project abstracts.
3. Review submitted project abstracts and rank them based on feasibility, innovation, and clarity.
4. Provide feedback on project abstracts and suggest improvements.
5. Approve or reject project abstracts based on evaluation criteria.
6. Schedule and manage project review sessions with students.
7. Track student progress and monitor if feedback has been addressed.
8. Receive and view automated email notifications about student submissions and updates.
9. View the status of projects (approved, rejected, or in progress).
10. Track and manage multiple projects from different student groups.

# CHAPTER 2

**SYSTEM ANALYSIS**

System analysis is a logical process; it is an important activity that takes place when a new system is being built. The objective of this phase is not actually to solve the problem but to determine what must be done to solve the problem. It is the central intact of system development and it includes gathering and interpreting facts, diagnosing and using this information to recommend improvements to the system. System analysis gives the structure and functioning of the system and it specifies what the system should do. It is helpful to understand the problem and emphasize what is needed from the system.

Before designing an application which will help its users, it is important that the way it currently operates should be clearly identified. The process of system investigation includes several methods of gathering the required information. It is important that the approach is appropriate to the application under consideration and the user of the system. System analysis should be creative and imaginative in producing new solutions to meet the user requirements. In short the system is analyzed by gathering various information such as system functionality, problems encountered, requirements by the proposed system, users, their tasks and responsibilities. Here in the system analysis phase, the system is viewed as a whole and inputs to the system are defined, and also the output from the system is traced through the various data collected.

A detailed study of these processes must be made by the various techniques like interviews, questionnaires, etc. The data collected by these sources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subject to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that weighed with the existing faces. The solutions given on a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for any endorsement by the user. The proposal is reviewed on user request and suitable changes that are made. This is a loop that ends as soon as the user is satisfied with the proposal.

### EXISTING SYSTEM

* Project submission is manual and time-consuming.
* Students submit project abstracts via email or printed documents.
* Guides face difficulty in tracking and managing submissions.
* There is no standardized ranking system for evaluation.
* Delays occur in giving feedback and making decisions.
* Students struggle to get timely responses from guides.
* There is no structured system to monitor project progress or previous feedback
  + 1. **Limitations of Existing System**
* Existing project submission methods like email and printed documents are manual and time-consuming, making the submission process inefficient and prone to delays.
* Guides face difficulty in tracking and managing multiple student submissions, leading to confusion and loss of information.
* There is no standardized ranking system for evaluating project abstracts, resulting in I inconsistent and delayed feedback.
* Students often struggle to receive timely responses from their guides, affecting their project progress.
* There is no structured system to monitor project progress or review previous feedback, which leads to miscommunication and lack of clarity.

### PROPOSED SYSTEM

The proposed system is designed to provide a comprehensive and efficient platform for managing the submission, evaluation, and tracking of student projects within educational institutions. It brings together three main user types: Admin, ProjectGuide, and Student, each with distinct roles and responsibilities. The Admin will have full control over the platform, including managing user accounts, approving submissions, and overseeing project evaluations. Project Guides will be responsible for evaluating student projects, providing feedback, and monitoring the overall progress of their students' work. Students will be able to submit their project abstracts, view feedback from their guides, and track their project's status and progress. The system focuses on enhancing transparency, simplifying the evaluation process, and fostering better communication between students and faculty, ensuring an efficient and streamlined project experience for all users.

### ADVANTAGES OF PROPOSED SYSTEM

The proposed ProjectXpertsystem for project submission, evaluation, and management offers several advantages over existing systems in academic institutions:

1. **Centralized Platform**: The ProjectXpert system centralizes all project-related activities, from submission to evaluation, providing a unified space for students, guides, and admins to interact. This reduces the confusion and inefficiencies of handling projects across various platforms or channels.
2. **Transparency in Evaluation**: By automating project rankings based on predefined criteria like feasibility, innovation, and clarity, the system ensures transparency in the evaluation process. Students and guides can easily track the progress and feedback, fostering trust and fairness.
3. **Efficient Project Management**: The system allows project guides to track and manage multiple student projects simultaneously, with features that make feedback and evaluations more streamlined. This reduces delays in communication and ensures that students receive timely responses..
4. **Role-Based Access**: The system provides role-based access for different users—Admin, Project Guide, and Student. This ensures that each user can only access the features relevant to their role, improving security and user experience.
5. **Automated Ranking and Feedback**: The automatic ranking of project abstracts based on feasibility and clarity eliminates the need for manual ranking, saving time and effort. Project guides can provide instant feedback, reducing the time between project submission and response.
6. **Enhanced Communication**: The platform facilitates efficient communication between students and guides, ensuring that students can receive guidance and feedback on time. This system reduces misunderstandings and improves the overall quality of the projects.
7. **User-Friendly Interface**: ProjectXpert is designed with a user-friendly interface, making it accessible even for users with minimal technical knowledge. This encourages more active participation from students and faculty alike.
8. **Security**: The system incorporates strong security measures, ensuring that user data and project information are protected. This includes secure login and password management for all types of users, safeguarding sensitive academic information.
9. **Comprehensive Project Tracking**: Students and guides can easily monitor the progress of projects, view feedback history, and track the overall development, ensuring that projects stay on track and meet the required standards.
10. **Flexibility and Scalability**: The system is adaptable to various types of academic projects, from simple abstracts to complex research proposals. It can handle a wide range of project types, making it suitable for diverse educational institutions.

### FEASIBILITY STUDY

An initial investigation led to a proposal to develop a centralized web-based platform for project submission, evaluation, and management within educational institutions. Once the proposed system was conceptualized and approved, a feasibility study was conducted. The purpose of the feasibility study is to explore various candidate systems and assess their practicality by evaluating technical, economical, operational, and legal aspects. After careful evaluation, ProjectXpert was identified as the best candidate system to streamline project handling, improve communication, and ensure a transparent evaluation process.

* + 1. **Technical Feasibility**

Technical feasibility study deals with the hardware as well as the software requirements. The scope was whether the work for the project is done with the current equipment and the existing software technology. The outcome was found to be positive. The system "ProjectXpert" is said to be technically feasible because the software and hardware required to develop this system are already available. For this system, we use Python as the technology, Django as the framework, and MySQL as the back end.

* + 1. **Operational Feasibility**

The proposed project would be beneficial to the organization, satisfying the objectives when developed and installed. One of the main problems faced during the development of a new system is getting acceptance from the users. There is support from the management of the institution towards the development of the project. All the operational aspects are considered carefully. Thus, the project ProjectXpert is operationally feasible.

* + 1. **Economic Feasibility**

The developing system must be justified by cost and benefit, ensuring that the effort taken on the project provides the best return at the earliest. One of the important factors affecting the development of a new system is the cost involved. Since ProjectXpert is developed as part of a project work, there is no manual cost required for the proposed system. All necessary resources, including software and hardware, are already available, indicating that the system is economically feasible for development. The system ProjectXpert is said to be economically feasible because the required technologies like Python, Django, MySQL are freely available or already accessible. No additional purchases or external tools are necessary. Furthermore, the maintenance cost of the system is very low. As this project is carried out as part of our academic curriculum, the overall development cost is minimized, making ProjectXpert a highly cost-effective and economically feasible system.

**2.3.4 Legal Feasibility**

Legal feasibility ensures that the system complies with laws regarding data privacy, intellectual property, and other regulations. The ProjectXpert system adheres to all applicable laws, ensuring no violations of intellectual property or data privacy. It complies with data protection laws to maintain the confidentiality and safety of student and faculty information. Additionally, all user data will be securely stored and processed according to relevant legal guidelines. The system will also respect licensing agreements for any third-party tools or software used. Therefore, ProjectXpert is legally sound and poses no legal risks.

### MODULE DESCRIPTION

The system has three users namely; Admin, Student module and Project Guide module

**Admin**

1. Add and manage project guides (create accounts for guides and set login credentials).
2. View a list of registered project guides.
3. Add and manage student accounts (assign students under specific guides).
4. View a list of registered student groups and their project submissions.
5. Review and approve/reject submitted project abstracts.
6. View the ranking list of submitted project abstracts based on feasibility, innovation, and clarity.
7. Track the status of approved and rejected projects.
8. Monitor feedback provided by guides to student groups.
9. Schedule and manage project review sessions.
10. Generate and view reports of project activities and student progress.
11. Change their own password.
12. View a list of all project-related notifications and contact inquiries.

**Student module**

1. Register and create student accounts (set login credentials).
2. Submit multiple project abstracts for evaluation under assigned project guides.
3. View the status of submitted projects (approved, rejected, or under review).
4. Receive feedback from project guides on project abstracts.
5. Track progress based on feedback and resubmit updates.
6. View a list of all ongoing review sessions and their schedules.
7. Receive automated email notifications about project approval, feedback, and deadlines.
8. View and attend scheduled project review sessions with the guide.
9. Access and update project details, including title, abstract, and description.
10. Change their own password for secure access to the platform

**Project Guide module**

1. Register and create guide accounts (set login credentials).
2. View a list of assigned student groups and their submitted project abstracts.
3. Review submitted project abstracts and rank them based on feasibility, innovation, and clarity.
4. Provide feedback on project abstracts and suggest improvements.
5. Approve or reject project abstracts based on evaluation criteria.
6. Schedule and manage project review sessions with students.
7. Track student progress and monitor if feedback has been addressed.
8. Receive and view automated email notifications about student submissions and updates.
9. View the status of projects (approved, rejected, or in progress).
10. Track and manage multiple projects from different student groups

# CHAPTER 3 SYSTEM SPECIFICATION

### SYSTEM SPECIFICATIONS

* + 1. **HARDWARE SPECIFICATIONS**

The selection of hardware configuration is a very important task related to software development. Insufficient random-access memory may affect adversely on speed and efficiency of the entire system. The process should be powerful to handle the entire operations.

The hard disk should have sufficient capacity to store the file and application**.**

* + - * Processor : 1 2.0 GHz Intel Core i5 or equivalent
      * Hard Disk : Minimum 500 GB storage
      * RAM : 4 GB or higher
      * Input Device : Standard Keyboard and Mouse
      * Output Device : High Resolution Monitor

### SOFTWARE SPECIFICATIONS

| * Operating System : |  | Windows 7 or above |
| --- | --- | --- |
| * Web Serve | : | XAMPP |
| * Web Browser | : | Microsoft Edge/  Chrome/Firefox |
| * Front End | : | HTML, Python,  JavaScript, Boot Strap |

| * Back End * Framework | :    : | MYSQL  Python Django |
| --- | --- | --- |

## SOFTWARE TECHNOLOGY OVERVIEW

## **3.2.1 Overview of Python:**

Python is a high-level, interpreted, and general-purpose programming language known for its simplicity, readability, and versatility. It has become one of the most popular languages for web development, data science, artificial intelligence, automation, and more.

• Python was created by Guido van Rossum in the late 1980s at the Centrum Wiskunde & Informatica (CWI) in the Netherlands. Van Rossum wanted to create a language that was easy to read and simple to use.

• The first official release, Python 0.9.0, was launched in 1991. It included classes with inheritance, exception handling, and core data types like strings, lists, and dictionaries.

• Python 2.0 was released in 2000, introducing features like list comprehensions and garbage collection through reference counting. It made Python more powerful but also introduced issues that led to breaking changes later.

• Python 3.0, also called "Python 3000" and "Py3k", was released in 2008. It was a major revision that was not backward compatible with Python 2. It focused on removing redundant programming constructs and modules.

• Over the years, Python has seen numerous updates: Python 3.4, 3.5 (asyncio added), Python 3.6 (f-strings introduced), Python 3.7, Python 3.8 (walrus operator :=), and Python 3.9, 3.10, 3.11 (pattern matching).

• Python 3.12 (latest versions) continue to improve speed, memory management, and new programming constructs.

• Today, Python powers web applications, artificial intelligence systems, data analytics, scientific computing, and even Internet of Things (IoT) devices. It has a vibrant ecosystem and a massive developer community.

**Most Important Python Features and How to Use them:**

In today’s technology-driven world, Python is essential not just for developers but also for engineers, analysts, researchers, and even non-programmers. It offers a balance between simplicity and powerful capabilities, making it a popular choice for diverse applications.

**Python Features and Advantages**

* **Simple and Easy to Learn**:  
  Python’s syntax is designed to be readable and clean. It looks similar to English language, allowing even beginners to learn quickly. There are no complicated symbols, making it a perfect start for new programmers.
* **Interpreted Language**:  
  Python code is executed line-by-line, which makes debugging easier and faster. There’s no separate compilation step needed.
* **Dynamically Typed**:  
  In Python, you don’t need to declare the type of variables. The interpreter automatically infers the type at runtime.
* **High-Level Language**:  
  Python handles low-level operations like memory management internally. Developers can focus on solving problems rather than dealing with system internals.
* **Object-Oriented and Functional**:  
  Python supports object-oriented programming (classes, objects, inheritance) and functional programming (functions, lambda, map, filter) giving flexibility in designing applications
* **Extensive Standard Library:**

Python comes with a rich set of built-in libraries and modules like math, datetime, os, sys, re, csv, and more — saving time for developers.

* **Portable and Cross-Platform:**

Python programs can run on Windows, Linux, Mac, Android, and other platforms without modification.

* **Open Source and Community Development:**

Python is open-source and maintained by a huge global community. Many improvements, frameworks, and libraries come from community contributions.

* **Third-Party Libraries and Frameworks:**

Python has a rich ecosystem of third-party libraries like:

* Web: Django, Flask
* Data Science: Pandas, NumPy, Matplotlib
* AI/ML: TensorFlow, Scikit-Learn
* Automation: Selenium, PyAutoGUI

**3.2.2 Python Django**

Django is a free and open-source, Python-based web framework that runs on a web server. It follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established in the US as a 501(c)(3) non-profit.​

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of "don't repeat yourself." Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update, and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Instagram, Mozilla, Disqus, Bitbucket,Nextdoor,and Clubhouse.​

**3.2.3 MySQL**

MySQL is a powerful and widely-used open-source relational database management system (RDBMS) that plays a critical role in the backend infrastructure of the ProjectXpert project. It is responsible for storing, organizing, and managing the vast amounts of data generated by users, including their profiles, published artworks, detailed project descriptions, and client transactions. MySQL’s robust querying capabilities and support for complex data relationships allow ProjectXpert to efficiently handle data retrieval and storage, ensuring that information is both accessible and secure. Additionally, MySQL’s scalability and performance make it an ideal choice for ProjectXpert, as it can easily accommodate the platform’s growth and increasing user interactions while maintaining high levels of reliability and speed.

**3.2.4** **HTML**

HTML, or Hypertext Markup Language, is a standard programming language used to create and structure content on the World Wide Web. HTML code is used to create web pages and is interpreted by web browsers to display the content on the internet.

HTML uses a set of tags and attributes to describe the structure and content of a web page. Tags are enclosed in angled brackets (< >) and are used to markup different elements such as headings, paragraphs, images, links, and tables. Attributes provide additional information about an element and are used in conjunction with tags to define characteristics such as the size, colour, and location of an element on the page.

One of the strengths of HTML is its flexibility and simplicity. HTML code can be easily edited and updated using basic text editors, and can be used to create a wide range of content, from simple web pages to complex web applications. It is also designed to be compatible with a wide range of browsers and operating systems, making it an ideal language for creating web content that can be accessed by a diverse range of users.

HTML has evolved over time, with new versions and updates being released periodically. The latest version, HTML5, includes new features such as improved multimedia support, new input types, and support for mobile devices. HTML5 is also designed to be more semantic, which means it provides more information about the meaning of the content on a page, making it easier for search engines to index and categorize the content.

Overall, HTML is a key technology for creating and sharing content on the web. Its simplicity, flexibility, and compatibility make it an essential tool for web developers and designers, and it will continue to be a vital part of the internet for years to come**.**

**3.2.5 CSS**

CSS, or Cascading Style Sheets, is a language used to describe the presentation of HTML documents. CSS allows web designers and developers to control the layout, colors, fonts, and other visual aspects of a web page. By separating the presentation from the content, CSS makes it easier to update the visual style of a web page without changing the underlying HTML code. CSS works by selecting HTML elements and applying styles to them. The styles are defined in a separate CSS file or embedded directly in the HTML code using the <style> tag. CSS styles consist of one or more properties and their values, which define the visual characteristics of the selected elements**.**

**3.2.6 JAVASCRIPT**

JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

**3.2.7 XAMPP**

XAMPP is an open-source, cross-platform web server solution that plays an essential role in the development of the ProjectXpert project by enabling local hosting during the development phase. It integrates Apache, MySQL, PHP, and Perl, providing a complete and easy-to-use environment for testing and debugging web applications before deployment. By using XAMPP, developers can run ProjectXpert on their local machines, allowing them to simulate a live server environment without the need for an external hosting service. This setup facilitates quick iterations, real-time testing, and seamless database management, ensuring that the ProjectXpert platform functions correctly before it is moved to a production server. XAMPP's simplicity and all-in-one package make it an invaluable tool for developers working on the backend, helping to streamline the development process and ensure the project is stable and secure.

**3.2.8 Normalization:**

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible; Normalization is carried out in this system for the following reasons.

* To structure the data so that there is no repetition of data, this helps in saving space
* To permit simple retrieval of data in response to query and report requests.
* To simplify the maintenance of the data through updates, insertions and deletions.

To reduce the need to restructure or reorganize data which new application requirements arise. Primary Key is assigned for this purpose. The primary Key fields in almost all the tables help to ease the search and improve efficiency .The proposed system is using a second Normal form as it is found most suitable. The second normal from each row must contain an associated field that describes an attribute of the entry that the table describes.

# CHAPTER 4

**SYSTEM DESIGN & DEVELOPMENT**

## INFRASTRUCTURE DESIGN

System design provides an understanding of the procedure details, necessary for implementing the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system and has a major impact on the testing and implementation phases. Design is the second phase in the system development life cycle Software design is the first of the three technical activities in the software development process such as design, code writing and testing.

During this phase, the analyst schedules design activities, works with the user to determine the various data inputs to the system, plans how data will flow though the system, designs required outputs and writes program specifications. Again the analyst's activities focus on solving a user's problem in logical terms.

During this second step, analysts employ a variety of tools such as data flow diagrams, entity- relationship diagrams, data dictionaries and Gantt charts. The system's design converts the theoretical solution introduced by the feasibility study into a logical reality.

During design the analyst:

* Draws a model of the new system, using data flow and entity-relationship diagrams
* Develop methods for collecting and inputting data
* Defines the detailed data requirements with a data dictionary
* Writes program specifications
* Specifies control techniques for the system's outputs, databases and inputs.
* Identifies and orders any hardware or software that the system will need.

In the physical design phase, necessary software is developed to accept input from the user, to perform necessary calculations through the manipulation of data stored in the databases to produce the appropriate result.

### INPUT DESIGN

Input Design is the process of converting a user-oriented description of the inputs to a computer based business system into a programmer-oriented specification. The aim of making input design is to make the data entry as easy, logical and free from errors as possible. An input format should be logical and easy to understand. In the design, the user oriented inputs are converted into computer recognizable format.

The collection of data is the most expensive part of the system in terms of the equipment used, time and no. of clients involved etc. in the input design data is accepted and it can be readily used for data processing or can be stored in a database for further use. The activities used for inputs are very user friendly. Different names are associated with each data entry activity item makes data entry an easy job. Each data entry contains separate buttons for submitting the form and proper validation checking is carried out and necessary message will be presented to the user in case of improper data entry. The proposed system satisfies the following input design objectives

* A cost effective method of input
* The highest possible level of accuracy
* The input is acceptable to and understood by the user.

Input objectives are

1. **Controlling the amount of input:** Wherever user input is required, the number of keystrokes is reduced by giving possible input values as default in that area. The viewer can select the answer in a single click. The amount of information entered by the viewer using the virtual keypad is reduced to the maximum and the software is made very user friendly.
2. **Avoiding Delay:** A processing delay resulting from data preparation or data entry operations is called a bottleneck. Such bottlenecks are avoided to the maximum. The only time the viewer has to wait is when the file is uploaded or downloaded. Progress bars or progress meters are displayed to keep the user waiting and also to show the speed and amount of download.
3. **Avoiding Errors in Data:** The rate at which errors occur depends on the quantity of data. Here the quantity of data is reduced to the lowest, and a text file is easily manageable.
4. **Avoiding Extra steps:** The viewer can quit at any point of time. Even at the time of upload or download the viewer can quit. The viewer need not wait for any specific event to happen for quitting the process.
5. **Keeping the process Simple:** This implies that the system has all the measures to keep the errors out even if the user is giving wrong data. It handles the situation with grace and doesn’t create much hype about the situation to the user.

### OUTPUT DESIGN

It has been an ongoing activity from the beginning of the project. It includes the process of finalizing the medium format and exact contents of each output to be produced by the proposed system. The primary objective in creating an output is accuracy and neatness.

Efficient output design should improve the system's relationship with the user and help in decision making. The system has got the capability to display standard screen layouts. These layouts should be designed around the output requirements and they must be designed with utmost care and details in the screen layouts must be simple, descriptive and clear to the user. While designing a system's output, the analyst must make several independent decisions. Every system produces some kind of report. No matter what the content of the report. The output design of this project is made with these objectives in mind.

* + 1. **FORM DESIGN**

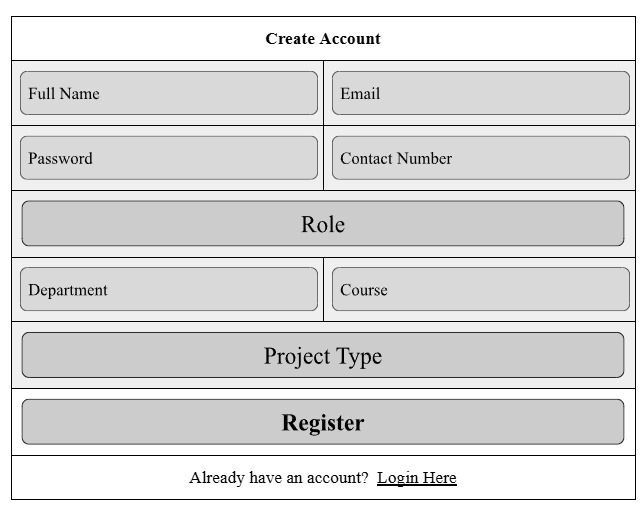
**LOGIN FORM**

| **Welcome Back**  Sign in to access your ProjectXpert dashboard | |
| --- | --- |
|  | |
|  | |
| Forgot password? | |
|  | |
| Don’t have an account? Create Account | |

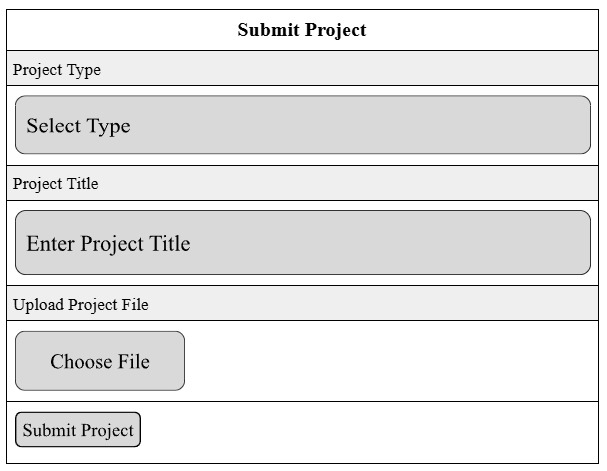
**FORGOT PASSWORD FORM**

| **Find Your Account** | |
| --- | --- |
| Please enter your email address or mobile number to search for your account. | |
|  | |
|  |  |

**REGISTRATION FORM**

****

**PROJECT SUBMISSION FORM**

****

**Output Types**

* External Outputs, whose destination is outside the organization and is the main image of the organization,
* Internal Outputs, whose destination is within the organization and which require careful design because it is the user's main interface with the Android mobile.
* Interactive Outputs, which involve the user in communicating directly with the Android mobile.

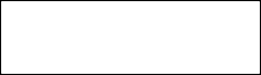
### DATA FLOW DIAGRAM

A DFD, also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flow in the system.

A data flow diagram may be used to represent a system or software at any level of abstraction DFD's can be partitioned into levels that represent increasing information flow and functional details. A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It is a structured analysis and design tool that can be used for flowcharting in place of or in association with, information oriented and process oriented system flowcharts.

When analysts prepare the DFD, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The DFD reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Basic data flow diagram symbols are:

*  A “Rectangle” defines a source or destination.
* An “Arrow” identifies data flow. It is a pipeline through which information flows.



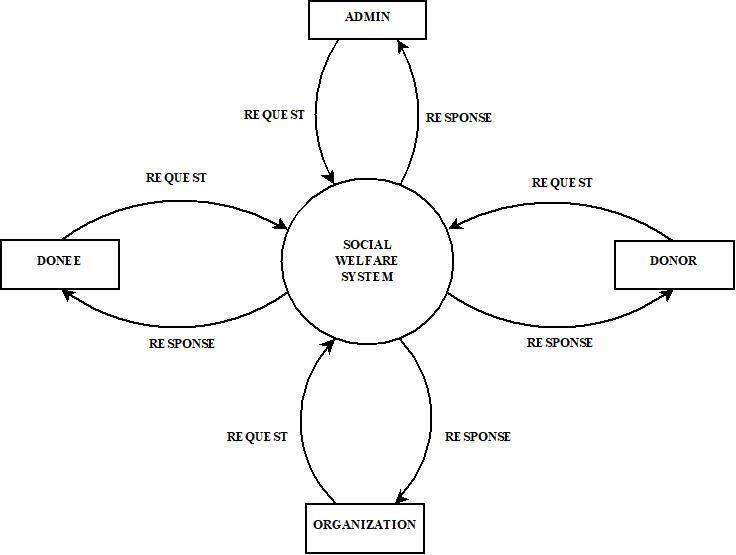
* A “Circle” represents a process that transforms incoming data flow(s) into outgoing data flow(s).
* An “Open Rectangle” is a data store

Steps to Construct Data Flow Diagrams:

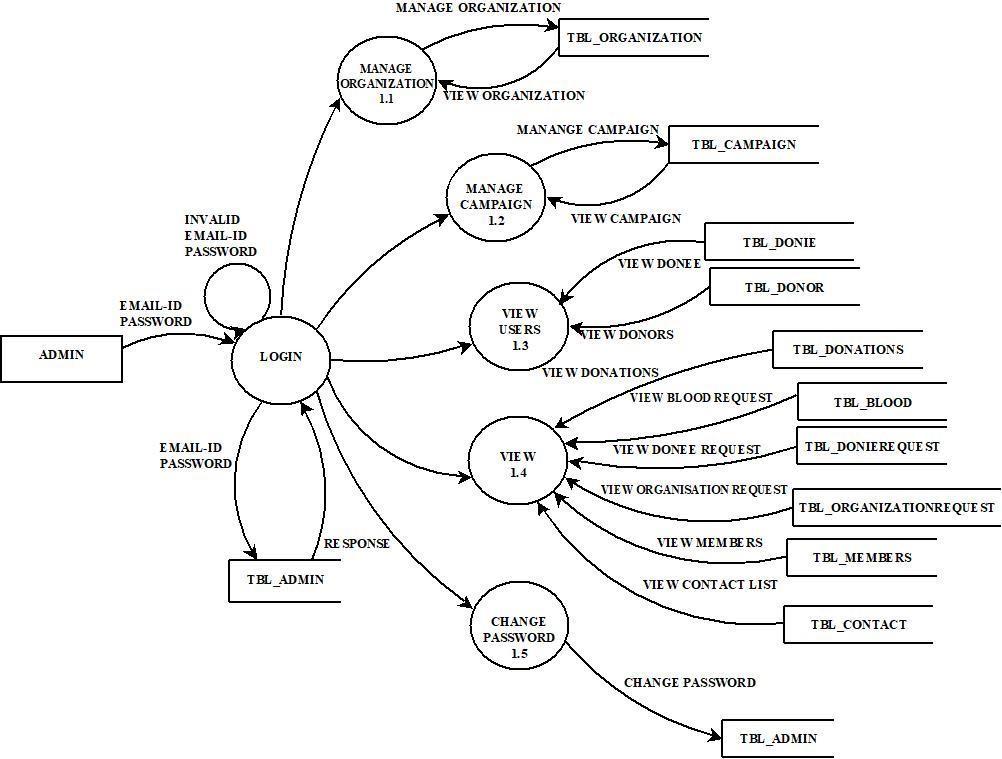
Three steps are commonly used to construct a DFD.

* Processes should be named and numbered for easy reference; each name should be representative of the process.
* The direction of flow is from top to bottom and from left to right.
* When a process is exploded into lower level details they are numbered.

### LEVEL 0 DFD (CONTEXT DIAGRAM)

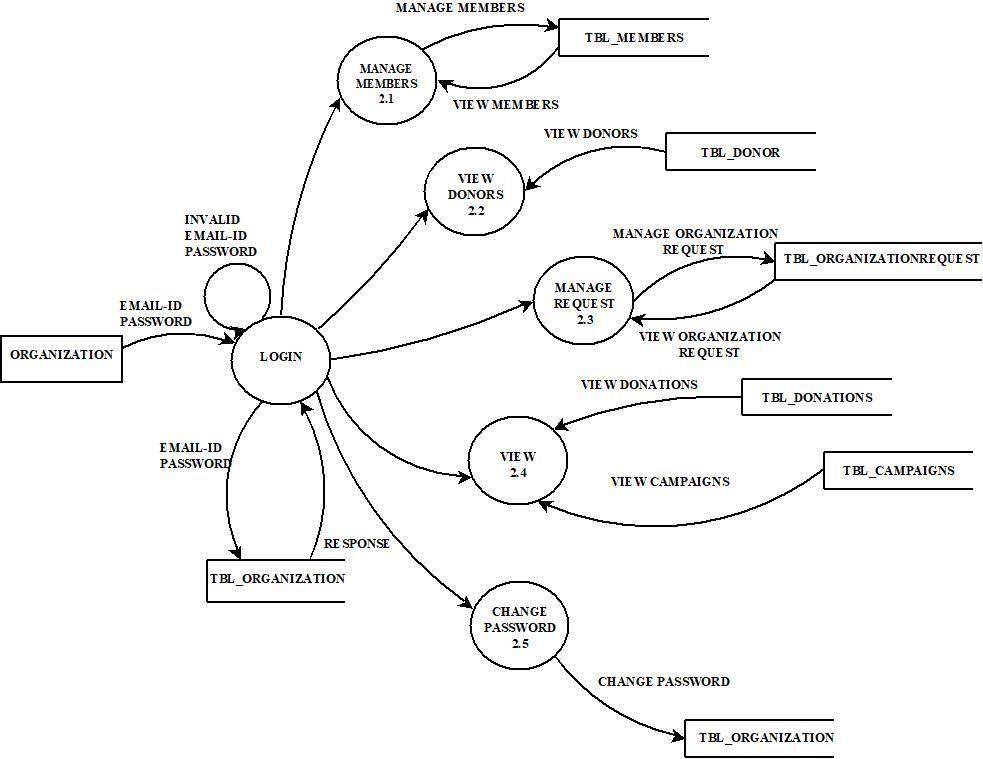


**Fig: 4.4.1 Context Diagram**

**Level 1 Admin**



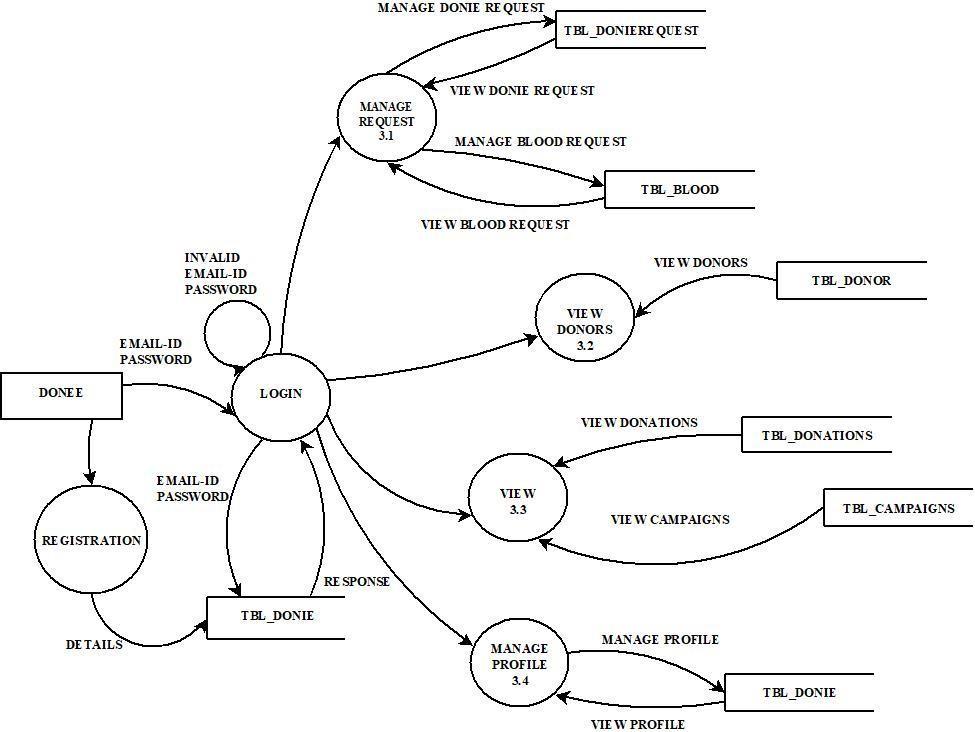
**Fig: 4.4.2 Level 1 Admin**

**Level 1 Organization**



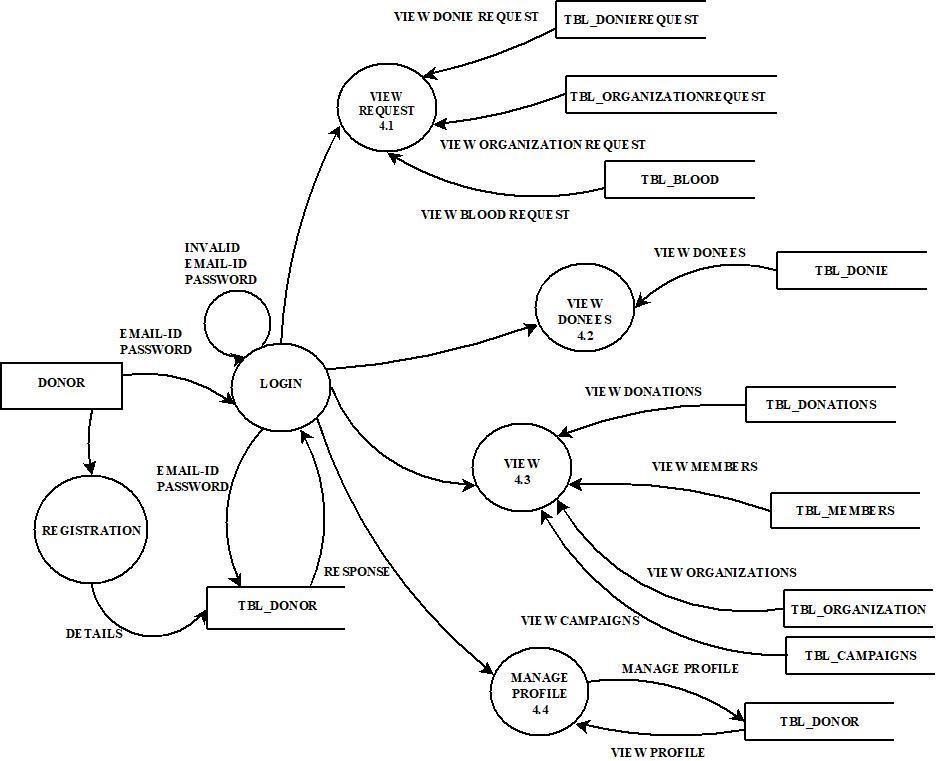
**Fig: 4.4.3 Level 1 Organization**

**Level 1 Donee**



**Fig: 4.4.4 Level 1 Donee**

**Level 1 Donor**



**Fig: 4.4.5 Level 1 Donor**

### ANALYSIS TOOLS

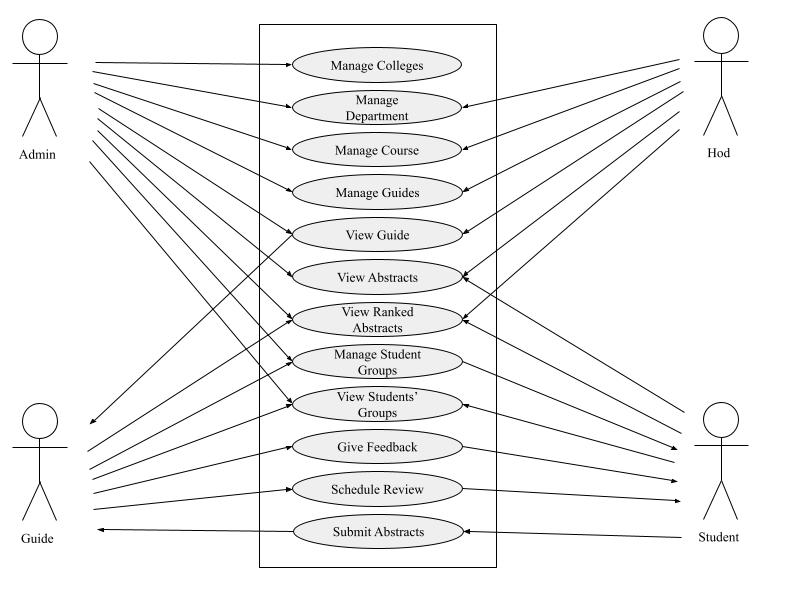
System analysis is the process of collecting and interpreting facts, understanding problems and using this information to suggest improvements in the system. This will helps to understand the existing system and determine how computers make their operations more effective.

In software engineering, a use case diagram in the Unified Modeling Language(UML) is a type of behavioral diagram defined by and created from a Use-Case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The use case diagram shows the position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/ owner/ user and the development team.

Use Case is a technique for capturing functional requirements of systems and systems-of-systems. Each use case provides one or more scenarios that convey how the system should interact with the users called actors to achieve a specific business goal or function. Use case actors may be end users or other systems. Use cases typically avoid technical jargon, preferring instead the language of the end user.

The UML standard describes graphical notation for relationships. Use cases are represented by ovals and the actors are represented by stick figures. The use case diagram of this application is given below.



**Fig: 4.5.1 Use Case Diagram**

* 1. **Database Design**

After designing the input and output activities, the designer moves to concentrate on database design. How data are organized depends on the data and response requirements that determine hardware configurations. The database is organized to ensure that the system resources are not wasted. The objective of the database design is to provide the effective auxiliary storage and contribute to the overall efficiency to the computer program components of the proposed system. The database design translates the data models that were developed for the system users during the definition phase into the data structures supported by the chosen database technology. The goals of database design are as follows:

1. A database should provide for the efficient storage, update and retrieval of data.
2. A database should be reliable; the stored data should have high priority to promote user trust in that data.
3. A database should be adaptable and scalable to new and unforeseen requirements and applications.

The techniques used to improve a data model in preparation for database design is called data analysis. Data analysis is a process that prepares a data model for implementation as a simple, non-redundant and adaptable database. The specific technique is called Normalization. Normalization is a technique that organizes the data attributes such that they are grouped to form stable, flexible and adaptive entities. The table involved in inspection process along with attributes, data types, constraints and brief description about the fields are stated in the below mentioned table.

* + 1. **Normalization**

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this application for the following reasons.

* + - * To structure the data so that there is no operation of data, this helps in saving space.
      * To permit simple retrieval of data in response to query and report request.
      * To simplify the maintenance of the data through updates, insertions and deletions.
      * To reduce the need to restructures or recognize data which new application requirements arise.

Primary key is assigned for this purpose. The primary key fields in the tables help to ease the search and improve efficiency. The proposed system is using second normal form as it is found most suitable. In second normal form each row must contain associated field that describes an attribute of the entry that the table describes.

**Table 4.6.1 tbl\_accountdetails**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of account details |
| cardname | varchar(225) | Not Null | card name |
| cardnumber | bigint(20) | Not Null | card number |
| expmonth | varchar(20) | Not Null | expiry month |
| expyear | int(11) | Not Null | expiry Year |
| cvv | int(11) | Not Null | CVV |
| amount | int(11) | Not Null | amount |

**Table 4.6.2 tbl\_admin**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(20) | Primary key | id of admin |
| email | varchar(30) | Not Null | email id |
| password | varchar(100) | Not Null | password |

**Table 4.6.3 tbl\_blood**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of blood donation request |
| doneeid | int(11) | Not Null | id of donee |
| name | varchar(30) | Not Null | name of donee |
| number | int(11) | Not Null | contact number |
| details | longtext | Not Null | details |
| date | date | Not Null | date of donation |
| regdate | timestamp | Not Null | date of registration |
| status | varchar(20) | Not Null | status of approval |

**Table 4.6.4 tbl\_campaign**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of campaigns |
| name | varchar(100) | Not Null | name of campaign |
| date | date | Not Null | date of campaign |
| venue | varchar(100) | Not Null | venue |
| purpose | longtext | Not Null | purpose of campaign |
| conductedby | varchar(100) | Not Null | conducted by |
| addeddate | timestamp | Not Null | date of registration |
| status | varchar(20) | Not Null | status of approval |

**Table 4.6.5 tbl\_contact**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of contact list |
| name | varchar(20) | Not Null | name of user |
| email | varchar(200) | Not Null | email id |
| message | longtext | Not Null | message |
| reply | longtext | Not Null | reply |
| status | varchar(20) | Not Null | status of verification |
| date | timestamp | Not Null | date of request |

**Table 4.6.6 tbl\_donations**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of donations |
| donorid | int(11) | Not Null | id of donor |
| recieverid | int(11) | Not Null | id of receiver |
| amount | int(11) | Not Null | amount |
| role | varchar(25) | Not Null | role |
| date | timestamp | Not Null | date of donation |

**Table 4.6.7 tbl\_donie**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of donees |
| name | varchar(30) | Not Null | name of donee |
| email | varchar(30) | Not Null | email id |
| gender | varchar(20) | Not Null | gender |
| age | int(11) | Not Null | age |
| address | longtext | Not Null | address |
| number | int(11) | Not Null | contact No. |
| purpose | longtext | Not Null | purpose of donee |
| password | varchar(100) | Not Null | password |
| status | varchar(20) | Not Null | status of verification |
| date | timestamp | Not Null | date of registration |

**Table 4.6.8 tbl\_donierequest**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of donee requests |
| donieid | int(11) | Not Null | id of donee |
| subject | varchar(100) | Not Null | subject |
| description | longtext | Not Null | description |
| holdername | varchar(50) | Not Null | name of acct holder |
| accountnumber | varchar(50) | Not Null | account number |
| ifsc | varchar(50) | Not Null | IFSC code |
| branchname | varchar(50) | Not Null | name of branch |
| date | timestamp | Not Null | date of request |
| status | varchar(20) | Not Null | status of approval |

**Table 4.6.9 tbl\_donor**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of donors |
| name | varchar(30) | Not Null | name of donor |
| email | varchar(30) | Not Null | email id |
| gender | varchar(20) | Not Null | gender |
| age | int(11) | Not Null | age |
| address | longtext | Not Null | address |
| number | int(11) | Not Null | contact No. |
| blood group | varchar(20) | Not Null | blood group of donor |
| photo | varchar(100) | Not Null | photo of donor |
| password | varchar(100) | Not Null | password |
| status | varchar(20) | Not Null | status of verification |
| date | timestamp | Not Null | date of registration |

**Table 4.6.10 tbl\_members**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of members |
| organizationid | int(11) | Not Null | id of organization |
| name | varchar(50) | Not Null | name of member |
| dateofbirth | date | Not Null | member DOB |
| gender | varchar(20) | Not Null | gender |
| date | timestamp | Not Null | date of registration |

**Table 4.6.11 tbl\_organization**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of organizations |
| name | varchar(50) | Not Null | name of organization |
| type | varchar(50) | Not Null | type of organization |
| address | longtext | Not Null | address |
| number | int(11) | Not Null | contact no. |
| email | varchar(50) | Not Null | email id |
| password | varchar(100) | Not Null | password |
| date | timestamp | Not Null | date of registration |

**Table 4.6.12 tbl\_organizationrequest**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| id | int(11) | Primary key | id of organization requests |
| organizationid | int(11) | Not Null | id of organization |
| subject | varchar(100) | Not Null | subject |
| description | longtext | Not Null | description |
| holdername | varchar(50) | Not Null | name of acct holder |
| accountnumber | varchar(50) | Not Null | account number |
| ifsc | varchar(50) | Not Null | IFSC code |
| branchname | varchar(50) | Not Null | name of branch |
| date | timestamp | Not Null | date of request |
| status | varchar(20) | Not Null | status of approval |

### E-R DIAGRAM

An entity-relationship (ER) diagram is specialized graphic that illustrates the interrelationship between entities in a database Boxes are commonly used to represent entity Diamonds are normally used to represent relationships and ovals are used to represent attributes.

An entity is piece of data is shared between entities.

* **Classifying Relationships**

Relationships are classifieds by their degree, connectivity, cardinality, direction, type and existence.

* **Degree of Relationships**

The degree of a relationship is the number of entities associated with the relationship. The n- array relationship is there general from for degree n. Special cases are binary, ternary where the degree is 2 and 3 respectively.

* **Connectivity and Cardinality**

The connectivity of a relationship describes the mapping of associated entity instances in the relationship. The values of connectivity are "one" or "many" The cardinality of a relationship is the actual number of related occurrences for each of the two entities. The basic types of connectivity of relations are: One-to-one, one-to- many, many-to-many.

**A one-to-one (1:1)** is when at most one instance of an entity A is associated with one instance of entity B.

**0-to-many (1: N)** Is when for an instance of a entity A, there are zero, one or many instances of entity B, but for instance of the entity B, there is only one instance of the entity A.

**A many-to-many (M: N)** relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one or more instances of entity B and for one instance of entity B there are zero, one or many instances of entity A.

The symbols used ER Diagram is Entity

Attributes

Relationships

Lines

**FIG: 4.7 ER Diagram**

# CHAPTER 5 SYSTEM TESTING

### INTRODUCTION

System testing is actually a series of different testes whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work together to verify that the system elements have been properly integrated and perform all functions clearly.

System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

Testing on this project can be done in many ways, such as module testing, where every single program module is examined thoroughly, this project is also divided into many modules such service, user interface, viewing schedules and deleting schedules etc. Also, the whole unit will be tested as every data entered and searched will also be tested; all the runtime errors can be detected and corrected accordingly. There are also other types of testing such as integration testing; validation testing etc. the whole project is integrated so it has to be tested at each and every point.

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. A test case is one that has a possibility of finding as yet undiscovered error. A successful test is one that uncovers as yet undiscovered error. The developed system is tested whether it works efficiently and whether it satisfies all the user requirements by taking a series of test cases.

* + 1. **Types of Testing**

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be least creative phase of the whole cycle of the system design. In the real sense it is the phase, which helps to bring out the creativity of the phases. No system design is

ever perfect. Errors occur due to communication problems, programmer’s negligence or time constraints. All these must be eliminated before the system is ready for user acceptance testing. Different levels of testing are employed during different stages of the system development.

* + - 1. **White Box Testing**

By using this technique, it was tested that all the individual logical paths were executed at least once as logical decisions were tested on both their true and false sides. All the loops were tested with data in between the ranges and especially at the boundary values.

* + - 1. **Black Box Testing**

By the use of this technique, the missing functions were identified and placed in their positions. The errors in the interfaces were identified and corrected. This technique was used to identify the initialization and termination errors and correct them.

* + - 1. **Unit Testing**

In the lines of this strategy, all the individual functions and modules were put to the test independently. By following this strategy, all the errors in coding were identified and corrected. This method was applied in combination with the White Box and Black Box testing techniques to find the errors in each module.

* + - 1. **Integration Testing**

This testing strategy has two different approaches namely the top down approach, in which the integrations are carried out from the top level module to the bottom and bottom up approach in which the integration is carried out from the low level module to the top. The modules were tested using the bottom up approaches by introducing stubs for the top-level functions. This test was used to identify the errors in the interfaces, the errors in passing the parameters between the functions and to correct them.

* + - 1. **Validation Testing**

Validation testing can be defined in many ways, but a simple definition that validation succeeds when the software in a manner that is reasonably expected by the customer.

Software validation is achieved through a series of black box test that demonstrate conformability with requirements. After validation test have conducted, one of the two possible conditions exists.

* + - * + The function or performance characterized confirm the specification and are accepted.
        + Deviation from specification is found and a deficiency list is created.
      1. **Output Testing**

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in a specific format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration.

The output format of a screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, output comes out as the specified requirements by the user. Hence output testing does not result in any correction in the system.

### TEST CASES

A specific set of steps and data along with expected results for a particular test objective. A test case should only test one limited subset of a feature of functionality. Test case documents for each functionality/ testing areas will be written, reviewed and maintained separately in Excel sheets.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error message s and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

# CHAPTER 6 SYSTEM IMPLEMENTATION

### INTRODUCTION

Implementation is the stage of project when the theoretical design is turned into a working system. At this stage, the main workload, the greatest upheaval and the major impact on existing practices shift to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. The implementation stage is a system project in its own right. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, training of staff in the changeover procedure and evaluation of changeover methods.

The implementation is the final and important phase. It involves user training system testing in order to ensure successful running of the proposed system. Once the system design phase is over, the next stage is to implement and monitors the operation of the system to ensure that it continues the work effectively and efficiently.

The three main phases in implementation take place in series. These are the initial installation, the test of the system as a whole and evaluation maintenance and control of the system. The implementation plan and action to implement should be bound closely together. The implementation plan is a function of line management at least as far as key decision or alternative plans are concerned. The implementation plan was to convert the existing clerical files to the computer. The implementation plan listed all sub tasks so that individuals in the organization may be assigned specific responsibilities.

The installation of the new system that is bound to replace the current one may require a major revision of computer facilities as well as completely new after space. Space planning took into account the space occupied by the people, space by equipment and the movement of people and equipment in the working investment. After conduction the initial testing the system is loaded on the client office's computer. Some of the user employees in this case are selected. These users are trained first and they run the system. A detailed documentation is prepared to this set of employees. There may be slight modifications to meet the organization.

After all modifications specified by the users in the documentation are made, the computer system is run along with manual system. Even though this kind of parallel run make extra burden to the employees and management, the system is run in parallel for the sake of checking reliability and efficiency.

After this document which compares the result of the manual system with those of the computerized is prepared. In the case of management many of whom participated in the development of the system short seminars were given. Particular attention was paid to the training of end users. The training sessions were aimed at giving the user staff the specific skills required in their new jobs. Education involved creating the right atmosphere and motivation of user staff. It explained the need for changes and helped to overcome the resentment caused by the feeling that computers took away the responsibility from individual departments.

Various measures have been taken by department officials in order to find suitable solutions by the following issues

* About the skill to be acquired.
* Reduction of man power in department.
* About the new form having all required option.

### IMPLEMENTATION PROCEDURE

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the department. If the implementation is not carefully planned and controlled, it can cause confusion. Implementation includes all those activities that take place to convert from the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements.

Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system into the actual use is called system implementation. This includes all those activities that

take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification of the system.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently. It involves careful planning, investigation of the current system and is constraints on implementation, design of methods to achieve the changeover. The more the complex system being implemented is, the more involved will be the system analysis and the design effort required for its implementation.

### EQUIPMENT ACQUISITION

Here, the necessary equipment is acquired to implement the system. Major steps involved in the implementation are installation of software. In case of this system there is no special hardware requirement for the working of the software. In addition to basic hardware requirements one thing that is essential is the software requirements with an operating system that has server support. The proposed system can run on any PC, which works on android and any versions of it. Our software is platform dependent and will run only on the windows platform after installing the software, it is essential to ensure that the software is working accordingly with the existing software.

## TRAINING

For this system, it was explained to the users how to use the system, what details are to be given while creating a new profile, how to use it so that we get the maximum output out of it. The proposed system may be entirely new, replacing an existing one or it may be modifications to the existing system. In either case, proper implementation is necessary to provide a reliable system to meet organizational requirements.

### EVALUATION

Evaluation of the system is performed to identify its strength and weaknesses.

### OPERATIONAL EVALUATION

Assessment of the manner in which the system functions, including ease of use, response time, overall reliability and level of utilization.

### ORGANIZATIONAL IMPACT

Identification and measurement of benefits to the organization in such areas like financial concerns, operational efficiency and competitive impact.

### USER MANAGEMENT ASSESSMENT

Evaluation of attitude of senior and user managers within the organization, as well

as end users.

### DEVELOPMENT PERFORMANCE

Evaluation of the development process based on overall development time and effort, conformance to budgets and standards, and other project management criteria includes assessment of development methods and tools.

### DOCUMENTATION

After the testing and implementation was completed, the whole system was presented and documented in a readable manner. This was done to ensure that any corrections, manipulations or updating are performed in future, the users would face no problem in performing those changes. Documentations include the source code, the tables that were used to construct the base for the system, the framework which is bound to the programs etc.

# CHAPTER 7 SOFTWARE MAINTENANCE

Software maintenance denoted any changes made to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. Most products need maintenance due to wear and tear by use. Although software does not wear out like a piece of hardware it “ages” and eventually fails to perform. So maintenance becomes necessity. Types of software maintenance:

### CORRECTIVE MAINTENANCE

Corrective maintenance of a software product is necessary either to rectify the bugs observed while the system is in use.

### ADAPTIVE MAINTENANCE

A software product might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to interface with new hardware or software.

### PERFECTIVE MAINTENANCE

A software product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to customer demands, or to enhance the performance of the system.

### PREVENTIVE MAINTENANCE

Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

# CHAPTER 8 CONCLUSION

The project was successfully completed within the time span allotted. Every effort

has been made to present the system in a more –user friendly manner. All the activities provide a feeling like an easy walk over to the user who is interfacing with the system. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for social welfare activities.

This project helped us in gaining valuable information and practical knowledge on several topics likes deigning web pages using PHP, usage of responsive templates and customized user forms. The project is created using MySQL as back end which also helped to study about the various operations associated with MySQL. The system as a whole is secured. Also the project helped us in understanding about the development phases of a project and software development life cycle. This project has given greater satisfaction because we developed a project which helps to promote the welfare activities in Kerala.

# CHAPTER 9 FUTURE ENHANCEMENT

Every application has its own merits and demerits. The project has covered almost

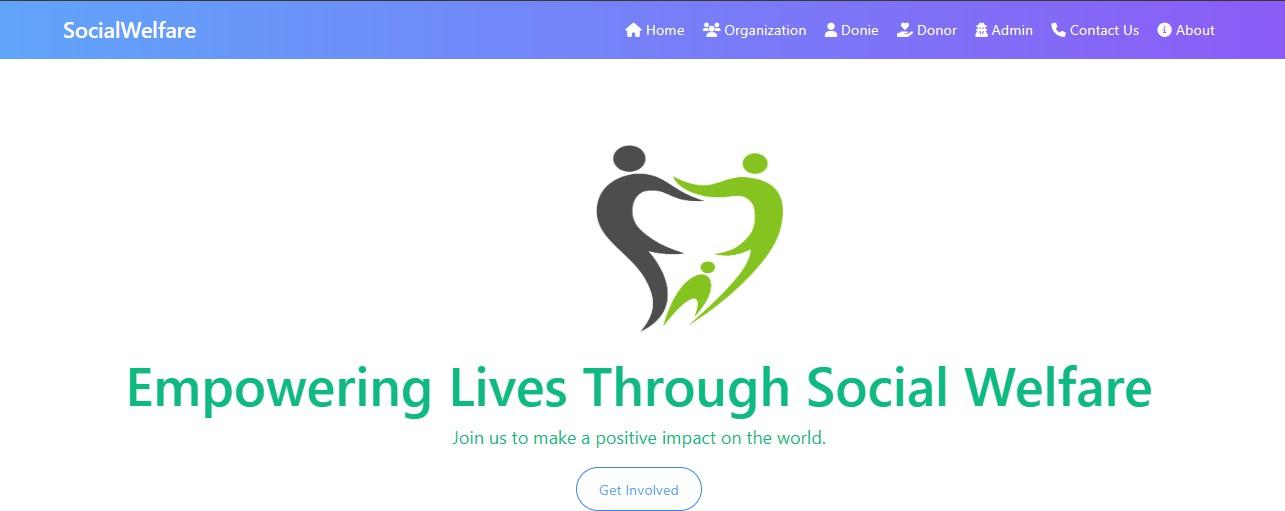
all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the website functions very attractive and useful manner than the present one. The project also helps in doing a global level social welfare activities.

Future enhancements for this website could include the development of a mobile application for greater accessibility, integration of geolocation features to find nearby assistance requests, real-time chat functionality for improved user communication, seamless payment gateway integration for enhanced donation processes, and the implementation of machine learning algorithms to refine matching between donors and recipients. Additionally, features like volunteer management, blockchain-based transparency, gamification elements, multilingual support, and accessibility features could further elevate the user experience. These enhancements aim to expand the platform's capabilities, user base, and impact in facilitating medical support, financial aid, and blood donation, ultimately creating a more connected and supportive community.

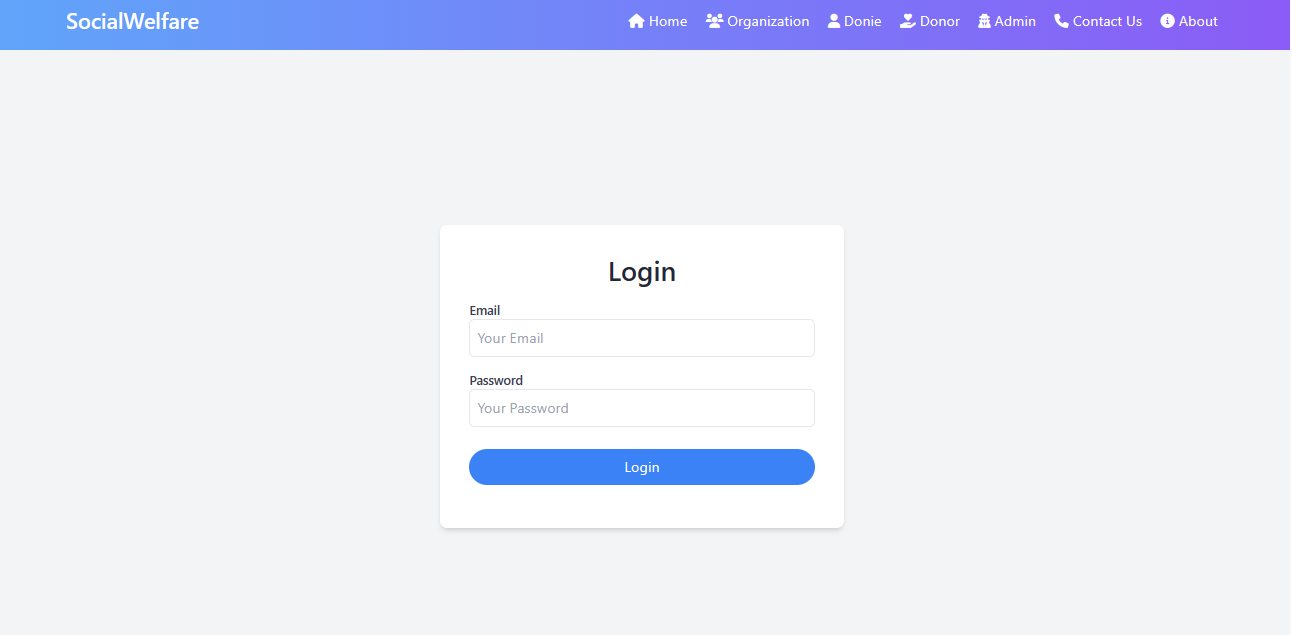
As with any software in any field, there is always room for improvement and changes that should be made over time in order to incorporate constantly evolving technology and also to continuously engage people who quickly adopt to those new tools.

# CHAPTER 10 APPENDIX

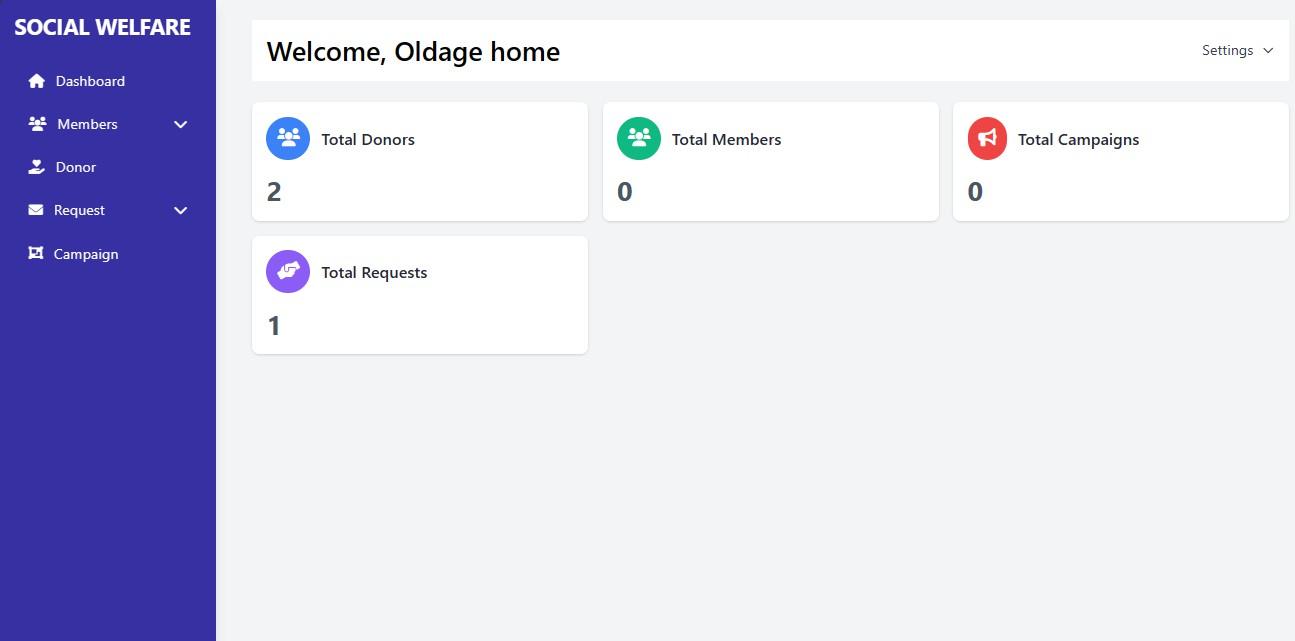
### SCREENSHOTS



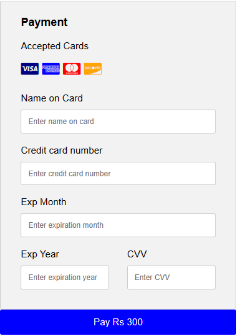
**Fig: 10.1.1 Home Pages**



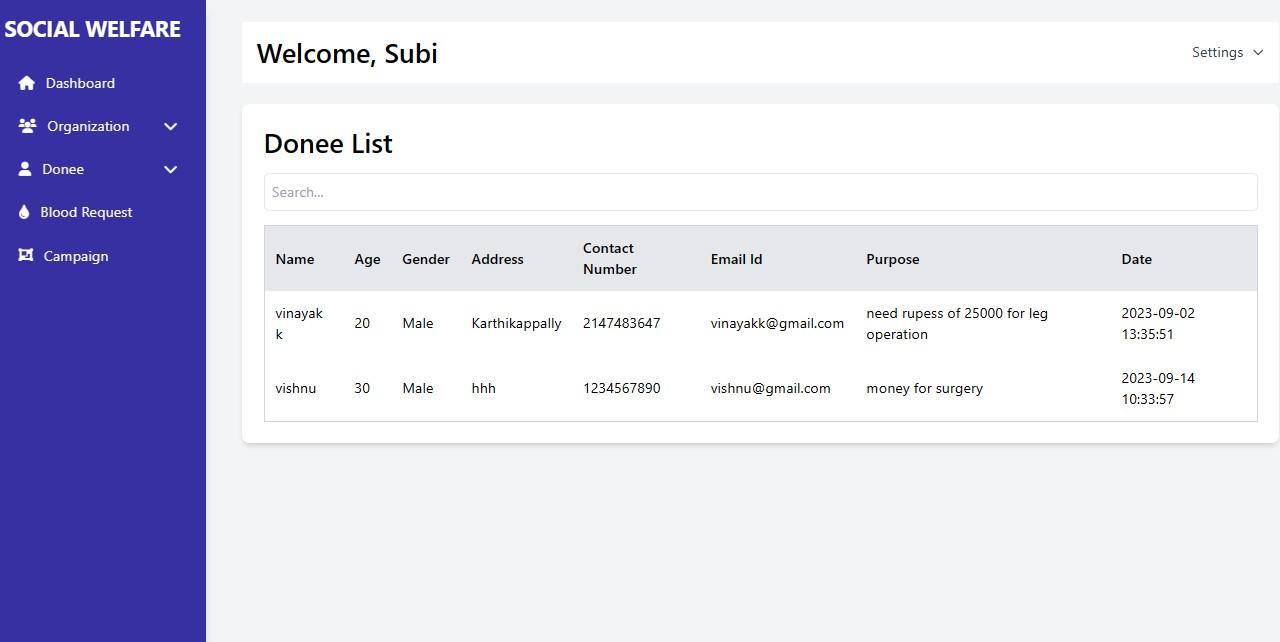
**Fig: 10.1.2 Admin Login Page**



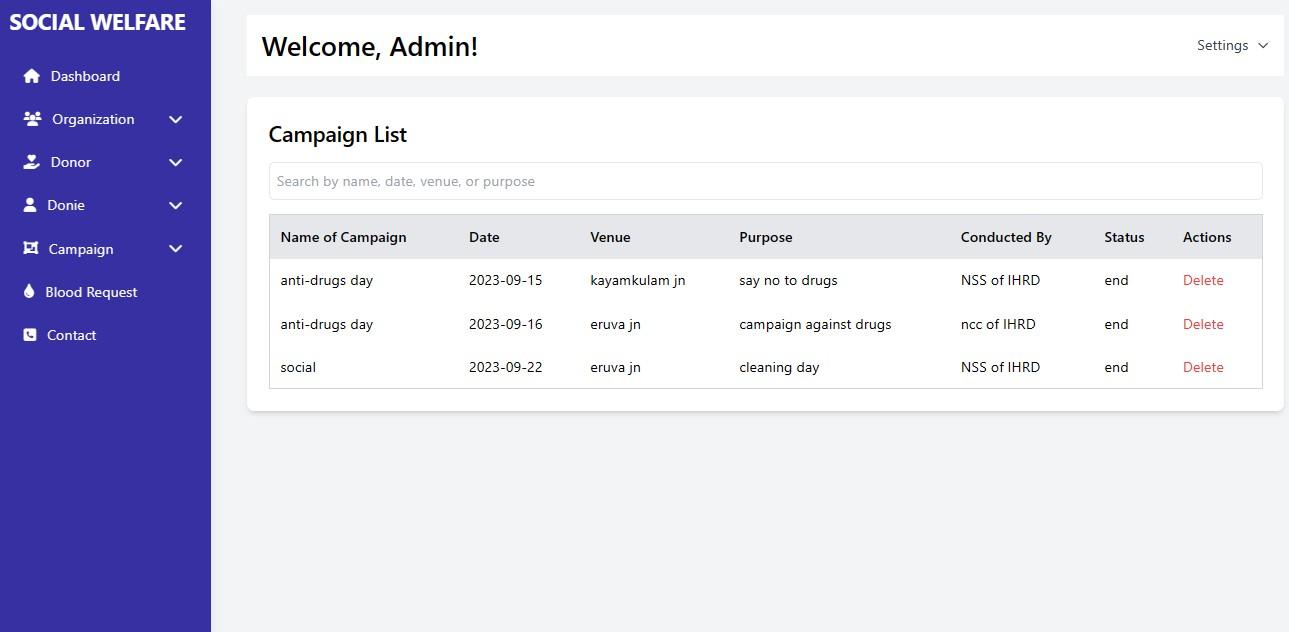
**Fig: 10.1.3 Organization Dashboard**



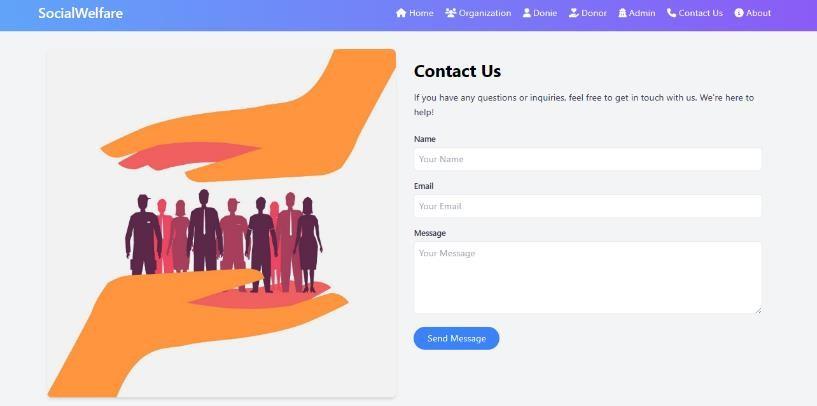
**Fig: 10.1.4 Payment page**



**Fig: 10.1.5 Donee List**



**Fig: 10.1.6 Campaign List**



**Fig: 10.1.7 Contact Form**

### SAMPLE SOURCE

**configure.php**

<?php define('DB\_SERVER','localhost'); define('DB\_USER','root');

define('DB\_PASS' ,''); define('DB\_NAME', 'social');

$con = mysqli\_connect(DB\_SERVER,DB\_USER,DB\_PASS,DB\_NAME);

// Check connection

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

$sql = "SELECT \* FROM campaign";

$result = mysqli\_query($con, $sql);

if ($result) {

while ($row = mysqli\_fetch\_assoc($result)) {

$id = $row['id'];

$endDate = $row['date'];

// Compare the current date with the end date

$currentDate = date('Y-m-d'); if ($currentDate > $endDate) {

// Update the status of the campaign to "end"

$updateSql = "UPDATE campaign SET status = 'end' WHERE id = '$id'";

$updateResult = mysqli\_query($con, $updateSql);

}

else if($currentDate==$endDate){

$updateSql = "UPDATE campaign SET status = 'active' WHERE id = '$id'";

$updateResult = mysqli\_query($con, $updateSql);

}

}

} else {

// echo "Error fetching campaigns from the database.<br>";

}

**index.php**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Social Welfare System</title>

<link href="[https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css](https://cdn.jsdelivr.net/npm/tailwindcss%402.2.19/dist/tailwind.min.css)" rel="stylesheet">

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">

<style>

@keyframes fadeInUp {

from { opacity: 0; transform: translateY(20px); } to { opacity: 1; transform: translateY(0); }

}

@keyframes scaleIn {

from { transform: scale(0.8); opacity: 0; } to { transform: scale(1); opacity: 1; }

}

</style>

</head>

<body class="bg-gray-100">

<!-- Navigation Bar -->

<?php include('navbar.php'); ?>

<!-- Hero Section -->

<div class="relative bg-white h-screen flex items-center justify-center text-white overflow- hidden">

<div class="absolute inset-0 mx-auto ml-96 mt-24">

<img src="image1.png" alt="Background Image" class="justify-center ml-72 w-52 h-52">

</div>

<div class="relative z-10 text-center p-6 animate-fadeInUp text-green-500 mt-24">

<h1 class="text-4xl md:text-6xl font-semibold mb-4">Empowering Lives Through Social Welfare</h1>

<p class="text-lg md:text-xl mb-8">Join us to make a positive impact on the world.</p>

<a href="#" class="bg-white text-blue-400 hover:bg-green-500 hover:text-white px-6 py-3 rounded-full border border-blue-500 transition-colors text-green-500 duration-300">Get Involved</a>

</div>

</div>

<!-- Featured Content Section -->

<div class="bg-gray-100 py-12">

<div class="container mx-auto text-center">

<h2 class="text-2xl font-semibold mb-4">Our Services</h2>

<div class="flex flex-wrap justify-center">

<div class="w-full md:w-1/2 lg:w-1/3 p-4 animate-fadeInUp">

<div class="bg-white p-6 rounded-lg shadow-md">

<i class="fas fa-hands-helping text-purple-500 text-4xl mb-4"></i>

<h3 class="text-lg font-semibold mb-2">Support Programs</h3>

<p class="text-gray-700">Explore various support programs for those in need.</p>

</div>

</div>

<div class="w-full md:w-1/2 lg:w-1/3 p-4 animate-fadeInUp">

<div class="bg-white p-6 rounded-lg shadow-md">

<i class="fas fa-people-carry text-purple-500 text-4xl mb-4"></i>

<h3 class="text-lg font-semibold mb-2">Community Outreach</h3>

<p class="text-gray-700">Engage with the community through our outreach initiatives.</p>

</div>

</div>

<div class="w-full md:w-1/2 lg:w-1/3 p-4 animate-fadeInUp">

<div class="bg-white p-6 rounded-lg shadow-md">

<i class="fas fa-heart text-purple-500 text-4xl mb-4"></i>

<h3 class="text-lg font-semibold mb-2">Volunteer Opportunities</h3>

<p class="text-gray-700">Become a volunteer and create positive change.</p>

</div>

</div>

</div>

</div>

</div>

<!-- Our Impact Section -->

<div class="bg-gray-100 py-12">

<div class="container mx-auto text-center">

<h2 class="text-2xl font-semibold mb-4">Our Impact</h2>

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">

<div class="bg-white p-6 rounded-lg shadow-md animate-fadeInUp">

<img src="https:/[/www.freeiconspng.com/uploads/education-png-3.png"](http://www.freeiconspng.com/uploads/education-png-3.png) alt="Impact Image 1" class="w-full h-64 mb-4">

<h3 class="text-lg font-semibold mb-2">Education Programs</h3>

<p class="text-gray-700">Providing quality education to underserved communities.</p>

</div>

<div class="bg-white p-6 rounded-lg shadow-md animate-fadeInUp">

<img src="health.jpeg" alt="Impact Image 2" class="w-full h-64 mb-4">

<h3 class="text-lg font-semibold mb-2">Health Initiatives</h3>

<p class="text-gray-700">Promoting healthcare awareness and access for all.</p>

</div>

<div class="bg-white p-6 rounded-lg shadow-md animate-fadeInUp">

<img src="enironment.png" alt="Impact Image 3" class="w-full h-64 mb-4">

<h3 class="text-lg font-semibold mb-2">Environmental Projects</h3>

<p class="text-gray-700">Sustainable efforts to protect our environment.</p>

</div>

</div>

</div>

</div>

<!-- Footer -->

<footer class="bg-purple-500 text-white text-center py-6">

<p>&copy; 2023 SocialWelfare. All rights reserved.</p>

</footer>

</body>

</html>

**contact.php**

<?php include('config.php'); if(isset($\_POST['submit']))

{

$name=$\_POST['name'];

$message=$\_POST['message'];

$email=$\_POST['email'];

$query=mysqli\_query($con,"insert into contact(name,email,message) values('$name','$email','$message')");

if($query)

{

echo "<script>alert('Successfully send admin will gave you reply through gmail');</script>";

//echo "<script>window.location.assign('contact.php');</script>";

}

else{

echo "<script>alert('failed');</script>";

}}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Contact Us - Social Welfare System</title>

<link href="[https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css"](https://cdn.jsdelivr.net/npm/tailwindcss%402.2.19/dist/tailwind.min.css) rel="stylesheet">

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">

<style>

@keyframes fadeInUp {

from { opacity: 0; transform: translateY(20px); } to { opacity: 1; transform: translateY(0); }

}

</style>

</head>

<body class="bg-gray-100">

<?php include('navbar.php'); ?>

<!-- Contact Us Section -->

<section class="py-4">

<div class="container mx-auto">

<div class="flex flex-col md:flex-row items-center">

<div class="md:w-1/2 p-4 animate-fadeInUp">

<img src="contact.png" alt="Contact Us" class="w-full h-auto rounded-lg shadow-

md">

</div>

<div class="md:w-1/2 p-4 animate-fadeInUp -mt-16">

<h2 class="text-3xl font-bold mb-4">Contact Us</h2>

<p class="text-gray-700 leading-relaxed">

If you have any questions or inquiries, feel free to get in touch with us. We're here

to help!

</p>

<form class="mt-6" method="post">

<label class="block text-sm font-medium text-gray-700">Name</label>

<input type="text" name="name" class="mt-1 p-2 w-full border rounded-md focus:outline-none focus:ring focus:border-blue-400" placeholder="Your Name" required>

<label class="block mt-4 text-sm font-medium text-gray-700">Email</label>

<input type="email" name='email' class="mt-1 p-2 w-full border rounded-md focus:outline-none focus:ring focus:border-blue-400" placeholder="Your Email" required>

<label class="block mt-4 text-sm font-medium text-gray-700">Message</label>

<textarea name='message' class="mt-1 p-2 w-full h-32 border rounded-md focus:outline-none focus:ring focus:border-blue-400" placeholder="Your Message" required></textarea>

<button type="submit" name='submit' class="mt-4 bg-blue-500 hover:bg-blue-600 text-white px-6 py-2 rounded-full transition-colors duration-300">Send Message</button>

</form>

</div>

</div>

</div>

</section>

<!-- Footer -->

<footer class="bg-purple-500 text-white text-center py-6">

<p>&copy; 2023 SocialWelfare. All rights reserved.</p>

</footer>

</body>

</html>

### GANTT CHART

The Gantt chart was developed by Henry Gantt. Those are used in software project management and enhanced version of standard Gantt chart. These are mainly allocating resources to activities. It is a special type of bar chart. Each bar represents an activity. Bars are drawn alone a time line. Length of each bar is proportional to duration of time planned for corresponding activity.



# CHAPTER 11 REFERENCE

**Text Books:**

* + - **Fundamentals of Software Engineering, Fifth Edition, Rajib Mall- PHI 2018**
    - **Software Engineering: A Practitioner's Approach (IRWIN COMPUTER SCIENCE) Hardcover – 16 March 2014**

**Websites:**

* + - **http**[**s://www.tutorialspoint.com/uml/uml\_class\_diagram.htm**](http://www.tutorialspoint.com/uml/uml_class_diagram.htm)
    - [**https://www.w3schools.com/php/**](https://www.w3schools.com/php/)
    - [**https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/**](https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/)
    - [**https://myiee.org/documentation-requirements**](https://myiee.org/documentation-requirements)
    - [**https://www.w3schools.com/mysql/default.asp**](https://www.w3schools.com/mysql/default.asp)