WEB BASED ORPHANGE SYSTEM

A PROJECT REPORT

Submitted by

ANIE H SYNDHIYA
(Reg. No: 24MCR003)

GOBINATH M
(Reg. No: 24MCR026)

MADHUBALAN C N
(Reg. No: 24MCR062)

in partial fulfilment of the requirements

for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638 052

DECEMBER 2024

DEPARTMENT OF COMPUTER APPLICATIONS

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638 052

DECEMBER 2024

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **“WEB BASED ORPHANGE SYSTEM”** is the bonafide record of project work done by **ANIE H SYNDHIYA (24MCR003), GOBINATH M (24MCR026)** and **MADHUBALAN CN(24MCR062)** in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications of Anna University, Chennai during the year 2022-2023.

SUPERVISOR	**HEAD OF THE DEPARTMENT**
Date:	**(Signature with seal)**

Submitted for the mini project viva voce examination held on ___________________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We affirm that the project entitled **“WEB BASED ORPHANGE SYSTEM”** being submitted in partial fulfilment of the requirements for the award of Master of Computer Applications is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidates.

ANIE H SYNDHIYA
(Reg. No: 24MCR003)

GOBINATH M(Reg. No:24MCR026)

MATHUBALAN C N
(Reg. No: 22MCR062)

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date: **Name and Signature of the Supervisor**

[T.Kalpana]

ABSTRACT

The Web-Based Orphanage Management System is an innovative project designed to enhance the operations of orphanages by creating a centralized and user-friendly online platform. The system facilitates seamless interaction among three key stakeholders: the admin, trust, and donors, streamlining their roles to ensure efficient management and effective allocation of resources. Developed using PHP as the programming language for dynamic web functionality and MySQL as the database for secure and efficient data storage, this platform addresses the need for transparency, accountability, and ease of use in managing orphanage operations and donations.

The admin plays a crucial role in the system by managing key operations, including maintaining records of orphaned children, tracking inventory levels for food and clothing, and monitoring incoming donations. The admin ensures that all contributions are recorded accurately and utilized effectively. The trust, as the primary caretaker and intermediary, oversees the proper use of resources, maintains transparency, and updates donors on the impact of their contributions. Trust members can track inventory levels, assess ongoing needs, and generate reports to provide accountability.

The donor module enables individuals to register and contribute to the orphanage conveniently. Donors can make contributions in the form of money, clothing, or food items. The platform provides donors with the ability to view the history of their contributions and receive real-time updates on how their donations are being used. This transparency builds trust and fosters a lasting relationship between the donors and the orphanage.

The system incorporates features such as secure authentication to protect sensitive data, role-based access control to define user permissions, and an intuitive interface optimized for both desktop and mobile devices to enhance usability. By providing real-time updates, generating detailed usage reports, and ensuring secure online transactions, the system instills confidence in donors and improves the operational efficiency of the orphanage.

The Web-Based Orphanage Management System serves as a bridge connecting donors with the orphanage, promoting collaboration and ensuring that resources are allocated where they are needed most. Through its efficient and scalable design, the system is capable of supporting the growing needs of orphanages while fostering a culture of transparency and accountability. This project not only simplifies donation processes but also ensures that the lives of orphaned children are significantly improved

ACKNOWLEDGEMENT

We respect and thank our correspondent, **Thiru.A.K.ILANGO BCom., MBA., LLB.,** and our Principal **Dr.V.BALUSAMY BE(Hons)., MTech., PhD.**, Kongu Engineering College, Perundurai for providing us with the facilities offered.

We convey our gratitude and heartfelt thanks to our Professor and the Head of the Department, Professor **Dr.A.TAMIARASI Msc.,M.phil., PhD.,M.Tech** Department of Computer Applications, Kongu Engineering College, for his perfect guidance and support that made this work to be completed successfully.

We would like to express our gratitude and sincere thanks to our project coordinator **Dr.T.KAVITHA MCA., MPhil., PhD.,** Assistant professor(Sr.G) Department of Computer Applications, Kongu Engineering College, who has motivated us in all aspects for completing the project in the scheduled time.

We would like to express our gratitude and sincere thanks to our project guide **Mrs.T.Kalpana MCA(Phd).,** Assistant Professor(Sr.G) Department of Computer Applications, Kongu Engineering College for giving her valuable guidance and suggestions which helped us in the successful completion of the project.

We owe a great deal of gratitude to our parents for helping overwhelm in all proceedings. We bow our hearts and head with heartfelt thanks to all those who thought us their warm services to succeed and achieve our work.

TABLE OF CONTENTS

ZZ

LIST OF FIGURES

ZZ

LIST OF ABBREVATION

ABBREVATION	EXPANSION
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	Java Script
PHP	Hypertext Preprocessor
MySql	My Structured Query Language

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

1.1.1 Overview

The Web-Based Orphanage Management System is a platform designed to connect donors with orphanages, making it easier to donate food, clothing, and money. The system allows orphanages to list their needs and manage donations efficiently. Donors can register, contribute to specific requirements, and track their contributions online.

The platform ensures transparency by allowing donors to view how their donations are used and provides receipts for every contribution. It also simplifies resource management for orphanages by keeping records of inventory and financial support. With a secure payment gateway, real-time donation tracking, and an easy-to-use interface, this system promotes community involvement and helps improve the lives of orphaned children.

1.1.2 Background

The Web-Based Orphanage Management System aims to connect donors with orphanages to simplify the process of contributing food, clothing, and money. Orphanages can list their needs, manage donations, and keep track of resources efficiently. Donors can register, view requirements, donate online, and track their contributions with transparency. The system promotes trust, accountability, and ease of access, encouraging more community involvement. By bridging the gap between donors and orphanages, it helps improve the quality of life for orphaned children.

1.1.3 Problem Description

The traditional methods of managing orphanages often rely on manual processes, which can lead to inefficiencies, lack of transparency, and limited accessibility for donors. Orphanages face challenges in tracking and utilizing donations effectively, maintaining accurate records of children's needs, and providing updates to donors. Similarly, donors may be reluctant to contribute due to the absence of real-time updates and clear insights into how their contributions are being used. This disconnect creates a barrier between donors and orphanages, hindering their ability to collaborate and support children in need.

The lack of a centralized system makes it difficult for orphanages to monitor inventory levels of essential resources like food and clothing, leading to either shortages or wastage. Additionally, the process of managing monetary donations is often prone to errors and lacks accountability. For donors, there is no convenient way to contribute to orphanages or track the impact of their donations. This results in decreased trust and a reduction in donor engagement, ultimately impacting the resources available to support orphaned children.

In the absence of a well-structured platform, orphanages struggle to communicate their needs effectively to potential donors. This problem is further exacerbated by the growing need for modern, scalable solutions that can handle the increasing volume of data and transactions as orphanage operations expand.

The Web-Based Orphanage Management System addresses these challenges by providing a centralized platform that streamlines the management of resources and donations, fosters transparency, and bridges the gap between orphanages and donors. By leveraging technology, the system eliminates inefficiencies, enhances accountability, and promotes a collaborative environment to improve the lives of orphaned children.

1.2 EXISTING SYSTEM

An existing system for a web-based orphanage project* typically involves user roles like Admin, Staff, Donors, and Visitors. It includes modules for child management, donor management, sponsorship and adoption processes, donation handling, event organization, and reporting.

The system stores child details such as name, age, health, and education. It manages donor records, processes online donations, tracks sponsorships, and supports adoption applications. Donations are categorized, and receipts are issued automatically. Events like fundraisers are showcased, and donors can contribute. Notifications and reminders are sent via email or SMS.

It generates reports for donations, sponsorships, and child growth, exporting them as PDFs or Excel. Role-based dashboards, secure login, and HTTPS ensure data security. Frontend technologies like React, with backend support from Django or Node.js, and databases like MySQL or MongoDB, form the technical backbone. Payment integrations like Stripe or PayPal facilitate transactions.

1.2.1 DRAWBACKS OF EXISTING SYSTEM

The existing system for managing orphanage donations has several drawbacks:

- Limited Reach
- Lack of Transparency
- Inefficient Resource Management
- Time-Consuming Process
- No Real-Time Updates.
- Security Concerns
- Minimal Engagement

1.3 PROPOSED SYSTEM

The proposed Web-Based Orphanage Management System provides a platform to simplify and streamline the donation process. Donors can view orphanage needs, donate food, clothing, or money, and track their contributions online. The system ensures transparency with real-time updates and automated receipts, building trust with donors. Orphanages can manage inventory, post specific requirements, and efficiently allocate resources. With secure payments and a user-friendly interface, the system connects donors and orphanages effectively, promoting community involvement and improving support for orphaned children.

1.3.1 ADVANTAGES OF PROPOSED SYSTEM

The proposed system offers several advantages:

- Ease of Use
- Transparency
- Efficient Resource Management
- Secure Transactions
- Wider Reach
- Time-Saving
- Community Engagement

SUMMARY

This chapter describes about the existing system and its drawbacks and proposed system and its advantages. The purpose of this project was to develop a web application for purchasing items from a shop. This project helped us in gaining valuable Information and practical knowledge on several topics like designing web pages using React JS, usage of responsive templates, management of databases using Firebase. The system analysis and software requirements and hardware requirements of the project has been explained in Chapter 2.

CHAPTER 2

SYSTEM ANALYSIS

2.1 IDENTIFICATION OF NEED

The need for a Web-Based Orphanage Management System arises from the challenges faced by orphanages in managing donations and resources efficiently. Traditional methods of seeking and managing contributions are time-consuming, lack transparency, and often result in mismanagement of resources. Donors, on the other hand, struggle with understanding specific needs and ensuring their contributions are used effectively. A centralized, digital platform is essential to bridge this gap, providing a streamlined process for connecting donors with orphanages, ensuring efficient resource management, and fostering trust and accountability. This system fulfills the need for a modern, scalable solution to support orphaned children and enhance community participation.

2.2 FEASIBILITY STUDY

The feasibility of the Web-Based Orphanage Management System is high due to the availability of reliable and scalable technologies like mysql . The use of open-source tools and affordable cloud hosting minimizes costs, making it economically viable. The system's user-friendly design ensures it can be easily operated by both donors and orphanage staff, requiring minimal training. Legal compliance is achievable through secure data handling and adherence to data protection regulations, while social acceptance is high as the system fosters transparency, accountability, and trust in the donation process . Overall, the project is feasible in all aspects and promises significant social impact.

2.2.1 Economic Feasibility

The Web-Based Orphanage Management System is an economically feasible solution leveraging open-source technologies like PHP and MySQL, minimizing development costs by eliminating licensing fees. The platform requires minimal hardware and utilizes affordable cloud hosting services, ensuring low operational expenses. Maintenance costs are kept low through widely available support for PHP-based systems and efficient data storage. By automating manual processes, the system reduces administrative expenses and improves resource management. Transparent donation tracking fosters trust, encouraging recurring contributions and increasing overall donations. The scalability of the platform allows it to grow without significant additional investment, ensuring sustainability. This cost-effective solution maximizes the orphanage's resources, improving the welfare of children.

2.2.2 Operational Feasibility

The Web-Based Orphanage Management System is operationally feasible, streamlining workflows for admins, trusts, and donors with its user-friendly interface. Admins can efficiently manage records and donations, while donors can easily register and track their contributions in real-time. The web-based design ensures accessibility from any device, and automated processes enhance efficiency and reduce errors. Its scalable architecture accommodates growing user numbers, making it a practical and sustainable solution.

2.2.3 Technical Feasibility

The Web-Based Orphanage Management System uses PHP and MySQL, both open-source technologies, ensuring cost-effectiveness and broad support. It is web-based, compatible with all major browsers, and requires no specialized hardware. Security features include role-based access control and encrypted transactions. The system is easily deployable on affordable servers or cloud platforms and supports scalability for future enhancements. Overall, it offers a robust and sustainable solution for managing orphanage activities.

2.3 SOFTWARE REQUIREMENT SPECIFICATIONS

A software requirements specification is a complete description of the behavior of the software to be developed. It contains the functional, non-functional and user interface requirements of the proposed system.

Functional Requirements

The Web-Based Orphanage Management System allows donors to register, log in, browse orphanage needs, and make donations in the form of food, clothing, or money. Donors can track their contributions, receive automated receipts, and schedule physical donations. Orphanages can create profiles, post specific needs, manage donations, and track inventory. The system includes secure payment gateway integration for monetary contributions and sends email notifications to donors and orphanages about donation status. Administrators oversee user accounts, monitor activities, and generate reports. The platform provides real-time updates on donation statuses, ensures mobile responsiveness, and supports user authentication for secure access.

System Feature 1

Authentication

- The trust can login by entering their correct email and password.
- If email and password are not correct it doesn’t open
- Turst can logout once they completed their process.
- If they don’t have a username and password, they can create their username and password. With the username and password, they can login.
- While creating the account, if the username exist it will throw an error message.
- Username check-After logging in, the username will be checked from the database and the user interface will be display accordingly.

Process Data

- Display- the login page will show up.
- View- The trust registered username can view the homepage and the available products information.

System Feature 2

Process Data

- View – Admin can view all the trust and donars information.
- Add – Admin can approve both trust and donars.
- Manage – Admin can manage the trust and donars details.

Software Requirements

Operating System	:	Windows / Linux/Mac
Language	:	PHP
Backend	:	PHP
Database	:	MYSQL

Hardware Requirements

Processer	:	Intel Core i5
RAM	:	2GB DDR4
Monitor	:	15’’VGA Monitor
Hard Disk	:	250GB
Keyboard	:	ACCUTYPE keyboard Mouse Optical Mouse

2.4 SOFTWARE DESCRIPTION

PHP

PHP (Hypertext Preprocessor) is a popular open-source server-side scripting language widely used for web development. It is designed to create dynamic and interactive web pages. PHP executes on the server, generating HTML that is sent to the client’s browser. It supports integration with databases like MySQL, PostgreSQL, and SQLite, making it ideal for dynamic and data-driven applications. PHP is platform-independent, easy to learn, and offers built-in security features to protect against common threats. Its extensive library support and ability to integrate seamlessly with HTML, CSS, and JavaScript make it a powerful tool for developing websites, content management systems, and e-commerce platforms. With its scalability and active community, PHP remains a reliable choice for modern web development.

Use of PHP

PHP is widely used for creating dynamic and interactive web applications. It is ideal for server- side scripting, where it generates web pages and content dynamically. PHP is used to develop websites, manage databases, process forms, handle user authentication, and build content management systems like WordPress. It is also used for creating e-commerce platforms, integrating payment gateways, and developing APIs. With its ability to connect to various databases, PHP is a powerful tool for building data-driven applications. Its ease of integration with HTML, CSS, and JavaScript makes it versatile for web development.

Working methods of PHP

PHP works as a server-side scripting language, processing code on the server and sending the resulting output (usually HTML) to the client’s browser. Here's how it operates:

1. **Request**: A client (browser) sends a request to the server for a PHP file by accessing a URL.
2. **Processing**: The web server (e.g., Apache or Nginx) receives the request and passes the PHP script to the PHP interpreter.
3. **Execution**: The PHP interpreter executes the code in the PHP file. It may include database interactions, form processing, or dynamic content generation.
4. **Output Generation**: After execution, PHP generates an output, typically in HTML format, which the server sends back to the client.
5. **Response**: The browser receives the output and renders the web page for the user to view.

PHP handles server-side logic, ensuring only the processed results (not the code) are visible to the client, making it secure and efficient for dynamic web applications.

Features of php

PHP offers a variety of features that make it a powerful tool for web development:

- **Open Source**: PHP is free to use, with an extensive community for support and resources.
- **Server-Side Execution**: PHP runs on the server, generating dynamic content for web pages.
- **Cross-Platform Compatibility**: PHP works seamlessly on multiple platforms, including Windows, Linux, and macOS.
- **Database Integration**: It supports a wide range of databases, such as MySQL, PostgreSQL, and SQLite.
- **Ease of Integration**: PHP can be easily embedded into HTML and integrates well with CSS and JavaScript.
- **Secure**: PHP provides built-in features to prevent security threats like SQL injection and XSS attacks.
- **Fast Performance**: PHP scripts execute quickly, even for dynamic and large-scale web applications.
- **Scalable**: It supports applications of any size, from small websites to complex enterprise systems.
- **Extensive Library Support**: PHP offers a wide range of libraries and frameworks, like Laravel and CodeIgniter, to speed up development.
- **Flexible and Dynamic**: PHP allows developers to create dynamic, interactive, and responsive web applications efficiently.

HTML,CSS, JAVASCRIPT

HTML (HyperText Markup Language):

HTML is the foundation of web development, used to structure the content on a web page. It defines elements like headings, paragraphs, images, links, and forms. HTML uses tags to create and organize content, forming the basic layout of a website.

CSS (Cascading Style Sheets):

CSS is used to style the HTML content by defining the visual presentation of a web page. It controls layout, colors, fonts, spacing, and responsiveness. CSS enables the creation of aesthetically pleasing and mobile-friendly designs, ensuring a consistent look across different devices.

JavaScript:

JavaScript is a scripting language that adds interactivity and dynamic functionality to web pages. It allows developers to implement features like form validation, animations, dynamic content updates, and real-time interactions. JavaScript works on the client side, enhancing the user experience by enabling seamless interactions without reloading the page.

Together, HTML, CSS, and JavaScript form the core technologies for building modern, interactive, and visually appealing web applications.

Features of HTML,CSS, JAVASCRIPT

Features of HTML

1. **Structure**: the basic structure for web pages using tags like <h1>, <p>, <img>, etc.
2. **Hyperlinking**: Enables linking between web pages using <a> tags.
3. **Media Integration**: Supports embedding images, videos, and audio files.
4. **Forms**: Facilitates user input through forms with elements like <input>, <textarea>, and <button>.
5. **Semantic Tags**: Includes tags like <header>, <footer>, and <article> to enhance readability and SEO.

Features of CSS

1. **Styling**: Defines the look and feel of web pages, including colors, fonts, and layout.
2. **Selectors**: Allows targeting specific elements for styling using classes, IDs, or tag names.
3. **Responsive Design**: Support media queries to create design for different screen sizes.
4. **Animations**: Enables animations and transitions for interactive effects.
5. **External Stylesheets**: Allows reusability and separation of style from content for efficient management.

Features of JavaScript

1. **Interactivity**: Adds dynamic features like form validation, sliders, and real-time updates.
2. **Client-Side Execution**: Runs directly in the user’s browser, reducing server load.
3. **DOM Manipulation**: Allows real-time updates to HTML and CSS elements without refreshing the page.
4. **Event Handling**: Responds to user actions like clicks, hovers, and keypresses.
5. **Cross-Browser Compatibility**: Works across all modern web browsers for consistent functionality.
6. **Integration**: Works seamlessly with HTML and CSS for a complete web development experience.

MYSQL

MySQL is a widely used open-source relational database management system (RDBMS). It is designed for managing and organizing data in a structured format using SQL (Structured Query Language). MySQL is known for its speed, reliability, and scalability, making it an ideal choice for applications ranging from small-scale websites to large enterprise systems.

Use of MYSQL

- Managing structured data in web applications.
- Powering content management systems (CMS) like WordPress, Joomla, and Drupal.
- Running e-commerce platforms, including inventory, orders, and customer data.
- Backend support for dynamic websites and web applications.
- Data warehousing and business intelligence systems.

MySQL is a dependable and flexible database system, widely adopted for its ease of use, robust features, and compatibility with modern technologies.

Features of MYSQL

- **Open Source**: MySQL is free to use and has an active community providing support and updates.
- **Relational Database**: Stores data in tables, allowing efficient relationships between data using primary and foreign keys.
- **Scalable**: Handles databases of any size, from small applications to large-scale enterprise systems.
- **High Performance**: Optimized for speed, ensuring quick execution of complex queries.
- **Cross-Platform**: Works on multiple operating systems, including Windows, Linux, and macOS.
- **Security**: Provides robust security features, including user authentication and data encryption.
- **ACID Compliance**: Ensures reliability, consistency, and data integrity in transactions.

- **Support for Multiple Storage Engines**: Offers flexibility with various storage engines like InnoDB and MyISAM for different use cases.
- **Integration**: Works seamlessly with popular programming languages like PHP, Python, and Java for web and application development.
- **Backup and Recovery**: Supports tools for data backup and disaster recovery, ensuring minimal downtime.

SUMMARY

This chapter the feasibility study. Before, the work is done manually, which takes more time and effort, and that all of these issues can be resolved by using this application, which offers a number of benefits such as time savings, reduced manual work, and so on. The system's economic, operational, and technical feasibility studies are also discussed. the hardware requirements, software requirements and technical requirements such as the fact that the front end and back end are done in this process. In Technical the performance, security, availability and reliability are also discussed. The system design, modules and data models for this project are described in Chapter 3.

CHAPTER 3

SYSTEM DESIGN

3.1 INPUT DESIGN

The input design for the Web-Based Orphanage Management System focuses on creating user-friendly, efficient, and secure data entry forms. Donors can register by providing their name, email, password, and mobile number, with validations for email format and password strength. The donation form allows donors to select donation type, specify quantity or amount, and choose an orphanage from a dropdown menu, ensuring accurate data entry with real-time validation. Orphanages can register by entering their name, location, contact details, and email, with checks for required fields and unique entries. They can also post specific needs by specifying item type, description, and quantity required, with validation for non-negative values. Admins manage users by entering names, roles, and contact details, ensuring accurate data through validations. Forms are designed with clear labels, placeholders, and error messages, offering real-time feedback and secure input for sensitive data like passwords and payments. The system is mobile-responsive, accessible, and optimized for error-free data collection.

OBJECTIVES

The objectives of the Web-Based Orphanage Management System are to create a platform that connects donors with orphanages, making it easy to contribute food, clothing, and money. It aims to ensure transparency by providing real-time updates and tracking of donations. The system streamlines resource management for orphanages, organizing inventory and records efficiently. It encourages community involvement by fostering long-term donor relationships and recurring contributions. By automating manual tasks like record-keeping and reporting, the system enhances operational efficiency. It provides a secure and user-friendly interface for all users and ensures donations are used effectively to improve the lives of orphaned children.

3.2 OUTPUT DESIGN

The output design for the Web-Based Orphanage Management System focuses on presenting clear, accurate, and user-friendly information to donors, orphanages, and administrators. Outputs are designed to ensure transparency, improve decision-making, and provide necessary feedback.

For donors, the system displays confirmation messages after successful donations, detailed donation history, and real-time updates on the status of their contributions. For orphanages, it generates inventory reports, lists of received and pending donations, and alerts for low stock or urgent needs. Admins receive system-wide activity summaries, user reports, and analytics to monitor donations and resource utilization. All outputs are visually structured for clarity, mobile-friendly, and include options for exporting data, such as reports in PDF format, to enhance usability.

3.3 MODULE DESIGN

A module description provides detailed information about the module and its supported components, which is accessible in different manners. In this application, it contains two main modules with many sub-modules.

ADMIN MODULE

This module will be accessible and handled by the admin that is admin can add or delete trust and doners details.

TRUST MODULE

Registration & Login

New users can register in this site and can login the system. If the login credentials are correct, then the user is allowed to login the site.

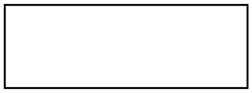
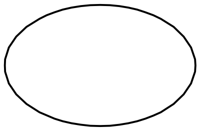
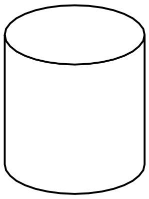
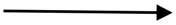
The new user is asked for the user name, password, email, mobile no, and address while registering the account in the website.

DONER MODULE

This module will be used by the donar in which the donar make their donaition on products which they need and make trust. When user need to purchase any product they can add that product into the cart and they can place their order.

3.4 DATA MODELS

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by thesystem.
	A **data store**, a place where data is held between processes.
	A **data flow.**

LEVEL 0

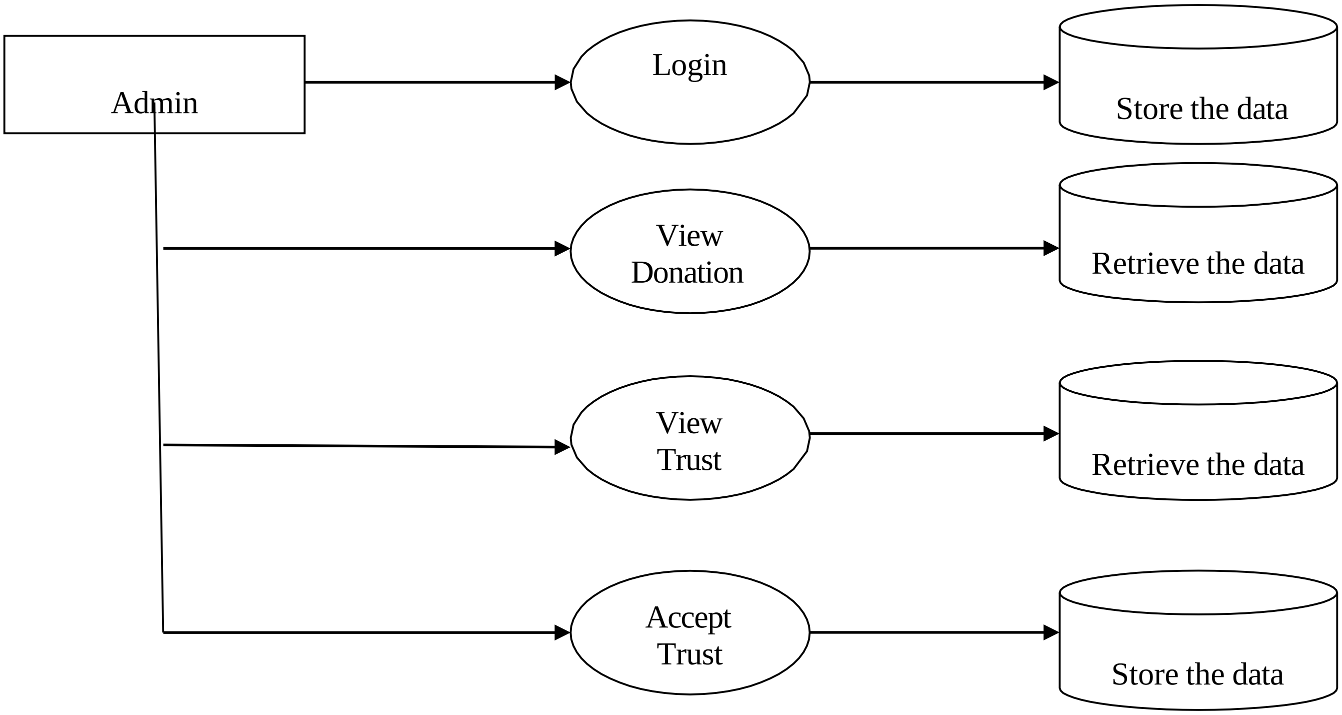


Figure:3.4.1 DATAFLOW DIAGRAM FOR WEB ORPHANAGE SYSTEM

LEVEL 1

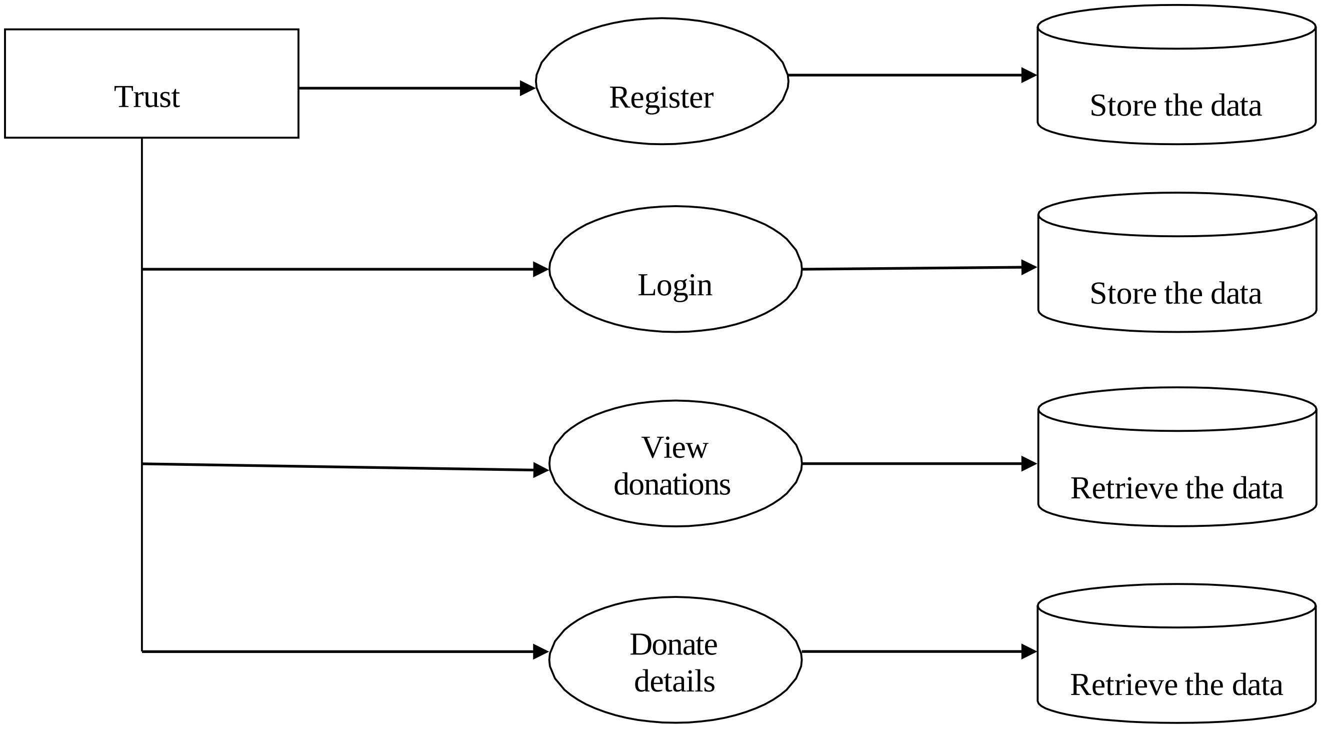


Figure:3.4.2 DATAFLOW DIAGRAM FOR WEB ORPHANAGE SYSTEM

DESCRIPTION

The image illustrates a data flow diagram, likely for a system related to trust and donations.

Components:

Trust: The starting point of the diagram, representing the entity or system initiating the process.

Register: The first action, likely for a user to register within the system.

Login: The second action, allowing a registered user to access the system.

View Donations: An action to view existing donations.

Donate Details: An action to provide details for a new donation.

Store the Data: Cylinders representing data storage, used for storing registration and login information.

Retrieve the Data: Cylinders representing data retrieval, used for fetching donation details.

Flow:

The flow starts with the "Trust" entity and progresses through "Register" and "Login," leading to data storage.

The user can then choose to "View Donations" or "Donate Details," both leading to data retrieval.

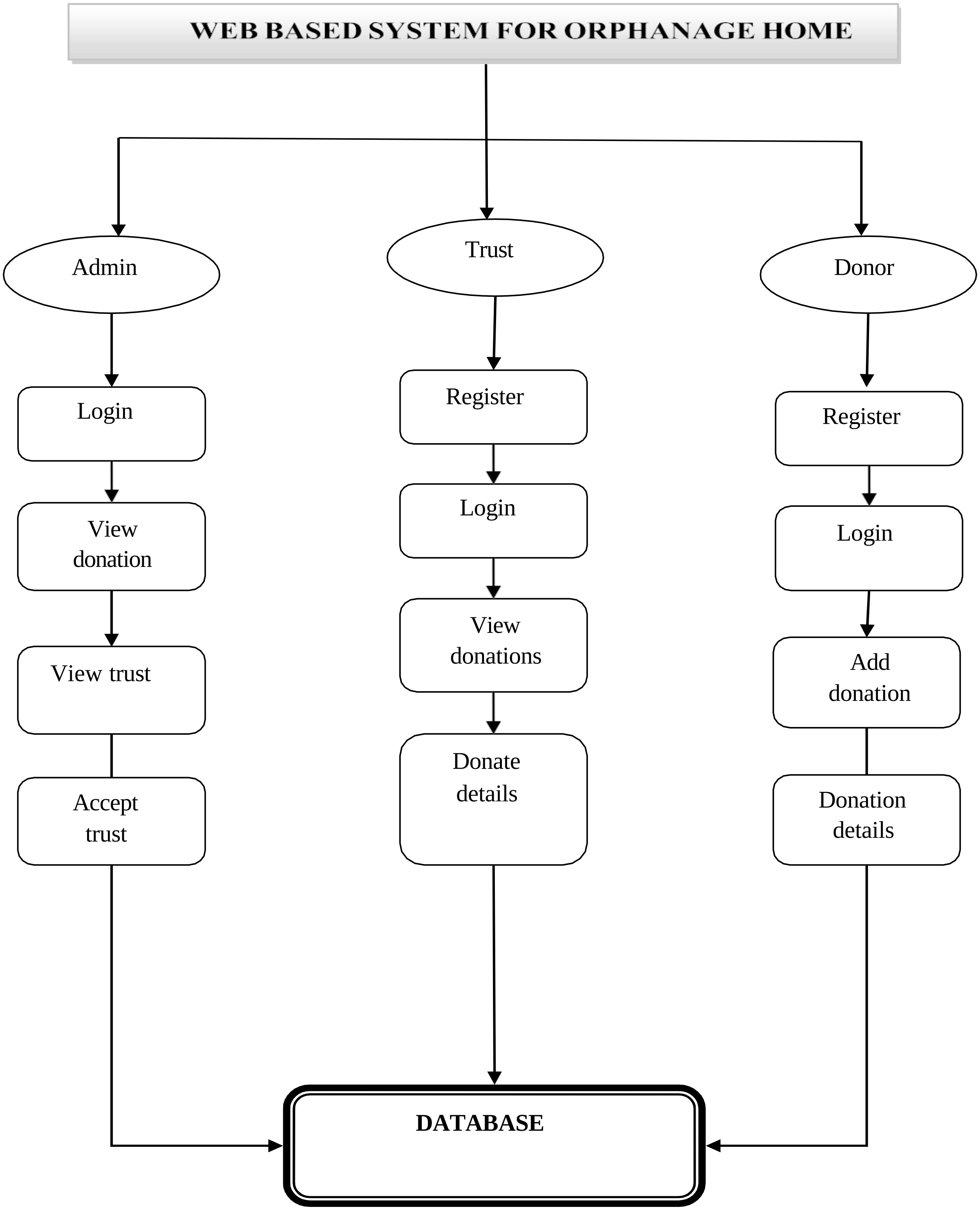


Figure:3.4.3 WEB BASED SYSTEM FOR ORPHANAGE HOME

DESCRIPTION

Key elements of the flowchart:

User Roles:

- Admin: The administrator manages the system, including accepting trust requests and viewing donation details.
- Trust: Represents the orphanage or trusted entity that receives donations. They can view donation details and accept donations.
- Donor: Individuals or organizations who make donations. They can register, log in, add donations, and view donation details.

Actions:

- Each user role has specific actions they can perform, such as:
- Register: New users can register to create an account.
- Login: Existing users can log in to access their features.
- View: Users can view relevant information like donation details or trust requests.
- Accept: Admin can accept trust requests, and Trust can accept donations.
- Add: Donors can add new donations to the system.

Central Database:

All user actions and data are stored in a central database, ensuring data integrity and consistency.

ER Diagram

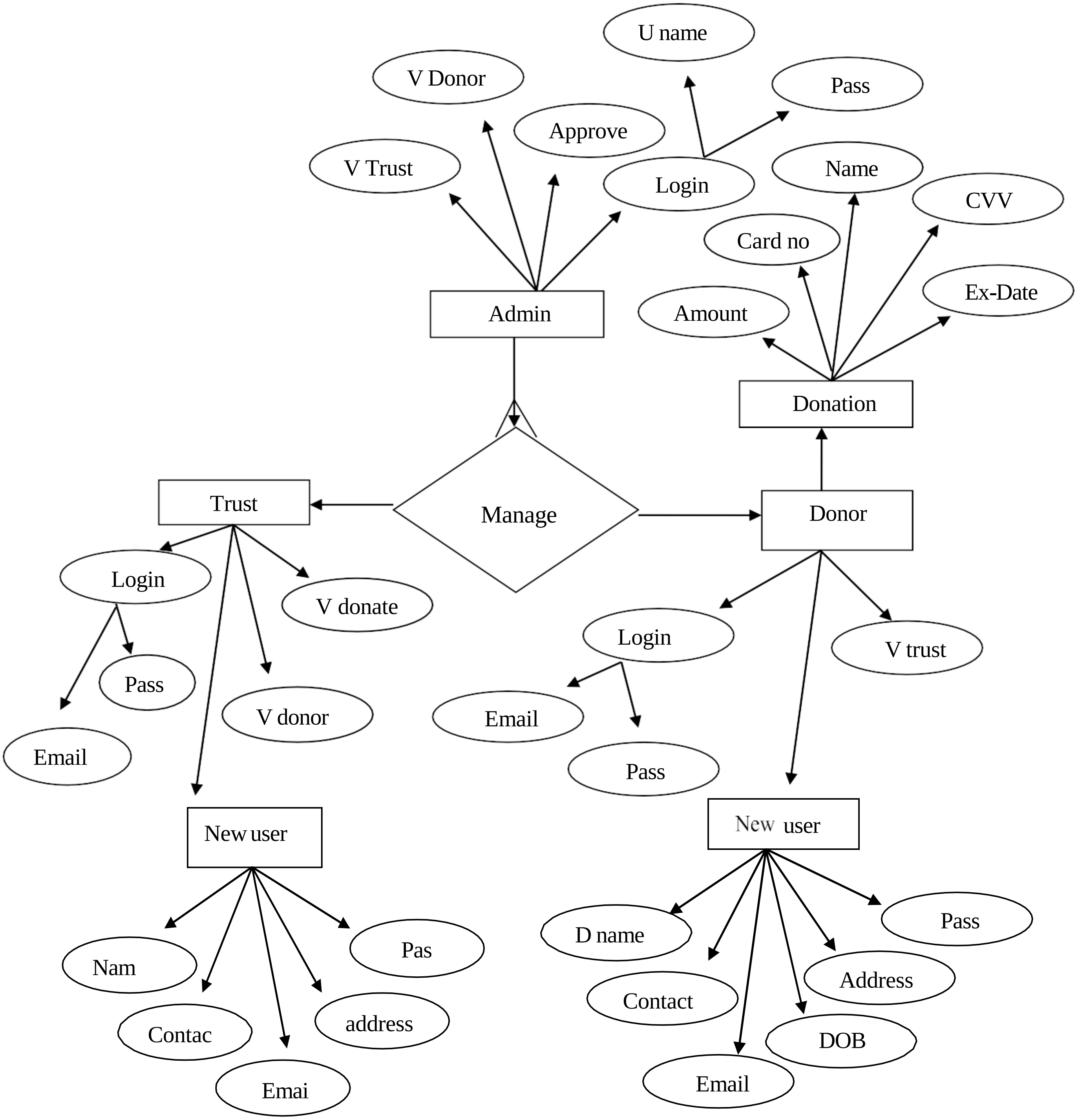


Figure:3.4.4 ENTITY RELATIONSHIP DIAGRAM WEB BASED ORPHANAGE SYSTEM

DESCRIPTION

The image shows an Entity-Relationship Diagram (ERD) for a donation management system. Here's a breakdown of the key components:

Entities:

* Trust: Represents a charitable organization or trust.

* Admin: Represents the administrator who manages the system.

* Donor: Represents individuals making donations.

* Donation: Represents the act of donating, including details like amount and date.

Relationships:

* Manage: Connects the Admin entity to the Donation entity, indicating that the admin manages donations.

* Donate: Connects the Donor entity to the Donation entity, representing the act of making a donation.

* Login: Connects various entities (Trust, Admin, Donor) to their respective login credentials (email, password).

* New User: Represents the process of creating a new user account, either for a Trust or a Donor.

Attributes:

* Each entity has several attributes listed within the oval shapes, such as:

- Trust: Name, Address, Email
- Admin: U-name (Username), Pass (Password)
- Donor: D-name (Donor Name), Address, Contact, DOB (Date of Birth)
- Donation: Amount, Card No, CVV, Exp Date .

SUMMARY

This Chapter describes the Input design, Output design, Module Description, such as Admin module and User module and data model. In the Admin module, the process is add products and delivery. User module, add details and login. The Data flow diagram, class diagram. database design, data integrity process also discussed in this chapter. The system implementation for this project are described in Chapter 4.

CHAPTER 4

IMPLEMENTATION

4.1 SYSTEM IMPLEMENTATION

The Web-Based Orphanage Management System involves setting up a web server (Apache or Nginx) and a MySQL database, using PHP for the backend and HTML, CSS, and JavaScript for the frontend. Donors can register, browse needs, make secure donations, and track contributions, while orphanages can post needs, manage inventory, and generate reports. Admins manage user accounts and monitor activities. The system includes email notifications and is deployed on a compatible hosting platform, with ongoing maintenance for performance and security. This approach facilitates a reliable connection between donors and orphanages.

4.1 STANDARDIZATION OF CODING

The Web-Based Orphanage Management System's coding standards focus on clean, consistent, and maintainable code. Variables, functions, and files follow consistent naming conventions like camelCase for variables and functions and lowercase with underscores for file names. Proper indentation, typically four spaces per level, ensures readability. Logic is separated from presentation by using templates or the MVC architecture, and external stylesheets and scripts are utilized. Database queries use prepared statements or stored procedures to prevent SQL injection, with tables and columns named in lowercase with underscores. Error handling involves meaningful error messages and try-catch blocks. Security standards include input sanitization, password hashing with bcrypt, and HTTPS for secure data transmission.

4.2 ERROR HANDLING

Error handling in the Web-Based Orphanage Management System ensures smooth functionality and meaningful feedback to users while maintaining security. All user inputs are validated on both the client and server-side using PHP to prevent invalid or harmful data. Runtime errors, like database connection failures or file operation issues, are managed using try-catch blocks, with user-friendly messages displayed. Detailed error information is logged on the server for debugging, while users see general error messages to protect sensitive information. Custom error pages handle HTTP errors like 404 and 500, providing a consistent user experience. Critical failures have fallback mechanisms, such as retrying database connections. For database queries, return statuses are checked to ensure meaningful messages like "Unable to process your request at this time" shown. PHP’s error reporting and display settings are configured to control visibility in production and development environments. All exceptions, especially for sensitive operations like payments, are caught and handled to ensure system security and reliability.

SUMMARY

In this chapter, Systems implementation is the process of defining how the information system should be built ensuring that the information system is operational and used, ensuring that the information system meets quality standard. we had gained knowledge on system implementation, and how to write a standardized code with command lines that a developer can understand and code readability and whenever exception occurs error can be handled. Testing and results for the project is described in chapter 5.

CHAPTER 5

TESTING AND RESULTS

5.1 SYSTEM TESTING

System testing for the Web-Based Orphanage Management System ensures that all components function correctly and meet user requirements. Functional testing verifies that all features, such as donor registration, donation tracking, and inventory management, work as intended. Integration testing checks that modules like donor contributions, orphanage inventory updates, and admin monitoring interact seamlessly. User acceptance testing involves real users validating the system’s usability and functionality. Security testing ensures that sensitive data, such as payment information and user credentials, is protected from vulnerabilities like SQL injection and cross-site scripting. Performance testing evaluates the system’s ability to handle concurrent users and database queries efficiently. Cross-browser and mobile responsiveness testing confirm that the application is accessible and functional across different devices and browsers. Finally, error-handling testing ensures the system provides meaningful feedback and gracefully manages unexpected issues.

5.2 FUNCTIONAL TESTING

Functional testing ensures the Web-Based Orphanage Management System works as intended for admins, trusts, and donors. Admins are tested for managing records, donations, and inventory, while trusts are validated for resource management and donor communication. Donors are tested for seamless registration, donation processes, and accessing their contribution history. System-wide tests ensure authentication, data accuracy, security, and scalability for smooth and reliable operations.

5.3 SECURITY TESTING

Security testing for the Web-Based Orphanage Management System is essential to protect sensitive user data and ensure application integrity. It begins with implementing role-based access control (RBAC) to restrict users to authorized functionalities, minimizing the risk of unauthorized actions. Sensitive data, such as login credentials and transactions, must be encrypted both in transit and at rest using secure protocols and strong algorithms. Session management is crucial, requiring automatic logout after inactivity to prevent unauthorized access. The system must also be tested against common threats like SQL injection and cross-site scripting (XSS) by using prepared statements and input validation. Regular security assessments, including penetration testing, help identify and address vulnerabilities. By prioritizing these security measures, the system can create a safe environment for users and foster trust among donors and orphanages.

5.4 VALIDATION TESTING

Validation testing is essential for the Web-Based Orphanage Management System to ensure accurate processing of user inputs and compliance with requirements. It involves checking input fields for correct data validation, such as verifying mandatory fields, valid email formats, and non-negative numerical values. Forms for registration, login, and donations are tested to ensure they handle invalid data properly, providing clear error messages to users. Additionally, interactive elements like dropdowns, checkboxes, and file uploads are validated for correct functionality. The system is designed to produce expected outputs for valid inputs while gracefully rejecting invalid data, enhancing user experience and maintaining data integrity. Overall, thorough validation testing contributes to a reliable and user-friendly application.

5.4 RESULT

The Web-Based Orphanage Management System successfully bridges the gap between donors and orphanages, providing an efficient, secure, and user-friendly platform for managing donations. The system enables donors to easily register, browse needs, and contribute food, clothing, or money, with real-time tracking of their contributions. Orphanages can post requirements, manage inventory, and generate reports, ensuring optimal resource utilization.

The system promotes transparency and trust through automated notifications, detailed donation histories, and secure payment processing. It also simplifies administrative tasks for orphanages and enhances community involvement by encouraging recurring contributions. Comprehensive testing ensures that the system is reliable, responsive, and secure, meeting all functional and non-functional requirements. Overall, the project achieves its goal of improving resource management, fostering trust, and enhancing the living conditions of orphaned children.

SUMMARY

In this chapter, it shows the system testing and its results. System testing is the testing of a system as a whole. End to end testing is performed to verify that all the scenarios are working as expected. In result, The challenges experienced have been thoroughly outlined in this chapter, and all of those challenges have been overcome by achieving this online shopping for flowers. The conclusion and future work for the project is described in Chapter 6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The Web-Based Orphanage Management System is a comprehensive solution that addresses the challenges faced by orphanages in managing donations and resources. By providing a centralized platform, the system simplifies the donation process for donors, enhances transparency, and ensures efficient resource management for orphanages. It bridges the gap between donors and orphanages, fostering trust and encouraging community involvement.

With features like real-time updates, secure payment processing, inventory management, and automated notifications, the system streamlines operations while promoting accountability. Thorough testing ensures the system is user-friendly, reliable, and secure. This project not only improves the efficiency of donation management but also contributes significantly to the welfare of orphaned children by enabling better utilization of resources and fostering a culture of giving.

6.2 FUTURE ENHANCEMENT

- **Mobile Application**: Develop a mobile app for easy access and on-the-go donations.
- **Recurring Donations**: Introduce options for donors to set up automated monthly or yearly donations.
- **Donation Pickup Scheduling**: Allow donors to schedule pickups for physical donations like food or clothing.
- **AI-Based Recommendations**: Use AI to recommend donation opportunities based on donor preferences and orphanage needs.
- **Feedback System**: Add a feedback mechanism for donors and orphanages to improve the platform.

APPENDICES

A. SAMPLE CODING

```
<?php
include("include\dbconnect.php");
extract($_POST);
session_start();
if(isset($btn))
{
 $name=$_REQUEST['username'];

$password=$_REQUEST['password'];
 $_SESSION['un']=$email;
 $qry="select * from admin_details where username='$name' && password='$password'";
$result = mysqli_query($conn, $qry);
 if (mysqli_num_rows($result))
  {
  ?>
<script language="javascript" type="text/javascript">
alert("Login Successfully");
window.location.href="admin_home.php";
</script>
<?php
 }
 else
{

?>
<script language="javascript" type="text/javascript">
alert("Username /  Password Incorrect");

</script>
<?php
}
$conn->close();
}
?>
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta  charset="utf-8">
  <meta  name="viewport"  content="width=device-width,  initial-scale=1,  shrink-to-fit=no">
  <title></title>
  <!--/google-fonts  -->
  <link
   href="//fonts.googleapis.com/css2?family=Hind+Siliguri:wght@300;400;500;600;700&displ
   ay=swap" rel="stylesheet">
  <!--//google-fonts  -->
  <!-- Template CSS -->
  <link  rel="stylesheet"  href="assets/css/style-starter.css">
</head>

<body>
  <!--/Header-->
  <header  id="site-header"  class="fixed-top">
    <div class="container">
      <nav class="navbar navbar-expand-lg navbar-light stroke py-lg-0">
        <h1><a class="navbar-brand pe-xl-5 pe-lg-4" href="#">
              Orphanage<span class="sublog">Home</span>
          </a></h1>
        <button class="navbar-toggler collapsed" type="button" data-bs-toggle="collapse"
   data-bs-target="#navbarScroll" aria-controls="navbarScroll" aria-expanded="false" aria-
   label="Toggle navigation">
          <span class="navbar-toggler-icon fa icon-expand fa-bars"></span>
          <span class="navbar-toggler-icon fa icon-close fa-times"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <ul class="navbar-nav me-lg-auto my-2 my-lg-0 navbar-nav-scroll">
            <li class="nav-item">
              <a class="nav-link " aria-current="page"
   href="index.php">Home</a>                    </li>
```

```
                    <li class="nav-item">
                        <a class="nav-link active" href="admin.php">Admin</a></li>
                    <li class="nav-item">
                        <a class="nav-link" href="trust.php">Trust</a></li>

                    <li class="nav-item">
                        <a class="nav-link" href="donor.php">Donor</a></li>

                </ul>



                <!-- //toggle switch for light and dark theme -->
                <!-- search popup -->

                <!-- /search popup -->
            </div>
            <!-- toggle switch for light and dark theme -->
            <div class="mobile-position">
                <nav class="navigation">
                    <div class="theme-switch-wrapper">
                        <label class="theme-switch" for="checkbox">
                            <input type="checkbox" id="checkbox">
                            <div class="mode-container">
                                <i class="gg-sun"></i>
                                <i class="gg-moon"></i>
                            </div>
                        </label>
                    </div>
                </nav>
            </div>
            <!-- //toggle switch for light and dark theme -->
        </nav>
    </div>
</header>
<!--//Header-->
<!-- breadcrumb -->
<section class="w3l-about-breadcrumb">
    <div class="breadcrumb-bg breadcrumb-bg-about py-5">
    <div class="container py-lg-5 py-md-4">
      <div  class="w3breadcrumb-gids  text-center  pt-5">
        <div   class="w3breadcrumb-right">
          <ul   class="breadcrumbs-custom-path">
            <li><a href="#">Home</a></li>
            <li class="active"><span class="fas fa-angle-double-right mx-2"></span>
 Admin </li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</section>
<!--//breadcrumb-->

 <section  class="w3l-contact-1  py-5"  id="contact">
  <div  class="contacts-9  py-lg-5  py-md-4">
    <div class="container">
      <div class="contactct-fm map-content-9">
        <div class="header-title text-center">

          <h3 class="title-w3l mb-2">ADMIN
            <span class="span-bold"> LOGIN</span>
          </h3>

        </div>


        <form action="" class="pt-lg-4" method="post" enctype="multipart/form-data">
         <table height="270" align="center">
          <tr>
           <td width="75">UserName</td>
           <td width="311"><input type="text" name="username"
 placeholder="Username" required=""></td>
          </tr>
          <tr>
           <td>Password</td>
           <td><input type="password" name="password" placeholder="Password"
 required=""></td>
          </tr>
          <tr>
            <td>&nbsp;</td>
            <td><span class="tp">
              <button type="submit" name="btn" class="btn btn-primary btn-style mt-lg-5
mt-4">Login</button>
            </span></td>
          </tr>
        </table>



      </form>
    </div>


  </div>
 </div>
</section>
<!--//w3l-project-->
<!--/w3l-footer-29-main-->
<footer   class="w3l-footer-29-main">
 <div  class="footer-29  py-5">
  <div class="container py-lg-4">
   <div class="bottom-copies text-center">
      <p class="copy-footer-29">All rights reserved. Design by
       <a href="#"> Admin</a>
      </p>
   </div>
  </div>
 </div>

 <!-- move top -->
 <button onClick="topFunction()" id="movetop" title="Go to top">
  <span class="fa fa-angle-up"></span>
 </button>
 <script>
  // When the user scrolls down 20px from the top of the document, show the button
  window.onscroll = function() {
   scrollFunction()
  };
```

```
        function scrollFunction() {
            if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
                document.getElementById("movetop").style.display = "block";
            } else {
                document.getElementById("movetop").style.display = "none";
            }
        }

        // When the user clicks on the button, scroll to the top of the document
        function topFunction() {
            document.body.scrollTop = 0;
            document.documentElement.scrollTop = 0;
        }

    </script>
    <!-- //move top -->

</footer>
<!--//footer-9 -->
<!-- Js scripts -->
<!-- Template JavaScript -->
<script src="assets/js/jquery-3.3.1.min.js"></script>
<script src="assets/js/theme-change.js"></script>
<!-- MENU-JS -->
<script>
    $(window).on("scroll", function() {
        var scroll = $(window).scrollTop();

        if (scroll >= 80) {
            $("#site-header").addClass("nav-fixed");
        } else {
            $("#site-header").removeClass("nav-fixed");
        }
    });

    //Main navigation Active Class Add Remove
    $(".navbar-toggler").on("click", function() {
            $("header").toggleClass("active");
        });
        $(document).on("ready", function() {
            if ($(window).width() > 991) {
                $("header").removeClass("active");
            }
            $(window).on("resize",  function()  {
                if ($(window).width() > 991) {
                    $("header").removeClass("active");
                }
            });
        });

    </script>
    <!-- //MENU-JS  -->

    <!-- disable body scroll which navbar is in active -->
    <script>
        $(function() {
            $('.navbar-toggler').click(function()   {
                $('body').toggleClass('noscroll');
            })
        });

    </script>
    <!-- //disable body scroll which navbar is in active -->

    <!-- //bootstrap  -->
    <script src="assets/js/bootstrap.min.js"></script>

</body>

<
/html>
```

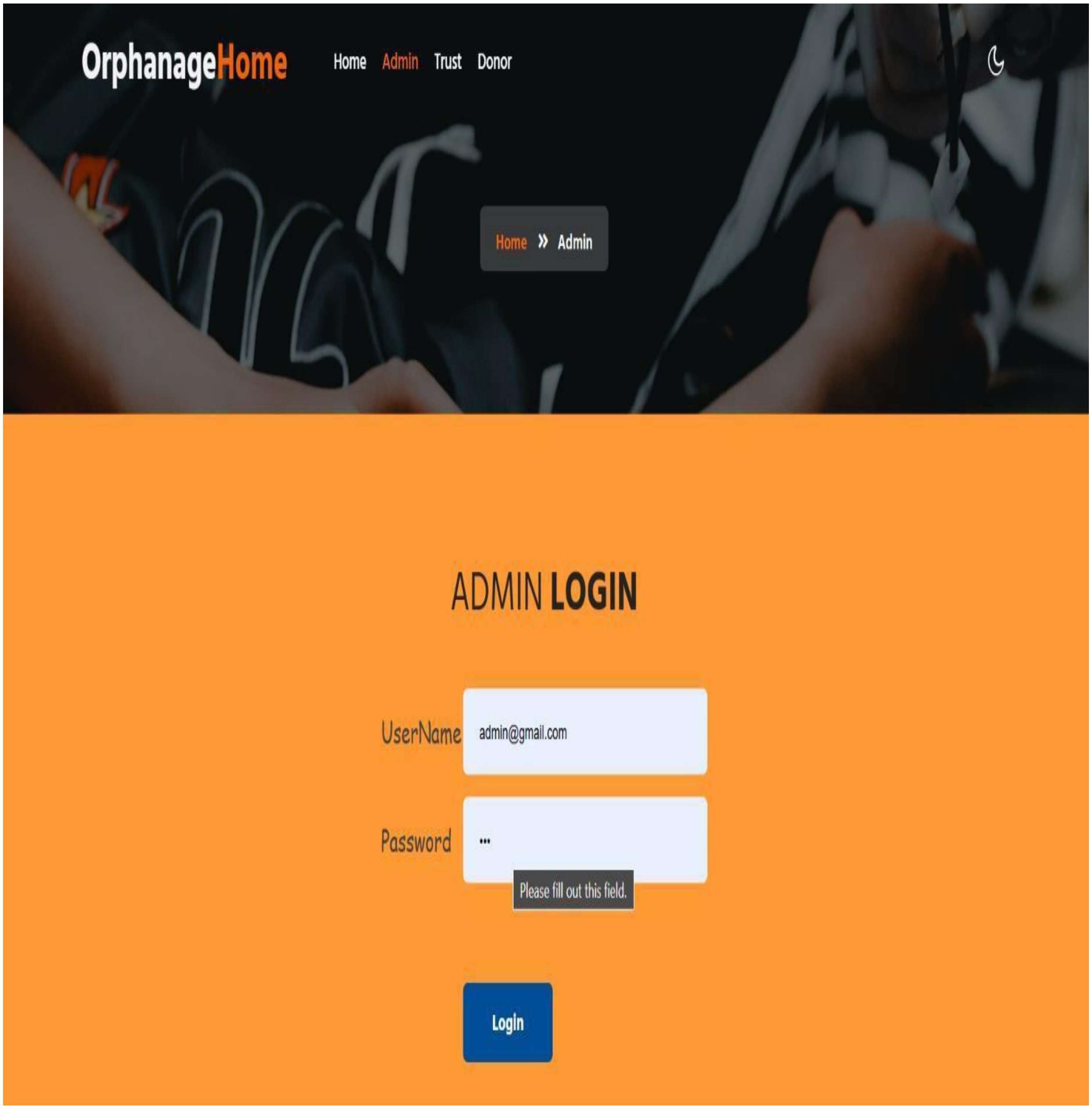
DONAR

```
<?php
extract($_POST);
session_start();
include("include/dbconnect.php");
if(isset($_POST['btn']))
{
  $username=$_REQUEST['username'];
  $password=$_REQUEST['password'];
  $_SESSION['username']=$username;
   if($conn->connect_error)
   {
     die("Connection failed: " . $conn->connect_error);
   }
   else
   {
     $qry="select * from donor_details where email='$username' and password='$password'";
     $result = mysqli_query($conn,$qry);
    $var=mysqli_num_rows($result);
      if($var>0)
      {
      ?>
      <script language="javascript">
      alert("login Success");
      window.location.href="donor_home.php";
      </script>
      <?php

      }
    else
     {
     ?>
     <script language="javascript">
     alert("Invalid user");
       window.location.href="donor.php";
     </script>
     <?php

     }
   }
  $conn->close();
}
?>
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title></title>
    <!--/google-fonts -->
    <link
href="//fonts.googleapis.com/css2?family=Hind+Siliguri:wght@300;400;500;600;700&display=swa
p" rel="stylesheet">
    <!--//google-fonts -->
    <!-- Template CSS -->
    <link rel="stylesheet" href="assets/css/style-starter.css">
</head>

<body>
    <!--/Header-->
    <header id="site-header" class="fixed-top">
        <div class="container">
            <nav class="navbar navbar-expand-lg navbar-light stroke py-lg-0">
                <h1><a class="navbar-brand pe-xl-5 pe-lg-4" href="#">
                        Orphanage<span class="sublog">Home</span>
                    </a></h1>
                <button class="navbar-toggler collapsed" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarScroll" aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle
navigation">
                    <span class="navbar-toggler-icon fa icon-expand fa-bars"></span>
                    <span class="navbar-toggler-icon fa icon-close fa-times"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarScroll">
                    <ul class="navbar-nav me-lg-auto my-2 my-lg-0 navbar-nav-scroll">
                        <li class="nav-item">
                            <a class="nav-link " aria-current="page"
href="index.php">Home</a>                        </li>
                        <li class="nav-item">
                            <a class="nav-link " href="admin.php">Admin</a>                        </li>
                        <li class="nav-item">
                            <a class="nav-link " href="trust.php">Trust</a>                        </li>


                        <li class="nav-item">
                            <a class="nav-link active" href="donor.php">Donor</a>                        </li>


                    </ul>




                    <!-- //toggle switch for light and dark theme -->
                    <!-- search popup -->


                    <!-- /search popup -->
                </div>
                <!-- toggle switch for light and dark theme -->
                <div class="mobile-position">
                    <nav class="navigation">
                        <div class="theme-switch-wrapper">
                            <label class="theme-switch" for="checkbox">
                                <input type="checkbox" id="checkbox">
                                <div class="mode-container">
                                    <i class="gg-sun"></i>
                                    <i class="gg-moon"></i>
                                </div>
                            </label>
                        </div>
                    </nav>
                </div>
                <!-- //toggle switch for light and dark theme -->
            </nav>
        </div>
    </header>
    <!--//Header-->
    <!-- breadcrumb -->
    <section class="w3l-about-breadcrumb">
        <div class="breadcrumb-bg breadcrumb-bg-about py-5">
            <div class="container py-lg-5 py-md-4">
                <div class="w3breadcrumb-gids text-center pt-5">
                    <div class="w3breadcrumb-right">
                        <ul class="breadcrumbs-custom-path">
                            <li><a href="#">Home</a></li>
                            <li class="active"><span class="fas fa-angle-double-right mx-2"></span> Donor
</li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
```

```
  </section>
  <!--//breadcrumb-->

   <section class="w3l-contact-1 py-5" id="contact">
    <div class="contacts-9 py-lg-5 py-md-4">
      <div class="container">
        <div class="contactct-fm map-content-9">
          <div class="header-title text-center">

            <h3 class="title-w3l mb-2">DONOR
              <span class="span-bold"> LOGIN</span>
            </h3>

          </div>

          <form action="" class="pt-lg-4" method="post" enctype="multipart/form-data">
           <table height="270" align="center">
            <tr>
             <td width="75">Email</td>
             <td width="311"><input type="email" name="username" placeholder="Email"
required=""></td>
            </tr>
            <tr>
             <td>Password</td>
             <td><input type="password" name="password" placeholder="Password"
required=""></td>
            </tr>
            <tr>
             <td>&nbsp;</td>
             <td><span class="tp">
              <button type="submit" name="btn" class="btn btn-primary btn-style mt-lg-5 mt-
4">Login</button>
             </span></td>
            </tr>
          <tr>
           <td>&nbsp;</td>
           <td><a href="donor_register.php">New User Register Here... </a></td>
          </tr>
         </table>


        </form>
      </div>

    </div>
  </div>
</section>
<!--//w3l-project-->
<!--/w3l-footer-29-main-->
<footer class="w3l-footer-29-main">
  <div class="footer-29 py-5">
    <div class="container py-lg-4">
     <div class="bottom-copies text-center">
        <p class="copy-footer-29">All rights reserved. Design by
          <a href="#"> Admin</a>
        </p>
     </div>
   </div>
  </div>

  <!-- move top -->
  <button onClick="topFunction()" id="movetop" title="Go to top">
    <span class="fa fa-angle-up"></span>
  </button>
  <script>
    // When the user scrolls down 20px from the top of the document, show the button
    window.onscroll = function() {
            scrollFunction()
        };

        function scrollFunction() {
            if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
                document.getElementById("movetop").style.display = "block";
            } else {
                document.getElementById("movetop").style.display = "none";
            }
        }

        // When the user clicks on the button, scroll to the top of the document
        function topFunction() {
            document.body.scrollTop = 0;
            document.documentElement.scrollTop = 0;
        }

    </script>
    <!-- //move top -->

</footer>
<!--//footer-9 -->
<!-- Js scripts -->
<!-- Template JavaScript -->
<script src="assets/js/jquery-3.3.1.min.js"></script>
<script src="assets/js/theme-change.js"></script>
<!-- MENU-JS -->
<script>
    $(window).on("scroll", function() {
        var scroll = $(window).scrollTop();

        if (scroll >= 80) {
            $("#site-header").addClass("nav-fixed");
        } else {
            $("#site-header").removeClass("nav-fixed");
        }
    });

    //Main navigation Active Class Add Remove
    $(".navbar-toggler").on("click", function() {
        $("header").toggleClass("active");
    });
    $(document).on("ready", function() {
        if ($(window).width() > 991) {
            $("header").removeClass("active");
        }
        $(window).on("resize", function() {
            if ($(window).width() > 991) {
                $("header").removeClass("active");
            }
        });
    });

</script>
<!-- //MENU-JS -->

<!-- disable body scroll which navbar is in active -->
<script>
    $(function() {
        $('.navbar-toggler').click(function() {
            $('body').toggleClass('noscroll');
        })
    });

</script>
<!-- //disable body scroll which navbar is in active -->
```

```
    <!-- //bootstrap -->
    <script src="assets/js/bootstrap.min.js"></script>

</body>

</html>
```

B. SCREENSHOTS



ADMIN LOGIN

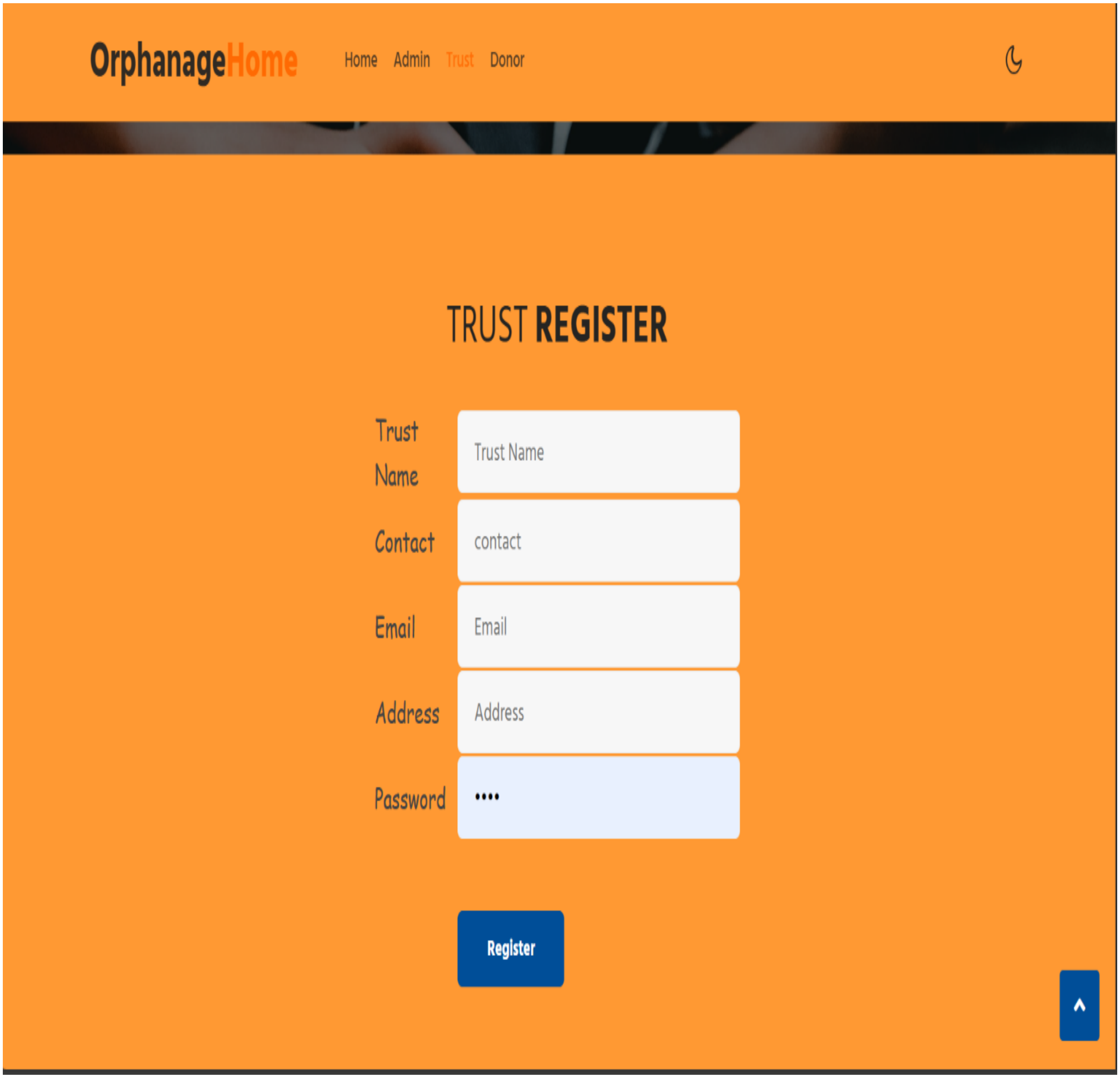
UserName admin@gmail.com

Password •••

Please fill out this field.

Figure: B1 Admin login page

This is the admin page to login and access to approve a trust and maintain the doner and trust datas.



TRUST REGISTER

Trust Name: Trust Name

Contact: contact

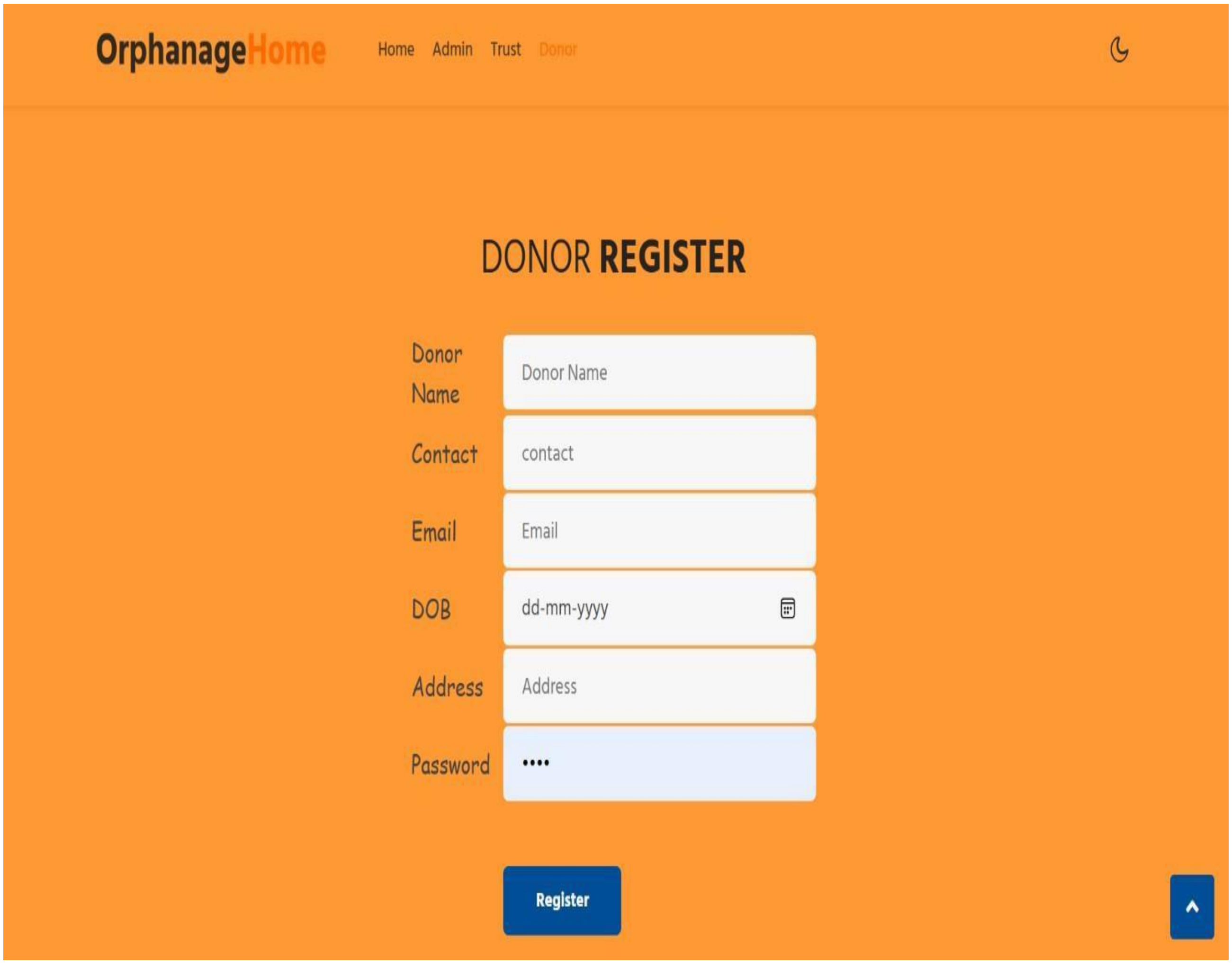
Email: Email

Address: Address

Password: ••••

Figure: B2 Trust login page

It is trust registeration page to register their informations and create their login.



DONOR **REGISTER**

Donor Name	Donor Name
Contact	contact
Email	Email
DOB	dd-mm-yyyy
Address	Address
Password	••••

Figure: B3 Donor register page

It is donor registration page to register their informations and create their login.

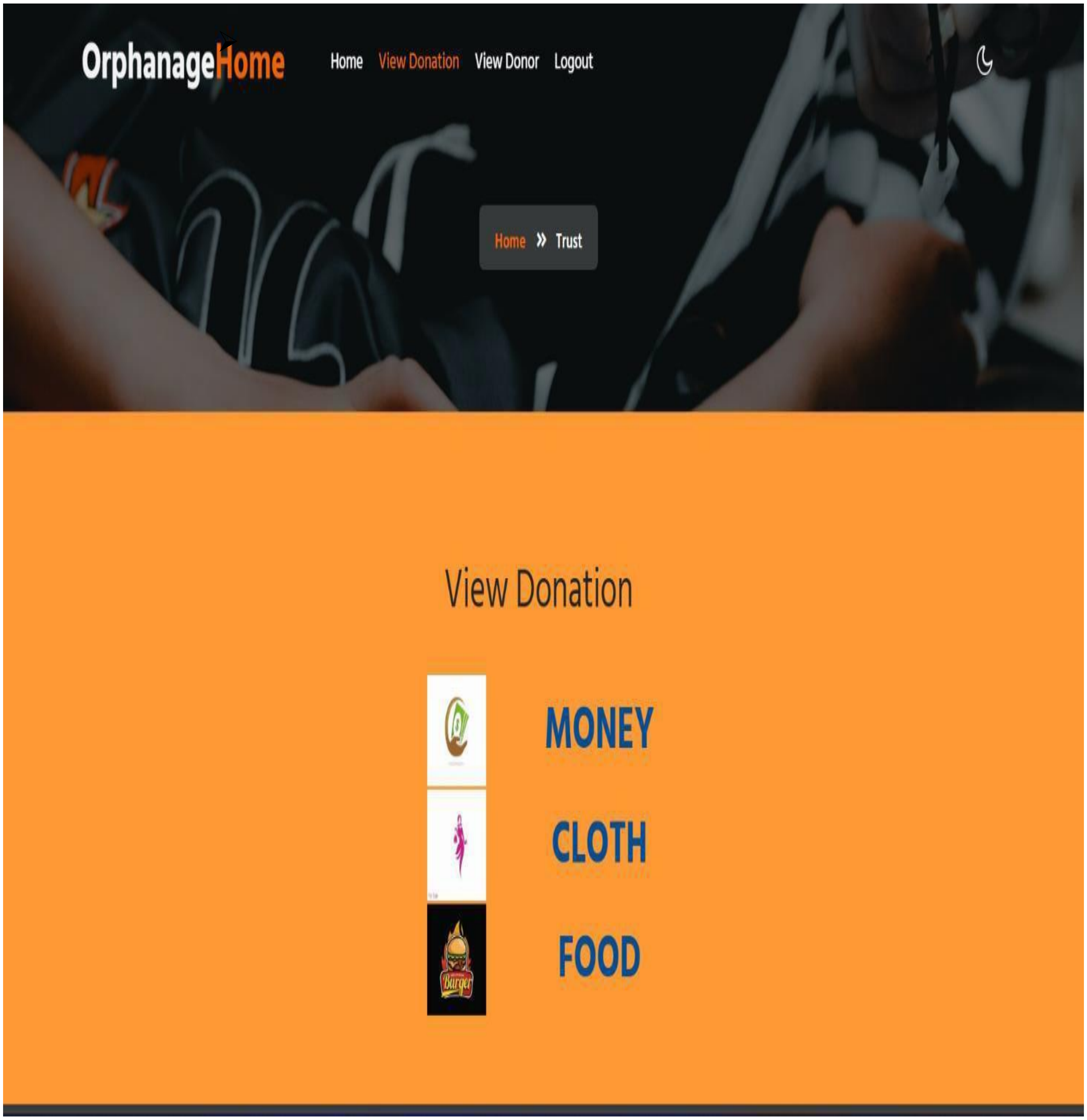


Figure: B4 Donation page

It is the donation page of the donar to give the donation by the following way of money,cloth,food

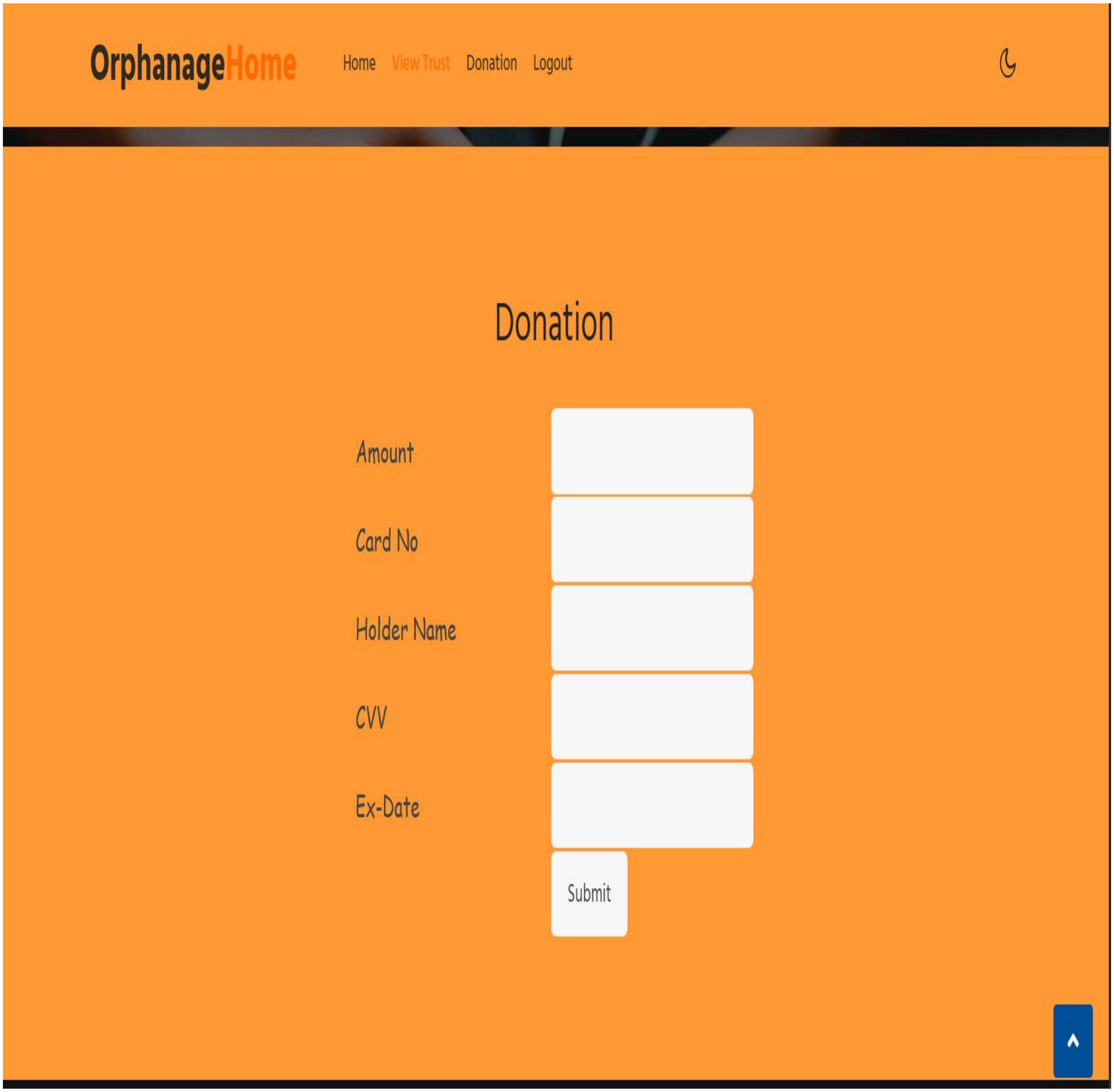
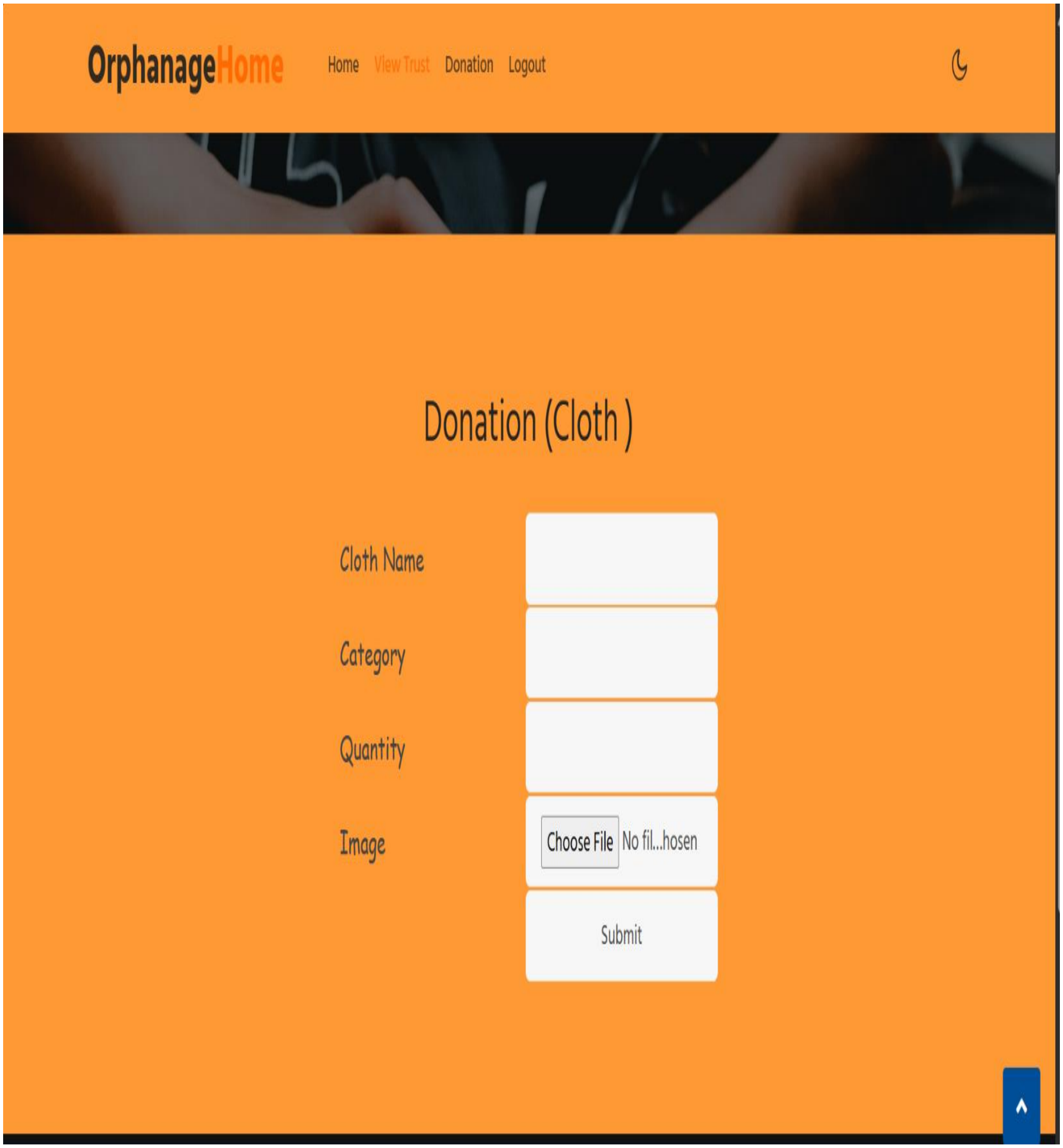


Figure: B5 Donar page

It is the donation page of the donar to give the donation by the following way of money.



Donation (Cloth)

Cloth Name	
Category	
Quantity	
Image	Choose File No fil...hosen
	Submit

Figure:B6 DONOR PAGE

It is the donation page of the donar to give the donation by the following way of cloth

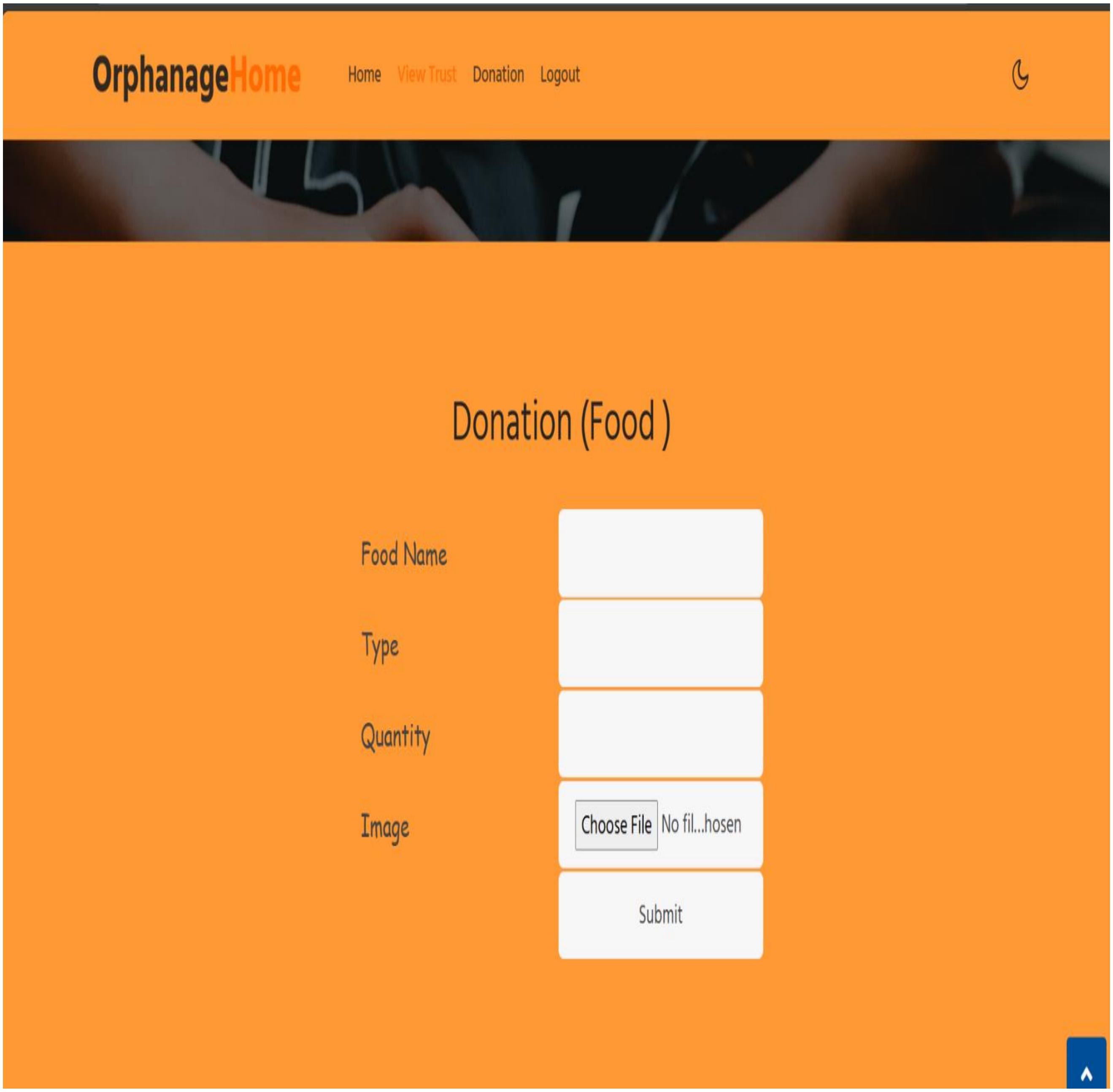


Figure:B6 DONOR PAGE

It is the donation page of the donar to give the donation by the following way of food

REFERENCES

Authors: Seyed M.M. Choudhury,” Get a Handle on Your Data”, 2nd Edition (2018)

Author: Paul DuBois, "High Performance MySQL”, 3rd Edition (2014).

Authors: Baron Schwartz, "MySQL Cookbook",3rd Edition (2012)

Author: David Flanagan, “The Definitive Guide",7th Edition (2020)

Authors: Luke Welling, PHP and MySQL Web Development"

Authors: Hassan Djirdeh, "Fullstack Vue: The Complete Guide to Vue.js",1st Edition (2018)

https://www.geeksforgeeks.org/mysql/

https://www.geeksforgeeks.org/javascript/

https://www.geeksforgeeks.org/php-introduction/