**Project Report on**

**“THE HOPE”**



**SRI BALAJI UNIVERSITY**

**SCHOOL OF COMPUTER STUDIES**

**Program: Masters of Computer Application**

**2023 Batch**

**SUBMITTED BY:**

**Ayush Verma(MCA2302012)**

**Dhiraj Agarwal(MCA2302123)**

**Abhishek Shukla(MCA2302113)**

# CERTIFICATE

Certified that the Project Report entitled **“THE HOPE”**, submitted by **Ayush Verma, Dhiraj Agarwal, Abhishek Shukla** of MCA, is their own work and has been carried out under my supervision. It is recommended that the candidates may now be evaluated for their work by the University.

**Name 1: Ayush Verma Roll no: MCA2302012**

**Name 2: Dhiraj Agarwal Roll no: MCA2302123**

**Name 3: Abhishek Shukla Roll no: MCA2302113**

Prof. Ashwini Shinde

(Assistant Professor)

# ACKNOWLEDGEMENT

We feel immense pleasure to introduce “**THE HOPE**” as our major project. I express my sincere thanks to our instructor **Prof. Ashiwini Shinde** who guided us to the successful completion of this project report. We take this opportunity to express our deep sense of gratitude for their individual guidance, constant encouragement and immense motivation which have sustained our efforts at all stages of this project report. We are grateful and appreciate all the staff members of the School of Computer Application for their cooperation and support.

I extend my sincere thanks to our principal Dr.G.Y.Shitole and vice principle Dr .Richa Purohit for his support and for all the facilities provided for the preparation of this project report.

Also, we wish to thank our parents & friends who helped us a lot in collecting data, pictures and continuous help and support.

Finally, we would wish to thank everyone involved in this project time.

# ABSTRACT

Help service provider is a website which provides a helping hand to the community. Our aim is to bring more and more people into our website and help them to get benefit, and reach more and more people so that they can join us for the better developmental side. **THE HOPE** was build using HTML, CSS, PHP, MYSQL.

Help service provider is basically website is proved to be a boon for people who don’t want to get out of there comfort zone to donate there resources by going through this website. It is really beneficial for the NGO, Charity etc.

The aim is to automate it’s existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling there requirement so that there valuable information/data can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describe how to manage a good performance and better service for the people.

# TABLE OF CONTENT

|  |
| --- |
| **Chapter 1 : INTRODUCTION TO THE LANGUAGES USED**  **6-8** |
| * 1. HTML & CSS |
| * 1. WHY HTML & CSS |
| 1.3 INTRODUCTION TO PHP |
|  |
| **Chapter 2 : INTRODUCTION TO “Help Service Provider” 9-13** |
| 2.1 INTRODUCTION |
| 2.2 PURPOSE & SCOPE |
| 2.3 ADVANTAGES |
| 2.4 SRS |
| 2.5 Use |
| 2.6 Purpose |
| 2.7 Functional Requirements |
| 2.8 Non-Functional Requirements. |
| 2.9 Software Tools |
| 2.10 Deployment |
| 2.11 FEASIBILITY STUDY |
|  |
| **Chapter 3 : DIAGRAMS 15-24** |
| * 1. Entity Relationship Diagram |
| * 1. Data Flow Diagram |
| * 1. Use Case Diagram |
| * 1. Class Diagram |
| * 1. Activity Diagram |
| * 1. Sequence Diagram |
| * 1. Deployment Diagram |
|  |
| **Chapter 4: USER INTERFACE 25-31** |
| 4.1 Home Page |
| 4.2 About Us |
| 4.3 Login Page |
| 4.4 Registration Page |
| 4.5 Donation Service Page |
| 4.6 Food Donation Page |
| 4.7 Clothes Donation Page |
| 4.8 Money Donation Page |
| 4.9 Book Donation Page |
| 4.10 Add On Cart Page |
| 4.11 Receipt of Food Page |
| 4.12 Download Food Donation Receipt |
| 4.13 Receipt of Money Page |
| 4.14 Download Money Donation Receipt |
|  |
| Chapter 5: Test Case 32-39 |
| 5.1 Black Box Testing |
|  |
| Chapter 6: Conclusion & Recommendations 40 |
|  |
| Chapter 7:Biblography & Reference 41 |

**Chapter 1:**

**INTRODUCTION TO THE LANGUAGES USED HTML & CSS** :

* **HTML:**

Html stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, which describes the structure of a Web page. HTML consists of a series of elements that tell the browser how to display the content. These elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Developer of HTML: Tim Berners-Lee

* **CSS :**

CSS stands for Cascading Style Sheets. This language describes how HTML elements are to be displayed on screen, paper, or in other media. It saves a lot of work, and can also control the layout of multiple web pages all at once. External stylesheets are stored in CSS files with .css extension. Developer of CSS : World Wide Web Consortium (W3C)

**WHY HTML & CSS?**

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, whereas CSS provides the layout, for a variety of devices along with graphics and scripting. HTML and CSS are the basics of building any Web pages or Web Applications.

You can learn more in detail as follows :

* **HTML**

In essence, HTML is used for creating the primary content of a webpage, giving it structure. You start by writing words, then apply tags or elements to these words. The web browser then reads this and can then understand the heading of a page, any paragraphs, and where the page starts and finishes, thus filling your web page with content. HTML is supported by every single browser and is established on pretty much every webpage in existence. You don’t need any licenses, you don’t need to pay for it, and it can be pretty easy to learn and code. If we can compare a webpage to the human body, then HTML is the bones of the body.

* **CSS**

If HTML is the bones of the body, then CSS is the skin that covers it. It’s used for background colour, styling, layout, borders, shadowing – all the essential design bits and bobs that make a webpage look slick and smart.

CSS enables you to distinguish between presentation and content by modifying the design and display of HTML elements.

Presentation and ease of use are a couple of the main things that CSS has brought to web design by translating the way content looks on a webpage and what else goes on it to complement that content.

While frequently used in correlation with HTML, it is actually independent of it, and can be used with any XML-based markup language.

**INTRODUCTION TO PHP:**

Certainly PHP (Hypertext Preprocessor) is a widely used server-side scripting language primarily designed for web development. It is used to create dynamic web pages by embedding PHP code within HTML. When a user requests a web page that contains PHP code, the server processes the PHP code and sends the resulting HTML to the user's web browser, which displays the page.

When integrating PHP with HTML and CSS for a project, you can create interactive and dynamic web pages.

**Chapter 2**

**INTRODUCTION TO “HELP SERVICE PROVIDERS”**

**Introduction:**

The “HELP SERVICES PROVIDER” is a non-profitable website which offers a wide range of help services for the orphanages as well as NGOs. We are committed to provide services which are reliable and helpful for all the orphanages and NGOs. Domestic help services is easy to navigate and user friendly.

We Provide a secure platform for you to get as well as provide help services such as donating and receiving books, money etc. we aim to be a helping hand for all our users. We as a service provider aim to help public find the best helping services in online mode.

Easier and Simpler to provide and receive help services at your place that’s what our goal is. Rather than an hassle searching for orphanages & NGOs to provide help and causing ruckus in offline mode, we try to take it all online with our innovative idea for providing kindness

**Purpose:**

A help service provider website for children is a place where people who care about kids can go to offer assistance. It's like a hub where those who want to help, like donating money, books, or groceries, can connect with kids who need these things. The website explains why these children need help, like for school supplies or food, and shows how donating can make a real difference in their lives. It's an easy way for kind-hearted folks to give what they can to make sure kids have what they need to grow and learn. The site also shares stories about how these donations help, so donors know their support matters. It's a friendly and welcoming place that makes it simple for anyone who wants to lend a hand to do so, making life a bit better for kids who could use some extra help.

**Scope:**

The main aim of our proposed system is to address the current problems and limitations of the existing help service website. The system will be designed to improve the overall user experience for the users and simplify the process of getting and receiving help services online.

The scope of the proposed system will include the development of a user-friendly platform that give users an easy experience of providing as well as getting help services. Also include the implementation of a feedback mechanism that enables customers to rate and review our services. This will provide valuable insights to improve the services and enhance user satisfaction.

Overall, the proposed system will aim to address the requirements and needs of NGOs & orphanages and help them fulfil their needs. It will provide a reliable and secure platform for users to provide as well as receive help with ease and convenience.

**Advantages:**

* **Convenience:** Online help service websites provide the convenience of showing your kindness and helping others from your home, eliminating the need for physical NGO visits.

* **Boundless Services:** Users can access a broader selection of service that they want to provide and receive, including money but it has its own conditions.

* **Timesaving:** Users save time by avoiding going in search of Orphanages and NGOs to provide to the one in need.

* **Reduce Frauds:** There are no chances of getting duped as our site provides a list of all authorized NGOs & orphanages, which are checked and cross checked for our users convenience.

* **Accessibility:** These platforms are accessible 24/7, allowing users to provide or receive help services at their convenience.

* **Information Access:** Users can access detailed information about each service, including Orphanage & NGOs phone number.

* **SRS- SOFTWARE REQUIREMENT SPECIFICATION:**

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all stakeholders (business, users) needs.

•Use: To create a platform where users can easily offer assistance to underprivileged children by providing various forms of aid

**3. Functional Requirements**

**User Registration and Authentication:**

* Use: Allow users to create accounts and log in securely.
* Purpose: To personalize user experiences and ensure security in transactions.

**Donation Management:**

* Use: Enable users to donate money, books, groceries, or other items easily.
* Purpose: To facilitate contributions and assist in fulfilling specific needs of children.

**Profile Creation for Children in Need:**

* Use: Admins or authorized users can create profiles for children requiring assistance.
* Purpose: To showcase specific needs and stories of children seeking support

**Information Display:**

* Use: Display information about the charity's mission, success stories, impact of donations, and ways to help.
* Purpose: Educate users, build trust, and encourage engagement.

**Donor-Child Matching System:**

* Use: Match donors with specific children or causes they wish to support.
* Purpose: To connect donors directly to those they want to help.

**Information Display:**

* Use: Display information about the charity's mission, success stories, impact of donations, and ways to help.
* Purpose: Educate users, build trust, and encourage engagement.

**Donor-Child Matching System:**

* Use: Match donors with specific children or causes they wish to support.
* Purpose: To connect donors directly to those they want to help.

**3. Non-Functional Requirements**

**Security and Privacy:**

* Purpose: Ensure that user data and financial transactions are encrypted and protected.

**Scalability:**

* Purpose: Allow the platform to handle increased traffic and accommodate more users and

donations over time.

**Usability:**

* Purpose: Provide an intuitive and user-friendly interface for easy navigation and donation

processes.

**Performance:**

* Purpose: Ensure the website's speed and responsiveness, minimizing downtime or delays

during donation processes.

**Accessibility:**

* Purpose: Design the website to be accessible to users with disabilities, considering features like screen readers and alternative text.

**Compliance and Regulation:**

* Purpose: Ensure compliance with legal regulations, especially regarding financial transactions and data protection laws.

**Software Tools**

1. You can use any software to run this website like google chrome, Firefox, etc. We are using google chrome.
2. We are working with HTML, CSS for the front-end.
3. For back-end support we are using PHP & SQL.

**Deployment**

Operating system server : Windows 10.

**FEASIBILITY STUDY**

Preliminary investigation examines the project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* + Technical Feasibility
  + Operation Feasibility
  + Economical Feasibility

**Technical Feasibility:**

**Definition:** Technical feasibility assesses whether the project can be implemented from a technological standpoint.

**Infrastructure:** Evaluate the availability of required technical infrastructure such as servers, hosting services, and bandwidth to support the website's expected traffic and data storage needs.

**Software & Development:** Determine if necessary software, programming languages, and development tools are available and appropriate for building the website.

**Integration:** Assess the feasibility of integrating different systems or third-party services (like payment gateways) necessary for website functionality.

**Operational Feasibility:**

**Definition:** Operational feasibility examines whether the project aligns with operational aspects and how well it will fit within existing procedures and operations.

**User Acceptance:** Conduct surveys or assessments to gauge the acceptance of the website among potential users (donors, recipients, and administrators).

**Training and Support:** Determine if the required training can be provided to users (administrators, moderators) to manage the website efficiently.

**Workflow Integration:** Assess how well the website aligns with existing workflow processes and how easily it can be incorporated into daily operations.

**Economic Feasibility:**

**Definition:** Economic feasibility evaluates the cost-effectiveness and financial viability of the project.

**Cost Analysis:** Evaluate the costs involved in building, hosting, and maintaining the website, including development costs, hosting fees, maintenance, and potential upgrades.

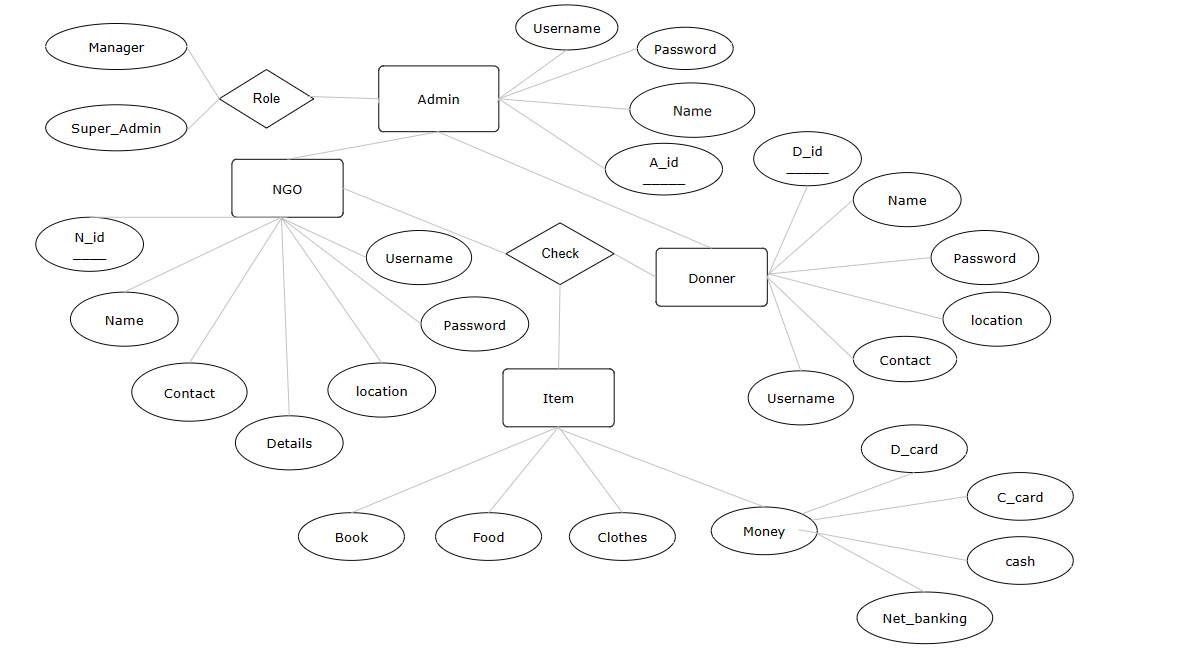
**Revenue Generation:** Determine potential sources of revenue or funding, such as donations, sponsorships, or grants, to sustain the website and its operations.

**Return on Investment (ROI):** Estimate the benefits or returns expected from the website in terms of impact on the community versus the costs incurred.

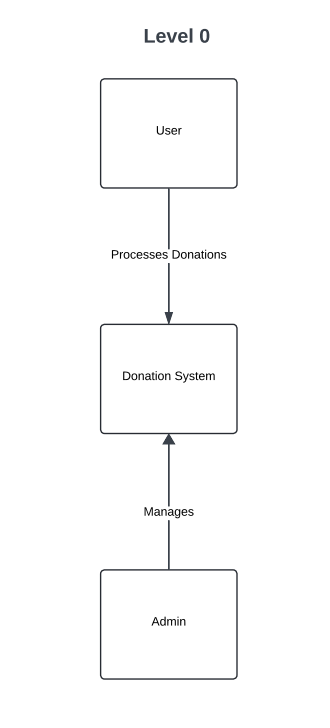
**Chapter 3**

**DIAGRAMS**

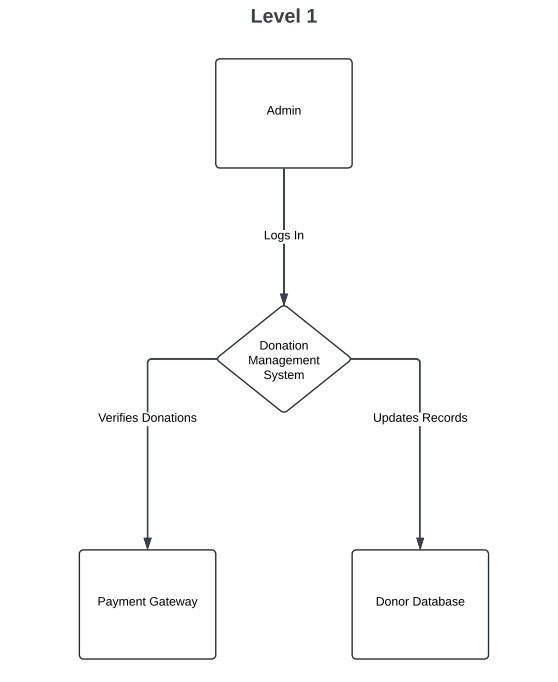
* **ENTITY RELATIONSHIP DIAGRAM**

****

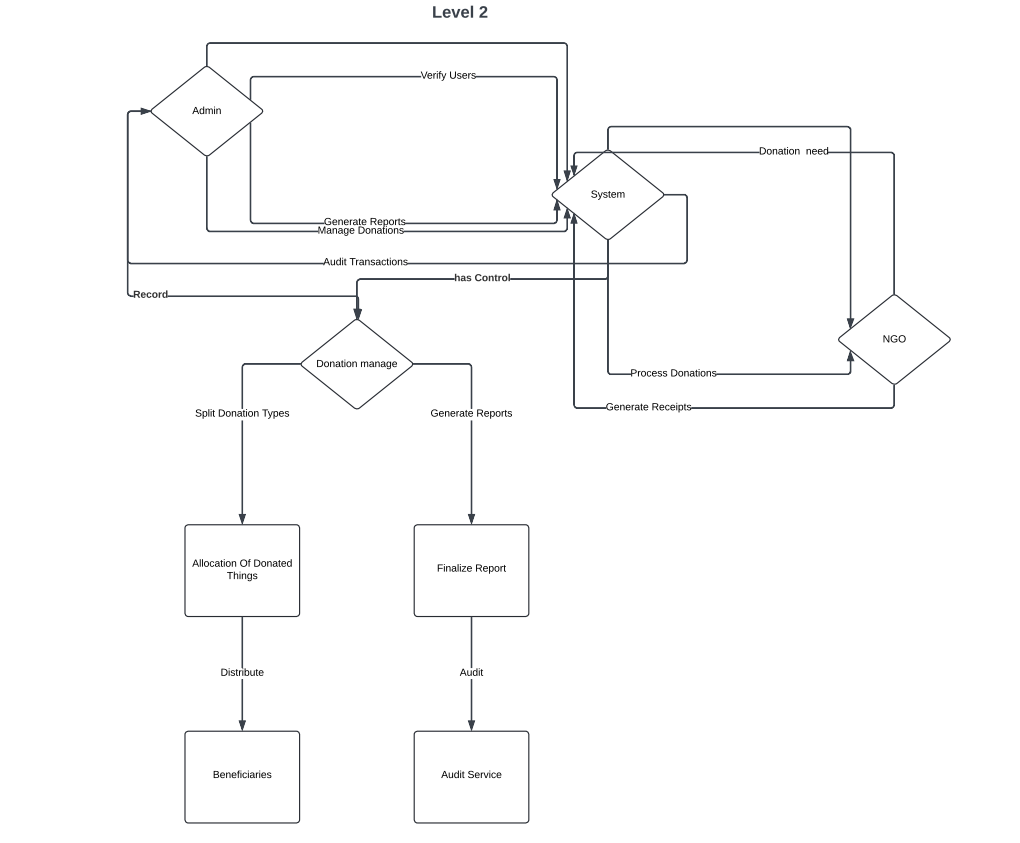
* **Data Flow Diagram**
* **Level 0**

****

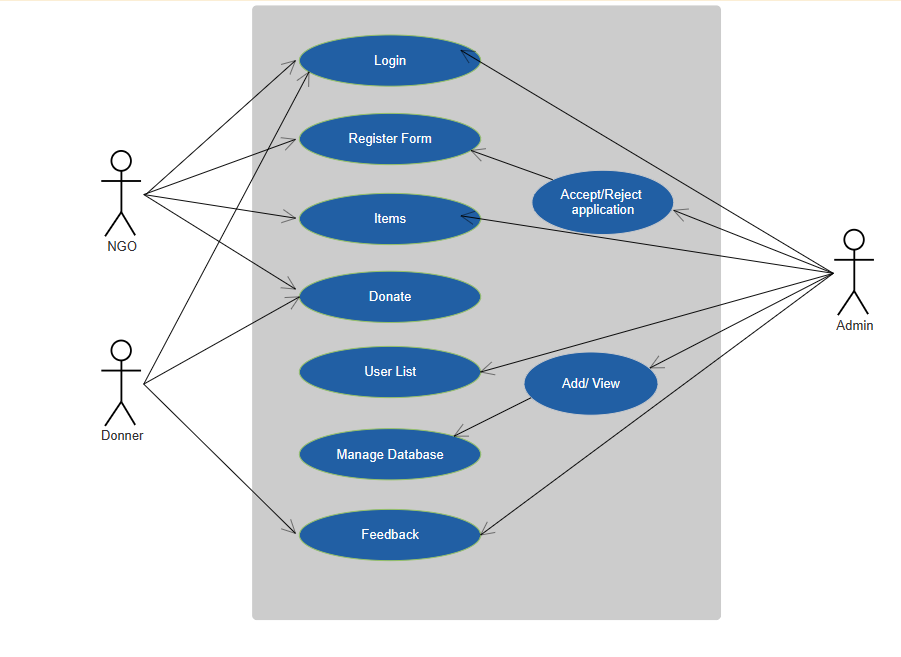
* **Level 1**

****

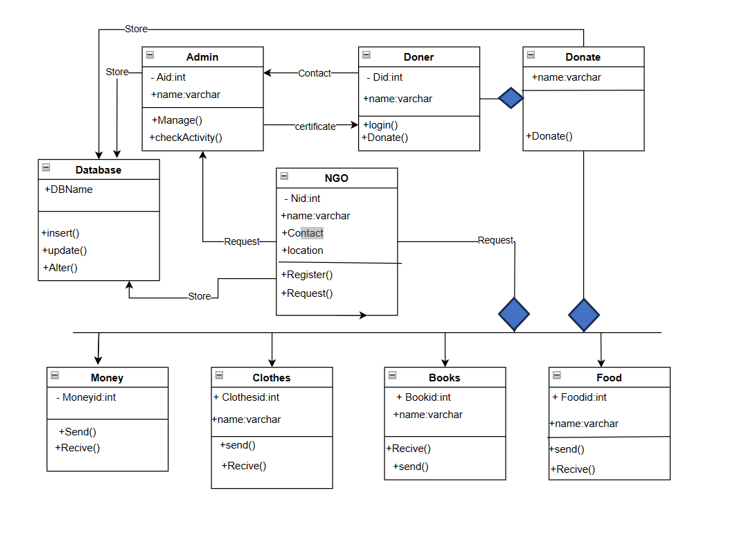
* **Level 2**

****

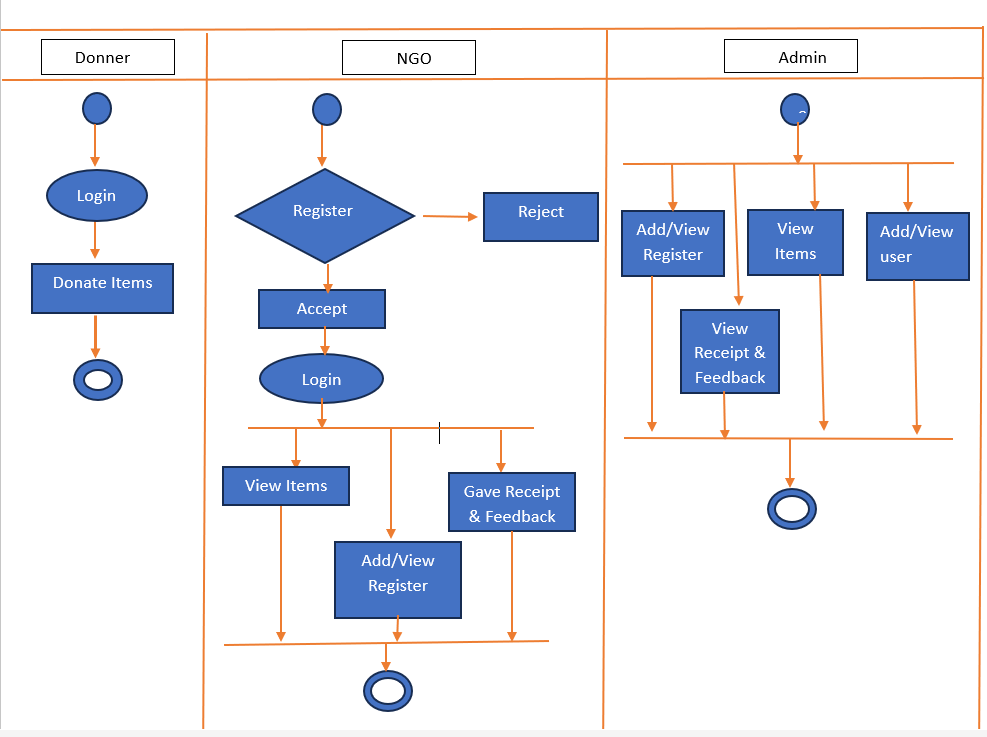
* **Use Case Diagram**

****

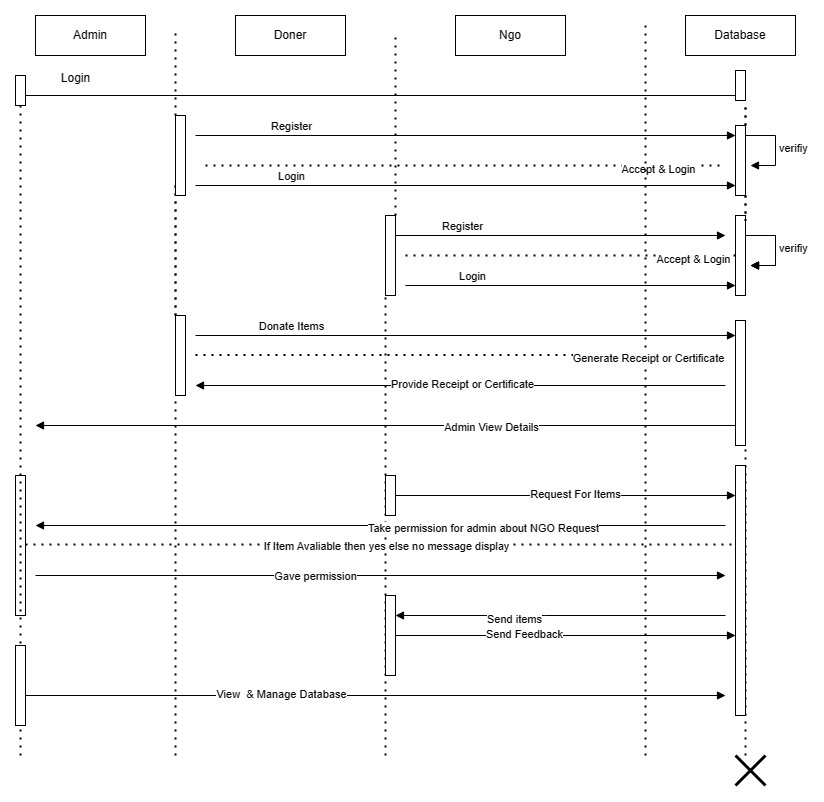
* **Class Diagram**

****

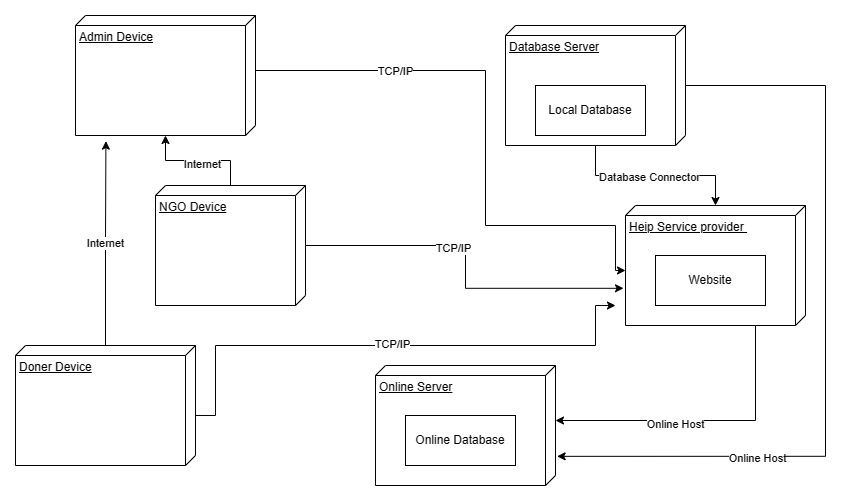
* **Activity Diagram**

****

* **Sequence Diagram**

****

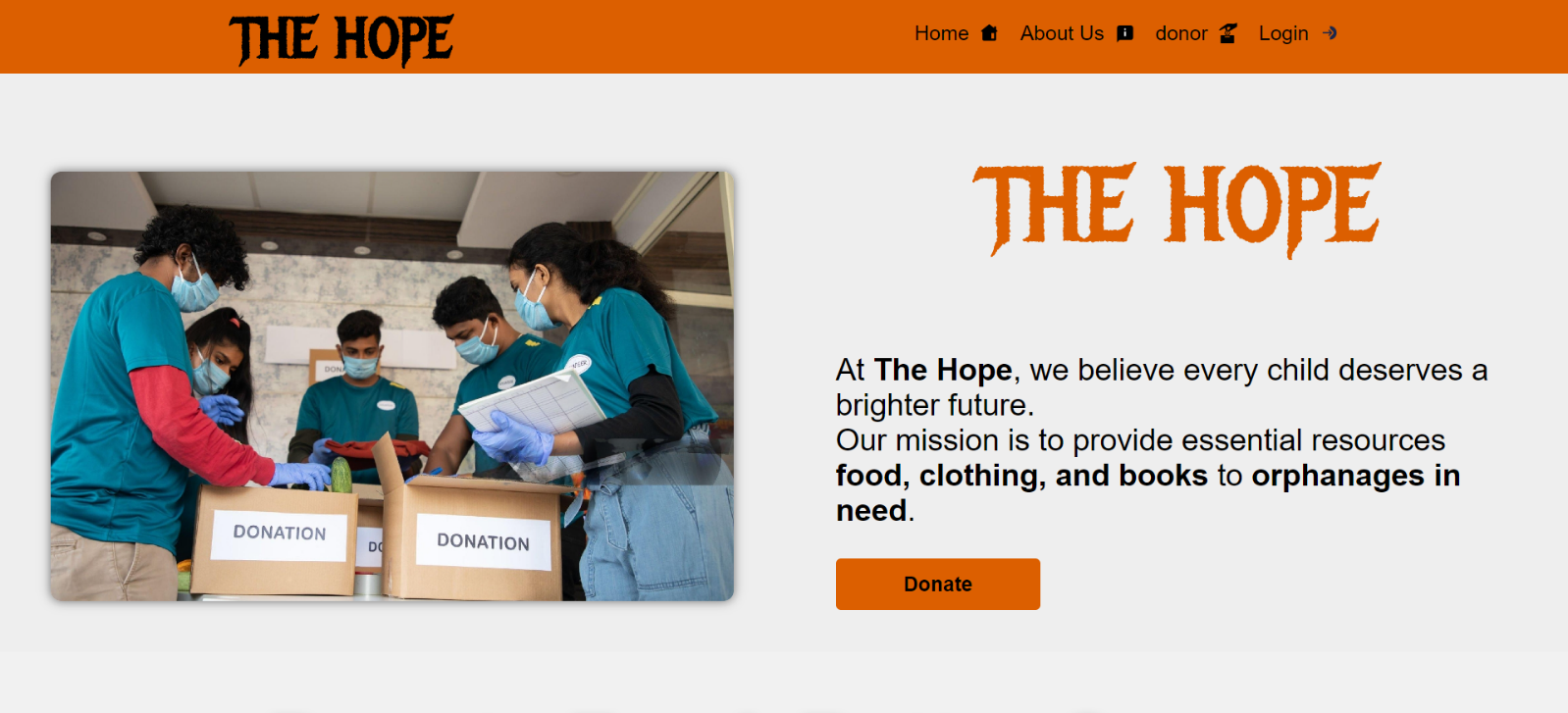
* **Deployment Diagram**

****

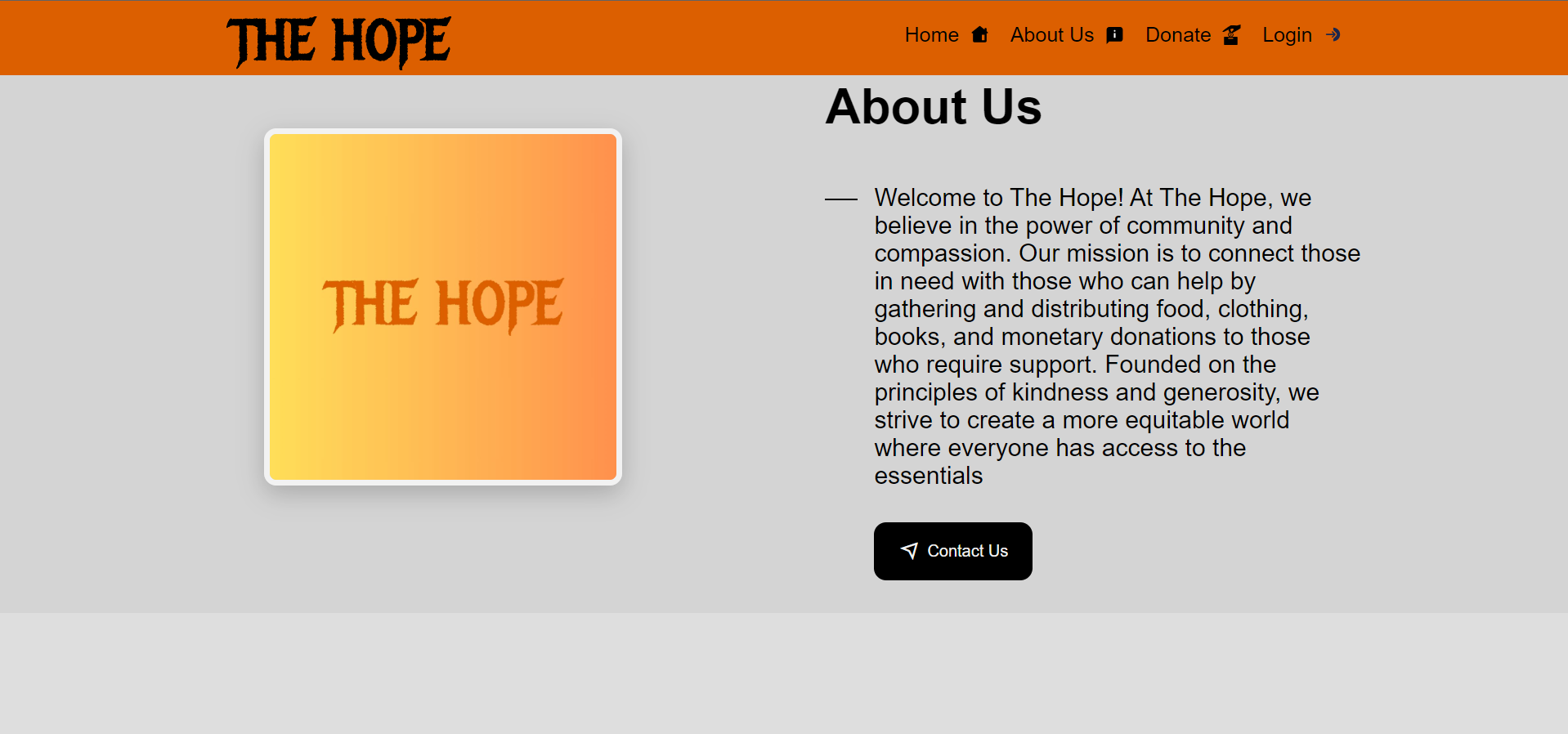
**Chapter 4:**

**USER INTERFACE**

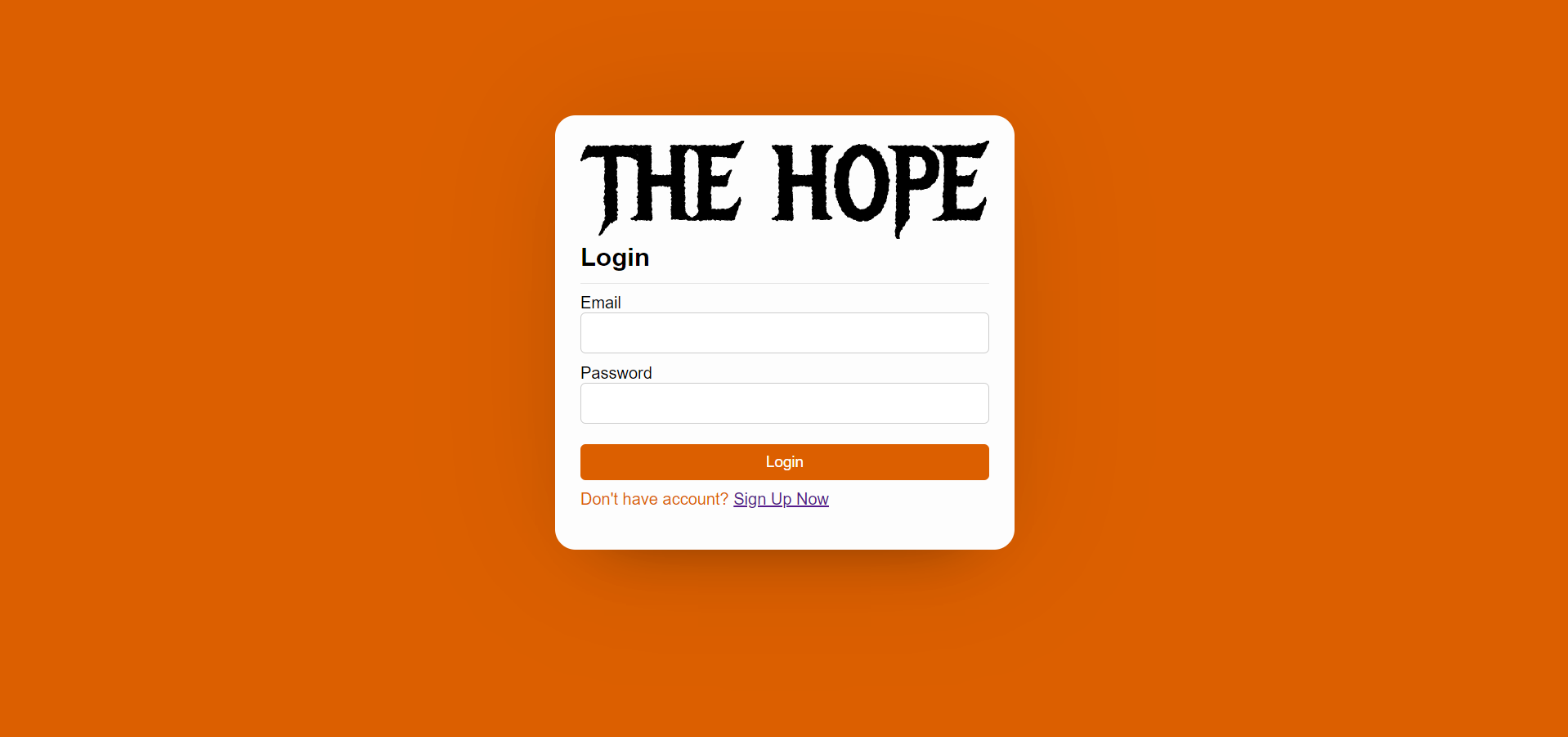
4.1Home Page

****

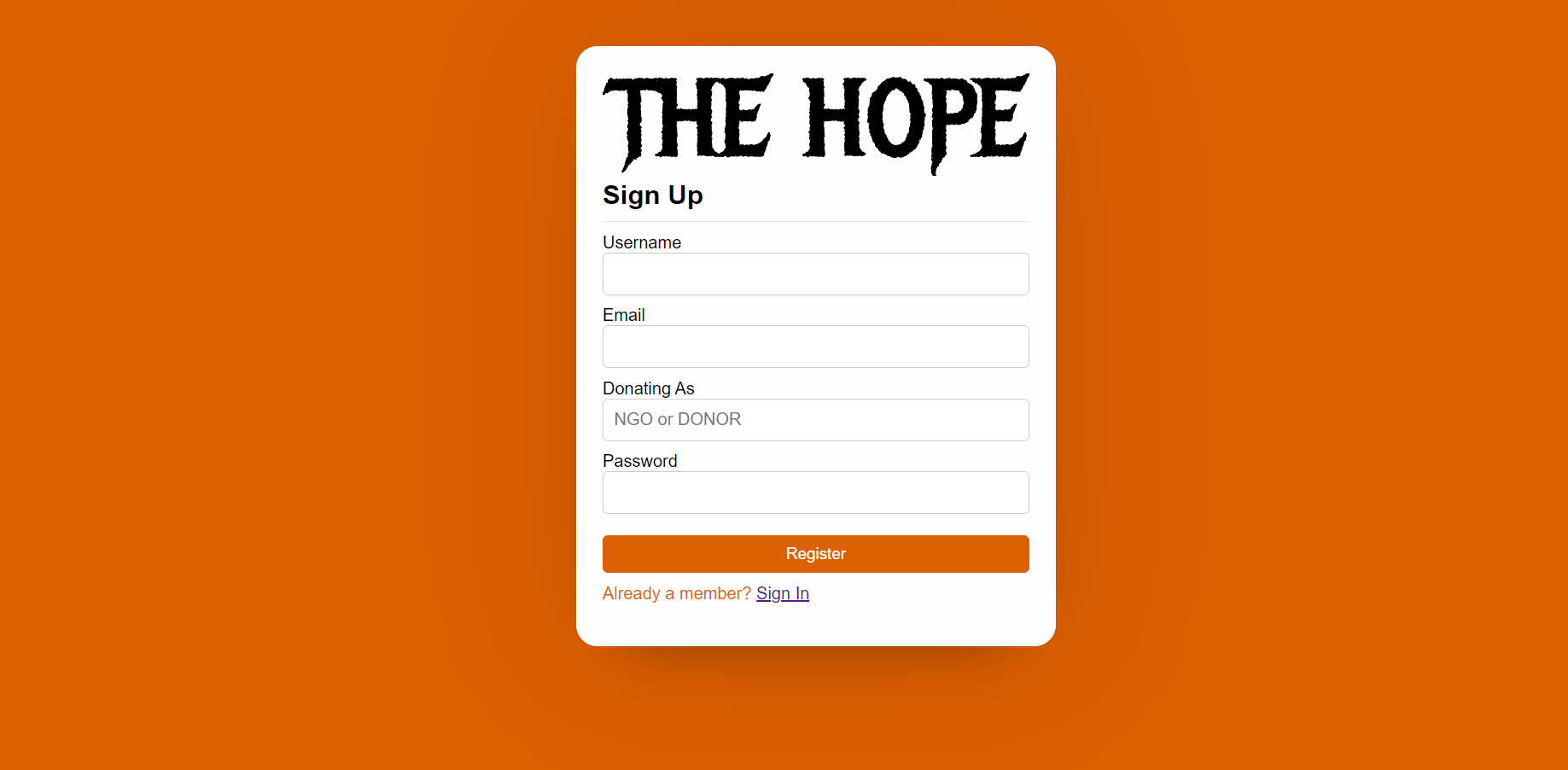
**4.2 About Us**

****

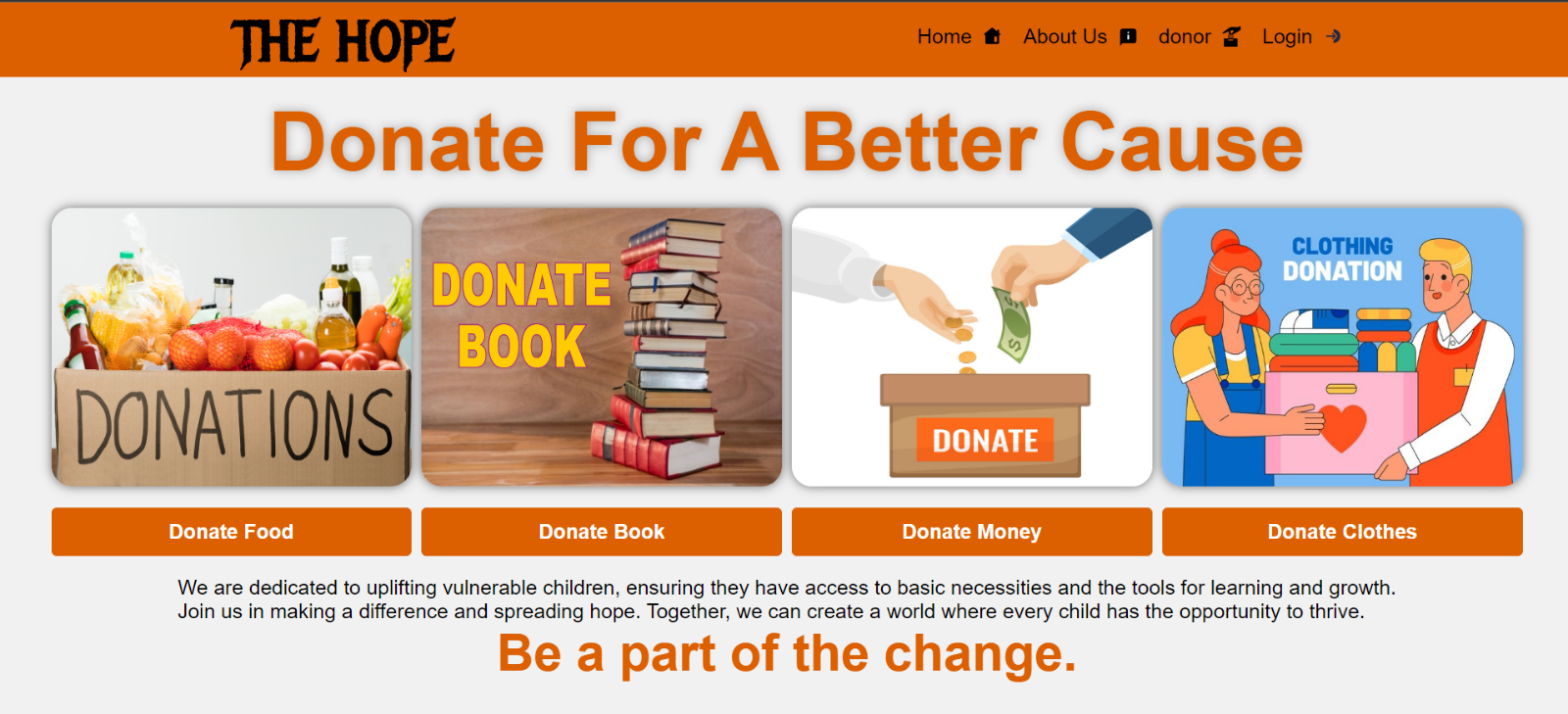
**4.3 Login Page**

****

**4.4 Registration Page**

****

**4.5 Donation Service Page**

****

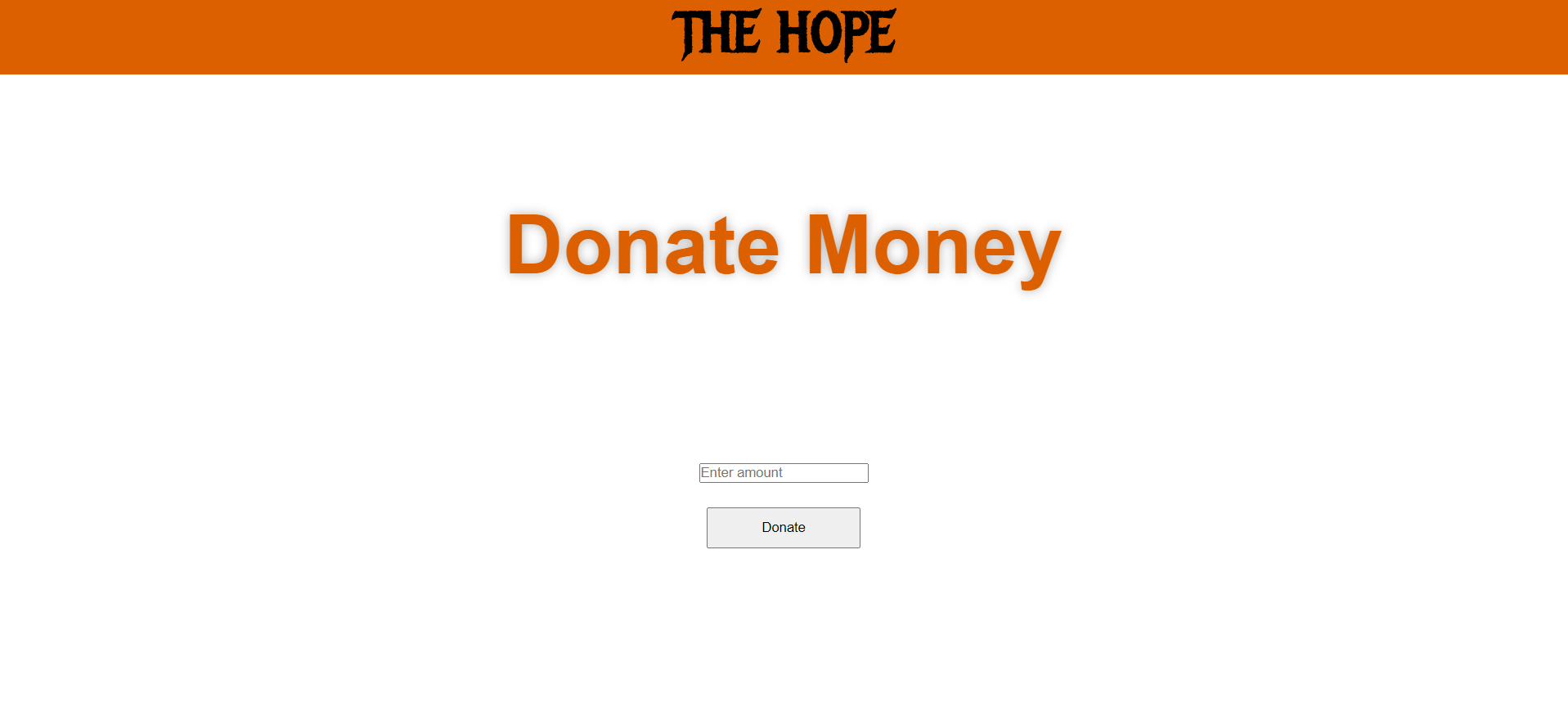
**4.6 Food Donation Page**

****

**4.7 Clothes Donation Page**

****

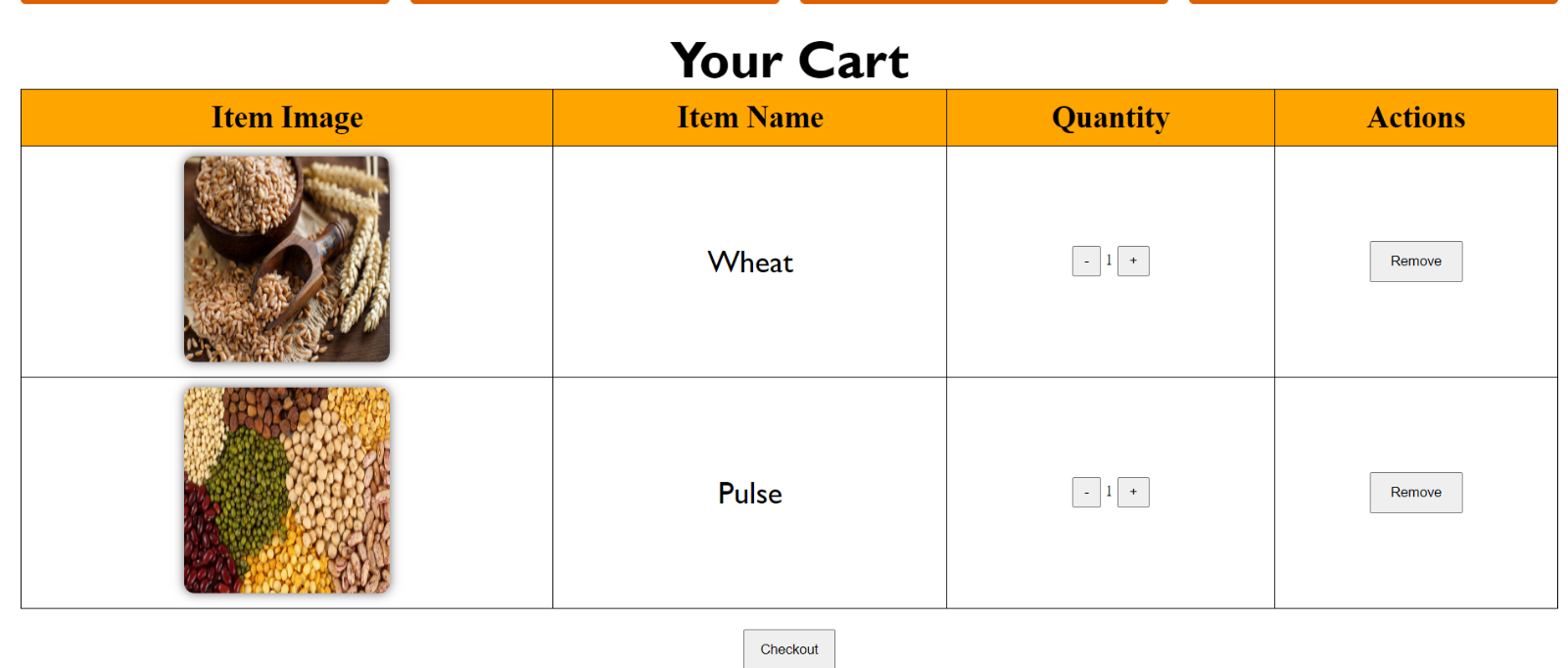
**4.8 Money Donation Page**

****

**4.9 Book Donation Page**

****

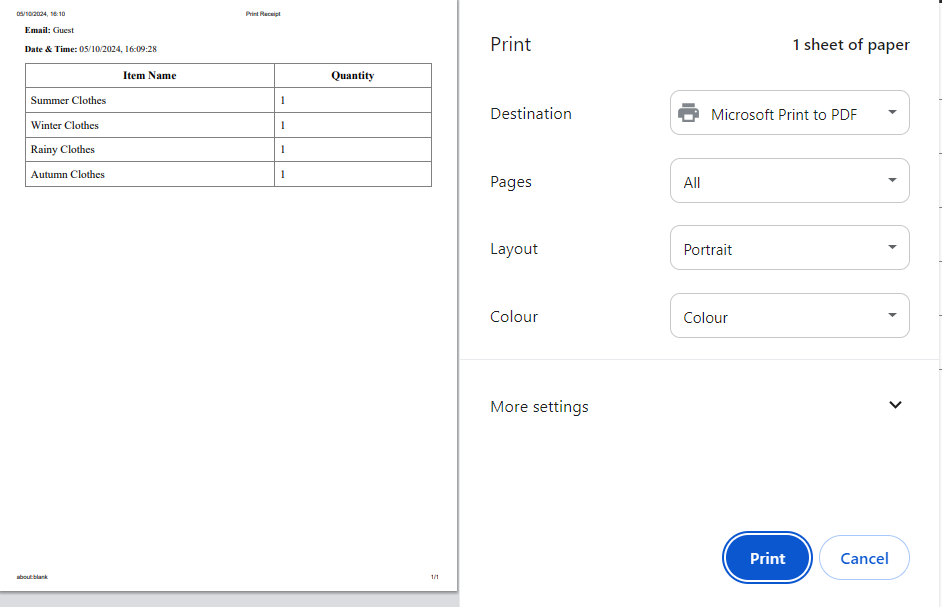
**4.10 Add On Cart Page**

****

**4.11** Receipt of Food Page

****

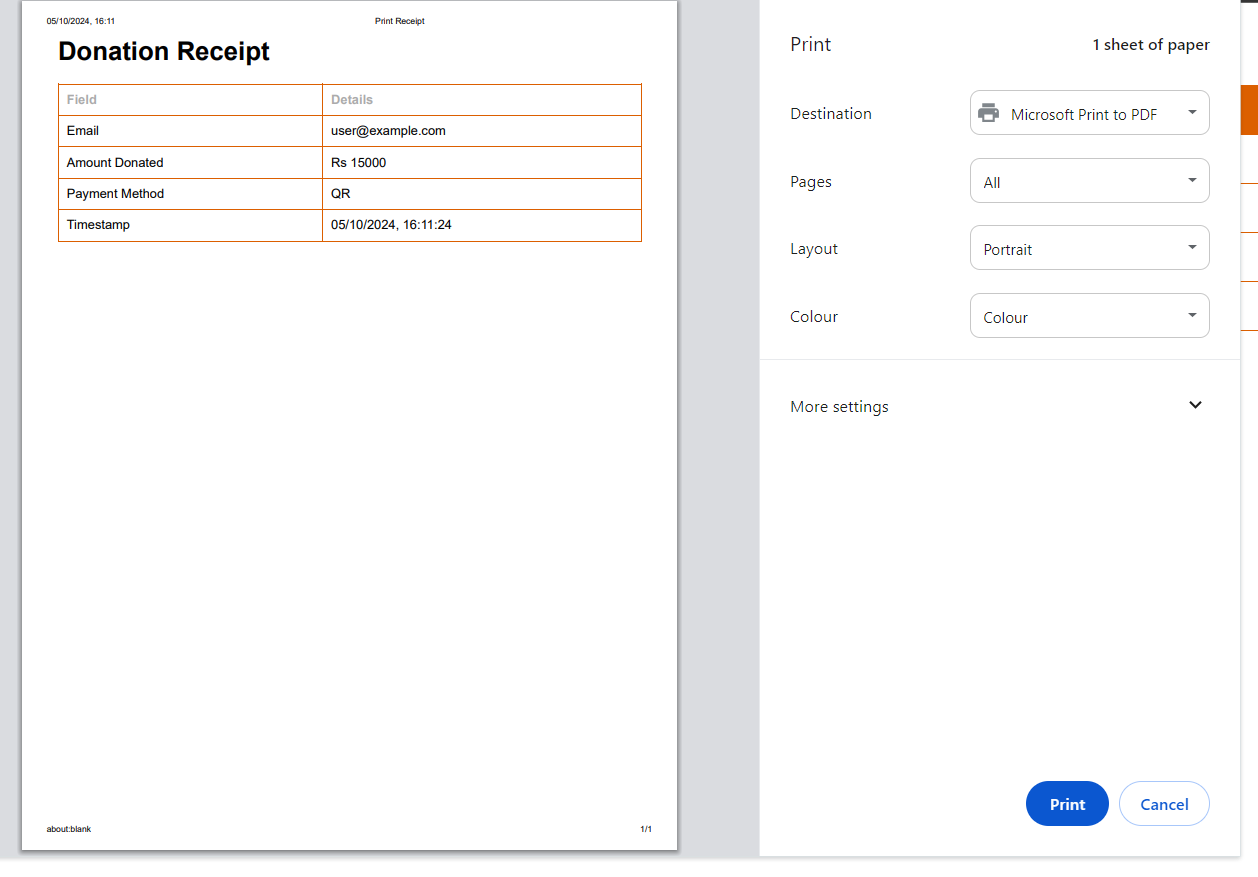
**4.12 Download Food Donation Recepit**

****

**4.13 Money Donation Receipt**

****

**4.14 Download Money Donation Receipt**

****

**Chapter No 5**

**Test Case**

Black Box Testing

|  |
| --- |
| **Test Case 1: Registration Page** |
| |  |  | | --- | --- | | **Test Case ID** | **TC001** | | **Test Case Name** | Registration with Valid Inputs | | **Description** | Test if a user can register with valid inputs. | | **Preconditions** | The user is on the registration page. | | **Test Steps** | 1. Navigate to the registration page.  2. Enter a valid username, email, donatingAs, and password.  3. Click on the "Register" button. | | **Expected Result** | User is registered successfully and redirected to the login page with a success message. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 2: Registration with Invalid Email** |
| |  |  | | --- | --- | | **Test Case ID** | **TC002** | | **Test Case Name** | Registration with Invalid Email | | **Description** | Test if the system prevents registration with an invalid email format. | | **Preconditions** | The user is on the registration page. | | **Test Steps** | 1.Navigate to the registration page.  2. Enter valid username, password, donatingAs but an invalid email format (e.g., "user@xyz").  3. Click on the "Register" button. | | **Expected Result** | The system shows a validation error stating the email format is invalid and does not allow registration. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 3: Registration with Already Used Email** |
| |  |  | | --- | --- | | **Test Case ID** | **TC003** | | **Test Case Name** | Registration with Already Used Email | | **Description** | Test if the system prevents registration with an email that is already in use. | | **Preconditions** | The user is on the registration page, and the email is already registered in the database. | | **Test Steps** | 1.Navigate to the registration page.  2. Enter valid username, donatingAs, and password, but use an email that is already registered.  3. Click on the "Register" button. | | **Expected Result** | The system shows a message stating "This email is used, try another one" and does not allow registration. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 4: Login Page with Correct Credentials** |
| |  |  | | --- | --- | | **Test Case ID** | **TC004** | | **Test Case Name** | Login with Valid Credentials | | **Description** | Test if a user can successfully log in with correct email and password. | | **Preconditions** | User is registered and on the login page. | | **Test Steps** | 1.Navigate to the login page.  2.Enter valid email and password.  3. Click on the "Login" button. | | **Expected Result** | User is logged in successfully and redirected to the home page. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| Test Case 5: Login with Incorrect Credentials |
| |  |  | | --- | --- | | **Test Case ID** | **TC005** | | **Test Case Name** | Login with Incorrect Credentials | | **Description** | Test if the system prevents login when an incorrect password or email is used. | | **Preconditions** | User is registered and on the login page. | | **Test Steps** | 1.Navigate to the login page.  2. Enter a valid email but an incorrect password.  3. Click on the "Login" button. | | **Expected Result** | The system shows a message "Wrong Username or Password" and the user is not logged in. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 6: Login with Unregistered Email** |
| |  |  | | --- | --- | | **Test Case ID** | **TC006** | | **Test Case Name** | Login with Unregistered Email | | **Description** | Test if the system prevents login when the email is not registered in the system. | | **Preconditions** | User is on the login page and has not registered yet. | | **Test Steps** | 1.Navigate to the login page.  2. Enter an unregistered email and any password.  3. Click on the "Login" button. | | **Expected Result** | The system shows a message "Wrong Username or Password" and the user is not logged in. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 7: Food Donation Page Access (Logged In)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC007** | | **Test Case Name** | Access Food Donation Page (Logged In) | | **Description** | Test if a logged-in user can access the food donation page. | | **Preconditions** | User is logged in and on the home page. | | **Test Steps** | 1. Log in successfully.  2. Navigate to the food donation page. | | **Expected Result** | The user is able to access the food donation page. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 8: Food Donation Page Access (Not Logged In)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC008** | | **Test Case Name** | Access Food Donation Page (Not Logged In) | | **Description** | Test if the system prevents access to the food donation page when the user is not logged in. | | **Preconditions** | User is not logged in. | | **Test Steps** | 1. Attempt to navigate to the food donation page without logging in. | | **Expected Result** | The system redirects the user to the login page with a message to log in first. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 9: Successful Food Donation Entry** |
| |  |  | | --- | --- | | **Test Case ID** | **TC009** | | **Test Case Name** | Successful Food Donation Entry | | **Description** | Test if a logged-in user can submit a valid food donation entry. | | **Preconditions** | User is logged in and on the food donation page. | | **Test Steps** | 1. Log in successfully.  2. Navigate to the food donation page.  3. Enter valid food donation details (e.g., food name, quantity, description).  4. Click the "Submit" button. | | **Expected Result** | The food donation entry is successfully added, and a confirmation message is displayed. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 10: Unsuccessful Food Donation Entry (Invalid Inputs)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC010** | | **Test Case Name** | Unsuccessful Food Donation Entry (Invalid Inputs) | | **Description** | Test if the system prevents submission of invalid or incomplete food donation entries. | | **Preconditions** | User is logged in and on the food donation page. | | **Test Steps** | 1.Loginsuccessfully.  2. Navigate to the food donation page.  3. Enter invalid details (e.g., leave required fields empty or input invalid data).  4. Click the "Submit" button. | | **Expected Result** | The system shows validation errors for the invalid fields, and the entry is not submitted. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 11: Add Item to Cart (Valid Item Selection)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC011** | | **Test Case Name** | Add Valid Item to Cart | | **Description** | Test if a user can successfully add a valid item to the cart. | | **Preconditions** | The user is logged in and on the food donation page or product list page. | | **Test Steps** | 1. Log in successfully. 2. Navigate to the item list. 3. Select a valid item. 4. Click the "Add to Cart" button. | | **Expected Result** | The item is successfully added to the cart, and a confirmation message is displayed. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 12: Add Item to Cart (Invalid Selection)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC012** | | **Test Case Name** | Add Invalid Item to Cart | | **Description** | Test if the system prevents adding an invalid or unavailable item to the cart. | | **Preconditions** | The user is logged in and on the food donation/product list page. | | **Test Steps** | 1. Log in successfully. 2. Navigate to the item list. 3. Select an invalid/unavailable item. 4. Attempt to click the "Add to Cart" button. | | **Expected Result** | The system shows a validation error, and the item is not added to the cart. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 13: View Cart and Checkout (Valid Cart)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC013** | | **Test Case Name** | View Cart and Checkout with Valid Items | | **Description** | Test if a user can view the cart and proceed to checkout with valid items. | | **Preconditions** | The user is logged in and has added valid items to the cart. | | **Test Steps** | 1. Add valid items to the cart. 2. Navigate to the "Cart" page. 3. Click "Checkout". | | **Expected Result** | The user can successfully view the cart and proceed to the checkout page. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 14: View Cart and Checkout (Empty Cart)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC014** | | **Test Case Name** | View Cart and Checkout with Empty Cart | | **Description** | Test if the system prevents checkout when the cart is empty. | | **Preconditions** | The user is logged in but has no items in the cart. | | **Test Steps** | 1. Log in successfully. 2. Navigate to the "Cart" page with an empty cart. 3. Attempt to click "Checkout". | | **Expected Result** | The system shows an error message stating "Cart is empty" and prevents the user from proceeding to checkout. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 15: Receipt Generation (Valid Order)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC015** | | **Test Case Name** | Generate Receipt for Valid Order | | **Description** | Test if the system generates a receipt after a successful checkout. | | **Preconditions** | The user has successfully added items to the cart and completed the checkout process. | | **Test Steps** | 1. Complete a valid checkout. 2. Confirm payment or order. 3. Check if the receipt is generated and available for download or print. | | **Expected Result** | The system generates a receipt with order details, total amount, and date, available for download or print. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 16: Receipt Generation (Invalid Order)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC016** | | **Test Case Name** | Attempt Receipt Generation with Invalid Order | | **Description** | Test if the system prevents receipt generation for invalid or incomplete orders. | | **Preconditions** | The user has not completed the checkout process or has an invalid cart. | | **Test Steps** | 1. Attempt to generate a receipt without completing the checkout. 2. Try to access the receipt section. | | **Expected Result** | The system shows an error message stating "No receipt available" and prevents receipt generation. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

|  |
| --- |
| **Test Case 17: View Previous Receipt (Order History)** |
| |  |  | | --- | --- | | **Test Case ID** | **TC017** | | **Test Case Name** | View Previous Receipt in Order History | | **Description** | Test if the user can view previously generated receipts in their order history. | | **Preconditions** | The user has successfully completed previous orders and logged in. | | **Test Steps** | 1. Log in successfully. 2. Navigate to the "Order History" or "Receipts" page. 3. View previous receipts. | | **Expected Result** | The user can view previous receipts with full details. | | **Actual Result** | (To be filled after execution) | | **Status** | (Pass/Fail) | |

**Chapter No 6**

**Conclusion**

The project "THE HOPE" successfully created a user-friendly website that helps connect donors with orphanages and NGOs. It makes it easy for people to donate money, books, or food from home, ensuring security and accessibility at any time. By improving the donation process, the platform aims to build a supportive community and encourage more people to give help.

**Recommendation**

To enhance "THE HOOP," focus on improving the website’s security with two-factor authentication and forming more partnerships with NGOs. Adding features like real-time donation tracking and a feedback system will boost transparency and user satisfaction. Automated reporting can help users and admins track activities easily. Since the website is mainly used on laptops, optimize it for desktop use, and integrate social media sharing to raise awareness and attract more donors.

**Chapter No 7**

**Bibliography & Reference**

** W3C. (n.d.). Cascading Style Sheets (CSS). Retrieved from** [**https://www.w3.org/Style/CSS/**](https://www.w3.org/Style/CSS/)

** Berners-Lee, T. (n.d.). HTML: Hypertext Markup Language. Retrieved from** [**https://www.w3.org/MarkUp/**](https://www.w3.org/MarkUp/)

** PHP Group. (n.d.). PHP: Hypertext Preprocessor. Retrieved from** [**https://www.php.net/**](https://www.php.net/)

** Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.**

** Sommerville, I. (2011). Software Engineering (9th ed.). Addison-Wesley**