HELPING HANDS

SOEN 6222

SAJANA BIDESI – ST10249843

SOEN 7311

DESIGN DOCUMENT

Sajana Bidesi
ST 10249843

Table of Contents

ARCHITECTURE PATTERN

1. What is an Architecture Pattern?

An architecture pattern is a widely recognized solution for addressing common software development challenges while promoting separation of concerns. It provides a blueprint for organizing code, defining component relationships, and managing data flow within an application (Interserver, 2016). These language-agnostic patterns can be applied across various programming environments, allowing developers to create well-structured, testable, and maintainable code, thereby streamlining the development process. (Sheldon, 2023)

In this project, I have chosen the MVC (Model-View-Controller) pattern to leverage these benefits. The MVC approach clearly separates data, user interface, and logic, facilitating modular development, easier testing, and improved scalability.

2. How Does an Architecture Pattern Work?

Architecture patterns segment an application into distinct components, each with a specific role. In the MVC pattern:
- Model: Manages data logic, including storage, retrieval, and business rules.
- View: Serves as the user interface, displaying data and capturing user interactions.
- Controller: Acts as an intermediary, processing user inputs from the View, updating the Model, and directing the data presentation back to the View.

These structured interactions create a clear division of responsibilities, enhancing modularity and allowing for independent development, code reuse, and simplified testing and debugging. (Sheldon, 2023)

https://i.sstatic.net/hchsD.png
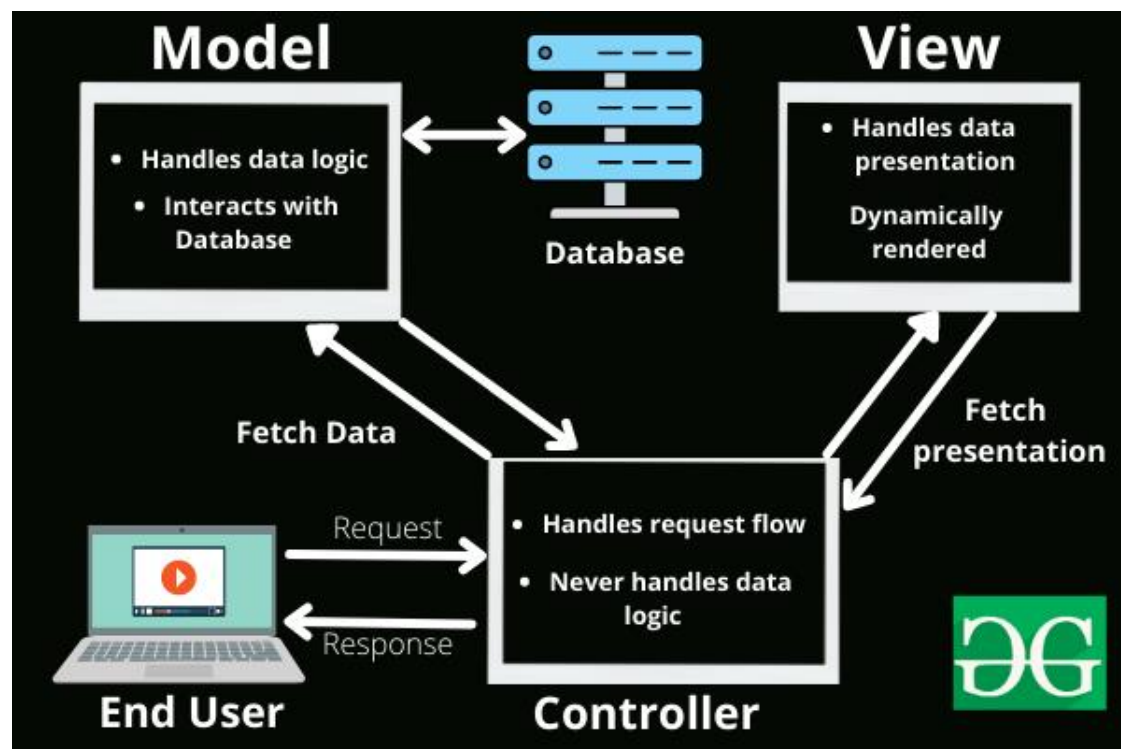
3. Why and How Would MVC Be Used for Helping Hands?
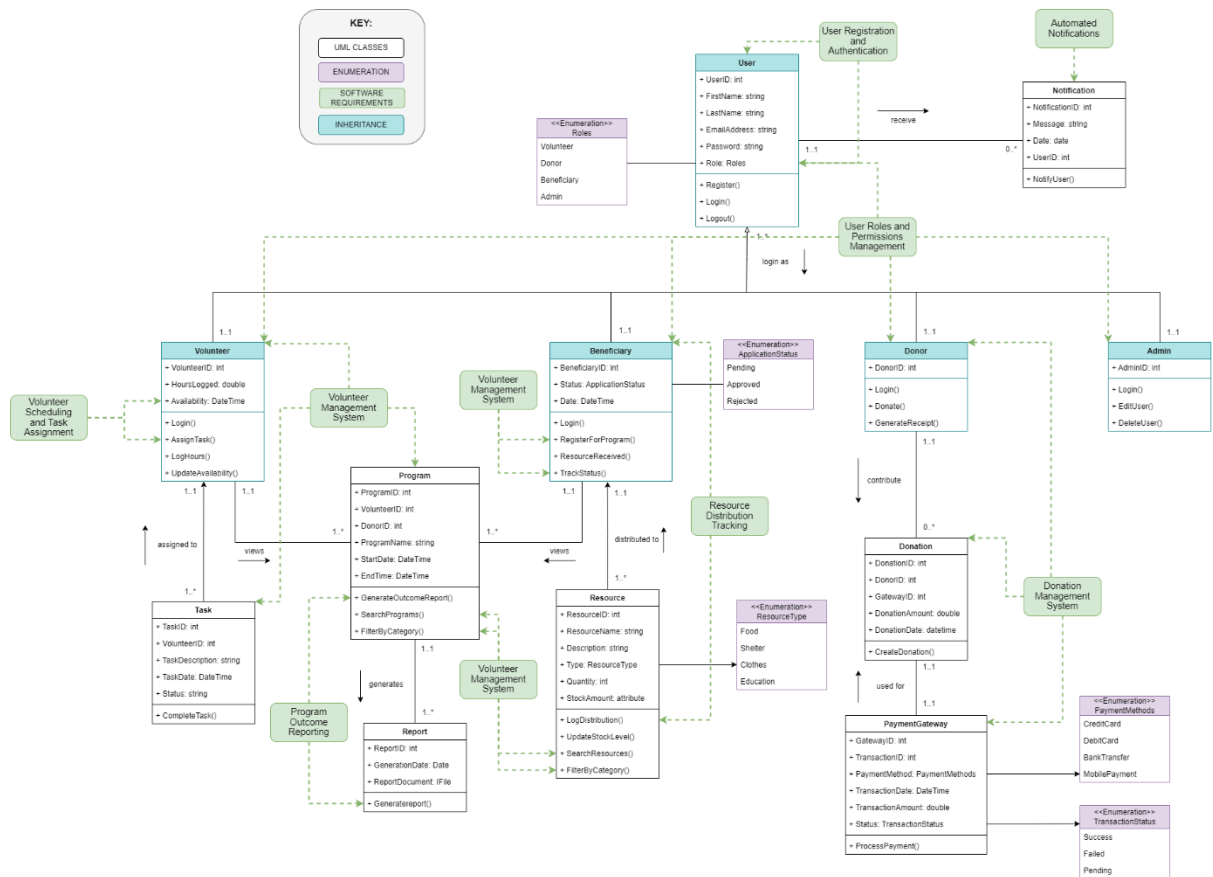
Implementing MVC for the Helping Hands project enhances organization and scalability in managing volunteers, donations, resources, tasks and beneficiaries. The Model Layer manages data logic and business rules for entities such as volunteers, donations, and beneficiaries, etc., often interfacing with a relational database for structured storage. The View Layer provides specific user interfaces for different roles, such as volunteers, donors, beneficiaries, and admin, allowing for personalized experiences without altering core logic. Meanwhile, the Controller Layer processes user inputs, interacts with the Model for data storage and retrieval, and manages workflows, including notifying users or assigning tasks based on availability. (Interserver, 2016)

Adopting the MVC architecture offers several benefits. It enables faster development through a clear separation of concerns (Interserver, 2016), allowing for more efficient collaboration among multiple developers working on different components. The structure also facilitates easier updates, as changes can be implemented with minimal disruption to the overall system. Additionally, the modular nature of MVC simplifies debugging, making it easier to isolate and fix issues.

However, there are some challenges associated with MVC. The architecture can be complex and may pose difficulties for new developers to understand. Additionally, adherence to strict rules and conventions can introduce a learning curve. Despite these minor drawbacks, the advantages of MVC significantly outweigh the disadvantages (Interserver, 2016). This architecture enables the Helping Hands system to evolve modularly, supporting new features like volunteer scheduling and donation tracking while aligning with agile methodologies for responsive and maintainable development.

UML DIAGRAM

KEY:
- UML CLASSES
- ENUMERATION
- SOFTWARE REQUIREMENTS
- INHERITANCE

User Registration and Authentication

Automated Notifications

User
- + UserID: int
- + FirstName: string
- + LastName: string
- + EmailAddress: string
- + Password: string
- + Role: Roles
- + Register()
- + Login()
- + Logout()

<<Enumeration>> Roles
- Volunteer
- Donor
- Beneficiary
- Admin

Notification
- + NotificationID: int
- + Message: string
- + Date: date
- + UserID: int
- + NotifyUser()

receive   1..1   0..*

User Roles and Permissions Management

login as

Volunteer
- + VolunteerID: int
- + HoursLogged: double
- + Availability: DateTime
- + Login()
- + AssignTask()
- + LogHours()
- + UpdateAvailability()

Volunteer Scheduling and Task Assignment

Volunteer Management System

Beneficiary
- + BeneficiaryID: int
- + Status: ApplicationStatus
- + Date: DateTime
- + Login()
- + RegisterForProgram()
- + ResourceReceived()
- + TrackStatus()

Volunteer Management System

<<Enumeration>> ApplicationStatus
- Pending
- Approved
- Rejected

Donor
- + DonorID: int
- + Login()
- + Donate()
- + GenerateReceipt()

Admin
- + AdminID: int
- + Login()
- + EditUser()
- + DeleteUser()

Program
- + ProgramID: int
- + VolunteerID: int
- + DonorID: int
- + ProgramName: string
- + StartDate: DateTime
- + EndTime: DateTime
- + GenerateOutcomeReport()
- + SearchPrograms()
- + FilterByCategory()

views   1..*   1..*   views

Resource Distribution Tracking

contribute

distributed to   1..*

Donation
- + DonationID: int
- + DonorID: int
- + GatewayID: int
- + DonationAmount: double
- + DonationDate: datetime
- + CreatesDonation()

Donation Management System

Task
- + TaskID: int
- + VolunteerID: int
- + TaskDescription: string
- + TaskDate: DateTime
- + Status: string
- + CompleteTask()

assigned to   1..1   1..*

Resource
- + ResourceID: int
- + ResourceName: string
- + Description: string
- + Type: ResourceType
- + Quantity: int
- + StockAmount: attribute
- + LogDistribution()
- + UpdateStockLevel()
- + SearchResources()
- + FilterByCategory()

<<Enumeration>> ResourceType
- Food
- Shelter
- Clothes
- Education

used for   1..1   1..1

Program Outcome Reporting

generates

Report
- + ReportID: int
- + GenerationDate: Date
- + ReportDocument: IFile
- + Generatereport()

Volunteer Management System

PaymentGateway
- + GatewayID: int
- + TransactionID: int
- + PaymentMethod: PaymentMethods
- + TransactionDate: DateTime
- + TransactionAmount: double
- + Status: TransactionStatus
- + ProcessPayment()

<<Enumeration>> PaymentMethods
- CreditCard
- DebitCard
- BankTransfer
- MobilePayment

<<Enumeration>> TransactionStatus
- Success
- Failed
- Pending

FUNCTIONAL (SOFTWARE) REQUIREMENTS

(AS PER PART 1 - AMMENDED WITH UML SOFTWARE REQUIREMENTS SATISFIED)

Functional Requirement	Requirement Name	UML INDICATION
User Registration and Authentication	The system will allow users (volunteers, staff, donors) to create accounts and log in securely.	- User class
Volunteer Management System	The system will allow admin to have control over volunteer accounts, with the ability to view, edit, or delete profiles.	- Volunteer class - Program class - Task class
Donation Management System	The system will allow admins to have full access to the profiles of donors, including the ability to view, edit, and delete donor details as needed.	- Donor class - Donation class - PaymentGateway class
Resource Distribution Tracking	The system will log and track the distribution of resources (food, clothing, etc.) to beneficiaries.	- Resource class - Beneficiary class
Program Outcome Reporting	The system will generate reports on program outcomes (e.g., families helped, volunteer hours logged).	- Program class - Report class
Volunteer Scheduling and Task Assignment	The system will assign tasks and schedules based on volunteer availability and display them on a dashboard.	- Volunteer, availability(attribute) - AssignTask(method)
Beneficiary Registration and Tracking	The system will allow beneficiaries to register for programs and track their application statuses.	- Beneficiary class
Automated Notifications for Volunteers and Donors	The system will send email and in-program notifications to volunteers (tasks) and donors (confirmation, thank you).	- Notification class
Search and Filter Functionality for Programs and Resources	The system will allow users to search and filter programs or resources by categories like location or type of support.	- Program class - Resource class
User Roles and Permissions Management	The system will control user access based on roles (admin, staff, volunteers, donors) to limit functionalities accordingly.	- User class - Admin class - Donor class - Beneficiary class - volunteer class

ANNEXTURE A

Although I received 100% for my POE Part 1 design document, I recognize the opportunity to make improvements and clarify any areas that may require revision or enhancement. One significant change I have decided to implement is in the donor and volunteer management system. Initially, my design allowed users to enter their details for donating to specific causes or for volunteers to provide their availability. However, after further consideration, I have decided to modify this functionality so that the admin of the system has full control over the volunteer and donor accounts. This includes the ability to view, edit, or delete user profiles, ensuring that administrators can manage the system more effectively and ensure data accuracy.

Additionally, I have reworked the notification system. Originally, it was designed to send notifications only via email. However, I realized that integrating an in-app notification system would enhance user experience. Now, users will receive notifications directly within the web application. These notifications will cover a range of activities, such as confirmation of successful donations, booking confirmations for volunteers, and updates like newsletters. This change will improve user engagement by providing real-time updates without relying solely on email, offering a more seamless and interactive experience for both donors and volunteers.

(All changes were made in page 5 of this document)

SOEN 7311

REQUIREMENT
DOCUMENT

Sajana Bidesi

INTRODUCTION

"Helping Hands" is a charity that provides support to individuals and families by giving them basic needs like food, clothes, residence, and education. The present manual processes, through the use of spreadsheets and paper-based systems, are quite ineffective and prone to mistakes, thus complicating the task of tracking the distribution of resources, the supervision of program efficiency and the involvement of two parties: the volunteers and the beneficiaries. To tackle these problems, a web-based software solution will be designed.

This software will simply be used to automate the program management, to manage the distribution of resources, to coordinate volunteers, and to communicate. This will improve the operation by reducing the time of work, provide real-time tracking, and make transparency easier during the management of the organization's activities. The system will also give Helping Hands the chance to involve the community better through improved communication and collaboration.

The aim of this project is the development and efficient running of a website that helps control programs, volunteers, beneficiaries, donations, and resource distribution. The site will enable both an administrator and a user to the directives that they will deliver.

The Agile methodology is a flexible approach that develops software in small, iterative cycles. For Helping Hands, this means the web-based solution will be built in phases, focusing on resource tracking, volunteer management, and donation handling. The project is divided into short sprints with regular stakeholder feedback, allowing continuous adjustments to meet real needs. Agile enables Helping Hands to respond quickly to new demands, improving efficiency by automating manual processes. By involving staff and volunteers throughout development, the system ensures it solves real problems, making Helping Hands more effective in serving individuals and families.

Scrum, as an Agile methodology, is particularly well-suited for Helping Hands. Its iterative process delivers small features in short sprints, focusing on specific functions like resource tracking, volunteer coordination, and donation management. With regular feedback and active involvement from stakeholders, Scrum ensures the system remains aligned with the charity's evolving needs. Daily standups and sprint reviews promote ongoing improvement, while each sprint delivers usable features that improve efficiency and transparency, allowing Helping Hands to better manage resources and respond quickly to changing demands. (Veritis, n.d.)

FUNCTIONAL REQUIREMENTS

Functional Requirement	Requirement Name	Testable	Feasible
User Registration and Authentication	The system will allow users (volunteers, staff, donors) to create accounts and log in securely.	Test registration form, password strength, error handling, and login success.	Implement with OAuth or JWT protocols. (Mukhadin Beschokov, 2024)
Volunteer Management System	The system will allow admin to have control over volunteer accounts, with the ability to view, edit, or delete profiles.	Test the admin's ability to view, edit, and delete volunteer profiles.	The Volunteer Management System is feasible by implementing role-based access control (RBAC), allowing the admin to securely view, edit, and delete volunteer profiles through a relational database and user interface.
Donation Management System	The system will allow admins to have full access to the profiles of donors, including the ability to view, edit, and delete donor details as needed.	Test the admin's ability to view, edit, and delete donor profiles, as well as manage donation tracking, receipt generation, and payment processing.	This is feasible as the admin can access donor profiles, view donation history, manage payment details, and track transactions through the system's backend, ensuring smooth donation processing and account management.
Resource Distribution Tracking	The system will log and track the distribution of resources (food, clothing, etc.) to beneficiaries.	Test resource logging, stock level updates, and distribution reports.	Implement using inventory management systems and databases. (Volha Belakurska, 2024)
Program Outcome Reporting	The system will generate reports on program outcomes (e.g., families helped, volunteer hours logged).	Test report generation accuracy based on program types and date filters.	Use SQL Server reporting tools or Business Intelligence (BI) tools for data visualization. (Atlassian, 2019)
Volunteer Scheduling and Task Assignment	The system will assign tasks and schedules based on volunteer availability and display them on a dashboard.	Test schedule updates, conflict management, and notification systems.	Use scheduling libraries and integrated calendars for task management.
Beneficiary Registration and Tracking	The system will allow beneficiaries to register for programs and track their application statuses.	Test application process, status updates, and notifications.	Implement using user profiles and automated workflows for program registration.
Automated Notifications for Volunteers and Donors	The system will send email and in-program notifications to volunteers (tasks) and donors (confirmation, thank you).	Test email notification triggers for different scenarios (e.g., donation, task assignment).	Integrate for communication services.
Search and Filter Functionality for Programs and Resources	The system will allow users to search and filter programs or resources by categories like location or type of support.	Test search queries for accuracy and filters for correct refinement.	Implement using database queries and search tools for fast performance.
User Roles and Permissions Management	The system will control user access based on roles (admin, staff, volunteers, donors) to limit functionalities accordingly.	Test role-based access to restricted areas and permissions for each user type.	Implement using Role-Based Access Control (RBAC) via middleware or authentication libraries. (Microsoft, 2022)

EXTERNAL INTERFACE REQUIREMENTS
External System: Payment Gateway

1. System Overview
The payment gateway plays a crucial role in managing online transactions for the Helping Hands website, enabling secure processing of donations through a variety of methods including credit and debit cards, mobile payments, and bank transfers. By integrating with payment gateways such as PayFast, Zapper, PayU, and Yoco, the website can offer flexible and reliable payment options to users, ensuring that all donations are handled securely and efficiently. (Stripe, n.d.)

2. Integration Requirements
API Integration
To manage payment transactions effectively, integration with the payment gateway's API is vital. This involves sending payment requests, handling the processing of transactions, and receiving responses from the gateway. The system must guarantee secure transmission of payment information and provide real-time updates on transaction statuses, as well as handle payment confirmations efficiently to ensure smooth and accurate processing. (Dantzer, 2024)

Data Security
Ensuring data security is essential to protect sensitive payment information. Payment data should be encrypted during transmission using SSL/TLS (Digicert, 2023) to safeguard communication between the website and the payment gateway. Additionally, implementing tokenization is important for securely managing sensitive payment data, thus minimizing the risk of exposing credit card information and enhancing overall security.

Transaction Management
Effective transaction management requires the payment gateway to support real-time processing, which offers immediate feedback on donation statuses and keeps records current. The system must also handle payment errors appropriately, providing users with clear guidance and information if any transaction issues arise. This ensures users are well-informed and supported throughout their payment experience. (Dantzer, 2024)

Reporting and Reconciliation
The website should record successful transactions in its database, capturing details such as donor information, donation amount, and transaction ID. The system must also support reconciliation by comparing transaction records with payment gateway reports to ensure accuracy and address any discrepancies. This helps maintain precise financial records and resolves any issues efficiently.

3. Testability
- Payment Methods: Test various payment methods supported by the gateway (e.g., credit/debit cards, mobile payments) to confirm transactions are processed correctly and efficiently.
- Error Scenarios: Simulate different error conditions (e.g., declined transactions, network issues) to ensure the system handles errors appropriately and provides useful feedback.
- Security: Perform security testing to verify that encryption (Digicert, 2023)and tokenization are effectively implemented, and that payment data is well-protected.

4. Feasibility

The feasibility of integrating a payment gateway into the Helping Hands website is supported by the availability of comprehensive APIs from various payment providers, such as PayFast, Zapper, PayU, and Yoco. These payment gateways offer well-documented APIs, which simplify the integration process by providing clear guidelines and technical details necessary for implementation. The thorough documentation ensures that developers have the resources needed to understand and utilize the APIs effectively, reducing the complexity of integration (Dantzer, 2024). Additionally, these payment gateways provide robust support to assist developers with any issues that may arise during the integration process, further enhancing the feasibility of incorporating these solutions into the website. This support helps to ensure that the integration is both efficient and effective, allowing the Helping Hands website to manage donations seamlessly and securely.

Non-Functional Requirements

Performance

The system must be designed to support up to 1,000 concurrent users while ensuring optimal performance. It is crucial that page load times consistently stay under 3 seconds to provide a responsive and efficient user experience, especially during peak traffic times. (Rome, 2020)

Security

To protect sensitive data, the system must use strong encryption protocols like SSL/TLS for data transmission. Additionally, user information storage must adhere to data protection laws, including the Protection of Personal Information Act (POPIA) in South Africa (POPIA, 2021). Following POPIA is vital for safeguarding personal data and ensuring compliance with local data protection regulations. (Rome, 2020)

Usability

The system should have a user-friendly interface that makes navigation and task completion easy for volunteers, donors, and beneficiaries. Good usability is essential so that users with different technical skills can interact with the system effectively and with minimal hassle. (Rome, 2020)

Availability

The system must maintain a minimum availability rate of 99.9%, ensuring it is almost always accessible. Scheduled maintenance should be communicated to users ahead of time to reduce disruptions and maintain user trust and satisfaction. (Rome, 2020)

Scalability

The system's design should allow for scalability, making it easy to add users, resources, and programs as the organization grows. This ability to scale ensures that the system can meet increasing demands and growth without sacrificing performance. (Rome, 2020)

ANNEXTURE B

The changes I made focused on enhancing the management systems for both volunteers and donors, as well as improving the notification system. For the Volunteer Management System, I updated the functionality to allow the admin to have full control over volunteer accounts, enabling them to view, edit, or delete profiles. This was implemented through role-based access control (RBAC), ensuring secure and efficient management via a relational database and user interface. Similarly, for the Donation Management System, I expanded the admin's access to donor profiles, including the ability to view, edit, and delete donor details. I also integrated features for donation tracking, receipt generation, and payment processing, ensuring smooth management of donor accounts and transactions. Additionally, I enhanced the notification system by incorporating both email and in-app notifications for volunteers and donors. This includes notifications for task assignments, donation confirmations, and thank-you messages, improving communication and engagement with users through both email and real-time updates within the web application.

(changes were made on page 9 of this document)

SOEN 7311

UI/UX DESIGN

Sajana Bidesi
ST 10249843

UI/UX DESIGN

UI (User Interface) and UX (User Experience) in Helping Hands:

UI (User Interface) and UX (User Experience) are integral to how users interact with the Helping Hands platform. Together, they shape the visual appeal, usability, and effectiveness of the web application, ensuring that it serves its purpose of connecting donors, volunteers, and administrators in a way that is both engaging and efficient. (CareerFoundry, 2021)

UI (User Interface):

The UI is the visual aspect of the platform. It encompasses the layout, design elements, and interactive components that users see and interact with. Good UI design ensures that all elements on the screen are intuitively placed and visually consistent, enabling users to easily navigate the system. Here's how UI is applied in Helping Hands:

1. Minimalistic Design:
   The homepage and other key pages adopt a clean, minimalist design. This allows users to focus on the essential elements, such as donation options, volunteer sign-up, and resource tracking, without being overwhelmed by unnecessary clutter. (CareerFoundry, 2021; Figma, 2023)

1. Clear Navigation:
   The interface offers a menu page that acts as a central hub, where users can access different sections (such as donations, volunteer management, and resource tracking). The buttons and links are clearly labelled, making it easy for users to find what they are looking for. (CareerFoundry, 2021)

2. Engaging Visuals:
   The home page features an impactful featured image that symbolizes the platform's mission. Visuals like icons, buttons, and imagery representing various causes (such as images of the organizations in the donation section) help emotionally connect users with the platform's goals. (CareerFoundry, 2021; Figma, 2023)

3. Interactive Components:
   The donation page, for example, features a split-screen layout, where donors can view the project's details on one side and enter payment information on the other. The input fields are styled like a bank card, making it familiar and intuitive for users who are completing transactions. (CareerFoundry, 2021; Figma, 2023)

UX (User Experience):

The UX is the overall experience of the user when interacting with the platform. It focuses on how users feel when using the application, ensuring they can navigate the system easily, accomplish tasks quickly, and have a positive overall experience. (CareerFoundry, 2021; Figma, 2023)Here's how UX is implemented in Helping Hands:

1. Intuitive Onboarding:
    The registration page has a simple, guided form where users provide their name, email, password, and role (either volunteer or donor). The registration process is quick and ensures that users are onboarded smoothly, without unnecessary friction. If users forget their password, the password recovery system is simple and easy to use, ensuring that they can quickly regain access to their account. (CareerFoundry, 2021; Figma, 2023)

2. Role-Based Access:
    The platform's functionality is tailored to the user's role (volunteer, donor, or admin). After logging in, users are taken to a personalized dashboard where they can access the features that are most relevant to them. For example, donors can browse and contribute to projects, while volunteers can sign up for causes, they want to help with. (Figma, 2023)

3. Search and Filter:
    In the donation section, donors can search for causes using filters like name, location, or keywords. This enables users to easily find projects that align with their interests, making the platform more user-centric and personalized. (CareerFoundry, 2021; Figma, 2023)

4. Real-Time Data and Feedback:
    The resource tracking page provides real-time data on resource distributions, showing stock levels and distribution details in a dynamic table. Volunteers can filter through the data based on various criteria (such as status or stock level). Additionally, they can visualize this data with interactive charts, such as pie charts and bar graphs. This ensures that users have quick access to important information in an easy-to-understand format, improving their ability to make informed decisions. (CareerFoundry, 2021; Figma, 2023)

5. Seamless Donation Process:
    The donation page offers predefined amounts for donations but also allows donors to input their own amount. The process is simple, guiding users from selecting a donation amount to entering payment information, all the way to the donation confirmation. The use of a "pay" button to complete the transaction ensures a seamless, straightforward experience, without any unnecessary steps. (Figma, 2023)

6. Automated Notifications:
    Users are kept informed through automated notifications that provide feedback on actions, such as successful donations or new updates on the causes they support. These notifications appear both on the platform and via email, ensuring that users are continuously engaged and informed. Volunteers also receive timely alerts when stock levels are low, helping them to act proactively and restock before shortages occur. (CareerFoundry, 2021)

7. Security and Trust:
   UX also includes the trust and security features of the platform. The login page is designed with security in mind, incorporating password encryption, account locking after failed login attempts, and third-party authentication options (such as Google or GitHub login). These measures ensure that users feel safe while using the platform, creating a sense of confidence in their interactions. (CareerFoundry, 2021; Figma, 2023)

8. Feedback and Communication:
   The notifications system also extends to volunteer and donor interactions, such as receiving a confirmation message after donating or receiving updates via email about future causes. This helps foster a sense of community and keeps users engaged with the platform. (CareerFoundry, 2021)

Both UI and UX are essential in ensuring that Helping Hands is not only visually appealing but also functional, user-friendly, and efficient. The UI ensures a clean, responsive, and visually engaging interface that facilitates easy navigation, while the UX ensures that users have a seamless and enjoyable experience as they interact with the platform. Together, they create a positive, effective environment for donors, volunteers, and administrators to collaborate, manage resources, and contribute to meaningful causes. (CareerFoundry, 2021)

HOME PAGE
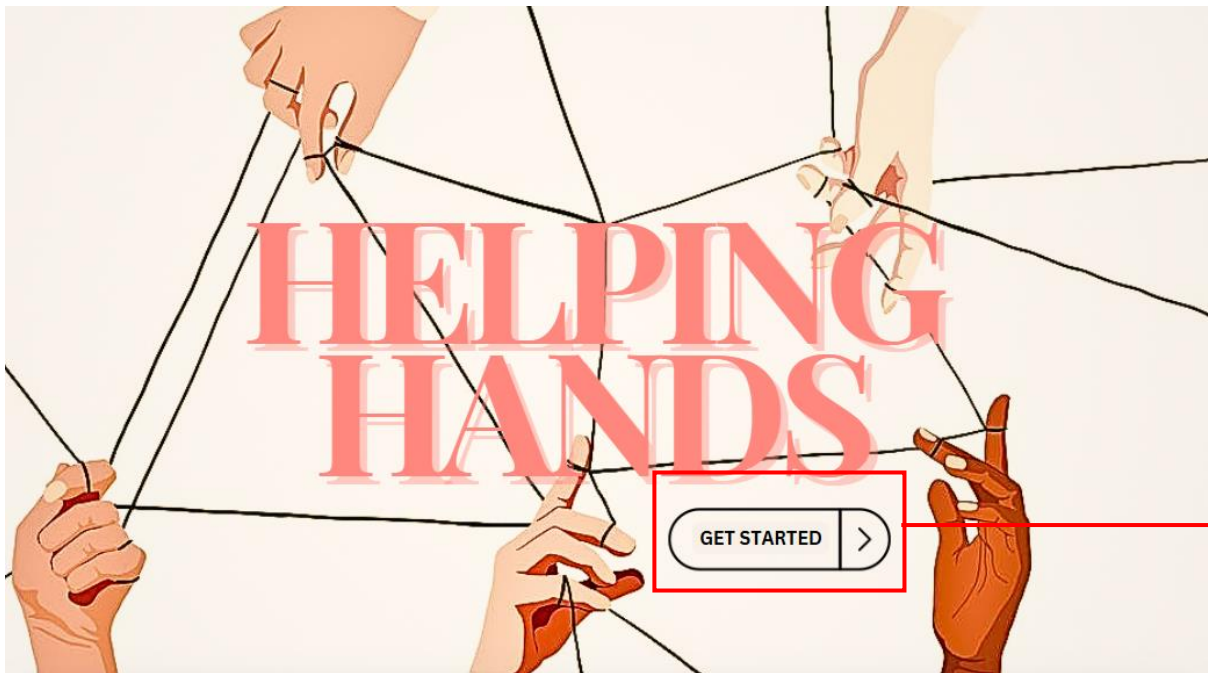
This is the startup or home page, serving as the welcoming screen for **all users, including volunteers, donors, and administrators**, to the Helping Hands platform. The featured image symbolizes hands connected by strings, reflecting the organization's collaborative efforts to spread love and care. This imagery represents how the organization works together to gather donations, provide assistance, and support those in need, with the shared goal of making the world a better place.

How to use this page?

Users can simply click the "Get Started" button to navigate to the menu page. While this page offers minimal interactivity, it is designed to be visually appealing and warmly inviting, effectively capturing the essence of Helping Hands and its mission to foster collective efforts in helping others.
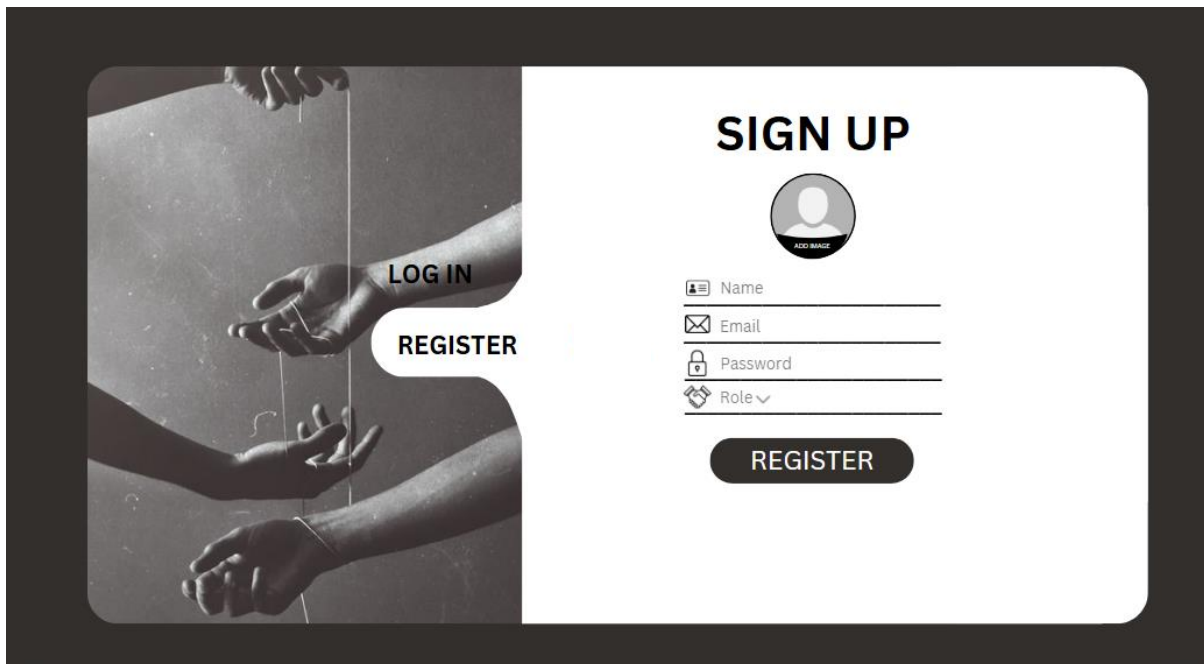
MENU PAGE
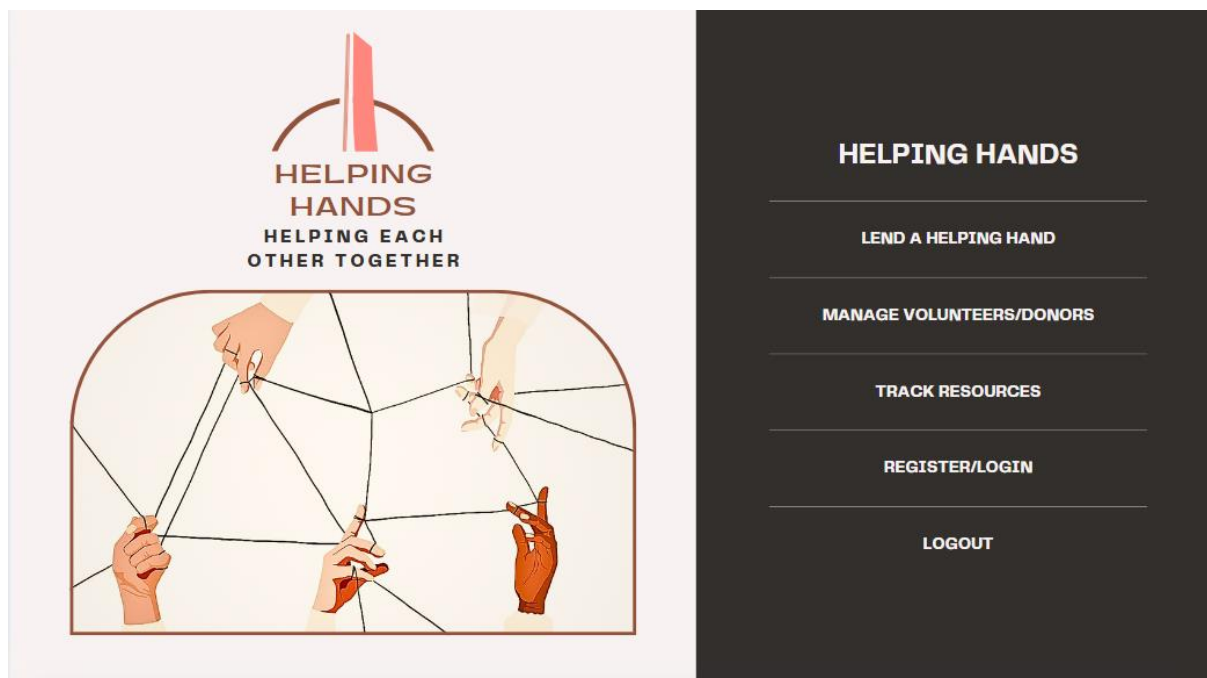
This menu page helps users navigate to the sections they need, offering the following functions as illustrated:

- **Lend a Helping Hand:** Donors can explore ongoing projects and contribute funds to specific organizations. Volunteers can sign up to assist these organizations by dedicating their time and efforts.
- **Manage Volunteers/Donors:** Accessible only to administrators, this page allows them to view a list of all volunteers and donors, edit user details, or delete profiles if necessary.
- **Track Resources:** Exclusively available to volunteers and administrators, this page provides an overview of all resources, details on beneficiaries receiving assistance, the amount distributed, and stock levels.
- **Register and Login:** Open to all users, this section facilitates registration and login, requiring users to provide their details and specify their role.
- **Logout:** Allows the current user to exit the system.

REGISTRATION PAGE

This page is accessible to all users and serves as the starting point for anyone looking to create an account to join the Helping Hands platform, either as a volunteer or a donor. This page satisfies the software requirement of "User Registration and Authentication" by offering essential functionality to ensure secure access.

To register, users need to fill out a simple form requiring the following details:
- **Name:** The user's full name.
- **Email:** A valid email address that will be used for communication and login purposes.
- **Password:** A secure password that meets specific criteria for strength and security, such as including uppercase letters, numbers, and special characters.
- **Role Selection:** Users must indicate their desired role—either as a donor or a volunteer.
- **Profile Picture:** Users must upload a profile picture of themselves. This will be displayed on their profile and helps personalize their account, ensuring authenticity and a sense of community.

Once registered as a donor, users gain immediate access to functions like donating to ongoing projects. For users aspiring to be volunteers, they can apply during registration or later by updating their profile. However, their request must go through an approval process. Administrators will review their application and, upon acceptance, update their access rights to enable volunteer-specific features.

This streamlined registration system ensures that users are securely onboarded, with a personalized and professional profile, and assigned the appropriate access based on their role while maintaining a smooth user experience.

LOGIN PAGE
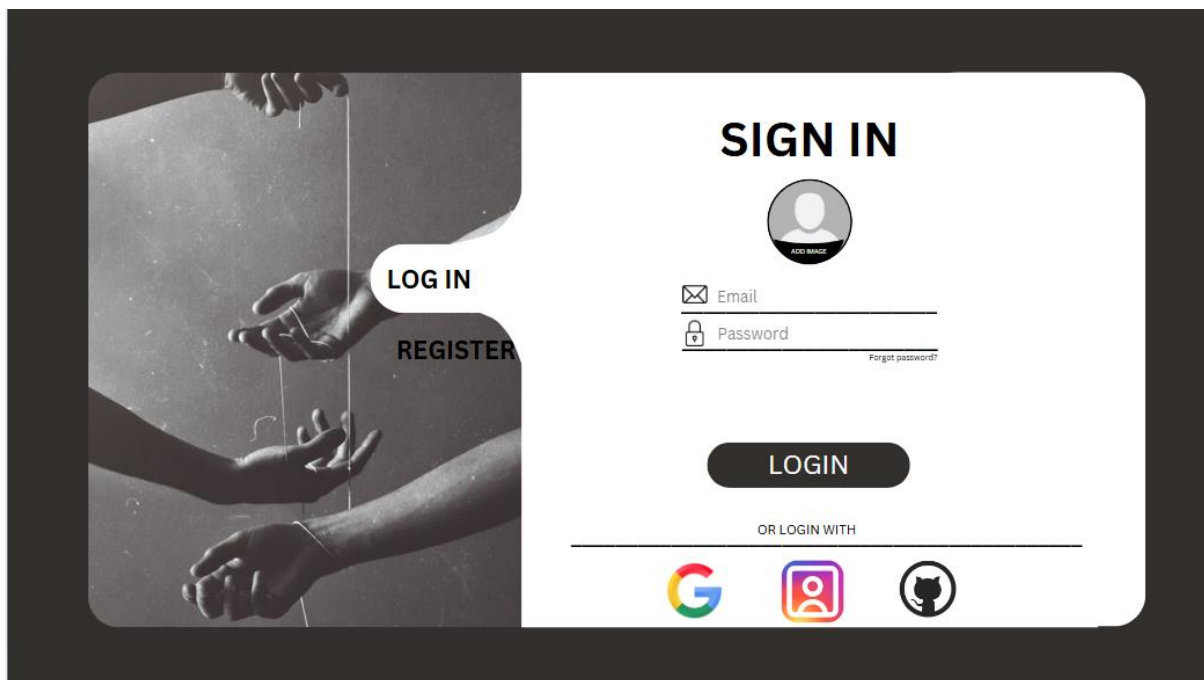
This is the login page, accessible to all users, designed to provide a seamless and secure entry point to the Helping Hands platform. The page is simple and user-friendly, ensuring a hassle-free login experience. This page satisfies the requirements of "User Registration and Authentication" by offering essential functionality to ensure secure access.

To log in, users must enter their email address and password. The system will validate these credentials against registered accounts. If the credentials are correct, the user will successfully log in and gain access to their respective dashboard and features based on their assigned role. If the credentials are incorrect or the account is not registered, the user will receive an error message prompting them to retry or register.

Additional Features:

Forgot Password Option

If a user forgets their password, they can click the "Forgot Password" button. This will initiate a password recovery process, sending a secure email to the registered address. The email will contain a link or instructions enabling the user to reset their password by creating a new one. This ensures both convenience and security in recovering account access.

Social Media Login

Users can alternatively log in using their Instagram, Google, or GitHub accounts, provided their email address is linked to one of these platforms. This feature streamlines the login process by allowing users to bypass manual credential entry, leveraging trusted third-party authentication for quick and efficient access.

Security Measures

To ensure account safety, the page incorporates measures such as:
- Email and password encryption during verification.
- Locking the account after multiple failed login attempts to prevent unauthorized access.
- Timed sessions requiring re-login after extended inactivity.

By integrating password recovery, third-party login options, and robust security practices, this page not only ensures secure authentication but also provides a modern, user-friendly experience that meets the platform's authentication requirements.

LEND A HELPING HAND PAGE
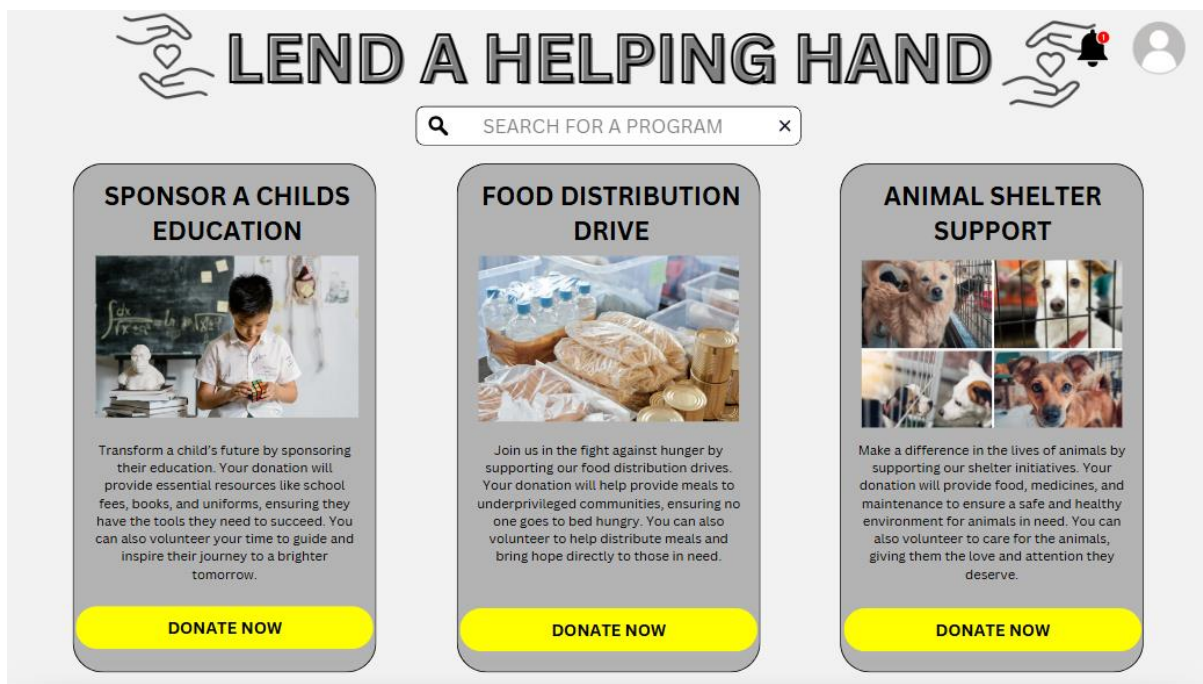
This page is designed exclusively for donors, offering them a comprehensive platform to explore and support meaningful causes. It features a curated list of ongoing projects that require funding, with each project displaying key details:

- **Organization Name:** Identifying the group behind the initiative.
- **Image:** A visual representation to help donors connect emotionally with the cause.
- **Detailed Description:** An in-depth explanation outlining the project's goals, purpose, and expected impact. This ensures donors are well-informed about where their contributions will go.

Features:

1. Search and Filter Options:

    Donors can use a search bar to locate projects by name, location, or keywords, ensuring a personalized experience. This feature satisfies the requirement of "Search and Filter for Programs and Resources" by enabling donors to easily navigate through available projects to find causes that align with their values and interests.

2. Donate Button:

    Each project includes a clear "Donate" button that directs users to a secure donation page where they can contribute an amount of their choice.

3. User-Friendly Interface:

    The page has an intuitive design, making it easy to browse and select projects while providing relevant details in an organized format.

This page ensures donors can engage with projects effectively, supporting causes they care about while fostering trust and transparency in the donation process.

DONATION PAGE

This is the Donation Page, where donors can securely contribute to causes they care about. The page provides an easy-to-use interface for making donations, offering both predefined donation amounts and card payment options. This page satisfies the software requirements of "Donation Management System" by allowing users to efficiently manage their donations, securely process payments, and understand the impact of their contributions.

Key Features:

1. Donation Amount Selection

Donors can choose from predefined amounts: 50, 100, 200, 500, 1000, or 2000. These options cater to different budgets and make the donation process simple and quick.

2. Split-Screen Layout

The page has a split-screen design:
- Left Section: Displays the organization's name and a detailed description of the cause, helping donors understand where their money will go.
- Right Section: Includes payment fields, formatted like a bank card, where donors enter their name, card number, expiry date, and CVV code for secure payment processing.

3. "Pay" Button

After entering their payment details, donors can click the "Pay" button to complete the donation.

This page streamlines the donation process, offering a transparent and secure experience, making it a core component of the Donation Management System.

DONATION PAGE – NOTIFICATION SYSTEM

The functionality of the donation page is consistent here, with the addition of a notification system to update users on the status of their donations. When a payment is successfully processed, a notification badge will appear. Clicking on this badge will expand the message, confirming the donation's success. The notification includes the messages:

1. **Donation Success Confirmation:** A message verifying that the donation was successfully processed, providing reassurance to the donor that their contribution has been received and will be used for the intended cause.

2. **Profile and Email Notifications:** Along with the on-screen notification, the donor will also receive a message on their profile and an email. This message contains:
   - A newsletter with updates on the cause and information about future projects.
   - Links to other organizations they may want to donate to, encouraging them to engage with additional causes.

These notifications are part of a comprehensive system aimed at keeping both volunteers and donors informed. As a result, this feature satisfies the requirement of "automated notifications for volunteers and donors", ensuring donors receive confirmation of their donation and ongoing communication. Volunteers will also receive similar newsletters and updates, keeping them engaged with the platform and its initiatives.

VOLUNTEER AND DONOR MANAGEMENT SYSTEM
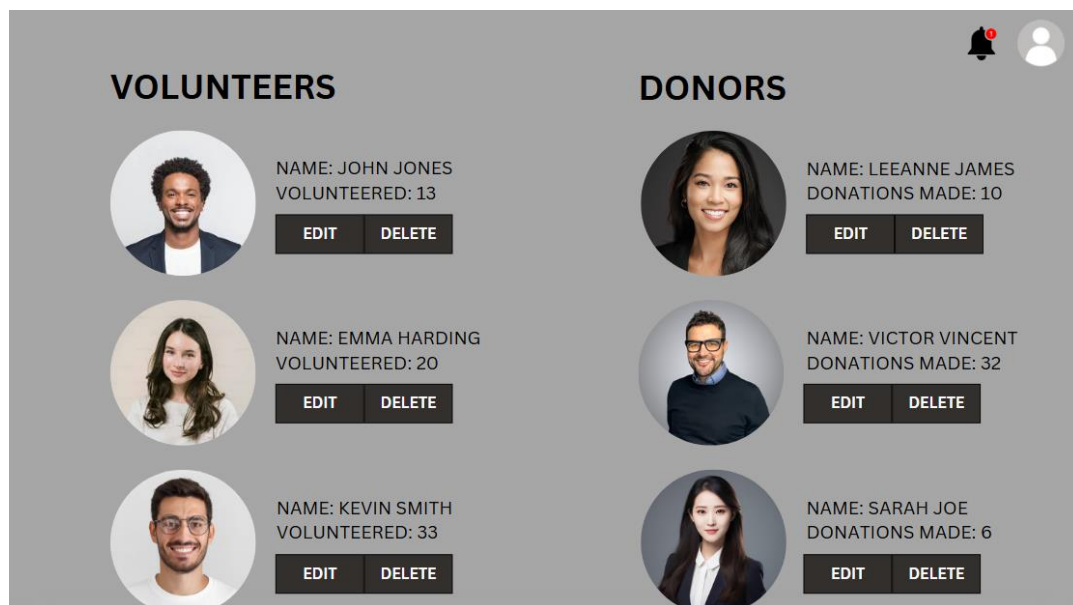
This page is dedicated to the Volunteer and Donor Management System and is exclusively accessible to admins. It plays a crucial role in managing user accounts and satisfies the requirements for a "Donor Management System", "Volunteer Management System", and "User Roles and Permissions Management". Only admins have the authority to edit or delete user profiles, ensuring that user access and data are securely controlled.

Key Features:

1. <u>User Profiles</u>
The page displays the profiles of all registered volunteers and donors. Each profile includes important details such as name, email, role (volunteer or donor), and other relevant information for the admin to review.

2. <u>Profile Viewing and Management</u>
Admins can easily view each user's profile, helping them track activities, contributions, and volunteer applications.

3. <u>Editing Profiles</u>
Admins have the ability to edit user profiles, allowing them to update personal information, modify roles, or adjust details as necessary to ensure the system is accurate and up to date.

4. <u>Deleting Profiles</u>
If needed, admins can delete user profiles, whether due to inactivity, user requests, or policy issues, maintaining a clean and secure system.

5. <u>Permissions and Access Control</u>
Only admins have access to this page, ensuring that regular users (volunteers and donors) cannot alter any profiles. This helps preserve system integrity and user privacy.

This page is integral to managing volunteers and donors, while also maintaining strict control over user permissions and data security.

RESOURCE TRACKING
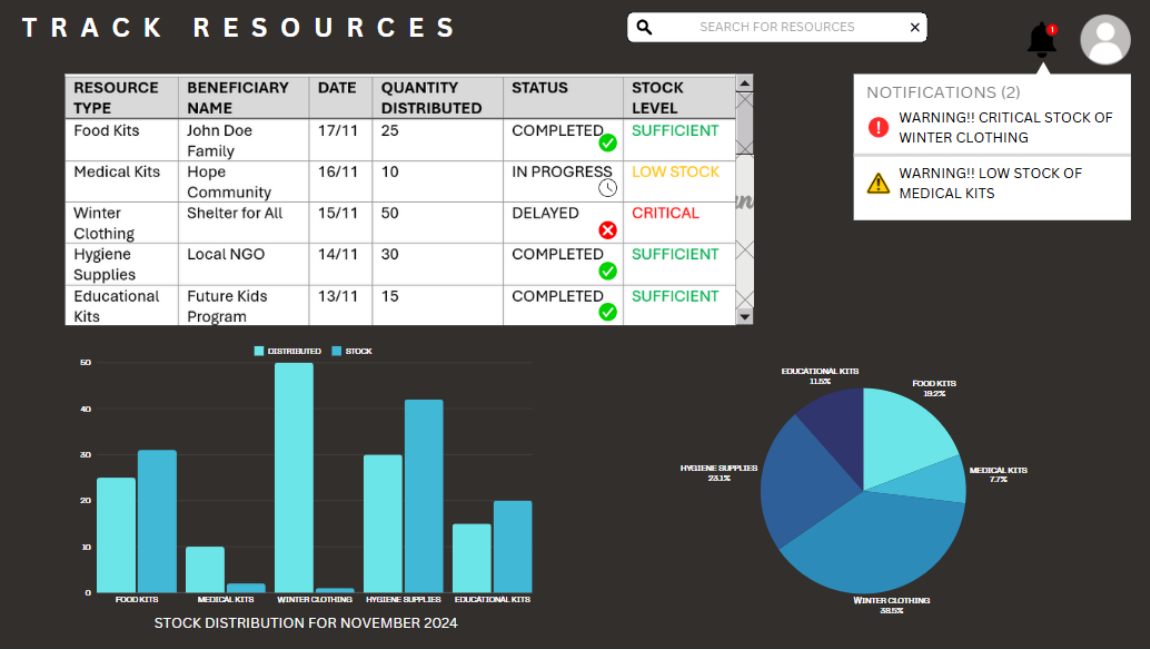
This page manages resource tracking for volunteers, enabling them to monitor where resources have been sent and to which beneficiaries. It fulfils the software requirement of "resource distribution tracking" by providing a detailed overview of all distributed resources and their status.

Key Features:
1. Resource Table
   Volunteers can access a detailed table showing the resource name, beneficiary name, date distributed, quantity distributed, status of distribution, and current stock level. This allows volunteers to quickly view each distribution and manage resources efficiently.

2. Filtering and Searching
   The table can be filtered using search terms like specific resources or beneficiaries. Volunteers can also sort by status or stock level to identify low-stock items that need restocking, making resource management more responsive.

3. Dynamic Visualization
   - Pie Chart: A dynamic pie chart provides a visual representation of current stock levels, helping volunteers quickly understand resource availability.
   - Double Bar Graph: This graph shows the distribution and stock levels of resources per month, allowing volunteers to track trends and manage resources proactively.

4. Notification System
   Volunteers receive notifications when stock levels are low. If a resource drops below a threshold, a warning is sent, ensuring timely restocking and preventing shortages.

This page ensures effective resource management by providing real-time data, visualizations, and notifications, helping volunteers make informed decisions and maintain smooth operations.

SELF EVALUATION

RUBRIC

POE SUBTOTAL				/100
RUBRIC POE — Self-Evaluation	**Levels of Achievement**			**Score**
	Excellent	**Good**	**Developing**	
	Score Ranges Per Level (½ marks possible)			
Criteria (3)	**3** • Bravo! You have done exceptionally well!	**2** You are on the right track, but you can do better!	**0 to 1** You have learned something – but you are not proving it!	3
Reflection (3)	**3** • Reflection shows thorough thoughtfulness. • Reflection has several supporting details and examples. • All parts of the reflection are complete and done.	**2** Reflection shows little thoughtfulness. Reflection has few details or examples. Most parts of the reflection are incomplete.	**0 to 1** • Reflection shows no thoughtfulness. • Reflection has no details. • Reflection is incomplete.	3
Demonstration of Learning (4)	**4** • Clearly explains what was learned. • Reflection is beyond simple description of event/experience to an analysis of how it contributed to learning and understanding.	**2 to 3** The reflection demonstrates student's attempt to analyse the event/experience but fails to demonstrate depth of analysis.	**0 to 1** • Reflection does not move beyond description of the event/experience.	4
POE – Self Evaluation TOTAL				/ 10

SELF REFLECTION

Reflecting on my experience with this software engineering module, I can confidently say that I have gained a deep understanding of various architectural patterns and methodologies, which have greatly enhanced my approach to developing applications. One of the most significant learnings was about MVC (Model-View-Controller), which I found to be the most suitable architecture for the Helping Hands platform. The separation of concerns in MVC made it easier to structure the application, ensuring scalability and maintainability. I found that designing the prototype using tools like Figma and Canva was a much smoother process since I could focus on the UI/UX without worrying about the backend at that stage. However, I am eager to dive into building a fully functional web application in the future, leveraging the MVC architecture and integrating the backend with payment gateways and API functionalities.

Throughout this module, I gained invaluable hands-on experience, especially when creating a UML diagram using Enum for the first time. This task required thorough research to ensure proper formatting and syntax, which enhanced my skills in understanding system structures. I also enjoyed specifying the functional requirements—documenting these in tables made the information much clearer and easier to understand, which I found rewarding. The process of learning and applying Agile methodologies helped me see the value of iterative development cycles when building applications. Moreover, integrating APIs and considering how they would work with the payment gateway functionality opened my eyes to the complexities of system interactions and how they drive user functionality.

The most enjoyable part, however, was working on the UI/UX section. Bringing the application prototype to life and explaining how each screen aligned with the functional requirements gave me a sense of fulfilment and completion. This module has not only equipped me with technical skills but also ignited my passion for software development, and I look forward to applying what I've learned in future projects, particularly in building a functional web application for Helping Hands.

REFERENCES

Atlassian (2019). *Top 10 BI Tools for Data Visualization & Analysis*. [online] Atlassian. Available at: https://www.atlassian.com/data/business-intelligence/10-data-visualization-tools#:~:text=Google%20Data%20Studio [Accessed 13 Sep. 2024].

CareerFoundry (2021). UX vs. UI Design: What's the Difference? [2024 Guide]. [online] CareerFoundry. Available at: https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/#ui-ux-salary [Accessed 18 Nov. 2024].

Dantzer, J. (2024). *Payment API explained: Connecting a payment gateway*. [online] Checkout.com. Available at: https://www.checkout.com/blog/online-payment-api-explained#:~:text=a%20payment%20API%3F- [Accessed 13 Sep. 2024].

Digicert (2023). *What is SSL, TLS and HTTPS? | DigiCert*. [online] www.digicert.com. Available at: https://www.digicert.com/what-is-ssl-tls-and-https.

Figma (2023). UI Design vs UX Design: What's the Difference? | Figma [2023 Guide]. [online] Figma. Available at: https://www.figma.com/resource-library/difference-between-ui-and-ux/.

Interserver (2016). What Is MVC? Advantages and Disadvantages of MVC - Interserver Tips. [online] Interserver Tips. Available at: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/.

Lacroix, B. (2022). *The Best Payment Gateways in South Africa*. [online] Portmoni. Available at: https://portmoni.com/best-payment-gateway-south-africa/.

Microsoft (2022). *What is Azure role-based access control (Azure RBAC)?* [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/azure/role-based-access-control/overview.

Mukhadin Beschokov (2024). *OAuth vs JWT*. [online] Wallarm.com. Available at: https://www.wallarm.com/what/oauth-vs-jwt-detailed-comparison#:~:text=JWT%20token%20vs%20oauth%20token.

POPIA (2021). *Protection of Personal Information Act (POPI Act)*. [online] POPIA. Available at: https://popia.co.za/.

Rome, P. (2020). *What are Non Functional Requirements — With Examples*. [online] Perforce Software. Available at: https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples.

Sheldon, R. (2023). What Is model-view-controller (MVC)? | Definition from TechTarget. [online] WhatIs.com. Available at: https://www.techtarget.com/whatis/definition/model-view-controller-MVC.

Stripe (n.d.). *What is a payment gateway? | Stripe*. [online] stripe.com. Available at: https://stripe.com/resources/more/payment-gateways-101#:~:text=A%20payment%20gateway%20is%20a.

Veritis (n.d.). 7 Popular Types of Agile Methodologies. [online] Available at: https://www.veritis.com/blog/7-important-types-of-agile-methodologies/.

Volha Belakurska (2024). *Inventory Management Database: Understanding Inventory Database and Inventory Management System*. [online] Resources. Available at: https://synder.com/blog/inventory-management-database/#:~:text=An%20inventory%20management%20database%20is [Accessed 13 Sep. 2024].