Ehsas-Hub

By:

Zain Muneer

35937

Abdullah shahid

35438

Hamza Ahmed

31967

Supervised by:

Tajamul Shahzad

Faculty of Computing

Riphah International University, Islamabad

Fall 2024

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the Degree of

Bachelors of Science in Computer Science

Faculty of Computing

Riphah International University, Islamabad

Date: December, 2024

Final Approval

This is to certify that we have read the report submitted by ***Zain Muneer 35937 Abdullah Shahid 35438 Hamza Ahmed 31967*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSCS).

Committee:

1

_____

Tajamul Shahzad
(Supervisor)

2

_____

Dr. Musharraf Ahmed
(Head of Department)

Declaration

We hereby declare that this document "**Ehsas hub**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor **Tajumal Shahzad,** if any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

Zain Muneer

35937

Abdullah Shahid

35438

Hamza Ahmed

31967

Dedication

Our project is dedicated to our parents, seniors, friends, and our supervisor "**Tajamul Shahzad**" who has been our continual source of inspiration and whose support has helped this project succeed. This project would not have been possible without their love and support.

Acknowledgement

First of all, we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We are deeply grateful to our friends who helped us along the way, our families for their support, and our supervisor, **Tajamul Shahzad**, for his direction.

Zain Muneer

35937

Abdullah Shahid

35438

Hamza Ahmed

31967

Abstract

Ehsas Hub is a community-driven platform to connect donors, students, and volunteers toward a common cause: that is, making education accessible to the needy. Unlike most platforms focusing on money, Ehsas Hub is more interested in book-sharing. By doing this, it pairs each donated book with a student who actually needs the book on what they are interested in and what they aim to do in the future. Ehsas Hub, through smart technology, makes book recommendations to each learner to learn and grow, system takes student interest like (Favorite book, author, genres) and provide top rated books. It ensures that all that is done is open and honest so that trust may be built. This doesn't only get the right resources to the right students but empowers them to reach their full potential.

Table of Contents

List of Tables

List of Figures

Chapter 01:

Introduction

Chapter 1: Introduction

Many students today who don't have much money and access to basic learning materials like books, which are important for both learning and personal growth. On the other hand, many people and groups are ready to donate books but don't know how to get in touch with people who need them. Ehsas Hub, a digital platform that makes it easy for donors, students, and volunteers to meet, can fill this gap. The main goal of this project is to make it easier for people to donate books and make sure that they get to the right people by using a personalized recommendation system that is based on academic interests, and preferences.

Ehsas Hub is more than just a place to donate; it's a step towards making education available to everyone. The platform improves the process of matching given books with people who can use them by adding a recommendation system. This way, every book donated has the chance to improve someone's education. The platform uses technology to get around the problems that come with traditional book donation methods, like matching people with the right books.

1.1 Goals and Objectives

1.1.1 Goals:

1. Bridge the Gap Between Donors and Needy:
2. Empower Volunteer Coordination:
3. Promote Transparency and Reliability:
4. Leverage Technology for Social Good:

1.1.2 Objectives:

1. Develop a Comprehensive Donation and Volunteer Management System:
2. Provide Intelligent Recommendations for Book Donations:
3. Streamline Volunteer Engagement and Task Management:
4. Ensure Platform Adaptability and User-Centric Growth:

1.2 Scope of the Project

- **Students:** Students from low-income families, who live in orphanages or underprivileged areas for their studies, need educational materials.
- **Donors:** People or groups eager to contribute books for the good of the community.
- **Volunteers:** Those who want to help with logistics, such picking up and distributing books, in order to support the cause.

1.3 Functional Features:

- **User Registration and Authentication:** encompass secure login, multi-factor authentication, and the definition of user roles, including student, donor, and volunteer.
- **Profile Management:** Customizable profiles for students (academic interests), donors (donation listings), and volunteers (availability and locations) are provided.
- **Recommendation System:** Proposes books, articles, and novels aligned with students' academic interests and the highest-rated books and authors.
- **Donation management enables the facilitation of book donations**: Offering options for categorization and presentation in a searchable catalog.
- **Request System:** Facilitates the process for students and institutions to request particular books or genres, aligned with available donations.

1.4 Conclusion:

With the use of technology, Ehsas-Hub hopes to improve book donations by boosting efficiency, equity, and personalization. The idea aims to provide a smart and organized platform so that no student will be denied the opportunity to learn because they cannot afford books.

Chapter 02:

Literature Review

Chapter 2: Literature Review

2.1 Introduction

In many areas, like e-commerce, entertainment, and education, recommendation systems are an important part of giving each user a personalized experience. This literature review is mostly about book recommendation systems, which try to match users with good books based on their likes, dislikes, and past actions. This chapter goes into definitions, linked research, and an analysis of methodologies. It then looks for research gaps and comes up with the Ehsas=Hub project's problem statement with an emphasis on book recommendation platforms; this chapter provides a thorough analysis of recommendation systems. It examines fundamental ideas, current studies, approaches used in comparable systems, and highlights important gaps pertinent to the Ehsas-Hub project.

2.2 Background and Problem Elaboration

Book recommendation systems have evolved from simple content-based methods to sophisticated hybrid approaches. The challenges addressed by these systems include handling vast datasets, improving recommendation accuracy, and overcoming issues like cold-start problems and sparsity in user feedback. For Ehsas Hub, the aim is to integrate a recommendation engine specifically tailored to students' interests and academic goals, leveraging techniques like collaborative filtering and machine learning.2.3 Detailed Literature Review

2.3 Detailed Literature Review

2.3.1 Definitions

- **Content-Based Filtering**: Recommends items similar to those the user has liked based on item attributes (e.g., genre, author).
- **Collaborative Filtering**: Makes recommendations by finding similarities among users or items based on user ratings or interactions.
- **Hybrid Systems**: Combines content-based and collaborative methods to overcome the limitations of each technique.

2.3.2 Related Research Work 1

A study by Gupta et al. (2020) explores the effectiveness of recommendation systems in e-commerce and library platforms. The research highlights the utility of content-based filtering for user-specific recommendations and discusses its limitation in handling new users (cold-start problem). Collaborative filtering, though powerful, requires extensive datasets to deliver accurate predictions.

2.3.3 Related Research Work 2

A personalized book recommendation system developed by Sarma et al. (2021) combines clustering techniques with cosine similarity to recommend books. The study uses datasets from Goodreads and applies machine learning models to improve recommendation accuracy. It effectively addresses sparsity and cold-start problems through clustering methods**.**

2.4 Literature Review Summary Table

Table 2.1: Literature Review Summary Table

STUDY	**Methodology**	STRENGTHS	**Limitations**
GUPTA ET AL. (2020)	Content-Based Filtering	Personalized recommendations	Struggles with cold-start problems
SARMA ET AL. (2021)	Clustering + Collaborative Filtering	High accuracy and handles sparsity well	Requires well-curated datasets
RAJPURKAR ET AL. (2015)	Hybrid (Content + Collaborative)	Improves recommendation relevance	Computationally intensive for large datasets

2.5 Research Gap

Existing systems largely focus on generic book recommendations and often fail to align with specific user goals, such as academic interests. Moreover, while hybrid systems improve accuracy, they introduce higher computational complexity. There is a lack of scalable solutions

tailored to nonprofit platforms like Ehsas Hub, which serve diverse user bases including students and donors.

2.6 Problem Statement

The challenge is to design a scalable and efficient book recommendation system for Ehsas Hub that:

1. Personalizes recommendations based on user interests, academic goals, and ratings.
2. Effectively addresses cold-start problems and data sparsity.
3. Operates within the constraints of a nonprofit organization serving varied stakeholders.

2.7 Conclusion

In conclusion, while numerous techniques exist for book recommendations, few are tailored for socially-driven platforms. The findings of this review validate the need for a hybrid, efficient, and student-centered system as proposed in Ehsas Hub

Chapter 03:

Requirements and Design

Chapter 3: Requirements and Design

3.1 Introduction:

In this chapter, we have developed our functional requirements for our actors i.e. (**Needy**, **Donor, Admin** and **Volunteer**). The requirements are designed for especially for Ehsas-Hub platform.

Ehsas-Hub is a web-based platform designed to connect or interact with Needy and Donors easily with each other with help of volunteer.

The platform is user-friendly, easy to navigate and search, and provide a convenient and efficient way for both parties to connect and interact with each other.

We created our system **use cases** against each functional requirement and created use case diagrams, fully dressed use cases for our actors i.e. (User, Admin, Donor and Volunteer).

3.2 Requirements

3.2.1 Functional Requirements

Needy:

Table 3.1: Functional Requirements of Needy

ID	REQUIREMENTS
FR-1.1	User shall be able to Sign Up.
FR-1.2	User shall be able to login to their account.
FR-1.3	User shall be able to Forget/Recover their password.
FR-1.4	.
FR-1.5	User shall be able to view profile
FR-1.6	User shall be able to edit/update their profile.
FR-1.7	User shall be able to View Books Based on Recommendation with respect to their interest.
FR-1.8	User shall be able to request specific books.

FR-1.9	User shall be able to add book to favorite .
FR-1.10	User shall be able to See favorite books.
FR-1.11	User shall be able to view book stats.
FR-1.12	User shall be able to view requested books list.
FR-1.13	User shall be able to search books.
FR-1.14	User shall be able to give feedback.
FR-1.15	User shall be able to Logout

Donor:

Table 3.2: Functional Requirements of Donor

ID	REQUIREMENTS
FR-2.1	Donor shall be able to Sign Up.
FR-2.2	Donor shall be able to Login.
FR-2.3	Donor shall be able to forget/recover their Password.
FR-2.4	Donor shall be to view profile.
FR-2.5	Donor shall be able to edit profile.
FR-2.6	Donor shall be able to donate books.
FR-2.7	Donor shall be able to view donated book stats.
FR-2.8	Donor shall be able to view donated book list.
FR-2.9	Donor shall be able to give feedback.
FR-2.10	Donor shall be able to Logout.

Volunteer:

Table 3.3: Functional Requirements of Volunteer

ID	REQUIREMENTS
FR-3.1	Volunteer shall be able to sign up.
FR-3.2	Volunteer shall be able to log in.
FR-3.3	Volunteer shall be able to forget/recover password.
FR-3.4	Volunteer shall be able to view profile.
FR-3.5	Volunteer shall be able to edit profile.
FR-3.6	Volunteer shall be able to check new request.
FR-3.7	Volunteer shall be able to accept request.
FR-3.8	Volunteer shall be able to check request in process.
FR-3.9	Volunteer shall be able to view completed request.
FR-3.10	Volunteer shall be able to logout.

Admin:

Table 3.4: Functional Requirements of Admin

ID	REQUIREMENTS

FR-4.1	Admin shall be able to sign up.
FR-4.2	Admin shall be able to login.
FR-4.3	Admin shall be able to forget/recover password.
FR-4.4	Admin shall be able to view profile.
FR-4.5	Admin shall be able to edit profile.
FR-4.6	Admin shall be able to manage accounts.
FR-4.7	Admin shall be able to approve account.
FR-4.8	Admin shall be able to reject account.
FR-4.9	Admin shall be able to freeze account.
FR-4.10	Admin shall be able to active account.
FR-4.11	Admin shall be able to view donor request.
FR-4.12	Admin shall be able to accept request.
FR-4.13	Admin shall be able to reject request.
FR-4.14	Admin shall be able to view approved request.
FR-4.14	Admin shall be able to view in process request.
FR-4.16	Admin shall be able to view completed request.
FR-4.17	Admin shall be able to view needy request.
FR-4.18	Admin shall be able to accept needy request.

FR-4.19	Admin shall be able to reject needy request.
FR-4.20	Admin shall be able to view needy approved request.
FR-4.21	Admin shall be able to view needy in process request
FR-4.22	Admin shall be able to view needy completed request.
FR-4.23	Admin shall be able to view volunteer request.
FR-4.24	Admin shall be able to view volunteer approved request.
FR-4.25	Admin shall be able to view volunteer completed request.
FR-4.26	Admin shall be able to view account statistics.
FR-4.27	Admin shall be able to view active list.
FR-4.28	Admin should be able to active book.
FR-4.29	Admin should be able to deactivate book.
FR-4.30	Admin should be able to view feedbacks.
FR-4.31	Admin should be able to log out.

3.2.2 Non-Functional Requirements

Reliability

- The system should handle errors gracefully and should not lose data during Donations.
- The platform must be able to recover quickly from unexpected failures Like passwords etc.

Security

- Strong authentication (JWT) and password encryption should be implemented to ensure that user data is secure.
- Sensitive data, such as user information and donation details, should be encrypted both during transmission and storage.

Usability

- The platform should have an intuitive and easy-to-use interface for all user roles (admin, donor, volunteer, needy).
- It should be accessible on both desktop and mobile devices, providing a responsive design for different screen sizes.

Maintainability

- The system should be modular, with clearly defined components that can be easily updated or replaced in the future.
- Proper documentation and error logs should be maintained to help with ongoing support and updates.

Compatibility

- The platform should work on common web browsers like **Chrome**, **Firefox**, and **Safari, Web Browsers.**
- It should be compatible with both **Windows** and **Ubuntu Linux** operating systems.

Resource Efficiency

- The system should operate efficiently without excessive consumption of memory or CPU, ensuring smooth operation on typical hardware.

3.2.3 Hardware and Software Requirements

Hardware Requirements:

Server: Dedicated or cloud-based server with at least 16GB RAM and 500GB SSD.
Storage: Sufficient storage for books metadata, user data, and logs.
Processing Power: Capable of handling concurrent user requests and machine learning tasks.

Software Requirements:

Operating System: Windows Server.
Database: MySQL for storing user profiles, book details, and donation records.
Frontend: React.js for building the user interface.
Backend: Node.js with Express.js for server-side logic.
Recommendation System: Google Collab.

3.3 Proposed Methodology

The project will follow the **agile methodology**, focusing on iterative development and user feedback:

- **Requirement Gathering**: Identify user needs and define functionalities.
- **System Design**: Develop architecture and UI mockups.
- **Development**: Build core modules, including registration, Account management, donation management, and recommendation system.
- **Testing**: Validate functionality, performance, and security.

3.4 System Architecture

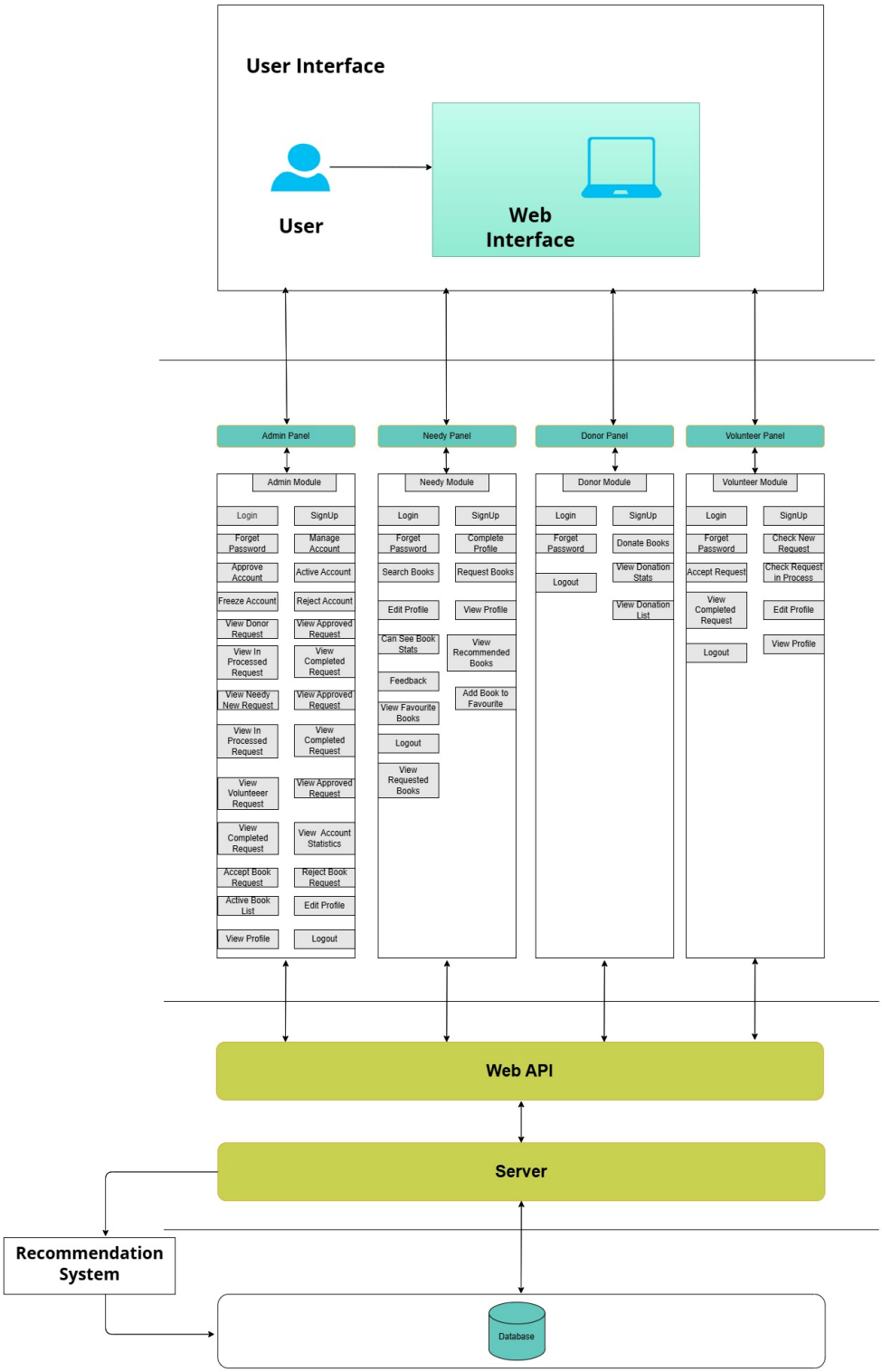


Figure 3.1: System Architecture

3.5 Use Cases

3.5.1 Admin Use-Case Diagram:

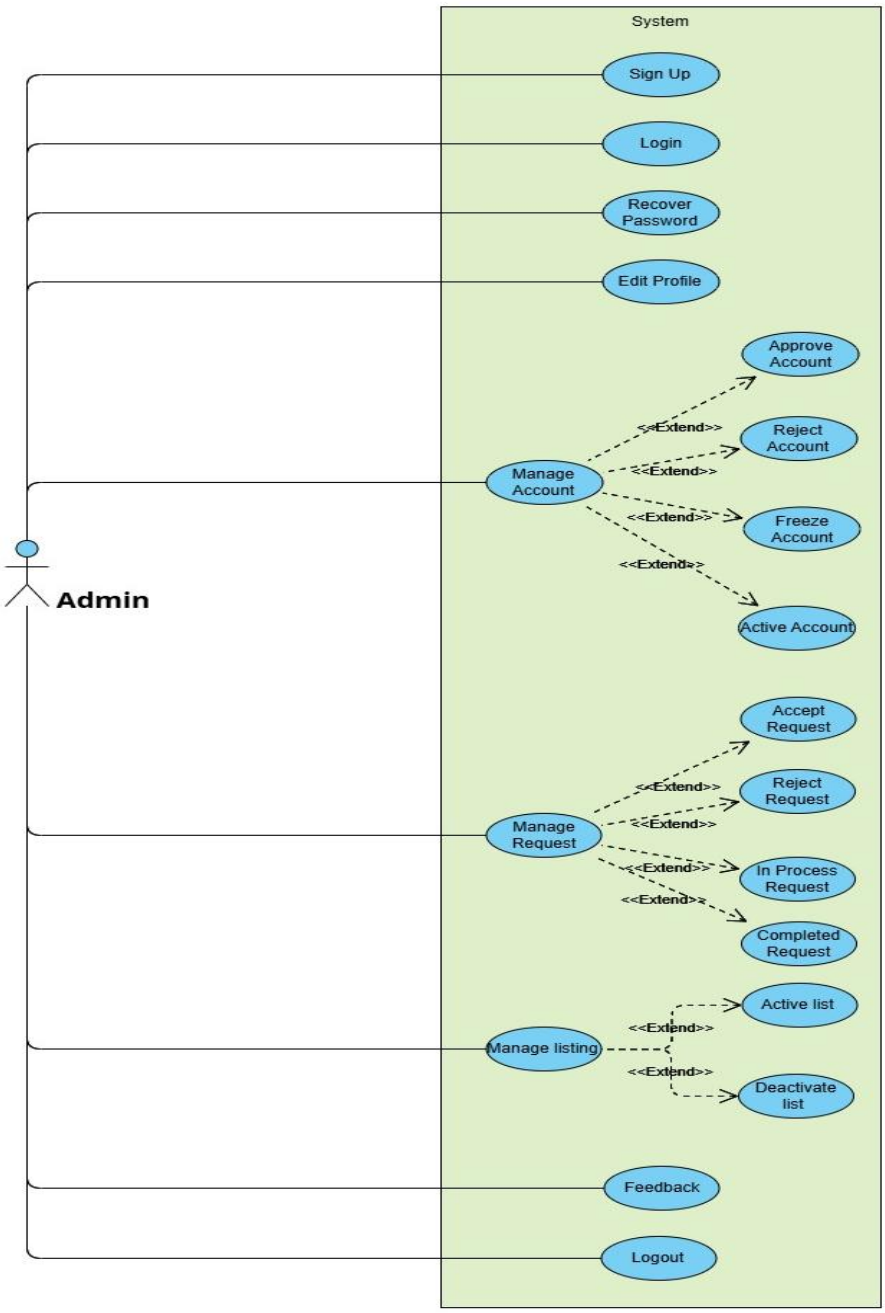


Figure 3.2: Admin Use-Case Diagram

3.5.2 Needy Use-case Diagram:

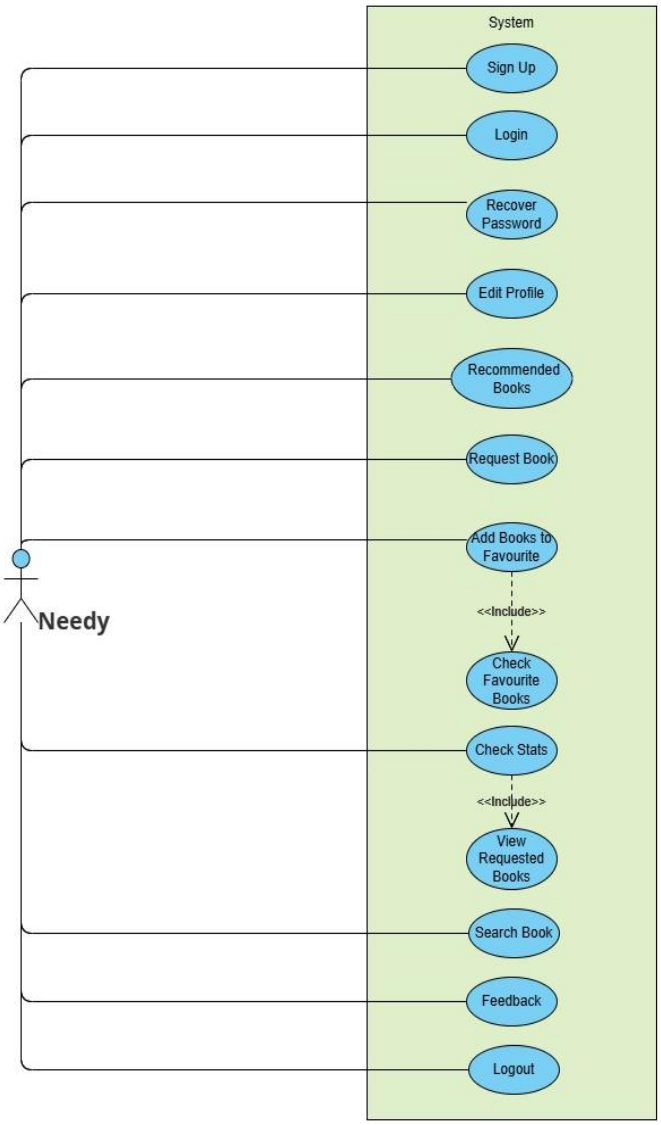


Figure 3.3: Needy Use-Case Diagram

3.5.3 Donor Use-Case Diagram:

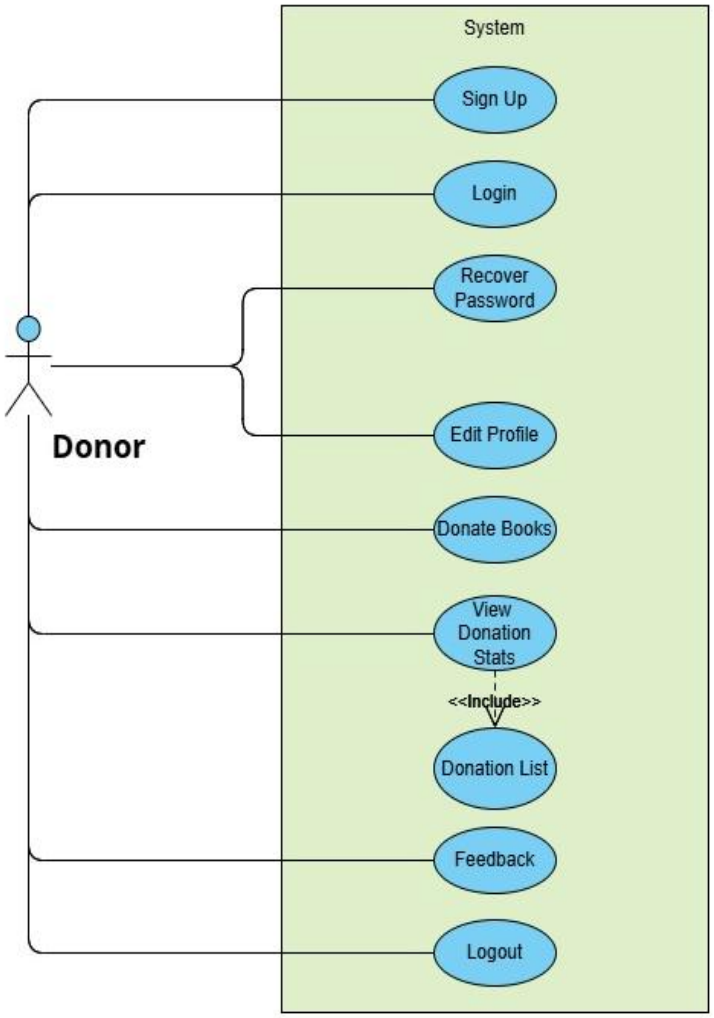


Figure 3.4: Donor Use-Case Diagram

3.5.4 Volunteer Use-Case Diagram

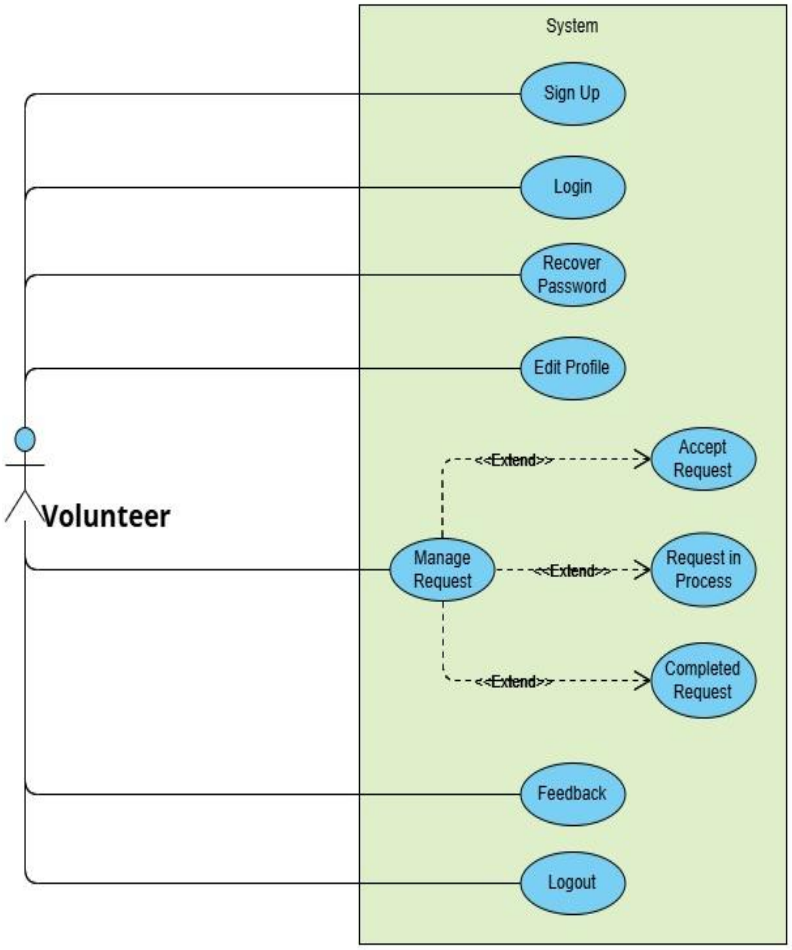


Figure 3.5: Volunteer Use-Case Diagram

3.5.5 Full System Use-Case Diagram:

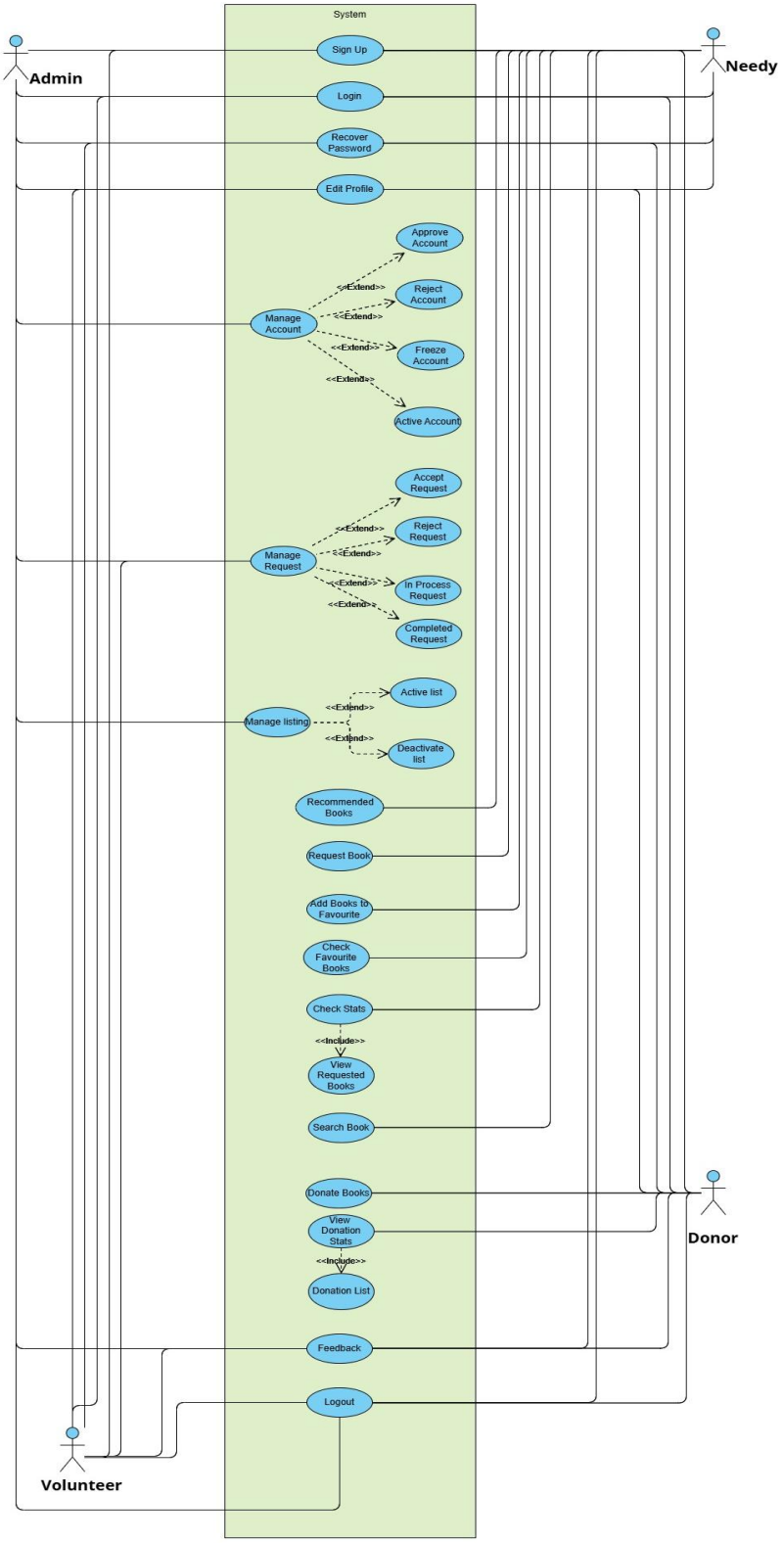


Figure 3.6: Full System Use-Case Diagram

3.6 Fully Dressed Use Cases:

3.6.1 Sign Up:

Table 3.5: Fully Dressed Use Case (Sign Up)

USE CASE ID	**UC-001**
USE CASE NAME	Sign Up
PRIMARY ACTOR	Donor / Needy / Volunteer / Admin
PRECONDITION	User is on the registration page
POSTCONDITION	User account is successfully created
MAIN FLOW	1. User accesses the Sign Up page 2. Enters required details (name, email, password etc) 3. Clicks Sign Up 4. System validates input 5. System saves data and creates account 6. User is notified of successful registration
ALTERNATE FLOW	1a. Missing or invalid data: system prompts correction 1b. Email already exists: system notifies and blocks submission

3.6.2 Login

Table 3.6: Fully Dressed Use Case (Login)

USE CASE ID	**UC-002**
USE CASE NAME	Login
PRIMARY ACTOR	Donor / Needy / Volunteer / Admin
PRECONDITION	User is registered and on login screen
POSTCONDITION	User is logged into the system
MAIN FLOW	1. User enters email and password 2. Clicks Login 3. System verifies credentials 4. Redirects to user dashboard
ALTERNATE FLOW	1a. Invalid credentials: system shows error

	1b. Account is frozen: system denies access and displays message

3.6.3 Recover password:

Table 3.7: Fully Dressed Use Case (Recover Password)

USE CASE ID	**UC-003**
USE CASE NAME	Recover Password
PRIMARY ACTOR	Donor / Needy / Volunteer / Admin
PRECONDITION	User is on password recovery screen
POSTCONDITION	Forgot Password OTP is sent to user's email
MAIN FLOW	1. User enters registered email 2. Clicks 'Verify Email' 3. System sends OTP via email 4. User resets password via OTP
ALTERNATE FLOW	1a. Email not registered: system notifies user

3.6.4 Edit Profile:

Table 3.8: Fully Dressed Use Case (Edit Profile)

USE CASE ID	**UC-004**
USE CASE NAME	Edit Profile
PRIMARY ACTOR	Donor / Needy / Volunteer / Admin
PRECONDITION	User is logged in and on profile page
POSTCONDITION	Profile is updated
MAIN FLOW	1. User navigates to profile page 2. Edits fields (name, email, Profile picture etc.) 3. Verify email with OTP 4. Clicks Save 5. System updates database 6. Confirms update to user
ALTERNATE FLOW	1a. Invalid inputs: system prompts correction

3.6.5 Donate Books:

Table 3.9: Fully Dressed Use Case (Donate Books)

USE CASE ID	**UC-005**
USE CASE NAME	Donate Books
PRIMARY ACTOR	Donor
PRECONDITION	Donor is logged in
POSTCONDITION	Book donation is recorded
MAIN FLOW	1. Donor selects 'Switch to donate' 2. Fills form with book details 3. Submits donation 4. System records donation and confirmation by email.
ALTERNATE FLOW	1a. Missing book details: system prompts for completion

3.6.6 Request Book:

Table 3.10: Fully Dressed Use Case (Request Book)

USE CASE ID	**UC-006**
USE CASE NAME	Request Book
PRIMARY ACTOR	Needy
PRECONDITION	Needy is logged in
POSTCONDITION	Book request is submitted
MAIN FLOW	1. See books on Dashboard 2. Selects a book and clicks 'Request' 3. System saves the request 4. Request Confirmation email sent to user
ALTERNATE FLOW	1a. Book unavailable: user shown notification 1b your request has not been submitted you have exceed your limit for this month

3.6.7 Manage Donor Request:

Table 3.11: Fully Dressed Use Case (Manage Donor Request)

USE CASE ID	**UC-007**
USE CASE NAME	Manage Request
PRIMARY ACTOR	Volunteer
PRECONDITION	Volunteer is logged in
POSTCONDITION	Request is processed
MAIN FLOW	1. Volunteer views pending requests 2. Accept one to manage 3. Accepted request status will be change to in process
ALTERNATE FLOW	1a.

3.6.8 Manage Account:

Table 3.12: Fully Dressed Use Case (Manage Account)

USE CASE ID	**UC-007**
USE CASE NAME	Manage Account
PRIMARY ACTOR	Admin
PRECONDITION	Admin is logged in
POSTCONDITION	Account Status is Updated to (approved, Reject, frozen, Active.)
MAIN FLOW	1. Admin views list of pending accounts 2. Approves, rejects, or freezes and active as needed 3. System updates status 4. System send email to user about the account status
ALTERNATE FLOW	1a. System shows message no data available

3.6.9 Manage Listing:

Table 3.13: Fully Dressed Use Case (Manage Listing)

USE CASE ID	**UC-009**
USE CASE NAME	Manage Listing
PRIMARY ACTOR	Admin
PRECONDITION	Admin is logged in
POSTCONDITION	Book listing is activated or deactivated
MAIN FLOW	1. Admin navigates to Active List 2. View active list 3. Chooses Activate/Deactivate 4. System update the status of book
ALTERNATE FLOW	1a. No data available

3.6.10 Feedback:

Table 3.14: Fully Dressed Use Case (Feedback)

USE CASE ID	**UC-010**
USE CASE NAME	Feedback
PRIMARY ACTOR	All Users
PRECONDITION	User on landing page
POSTCONDITION	Feedback is Saved Successfully
MAIN FLOW	1. User accesses feedback form 2. Fill and submits feedback 3. System saves feedback
ALTERNATE FLOW	1a. system prompts to fields are reuqired

3.6.11 Recommended Books:

Table 3.15: Fully Dressed Use Case (Recommended Books)

USE CASE ID	UC-011
USE CASE NAME	Recommended Books
PRIMARY ACTOR	Needy
PRECONDITION	Needy Must be logged in
POSTCONDITION	Can view recommended Books
MAIN FLOW	1. Needy will log in 2. Needy navigates to dashboard 3. System shows recommended Books
ALTERNATE FLOW	1a.

3.6.12 Add to Favorites:

Table 3.16: Fully Dressed Use Case (Add to Favorites)

USE CASE ID	UC-012
USE CASE NAME	Add to Favorites
PRIMARY ACTOR	Needy
PRECONDITION	Needy must be Logged in and on Dashboard
POSTCONDITION	Book added to favorites
MAIN FLOW	1. User will view available books on dashboard 2. User will click on like button 3. Book will be added to favorites
ALTERNATE FLOW	1a. No Available Books

3.6.13 Requested Book Statistics:

Table 3.17: Fully Dressed Use Case (Requested Book Statistics)

USE CASE ID	UC-013
USE CASE NAME	Requested Book Statistics
PRIMARY ACTOR	Needy
PRECONDITION	Needy Should be on profile page
POSTCONDITION	View request book
MAIN FLOW	1. Needy clicks to profile page 2. Needy view requested books statistics 3. Clicks on requested books 4. View all requested book details
ALTERNATE FLOW	1a. No Requestedbook requested

3.6.14 Search Book:

Table 3.18: Fully Dressed Use Case (Search Book)

USE CASE ID	UC-014
USE CASE NAME	Search Book
PRIMARY ACTOR	Needy
PRECONDITION	Needy must be on the dashboard
POSTCONDITION	Searched books will be displayed
MAIN FLOW	1. Needy accesses the dashboard 2. Needy enters keywords in the search bar 3. System fetches matching books from the database 4. System displays the list of books that match the search keywords
ALTERNATE FLOW	1a. If no match is found, system displays “No books found” 2a. If Needy enters invalid characters or empty string, system prompts: “Please enter a valid book title or keyword”

3.6.15 Donated Book Statistics:

Table 3.19: Fully Dressed Use Case (Donated Book Statistics)

USE CASE ID	**UC-015**
USE CASE NAME	Donated Book Statistics
PRIMARY ACTOR	Donor
PRECONDITION	Donor Should be on Profile page
POSTCONDITION	View donated books
MAIN FLOW	1. Donor clicks on profile page 2. Donor view donated book statistics 3. Clicks on donated books 4. View all donated books
ALTERNATE FLOW	1a No donated books found

3.6.16 Manage Request:

Table 3.20: Fully Dressed Use Case (Manage Request)

USE CASE ID	UC-016
USE CASE NAME	Manage Request
PRIMARY ACTOR	Admin
PRECONDITION	Admin must be logged in and, on the dashboard,
POSTCONDITION	Request status will be updated to either **Accepted**, **Rejected**, or **Completed**
MAIN FLOW	1. Admin logs in and navigates to the Dashboard 2. Admin clicks on "Manage Requests" tab 3. System displays all pending book requests 4. Admin selects a request 5. Admin updates the request status (Accepted/Rejected/Completed) 6. System updates the request status in the database
ALTERNATE FLOW	1a. If no requests are available, system shows message: “No requests to manage.”

3.6.17 Logout:

Table 3.21: Fully Dressed Use Case (Logout)

USE CASE ID	**UC-017**
USE CASE NAME	Logout
PRIMARY ACTOR	Admin / Donor / Volunteer / Needy
PRECONDITION	All user must be Logged in
POSTCONDITION	Logout Successfully
MAIN FLOW	1. User is on dashboard 2. Clicks on profile picture or dropdown 3. Selects "Logout" from the menu 4. System terminates session and redirects to login screen
ALTERNATE FLOW	

3.7 Entity Relationship diagram

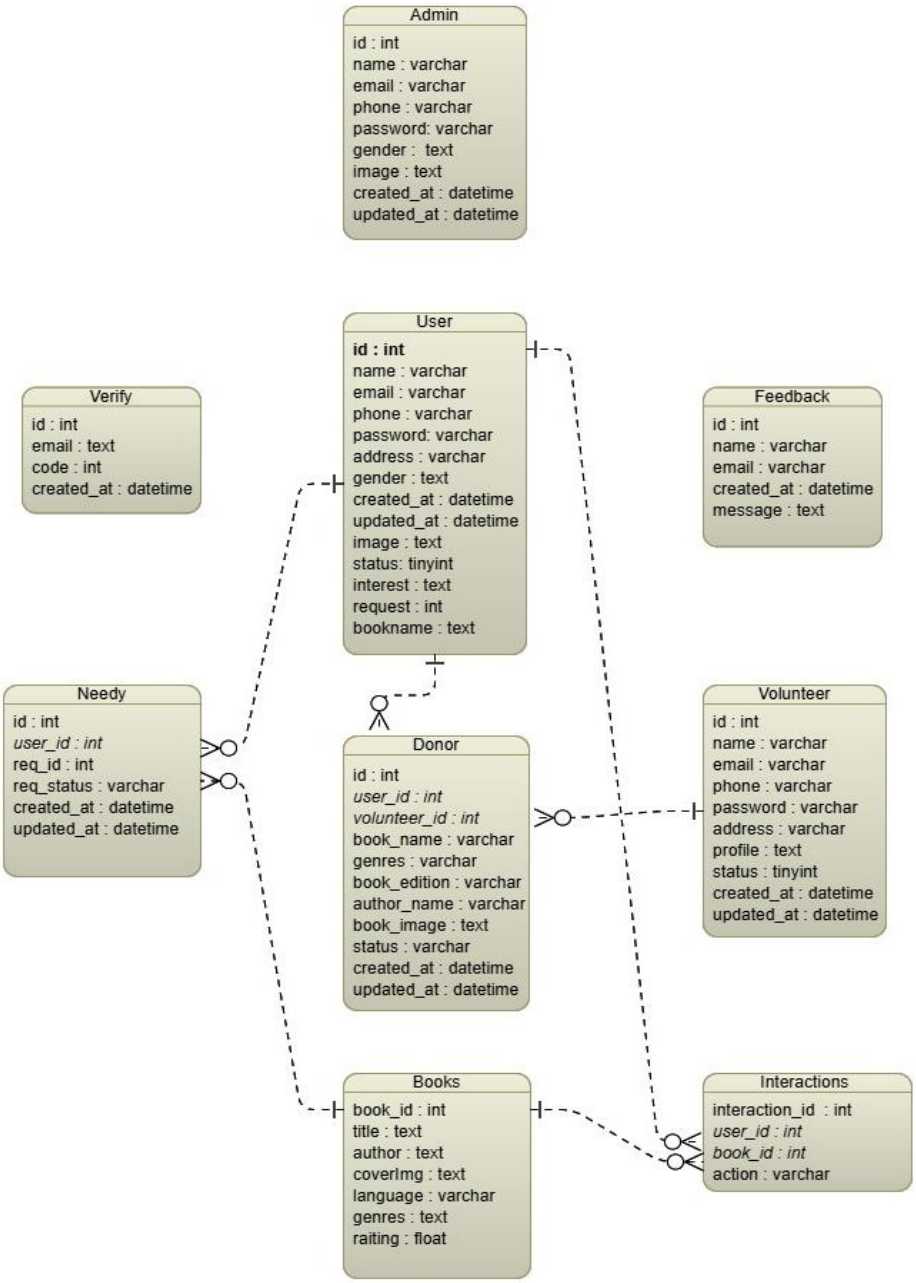


Figure 3.7: Entity Relationship Diagram (ERD)

3.8 Activity Diagram

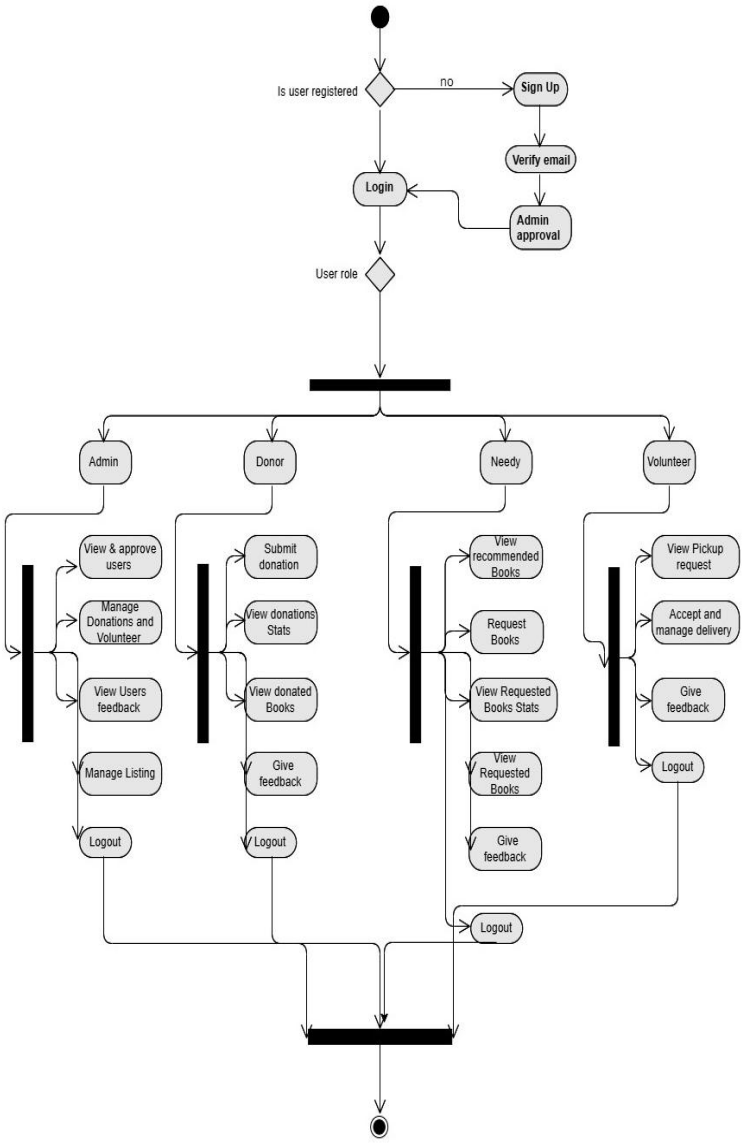
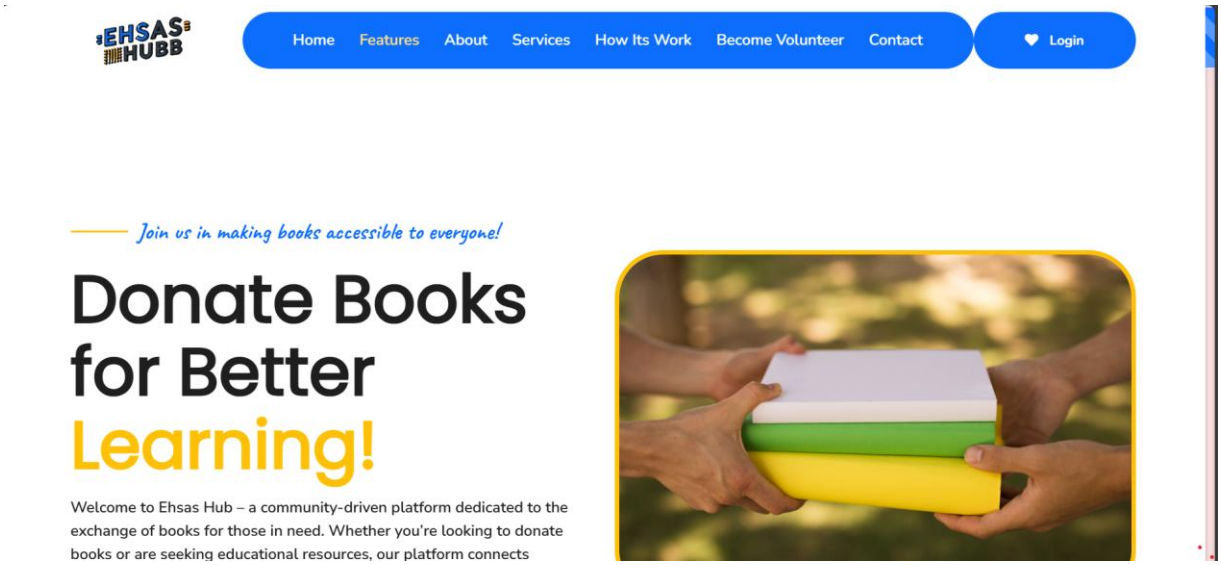
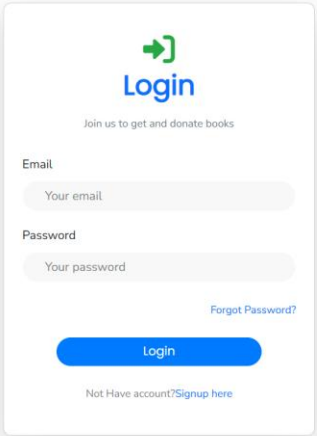


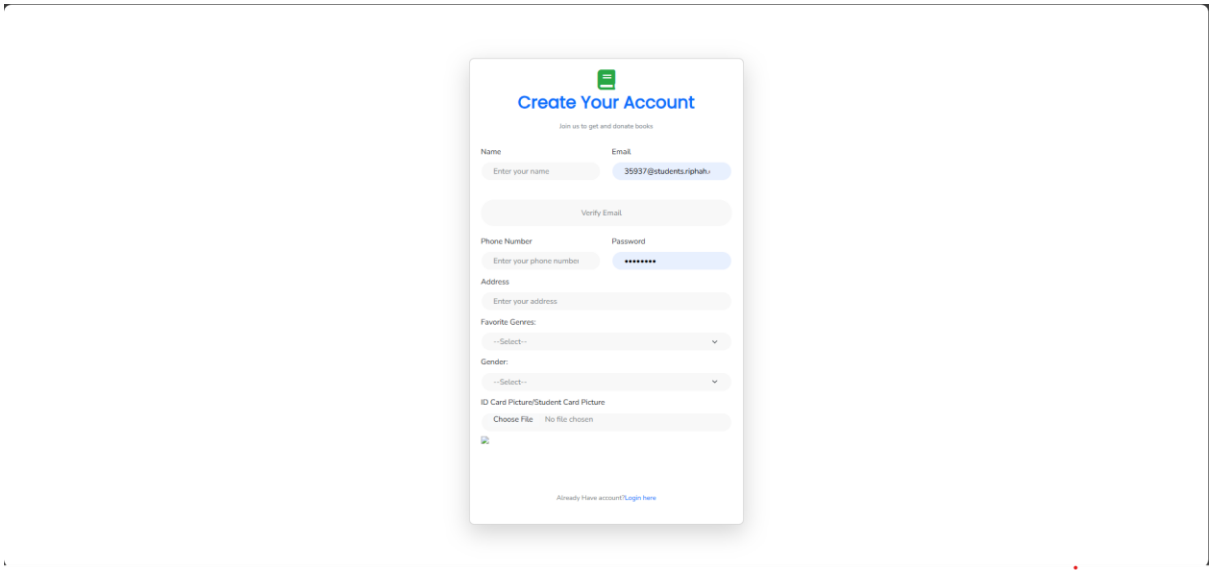
Figure 3.8: Whole system Activity Diagram

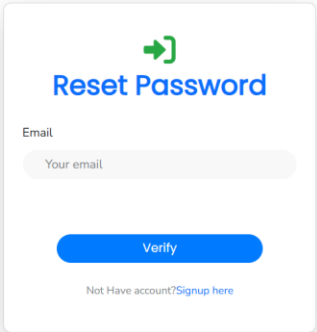
3.9 GUI Graphical User Interfaces

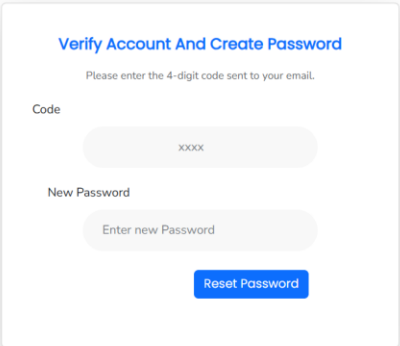
3.9.1 User Role GUI:

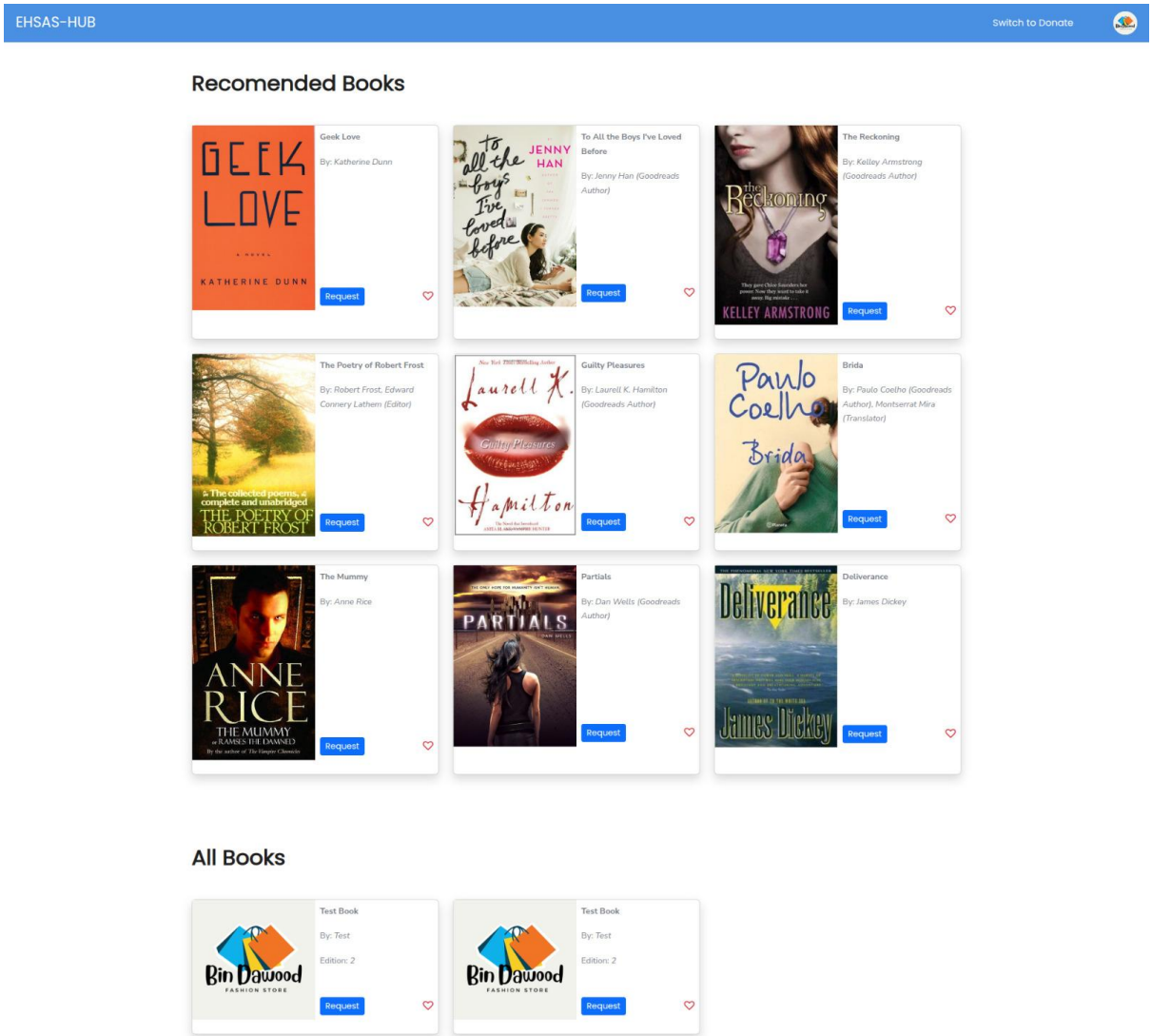


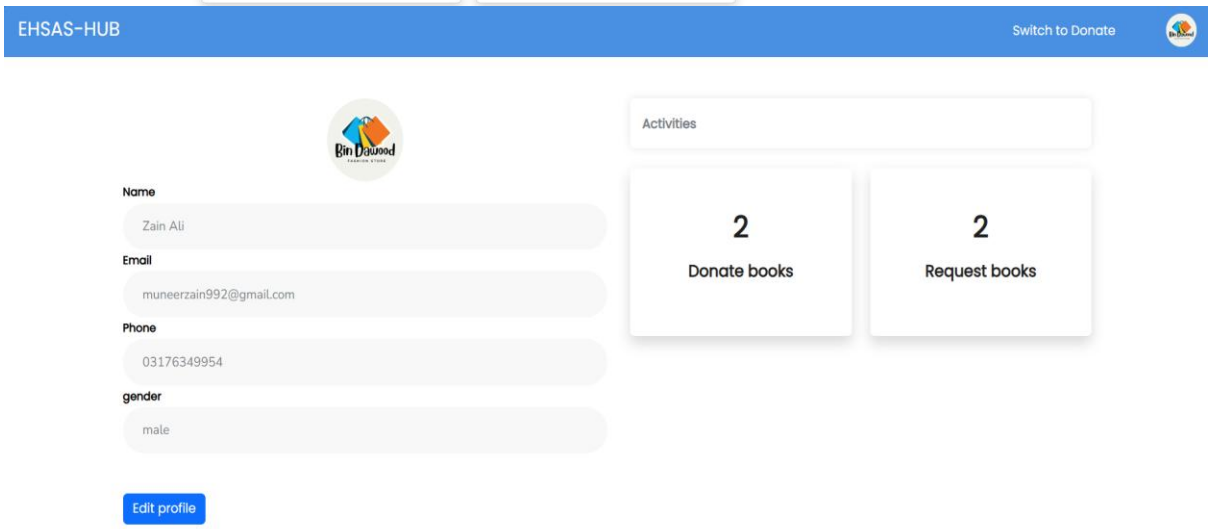


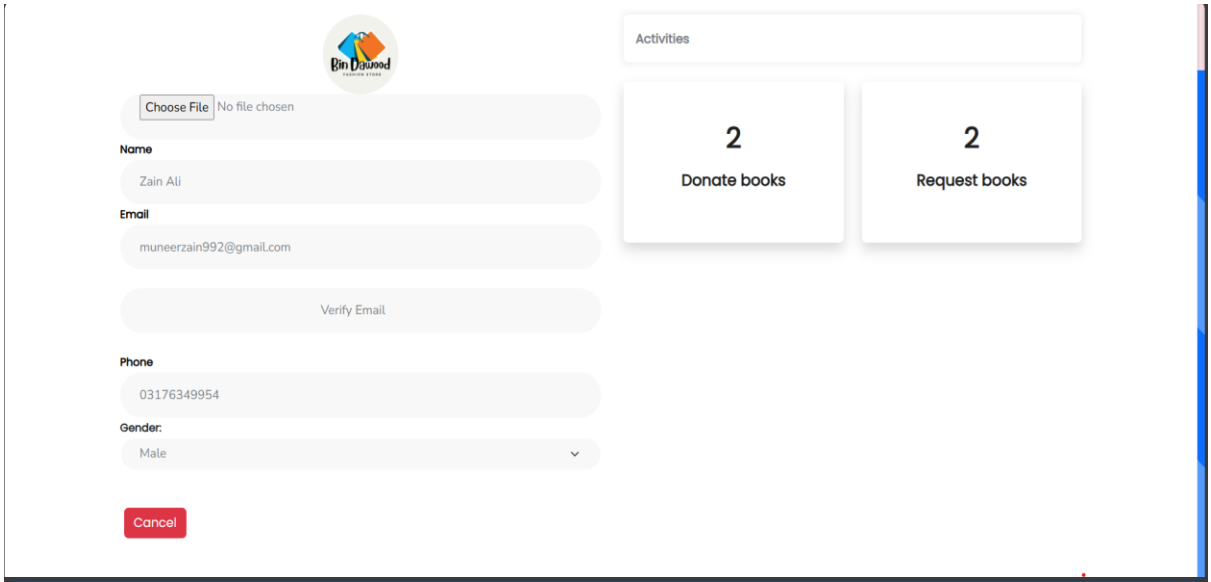

Create Your Account
Join us to get and donate books
Name
Enter your name
Email
35937@students.riphah.
Verify Email
Phone Number
Enter your phone number
Password
Address
Enter your address
Favorite Genres:
--Select--
Gender:
--Select--
ID Card Picture/Student Card Picture
Choose File
No file chosen
Already Have account?Login here

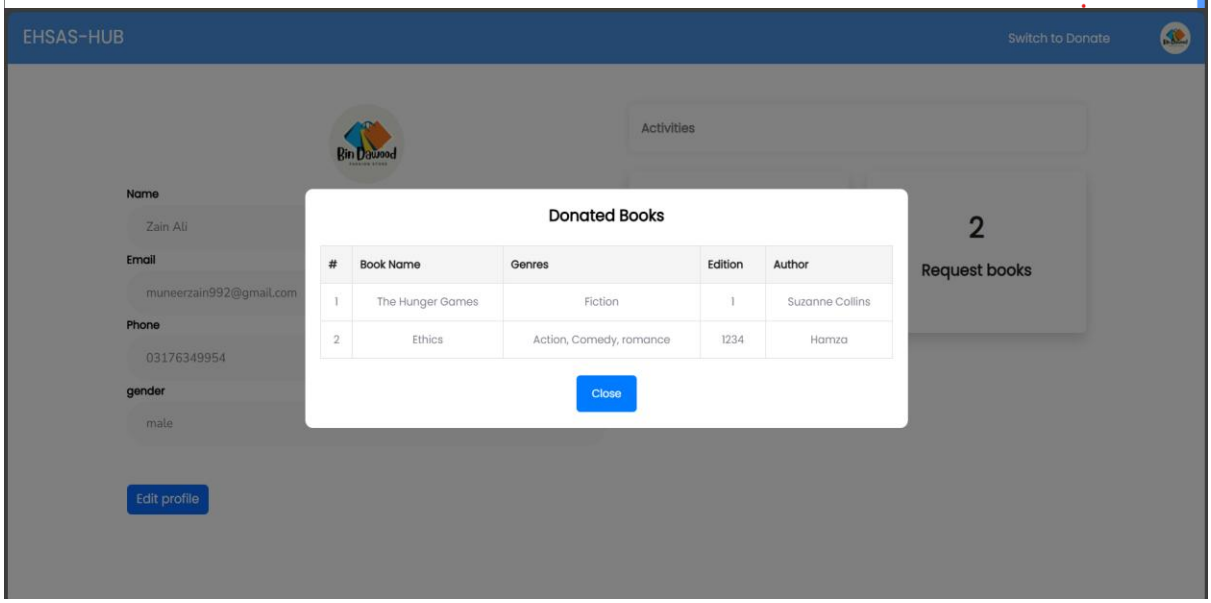

Reset Password
Email
Your email
Verify
Not Have account?Signup here

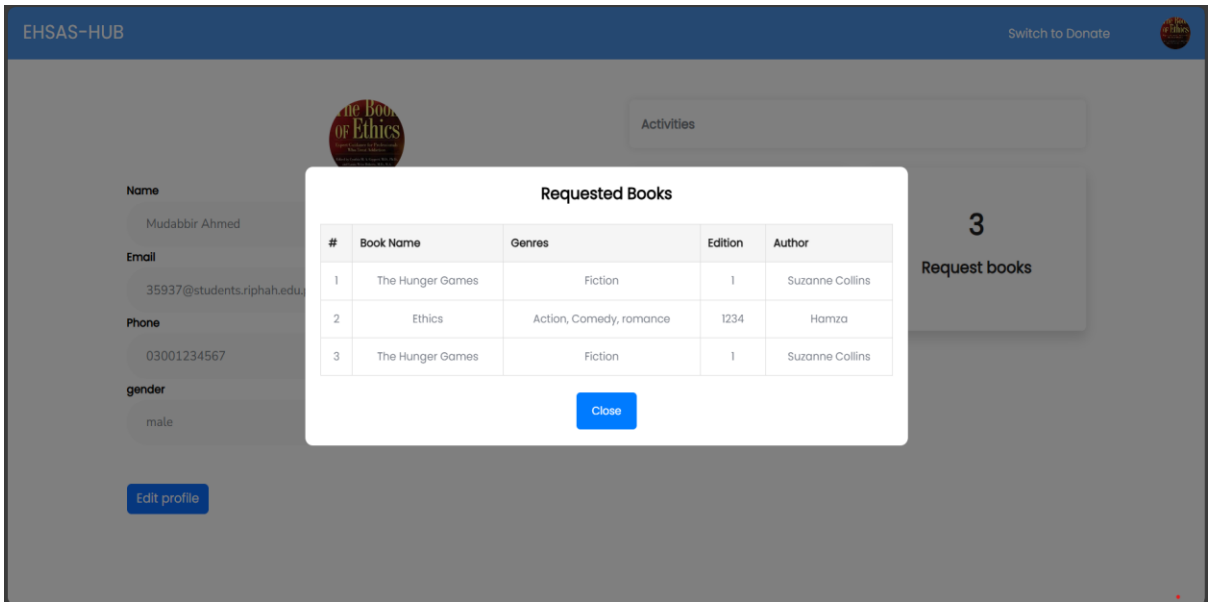

Verify Account And Create Password
Please enter the 4-digit code sent to your email.
Code
XXXX
New Password
Enter new Password
Reset Password

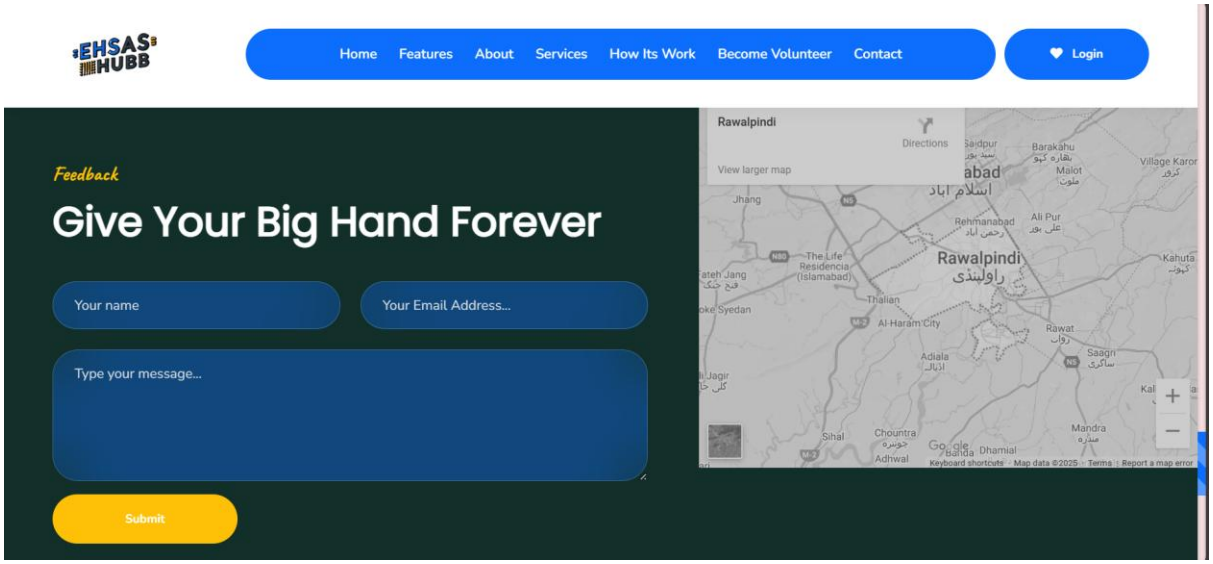

EHSAS-HUB
Switch to Donate
Recomended Books
Geek Love
By: Katherine Dunn
Request
To All the Boys I've Loved Before
By: Jenny Han (Goodreads Author)
Request
The Reckoning
By: Kelley Armstrong (Goodreads Author)
Request
The Poetry of Robert Frost
By: Robert Frost, Edward Connery Lathem (Editor)
Request
Guilty Pleasures
By: Laurell K. Hamilton (Goodreads Author)
Request
Brida
By: Paulo Coelho (Goodreads Author), Montserrat Mira (Translator)
Request
The Mummy
By: Anne Rice
Request
Partials
By: Dan Wells (Goodreads Author)
Request
Deliverance
By: James Dickey
Request
All Books
Test Book
By: Test
Edition: 2
Request
Test Book
By: Test
Edition: 2
Request


EHSAS-HUB
Switch to Donate
Name
Zain Ali
Email
muneerzain992@gmail.com
Phone
03176349954
gender
male
Edit profile
Activities
2
Donate books
2
Request books

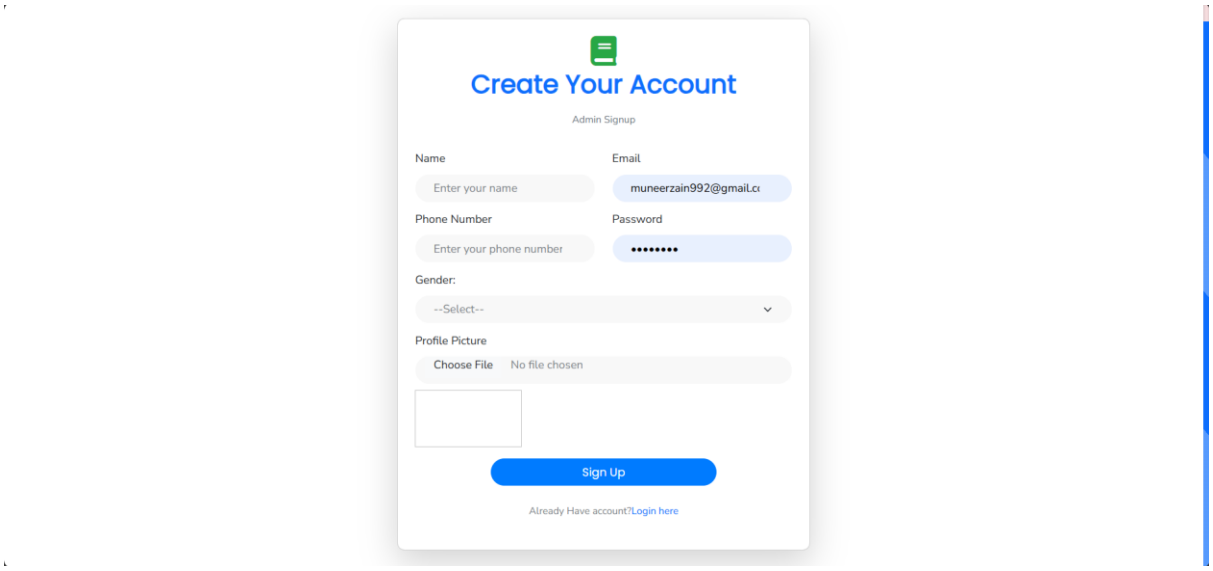

Choose File No file chosen
Name
Zain Ali
Email
muneerzain992@gmail.com
Verify Email
Phone
03176349954
Gender:
Male
Cancel
Activities
2
Donate books
2
Request books

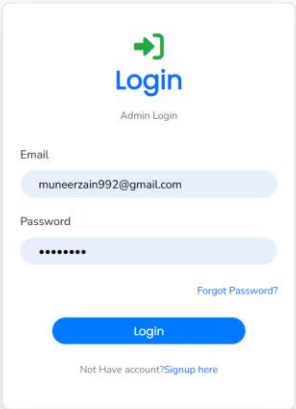

EHSAS-HUB
Switch to Donate
Activities
Name
Zain Ali
Email
muneerzain992@gmail.com
Phone
03176349954
gender
male
Edit profile
Donated Books
| Book Name | Genres | Edition | Author
1 | The Hunger Games | Fiction | 1 | Suzanne Collins
2 | Ethics | Action, Comedy, romance | 1234 | Hamza
Close
2
Request books

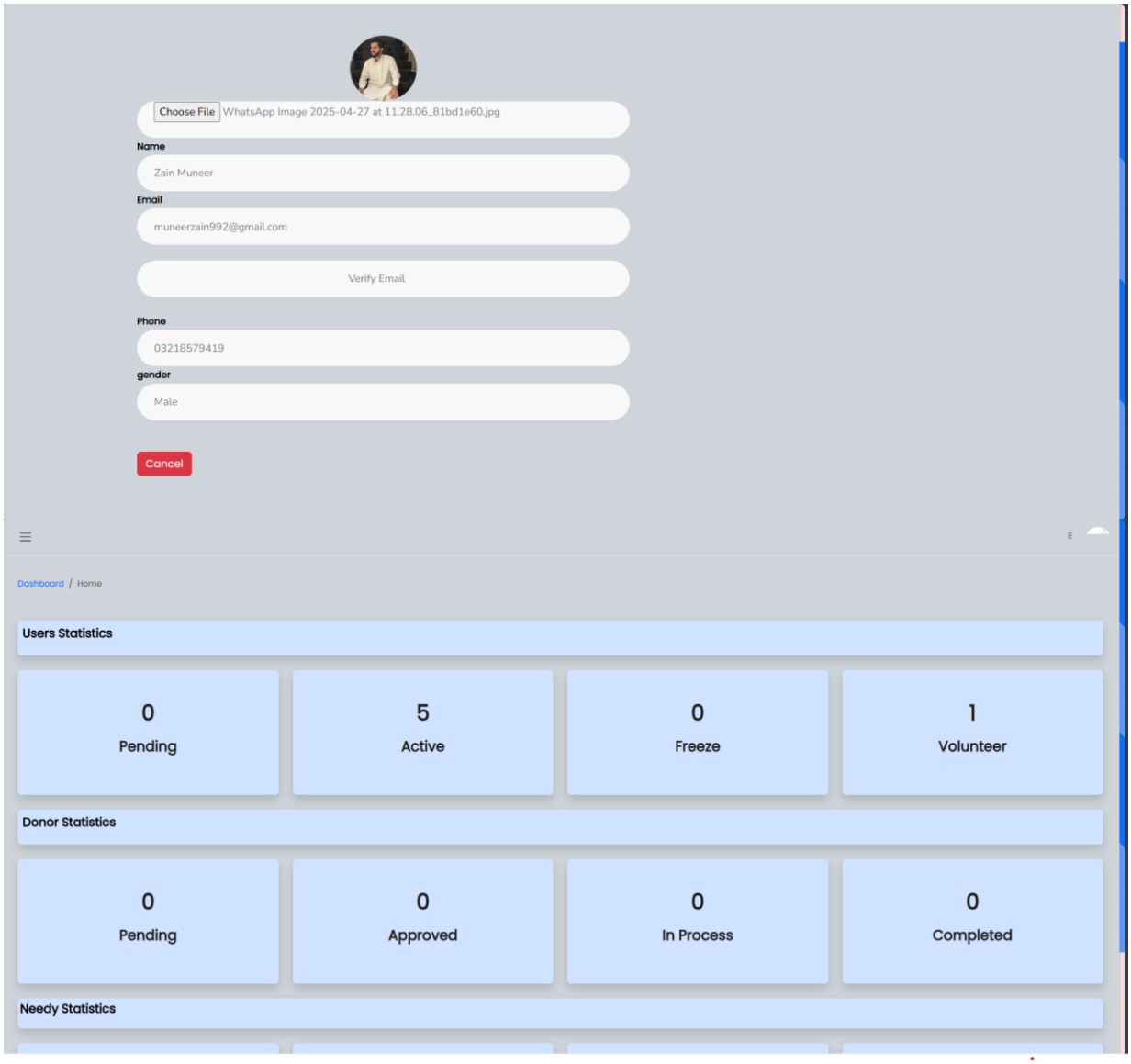

EHSAS-HUB
Switch to Donate
Activities
Name
Mudabbir Ahmed
Email
35937@students.riphah.edu.
Phone
03001234567
gender
male
Edit profile
Requested Books
| Book Name | Genres | Edition | Author
1 | The Hunger Games | Fiction | 1 | Suzanne Collins
2 | Ethics | Action, Comedy, romance | 1234 | Hamza
3 | The Hunger Games | Fiction | 1 | Suzanne Collins
Close
3
Request books



3.9.2 Admin Role GUI:






Choose File
WhatsApp Image 2025-04-27 at 11.28.06_81bd1e60.jpg
Name
Zain Muneer
Email
muneerzain992@gmail.com
Verify Email
Phone
03218579419
gender
Male
Cancel
Dashboard / Home
Users Statistics
0
Pending
5
Active
0
Freeze
1
Volunteer
Donor Statistics
0
Pending
0
Approved
0
In Process
0
Completed
Needy Statistics

Ehsas Hul

Active List

book_name	auther_name	book_edition	Status	Action
The Hunger Games	1	Suzanne Collins	Active	Deactivate
Ethics	1234	Hamza	Active	Deactivate

Ehs

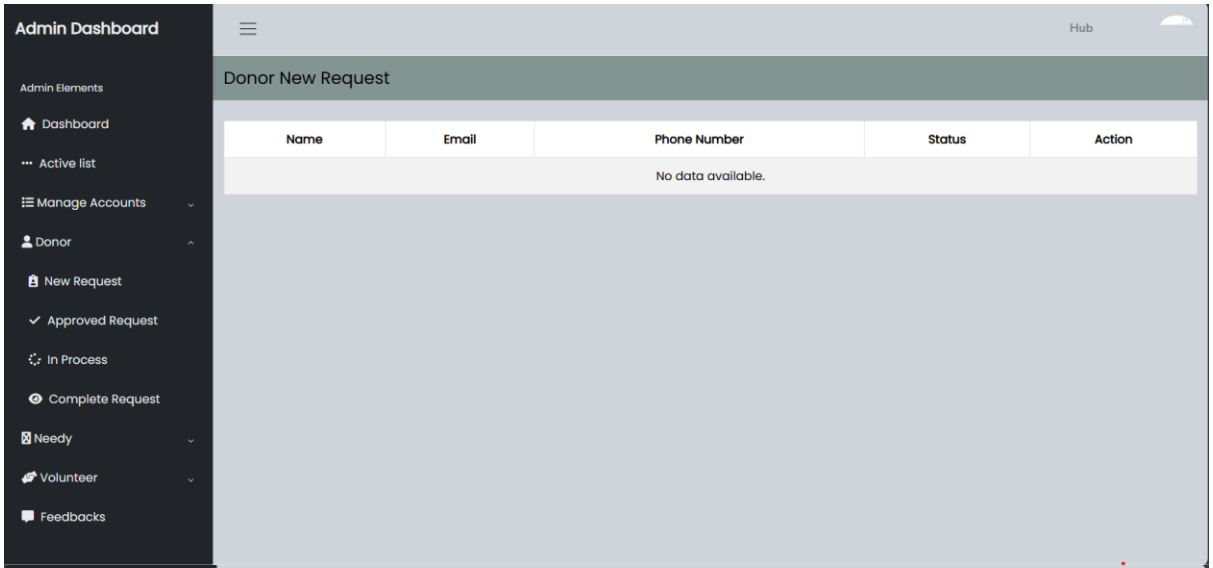
User List

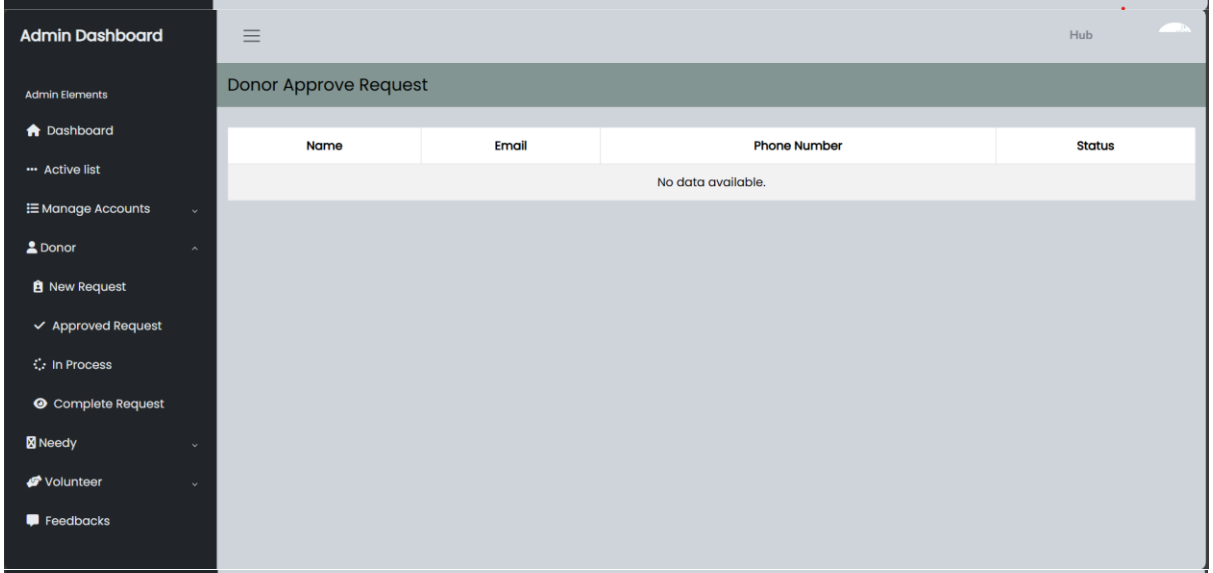
Name	Email	Phone Number	Status	Action
Test User	liyeye1256@miracle3.com	03176349954	Active	Action
Hamza Ahmad	humzaahmed628@gmail.com	03176349954	Active	Action
Abdullah Shahid	abdullahsatti200@gmail.com	03176349954	Active	Action
Zain Ali	muneerzain992@gmail.com	03176349954	Active	Action
Mudabbir Ahmed	35937@students.riphah.edu.pk	03341012642	Active	Action

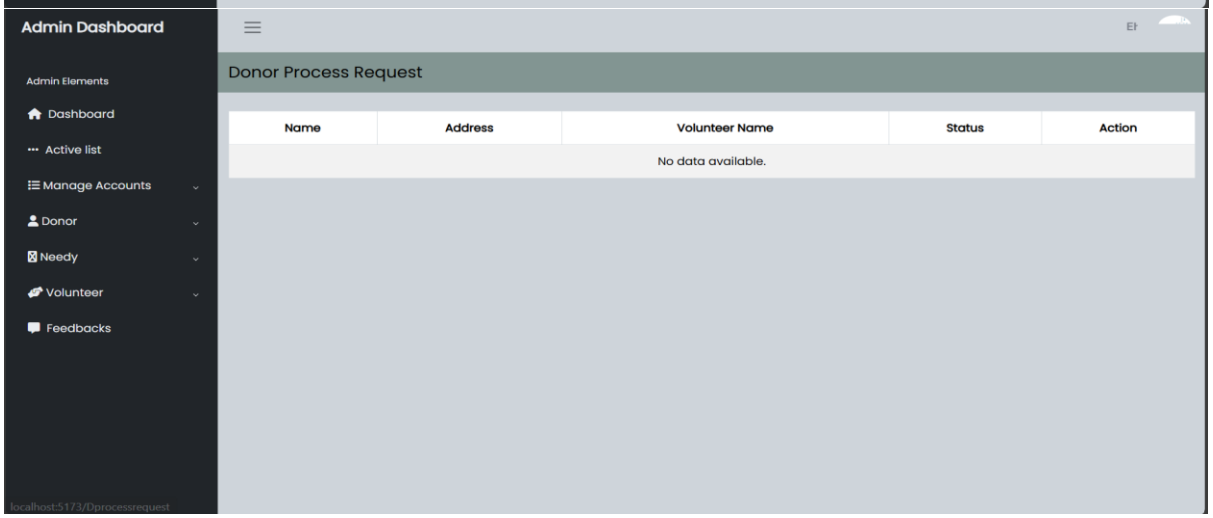
Ehsas Hub

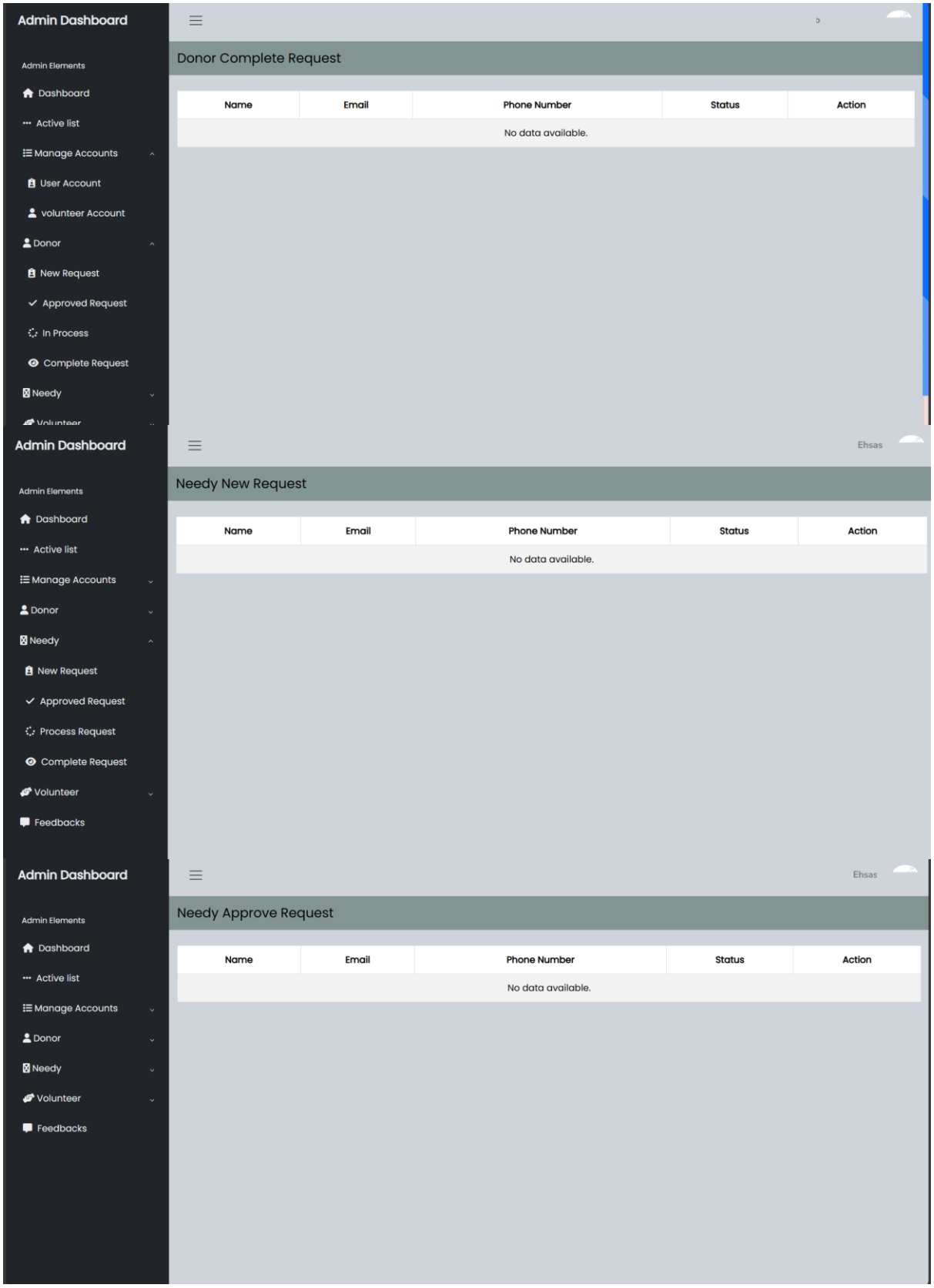
Volunteer

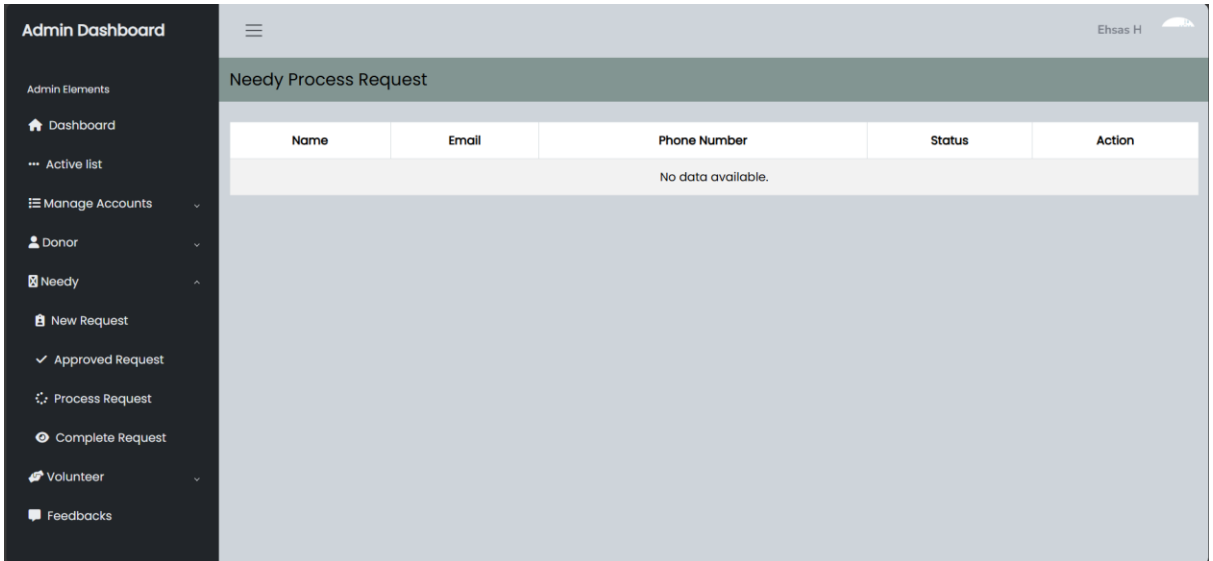
Name	Email	Phone Number	Status	Action
Zain Muneer	muneerzain992@gmail.com	03218579419	Active	Action


Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
New Request
Approved Request
In Process
Complete Request
Needy
Volunteer
Feedbacks
Hub
Donor New Request
Name
Email
Phone Number
Status
Action
No data available.

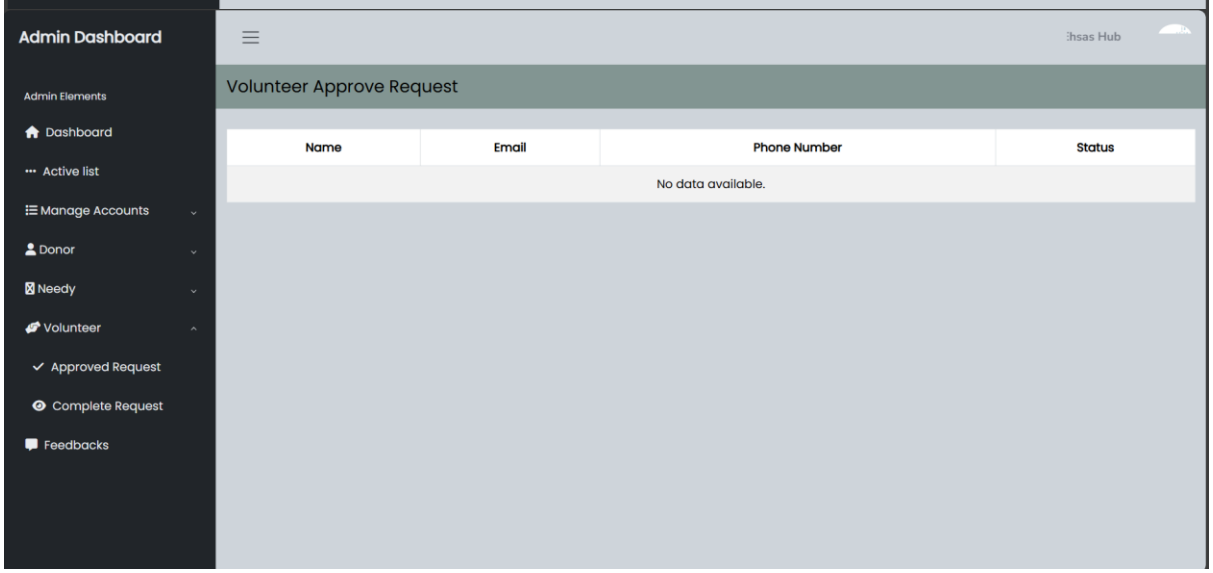

Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
New Request
Approved Request
In Process
Complete Request
Needy
Volunteer
Feedbacks
Hub
Donor Approve Request
Name
Email
Phone Number
Status
No data available.

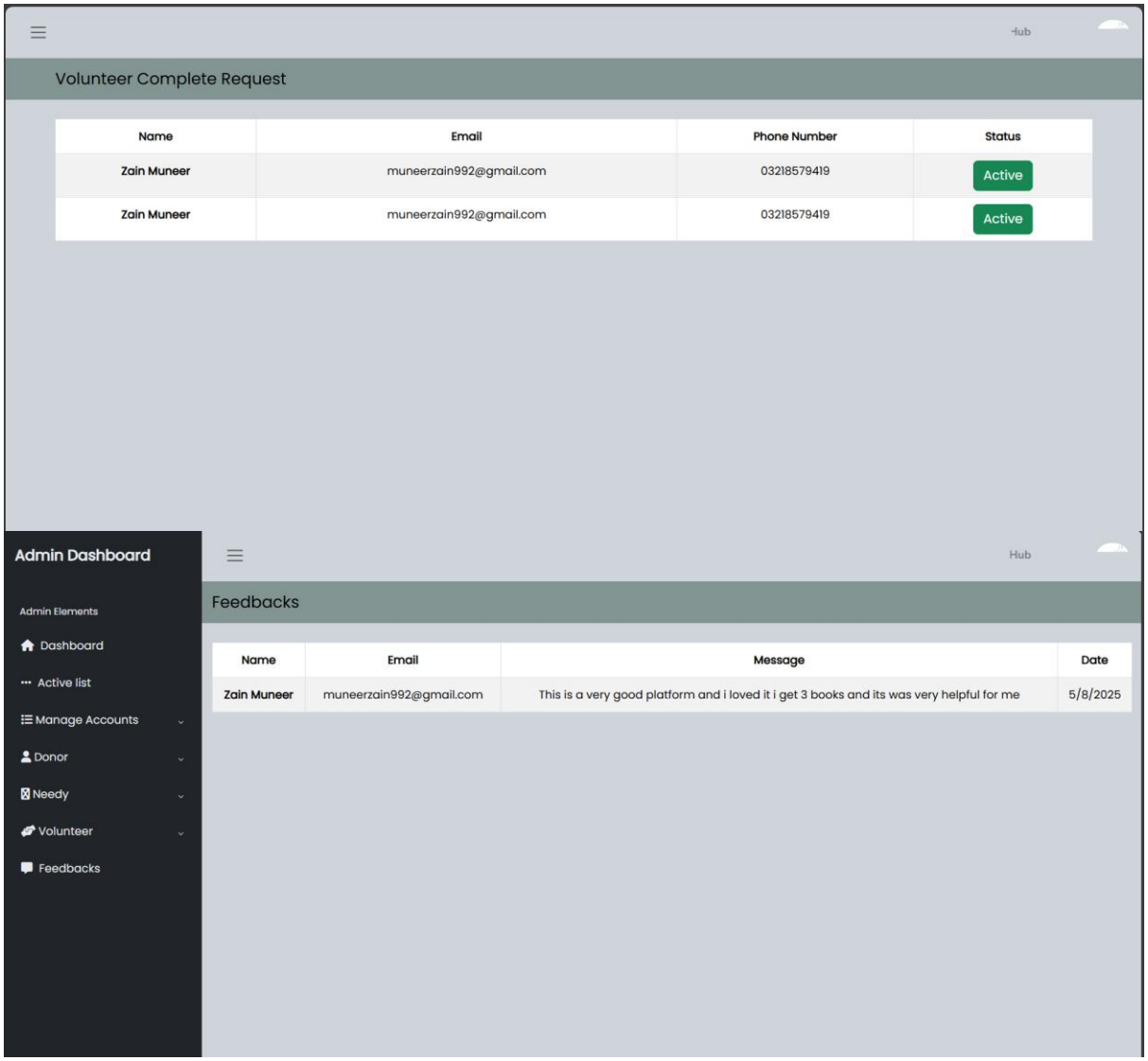

Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
Volunteer
Feedbacks
Donor Process Request
Name
Address
Volunteer Name
Status
Action
No data available.


Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
User Account
volunteer Account
Donor
New Request
Approved Request
In Process
Complete Request
Needy
Donor Complete Request
Name
Email
Phone Number
Status
Action
No data available.
Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
New Request
Approved Request
Process Request
Complete Request
Volunteer
Feedbacks
Ehsas
Needy New Request
Name
Email
Phone Number
Status
Action
No data available.
Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
Volunteer
Feedbacks
Ehsas
Needy Approve Request
Name
Email
Phone Number
Status
Action
No data available.

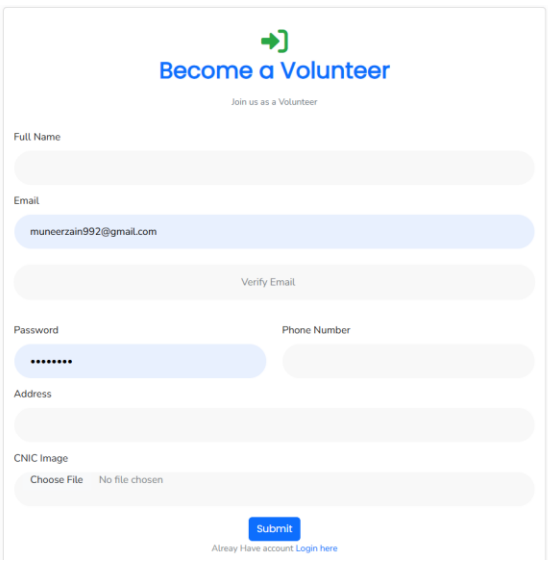

Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
New Request
Approved Request
Process Request
Complete Request
Volunteer
Feedbacks
Ehsas H
Needy Process Request
Name
Email
Phone Number
Status
Action
No data available.

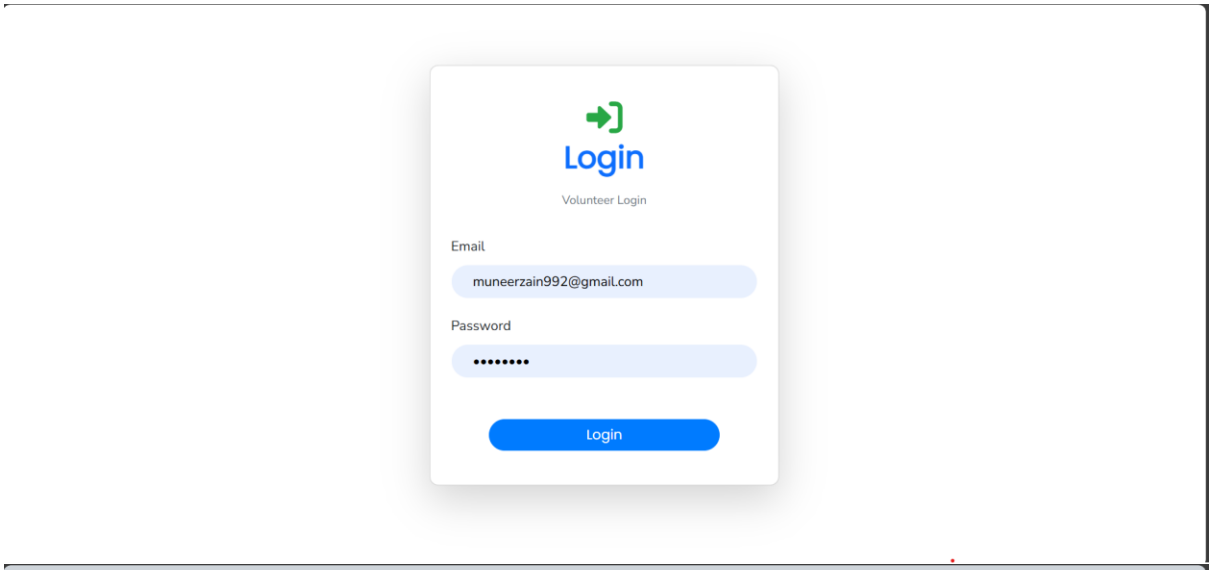

Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
New Request
Approved Request
Process Request
Complete Request
Volunteer
Feedbacks
Ehsas Hub
Needy Complete Request
Name
Email
Phone Number
Status
No data available.

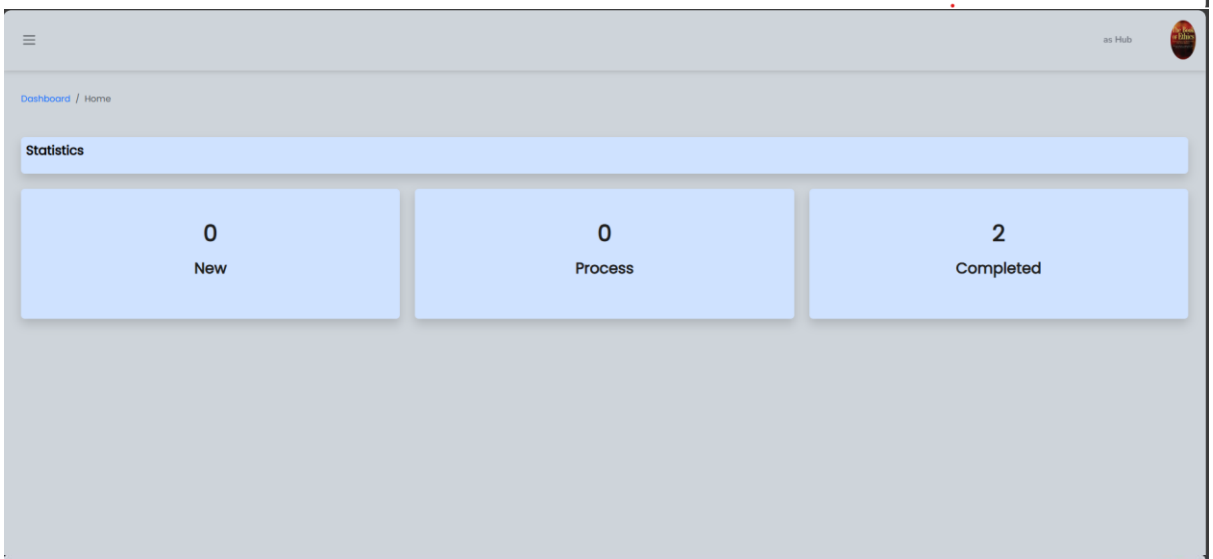

Admin Dashboard
Admin Elements
Dashboard
Active list
Manage Accounts
Donor
Needy
Volunteer
Approved Request
Complete Request
Feedbacks
:hsas Hub
Volunteer Approve Request
Name
Email
Phone Number
Status
No data available.

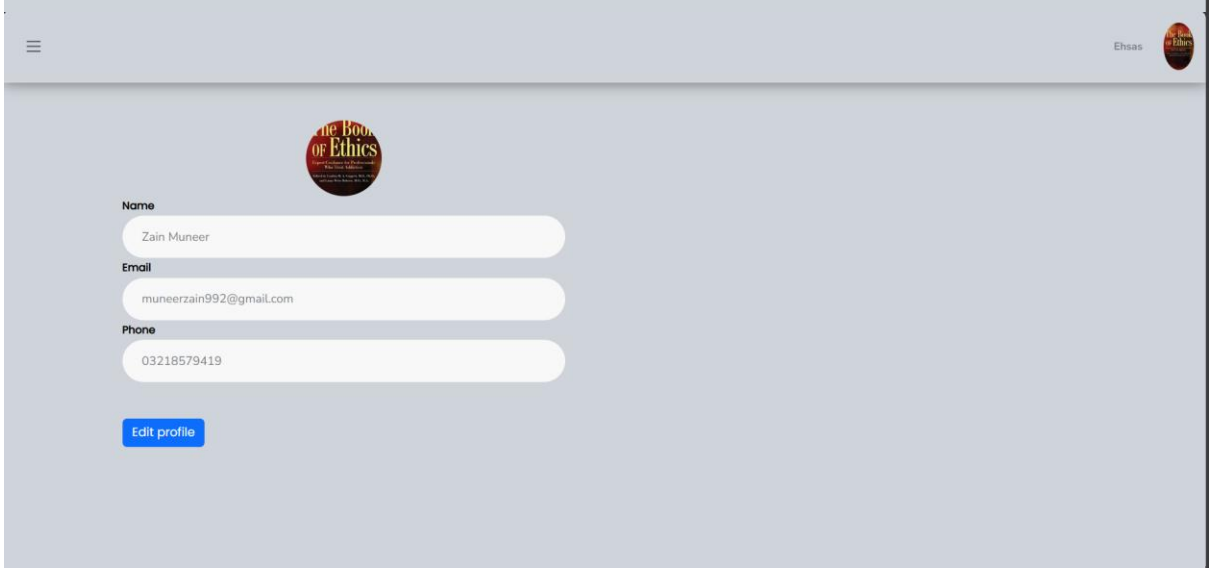


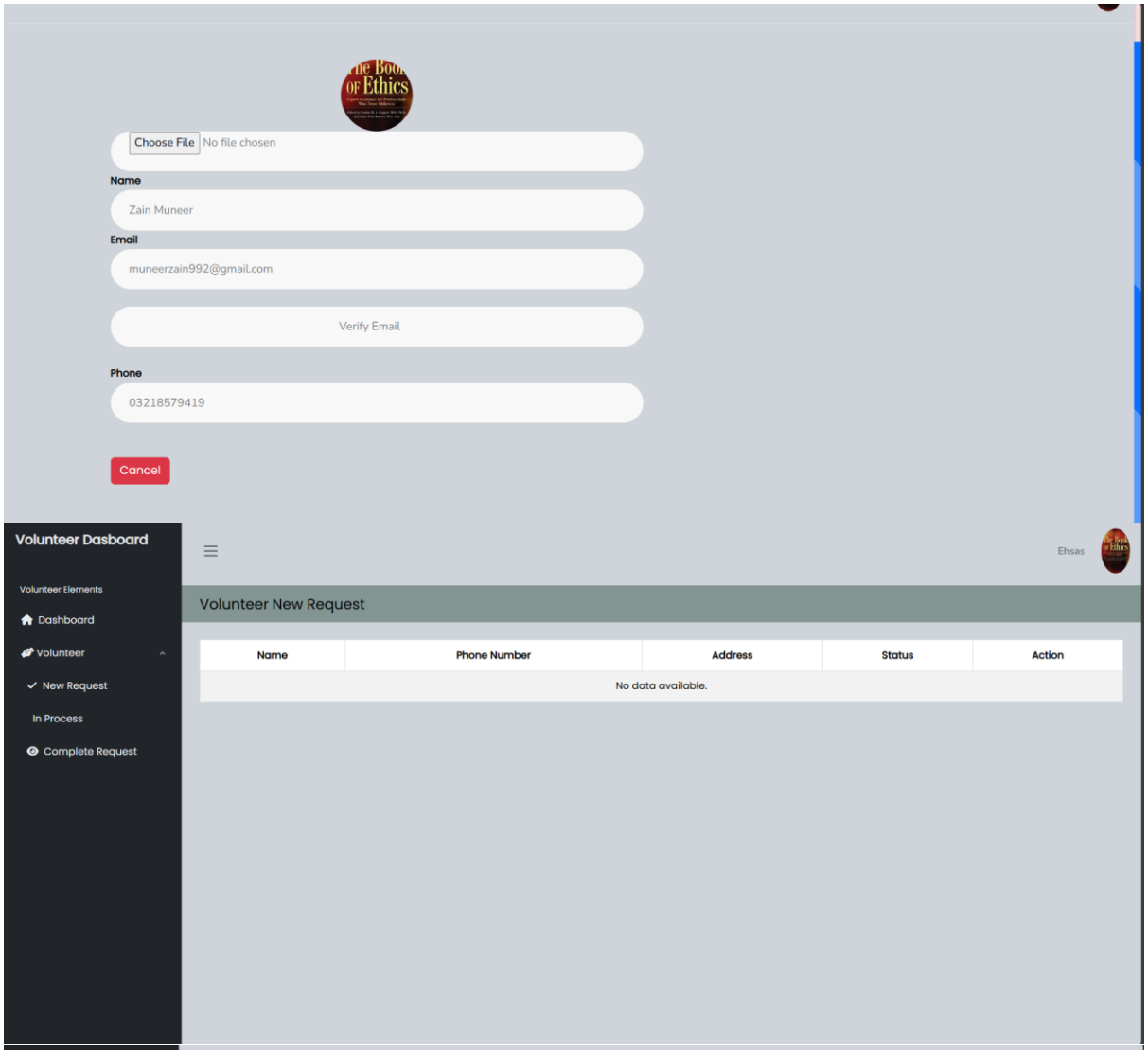
3.9.3 Volunteer Role GUI:




Login
Volunteer Login
Email
muneerzain992@gmail.com
Password
Login


Dashboard / Home
Statistics
0
New
0
Process
2
Completed


Ehsas
Name
Zain Muneer
Email
muneerzain992@gmail.com
Phone
03218579419
Edit profile


Choose File No file chosen
Name
Zain Muneer
Email
muneerzain992@gmail.com
Verify Email
Phone
03218579419
Cancel
Volunteer Dasboard
Ehsas
Volunteer Elements
Dashboard
Volunteer
New Request
In Process
Complete Request
Volunteer New Request
Name
Phone Number
Address
Status
Action
No data available.


Volunteer Dasboard
Volunteer Elements
Dashboard
Volunteer
New Request
In Process
Complete Request
Volunteer New Request
Name
Phone Number
Address
Status
No data available.


Volunteer Dasboard
Volunteer Elements
Dashboard
Volunteer
New Request
In Process
Complete Request
Ehsas
Volunteer Complete Request
Name
Edition
Author
Status
The Hunger Games
1
Suzanne Collins
Active
Ethics
1234
Hamza
Active

Chapter 04:

Implementation and Test Cases

Chapter 4: Implementation and Test Cases

4.1 Implementation

4.1.1 Implementation Overview

Ehsas Hub is implemented using the MySQL database through XAMPP server for local development, along with Express.js, React.js, and Node.js for backend and frontend development. The AI recommendation system is built in Python using Flask. The system integrates multiple roles—donors, (needy), volunteers, and admins—with functionalities like book donation/request, OTP-based authentication, recommendation engine, and profile verification.

4.1.2 Introduction

In this chapter, we delve into the implementation of the **Ehsas-Hub** platform. We will cover the core components of the system that have been developed so far, focusing on the major algorithms implemented, such as the **Recommendation System**, and **Volunteer Coordination** functionalities. Additionally, we will describe the platforms, APIs, and libraries that were used in the system. This chapter will also discuss the test cases that validate the system's functionality, ensuring its performance, security, and reliability.

4.1.3 Prototype

The initial prototype of **Ehsas-Hub** has been developed to showcase the core functionalities of the platform. This prototype focuses on the primary use cases, such as user registration, book donation management, and personalized recommendations. It provides a basic structure for the system's user interface, backend logic, and database integration, demonstrating how different user roles (students, donors, volunteers, and admins) interact with the platform.

The prototype is built using the (**MySQL**, **Express.js**, **React.js**, and **Node.js**) and integrates key features like a hybrid recommendation system and volunteer task coordination.

4.1.4 Key Implementation Components

Frontend (React.js):

- **User Interfaces:** Dynamic views for different roles.
- **Routing:** Implemented using React Router DOM.
- **HTTP Requests:** Axios used for connecting frontend to backend APIs.

Form Validation:

- Passwords must include at least 8 characters, one uppercase, one lowercase, and one numeric digit
- All fields must be filled. Any missing field will trigger an error message.
- Book donation/request forms validate genre, edition (numeric only), image format (JPG/PNG only), and title length (minimum 3 characters).

Backend (Node.js & Express.js):

- RESTful APIs for login, registration, book management, and volunteering.
- Authentication: JWT and bcrypt for token-based secure login.
- Email Services: Node mailer for OTP email verifications.
- Profile Control: OTP verification before allowing any profile updates.
- Validations: Strong validation for registration (email, password format), login, donations, book requests, and mandatory field checks.

Recommendation System:

- **Algorithm:** Hybrid system using Neural Collaborative Filtering (NCF) .
- **Rationale:** NCF models latent user-book interactions, while cosine similarity handles explicit preferences.
- **Tech Stack & Libraries:** pandas, numpy ,flask, tensorflow, sentence-transformers, scikit-learn, faker, tf-keras, dotenv, openpyxl, sqlalchemy, pymysql, hf_xet.
- **Data Handling:** Cleaned and structured user interest data for model training.

Database (MySQL)

Tables

- **users**: Stores general users including students (needy). Fields: user_id (PK), name, email, phone, password, gender, address, image, status, preferred_genre, request_id (FK to needy), created_at, updated_at.
- **admin:** Stores administrators. Fields: id (PK), name, email, phone, password, gender, image, created_at, updated_at.
- volunteer: Stores volunteer data. Fields: id (PK), name, email, phone, password, address, image, gender, status, created_at, updated_at.
- **donor:** Book donors. Fields: id (PK), user_id (FK), volunteer_id (FK), book_name, genres, edition, author_name, image, status, created_at, updated_at.
- **books:** All books in the system. Fields: bookid (PK), title, author, covering, language, genres, rating.
- **needy:** Links students to their book requests. Fields: id (PK), user_id (FK), req_id, request_status, created_at, updated_at.
- **interactions:** Tracks recommendations and user-book interactions. Fields: interaction_id (PK), user_id (FK), book_id (FK), action.
- **feedback:** Collects user feedback. Fields: id (PK), name, email, message, created_at.
- **verify:** OTP verification store. Fields: id (PK), email, code, created_at.

Validation enforced through:

- NOT NULL constraints for mandatory fields
- Strong password policy
- ENUM values for controlled status and gender fields
- Foreign key constraints for relational integrity

4.2 Test Cases

4.2.1 Admin Test Cases

Table 4.1: Admin Test Case

ID	TEST CASES	PRECONDITIONS	INPUT DATA	STEPS	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Test Admin Registration Successfully	None	Name, Email, Phone, Password, Gender, Image	Fill registration form and press Submit	Admin account created and stored in DB	Admin created successfully	Pass
2	Test Admin Login Successfully	Admin must be registered	Correct Email and Password	Enter Email and press Login	Admin logged in successfully	Logged in successfully	Pass
3	Test Admin Login with Incorrect Password	Admin must be registered	Correct Email, Wrong Password	Enter Email and wrong Password → Press Login	Error message: Invalid credentials	Error displayed	Pass
4	Test Admin Login with Unregistered Email	None	Unregistered Email, Any Password	Enter Email → Press Login	Error: Email not registered	Error displayed	Pass
5	Test Admin Update Profile	Admin must be logged in	Updated Name, Phone, Image	Update fields → Press Save	Admin profile updated	Profile updated	Pass

4.2.2 Needy Test Case

Table 4.2: Needy Test Case

ID	TEST CASES	PRECONDITIONS	INPUT DATA	STEPS	EXPECTED RESULT	ACTUAL RESULTS	PASS/FAIL
1	Test user registration with valid data	User not registered	Correct name, email, phone, password, address, gender, image	Fill all fields and press Register	User is successfully registered	User registered successfully	Pass
2	Test user registration with duplicate email	Email already registered	Existing email, new name, phone, password	Enter duplicate email and press Register	"Email already exists" error should appear	Duplicate email error displayed	Pass
3	Test user login with correct credentials	User already registered	Correct registered email and password	Enter email/password and press Login	User logged in successfully	Login successful	Pass
4	Test user login with wrong password	User already registered	Correct email, wrong password	Enter email and wrong password, press Login	"Invalid credentials" error should appear	Login failed with error	Pass
5	Test update user profile information	User logged in	Updated phone or address	Change profile fields and verify email with OTP and save	Profile updated successfully	Profile updated successfully	Pass
6	Test registration with invalid email format	No user registered	Wrong email format	Enter wrong email and press Register	"Invalid Email Format" error should appear	Invalid email error displayed	Pass

7	Test user status field behavior	New registration	Correct user details	Register and check status field	Status should be Active (1)	Status set correctly	Pass
8	Test phone number field validation	No user registered	Phone number less than 10 digits	Enter short phone number and press Register	"Invalid Phone Number" error should appear	Phone validation error shown	Pass
9	Test password encryption	New registration	Correct user data	Register and check database password field	Password should be encrypted (hash)	Password saved encrypted	Pass

4.2.3 Volunteer Test Case

Table 4.3: Volunteer Test Case

ID	TEST CASES	PRECONDITIONS	INPUT DATA	STEPS	EXPECTED RESULT	ACTUAL RESULTS	PASS/FAIL
1	Test volunteer registration with valid data	Volunteer not registered	Correct name, email, phone, password, address, profile image	Fill all fields and press Register	Volunteer registered successfully	Volunteer registered successfully	Pass
2	Test volunteer registration with duplicate email	Email already registered	Existing email, new name, phone, password	Enter duplicate email and press Register	"Email already exists" error should appear	Duplicate email error displayed	Pass
3	Test volunteer login with correct credentials	Volunteer already registered	Correct email and password	Enter email/password and press Login	Volunteer logged in successfully	Login successful	Pass
4	Test volunteer login with incorrect password	Volunteer already registered	Correct email, wrong password	Enter correct email and wrong password	"Invalid Credentials" error should appear	Login failed with error	Pass
5	Test update volunteer profile	Volunteer logged in	Updated phone/address details	Change and save profile	Profile updated successfully	Profile updated successfully	Pass
6	Test delete volunteer	Volunteer present in database	Existing volunteer data	Press Delete on the volunteer	Volunteer deleted successfully	Volunteer deleted successfully	Pass
7	Test volunteer phone number validation	No volunteer registered	Phone number with letters or special chars	Enter invalid phone number and press Register	"Invalid Phone Number" error should appear	Validation error displayed	Pass
8	Test volunteer password strength	No volunteer registered	Weak password without special char, uppercase	Enter weak password and submit	"Weak Password" error should appear	Weak password error displayed	Pass
9	Test volunteer profile image upload	No volunteer registered	Correct image file (JPG/PNG)	Upload profile image and press Register	Image uploaded and saved	Image saved successfully	Pass

10	Test volunteer status after registration	New volunteer registration	Correct registration data	Complete registration and check status field	Status should be Active (1)	Status set correctly	Pass

4.2.4 Donor Test Case

Table 4.4: Donor Test Case

ID	TEST CASES	PRECONDITIONS	INPUT DATA	STEPS	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Test book donation with valid inputs	User logged in	Valid book name, genre, edition, author, image	Fill donation form → Submit	Book added to DB	Book saved	Pass
2	Test missing book name	Logged in	Leave book name empty	Submit form	Error: Book name required	Error shown	Pass
3	Test invalid genre	Logged in	Leave genre empty	Submit form	Error shown	Error displayed	Pass
4	Test edition field with text	Logged in	Enter "First" in edition	Submit	Error: Numbers only	Error shown	Pass
5	Test book image field	Logged in	Upload PDF instead of image	Submit	Error: Invalid format	Error shown	Pass
6	Test donation form reset	Book donated	Check if form resets	Submit → Check fields	Fields cleared	Cleared	Pass
7	Test minimum book title length	Logged in	Enter 2 characters	Submit	Error: Title too short	Error shown	Pass
8	Test cancel donation form	Logged in	Click Close on modal	Close the modal	Modal closes, no action	Closed	Pass

4.3 Test Metrics

4.3.1 Common Attributes of Test Case Metrics

Test case metrics provide a structured approach to evaluate the quality and performance of software testing. In Ehsas Hub, the following common attributes were used across all modules:

- **Total Number of Test Cases:** Indicates the overall coverage of testing across all modules and functionalities.
- **Test Case Pass Rate:** The ratio of test cases that passed successfully against the total executed.
- **Test Case Failures:** Number of tests that did not meet expected outcomes, helping identify bugs or logic flaws.
- **Defect Density:** Represents the percentage of test cases that failed out of the total executed, calculated as
  Defect Density = (Failed Test Cases / Total Test Cases) * 100.
- **Test Case Effectiveness:** Measures the proportion of test cases that successfully detected defects, calculated as
  Effectiveness = (Defects Found by Tests / Total Defects) * 100.
- **Traceability Matrix:** Ensures that each requirement is linked to corresponding test cases to verify that all features are tested and validated.
- **Validation Checks:** Common validation logic (e.g., non-empty fields, password complexity, file format, numeric inputs) was standardized and reused across different test forms.

To provide comprehensive and consistent testing coverage, these metrics were used uniformly throughout the admin, user, volunteer, and book donation/request modules.

4.3.2 Test Summary Table

Table 4.5: Test Summary Table

METRIC	DESCRIPTION	VALUE
TOTAL TEST CASES	Combined across all modules	37
PASSED	All test cases executed successfully	37
FAILED	-	0
TEST CASE EFFECTIVENESS	(37/37) *100	100%
DEFECT DENSITY	(0/37) *100	0%

4.4 Conclusion

This chapter discussed the Ehsas Hub implementation process, including the technical architecture, component breakdown, and project-wide validation techniques. Our backend makes use of MySQL, which has stringent database and code validation guidelines. Complexity criteria are enforced by password validation. Security for profile updates is ensured by OTP verification. To better match recommendations with user preferences, the recommendation system employs a hybrid approach. Every aspect of the system passed the first functional testing with 100% efficacy, demonstrating that Ehsas Hub is safe, scalable, and designed with the goal of enabling students to access educational materials with ease.

Chapter 05:

Experimental Results and Analysis

Chapter 5: Experimental Results and Analysis

5.1 Introduction

This chapter presents the experimental setup, performance evaluation, and result analysis of our application "Ehsas Hub". Ehsas Hub is a platform that links administrators, volunteers, needy users, and contributors to donate and suggest books. Validating the efficacy of key features such user interaction flow, account approval procedures, book donation/request processing, and the hybrid recommendation system is the goal of these trials. In order to guarantee correctness, usability, and dependability, we additionally assess platform performance in a variety of user roles and scenarios.

5.2 Experimental Setup

5.2.1 Platform Performance Evaluation

Objective

To evaluate each role's essential characteristics and user experience from start to finish, admin, volunteer, needy, and donor.

Environmental tools

- **Device Used:** Dell Latitude Laptop
- **Specifications:** 16gb Ram 512 SSD Core i5 8th gen
- **Network:** 4G , Nayatel Wi-Fi
- **Internet Speed:** 3-5 Mbps
- **Software:** Ehsas-Hub Web Application (Mobile responsive)

Table 5.1: Functional Performance Evaluation

TEST PROCEDURE	ACTION	EXPECTED TIME	ACTUAL TIME	RESULT	NOTES
ACCOUNT REGISTRATION	Sign up with details, genre, and email OTP	≤ 5 sec	6 sec	90%	Network dependent
ADMIN ACCOUNT APPROVAL	Admin dashboard accepts new users	Instant	Instant	100%	Works as intended
LOGIN	Enter email/password	≤ 5 sec	3 sec	100%	Secure and smooth
DONATE BOOK	Fill form and submit book	≤ 5 sec	5-6 sec	90%	Image upload takes time
REQUEST BOOK	Choose and request a book	≤ 5 sec	4 sec	100%	Success confirmation email
VOLUNTEER ACCEPT REQUEST	Volunteer accepts pickup nearby	≤ 3 sec	3-4 sec	95%	Needs location optimization
VIEW RECOMMENDED BOOK	Browse all books	≤ 4 sec	3 sec	100%	Smooth rendering
EDIT PROFILE / LOGOUT	Update info / logout	≤ 3 sec	2 sec	100%	No issues found

5.2.2 Recommendation System Effectiveness

Objective

To test the hybrid recommendation system, which uses Neural Collaborative Filtering (NCF) and content-based filtering using BERT embedding is for suggesting books, based on preferred genres and interaction history.

Environment & Tools

- **Libraries Used:** pandas, numpy ,flask, tensorflow, sentence-transformers, scikit-learn, faker, tf-keras, dotenv, openpyxl, sqlalchemy, pymysql, hf_xet
- **Backend:** Flask REST API
- **Database:** MySQL with SQLAlchemy
- **Synthetic Data:** User interactions generated via Faker

Table 5.2: Recommendation System Evaluation

TEST COMPONENT	ACTION	EXPECTED OUTCOME	ACTUAL OUTCOME	ACCURACY	NOTES
GENRE-BASED SUGGESTIONS	Display relevant books after login	Relevant book list shown	90% match	90%	Based on initial signup genre
INTERACTION LEARNING	Recommend based on Search, likes, requests	Personalized suggestions	85% accuracy	85%	Improves over time
RESPONSE TIME	Load recommendations	≤ 5 sec	4-5 sec	100%	Acceptable speed under load
COLD START TEST	New user with no interactions	Genre-only based suggestions	80% match	80%	Initial fallback to genre model

5.2.3 Authentication and Security

Objective

To confirm the safe and effective operation of the password reset and edit profile of any user, OTP verification, and login functions.

Environment & Tools

- **Device Used:** Dell Latitude Laptop
- **Specifications:** 16gb Ram 512 SSD Core i5 8th gen
- **Software:** Web Frontend with email services (Mailer linked to Ehsas-Hub domain)

Table 5.3: Authentication Metrics

FEATURE	ACTION	EXPECTED TIME	ACTUAL TIME	SUCCESS RATE	NOTES
EMAIL OTP VERIFICATION	Register + receive code	≤ 2 min	1.5 min	100%	Code received on Gmail
LOGIN AUTHENTICATION	Email/password login	≤ 5 sec	2-3 sec	100%	Token stored securely
FORGOT PASSWORD FLOW	Request Forgot password	≤ 3 min	2.5 min	100%	Secure via email confirmation
VERIFY EMAIL ON EDIT PROFILE	Verify email OTP before profile is updated	≤ 2 min	1.8 min	100%	OTP ensures security for user changes

5.3 Conclusion

All of the Ehsas Hub platform's modules—registration, book donation/request, volunteer coordination, and personalized recommendations—show excellent functioning and user satisfaction, according to the experimental research. When it came to genre-based and interaction-based recommendations, the hybrid recommendation algorithm achieved up to 90% accuracy. With secure email-based verification, user approval and authentication processes operated effectively. While there is need for improvement in terms of response times and volunteer location optimization, the platform is reliable and prepared for practical use. These findings support Ehsas Hub's usefulness in expediting book contributions via an ecosystem powered by technology.

Chapter 06:

Conclusion and Future Directions

Chapter 6: Conclusion and Future Directions

6.1 Introduction

The main objective of Ehsas Hub's conception and development was to provide a centralized, AI-assisted platform that would accelerate volunteer and admin coordination, allow book contribution, and enhance needy kids' ability to access learning resources. This chapter offers a thorough summary of the findings from the implementation process, evaluations of every element, and suggestions for further improvement. We review our achievements and consider areas that might use improvement.

6.2 Achievements and Improvements

Throughout the development of Ehsas Hub, several technical and operational milestones were achieved that validate the robustness and feasibility of the platform:

6.2.1 Front-End Achievements:

- **User Experience Optimization:** React.js was used to design an intuitive and responsive user interface for multiple roles (admin, donor, volunteer, and Needy).
- **Validation Enhancements:** All forms enforce strong password rules, mandatory field checks, and file type validations, improving data consistency and security.
- **Modular Navigation:** Seamless routing between modules such as donation, registration, login, and feedback has been established using React Router.

6.2.2 Backend Achievement's

- **Secure Authentication:** JWT-based login with password encryption (bcrypt) and OTP verification using Node mailer ensures secure user operations.
- **Role-Based Functionality:** Each role accesses specific APIs designed to maintain operational clarity and data segregation.
- **API Validation:** Express-validator ensures structured input validation across all endpoints.

6.2.3 Recommendation Engine:

- **Hybrid Model Integration:** We developed a Neural Collaborative Filtering (NCF) model for personalized book recommendations.
- **Successful Training and Evaluation:** The system uses actual interaction data to fine-tune suggestions, boosting usability.

6.2.4 Database Enhancements:

- **MySQL with XAMPP:** Relational schema designed with foreign keys, NOT NULL constraints, and ENUM types to maintain integrity.
- **Modules Covered:** Admin, Users, Donors, Volunteers, Feedback, Book Interactions, and OTP Verification modules were fully developed and interconnected.
- **Detailed ERD Mapped:** Relationships and constraints were implemented exactly as mapped in the ER diagram.

6.3 Critical Review

The Ehsas Hub platform tackles the issues of needy empowerment, donation transparency, and book accessibility. The creation of a full-stack platform with multi-role support and integrated AI recommendation was part of the scope.

6.3.1 Strengths

- **Social Impact Focused:** Aimed at educational upliftment using technology
- **Machine Learning Integration:** Used modern algorithms for personalized learning support.
- **Secure and Scalable:** Clean modular codebase and robust authentication mechanisms.

6.3.2 Weaknesses:

- **UI Aesthetics:** Visual design could benefit from improved styling and user interaction cues.
- **Performance Optimization:** Database queries can be further optimized for high concurrency.
- **Limited Real-Time Updates:** Chat or live support functionalities were not included but could enhance coordination.

- **Google Map:** Use Map location will be Helpful to enhance the Donor location for pick up.

6.4 Future Recommendations

For Future, the following future improvements and scope extensions are proposed:

6.4.1 Enhancements to Current Modules:

- Improve UI with animated transitions and better visual hierarchies.
- Add file format previews for book cover uploads.
- Provide status tracking for donation and request submissions.

6.4.2 Additional Features:

- **Chat System:** Real-time communication between users and volunteers.
- **Mobile Application:** Flutter-based cross-platform app for accessibility.
- **Gamified Volunteering:** Add badges and leaderboards to motivate volunteers.
- **Feedback Analytics:** Automatically categorize user feedback using NLP.

6.4.3 Future Specific Work Plan:

- Implement real-time notification system (Node.js + Socket.io).
- Deploy the application using cloud services (e.g., Vercel/Heroku for frontend, Render for backend).
- Conduct user testing in real environments (e.g., colleges, libraries).

6.5 Conclusion

In conclusion, Ehsas Hub successfully met its objectives, including secure user registration, book donation workflows, multi-role access, and personalized book recommendations. All core functionalities were implemented and tested with 100% success rate in unit testing. The project offers a scalable base for further work and meaningful social contribution. Future directions include the expansion of features,, UI improvement, integration of real-time components, and deployment for public use. With a clear roadmap for Future, Ehsas Hub stands ready for refinement and broader impact in the educational tech domain.

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA: Springer, 2011.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[3] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[4] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 2009.

[5] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[6] K. R. Seeja and M. Zareapoor, "FraudMiner: A novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, Article ID 252797, 2014.

[7] M. Omokaro, P. C. Zhang, and E. H. Chi, "Volunteerism through self-tracking and personal informatics: How quantified self data can support civic engagement," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2015, pp. 4873–4882.

[8] S. Wu, J. M. DiMicco, and D. R. Millen, "Detecting professional versus personal closeness using an enterprise social network site," in *Proc. 32nd ACM Conf. Human Factors in Computing Systems*, 2010, pp. 1955–1964.

[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182.

[10] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Systems (RecSys)*, 2016, pp. 191–19