The Library Management

Project Overview

The Library Management System (LMS) is designed to efficiently manage the operations of a library. It includes two separate websites: one for clients (students) and one for administrators (admins). The system handles various functions such as managing books, book categories, clients, and borrowing transactions.

Entities and Features

Admin

Admins are responsible for managing all aspects of the library system. The key features and tasks include:

1. **Manage Books:**
    - **Add New Books**: Admins can add new books by entering details such as title, author, ISBN, category, and availability status.
    - **Update Books**: Admins can update the information of existing books.
    - **Delete Books**: Admins can remove books from the library system.
    - **Display All Books**: Admins can view a comprehensive list of all books in the library.
    - **Search Books**: Admins can search for books by title, author, or ISBN.

2. **Manage Students**:
    - **Add New Students**: Admins can add new student records including details such as name, student ID, contact information, and address.
    - **Update Student Information**: Admins can update the details of existing students.
    - **Delete Students**: Admins can remove student records from the system.
    - **Display All Students**: Admins can view a comprehensive list of all students.
    - **Search Students**: Admins can search for students by name or student ID.

3. **Manage Book Categories:**
    - **Add New Categories**: Admins can add new categories for books.
    - **Update Categories**: Admins can update the details of existing categories.
    - **Delete Categories**: Admins can remove categories from the system.
    - **Display All Categories**: Admins can view a comprehensive list of all book categories.
    - **Search Categories**: Admins can search for categories by name.

4. **Manage Borrowed Books:**
    - **Record Borrowed Books**: Admins can record when a book is borrowed by a student, including due dates.
    - **Update Borrowed Book Records**: Admins can update the borrowing records.
    - **Delete Borrowed Book Records**: Admins can remove borrowed book records once the book is returned.
    - **Display All Borrowed Books**: Admins can view a list of all borrowed books along with their due dates.
    - **Search Borrowed Books**: Admins can search for borrowed books by student name, book title, or due date.

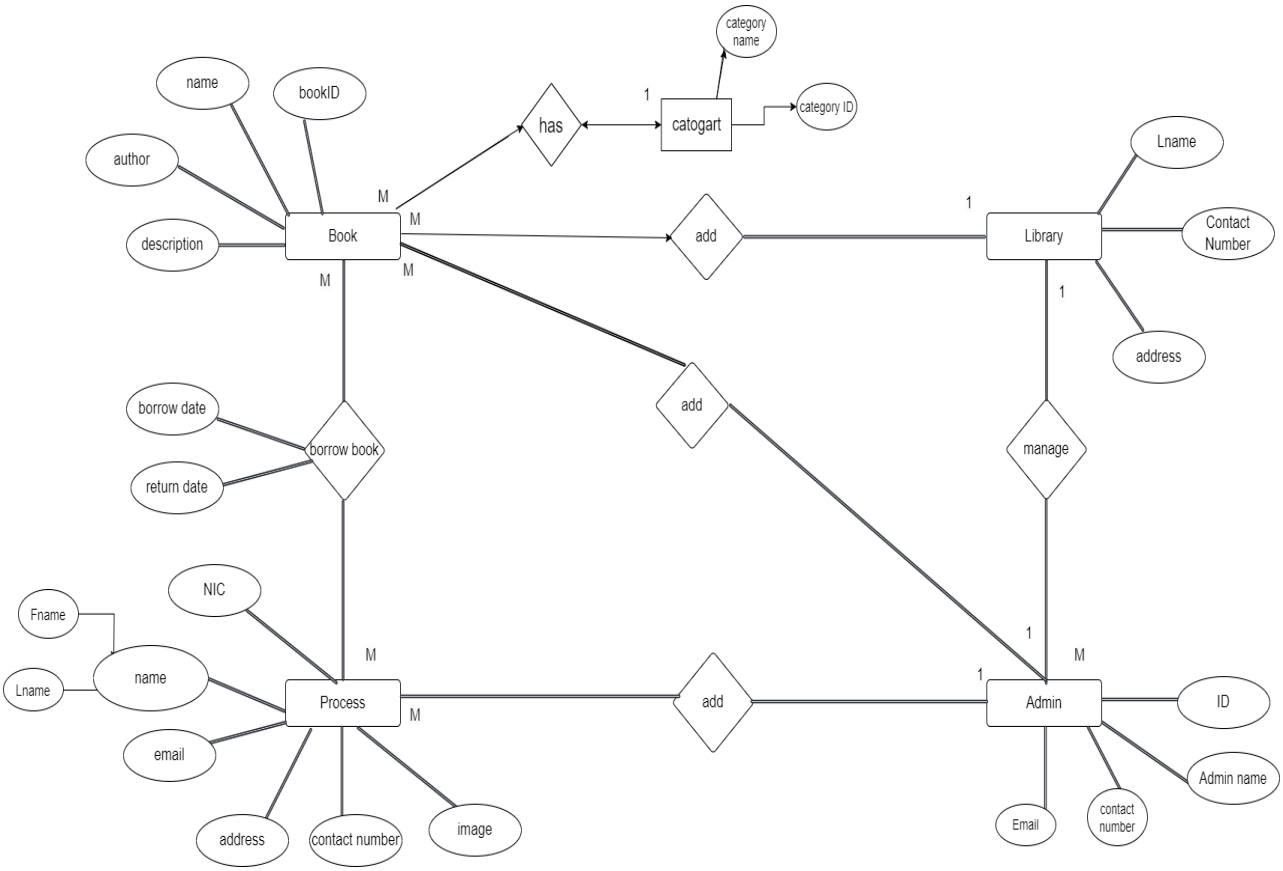
5. **Manage Other Admins:**
    - **Add New Admins**: Super admins can add new admin users.
    - **Update Admin Information**: Super admins can update the details of existing admins.
    - **Delete Admins**: Super admins can remove admin users from the system.
    - **Display All Admins**: Super admins can view a comprehensive list of all admins.
    - **Search Admins**: Super admins can search for admins by name or admin ID.

Client (Student)

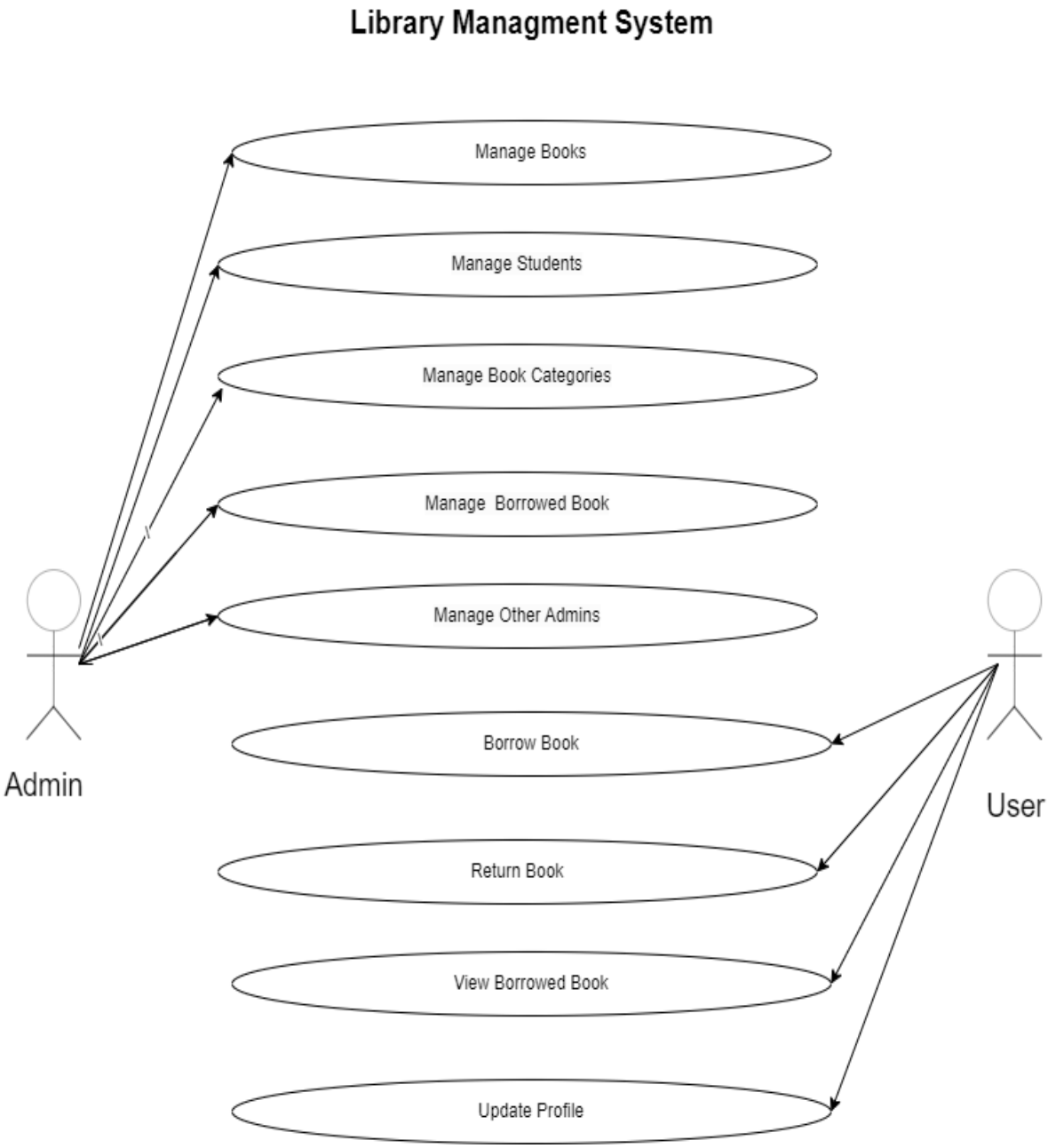
Clients (students) can perform a variety of actions related to their library usage. The key features and tasks include:

1. **Show Books:**
    - **View Available Books**: Students can view the list of books available in the library.
2. **Borrowing History:**
    - **View Borrowing History**: Students can view the list of books they have borrowed along with the due dates.
3. **Client's Profile:**
    - **Update Profile Image**: Students can update their profile image.
    - **Update Profile Password**: Students can update their profile password.

ER Diagram:


category name
name
bookID
has
1
catogart
category ID
author
M
Lname
1
Book
M
add
Library
Contact Number
description
M
M
1
address
add
borrow date
borrow book
manage
return date
NIC
Fname
1
M
M
1
name
Lname
Process
add
Admin
ID
M
email
Admin name
address
contact number
image
Email
contact number

User Diagram:



Requirements Analysis

1.1 Functional Requirements

- **Admin Functionalities:**
    1. Manage Books:
        - Add, update, delete, display, and search books.
    2. Manage Students:
        - Add, update, delete, display, and search student records.
    3. Manage Book Categories:
        - Add, update, delete, display, and search book categories.
    4. Manage Borrowed Books:
        - Record, update, delete, display, and search borrowed book records.
    5. Manage Admins:
        - Add, update, delete, display, and search admin users.
- **Student Functionalities:**
    1. Show Books:
        - View available books.
    2. Borrowing History:
        - View borrowing history and due dates.
    3. Client's Profile:
        - Update profile image and password.

1.2 Non-Functional Requirements

- **Performance Requirements:**
    - The system should handle simultaneous requests from multiple users.
- **Security Requirements:**
    - User authentication for both admin and student portals.
    - Data encryption for sensitive information.
- **Usability Requirements:**
    - User-friendly interfaces for both portals.
- **Reliability Requirements:**
    - Regular data backups and recovery options.

Plan

Testing Plan

1.1 Types of Testing

- **Unit Testing:** Test individual components of the system.
- **Integration Testing:** Test the interaction between different components.
- **System Testing:** Test the complete system for defects.
- **Acceptance Testing:** Validate the system against user requirements.

1.2 Test Cases

- **Admin Portal:**
  - Add new book and verify its addition.
  - Update book details and verify updates.
  - Delete a book and verify deletion.
  - Search for a book and verify results.
  - Add, update, delete, and search students.
  - Add, update, delete, and search categories.
  - Record, update, delete, and search borrowed books.
  - Add, update, delete, and search admin users.

- **Student Portal:**
  - View available books.
  - View borrowing history.
  - Update profile image and password.

Deployment Plan

2.1 Deployment Steps

- **Step 1:** Prepare the production environment.
- **Step 2:** Deploy the backend server.
- **Step 3:** Deploy the frontend applications.
- **Step 4:** Configure the database.
- **Step 5:** Perform system testing in the production environment.
- **Step 6:** Go live and monitor the system for any issues.

2.2 User Training and Support

- **User Manuals:** Provide detailed user manuals for both admins and students.
- **Training Sessions:** Conduct training sessions for admin users.
- **Support:** Offer ongoing support and maintenance for the system.

Implementation

The Library Management System (LMS) is structured into several components including frontend views, backend controllers, utility classes, models, and services. This section provides an overview of the implementation details, ensuring maintainability and scalability.

1.1 Project Structure

The project is divided into the following key directories and files:

- **Views:**
  - Contains JSP files for the frontend user interface.
  - Subdirectories for Admin and Client views.
- **Source Packages:**
  - Contains Java packages for controllers, utilities, models, and services.

1.2 Frontend Implementation

The frontend is implemented using JSP (JavaServer Pages). Key directories include:

- **styles:**
  - admin.css: Stylesheet for admin pages.
  - client.css: Stylesheet for client pages.
  -
- **views:**
  - **Admin:**
    - Admin_manage: Managing admins (admin_form.jsp, show_admins.jsp).
    - Books_Categories: Managing book categories (show_bookcategories.jsp).
    - borrowed_books: Displaying borrowed books (display_borrowed_books.jsp).
    - books: Managing books (adminAddBook.jsp, adminBook.jsp, adminEditBook.jsp, adminSearchBook.jsp).
    - home: Admin home page.

        - students: Managing students (show_students.jsp, student_form.jsp).
        - admin_page_template.jsp: Admin page template.
    - **auth:**
        - login_form.jsp: Login form for authentication.
    - **client:**
        - clientCommon: Common components like footer and header.
        - home: Client home page.
        - my_books: Viewing borrowed books.
        - profile: Managing client profile.
    - **common:**
        - common_url.jsp: Common URLs for the application.
    - index.jsp: Main entry point of the application.

1.3 Backend Implementation

The backend is implemented using Java. Key packages and classes include:

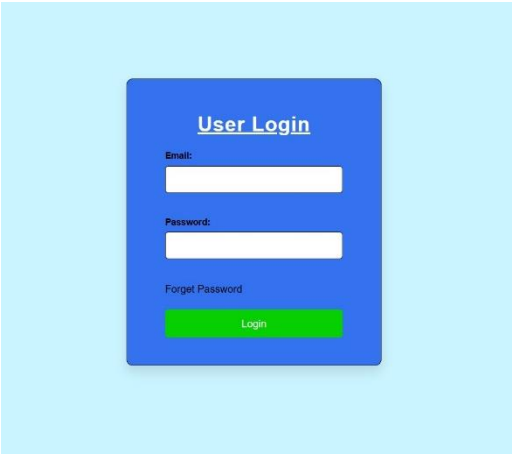
- **com.team2.controller.admin:**
    - Manages various admin functionalities (books, students, categories, borrowed books, admins).
- **com.team2.controller.auth:**
    - AuthController.java: Handles authentication processes.
- **com.team2.controller.client:**
    - Manages client functionalities (home, books, profile).
- **com.team2.controller.util:**
    - Utility classes for database connections, image paths, password generation, and asset URLs.
- **com.team2.models:**
    - Models for borrowed books, book categories, books, and users.
- **com.team2.service:**
    - Services for authentication, borrowed books, book categories, books, and users.
- **com.test:**
    - Test.java: Contains test cases for the application.

1.4 Implementation Details

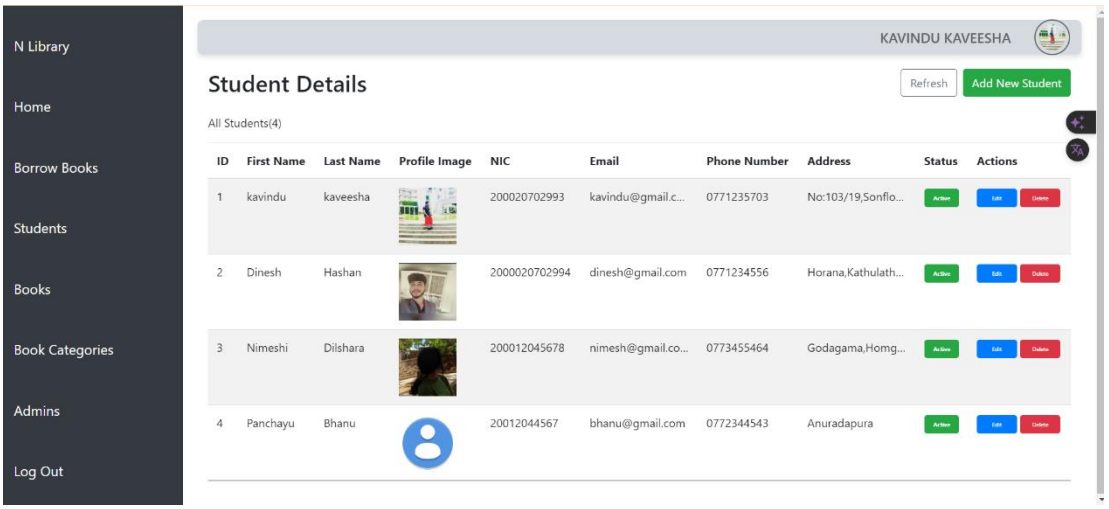
- **Admin Portal Implementation:**
  - **Book Management:** Handled by AdminManageBooksController.java.
  - **Student Management:** Handled by AdminStudentController.java.
  - **Category Management:** Handled by AdminManageBooksCategoriesController.java.
  - **Borrowed Books Management:** Handled by AdminManageBorrowedBooksController.java.
  - **Admin User Management:** Handled by ManageAdminsController.java.
- **Client Portal Implementation:**
  - **Book Viewing:** Handled by HomeController.java.
  - **Borrowing History:** Handled by MyBooksController.java.
  - **Profile Management:** Handled by ProfileController.java.
- **Authentication Implementation:**
  - AuthController.java handles login and logout functionality.
- **Utility Classes:**
  - DBConnection.java manages database connections.
  - Utility functions provided by PasswordGenerator.java and RandomBookIdGenerator.java.
- **Models and Services:**
  - Models define data structure and relationships.
  - Services implement business logic and interact with the database through models.

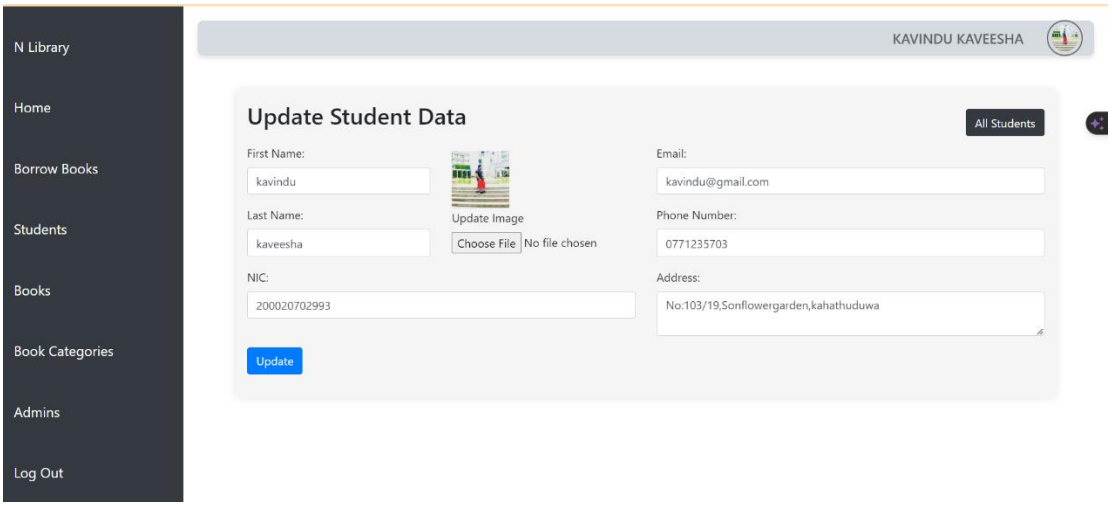
Design

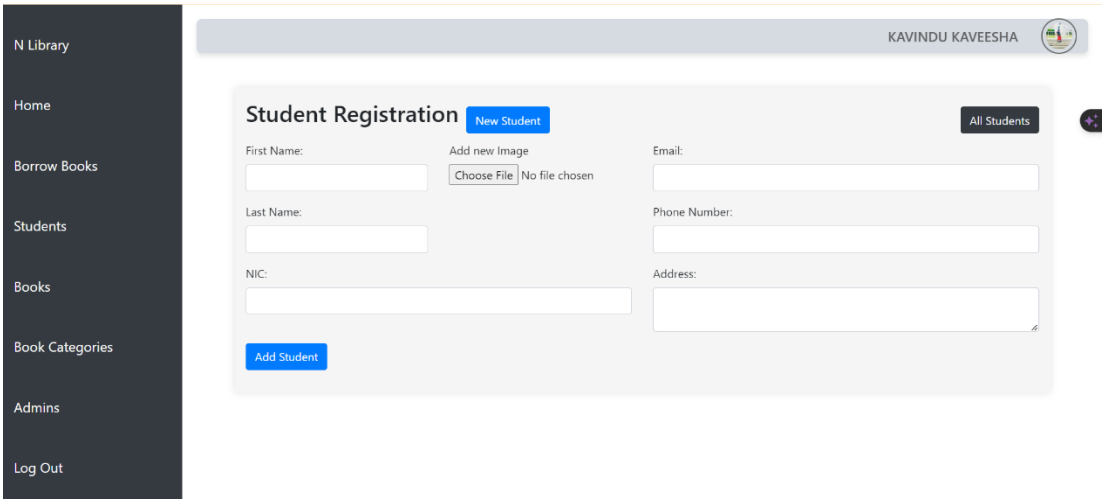
Login



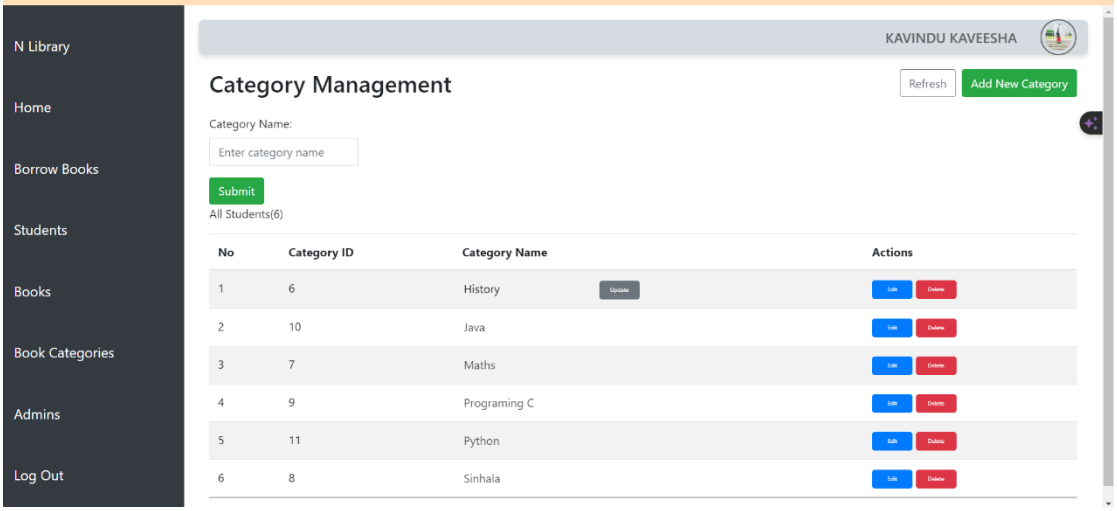
Student Management

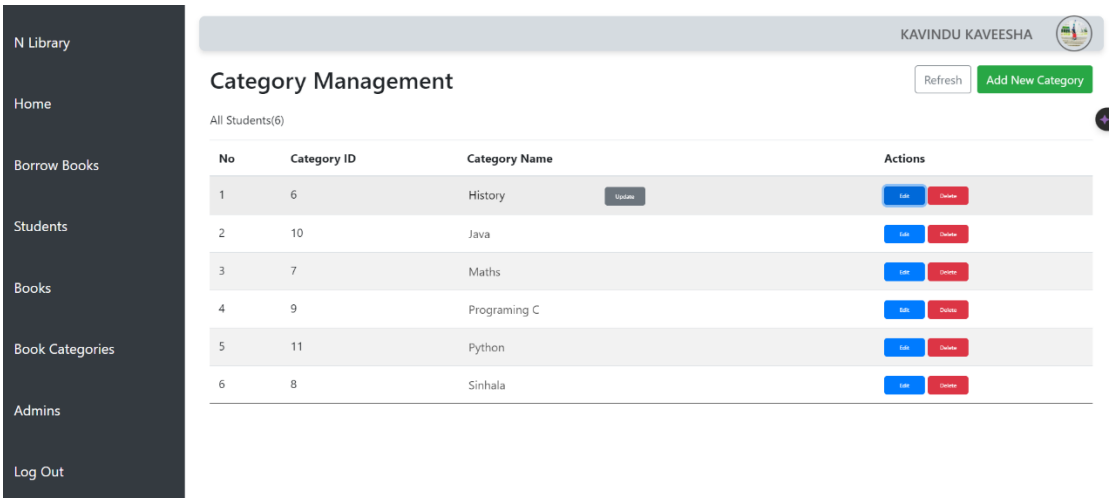


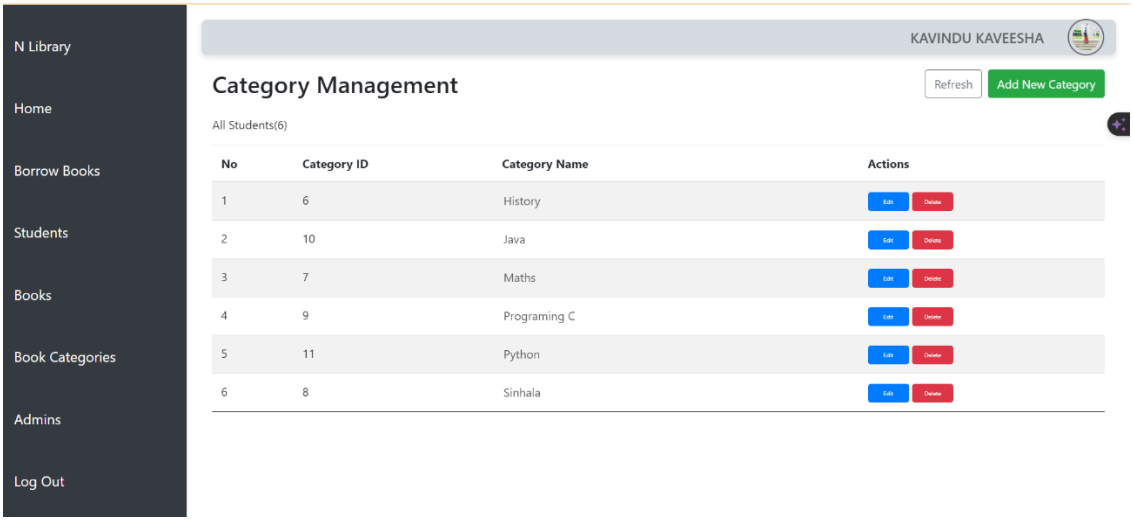




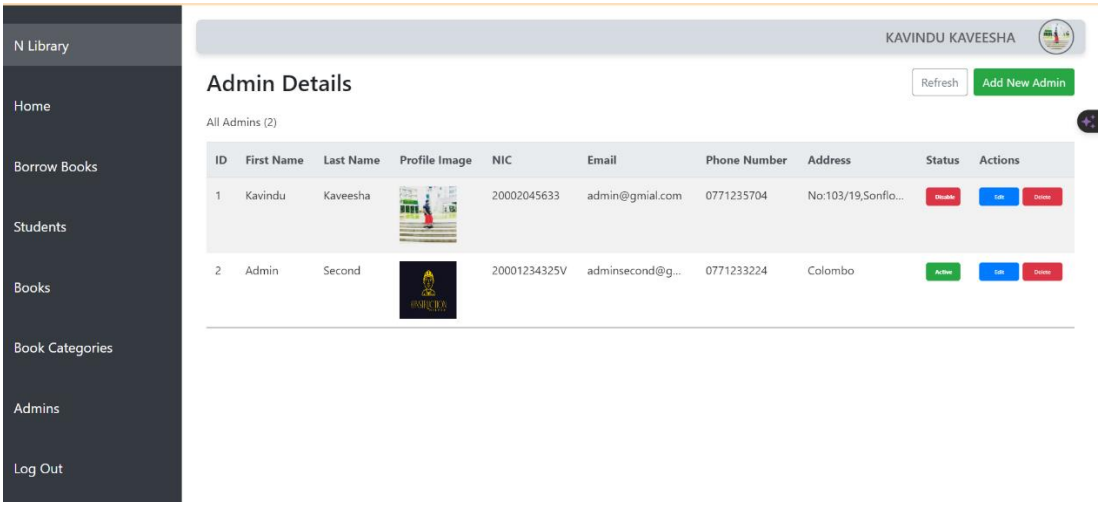
Book Categories

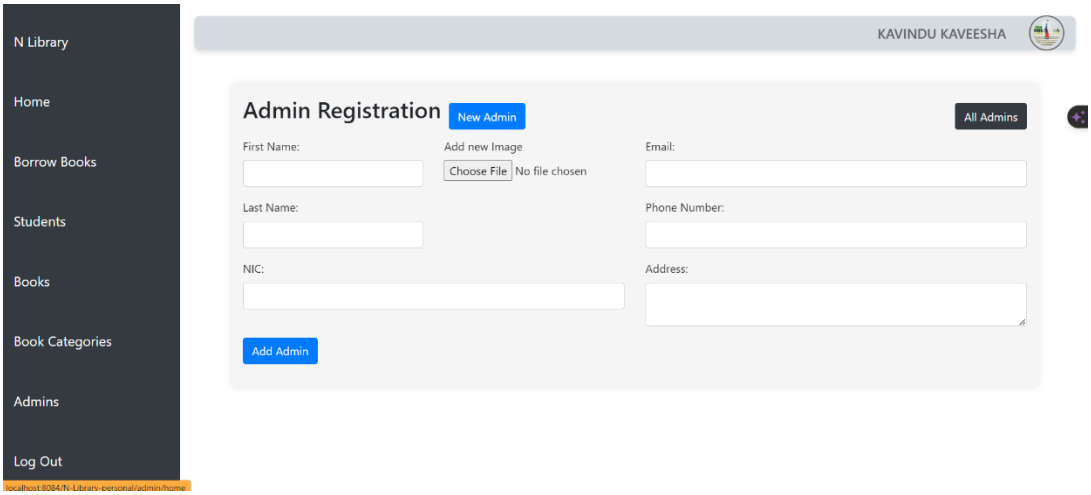


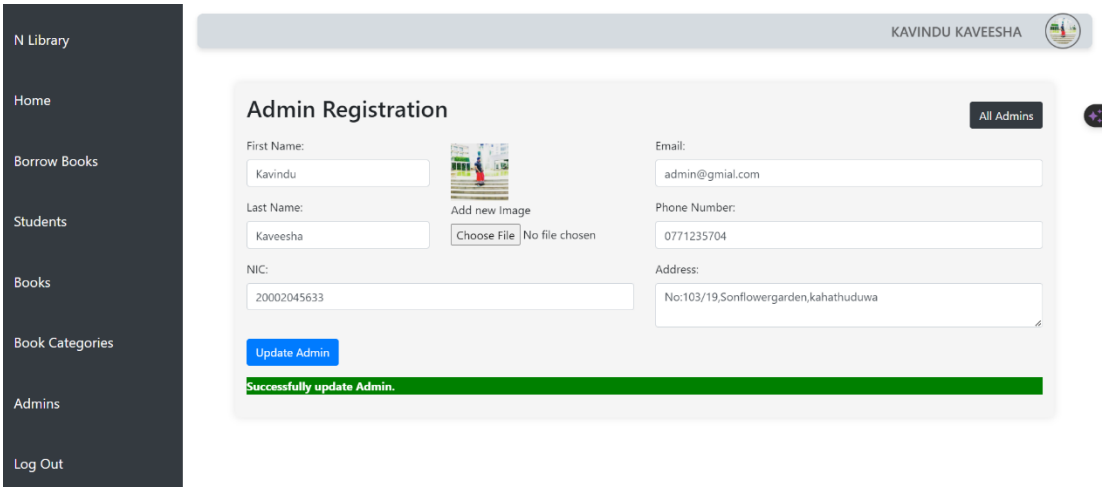




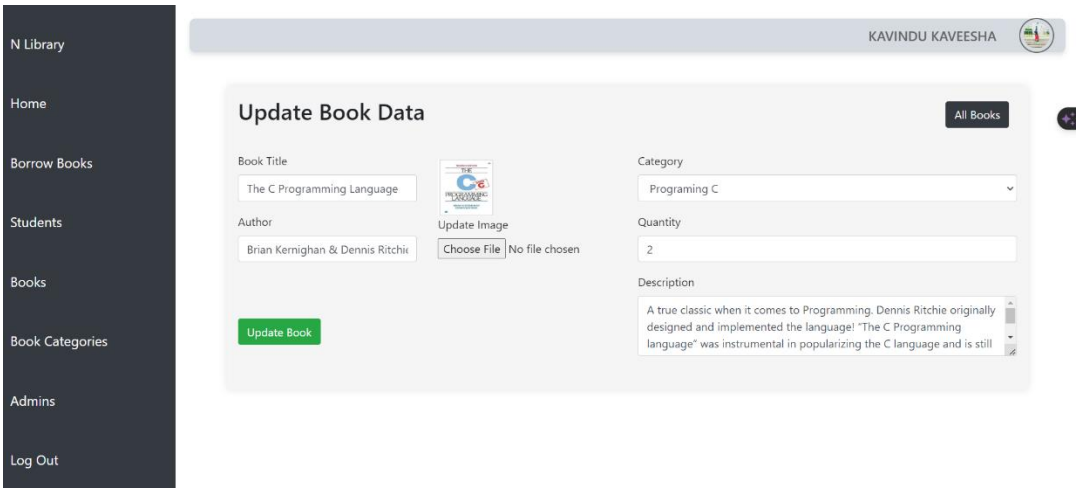
admin management

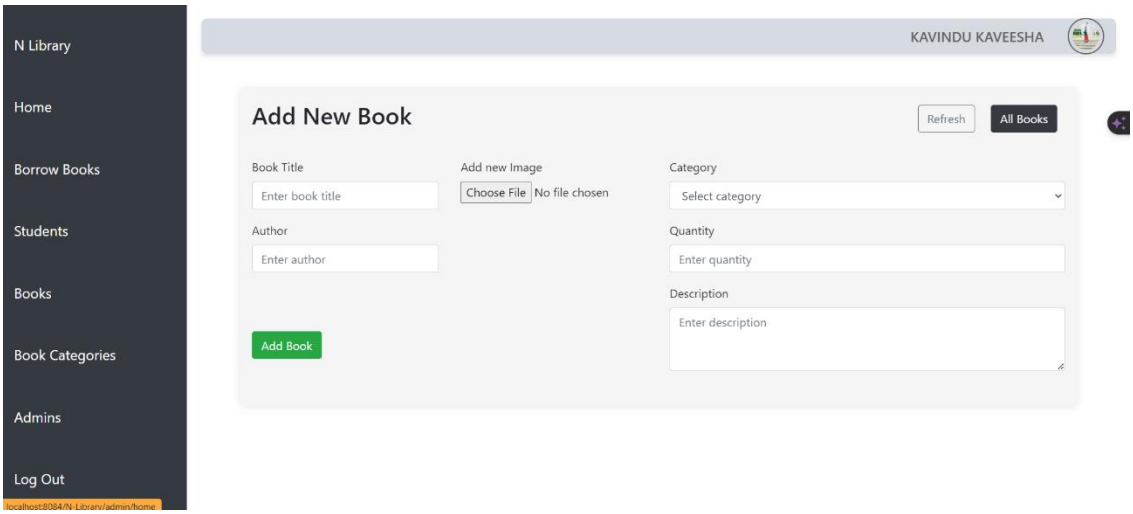


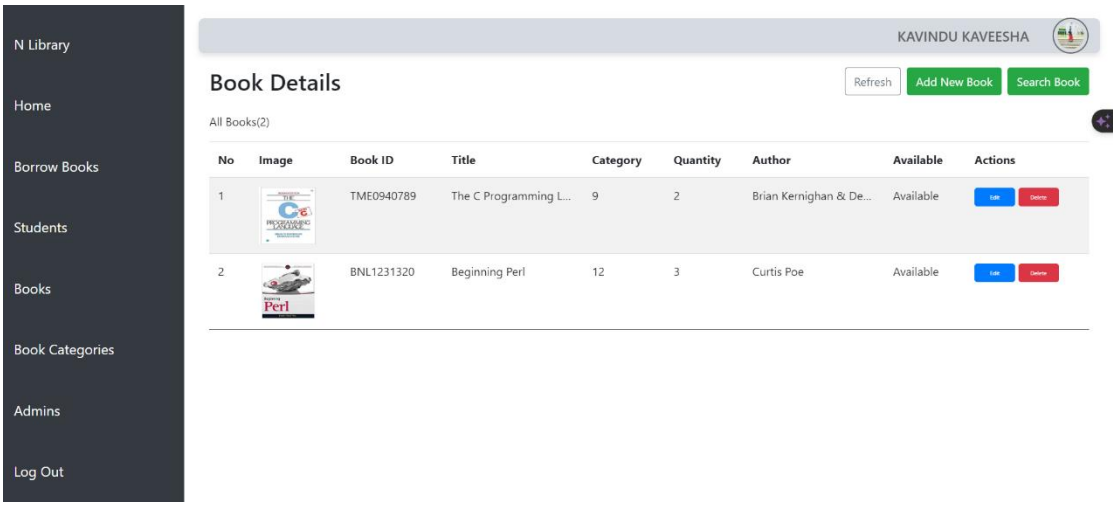




Manage Books







Barrow Books