**Project Title**: BookKeeper

**Project Description**: BookKeeper is a Java EE web application that simulates a book management system developed with user authentication, CRUD(Create, Read, Update, Delete) operations, and admin privileges. Users can view books in the database, checkout, and return books while admins manage users, books, and other admins accounts.

**Technologies used**:

1. Java servlets
2. Java Server Pages(JSP)
3. JDBC
4. CSS
5. Mysql

BookKeeper is implemented using the Model-View-Controller(MVC) design pattern.

**Features**:

* Implemented a login and registration page for users. In the login page, users will input username, password, and select whether they are logging in with a regular user or admin account. In the registration page, users can create regular user accounts by inputting a unique username, creating and confirming a password.
* A user account can view books, checkout books, and return books they checked out.
  + In the view book page, all books in the database will be displayed and the user can search for specific books based on book id, title, author last name or first name, and genre.
  + In the checkout page, users can checkout a particular book, if they have not already checked out one, by book id or by title, author last name, author first name, and genre.
  + In the return page, all the books the user checked out will be displayed, each book will have a “return” button that will return the book when clicked.
* An admin account can view books similar to regular user accounts, but admins can also view/remove user accounts, add new books into the system, remove old books, create or remove new or old admin accounts.
  + In the remove user page, all non-admin user accounts will be displayed with a “Remove user” button next to each user. Clicking the remove user button will return all books the user has checked out before finally removing the user account from the system.
  + In the add new book page, admins can add new books into the system, if the book does not already exist in the system, by inputting the new book’s title, author last name, author first name, and genre to a form.
  + In the remove book page, admins can remove books from the system, if they exist in the system, by inputting the book’s id or the book’s title, author last name, author first name, and genre.
  + In the add/remove admin page, admins can create new admin accounts or remove existing admins. However, the admin cannot remove the admin account they are currently signed into.

**Installation Instructions**:

**Prerequisites**:

1. Eclipse IDE for Java EE Developers (Version 2021-03 or later)
2. Apache Tomcat 8.5 or later
3. Mysql Server 5.7 or later
4. Mysql workbench 8.0 or later

**Installation**:

1. Download and install Eclipse IDE for Java EE Developers: https://www.eclipse.org/downloads/packages/release/2021-03/r/eclipse-ide-enterprise-java-and-web-developers
2. Download Apache Tomcat 8.5: https://tomcat.apache.org/download-80.cgi
3. Extract the downloaded Apache Tomcat package to a directory of your choice
4. Download and install MySQL server and MySQL Workbench: https://dev.mysql.com/downloads/

**Setup Database**:

1. Open MySQL Workbench and create a new schema (database).
2. Execute the SQL scripts bookdb\_books.sql, bookdb\_user\_book\_checkout.sql, bookdb\_users.sql to create the required tables and populate the database with sample data.

When the SQL scripts are executed, it will populate the database with three sample user accounts and 1 sample admin account that can be used:

User accounts:

1. username: johndoe, password: abc12345
2. username: sarahsmith, password: xyz12345
3. username: bookworm115, password: npr123zr

Admin account:

1. username: admin1, password: 12345678

**Import project into Eclipse**:

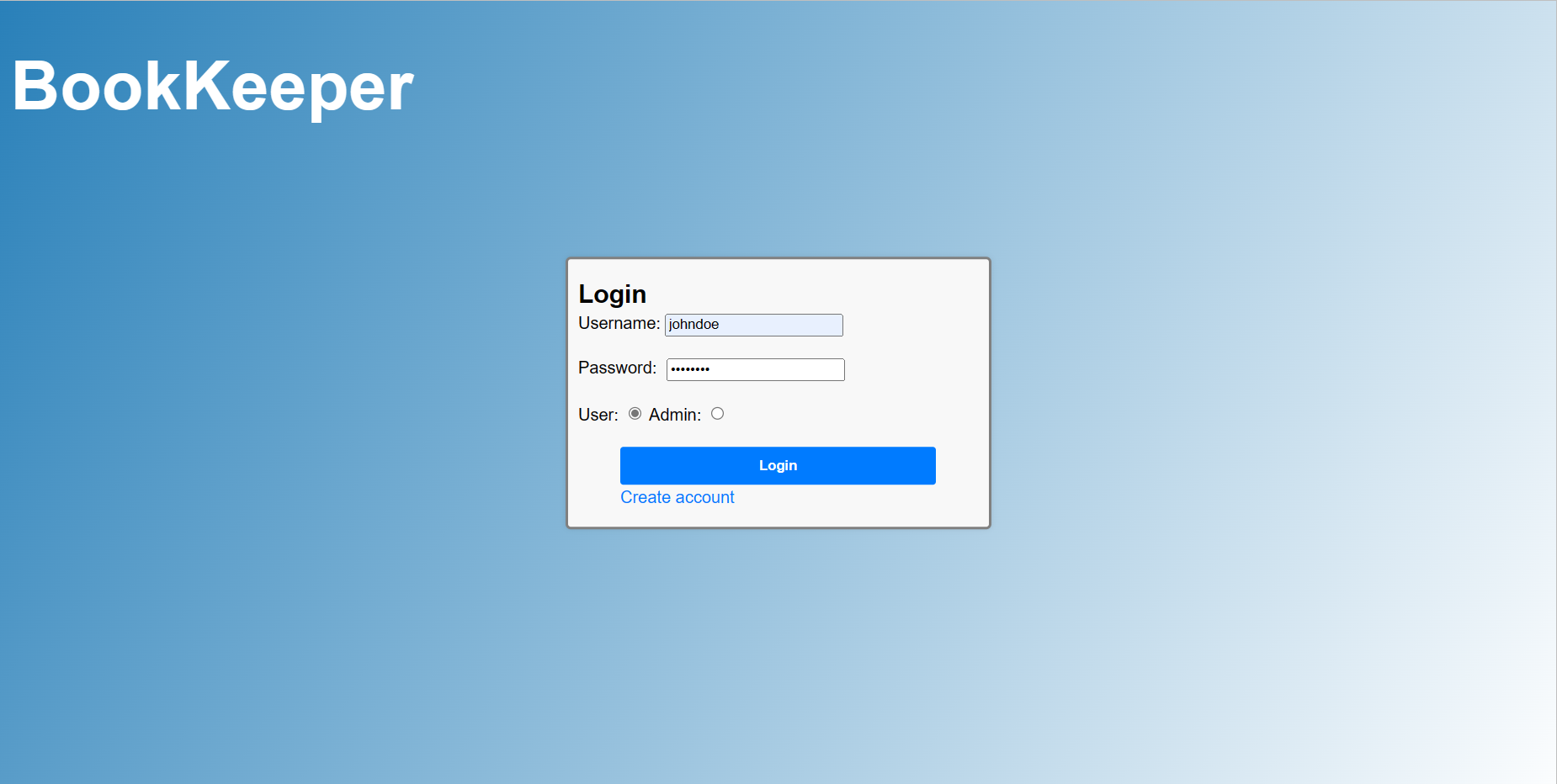
1. Open Eclipse IDE and select “File” > “Import”.
2. In the import dialog box, expand the “General” folder and select “Existing Projects into Workspace”.
3. Choose the directory where the project BookManagementSystem is.

**Configure Tomcat Server**:

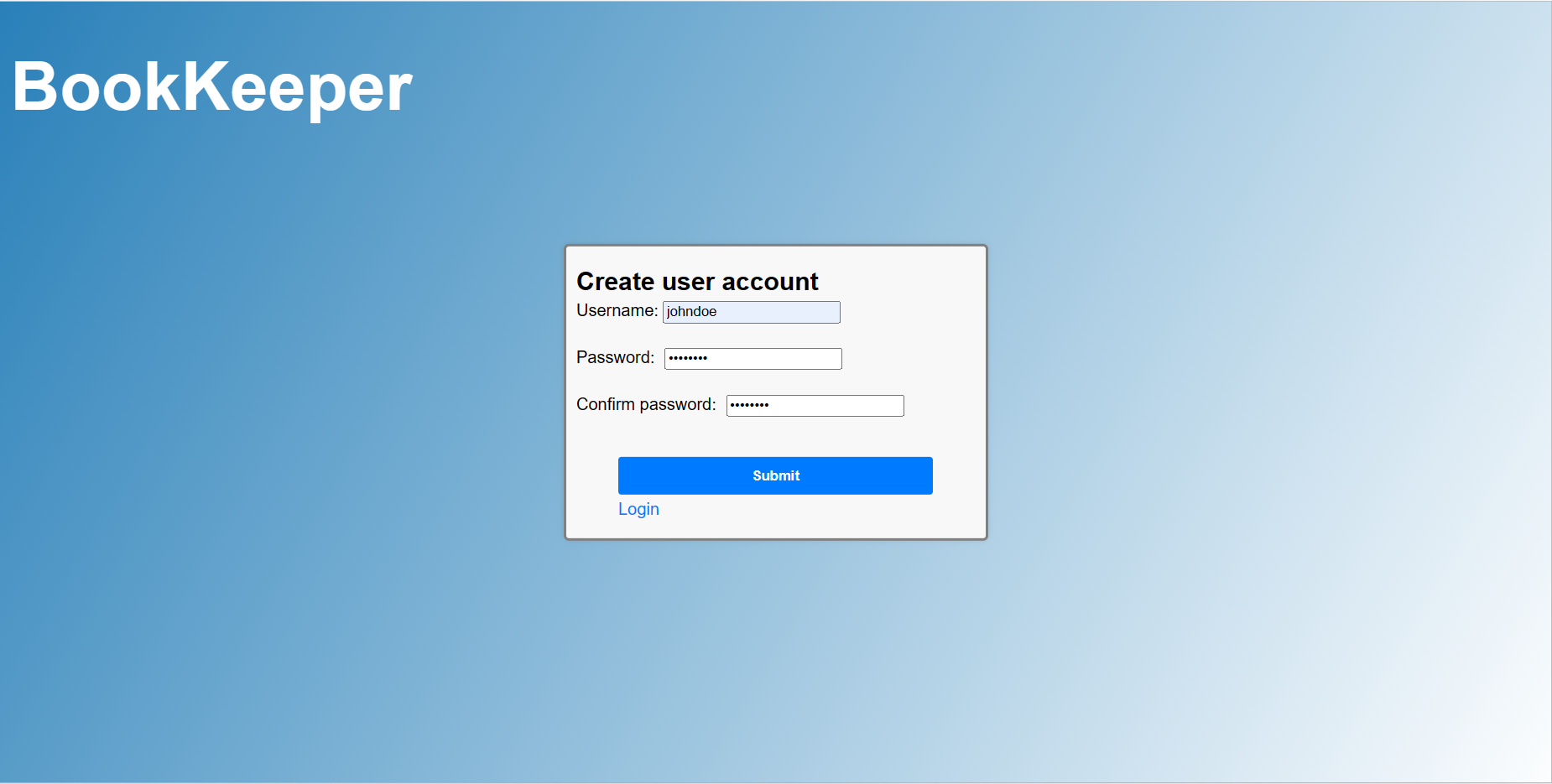
1. In Eclipse, select “Window” > “Preferences” from the menu bar.
2. Expand the “Server” section and select “Runtime Environments”.
3. Click “Add” to create a new Tomcat server runtime environment.
4. Choose “Apache Tomcat v8.5” as the server type and click “Next”.
5. Browse to the directory where you extracted the Apache Tomcat and finish.
6. In the “Servers” view, right-click and select “New” > “Server”.
7. Choose “Apache Tomcat v8.5 Server” as the server type and click “Next”.
8. Select the Tomcat installation directory and click “Finish”

**Images**:

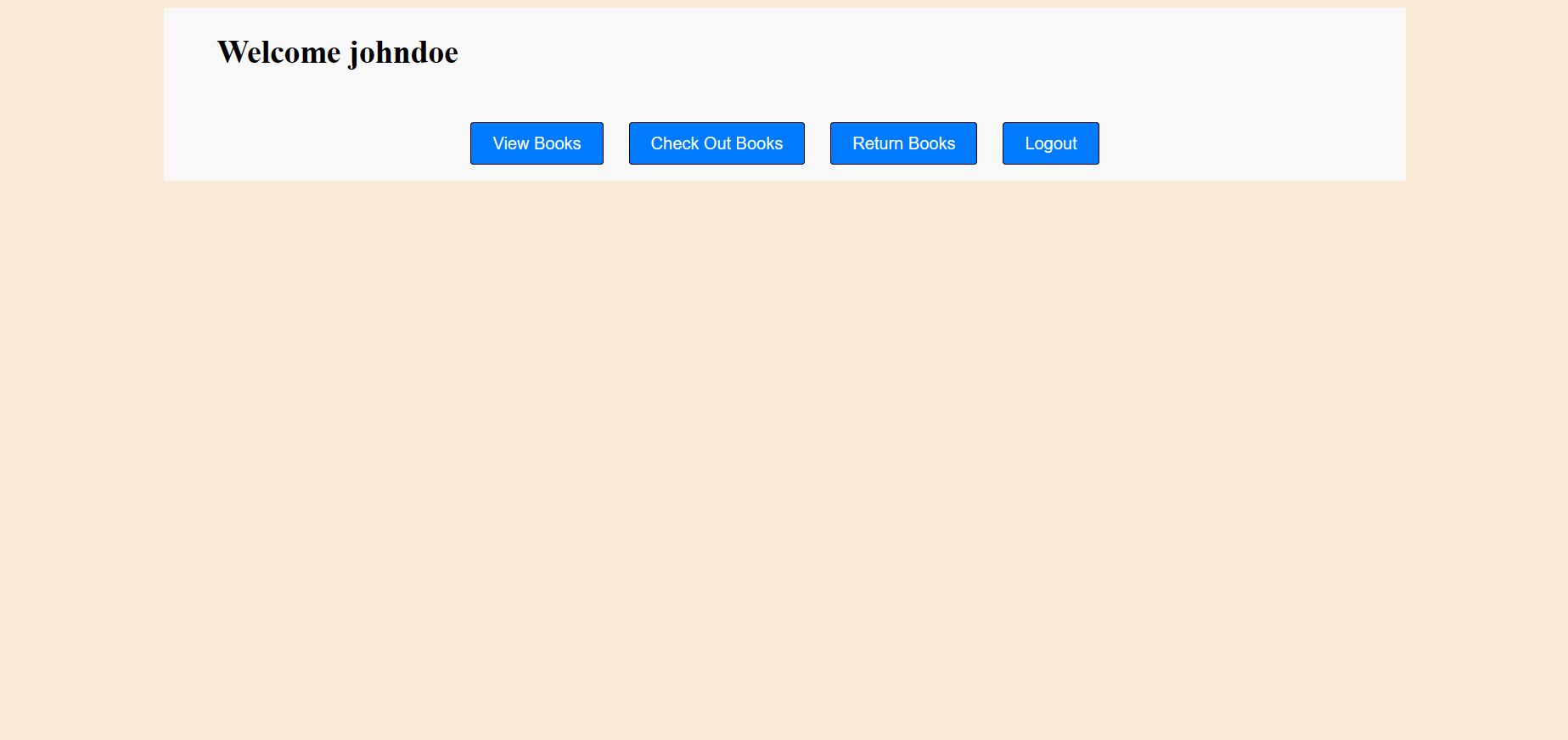
Login page:



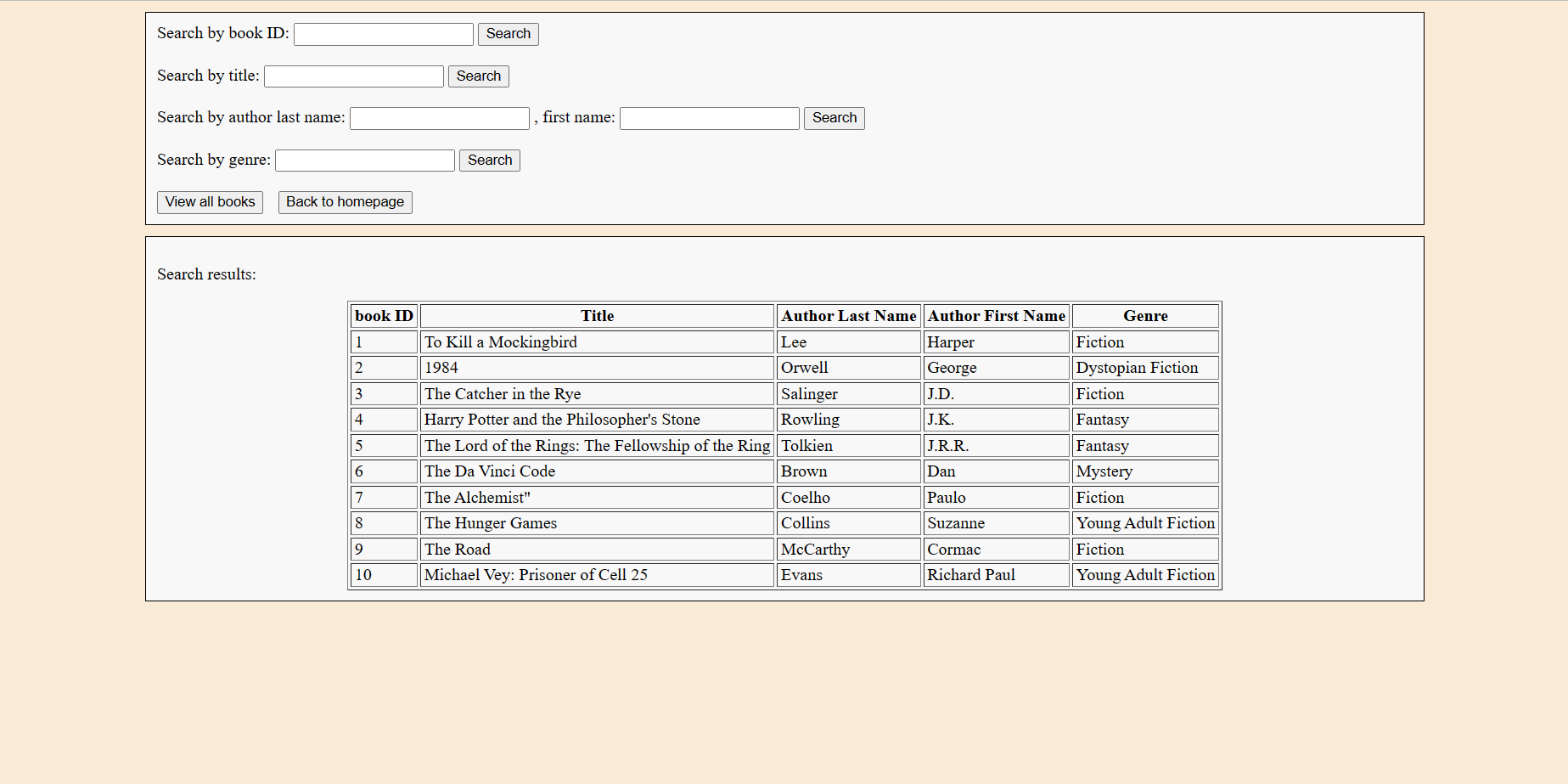
Registration page:



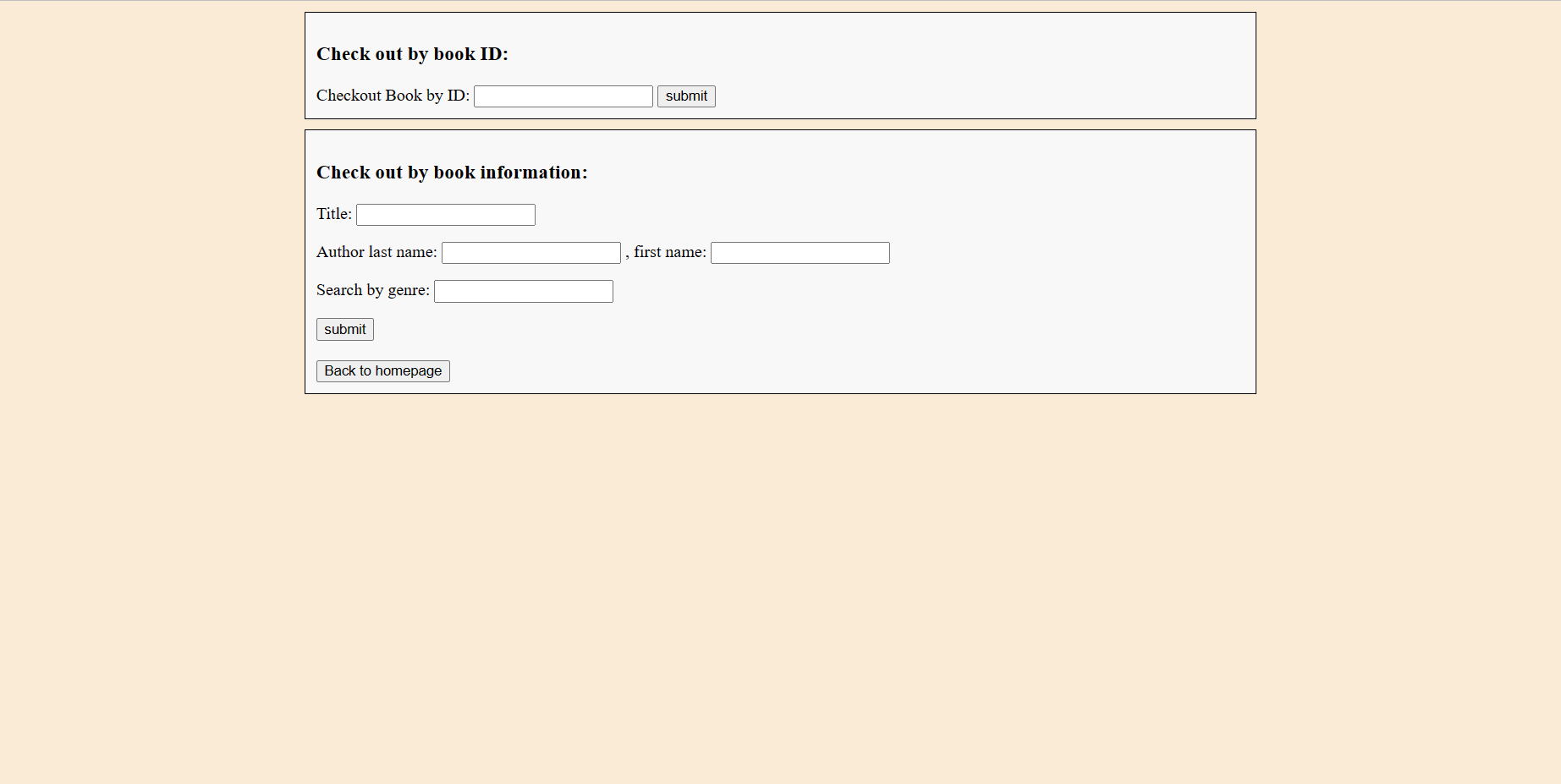
User home page:



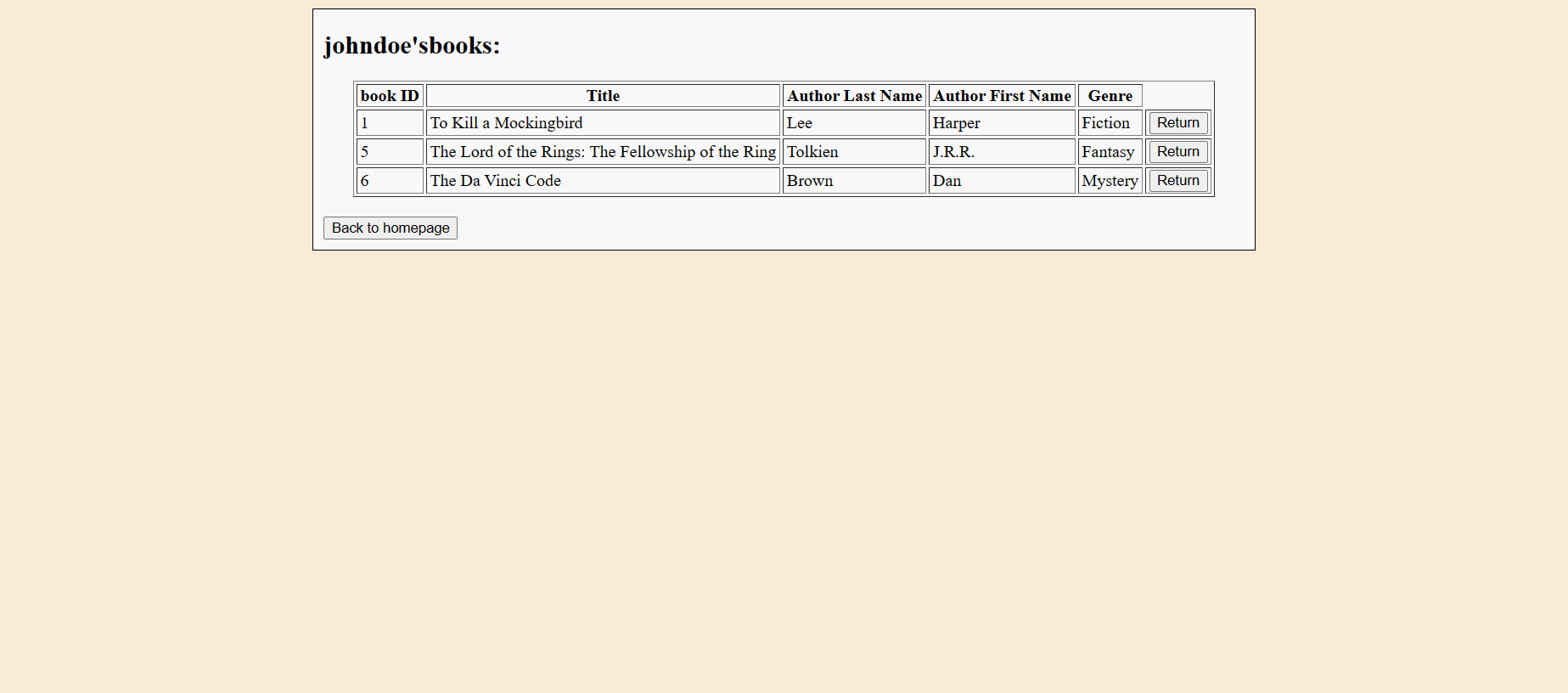
View book page:



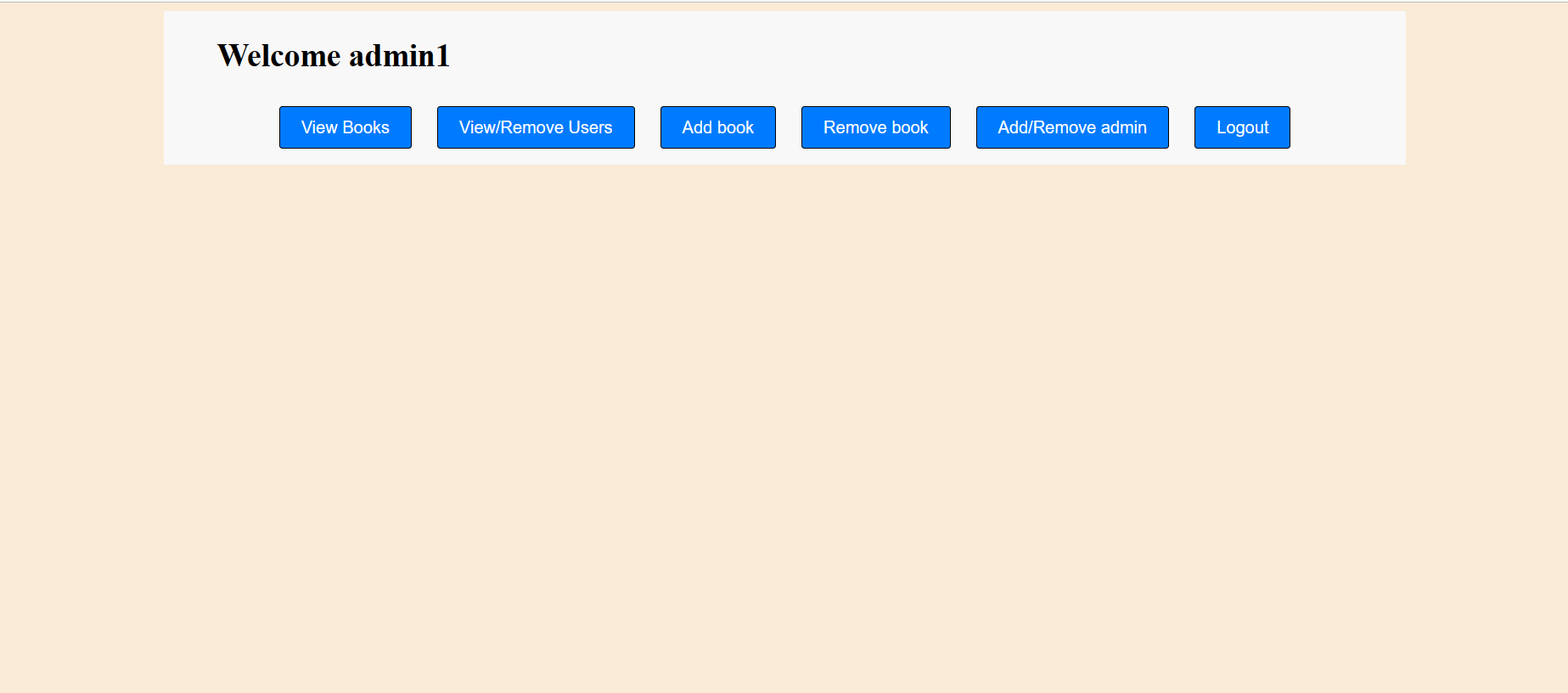
Check out book page:



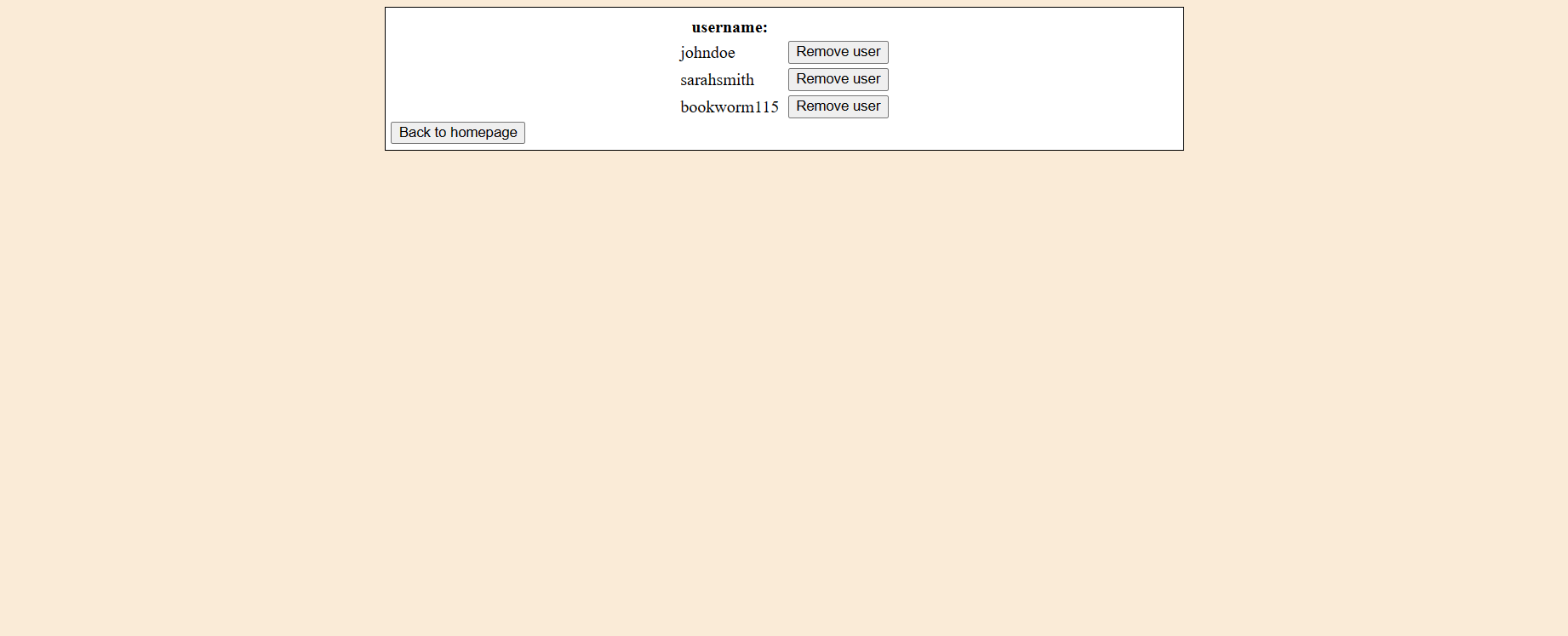
Return book page:



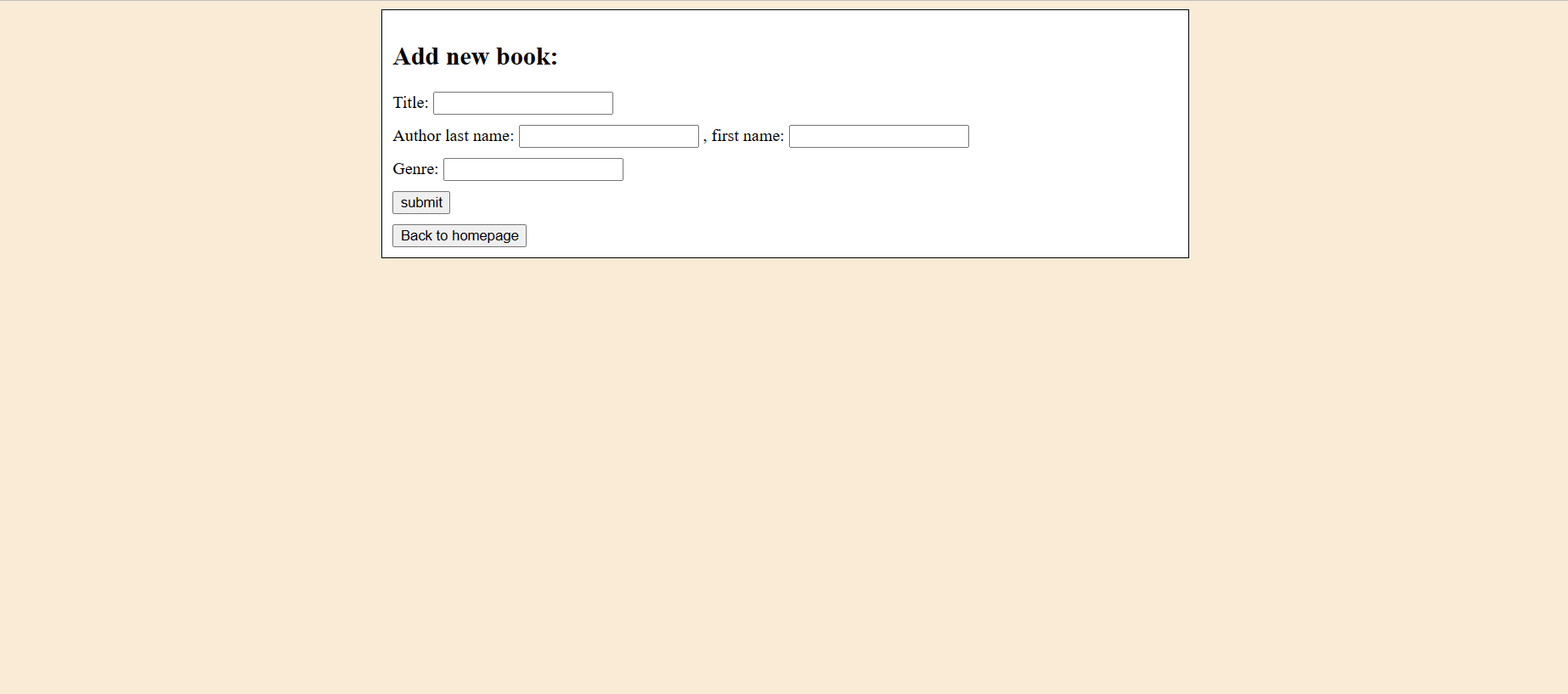
Admin home page:



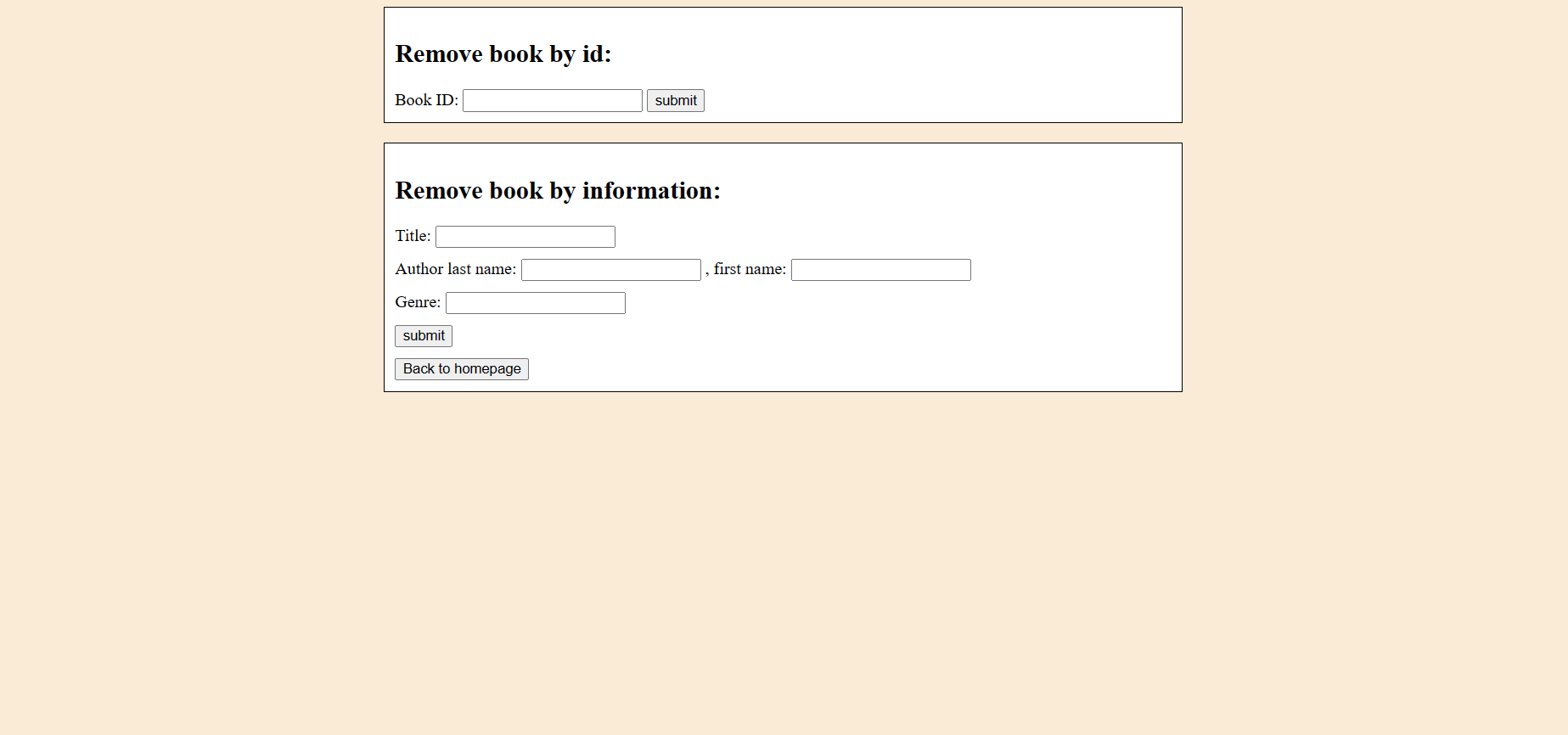
Admin view/remove user page:



Admin add new book page:



Admin remove book page:



Admin add/remove admin page:

