

Project Report: Online book borrowing Website

Part 1: Introduction

Overview:

- **Project Topic:** Online library
- **Project Use:** Assist users (library users and librarians) in quickly finding the books they need
- **Project Type:** Website
- **Technologies:** HTML and CSS (frontend); Python and Flask (backend); SQLite (database)

Background:

The project aims to help librarians borrow and search books efficiently. Through a simple web-based borrowing and searching system, library users and librarians can easily access and update information on existing books.

Part 2: Project Plan

Below is an overview of the project plan with each part and its status:

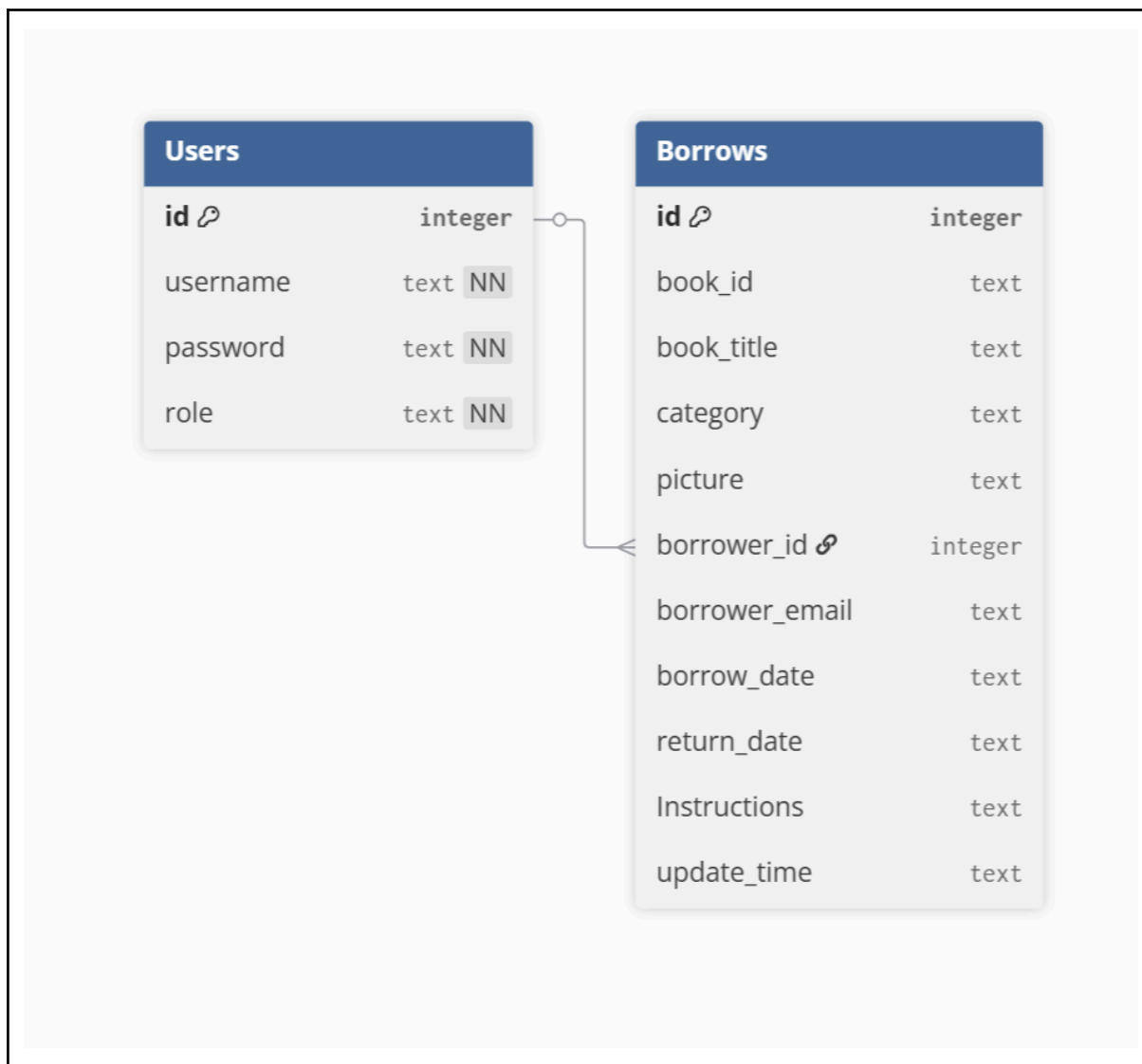
Part	Main Content	Tool Type	What's Included	Completion
Part 1	Database Creation and Data Entry	SQLiteStudio	- Database design diagram - Creation of tables: Books, Authors, BorrowRecords, Users - Query views and sample data	Completed

Part 2	Backend Code Production	Python and Flask	<ul style="list-style-type: none"> - Flask application setup - Database connection and API development - Routing and business logic 	Completed
Part 3	Frontend Code Production	HTML	<ul style="list-style-type: none"> - Web page templates (Home, Search, Details pages) - Using Jinja2 to display data from the backend 	Completed
Part 4	Web Page Styling	CSS	<ul style="list-style-type: none"> - Overall website styling - Layout, fonts, colors, and responsive design 	Completed
Part 5	Testing	Manual Testing	<ul style="list-style-type: none"> - Overall system functionality testing - Bug fixing and performance optimization 	In Progress
Part 6	Update new content	All of the above tools	<ul style="list-style-type: none"> - Add new content and update - Final summary 	In Progress

Part 3: Visual Evidence

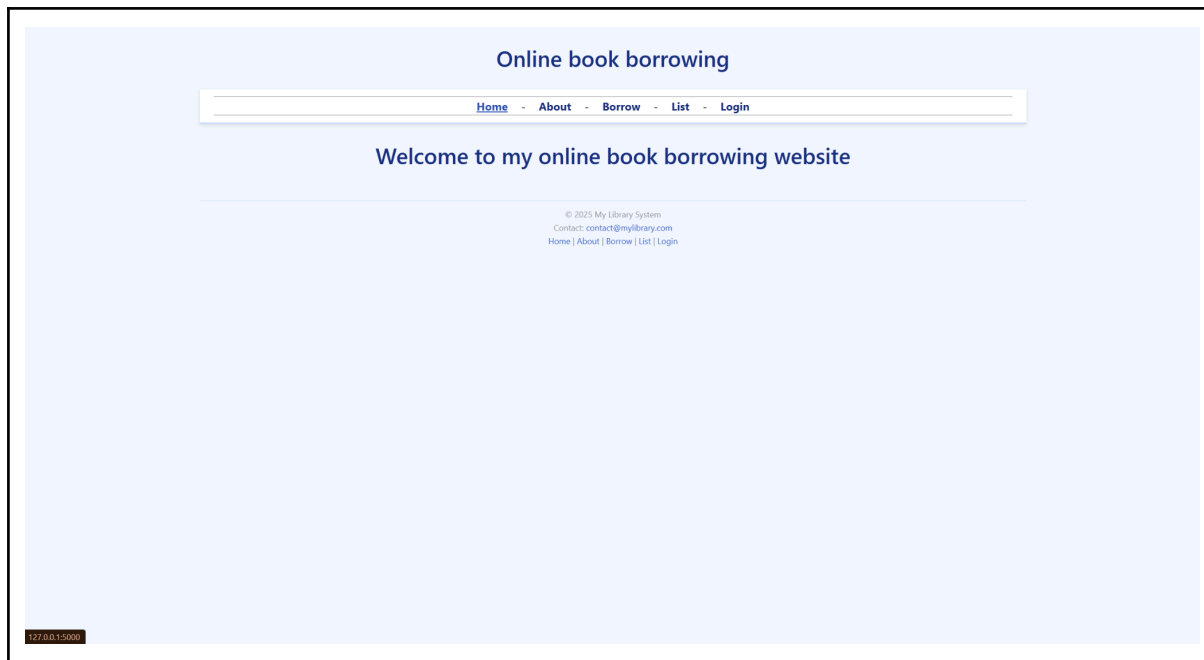
Screenshot 1: Database design diagram

Description: This screenshot show the Database connection diagram



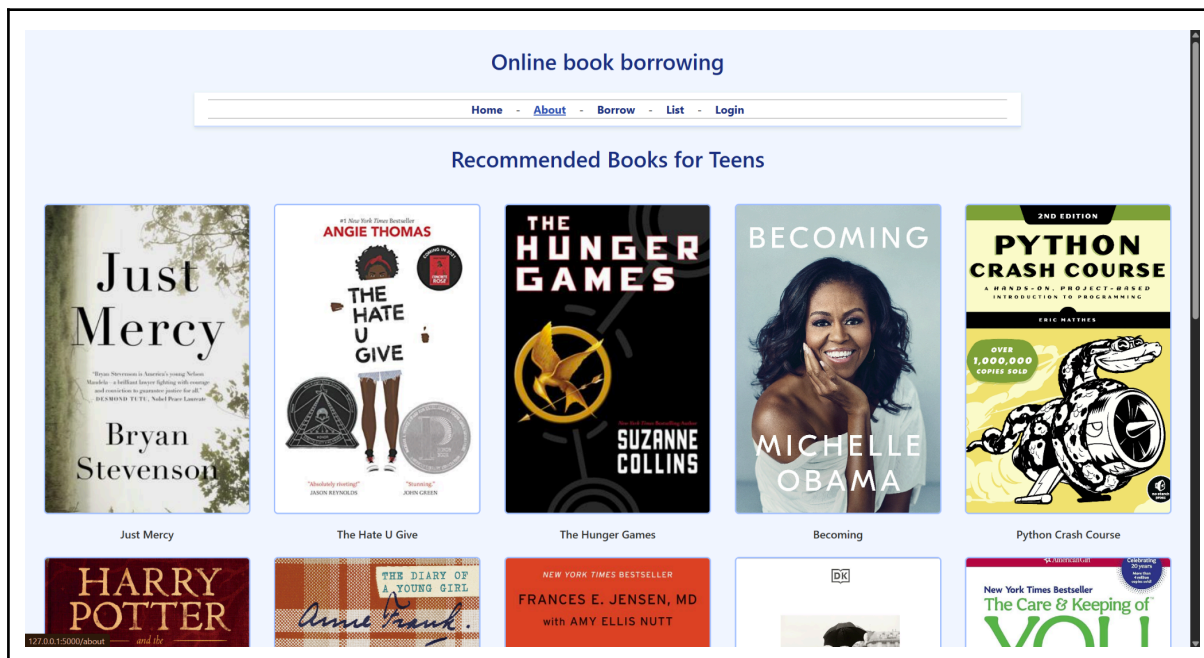
Screenshot 2: Home Page Display

Description: This screenshot shows the Home page of the Online library.



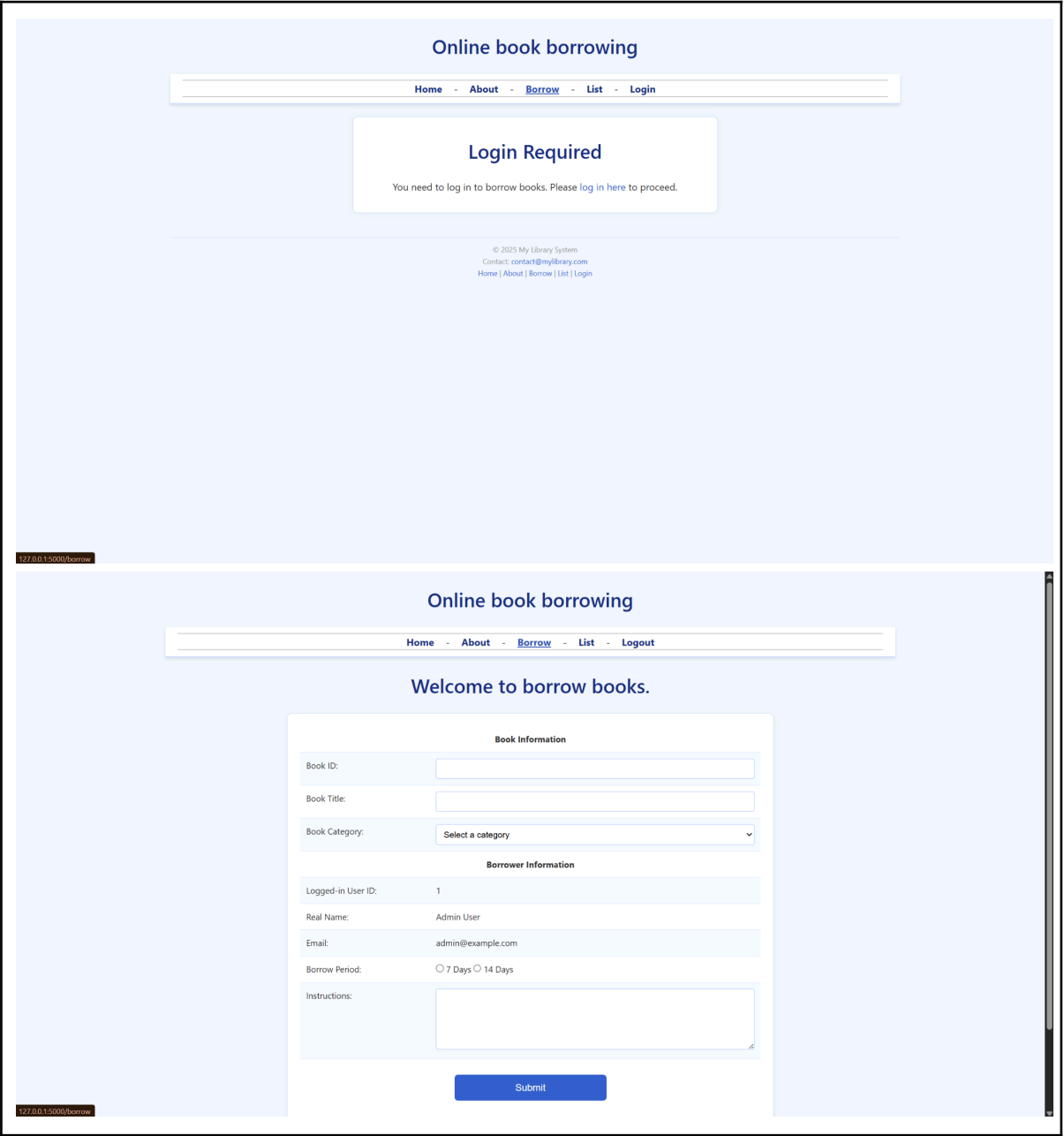
Screenshot 3: About Page Display

Description: This screenshot shows the **About** page of the Online library.



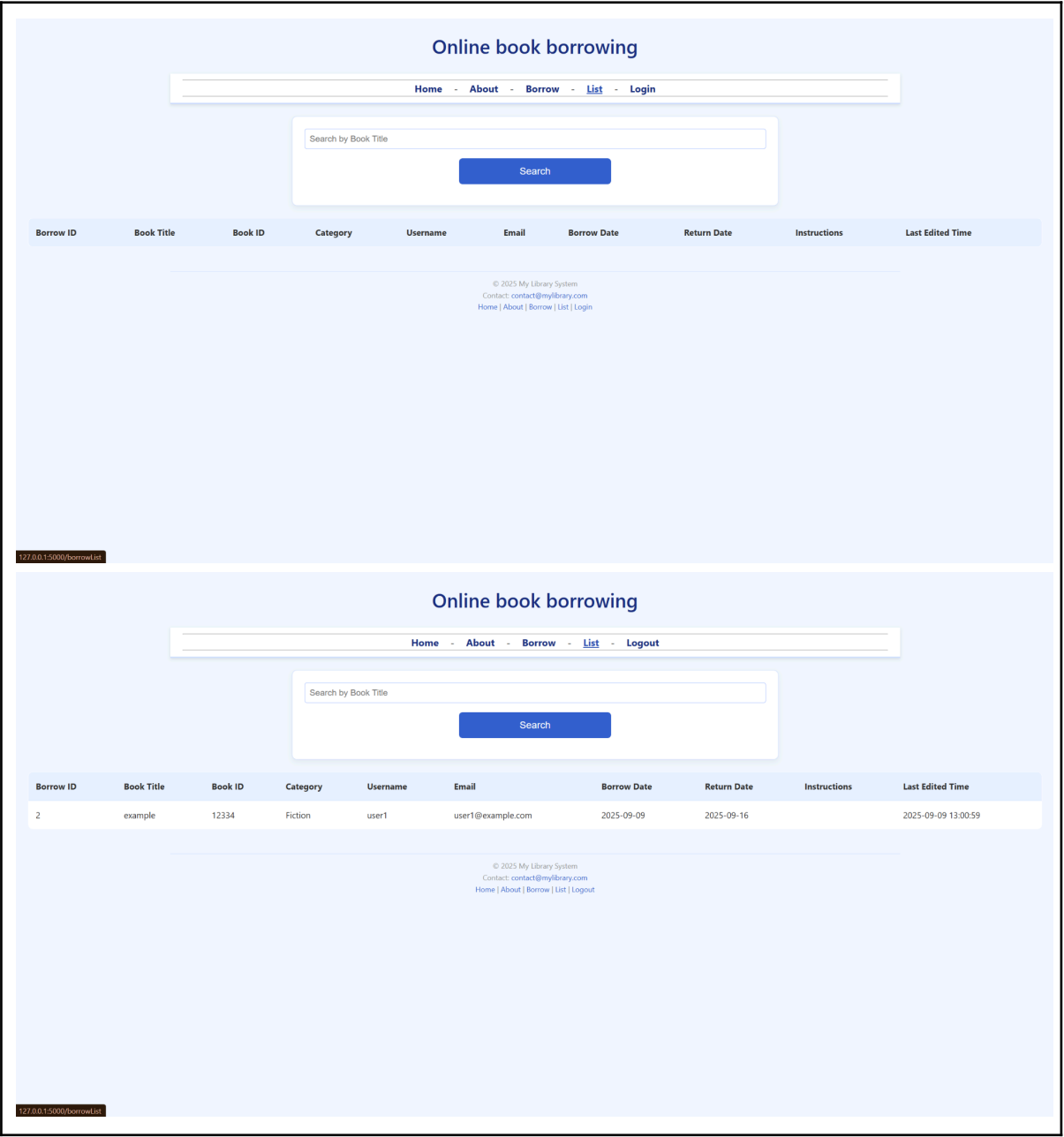
Screenshot 4: Borrow Page Display

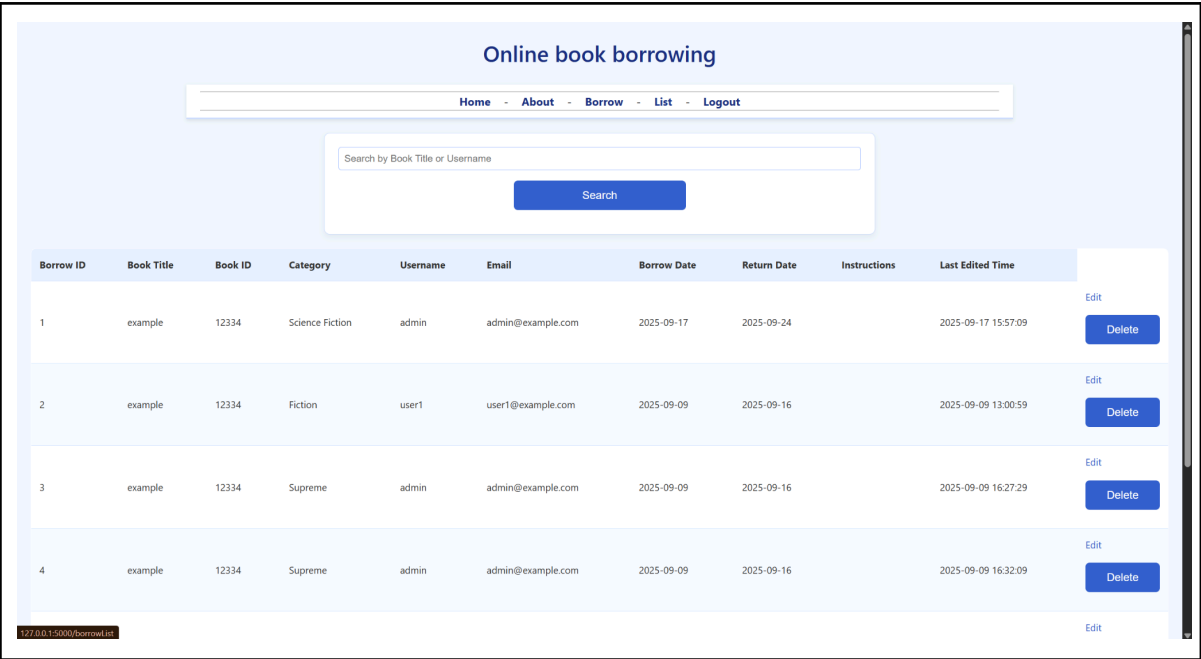
*Description: This screenshot shows the **Borrow** page of the Online library. (There are two types of borrow pages: when not logged in and when logged in)*



Screenshot 5: List Page Display

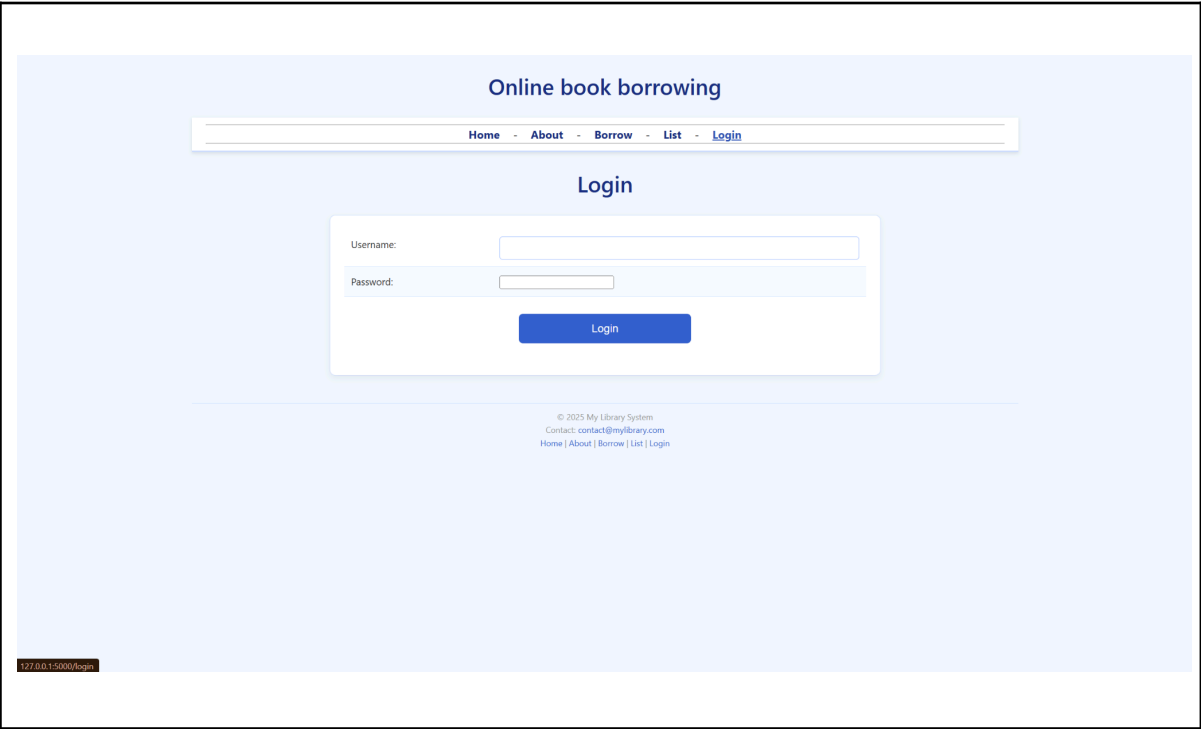
*Description: This screenshot shows the **List** page of the Online library. (There are three types of list pages: when not logged in, when logged in as a user account, and when logged in as an administrator account.)*





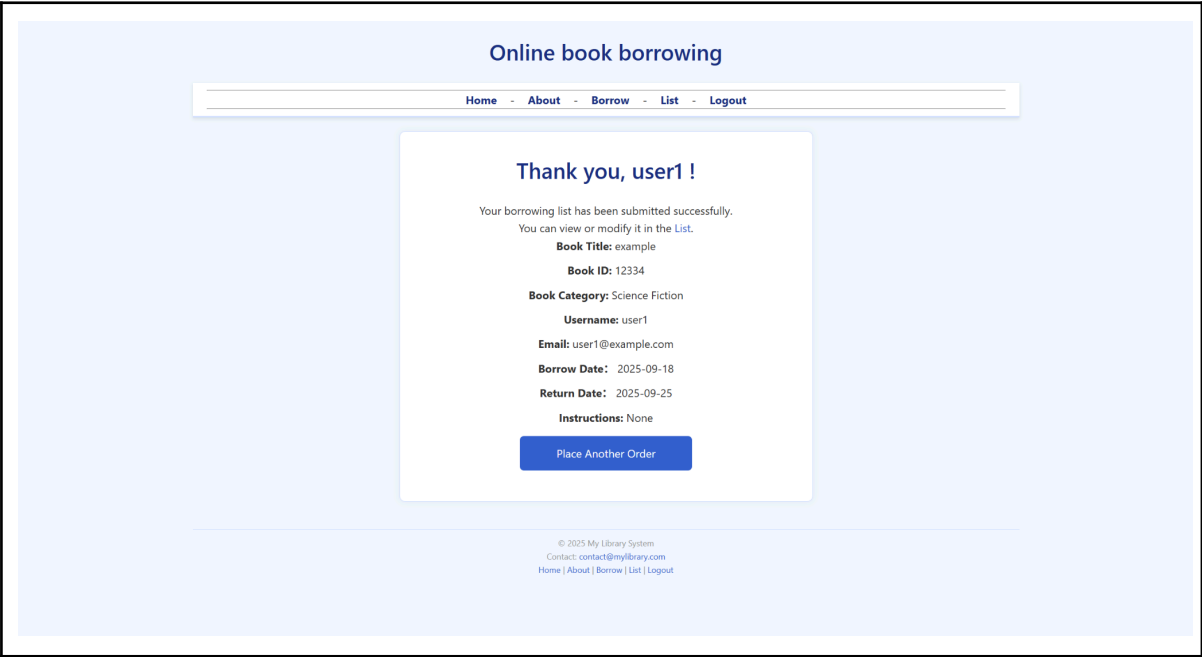
Screenshot 6: Login Page Display

Description: This screenshot shows the Login page of the Online library.



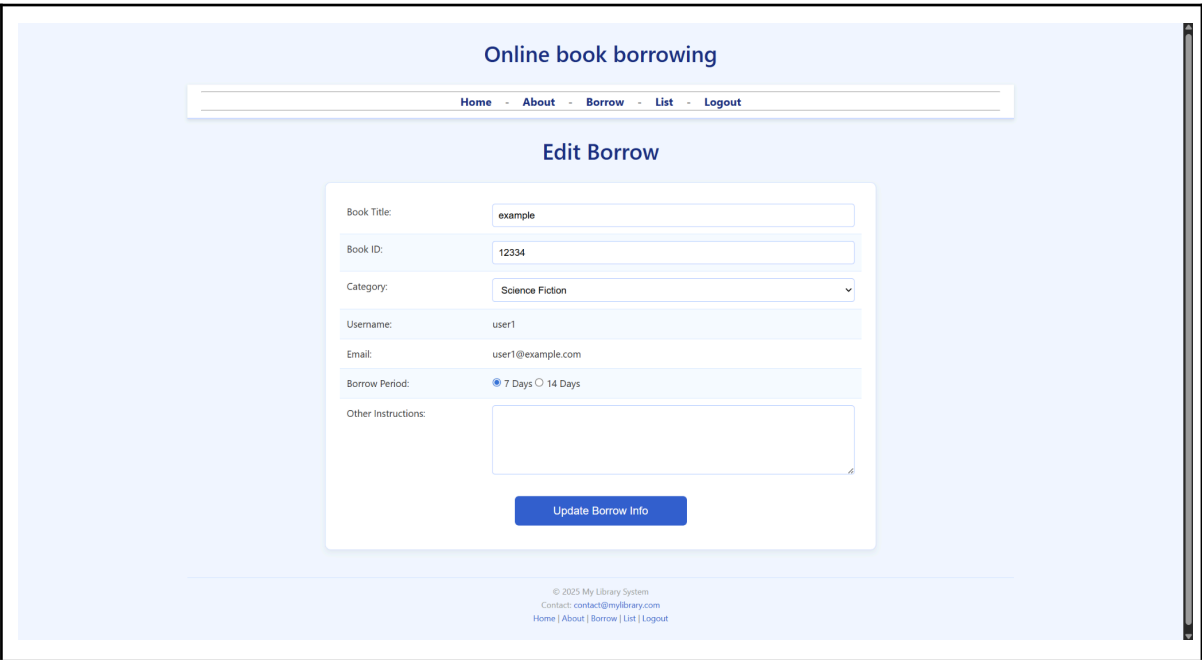
Screenshot 6: Confirmation Page Display

Description: This screenshot shows the **Confirmation** page of the Online library.



Screenshot 7: Edit Page Display

Description: This screenshot shows the **Edit** page of the Online library.



Routes* / function signatures** for each page

```
# Home page route
@app.route('/')
def home():

# About page route
@app.route('/about')
def about():

# Login page route
@app.route('/login', methods=['GET', 'POST'])
def login():

# Logout page route
@app.route('/logout')
def logout():

# Borrow form and submission
@app.route('/borrow', methods=['GET', 'POST'])
def borrow():

# Borrow list with search
@app.route('/borrowList')
def borrowList():

# Edit borrow list
@app.route('/edit_borrow/<int:id>')
def edit_borrow(id):

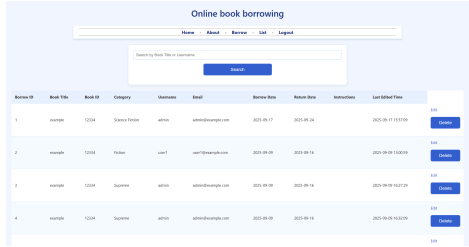
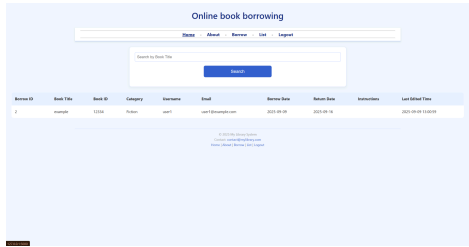
# Update borrow list
@app.route('/update_borrow/<int:id>', methods=['POST'])
def update_borrow(id):

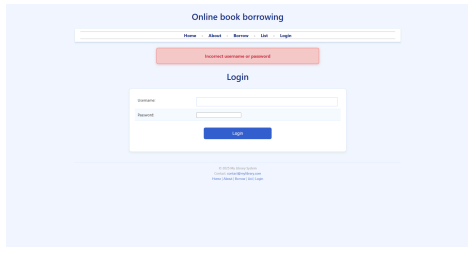
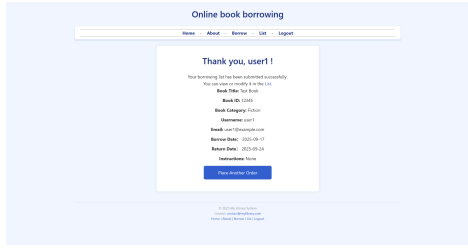
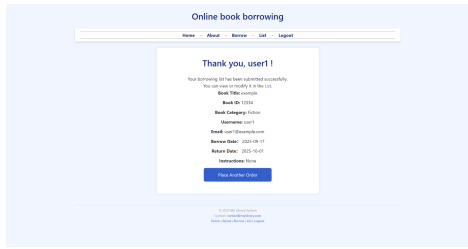
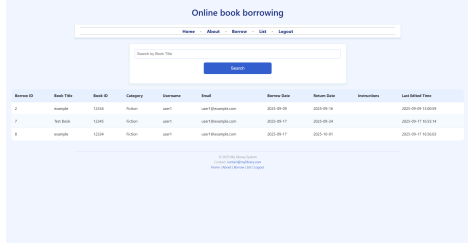
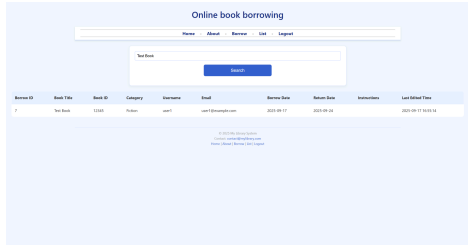
# Delete borrow list
```

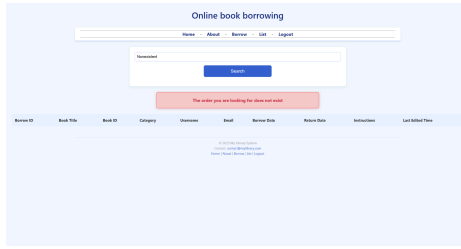
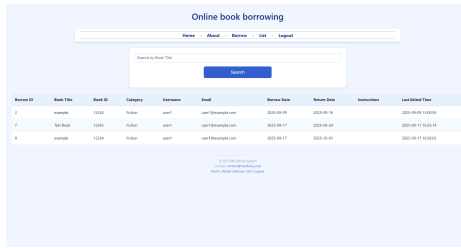
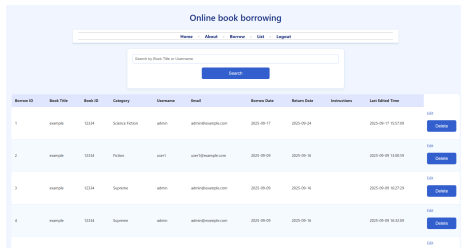
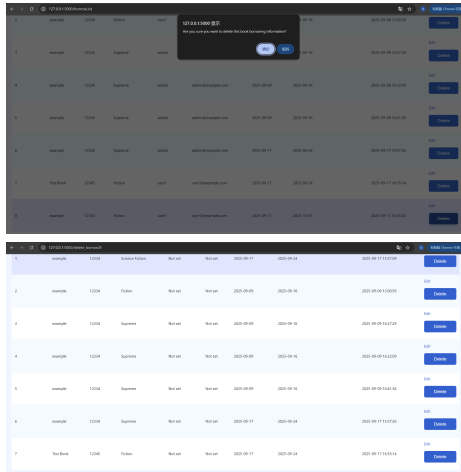
```
@app.route('/delete_borrow/<int:id>', methods=['POST'])
def delete_borrow(id):

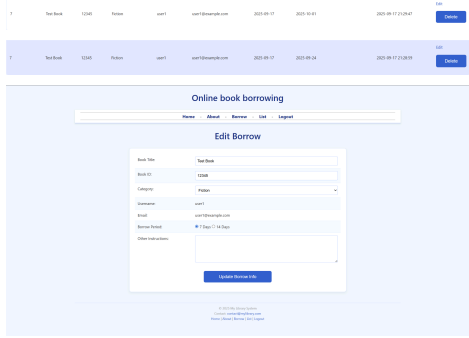
# Custom 404 error handler
@app.errorhandler(404)
def not_found(e):
```

Part 4: Testing

WHAT IS BEING TESTED	HOW IS IT TESTED	WHAT IS EXPECTED	WHAT DID I GET	PASS/ FAIL/NOTES
Log in to the administrator account with correct credentials	Access the /login page, enter "admin" as username and "admin123" as password, submit the form	Successfully receive user information and redirect to the borrow_list_admin.html page	The information was submitted successfully and redirected to the borrow_list_admin.html page (picture1) 	PASS - Login successful
Log in to a normal user account with correct credentials	Access the /login page, enter "user1" as username and "user123" as password, submit the form	Successfully receive user information and redirect to the borrow_list_guest.html page	The information was submitted successfully and redirected to the borrow_list_guest.html page (picture2) 	PASS - Login successful
Login with incorrect credentials	Access the /login page, enter "wrong" as username and "pass" as password, submit the form	Display the error message "Incorrect username or password"	Error message displayed correctly (picture3)	PASS - Error handling works

	form			
Verification of successful book borrowing submission	Log in as "user1", navigate to /borrow, input a valid book_id, "Test Book" as title, "Fiction" as category, select "7days", and submit	Redirect to the confirmation page displaying borrow details	The content is submitted successfully and the corresponding information can be seen on the confirmation page (picture4) 	PASS - Borrowing function work
Accuracy of day calculation in the borrowing system	Log in as "user1", borrow a book with a "14days" period, with borrow date set as 2025-09-17, and submit	Return date calculated as 2025-10-01	The borrowed content was submitted successfully and the return date was 2025-10-01. The date calculation was correct 	PASS - Date calculation is working
Correct display of submitted content on the list page	Log in as "user1", submit a borrow, then access /borrowList	Display the submitted borrow record, including book_title and username	The detailed information of the previous book borrowing appears on the list page 	PASS - List page data display function is working
Functionality of the list form query	Log in as "admin", enter "Test Book" in the search field on /borrowList	Display borrow records containing "Test Book"	Successfully searched for the borrowing list with the title "test book" 	PASS - Query function work

Feedback provision for querying incorrect content	Log in as "admin", search "Nonexistent" on /borrowList	Display the message "The order you are looking for does not exist"	<p>Error message displayed correctly</p> 	PASS - Error handling works
Restriction of user account from editing or deleting list forms	Log in as "user1", visit /borrowList, attempt to edit or delete a record	No edit/delete options available	<p>There are no edit and delete buttons on the list page, so the account does not have these two functions</p> 	PASS - User account permissions are restricted
Authorization of admin account to edit or delete list forms	Log in as "admin", visit /borrowList, attempt to edit or delete a record	Display edit and delete links, with updates reflected after operation	<p>The edit and delete buttons are correctly displayed on the list page, so the account has these two functions</p> 	PASS - Unlimited administrator account permissions
Feasibility of the delete data function	Log in as "admin", select a record, submit delete request	Record removed from /borrowList	<p>Click the Delete button and a deletion warning prompt will appear: "Are you sure you want to delete this book borrowing information?" Click OK and the data will be successfully deleted from the list page</p> 	PASS - Delete function is working

Feasibility of the edit data function	Log in as "admin", select a record, change return_date to "14days", submit	return_date updated to 14 days later, list reflects change	<p>Click the Edit button to jump to the Edit page, change the borrowing date to 14 days, click Confirm, and the final return date is 2025.10.01. The original return date is 2025.09.24. The date change is correct.</p> 	PASS - Editing functionality available
---------------------------------------	--	--	---	--

Part 5: Explanation and Solution

Part 1: Explaining Relevant Implications (Database)

The development and management of a database, such as the one used in my Flask-based borrowing system (library.db), have significant implications related to privacy and legal standards. Two key aspects from these standards—Privacy Requirements and Legal Compliance—are particularly relevant to a database.

- Privacy:** This aspect is relevant because it ensures that data is kept private, personal information is safeguarded, and sharing is minimized. The reason is that the database stores sensitive user data, such as usernames, emails, and borrowing records in the Users and Borrows tables. Failure to protect this data could lead to privacy breaches, causing harm to users and potential legal issues.
- Legal:** This aspect is relevant because the database must obey all relevant laws, including copyright and intellectual property rights, the Harmful Digital Communications Act, and data protection/privacy/libel regulations. The reason is that the system handles user-generated content (e.g., borrowing records) and personal data (e.g., emails), and non-compliance could result in legal penalties, fines, or reputational damage due to violations such as unauthorized data use or libelous content.

Part 2: Addressing relevant implications (Database)

- Privacy: To address the privacy requirements, I took measures to safeguard personal data within the database. I implemented password hashing using `generate_password_hash` from the `werkzeug.security` module to encrypt passwords in the Users table, ensuring they are kept safe from unauthorized access. The `borrowList` route was designed to display only the logged-in user's own records (using `WHERE b.borrower_id = ?`), preventing the sharing of personal information with others. Since the system does not use images, no permissions are required, further reducing privacy risks. However, emails are visible to users and administrators for contact purposes, and while this is functional, I could enhance privacy by masking them (e.g., `u***@example.com`) in future iterations.
- Legal: For legal compliance, I ensured the system adheres to relevant laws. The borrowing system focuses on educational content, avoiding copyright or intellectual property violations by not storing or distributing protected materials. I also restricted data modification to administrators (`role=1`) while limiting regular users (`role=0`) to viewing their own records, aligning with data protection laws to prevent libel or harmful communications. Although I did not implement a formal privacy policy, the design reflects a commitment to legal standards by minimizing unauthorized data exposure.

Part 3: Explaining Relevant Implications (Website)

The design and functionality of my Flask-based borrowing system website carry significant implications related to end user needs and aesthetics. Two key aspects—End User Requirements and Aesthetic Appeal—are particularly relevant to ensuring the website's effectiveness and user satisfaction.

- End User: This aspect is relevant because it addresses the specific needs of the end users, including ordinary users and administrators, in a publicly accessible book borrowing platform. The reason is that the website must perform essential tasks (e.g., borrowing books, viewing records) and be accessible on suitable devices, using appropriate language and appealing to the right demographic. Failure to meet these needs could result in poor user engagement or accessibility issues.
- Aesthetic: This aspect is relevant because the website's visual design influences user experience and retention. The reason is that a well-designed interface, following design conventions and conveying the right image (e.g., an educational theme), ensures it appeals to the

target audience, enhancing usability and credibility.

Part 2: Addressing relevant implications (Website)

- End User: To address end user requirements, I designed the website to meet the specific needs of a broad audience, including ordinary users and administrators, as it is a publicly accessible book borrowing platform with no major restrictions. The website fulfills core functions such as borrowing books and viewing personal records via the borrowList route (using WHERE b.borrower_id = ?), ensuring it does what users need. The language is kept basic and clear, suitable for non-native English speakers like myself, avoiding complex terminology to maintain an accessible reading level. While the website has been tested only on laptops and desktops, its responsive design (e.g., centered layouts with max-width: 1200px) aims to support wider device compatibility, though mobile testing is limited. The open access nature appeals to a diverse demographic, encouraging anyone interested to use it.
 - Aesthetic: For aesthetic appeal, I crafted a visually pleasing design using a minimalist palette of light blue (#f0f8ff), light gray, and white, with red as a contrast color for warnings (e.g., .alert with #ffc000). This follows design conventions such as centered navigation (nav with text-align: center) and consistent button styles (input[type="submit"] with border-radius: 6px). The design conveys an educational image suitable for a library context, enhanced by features like mouse hover feedback (transform: scale(1.05) on nav a:hover) and page transition animations (@keyframes fadeIn). Small functionalities, such as button scaling on click (transform: scale(0.98) on input[type="submit"]:active), provide user feedback, making the interface intuitive and appealing to the target audience.
-

Part 6: Feedback

Part 1: Formal Feedback from Teacher Discussion (All completed)

Enhancement of Search Functionality

- Feedback: It is recommended to implement a search feature allowing users to search for books by title or borrowers by name. This will improve the usability and efficiency of the website, enabling quick access to relevant information.

Incorporation of Visual Elements

- Feedback: Consider adding pictures to the website and creating a dedicated section for images within the borrowers table. This enhancement will enrich the user experience by providing visual context to the borrowing data.

Database Connectivity and Relationship Integrity

- Feedback: Ensure the database is properly connected, with the borrow_id in the borrowers table accurately linked to the id in the users table. This relational integrity is essential for maintaining data consistency and functionality across the system.

Testing and Documentation of Visual Elements

- Feedback: Conduct thorough testing of the picture integration and provide a detailed explanation of at least 10 test cases. This will validate the functionality and accessibility of the new visual features, ensuring a robust implementation.

Part 2:Feedback from Non-Programmer People

Person 1 - The website has a clean look, with a pleasant light blue background, but the colors are a bit dull, and the red warning text is a bit glaring. I'd like more interesting animations when transitioning between pages.

Note - I might adopt it, but it's not very practical

Person 2 - The design is minimalist, reminiscent of a library, and the buttons are interestingly responsive, but the layout is a bit crowded. The font could be larger, and the animations could be more vivid.

Note - Font size suggestions can be adopted

Part 2:Feedback from Software Engineer People

Person 1 - feel the email display looks unsafe and suggest adding stars in the borrowList page, like u***@example.com, to make users feel more secure.

Note - Effective suggestions will be adopted

Person 2 - recommended to add a new database to manage book content, and connect it to the borrowing database to facilitate book management.

Note - Very good suggestion, will adopt it

Part 7: Changelog

[1.0] - 2025-XX-XX

Detail:

This version of the online library website is my first time programming, so the content is relatively simple. It only has one function to query the book content in the database. This version adds some CSS for minor decoration, and there are some book contents that can be queried in the database. Since this is just an attempt and the content is too little, I may re-make a new website with other templates in the future.

[2.0] - 2025-XX-XX**Detail:**

This version of the library website was modified by me based on my teacher's pizza subscription website. Most of the content is similar to the pizza subscription website. This version of the website does not use a lot of CSS for decoration. There is no content in the about interface. There are also some problems with the code of the borrow page. Among them, the selection of borrowing time cannot be confirmed, which may need to be improved in the future.

[3.1] - 2025-XX-XX**Detail:**

This version of the online library is an update to the previous one. It uses some simple AI-generated CSS for simple decoration, with a predominantly blue colour scheme. I've also added a brief introduction to the "about" page. Available features include:

1. Book borrowing form;
2. Data upload;
3. Querying, modifying, and deleting borrowing records.

Since modifying and deleting borrowing records shouldn't be accessible to ordinary users, I may implement some restrictions in the future, such as adding administrators.

[3.2] - 2025-XX-XX

Detail:

This version of the online library is a branch of the third edition. Most of the content is almost the same as the third edition, but the two database tables used in the code are for more convenient data management. However, after modification, it was found that the cost-effectiveness for this small website is not as high as the single website management of the third edition, so this solution was abandoned.

[4.0] - 2025-08-xx**Detail:**

This version optimizes the borrower's borrowing check-in process, specifies the borrowing time as the current date, and the return time as the borrowing date plus a specific time (such as 7 days or 14 days), and adds a dedicated footer. The next plan is to add a user authentication system to the website, such as administrators and borrowers. Administrators can use all website functions, while borrowers can only borrow and search data and cannot modify the data. In terms of web page aesthetics, In the future we may make certain modifications to the layout of the edit and delete buttons on the list page to improve the overall aesthetics.

[4.0] - 2025-08-xx**Detail:**

This version adds a significant amount of new functionality to the online library website. First, I added a new login system. Since regular visitors shouldn't have the ability to modify or delete borrowed items, I restructured the original separate borrow list page into a guest and administrator version. The guest version retains only basic query functionality, while the administrator version grants access to modify, edit, and delete borrowed items.

Login is handled by newly added login and logout screens.

The basic login username is "admin" and the password is "admin123."

To implement this login functionality, I used flask login, which led to significant changes to the underlying backend code.

I moved database generation to config.py, added login code to auth.py, and added login and logout code to main.py.

Future plans include adding library book data, images, and descriptions to the about.html page to facilitate borrowing by book ID.

Since the previous CSS is insufficient to support the new features, I may also update and update the overall CSS.

Part 8: Conclusion

git hub code clone link:

https://github.com/Sky-23758/12DTP-project_library.git