Project Report: Online Library

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	- Flask application setup - Database connection and API development - Routing and business logic	Completed
Part 3	Frontend Code Production	HTML	- Web page templates (Home, Search, Details pages) - Using Jinja2 to display data from the backend	Completed
Part 4	Web Page Styling	CSS	- Overall website styling - Layout, fonts, colors, and responsive design	Completed
Part 5	Testing	Manual Testing	- Overall system functionality testing - Bug fixing and performance optimization	In Progress
Part 6	Update new content	All of the above tools	- Add new content and update - Final summary	In Progress

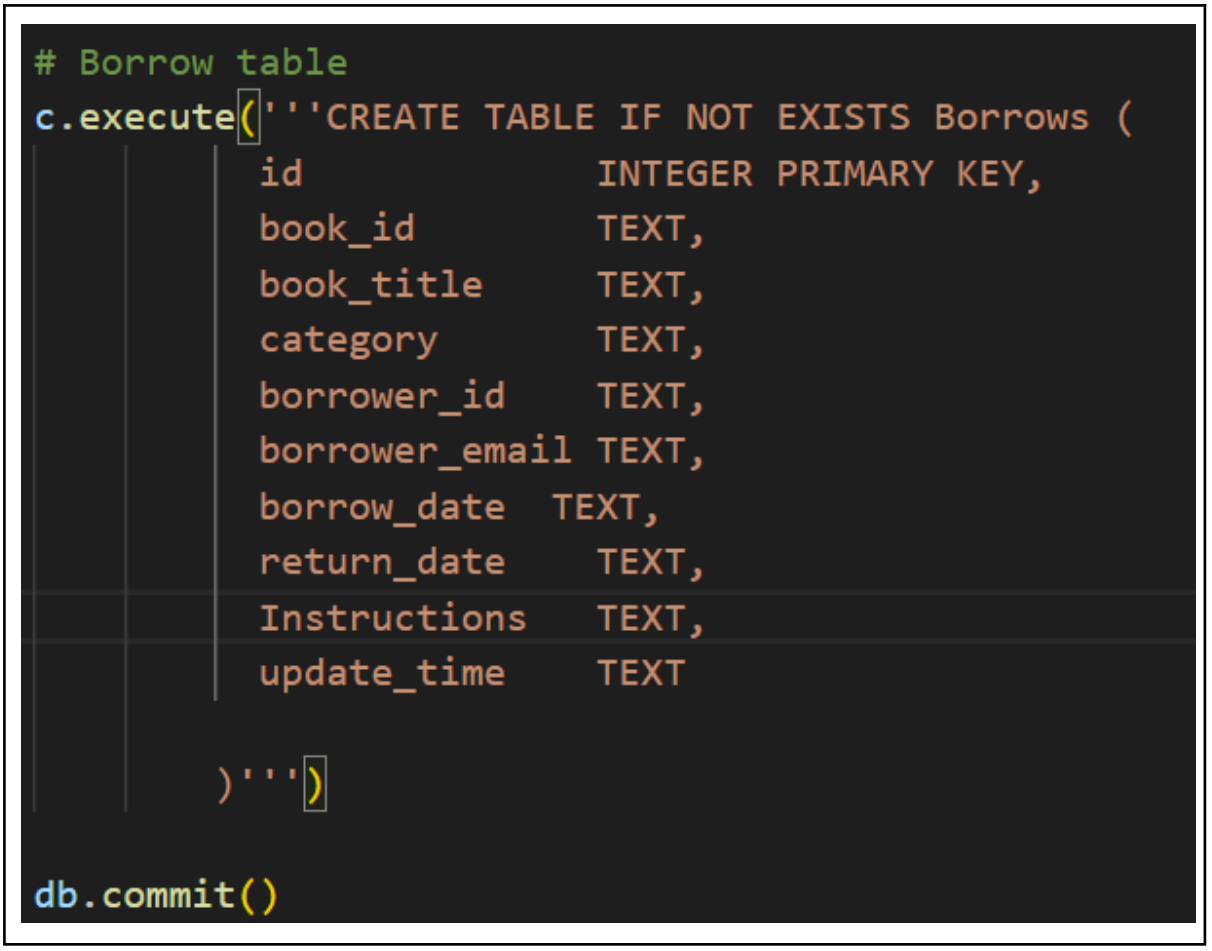
---

Part 3: Feedback & Visual Evidence

Feedback Summary:

1. New database information in the pictures. (from classmate)
2. The webpage is too monotonous（from myself)
3. Missing footer（from myself)
4. Insufficient search results（from myself)

Screenshot 1: Database design diagram



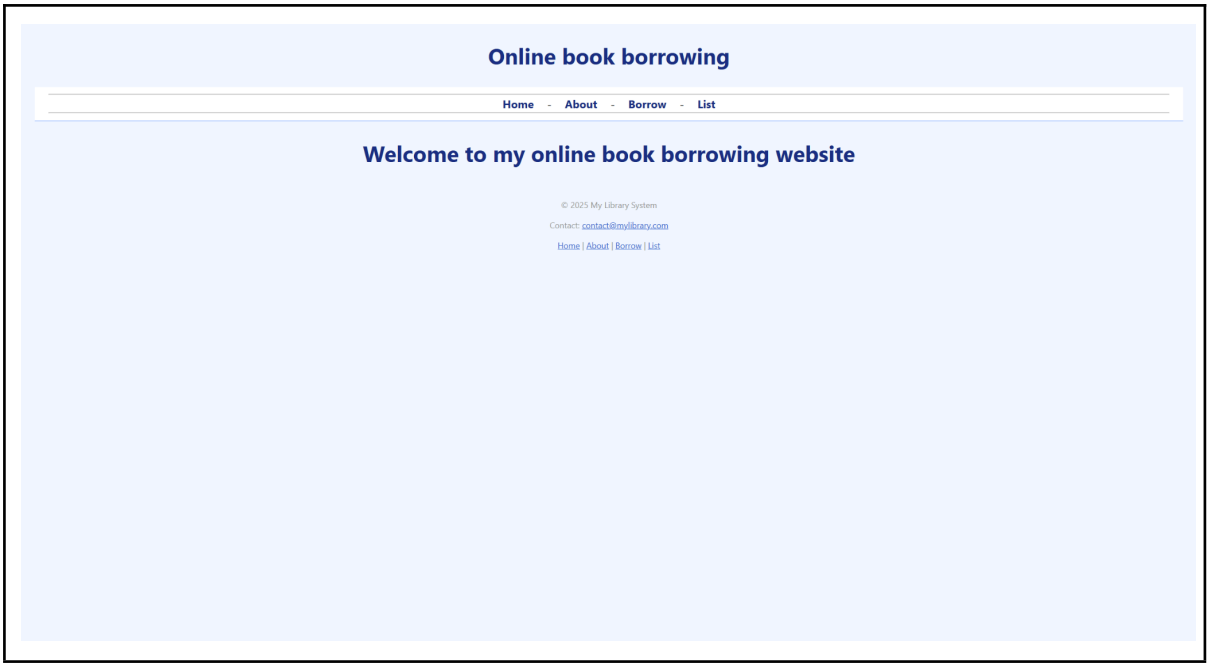
```
# Borrow table
c.execute('''CREATE TABLE IF NOT EXISTS Borrows (
            id              INTEGER PRIMARY KEY,
            book_id         TEXT,
            book_title      TEXT,
            category        TEXT,
            borrower_id     TEXT,
            borrower_email TEXT,
            borrow_date   TEXT,
            return_date     TEXT,
            Instructions    TEXT,
            update_time     TEXT

        )''')

db.commit()
```

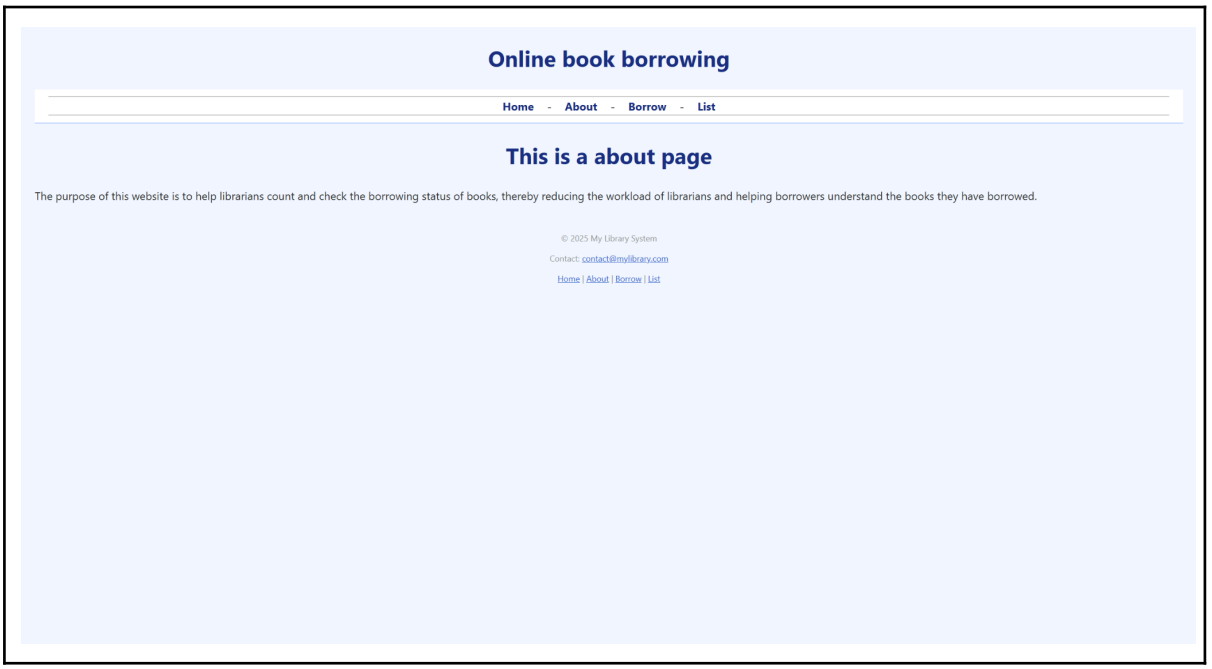
Screenshot 2: Home Page Display

Description: This screenshot shows the **Home** *page of the Online library.*（*Not the final result*）



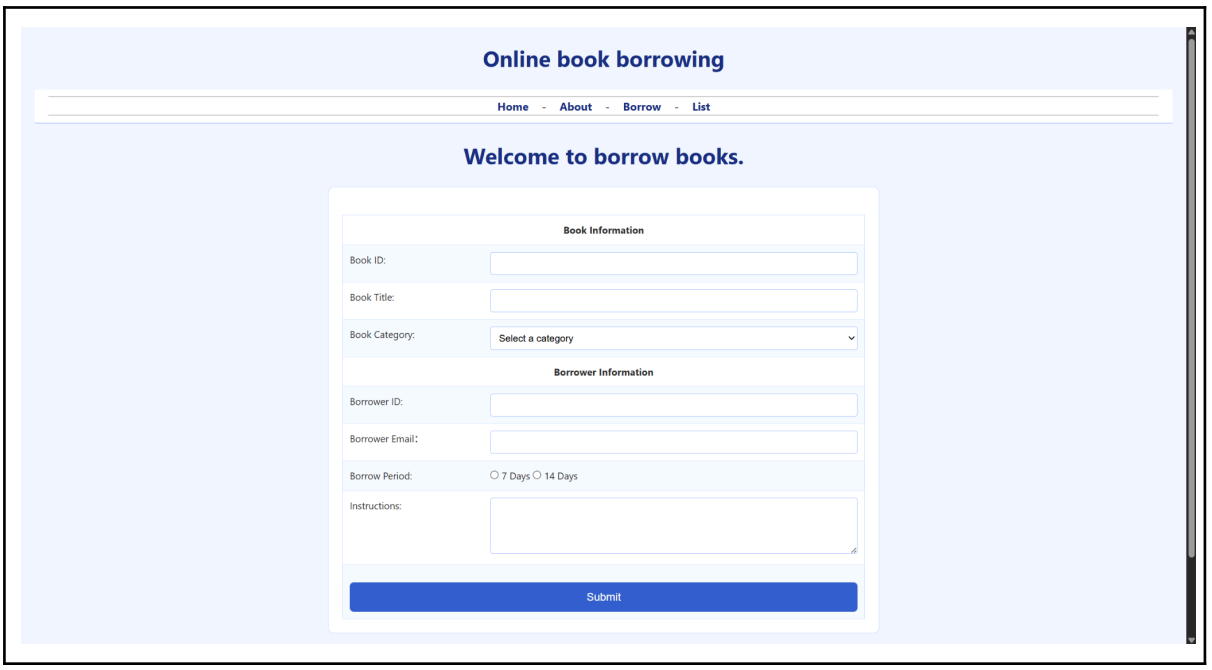
Screenshot 3: About Page Display

Description: This screenshot shows the **About** *page of the Online library.*（*Not the final result*）



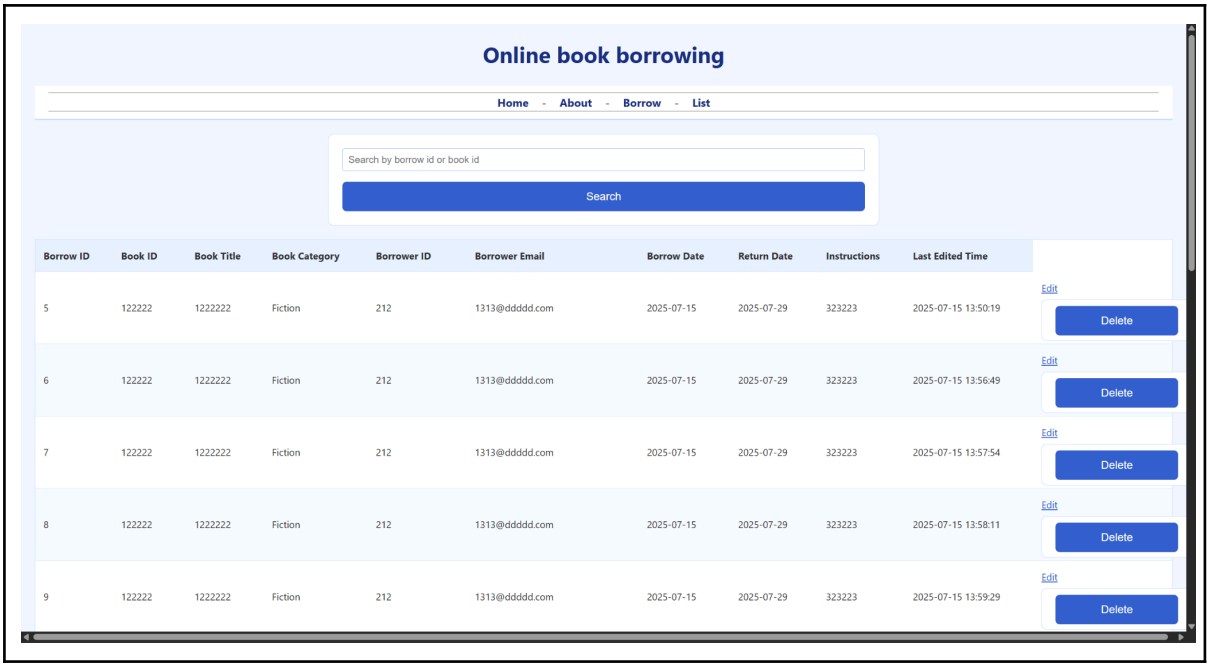
Screenshot 4: Borrow Page Display

Description: This screenshot shows the **Borrow** *page of the Online library.（Not the final result）*



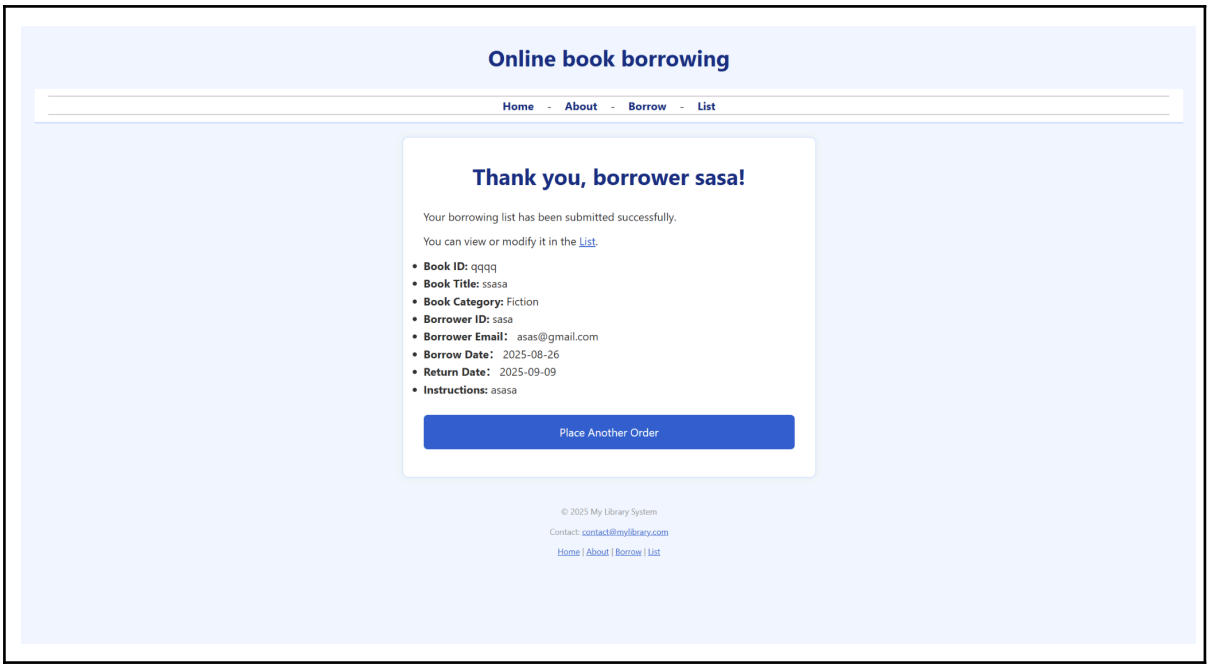
Screenshot 5: List Page Display

Description: This screenshot shows the **List** *page of the Online library.（Not the final result）*



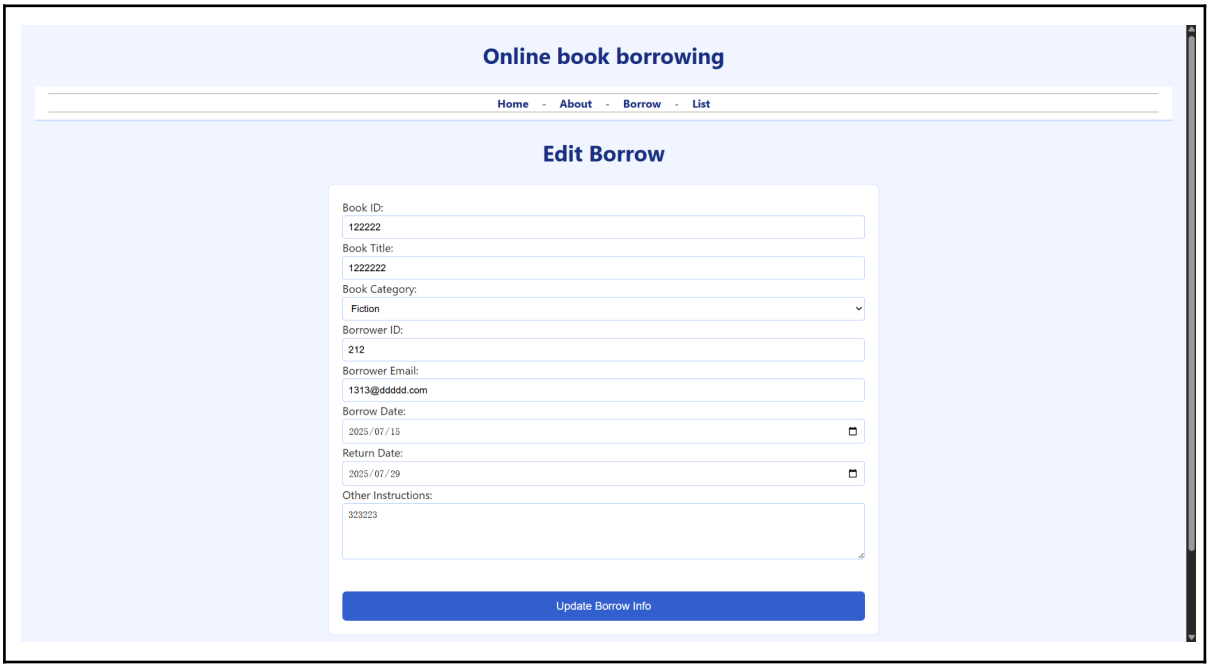
Screenshot 6: Confirmation Page Display

Description: This screenshot shows the **Confirmation** *page of the Online library.*（*Not the final result*）



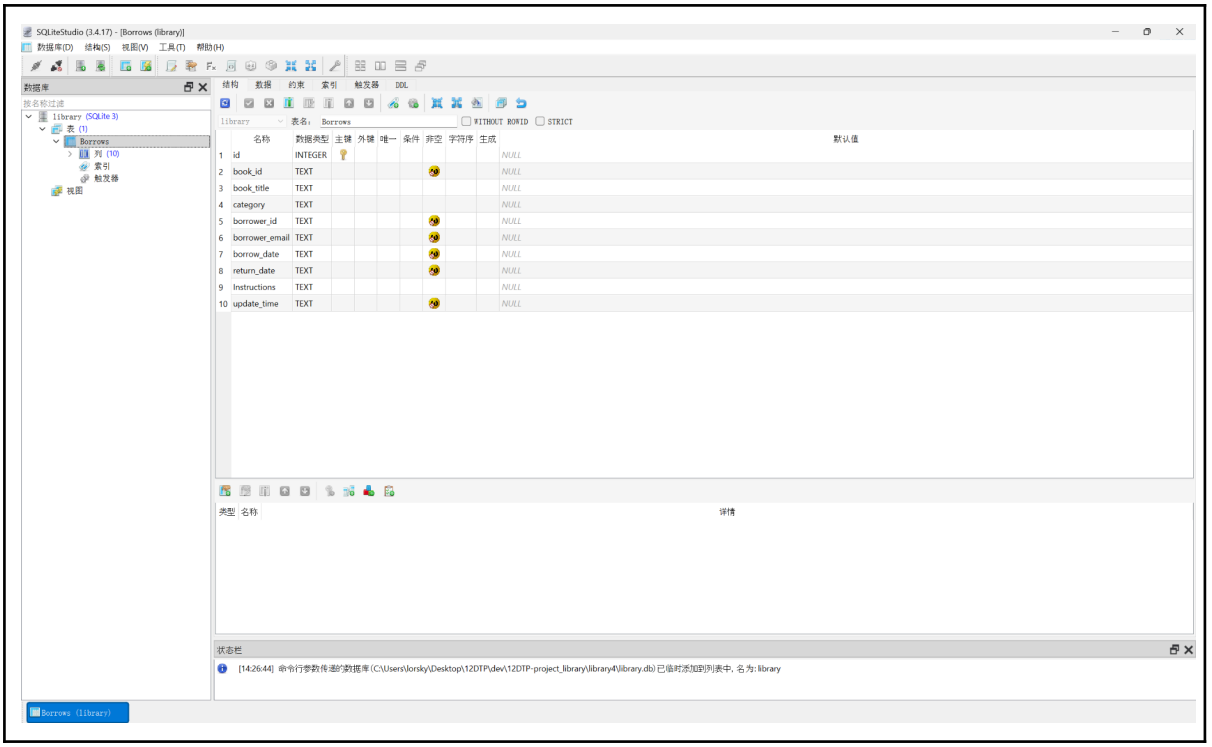
Screenshot 7: Edit Page Display

Description: This screenshot shows the **Edit** *page of the Online library.（Not the final result）*



Screenshot 4: Database queries for each route (SQL query/queries)

Description: This screenshot shows database queries for each route (SQL query/queries)（Not the final result）



Routes* / function signatures for each page

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

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

# 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: 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.