CS 421/CS 621 Adv Web Application Dev

Final Project Report

Yixuan Liu, Frank Wu and Yixin Zhang

Aug 5, 2022

**Introduction**

In this project, we design a web-based application for university students to purchase books at an online bookstore. All students can create an account with their email and name. Also, the bookstore manager has an admin account which can check orders and account. Only admin account can add and modify book (books information, books stock and price…). Also, the students can easily find the book they want.

**Technologies**

In this project, we mainly use HTML, Python and CSS. The HTML is used for all website structure. All styling is using CSS. For the back end we mainly using Python. In the front end, we also using bootstrap to build the sign-in page and navigation bar (top navbar and side navbar). The advantage to using that is it can build a UI fast and looks good (some detail still needs to improve). Awesome5 is also used in this project. The icon in this project are mainly from Awesome5. For the back end, we mainly using flask, SQLite and SQLAlchemy.

Flask framework is a web server framework and used in backend. Flask supports the dynamic web pages and provides most of the functions in the project. In this project we are using flask for most of action in the web app. For example, we use render\_template and redirect to connect each page and it used to get users input information and store to the database.

For the database, we use SQLite3. In this scenario, the database should deal with a large data with thousands of books, students’ information, and orders. And sometime the database could be used to deal with complex queries, so a stable DBMS which can solve complex relational query is very important. SQLite3 is a solution and works well with flask.

Diagram

Description automatically generated Diagram

Description automatically generated Diagram

Description automatically generated

Ajax is a front-end technology. By using Ajax, the web pages can send data to web server. In this project, we use Ajax to send notification to manger.

**Results**

Sign in & Sign up

When user (new user)/ admin use this app, it will state at route ‘/’, which is the main login page. For the user who already have account can enter user’s name and password here. With incorrect username or password, message will be display after click login button. The “Sign Up” button is used for new user to create an account and the admin button is used for administrator or manage to login. For the sign up page, the user’s email is used as ID in DBSM, which means one email could one register one bookstore account. After create an account, the user can easily redirect to the login page.

**Graphical user interface, application

Description automatically generated**

The homepage

After user log in, user will redirect to the homepage. At the top navigation bar, there are “Book Store”, “Home”, “Cart”, “My Account”, and “Log Out”. In this stage, the “Book Store” and “Home” button will redirect to the homepage (same function). The list at the homepage shows all book to the user. The user can simply use the checkbox in front of the list to add the book to the cart.

The side navigation bar is used to choose the book category. By click the list, the user could find the book which is user’s want.

**Table

Description automatically generated**

Cart page

By click the “Cart” button, the user will redirect to cart page. In this page, all item in the cart will be listed and the total price will be shown at the end of the list. The user can “Place Order” button to confirm this order.

**Graphical user interface, application, table, Excel

Description automatically generated**

Admin

By log in to admin account, the manager can see all the orders and edit and add books. To add a book, the manager should type all the book’s information and submit it, the information will send to the database directly. The manager can also use ISBN to search the book in the database to update the information.

**Graphical user interface, application

Description automatically generatedTable

Description automatically generated**

**Graphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

Future Work

1. Separate the admin registration, admin login and admin database to improve the security.
2. Set only one admin account when release the application. And only the first admin can add new admins to the system.
3. The user’s database should be encrypted. The real user’s information will never show in plaintext. Using private key at client side to encrypt the information.
4. The user’s email could be used to authenticate user’s log in. Also, the system can send a email to user to confirm the email address. With this function, the user can also use email to reset the password, if they forget the password.
5. The database schema could be improved by add a new table “order\_item” (order\_item (item\_id, order\_id)). By doing this, we can use SQL to show all items under the specific order.
6. The web crawler could also be used to find book information when manager add a new book. When manager type the ISBN, the application can return a result to the manager to confirm. If everything of the crawler result is correct, the manager can just type the quantity of the book and upload to the database. If there are something wrong, the app can fill all information to the input box, and let manage to make some change.

**References**

CSS navigation bar. (n.d.). Retrieved August 4, 2022, from https://www.w3schools.com/css/css\_navbar.asp

*How to - fixed sidebar*. How To Create a Fixed Sidebar. (n.d.). Retrieved August 4, 2022, from https://www.w3schools.com/howto/howto\_css\_fixed\_sidebar.asp

*How to - login form*. How To Create a Login Form. (n.d.). Retrieved August 4, 2022, from https://www.w3schools.com/howto/howto\_css\_login\_form.asp