Design Report

1. A statement of the purpose of your website and a list of features implemented.

In this coursework, I designed a book management system containing a super admin, several admins, and normal users. In this system, both the user and admin can register and login for the system. Users, Admins and the super admin enjoy different privilege and can do different things in this system.

- **User**

A normal user can:

1) View all book in the library. Books in the library are also divided into several categories which is more convenient for users to find types that they want.
2) Borrow and return books. If users show their interest in a book, they can borrow it. However, there is a limitation of borrowing books: a user can only borrow 3 books at the same time.
3) Find books by book name or author name. users can search books on the home page of the library. As a user, if you enter ‘a’ into the search form, name of books and authors that contains ‘a’ will be all displayed on the result page.
4) Also, there is another limitation for users that a book can only be borrowed for at most a month. If a user borrows a book for more than one month, the criminal number of the user will add one. If the criminal number increases to five, the user will never accessible to borrow books.

- **Admin**

An admin can:

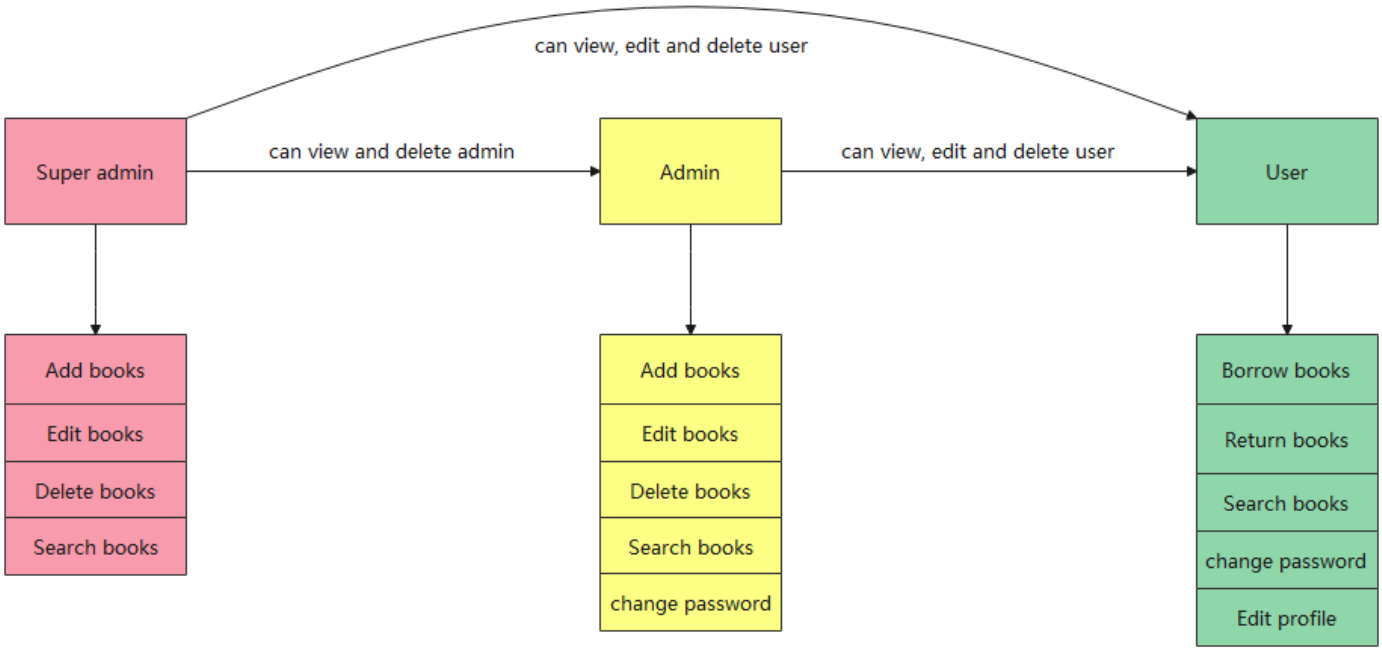
1) Add, edit and delete books. An admin can add the picture of a book as well as its name, author and so on. All the information of books will be stored in the database.
2) Delete and edit information of users. As mentioned above, there are limitations of number of borrow books as well as time of borrowing books. Thus, if there exist some error in the system or other personal condition, the admin can manually adjust those two values.

- **Super admin**

As for the super admin:

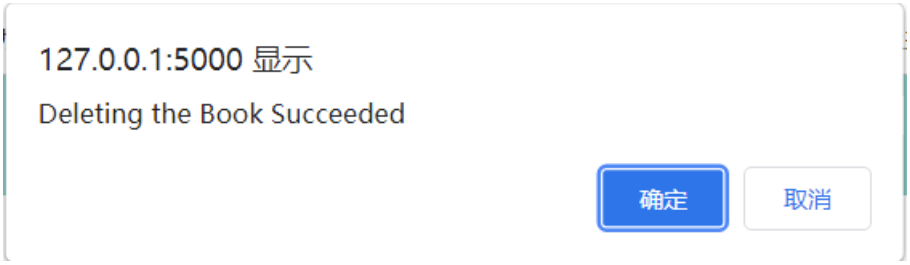
There is just single one super admin who enjoys the highest privilege in the system. The super admin can do everything that admins can do and except for this, he/she can see all admins and has the right to delete some of admins.

Hope that the following image could illustrate the above words clearer.

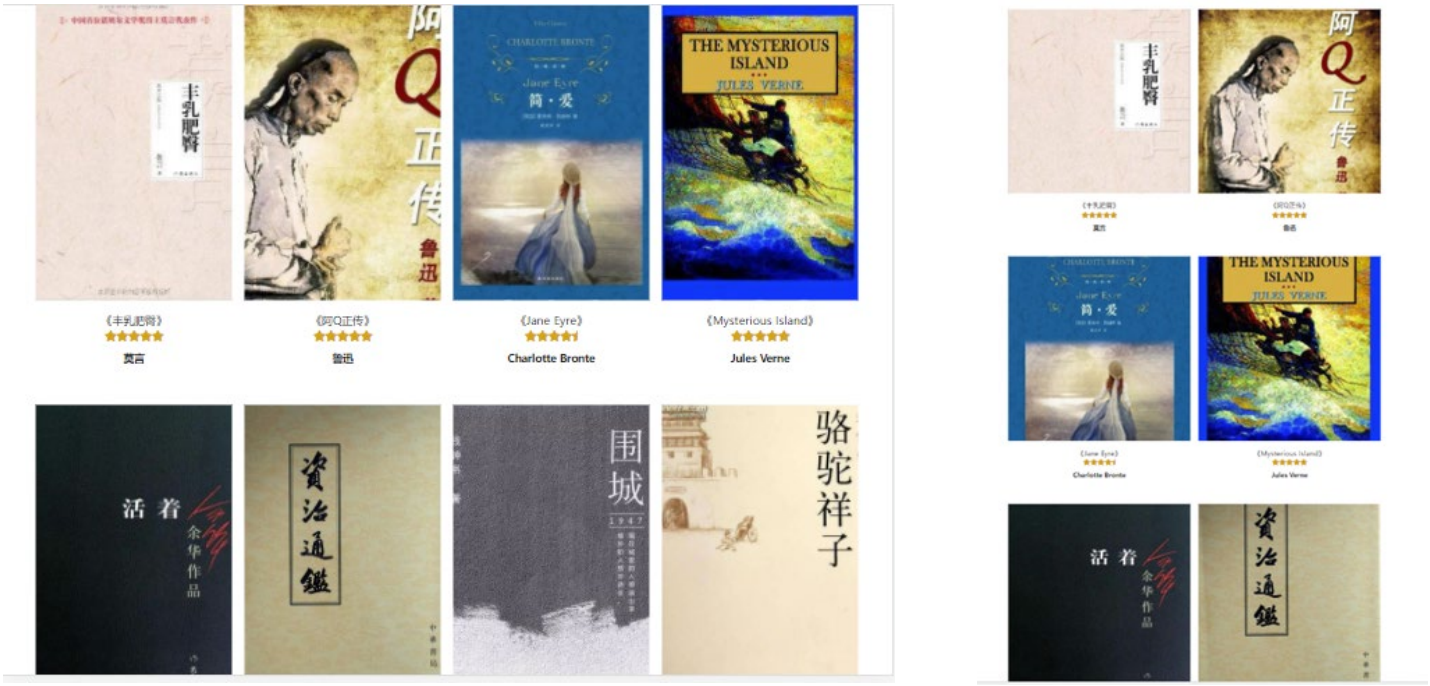


Feature implemented:

1) Registration, login, and authentication of users and administrators. If the entered password is incorrect or the user does not exist, a message is displayed.
2) Add, delete, change and search functions for users, administrators and books. Additionally, the system allows users and admins to change their password.
3) Some prompt information after operation implemented by JavaScript. For instance, after deleting a book, the admin or super admin will receive a confirm message like this:



4) Responsive layout. In this coursework, a responsive layout is implemented so that no typography errors will occur on any device. It is worth mentioning that the site has a very successful way of responding:



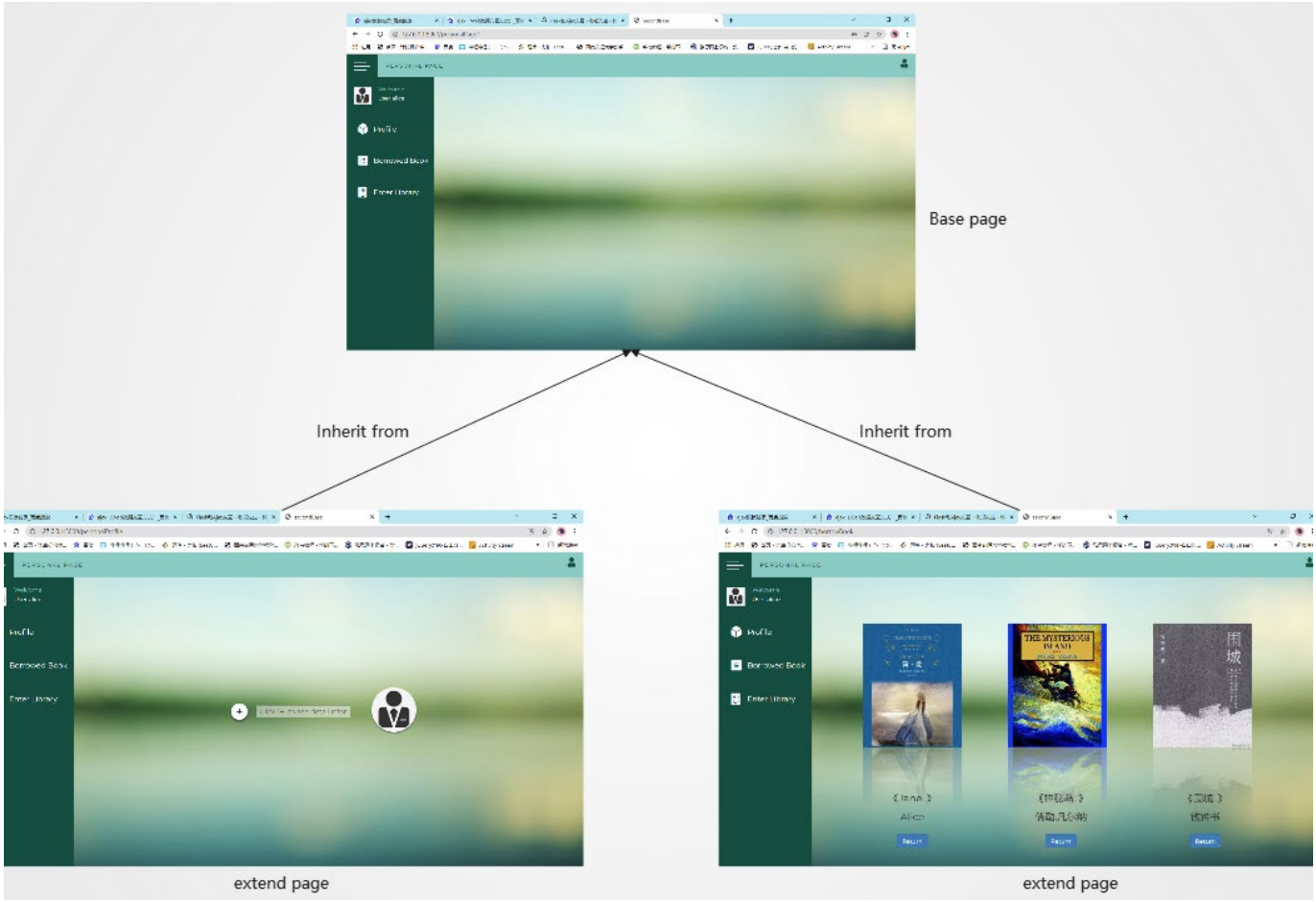
On the left, I view the page in landscape, and on the right, in portrait. As you can see, the layout is four images in a row in landscape and two in portrait.

5) Automatic rotation of pictures and some advanced features of HTML like svg as well as some other tools like jQuery, bootstrap and so on. In the library home page, svg is used for the rotation diagram. At the same time, my login and registration interface and navigation bar interface have adopted some templates of jQuery, which not only greatly improves my programming efficiency, but also makes the page more beautiful.

6) Uploading images onto website from local. By implementing this, the filename of the image as well as absolute path of file run.py is required. Then, the uploaded image can be stored in a specific directory and the path of this directory plus the filename will be stored in the database called app.db. When html page needs the image, database will send the path to html and the image will be shown in html file. This is such a difficult work that I think I had made a breakthrough after implemented it.

7) Many to many relationships in database. Since there are more than one admins, an admin can manage many users and, in the meantime, a user can be managed by many admins. Secondly, a user can borrow more than one books and a kind of book can be borrowed by many users. A many to many relationship is a little bit more complex than one to many or one to one relationship.

8) The application of Ajax technology. In this coursework, I have implemented several web template inheritances. Template inheritance can greatly reduce code duplication, making web applications run faster.



Let’s see it in the above picture. The upper one is the base page and the lower two pictures inherit from the upper page. In this way, when user click ‘profile’ button, there is no necessity to load the whole page, which will be a great promotion in efficiency.

2. **A link to the deployed website and username/password if required to access.**

   The coursework was deployed using Ali Cloud, and the website of deployment is: http://47.103.100.101:5000/.

   For normal user and admin, you are supposed to click register to get a new account. If you want to be the super admin, both username and password are “superadmin”. Hope you can enjoy yourself when using my web application!

3. **an evaluation of your implementation.**

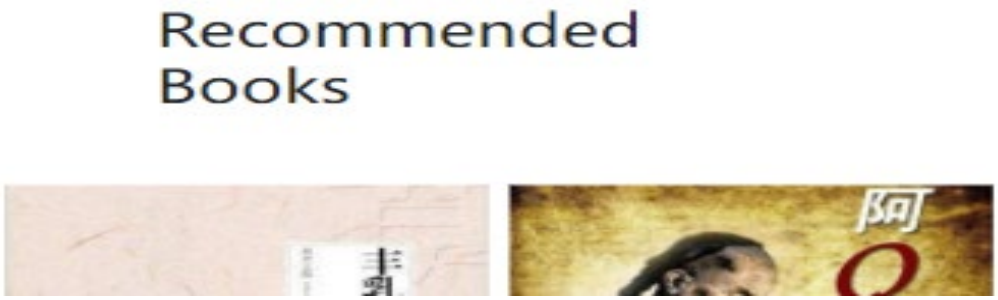
   In this coursework, I have generally implemented the functions required for a library management system. On the front end, the page looks good from the overall layout to every part. The web also achieves responsive layout to a certain extent. In function, add, delete, change, check and other functions can be well realized, flexible use of various forms to obtain data. In terms of the overall architecture, I have realized the three-tier architecture from the front end to the back end and then to the data end. Meanwhile, the whole project has completed the deployment on Ali Cloud and is running in good condition with no abnormal situation found.

   However, there are also some problems that need to be corrected and solved.

- The web page is only partially responsive, and for the rest of the page, when the page shrinks or zooms in, it still doesn't look pretty.

‘Recommended Books’ can display properly when the page is wide enough

However, if the page is not wide enough, Books will wrap itself which makes the page not beautiful. If when the width gets smaller and smaller, the font size can get smaller and smaller and keep the whole text in a single line, it will looks more beautiful.

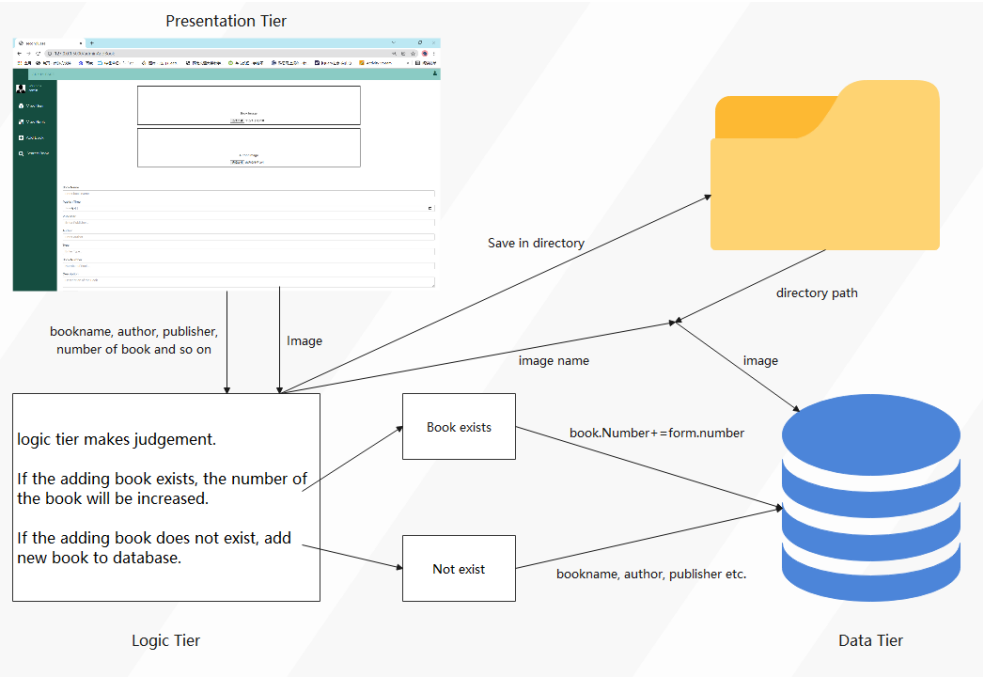


- Currently, the system can only be used for one user, and there is no way to handle high concurrency events, which means that if multiple users visit the site at the same time, the system will likely crash.
- Due to time constraints, some functions have not been implemented, such as users can modify their personal information (avatar, username, etc.).
- Seldom usage of JavaScript, which plays a pivotal role in the production of front-end web pages. Due to the lack of in-depth learning of JavaScript, I can only achieve some simple effects at present, which also leads to the failure of my web pages to achieve perfect adaptive effects.

4. An analysis of your web applications architecture.

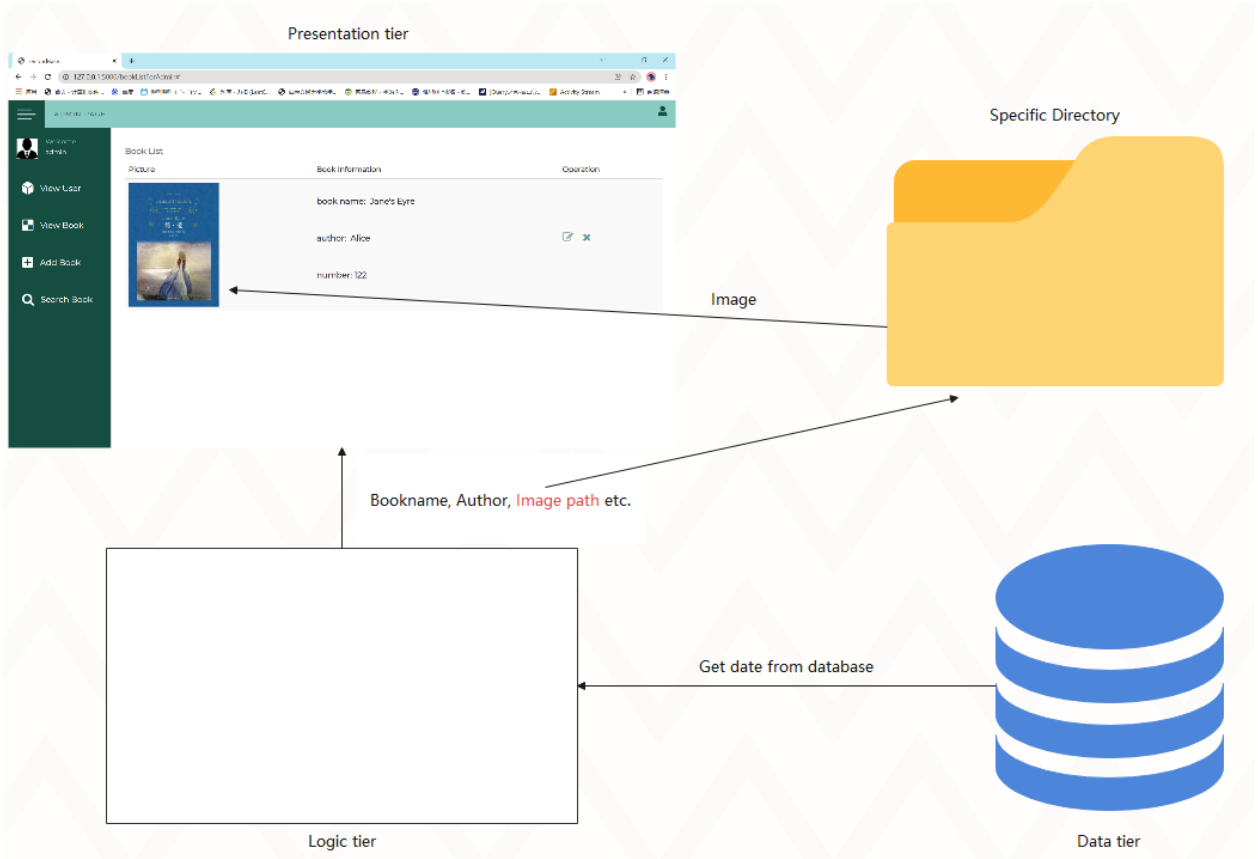
The library management system is designed and developed in strict accordance with the three-tier (presentation tier, logic tier and data tier) architecture of the network.

I would like to take the process of add a book as an example to explain my web applications architecture.



First, the user will input data from the Presentation Tier. There are various types of data, including file data, text data, numeric data, and time data. Logic Tier receives the data and determines and processes it accordingly. Logic Tier determines data other than images. For image data, Logic Tier processes the file name and storage address. Finally, the Logic Tier transfers data to the Data Tier for storage.

When we need to display data or images on the Presentation Tier, we need to get the data we need from the database (title, author, publisher, etc.) and then transfer this data to the Presentation Tier via Logic Tier. The whole process is shown below:



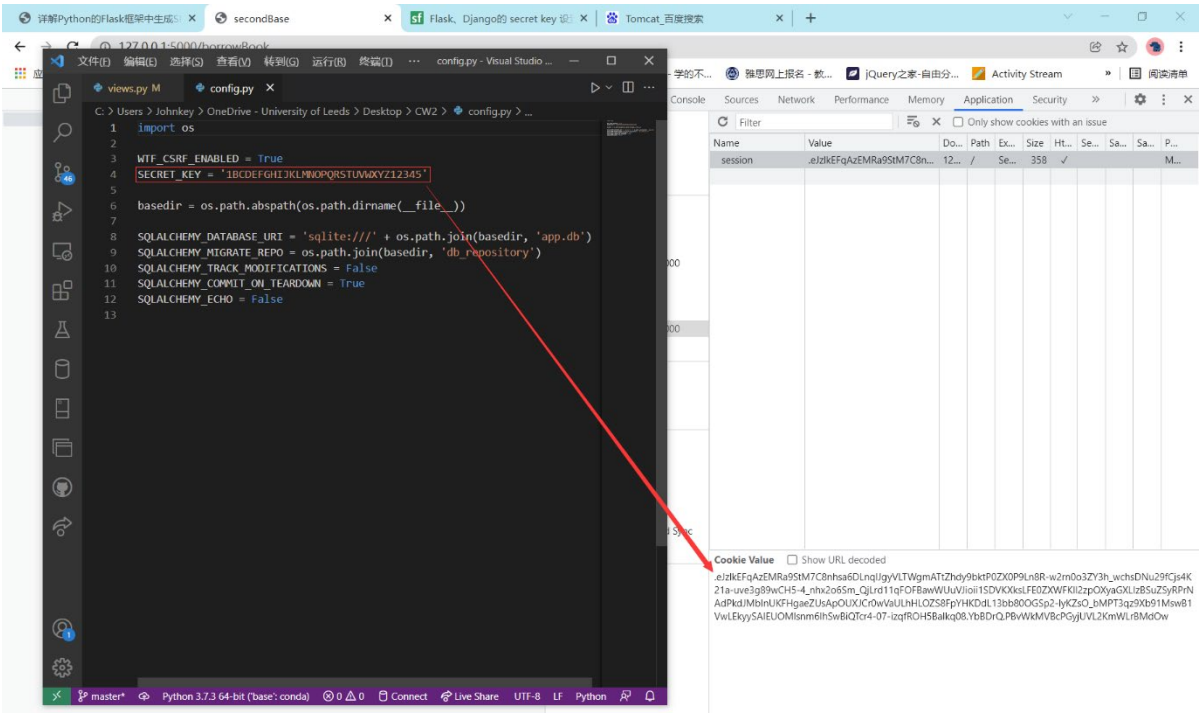
In other cases, registration and password change are similar. The process is that the user inputs on the Presentation Tier, and the data is stored in the database if the Logic Tier judges that the data is successful. If the Logic Tier fails, the front-end page is rendered again and an error is reported. Similarly, login verification is to

compare the data entered by the Presentation tier with the data in the Data Tier in the Logic Tier. If the comparison succeeds, it will be redirected to a new page; if the comparison fails, it will be rendered again and an error will be reported.

5. A description of potential security issues your web application might encounter and how you have mitigated to remove/reduce their impact.

1) Usage of SECRET_KEY

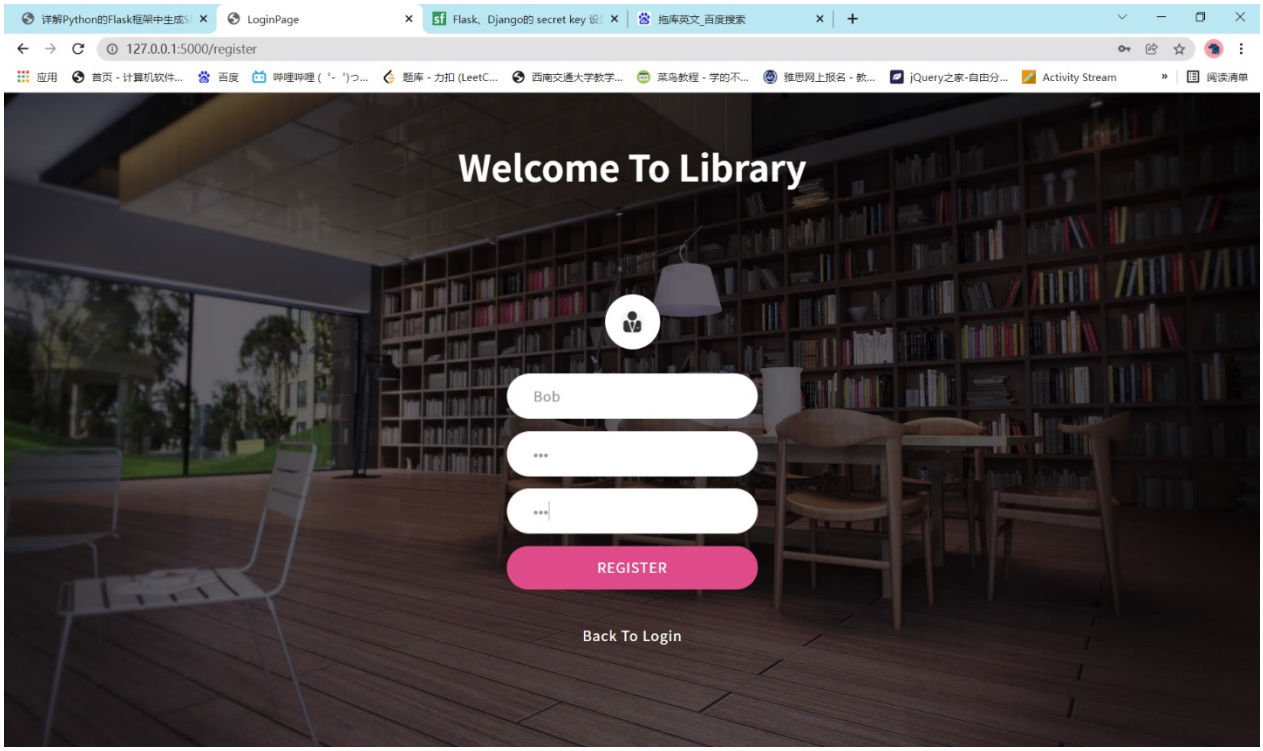
If the data is not encrypted during transmission and storage, the original data will be directly exposed, which is very dangerous. For solving this problem, 'SECRET_KEY' is used. Session, Cookies, and some third-party extensions all use the SECRET_KEY value, which can encrypt the transmitted data. The main method is to provide a value to hash the data, so that the Session data we see are encrypted, which greatly improves the security.



Additionally, this SECRET_KEY value is set in file 'config.py' instead of 'views.py' so that it is less possible to find it.

2) The database may be dragged by hackers, in which case all user and administrator accounts and passwords will be stolen, and their personal information will be leaked. The entire web application is at risk of crashing. However, if we can encrypt the data while it is stored, even if the database is stolen by hackers, all they get is a long list of processed hashes that need to be decoded before they can be used. In this way, users and administrators in the database information is well protected, even if the database is unfortunately stolen, users' personal information is unlikely to be leaked out.

Let's see this process in the following picture:



Now a new user is setting up an account whose username is Bob and password is ‘123’ (Invisible since the type of form is “password”). Now when the user clicks “register” button, let’s see what will happen.

```
C:\Users\Johnkey\Anaconda3\Library\bin\sqlite3.exe
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.
sqlite> .header on
sqlite> .mode column
sqlite> select * from User;
id          userName    password_hash                                                                                       bookCount   criminalRecord  firstBorrowBookId  secondBorrowB
ookId  thirdBorrowBookId
----------  ----------  --------------------------------------------------------------------------------------------------  ----------  --------------  -----------------  -------------
-----  -----------------
1           alice       pbkdf2:sha256:50000$KCVOFUNv$a91bf765aeef1b71718785ca40afca423462c78c8ec15e441c652239d3e13557  2           0               1                  2
       0
2           Johnkey     pbkdf2:sha256:50000$aCZPagbr$c3abcc1a2527b8c42e71e9d12d053cf10252fd72d39e65c589245110737f413c  0           0               0                  0
       0
3           Bob         pbkdf2:sha256:50000$e7m5Gp8I$c72777481429f29774dc5f179c895bbcdfd9858f34d578e85218cb25bb586cdf  0           0               0                  0
       0
sqlite>
```

As you can see, the database already has Bob as the user, and the password column is not 123, but a long hash string. In this way, the user's password is encrypted and protected, thus improving the security of the entire Web application.

6. Reference

Images all gotten from: https://www.ivsky.com/ and https://pixabay.com/.

Some HTML and jQuery templates gotten from: http://www.htmleaf.com/ and https://www.codehim.com/.