****

**COMP9900 Information Technology Project**

**Project Report**

**Term 3 2020**

Team chongchongchong

Mentor: Iwan Budiman

Submission Date: 2020-11-16

|  |  |  |
| --- | --- | --- |
| Name | zID | Role |
| Yipeng Han | Z5192958 | Scrum Master/Developer |
| Mingyan Yu | z5196418 | Developer |
| Ziwei Li | z5187737 | Developer |
| Kaisen Luo | z5185842 | Developer |

**Project Overview**

**Background**

In recent years, watching movies has become one of the most enjoyable activities for people to release their pressure and share their time with friends and families. When the market of movies is developing rapidly, thousands of new movies are released annually which make it hard for people to find the movies they want most in a short time. Currently, there are some websites related to finding movies and providing the information about the movies at the same time but however, their efficiency and functions are not totally user friendly. Some of their functions are useless for most of the users and slow down the searching period. Therefore, we want to implement a website to help the users easily achieve their goal of finding movies and even sharing their ideas with the other users.

**What is FilmFinder?**

The FilmFinder is a professional website for searching movies. The users can search the movies they want using the name, genre, description, and director of the movie. The searching results will show all the relevant movies and their rates. The users can also search for the detailed information of a movie including the name, released time, director, description, newest rates, and the comments from the other users. Besides, the users can add their favourite movies to their wishlist for reference and manage them at any time. Meanwhile, the users can rate and comment a movie based on their own idea and share it with the other users. The system will recommend movies to the users based on their viewing and rating history.

This report will describe the implementation of FilmFinder. The whole implementation process follows the sprints in Jira and all the user stories supposed in the submitted project proposal are achieved. The outcome of each implementation process has been uploaded to GitHub and demoed twice to the course supervisor in arranged time.

During the implementation, our group members use GitHub and WeChat to share the files and write weekly working diary. We use WeChat group and Zoom conference to discuss the problems and ideas happened in the implementation and complete the retrospectives after each demonstration.

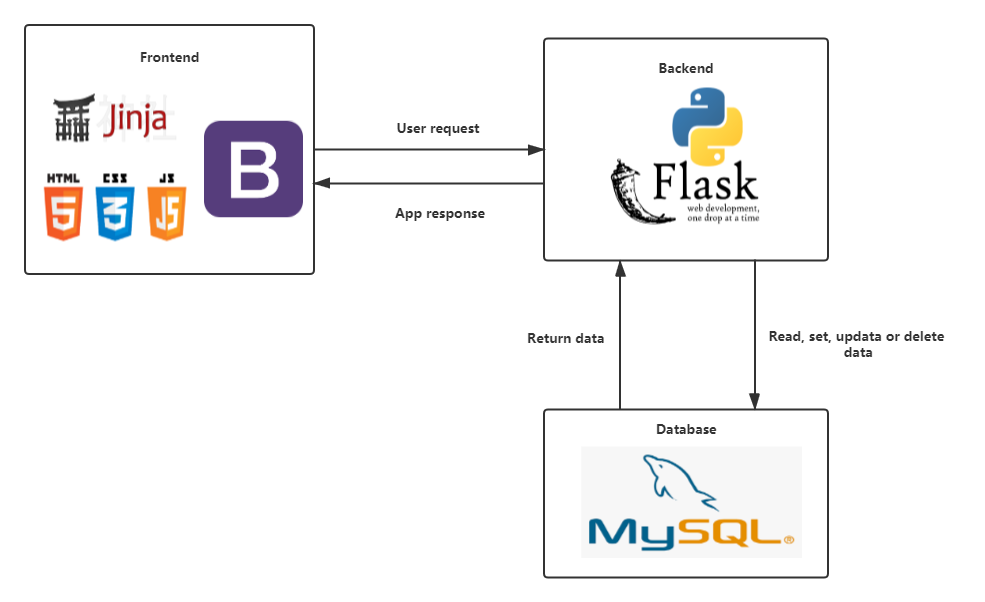
In the following part of this report, we will show the features, architecture, and highlights of FilmFinder as well as the challenges we faced during the implementation. Before the end of this report, we will provide the documents and guidance to install and operate the FilmFinder.

**System Architecture**

**System Design**

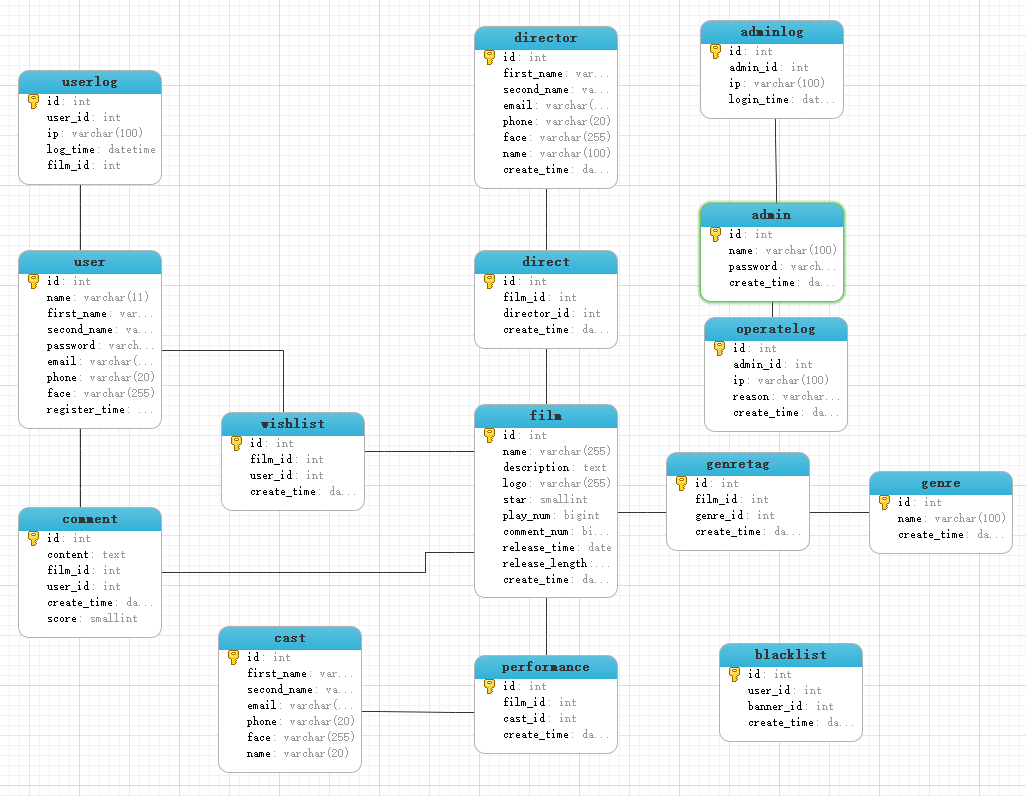
The FilmFinder system consists of three layers, namely the front-end, the back-end and the database. The front-end is equivalent to the face of the system. For users, the front-end is where users and administrators interact directly with the program. The back-end is the main logic module for program operation, mainly processing front-end interactive information, and direct interaction with the database. After getting the front-end request, the back-end can read data from the database and return to the front-end for display. The database is mainly used to store all information related to the program.

In this project, we use python's flask framework for back-end development and use SQLAlchemy to separate the back-end from the database. SQLAlchemy is an object-relational mapping library. In the database, we chose the powerful relational database Mysql. For the front end, we used the Jinja module of python to render the basic HTML, CSS, JavaScript and bootstrap framework.

****

**Data View**

We chose mysql as the database of our system. Mysql is a relational database. It has a strong versatility, which is conducive to transplantation on different operating systems. Below is the ER diagram of our database:

****

**Key Feature Highlights**

**Information retrieval and restoration**

FilmFinder can help the users search the movies via different kinds of information including movie name, description, genre, and director. After searching, the results will show all relevant movies and its detailed information including name, description, genre, director, average rates, and reviews. This feature helps the users achieve all the information they want by just input a single relevant word and click searching button.

**Recommendation system**

FilmFinder provides a recommendation system to its users. This feature can recommend relevant movies to the users based on the users’ searching and viewing history. It will help the users easily achieve their favourite kind of movies and make the whole website much more personalized.

**Data maintaining and operating**

The database of FilmFinder not just can store the information of plenty of movies, it can also store and keep all the users’ information and preference. The user can modify their own information like their own username or login password in the account page. For the movies, the user likes, he or she can add them into their own wishlist and remove them when they change their mind. All the operations above can be achieved once the user clicks without any delay.

**Information sharing and blacklist**

The users in FilmFinder can rate and comment movies according to their own ideas. At the same time, users can see the others’ rates and comments about the movie and have the access to put the users into their own blacklist if they hold different ideas. Once addressed into the blacklist, that user’s rates and comments about all movies will not be seen by the user who holds the blacklist. The rates of each movie will be recounted automatically except the rates by the users in the blacklist. This feature can help to minimize the influence of malicious users and allow the users to communicate with the people who have similar ideas or interests.

**System and Feature Walkthrough**

**Customer App**

**Landing Page, Register and Sign in**

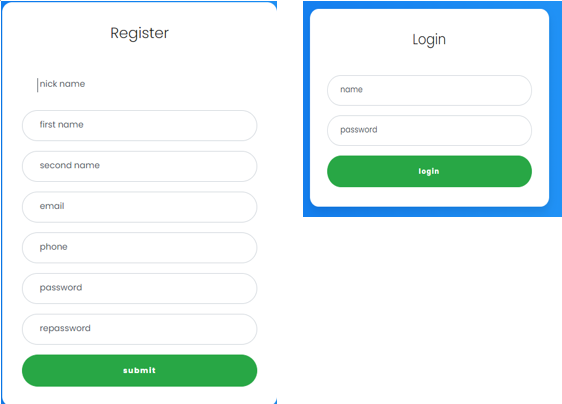
The FilmFinder program allows users to search for movies on the website, view related information and film reviews of movies in the state of visitor. In addition, other functions require users to log in to use them.



Users can enter nickname, name, email, mobile phone number and other information to register an account. After registering an account, users can change their information in the profile, and upload an avatar to show their personality.

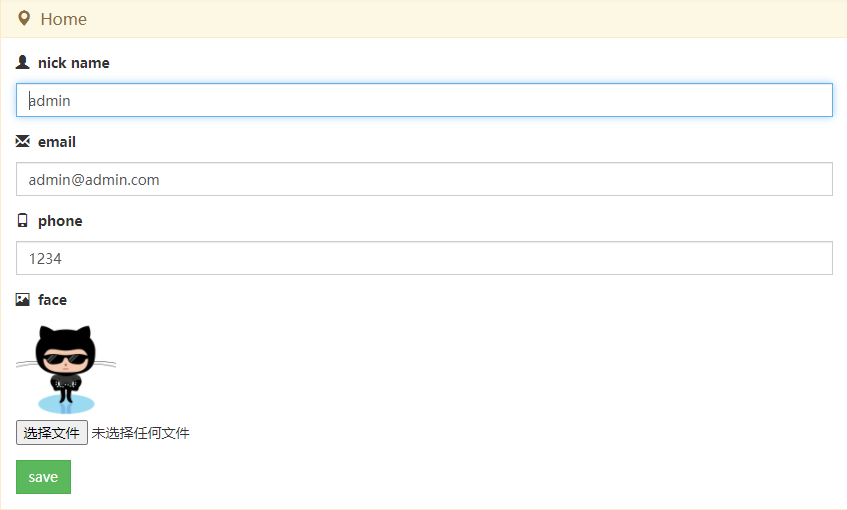
To protect user passwords from potential leaks, the system stores user passwords

As a SHA256 hexadecimal digest (64 characters long) instead of a plain text password, it is displayed as the password varchar (64) column in the database system. It ensures that the user’s password will do not disclose even if there is a database leak.



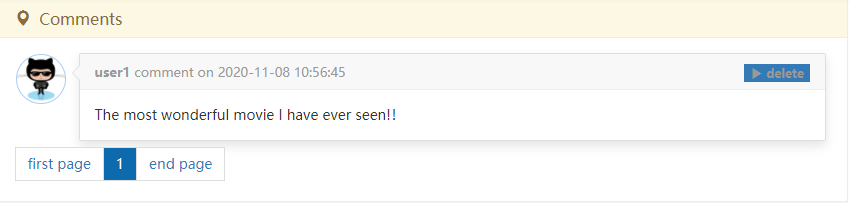
**User’s profile**

After logging in to the website, users can click the user button at the top right to enter their homepage at any time. Here users can change their login name, email address and mobile phone number, as well as upload new avatars, which support jpg format.



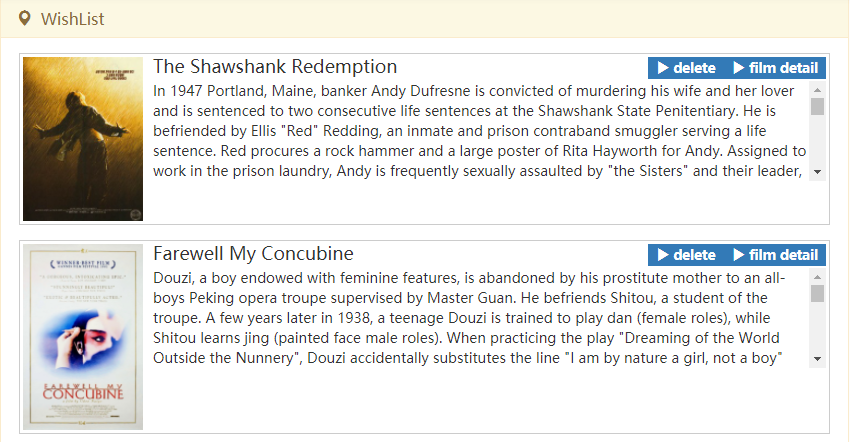
**User’s comment page**

The comment button can be found in the navigation bar on the left side of the user homepage. The user clicks to enter the user comment page, where the user can view all the film reviews they have posted, and the user can turn the page to view the film reviews. Users can delete the film reviews they do not like, and the scores they gave when they posted this film review are also deleted.

****

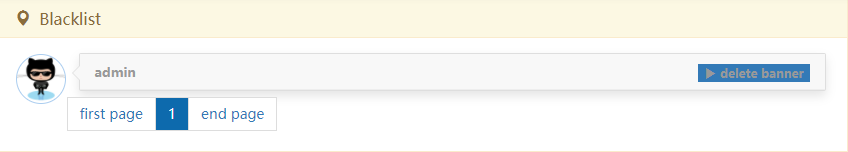
**User’s wishlist page**

Users can click wishlist button which is on the left side of the homepage to enter the user’s wishlist interface. Users can view all the movies they put on wishlist here. Users can click on film detail on the right side of the movie to enter the detailed information page of the movie. Users can also click delete to remove the movie from the wishlist.

****

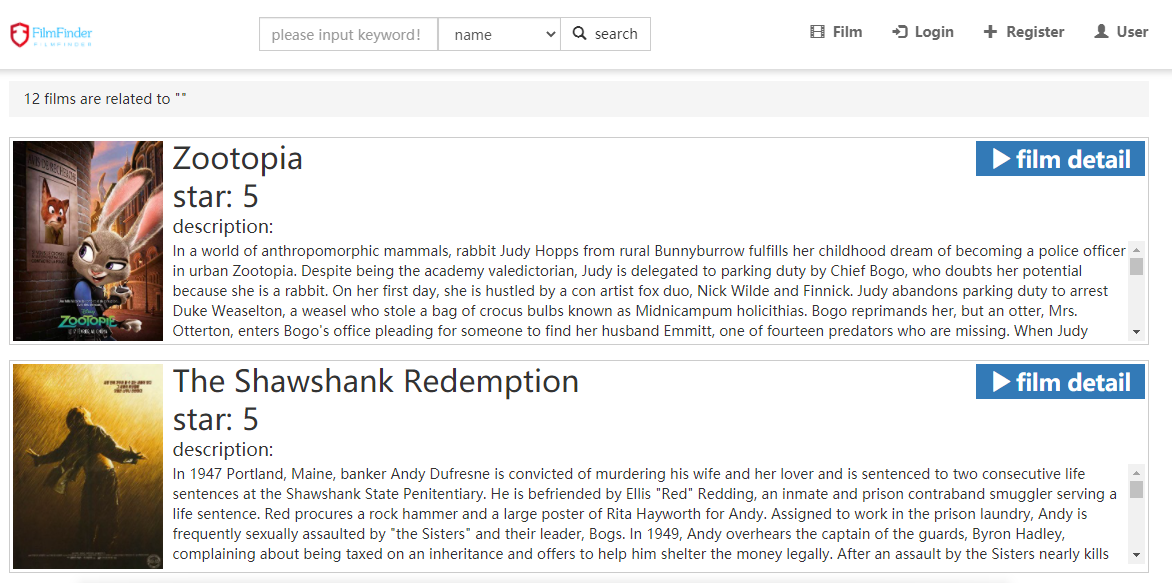
**User’s blacklist page**

The user can click the blacklist button to enter the user's blacklist page. Users can view all users in the blacklist on the page. Users can also remove them from the blacklist by clicking the delete button on the right side.



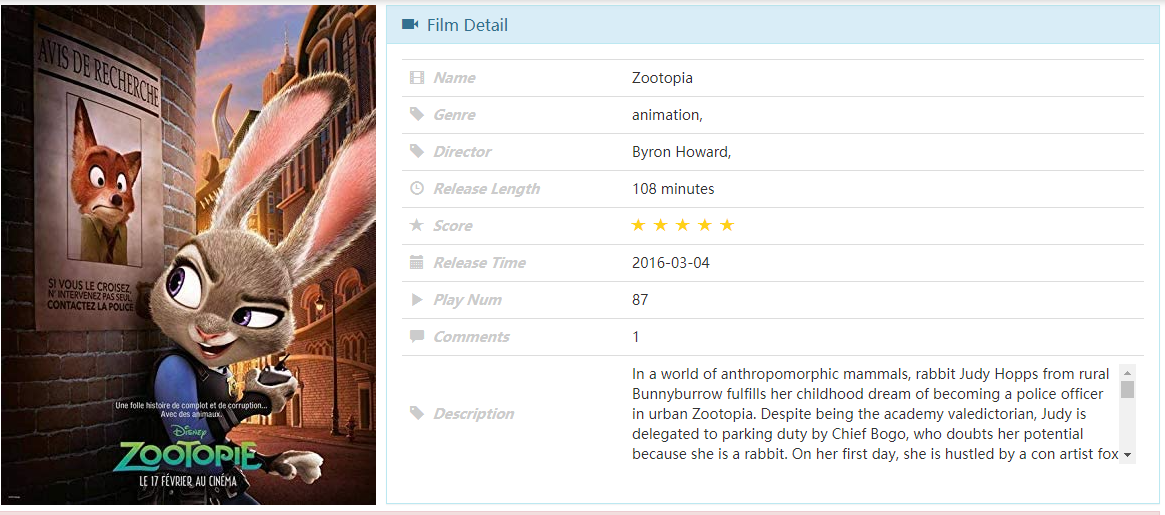
**Movie browsing and searching**

This feature allows all website users to search for movies and view the search results. Searching function is not only effective for registered users, visitors without a registered account can also use the search function of the website. The website provides a total of four search methods, which can be searched by name, director, genre, and description. The search supports fuzzy search and can ignore the difference in capitalization.

****

**Movie detail page**

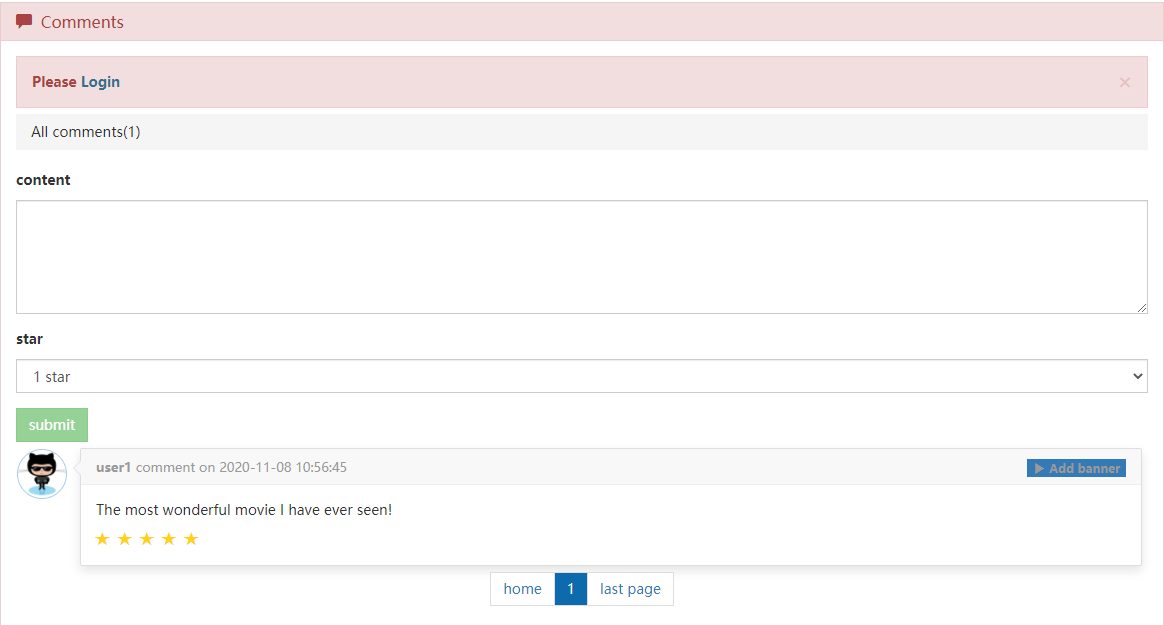
The user can click on the film detail on the right side of the search results to enter the detailed information page of the movie. The upper side of the page displays the detailed information of the movie, including director, genre, duration, description, etc. On the left is the movie poster.



**Film review, wishlist and blacklist**

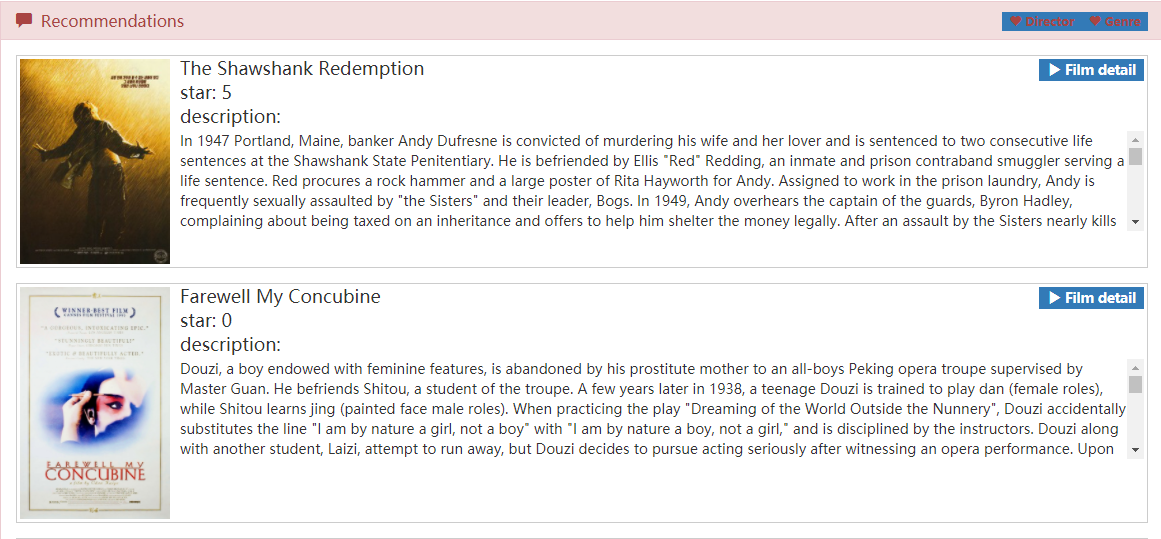
Scroll down on the detail page is the film review section of the movie. Here users can view the relevant ratings of the movie in the login and guest status. In addition to viewing movie reviews, users can also write movie reviews and rate movies. The user cannot be in the guest state when using this function and can only use this function after logging in to the account. Users can also click on wishlist on this page to add movies to their wishlist. This function also requires the user to log in before it can be used.

If users find an offensive comment, they can click the Add banner button next to the user. This will add users to the blacklist. Their comments will no longer be displayed on the page, and the movie ratings will be recalculated after excluding their ratings. This function also requires the user to be logged in.



**Film recommendation**

In the recommendation section, based on the user's current latest movie reviews, recommend movies that may be of interest to the user. If the user is not logged in to the system, the user is recommended to the movie with a higher score in the current system library. If the user has logged into the system, recommendations are made based on the user's reviews. Moreover, the user can also select the recommendation type, and the user can choose to recommend according to the director or genre.



**Admin Dashboard**

**Future Feature Roadmap**

**Implementation Challenges**

**Third-party Functionalities/Frameworks**

**Library List**

We will introduce here some third-party libraries that we use in the project. We use Flask as a web framework to implement our website. We also use Flask-SQLalchemy as the connector between the backend and the database. It converts data tuples in the database into Python objects, which allows us to easily manipulate the data. In addition, we use Flask-WTF for all form validation to check whether user input is valid on different pages. The specific list is as follows:

|  |  |
| --- | --- |
| **Libraries** | **Edition** |
| click | 7.1.2 |
| Flask | 1.1.2 |
| Flask-SQLAlchemy | 2.4.4 |
| Flask-WTF | 0.14.3 |
| itsdangerous | 1.1.0 |
| Jinja2 | 2.11.2 |
| MarkupSafe | 1.1.1 |
| PyMySQL | 0.10.1 |
| SQLAlchemy | 1.3.19 |
| Werkzeug | 1.0.1 |
| WTForms | 2.3.3 |

**Library Introduction**

**PyMySQL 0.10.1**

PyMySQL is a python MySQL client library. It is used to connect SQLAlchemy to the database system. Most of the APIs in PyMySQL are able to use with MySQLclient and MySQLdb.

**SQLAlchemy 1.3.19**

SQLAlchemy is an open-source SQL toolkit for python programming. It helps to connect the database and PyMySQL and provides a series of enterprise-level persistence patterns.

**Flask 1.1.2**

Flask is a web framework which serves the development and extension during implementation. The flask is based on Werkzeug and Jinja.

**Flask-WTF 0.14.3**

Flask-WTF is a feature to provide integration with WTForms. It is safely secured and allowed file upload works with Flask upload.

**Flask-SQLAlchemy 2.4.4**

Flask-SQLAlchemy is an extension for Flask and it provides support for SQLAlchemy in the implementation by some defaults and helpers. It will be easier and efficient to implement and achieve our goal when using Flask-SQLAlchemy.

**Werkzeug 1.0.1**

Werkzeug is a comprehensive WSGI web application library for python programming.

**Jinja2 2.11.2**

Jinja2 is a template engine for python programming. It is a text-based template so that it can be used to produce different markup or python source code.

**Click 7.1.2**

Click is a python package that allows to make command line interfaces with limited amounts of code. Besides, Click can continue to act as intended when multiple Clicks are used together.

**MarkupSafe 1.1.1**

MarkupSafe can be used in HTML and XML to transfer those characters with special meanings into real meaningful characters. It helps to display the inputs by those untrusted users.

**WTForms 2.3.3**

WTForms is a flexible forms validation and rendering library for python web development. It can help to create web forms and use the information in the forms easily with a python model.

**Itsdangerous 1.1.0**

Itsdangerous is used to help send the data safely in an untrusted environment. The basic idea is to cypher the data you want to send and only the true receiver can decipher and read your data. The other people who are not the matched receiver or do not have the secret key cannot read or modify the data.

**System Setup Documentation**

**Reference**