#--------READ ME-----------

**To initialize:**

1. In the generate\_data.py file, modify the host, user and passwd variables to the appropriate values (lines 10-13). Add the host, user and password of your MySQL account.

2. In the app.py file, modify the host, user and password variables to the appropriate values (lines 6-9).

**To set up the database:**

1. Login to MySQL and create a database named book\_management\_project by using the command

create database if not exists book\_management\_project;

2. Run the generate\_data.py file by using the command on terminal

python generate\_data.py

**To use the application:**

1. Run the command on terminal:

python app.py runserver -d

2. The terminal shows a message: "Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)". Right click on the url and

click on the "Open Link" option.

3. The application opens up on a web browser (Mozilla Firefox in my case).

Note: If you close the tab and try to repeat the above command, it may show an error since the socket is already in use. To get rid of this error, type the commands

sudo lsof -i | grep "py"

This gives you the process with its ID (second entry). Then type:

sudo kill -9 <process\_id>

*To Login as user or manager:*

1. Enter the correct user credentials in the appropriate form. Since data generated is random, one can look up the User table for credentials. Just type ‘select \* from User;’ and choose any user credentials.

Valid credentials for the manager table are : Username: m012, Password: mwer

*To Sign Up:*

1. Create a profile by providing the necessary information. The valid genres are :

genre = ['Mystery', 'Fantasy', 'Drama', 'Romance', 'Non-fiction']. No quotations needed.

2. Login using the credentials of the newly created account.

*User Main Page:*

1. To find books by author, specify the author's name (such as "Oscar Wilde" or "Jane Austen"- no quotes needed) in the dialog box.

2. To find books by genre, specify the genre. Use only the ones shown below.

genre = ['Mystery', 'Fantasy', 'Drama', 'Romance', 'Non-fiction']

3. To add book to your Read bookshelf, specify the book title (such as "Rebecca" or "Persuasion") and a rating (integer 1-5, inclusive of both)

4. Press the respective buttons for the functionalities:

Find Books Read by User , Recommend book , Find the Most Critically-Acclaimed Book , Find Books Ordered by User, Find Book with Maximum Orders

5. To find highest rated book of author, specify the author's name (such as "Oscar Wilde")

6. To find books written by authors who have won a specific award, use either of the 3 awards:

'Hugo Award', 'Man Booker Prize', 'Pulitzer Prize'. (No quotes needed)

7. To find bookstores which carry a book, specify the book title. The output is a storeID.

8. To order a book from a bookstore, specify the book title, number of copies (an integer) and the storeID (which is the output of the previous functionality).

9. Click on Logout button to logout.

*Manager Main Page:*

1. To find books ordered by a user from the bookstore at which the manager works, specify the username in the input box.

2. Click on the button " Find Orders Made to the Bookstore " to find all orders made by users to the bookstore.

3. Click on the button " See Pending Orders " to see all pending orders made by store employees to the inventory.

4. To approve a pending order, specify a supplyID for the " Approve Pending Orders " form. The supplyID can be found from the functionality "See Pending Orders".

5. To logout, click on the logout button.

Data Preparation and Setup

1. Database System: MySQL

Version: 5.7.28

Installation Link: <https://dev.mysql.com/downloads/mysql/5.7.html>

mysqld Ver 5.7.28-0ubuntu0.16.04.2 for Linux on x86\_64 ((Ubuntu))

2. I have used automatically generated database. Run the python file ‘generate\_data.py’ to create the tables in a database. Instructions above.

3.

1. Create a database ‘book\_management\_project’ in your MySQL account. Run the python file ‘generate\_data.py’ to create the tables in the created database. This automatically loads the data into the database.

2. The tools used for loading the database include the mysql.connector library of Python. The commands ‘pip **install MySQL**-python’ and ‘pip **install mysql**-**connector**-python’ work on the terminal.

Installation link: <https://dev.mysql.com/doc/connector-python/en/connector-python-installation-binary.html>

Note that this library was used in the generate\_data.py script to insert tuples into relations, create relations and update relations. The string and random libraries of Python were used to generate random data.

4. Not applicable. Did not benchmark database systems.

5. I did generate data. This was done by the generate\_data.py script.

To generate data, create a database ‘book\_management\_project’ in your MySQL account. Edit the lines (10-13) in generate\_data.py and add your own credentials. Run the python file ‘generate\_data.py’ to create the required tables. You can use the command

python generate\_data.py

Application and code

1. Programming language: Python

2. Third party libraries:

The random and string libraries are part of Python.

To install the mysql.connector library for generate\_data.py (to insert tuples, etc): The commands ‘pip **install MySQL**-python’ and ‘pip **install mysql**-**connector**-python’ work on the terminal. I used the version '2.2.2b1'

Installation link: : <https://dev.mysql.com/doc/connector-python/en/connector-python-installation-binary.html>

Flask library used to connect the Python code to HTML webpages. Can use the command ‘pip install Flask’ or ‘pip install -U Flask’ in terminal/command line.

Installation link: <https://flask.palletsprojects.com/en/1.1.x/installation/>

I used the version '1.1.1'

flask\_mysqldb library to connect the HTML webpages to MySQL database. Can use the command ‘pip install flask-mysql’ on the terminal/command line.

Installation link: <https://pypi.org/project/Flask-MySQLdb/> (works with Linux/Mac)

<https://stackoverflow.com/questions/47015590/python-how-to-run-flask-mysqldb-on-windows-machine> (instructions for Windows)

<https://stackoverflow.com/questions/54043953/how-to-fix-install-flask-mysqdb-module-in-windows-10> (instructions for Windows)

3. The GUI can be used by running the command on terminal/command line:

python app.py runserver -d

If you want to run this again, you may get an error saying that the socket address is already in use (OSError: [Errno 98] Address already in use). To overcome this run the command:

sudo lsof -i | grep "py"

This gives the python files running along with their process id (second entry in the row). Kill that process by using the below command.

sudo kill -9 <process\_id>

Note: Since sudo commands are used, you will have to enter your system’s password for the commands to work.

4. Make sure that the files are not moved. The ‘books.jpg’ image should be in the *static* folder and the .html and .css files should be in the *templates* folder.

IMPORTANT: Make sure that before you use the generate\_data.py and app.py files, change the host, user and password values to those matching yours. In generate\_data.py, these are lines 10-13 whereas in app.pym, these are lines 6-9.

Code Documentation and References

1. No. All code was written by me.

2. No online code used.

3. List of files written by me:

In the templates folder:

index.html

login.html

manager.html

styles.css

In the Project (main) folder:

app.py

generate\_data.py

I have not used any code from online sources but I have used websites like StackOverflow and TutorialPoint to read up on MySQL syntax and common Flask mistakes. No code was copied though.