READ ME INSTRUCTIONS FILE

Clear Room Application and Database

Front of House Application

To be able to run the FOH applications relevant to the Clear Room data system, all files listed below will be needed. You can find this in the current repository.

Files needed are in the Flask folder, this includes:

- _pycache__ folder
- static folder
    - Logo
        - Class1.jpg
        - Class2.jpg
        - Class3.jpg
        - Clear Room.png
        - Group 27.png
        - uts logo.png
        - uts-logo-University-Of-Technology-Sydney.png
    - bookings CSS
    - confirmation CSS
    - feedback CSS
    - home CSS
    - login CSS
- templates folder
    - app.py.code-profile
    - bookings.html
    - confirmation.html
    - feedbackreceived.html
    - home.html
    - login.html
- main.py

Prerequisites

Python and MySQL will be required to run a connection on local host.

Ensure the necessary packages seen below are installed:

- mysql-connector-python==8.0.26
- Flask
- Datetime

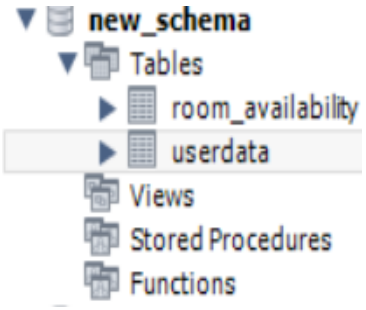
If not already installed, you can do this by executing the following commands into terminal:

- ‘pip install mysql-connector-python’
- ‘pip install flask’
- ‘pip install datetime’

Step 1

To create the database:

Open mySQL Workbench and set a root user password and then load the ‘userdata.csv’ to table called ‘userdata’ and ‘studyroom.csv’ to another table called ‘room_availability’ under the same schema named ‘new_schema’. This will create your SQL database.



Update your details of host, user, password that you will be referring to in each function in main.py. If the program is being run on a localhost server, this line of code can remain the same as seen in the downloaded file.

Once these changes have been successfully made, the mySQL connection should be set up.

Step 2

In Flask you may load existing CVSs with the same name as ‘bookings.csv’ and/or ‘feedback.csv’ if you wish to add the bookings and feedback collected to those files. You could alternatively use the provided csv files, create your own, load an empty file or not load any file at all. However, the file must be named ‘bookings.csv’ and ‘feedback.csv’ if one is uploaded. Ensure that when running the code, the csv is not open on any other applications.

Step 3

To make use of the booking and feedback system, confirm that the main.py file should include all necessary functions. Ensure the remaining files in the flask folder are also in the same directory when running the program.

The functions available following importing the main.py into the application are shown below:

Defines a route for the default page - in this case the log in page

@app.route('/')

def loginpage():

Defines a route for home page

@app.route('/home')

def home():

Defines a route for confirmation page

@app.route('/confirm', methods=['POST'])

def confirm():

Defines a route for the login page if redirected after initial webpage opening

@app.route('/login', methods=['POST', 'GET'])

def login():

Defines a route for log out -> takes you back to log in

@app.route('/log out')

def logout():

Writes data recorded from user interface into corresponding csv files

write_feedback_to_csv(data)

write_booking_to_csv(data)

Defines a route for bookings page

@app.route('/bookings', methods=['GET', 'POST'])

def bookings()

Converts check-in and check-out times to datetime objects

check_in_datetime = datetime.strptime(check_in_date + ' ' + check_in_time, '%Y-%m-%d %H:%M')

check_out_datetime = datetime.strptime(check_in_date + ' ' + check_out_time, '%Y-%m-%d %H:%M')

Checks room availability

is_room_available(room_number,num_guests):

Step 4

After running the code, open the link provided. This should open to your web browser that displays the log in page. To log in successfully retrieve a matching pair of email and password from the userdata.csv that was loaded into the SQL database from before. Once logged in you will be able to make bookings and feedbacks successfully.