COP 4521 Spring 2021
Homework 3

Total Points: 20

Due: Tuesday, 02/16/2020, 11:59:00 PM EDT

1 Objective

The objective is to make sure

1. You are familiar with the principles of Information Management.
2. You can set up a simple Database and interact with it using SQL.
3. You can set up a `flask` server in a Python application to work with the Database.

2 Specifications

For this homework you will be using the `sqlite3` and `flask` libraries to create a database for a text-only ratings and reviews website, to be used by local restaurants. Samples for the different pages of the web application are shown on Section 3. Please make sure you conform to the following specifications:

1. Create a `sqlite3` database for your application. Call it `reviewData.db` (1 points)
2. Write a Python script called `setup.py` that connects to the database and creates the tables specified in Steps 3 and 4. (1 points)
3. Create a table called `Reviews` with the following fields (1 points):
   - Username - 40 character text field
   - Restaurant - 50 character text field
   - ReviewTime - SQL Datetime
   - Rating - float
   - Review - 500 character text field
4. Create a table called `Ratings` with the following fields (1 points):
   - Restaurant - 50 character text field
   - Food - float
   - Service - float
   - Ambience - float
   - Price - float
   - Overall - float
5. Create an HTML page called `index.html`. This page will serve as the homepage for the website. It should include a Welcome Message and links to take users to the other pages in the website. (1 points)

6. Create an HTML page called `addReview.html`. This page will display a form for a user to submit a review for a local restaurant. This will include a "Submit" button. Upon clicking the submit button, the application should insert a row in the Reviews and `Ratings` tables. (1 points)

7. Create an HTML page called `getReviews.html`. This page will display a form for a user to get all the reviews for a particular restaurant. This will include a "Submit" button that will fetch all the reviews for the given restaurant from the `Reviews` table and show them in the `showReviews.html` page. (1 points)

8. Create an HTML page called `showReviews.html`. This page will display the Username, Review and Overall ratings for a particular restaurant in a table format. This page will be rendered as a result of clicking on "Submit" in the `getReviews.html` page. (1 points)

9. Create an HTML page called `showReport.html`. This page will fetch and display all the ratings of the top - 10 restaurants by overall rating. If different restaurants have the same overall rating, order them alphabetically. If there are fewer than 10 restaurants in the database, show all of them. (1 points)

10. All pages except the homepage should have a "back to home" link. (1 points)

11. Write a Python script called `app.py` that is going to essentially run the application. This script should contain functions to:

    (a) A function to render the homepage for the root directory of the application. (1 points)

    (b) A function to add rows to the 2 tables triggered by submitting the form on `addReview.html`. The data from a single form should be separated into data pertinent to each table and then run 2 SQL insert queries. Make sure you handle exceptional situations by rolling back. (2 points)

    (c) A function to fetch all reviews for a restaurant, triggered by submitting the form on `getReviews.html` and rendering `showReviews.html`. (2 points)

    (d) A function to run the query to get the top-10 restaurants and render the `showReport.html` page. (3 points)

    (e) If this were the main module, run the application. (1 points)

12. Please make sure your submission is well documented (1 points)

13. Your submission `HW3.tar` should contain the following files:

    - `setup.py`
    - `app.py`
    - `index.html`
    - `showReviews.html`
    - `showReport.html`
    - `addReview.html`
    - `getReview.html`

14. Please note that the SQL commands should handle ALL of the database interactions. You should not be using Python functions to filter the results returned by a slecte all or similar SQL command.

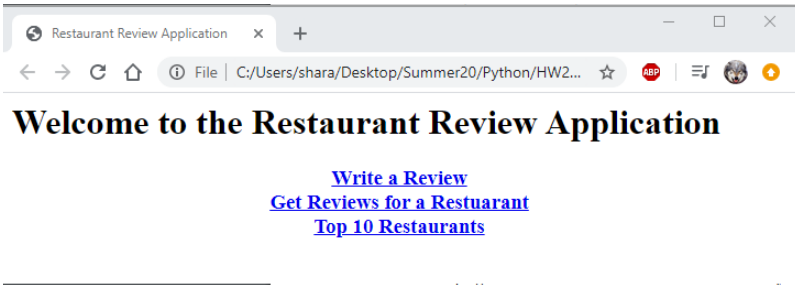
15. flask requires that the templated HTML files will be in a folder called "templates" in the same folder as the python file. We will use the same setup for tests, even if folder organization is not required for submission.

3 Sample

You do not have to worry about verifying if the data entered by the users would fit in the text fields. The user ratings will also definitely be a number between 1 and 5 (inclusive). This verification will usually be handled by client-side Javascript. This is not the objective of the assignment, nor is it one of the learning objectives of the course.

The samples below are strictly functional and are not very aesthetic, again because that would be handled by client-side CSS, which is not one of the objectives of the assignment.

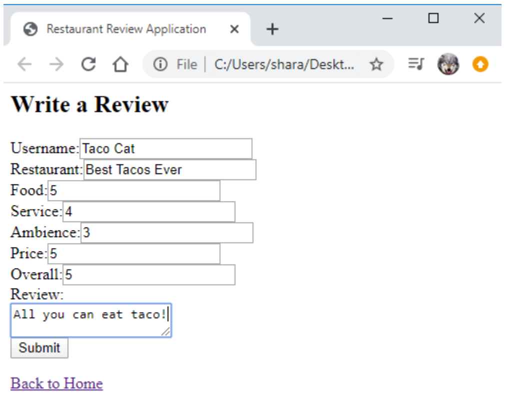
1. `index.html`


Restaurant Review Application
File | C:/Users/shara/Desktop/Summer20/Python/HW2...

Welcome to the Restaurant Review Application

Write a Review
Get Reviews for a Restuarant
Top 10 Restaurants

2. `addReview.html`



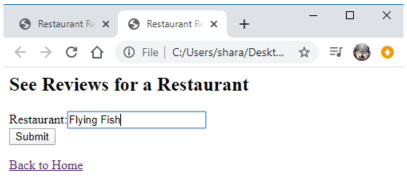
Write a Review

Username: Taco Cat
Restaurant: Best Tacos Ever
Food: 5
Service: 4
Ambience: 3
Price: 5
Overall: 5
Review:
All you can eat taco!

Submit

Back to Home

3. `getReviews.html`

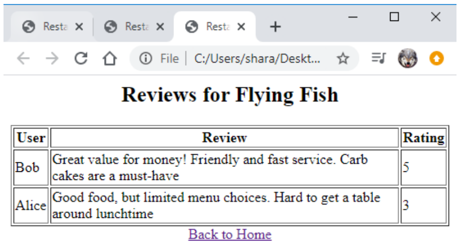


See Reviews for a Restaurant

Restaurant: Flying Fish
Submit

Back to Home

4. `showReviews.html`

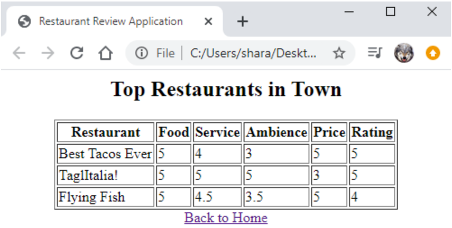


Reviews for Flying Fish

User	Review	Rating
Bob	Great value for money! Friendly and fast service. Carb cakes are a must-have	5
Alice	Good food, but limited menu choices. Hard to get a table around lunchtime	3

Back to Home

5. `showtReport.html`



Top Restaurants in Town

Restaurant	Food	Service	Ambience	Price	Rating
Best Tacos Ever	5	4	3	5	5
TaglItalia!	5	5	5	3	5
Flying Fish	5	4.5	3.5	5	4

Back to Home

4 General Guidelines

- Add a comment with your name, FSUID, the due date and the line "The program in this file is the individual work of <your name>" on all files. Not having this in each of your files would result in a loss of 1 points.
- Create a tarball called `HW3.tar` with all your files. This is the only file you will be turning in.
- If we have listed a specification and allocated point for it, you will lose points if that particular item is missing from your code, even if it is trivial.
- Your outputs will be different from mine depending on the data in your database.
- Your program should load and run without issues. Every interpretation error or HTML/flask exception or SQL error will result in a loss of 1 points each.
- You are restricted to standard Python (built-ins), `sqlite3` and `flask` libraries. Use of any other libraries would result in loss of 2 points per library.
- Testing your program thoroughly is a part of writing good code. We give you sample runs to make sure you match our output requirements and to get a general idea of how we would test your code.
- Only a file turned in through Canvas counts as a submission. A file on your computer, even if it has not been edited after the deadline, does not count.
- Please adhere to the Academic Honor Policy.
- The student is responsible for making sure they have turned in the right file(s). We will not accept any excuses about inadvertently modifying or deleting files, turning in incomplete tarballs, or turning in the wrong files.
- **Program submissions** should be done through the Canvas class page, under the assignments tab (if it's not there yet I'll create it soon.) Do not send program submissions through e-mail – e-mail attachments will not be accepted as valid submissions.

- The ONLY file you will submit via Canvas is HW3.tar
- **General Advice** - always keep an untouched copy of your finished homework files in your email. These files will have a time-stamp which will show when they were last worked on and will serve as a backup in case you ever have legitimate problems with submitting files through Canvas. Do this for ALL programs.