THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE

EXAMINATIONS OF JUNE 2020

Code and Name of Course: INFO 2602 Web Programming and Technologies 1
Paper: 1

Date and Time: Duration: 2 Days

INSTRUCTIONS TO CANDIDATES: This paper has 9 pages and 5 questions.

Answer All Questions

1 Prompt

Learning Outcomes & Tasks

1. Identify alternative ways to organize information based on its inherent structure in a way that enables effective and efficient completion of tasks.
2. Discuss issues involved in the development of client-side and server-side solutions
3. Discuss the contrast between entry and validation techniques in client-side and server-side programming
4. Demonstrate the management of state and data using technologies such as databases and server-side sessions.
5. Discriminate between types of web servers including application servers, streaming media servers and transformation servers
6. Evaluate selected emerging and existing web technologies for web based solutions
7. Develop server-side programs
8. Implement a web site integrated with other IT applications

Task	Outcomes Assessed
Implement a SQLAlchemy model design in models.py	1,
Implement a Flask Application Server using the models specified	1, 4, 7
Implement a web application front-end in index.html and app.html	4, 8
Extend the model of an application for new functionality.	1
Specify a web application solution.	2, 5

2 Application Specification

A social media application whereby users can create posts and other users can view 'like' or 'dislike' posts. The following are the features of the application:

1. Authenticated users can create posts
2. Authenticated users can view posts by all other users of the application
3. Authenticated users can delete their own posts. When a post is deleted, all of its associated reactions (reacts) should also be deleted.
4. Authenticated users can react (like/dislike) to any post

Question 1 (a): Login Page Design

Using HTML, CSS, JavaScript, AJAX and/or Jinja, implement the user interface design given in Figure 1 in a file named index.html.

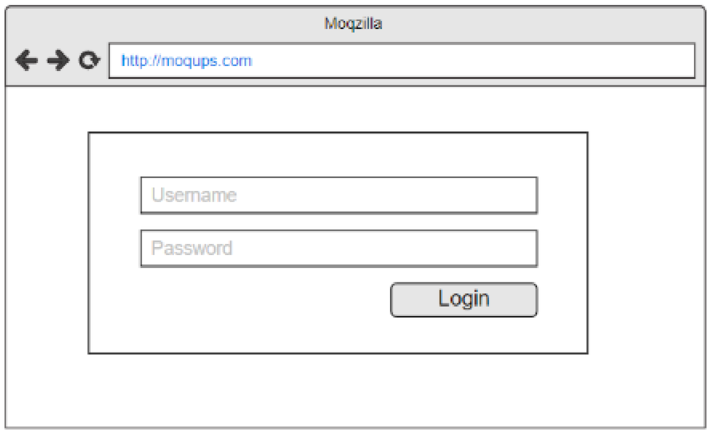

Figure 1: Login Page

Question 1 (b): Posts Page Design

Using HTML, CSS, JavaScript, AJAX and/or Jinja, implement the user interface design given in Figure 2 in a file named app.html. You may add additional form controls if necessary.

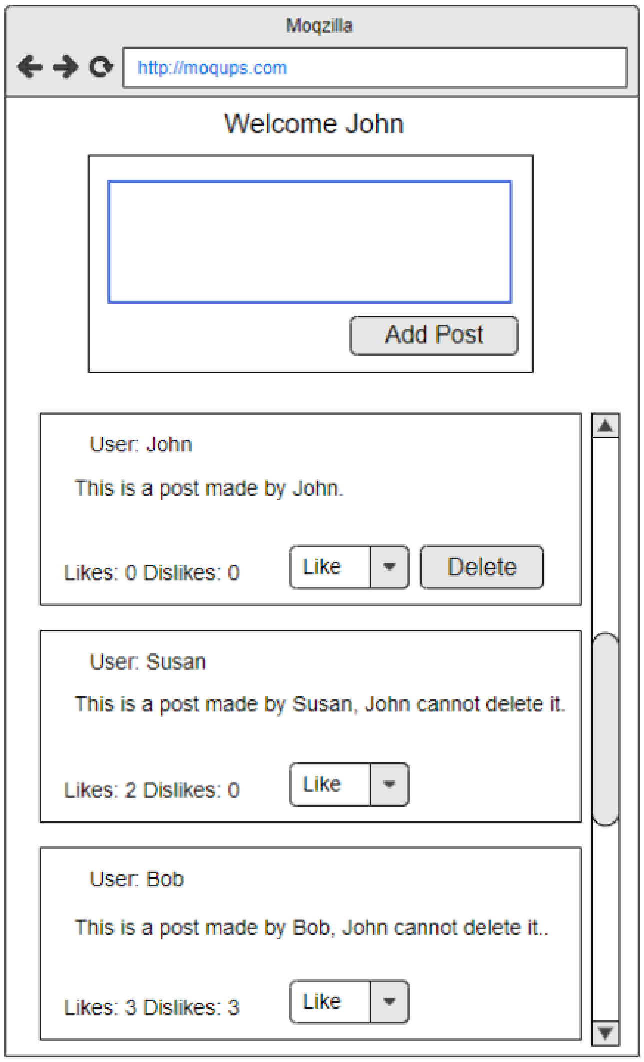

Figure 2: Posts Page

Question 2: Model Design

Implement the class diagram given in Figure 3 in a file called models.py. Create an initDB.py script which initializes the database and creates two users.

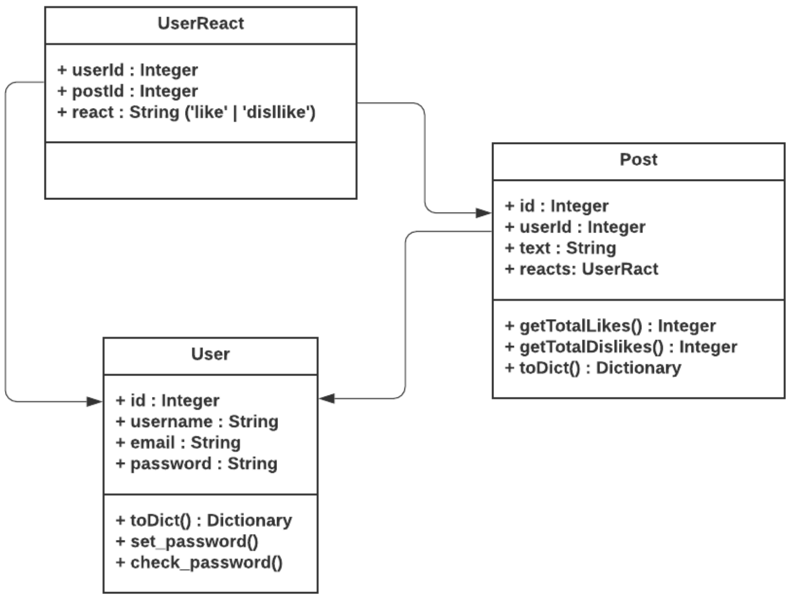

Figure 3: Model Design

Methods Description

- Post.getTotalLikes(): Queries UserReact and counts the total ‘like’ values of the react property for that post.
- Post.getTotalDislikes(): Queries UserReact and counts the total ‘dislike’ values of the react property for that post.
- Post.toDict(): Returns a JSON representation of the post including the username of the post author and the results of getTotalLikes() and getTotalDislikes() as “likes” and “dislikes” respectfully.

Question 3: Server Side Programming

You may implement any python flask routes necessary to facilitate the functionalities of the application. You may use the following as an example of posts data sent to the view when an authenticated user “Bob” (id 1) visits app.html.

```json
[
    {
        "id": 1,
        "userid": 1,
        "text": "Example post by Bob",
        "likes": 0,
        "dislikes": 1,
        "react": "dislike",
        "owner": true
    },
    {
        "id": 2,
        "userid": 2,
        "text": "Example post by Alice",
        "likes": 0,
        "dislikes": 4,
        "react": null,
        "owner": false
    },
    {
        "id": 3,
        "userid": 1,
        "text": "Another post by Bob",
        "likes": 4,
        "dislikes": 1,
        "react": "like",
        "owner": true
    }
]
```

Client Side Implementation Rubric

	Look and Feel	Front end Functionality
Advanced	Neat format and layout. Makes good use of 3rd party library components for messages. Page redirects after login. 5 marks	Data displayed from database. HTTP requests contain the required authorization headers. Forms are functional. Error handling. View refreshes when data changes on server. 15 marks
Proficient	Page redirects after login. Alerts user when appropriate. Resembles the design given. 4-3marks	Data displayed from database. HTTP requests contain the required authorization headers. Forms are functional. Users' previous reactions to posts are not shown. View only reloads when the page is refreshed. 14-11 marks
Approaching Proficient	Fairly resembles the design given. Styling and colour. Alerts user when appropriate. 2 marks	Data displayed from database. No error handling. No authorization headers in request. Likes/Dislikes not shown on posts. Authentication implemented. 10-8 marks
Beginning	Minimal styling. No error messaging. 1-0 marks	Data displayed from database. Forms somewhat functional. No authorization or authentication used. 7-0 marks

Server Side Implementation Rubric

	Model Implementation	Routes
Advanced	Model fully implemented according to the specification. 10 marks	Works according to the specification. Code is neatly formatted. Authorization and data restriction error handling done. 10 marks
Proficient	Slight deviation from the specification given. Implements foreign key relationships. Model methods implemented. 9-7 marks	Posts endpoint does not indicate the user’s reaction to the post. Authorization and data restriction done. 9-7 marks
Approaching Proficient	Covers most relationships. Model methods implemented. 6-5 marks	Not all features are supported. Posts data sent to view do not contain accurate counts of reactions for each post. 6-5 marks
Beginning	Missing model methods. Model fields incorrect. Code not functional. 4-0 marks	Only post model fields are sent to as post data in the view. Improper access restriction. Code not functional. 4-0 marks

3 Short Answer Questions

Question 4

You are required to add a “follow” feature to the application described in section 2. Users should only see posts from themselves and other users they follow. Explain what changes to the models and endpoints you would make in implementing this feature. You may use diagrams to aid your explanation. [10 Marks]

Question 5

You are required to build a Public Transport Mapper application whereby users can:

1. Search and view public transport routes.
2. Create and submit routes to be approved by a moderator.
3. Upload pictures of locations

Identify several technologies (frameworks, libraries, integrations, servers, software, tools, services) and explain their use in fully implementing and hosting the solution.

[10 marks]

END OF PAPER

TOTAL 60 MARKS