CS139: Web Application Development

Coursework Assignment

Introduction

You will be developing a brand-new web application as your coursework for this module. It may seem like a daunting task, but you'll be able to use ideas from the labs, and you will get credit not only for meeting the specification, but also for exceeding it.

Important Dates

Coursework Issued: Monday 14 February 2022 9am (week6)
Tabula Submission: **Monday 14 March 2022 12pm (week10)**

The Task

Next year you will be moving out of your luxurious on-campus accommodation into shared houses. This introduces the issue of shared responsibility for household bills like gas, electricity, and internet. Keeping track of who owes money to whom can be a difficult task without some sort of tool. For example:

- I live in a house with 4 other people
- I receive an electricity bill for £50.00
- I input the bill into the application
- Each housemate receives a notification to pay me £10.00

You will be developing The Splitinator! (Please feel free to come up with your own awesome application names). The basic functionality for this application is as follows:

- User registration
    - Minimum to include: name, email, password
- User authentication
    - Minimum to include; email, password (including secure password storage)
- Adding a bill
    - Minimum to include name, amount
- Splitting the bill between the housemates
- Settling payment between parties
- Displaying the status of bills
    - Minimum to include: pending/complete, balances
- Flash notification of a new bill, and monies owed

In addition to implementing the basic specification above, 10% of the coursework mark will be awarded for implementing additional features. The two most significant features will be selected to contribute to your mark.

Possible ideas for additional features:

- Uploaded image of the bill
- Person to person bill splitting (e.g. a dinner split between a subset of the housemates)
- Proportional bill splitting (e.g. 40%,20%,20%,20%)

You may have other interesting and useful ideas, which will also earn credit. Build the best application that you can.

Deliverables

You should provide all files in a `zip` file to be uploaded to Tabula by noon on Monday 14th March 2022. The `zip` file must not include the Python virtual environment so you must run `./clean.sh` prior to compressing and uploading your application. Make sure to have a backup first! Please decompress your submission to a different location and run `./setup.sh` before submitting to check that your `zip` file is what you really want to submit.

A complete application should:

- Implement the functionality of the basic specification above using Python, Flask and Flask-SQLAlchemy with sqlite3.
- Make use of JavaScript to enhance the user experience.
- Have a consistent styling.
- Consider usability and accessibility to disabled users.
- Consider security issues.
- Possibly implement additional features to attain the additional 10%.

Include in the `zip` file, a 5-minute demo video (.`mp4` format) that can be viewed to demonstrate your application to a marker. You should also include a `README` file describing the application's main features, with any additional features clearly marked and explained from the user's perspective.

The video must show:

1. Registering for a new account
2. Logging into that account
3. Failure to log into a false account
4. Adding and removing bills
5. Distributing bills
6. Notification of a new bill and monies owed (within the app)
7. Settling a payment
8. Demonstration of any extra features

Your marker will install and execute your Flask application from a secure server that will not have access to external services. It will run on a free port (e.g. 5000, 5001, etc.) so you must **not** hardcode port numbers. Your core application should work without the need to download files or query services outside your application. The basic installation framework has been used throughout the labs and can be found in the `/modules/cs139/coursework.zip` file and on Moodle.

Support

The tutors will be happy to provide some help on this coursework assignment during the lab sessions; however, that help won't be as substantial as for the lab assignments (so it makes sense to complete those). As things may get busy, you may wish to contact the tutors with any coursework questions earlier rather than later. All work in the final submission should be your own: sharing ideas is welcome; sharing code is **not**.

Allowable Frameworks

Here are some examples of libraries and frameworks that may be used for the **core requirements** of the project in addition to Python and Flask modules installed via `coursework.zip`.

- Google Fonts
- jQuery and jQuery UI
- Reset CSS

Include attribution / acknowledgments for these in a separate web page on your site, where necessary. The use of jQuery and AJAX is optional, but its use can contribute to your additional functionality.

If you make use of additional libraries (such as **Bootstrap**) to support your **additional features**, code provided by such libraries will not be graded, naturally. Only your own work will be graded.

Finally, you should ensure that your application works on your DCS linux account and that it will work from a clean install. That is, the marker will run the provided `./setup.sh` with the standard `requirements.txt`, which will install packages and run your application as its last step.

Marking

The marking criteria are as follows:

- 20% - Appropriate markup
- 25% - Design and usability
- 35% - Functionality and implementation quality
- 10% - Security
- 10% - Additional functionality (1%, 3% or 5% for small/medium/big feature at the marker's discretion; you may add more than one feature)

We are excited to see how you solve the problem and the solutions you produce!