Exercise 7: A First Flask Application

In this lab we will start building up the tools that will be required for building the backend of web applications. The backend needs to be able to respond to web requests, store data and dynamically build web pages and data structures. As such, there are several key technologies you will need to be familiar with.

- **The Flask web application framework**: This framework runs the web application and is written in Python3. Similar alternatives are Django (Python3), Ruby on Rails, and NodeJS.
- **A templating language**: This allows us to programmatically build HTML pages, by mixing Python in with HTML. We will be using Jinja, but there are many alternatives including; pug and erb.
- **A database management system**: We will use SQLite, which is good for prototyping but should not be used for production systems. Alternatives include MySQL and PostgreSQL for relational databases, and MongoDB or Neo4j for NoSQL databases
- **Tools for deploying the system**: We will use either PythonAnywhere or Heroku, which are docker container services that can host small applications. Some alternatives include Linux virtual machines through Digital Ocean or other web services, and various solutions through Amazon Web Services. You can also host your own server.

First things first....

By now you should have formed a project group and begun discussing your ideas for the project. Once you have registered your group, discuss your idea with the lab demonstrator. Try and identify the user stories involved with the project, some sample polls/ranking it may show, and sketch some interface designs.

The application is running on the development flask server, and using *SQLite* as a database, so is not ready for full deployment, but it serves as a basic template for the sort of things you will require in your project. You can find the source code for the project on GitHub. This includes the basic instructions for running the application via the README.md file. You can download a copy of the application from GitHub to run yourself. However, complete the **First Flask App** exercise below first.

First Flask App

Miguel Grinberg has produced an excellent set of tutorials for building Flask apps. Work through the first three chapters of their mega tutorial. It is important to get your Python and Flask environment set up correctly. The instructions in the mega tutorial are good but ask for help if you do not understand something, as this development environment will be especially important for the rest of semester.

Lauren Gee has prepared some detailed notes on setting up a Flask project within VSCode which are available on the LMS above the link for this exercise.

Running Static HTML Files

If you would like to run your web pages from early labs on your flask server, you can create a folder called static in the app directory and place the files there. When you navigate to localhost flask will

look in the static directory for resources, so you can put the webpage name at the end of the local host URL. You can then see your requests being sent to the flask webserver.

For example, if you have a static HTML file called calculator.html, and you have placed it under app/static/calculator.html, you should be able to access it via http://localhost:5000/static/calculator.html