Assignment 5

It is advisable to read over the whole assignment before making a start.

When submitting screenshots, please resize them to keep the total submission file size down to below 8 MB, the Hub's upload size limit (the smaller the better, as long as the screenshots are readable).

SQL databases are used to store and manage the data used by web apps. In this assignment, you will create a simple web app using the Flask server in Python, connect it to our course database using an ORM, and deploy it to the web using Heroku, a hosting platform for web apps.

You will also learn about Git, a version control tool used to store and backup source code; this tool is required to deploy code to Heroku.

You will need to refer to online documentation and possibly Stack Overflow to do research for this assignment.

Question 1

What is an ORM (object-relational mapper) and how are such tools useful in database programming?

Question 2

Give a brief overview of each of the following software tools. Your overview should mention functionality, applications, history, and the meaning of the tool's name.

- The Flask server (Python library)
- The SQLAlchemy ORM (Python library)
- The Heroku hosting platform
- The Git version control tool

Question 3

Install Flask using **pip**. If you get stuck, check some examples or try **conda** instead of **pip**.

Question 4

Create a folder called **server**. In this folder, save the following as a Python file called **app.py**:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

Then run this file using the **flask run** command on the command line. You will need to set an environment variable using set or export to get this to work.

The Flask server is now serving a web page. Visit this page in your browser and show a screenshot.

Question 5

With the Flask server still running, change ‘Hello, World!’ to something else. Refresh the web page. Is the new value shown?

Configure your Flask environment to automatically reload the code when you change it. (Hint: look up the **FLASK_DEBUG** environment variable). Describe how you did this and show a screenshot.

Question 6

To show web pages, Flask uses templates. A template is a file containing HTML code which will eventually be rendered as a web page.

Research how to add a template to your Flask app, then add a simple HTML page. Show your code as well as a screenshot of the page.

Question 7

Research the **git init** command and use it to create a git repository in your server folder.

Question 8

Research the **git add** command and use it to add the code files in **server** to your git repository.

Question 9

Research the **git commit** command and use it to make a commit (to save a snapshot of your source code). Hint: research the **-m** option, which makes it much easier to save commit messages.

Question 10

Sign up for a free Heroku account (no credit card is required) and install the Heroku toolbelt.

Question 11

With reference to Heroku documentation and tutorials, use Heroku to deploy your app.

You will need to use the **heroku login** and **heroku create** commands from the Heroku toolbelt; research these. If you can't run these commands, make sure the toolbelt is on your PATH. For **heroku create,** you should be in the same folder which contains app.py.

A Heroku deploy is done by adding a Heroku server as one of your git remotes (**heroku create** should do this for you). Then, to deploy, you perform a **git add**, a **git commit** and then finally a **git push** to send your code to Heroku, which will attempt to run it on a web server.

As soon as you have created an app with **heroku create**, and before continuing, open a second command line window and use the **heroku logs** command to monitor what Heroku is doing. Keep this window open and visible at all times.

Hints:

- Heroku requires that a requirements.txt file be present in the server folder to indicate that this is a Python app. This should also contain the Python libraries Heroku needs to install to run your app.
- When your app is running on Heroku, an environment variable is present to tell it which port it should listen on. Research how to get this variable and pass it to Flask.
- A file called Procfile is required to tell Heroku how to run your server. Research what Procfile to use with Flask.

If you get stuck, check the Heroku logs, research on Google, and look at the Heroku tutorials and documentation before asking the instructor for help.

Describe what you did, then visit your app's Heroku web address and show a screenshot.

Question 12

Drawing on the Flask sample code made available, add Flask-SQLAlchemy to your app. Add a database model for one of the database tables in **northwind** or **dvdrental** (not the movies table), then use a template to create a page listing all the rows in this table. Running the app locally, show a screenshot. Show also your code.

Question 13

Deploy the new version of your app to Heroku and show a screenshot. If you get stuck, ensure the new libraries are in your requirements.txt file and look at the Heroku logs.