Software Development Year 3 - Academic Year 2019/2020

Wed and Cloud Development – Assignment 1

Develop a web-based game which exercises all of the code and web development techniques covered to date. Deploy your locally running webapp to "the Cloud".

Assignment Description

You are to develop a web-based guessing game, which implements the following game behaviour:

- A new user signals the start of a new game by visiting the `/game` URL, at which point a timer starts.
- The system generates a random integer between 1 and 1000, which it keeps secret, then asks the player to guess the number.
- The player makes a guess, and the system responds with one of three messages:
  - Congratulations, you guessed correctly!
  - Sorry, your guess is too high.
  - Sorry, your guess it too low.
- If the player guessed correctly, the system stops the timer, then records the player's name, time, and the number of guess attempts in your system's high-scores table (which can also be viewed at the `/highscores` URL). The system tells the user where in the high-scores table they placed. In the event of a tie, the player with the quicker time places higher in the high-scores table.
- If the player's guess is too high or too low, the system asks for another guess, and increments the number of guess attempts so far (and all the while the clock is ticking).
- The player keeps going until they guess the correct number, finishing the game.

Assignment Specification

1. Use Python 3 as your programming language for this game – which is to run as a web application, built using Flask/Jinja2. Note that the system needs to support more than one player interacting with it at any point in time[1].

2. Once you have your game running locally, deploy it to PythonAnywhere using the account that you set-up in class.

3. Once on PythonAnywhere (and working), use the PythonAnywhere Dashboard to identify me (userid: **barryp**) as your Teacher. This allows me to check your work on PythonAnywhere without needing to know your password. Note that PythonAnywhere emails me when you do this (so you don't have to).

4. You are to ensure that all submitted code conforms to Black's formatting conventions (note: I'll be checking your code for this).

---

1 This can be simulated on a single computer by accessing the system from multiple browsers (note: multiple tabs within the same browser breaks really badly). Flask's **session** technology can help here.

5. You are to write and submit tests which exercise the functionality of your webapp. You also need to create the **htmlcov** test coverage folder, too, and include it with your submission (note: I'll be running **pytest .**, **pytest -v**, and **pytest –cov** on your submitted code).

6. Use GIT to manage your code.

7. Marks will be allocated for a functioning webapp which implements all the requried features (and which runs locally on my computer), successful deployment to the cloud, the quality of your tests, Black conformance, as well as evidence of your use of GIT (include a copy of your **git log** output in your submission).

8. The actual "look" of your webapp does not matter – it's the **functionality** which is important. I don't care if it is beautiful or ugly, only that it works[2].

9. E-mail your solution folder as a ZIP file (named per your login ID, i.e., c00123456.zip) to `paul.barry@itcarlow.ie` by the due date/time. Your folder has to include all your code, your test code, your **htmlcov** folder, but **not** your GIT repository.

10. This CA is worth 20% of your final mark. This is an individual assignment, and you are expected to work on this assignment on your own. You must declare if you collaborated with anyone else.

11. Due date: **Friday, November 8th 2019**. Due time: **5:00pm**.

Useful Websites

Jinja2 Docs: https://jinja.palletsprojects.com/en/2.10.x/ - which includes information on how you can add IF statements to your templates to control which chunks of HTML appear where and when needed.

Flask Docs: https://flask.palletsprojects.com/en/1.1.x/ - which includes information on everything related to Flask (as we've only really scratched the surface). The **session** stuff should be interesting.

2 I should probably put that on a T-shirt.