C3 – Assignment #1 – WordGame for the WWW

Probably the best game never invented...

Some Useful Background Information

WordGame is a word game that tests your skill at word creation, checks if you can spell, and assesses your performance under pressure (tick ... tick ... tick ...).

When contacted by any web browser, the WordGame web server selects a word of seven or more letters from a dictionary. This word is known as the **source word**. The source word is sent to the requesting web browser and displayed on screen. Note that upon sending the source word to the web browser, the WordGame web server records the current timestamp value.

The user of the web browser now has to think up seven three-or-more letter words made up from the letters contained within the source word, and they have to do this as quickly as possible. Each word is entered into the web browser, then submitted to the web server, which then records another timestamp value.

Upon receipt of the seven words, the web server checks to ensure that:

1. each word is made up from letters contained within the source word,
2. each word exists within the dictionary (i.e., it's a "real" word),
3. the words all have three letters or more,
4. there are no duplicates[1], and
5. none of the seven is the source word.

If the seven words meet the above criteria, the web server computes how long the process took using the two timestamp values, then sends a webpage to the browser which needs to include the amount of time taken. The webpage asks for the user's name to add to the *Top Scorers List*.

Upon receipt of the name from the browser, the web server adds the user's name and their time into the appropriate place within the *Top Scorers List* then sends the current "Top 10" entries from the *Top Scorers List* to the browser, where is then displayed for all to see.

And, of course, because this game is *so* cool, the user willingly accepts your kind offer to play again.

1 Note: Case is not an issue here. So, "PET" is the same as "Pet" - that is, they are NOT two different words.

The Interaction Description

Here's a high-level description of a typical "interaction" between a web browser and web server:

- *browser* -> send a request to web server for WordGame's opening webpage.
- *server* -> determine a new source-word, record the timestamp, then send the source-word to the browser as part of a webpage.
- *browser* -> display the received webpage and wait for the user to enter seven words.
- *browser* -> submit the source-word, together with the seven words to the server.
- *server* -> record another timestamp, then check the seven words against the source-word for **validity**.
- *server* -> send the "result" to the browser and ask for the user's name.
- *browser* -> submit the user's name to the server.
- *server* -> update the *Top Scorers List* and send the Top 10 list to the browser.

Points to note from this interaction:

1. If one or more of the seven words are invalid, the web server needs to tell the browser which words were wrong so that the browser can display them to the user[2].
2. If the user does not make the Top 10, they should be told where they were placed within the list (e.g., *"Nice try, Jerry: You were ranked 4396th. Better luck next time."*).

Assignment Specification & Notes

1. Use Python as your programming language on the web server.
2. Be sure to test your solution (and maybe even ask your friends to play your game).
3. E-mail your solution code to `paul.barry@itcarlow.ie` by the due date/time, and deploy your game to PythonAnywhere.
4. This is your first CA and is worth 15% of your final mark. While this CA is active, no new material will be presented in class.
5. Due date: **Tueday, November 15th 2016**. Due time: 5:00pm.

2 And the server needs to appropriately chastise the user of the browser in the most "reasonable" way possible. Use your imagination.