Advanced Programming: Coursework 3

For the final coursework of the term, we'll be combining several skills we've learnt.

The task is to make a website which **gives the weather forecast for a given district of Uganda**.

Use the web server you used in class (running python httpserver.py).

Task 1: Make a web form

Make a form using the demo.html file (we looked at in class) as an example. Call this one **weather.html**. This form should ask which district you want a weather forecast for, and the action of the form should go to the script you are about to write (called **weather.py**).

Task 2: Write a script

You need to write a cgi-bin file (maybe based on demo.py, that we also looked at in class) which takes the district the user wants as its input. To begin with, maybe just try printing this district out. Call this script **weather.py**.

Task 3a: Find the coordinates of the district: Access wikipedia

To do this I want you to access the **wikipedia page** for the district, using...

```
import urllib2;
website = urllib2.urlopen('http://en.wikipedia.org/wiki/Jinja_District');
content = website.read();
```

Try outputting the content, as an output of the script.
Obviously you'll need to change the url depending on which district the user has entered.

Task 3b: Find the coordinates of the district: Parse the content of the webpage

Use a **regular expression** (using the re library we used in class) to search the page for the coordinates. It will be something that looks like:
 0.500°N 33.200°E
or maybe:
<span class="geo">0.500; 33.200</span>
– remember to handle if the number of digits might change, and weather it's in the southern or northern hemisphere.

Task 4a: Query a web service to get a forecast for that location: Get the data

We'll be using **forecast.io** to get a weather forecast. This is a webservice which provides weather forecasts for given latitudes and longitudes. The url for Kampala is:

```
https://api.forecast.io/forecast/6c8485e693d2f834905eea450403ac9c/0.31,32.58
```

(try visiting this page in a browser, to see the data you get).
Where, the large number (6c8485e693d2f834905eea450403ac9c) is an API key.
The two small numbers are the coordinates of Kampala.
We want to replace these with the coordinates we got from wikipedia...

Task 4b: Query a web service to get a forecast for that location: Parse the data

The web service returns a JSON file. For example:

```
{"latitude":0.31,"longitude":32.58,"timezone":"Africa/Kampala","offset":3,"currently":{"time":1416245978,"summary":"Mostly Cloudy","icon":"partly-cloudy-night","precipIntensity":0.0026,"precipProbability":0.02,"precipType":"rain","temperature":64.09,"apparentTemperature":64.09,"dewPoint":59.86,"humidity":0.86,"windSpeed":0.14,"windBearing":38,"cloudCover":0.83,"pressure":1008.77,"ozone":250.03},"hourly":{"summary":"Light rain tonight and tomorrow mo... etc
```

We need to extract from this the current summary of the weather.

To do this we'll use the json library:

```
import json
weather = json.loads(content)
```

look in this weather variable and see if you can get find the 'summary' of the current weather.

Task 5: Output this in a message on the webpage.
I want the result to say something like:
“The weather in Jinja district is sunny”

Task 6: Make sure your program can handle errors or wrong inputs. Is it secure?

Task 7a: Store the weather results in a database
Finally, to make the program quicker, we should cache the result of the weather summary in a database. Make the database using sqlite3 called weather.db with one table called weather. This table should have two columns, the district and the summary. Insert the summary into the table for the given district, the code below might help. Also: **Use google**!

Example code to help:

```
import sqlite3 as lite
try:
    con = lite.connect('weather.db')
    cur = con.cursor()
    cur.execute('SELECT something FROM atable WHERE stuff=?',(variable,))
    data = cur.fetchone()
    if (data==None):
        print “Not found anything”
    cur.execute('INSERT INTO atable (something, blah) VALUES (?,?)',(variable1,variable2,))
        con.commit()
except lite.Error, e:
    print "Error %s:" % e.args[0]
    sys.exit(1)
if con:
    con.close()
```

Task 7b: Use these cached results to speed up the webpage
Add code before you get the coordinates to see if we know the summary of the weather for that district.

Task 8: Submission
Zip the weather.py, weather.html and weather.db files and email them to me at: msmith@cit.ac.ug
Deadline: **19th December, 2014.**