Cloud Computing, Semester 2, 2016

Assignment 2 - Mood Quiz Web App  
By James Snee, s3369721 and Sean Lee, s3301214

1. Summary

The Mood Quiz is a web app that is designed to find out what particular mood the user is in. It features 500 random questions to ensure the user will get a unique game every time (Amount of questions may change as we go along) Each question retrieved from the database will also have 4 possible answers, and in turn these answers will have 4 possible values associated with them. Each question will have up to 10 points available, so the more positively the user answers, the higher the score (in a 20 question game, a score of 200 signifies perfect mental health, more on this later.) When the user gets to the end of the quiz, their score will be presented to them.

1. Introduction

I - Motivations:

II - What does it do?: The Mood Quiz is a web app designed to find out what particular mood the user is in.

III - Why is it required?:

IV - How can it be used as a real life application?:

V - The advantages/positives/new features of our application

1. Software Design/Architecture

I have decided to implement this web app using Python 2.7 as well as the Flask webapp framework, and for hosting have chosen Amazon AWS and DynamoDB. For email functionality, I have used the smtplib module. The app consists of five HTML pages (index, question and results, sendemail and emailconfirmation) and various .py files for data upload to DynamoDB (using the Boto3 module) and the app controller script itself. Included in this package are also some test python scripts which shows my experimentation of getting connected to the DynamoDB, although these should not be run. The app also reads the included credentials file which includes the AWS ID and AWS Secret Key.

1. Related Works

Originally, I wanted to implement a web app version of my Android/iOS app, Epic Banter, which just pulls random strings out of a data store and display them. However, I found that implementing this would not pose much of a challenge to us as the app is quite a simple one. I opted to design a quiz type game with some additional features to keep track of, such as score as well as the pulling of random data out of a database, and a more advanced feature like email results to any email address, the implementation of which will be discussed in the next section.<Will try to incorporate and advanced feature like MapReduce in Python>

1. Implementation

Tutorial sheet

For installation, refer to the next section

How to push the questions.csv data into DynamoDB

How to deploy app to cloud

Deploying app to cloud

You will need an additional python framework - virtualenv. This is installed using Python pip: pip install virtualenv

Once it is installed, on Windows go to C:\Python27\Scripts\ and run virtualenv flask-aws to create a new Virtual Environment

Install all of the frameworks in the next step inside the Virtual Environment (pip install <name of framework>)

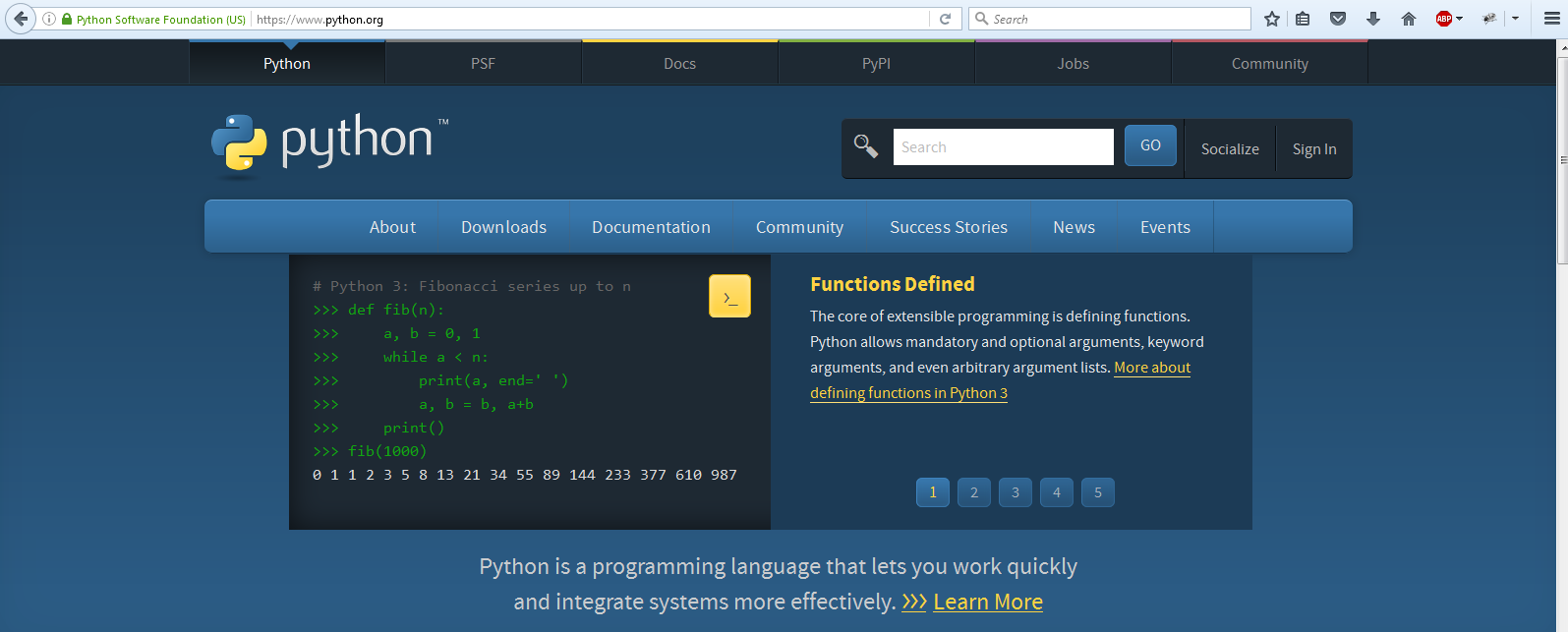
When the user goes to the URL of the app, it will start the app.py script. This is the main controller script of the app. Before anything is done, it will create a secret key for the session variables used, to ensure unique sessions. It will read the credentials file to ensure that we have the database connection.

Send results via email - I have included functionality to send the results score and results string via email. This is achieved using the smtplib library, and gets sent for a special gmail account I created just for this purpose. The app will login to this special account, and send the email from it. It was very easy to implement this feature, but powerful!

1. User Manual - How to Install

You will need to install a few things before you can run the app. Dependencies required are as follows:

* Python 2.7 (Can be downloaded from www.python.org)
* Python pip installer (used to easily install Python modules we require, look for the get-pip.py script on the Python website “https://pip.pypa.io/en/stable/installing/#installing-with-get-pip-py)”
* Boto/Boto3 module for Amazon AWS, DynamoDB and Python connectivity
* Flask webapp framework for Python



To install PIP, double click on the get-pip.py file downloaded to your PC and let the script run.

Make sure the Path to Python has been set on your local PC. To set it up :-

> Open up Command Prompt by right clicking on your windows button on the bottom left of your windows desktop, selecting run and typing cmd.

> To set up the Environment Variable. Input [setx -m SBT\_HOME [PATH TO SBT FILE]]

setx -m PYTHON\_HOME C:\Python27

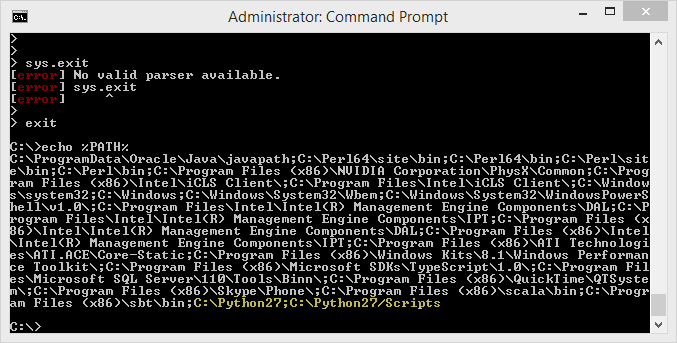
> To set up the Python Path. Input [setx -m PATH [PATH TO SBT BIN DIRECTORY]]

setx -m PATH %path%;C:\Python27\;C:\Python27\Scripts

Note: %path% appends the new path to the old one, if this is not entered your old paths will be replaced with the new one you have set.

> To check if the installation is successful. Enter on the cmd line:-

echo %PATH%



This or something similar is what should be displayed on the command prompt.

> Alternatively, you can perform this task manually by going to the properties option on your C:\ to open the System Window.

🡪 Selecting Advanced system settings from the menu to the left of the window.

🡪 Select the Environment Variables towards the bottom of that window.

🡪 Then, to set the PYTHON\_HOME environment variable. In the System Variables part of the window. Select new and enter

PYTHON\_HOME and [C:\Python27]. The path to the Python folder

🡪 Then, to set the Python Path. In the System Variables part of the window. Select the path Variable and enter

;C:\Python27\; C:\Python27\Scripts

[path to the Python Sripts directory]

Note: Do no forget the ; as it separates the mappings different path variables.

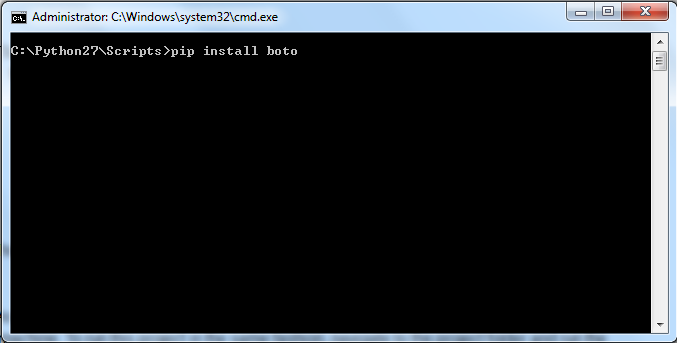
> Run the same commands in the cmd to check if the installation and path variables are set and successful.

To install these, once pip is installed, run the following commands from the Python/scripts folder (default on Windows is C:\Python27\Scripts) in your command prompt/terminal:

Pip install boto

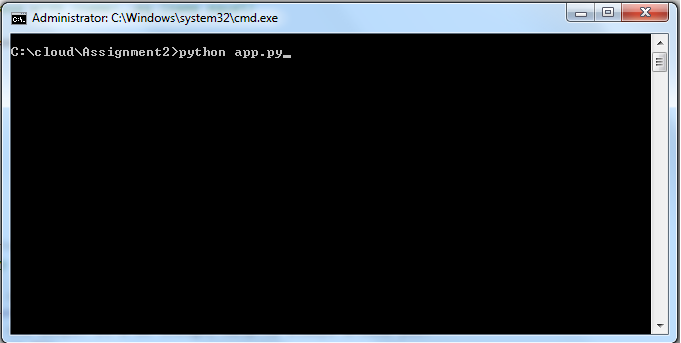
Pip install boto3

Pip install flask



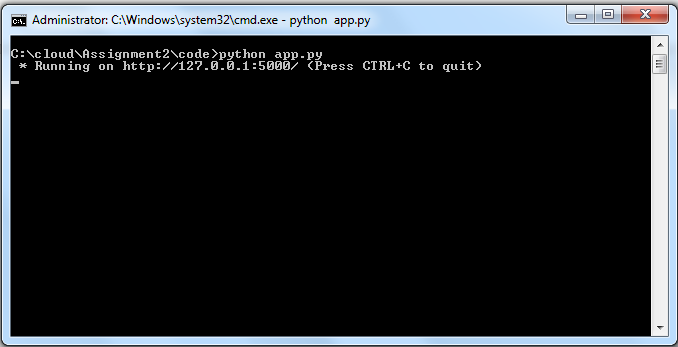
This will get the required files from the Pypi server. This is also useful because it will get the latest version of each module.

As we have seen from Implementation, I have developed the app using Localhost as a test machine. To run this project in the same fashion, navigate to the project folder and run the following command from the command prompt:



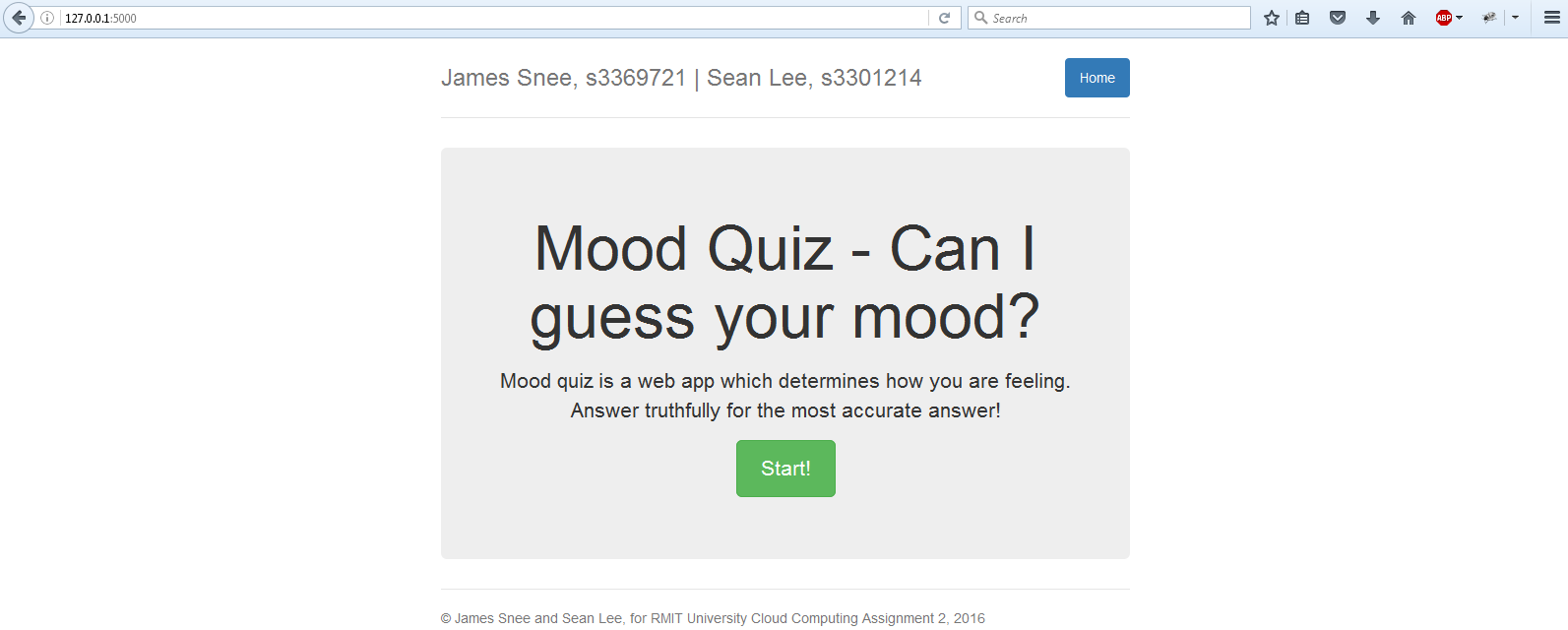
Python app.py

This will give you a message similar to the following:



\*running on <http://127.0.0.1:5000> (Press Ctrl-c to quit)

Open a web browser and go to the address above and you should see the following:



The app is now ready to run.

1. Contribution

James Snee - Compiling question data, GUI/HTML design, uploading question data to DynamoDB, code including DynamoDB connectivity (read/write), flask app script, deploying to Amazon AWS, Report

Sean Lee -

1. Links

* Flask how to: <https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972>
* Deploying Flask app to Amazon AWS and BeanStalk:

<https://medium.com/@rodkey/deploying-a-flask-application-on-aws-a72daba6bb80#.xdg0ksmkr>

-Deploying Flask App to Amazon AWS and BeanStalk (Official Documentation):

<http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-flask.html>  
- Deploy onto Amazon EC2:

http://www.datasciencebytes.com/bytes/2015/02/24/running-a-flask-app-on-aws-ec2/

* Send email from Python script:

<http://naelshiab.com/tutorial-send-email-python/>