Homework 4

Cloud9 Final Project

Brian Prost
Dr. Erroll Waithe
SDEV 400 7380
14 December 2021

Application Description

Homework4 is a Python menu driven application that allows you to:

- Learn about wine
    * A specific wine
    * or a random wine, if you don't know much about wine
- Lookup COVID statistics from the New York Times
    * not in realtime, data was imported on 10 Dec 2021
    * you can enter a state and receive the case and death counts for it
- input a url of a picture on the web and run a semi-accurate facial recognition
    * you'll get a link that hosts the image with a boxes around the faces it found
- read a random selection of a dad joke

Here's how to use the program

Initial Menu

Here is the first layer of the menu. Instead of typing in what you want to do, simply use the up and down buttons on your keyboard to select an option. Then press enter.

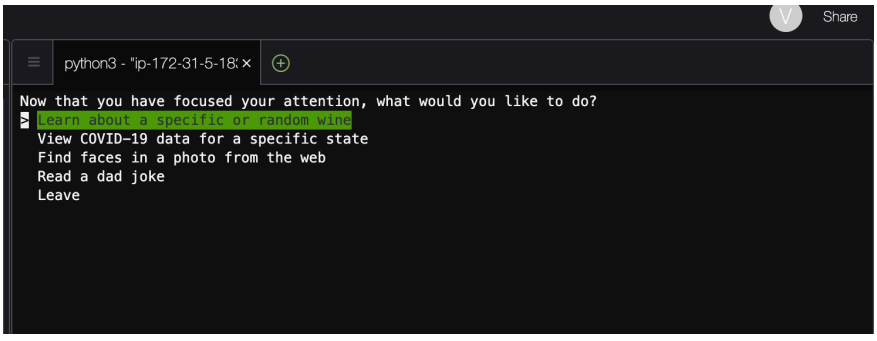

Fig. 1: Main menu

Amazon Wine Services

The next thing we're greeted with is two options: to learn about a specific wine or to choose a specific wine to learn about.

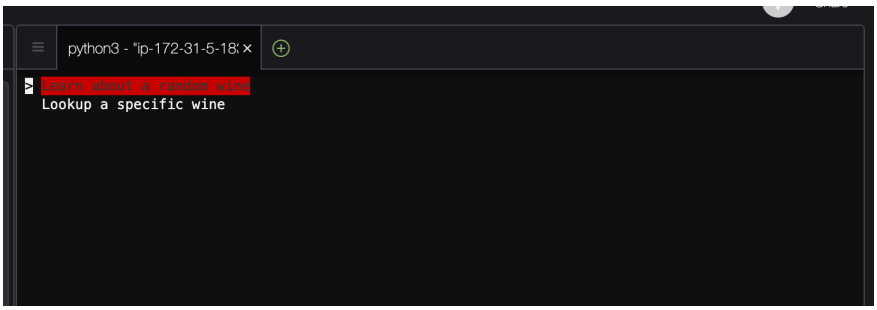

Fig. 2: Top level of wine menu

Specific Wine

The next screen asks if you want to learn about good wines (reds) or bad wines (whites). Choose wisely:

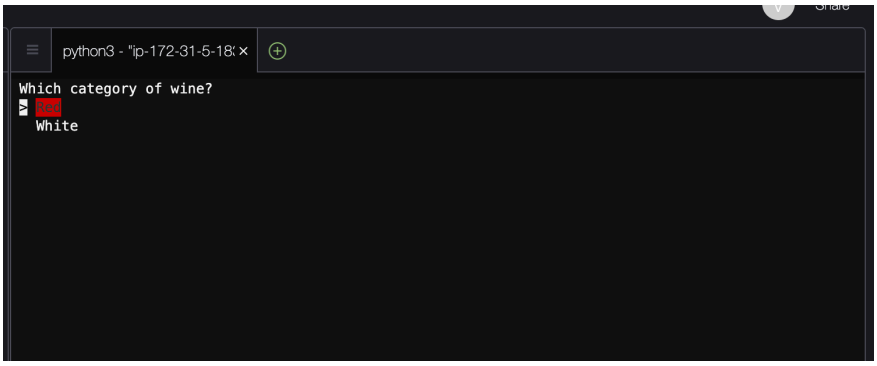

Fig. 3: Choosing a color of wine

After selecting the color, select the specific wine you'd like to learn about.

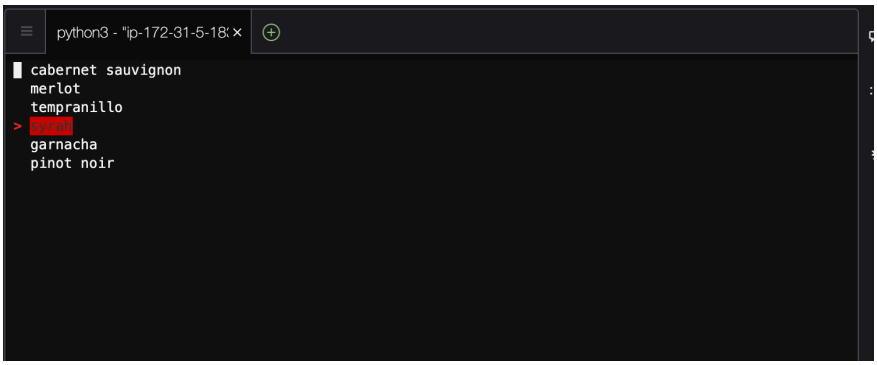

Fig. 4: Choosing a specific grape of wine

Here we selected 'Syrah.' We're then given a description of the grape and my rating, represented by stars.

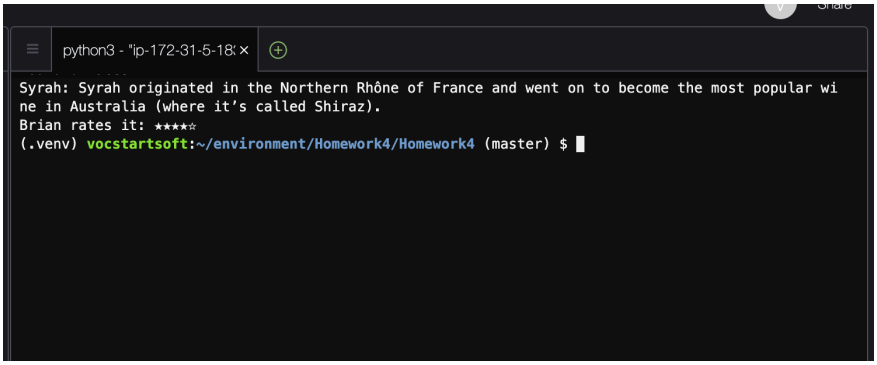

Fig. 6: a description and rating of the wine

Random Wine

If random wine is selected, then the user is brought right to the last step from before where the wine description and rating are printed:

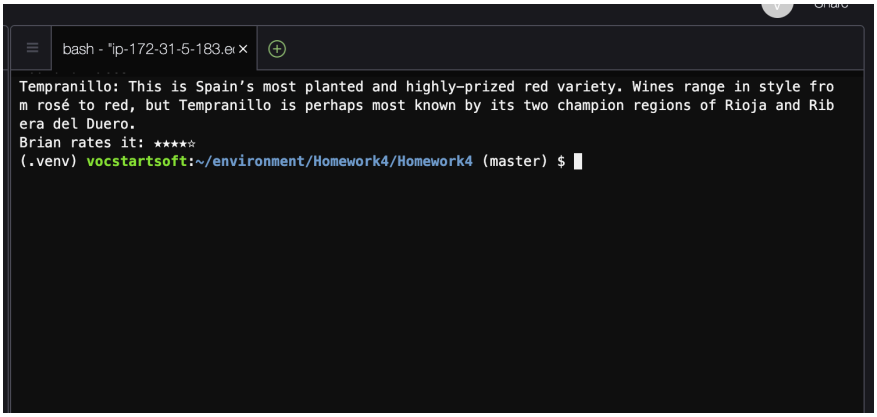

Fig. 7: Result of random wine option

About the implementation and integration with AWS

This service is achieved by loading a JSON file stored in S3 that is loaded locally, parsed, and selectively printed to the user.

"Hey Alexa, do I have COVID?"

If the option for COVID data is selected, then the next thing you'll see a prompt to type in the state you're looking for information about:

Fig. 8: entering and receiving a state's COVID cases & deaths

As you can tell, the application will properly handle any situations in which an invalid entry is entered. But when the correct one is entered, the corresponding data is displayed.

About the implementation and integration with AWS

This service uses an API call, AWS Lambda, and DynamoDB to achieve this solution. The state input is prompted and verified locally and then a Lambda function will access the data from a DynamoDB table, all triggered by an API call.

Facebox

If the option to recognize faces present in a photo is selected, the user will see a prompt (as seen below) and can paste or type in a direct link to an image on the web. Then, using OpenCV, a rudimentary facial scan will take place to attempt to identify any faces in the photo. If it finds any, it will upload it to an S3 bucket and give the user a URL that they can visit to download and/or share the photo. An example:

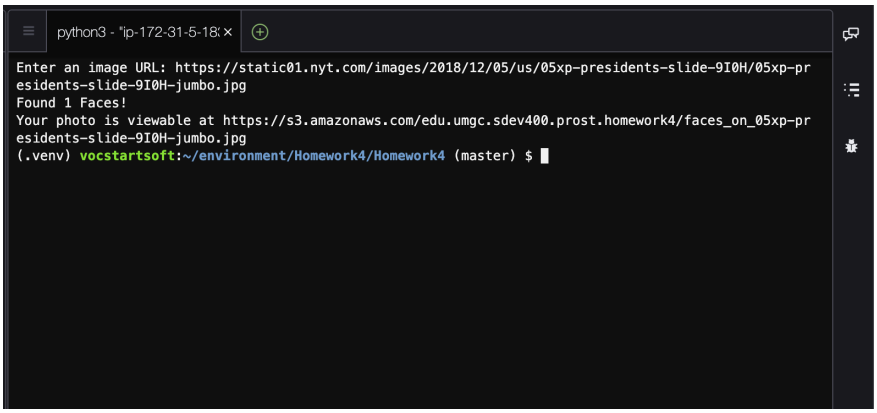

Fig. 9: Using the facial recognition feature of Homework4

As previously mentioned, the scan is very basic and has a low accuracy rate. This is of no concern because it's not pertinent to the assignment, but I would love to improve this over the holiday season. Here's the photo that was used in the example with the box that the program drew over it:

Fig. 10: The resulting image. Note the red square around George Bush Sr.'s face.

About the implementation and integration with AWS

This service uses S3 to save and access both the original photo and the edited one. The goal was to have this be a Lambda function, but using OpenCV with AWS Lambda proved difficult since one needs to install OpenCV not only as a Python package but also a system package.

Dad Joke

Last and certainly not least, the dad jokes. If the user selects this, the application prints out a dad joke that is saved in a database. No further action is required after selecting this option from the main menu.

Fig. 11: Result of selecting get a dad joke. Did you laugh? Cringe?

About the implementation and integration with AWS

If the dad_joke option is selected, an API call to a Lambda function is made. The Lambda function generates a random integer between 0 and the length of the rows in the DadJoke table. Then it uses the `boto3.get_item` function to get the joke from DynamoDB. It returns the joke as a string inside of the JSON response.

Saying Goodbye

When leaving the app, I made a stupid little animation to say goodbye and thanks for all the fish. It's of a guy throwing a table like, "Hey you, get out of here! Leave!" I don't know if the PDF will embed a video, so if it doesn't display, see fig_12.mov in the project folder.