**Dynamic Language Programming Assignment**

**For NortonLifeLock**

**By Nikhil Prakash Balbadri**

Table of Contents

[1. Details of the application. 3](#_Toc73087386)

[2. Structure of the application 3](#_Toc73087387)

[3. Data Information 4](#_Toc73087388)

[4. Instructions to run the application(Windows System). 4](#_Toc73087389)

[5. Testing the application 5](#_Toc73087390)

[6. Questions 5](#_Toc73087391)

# Details of the application.

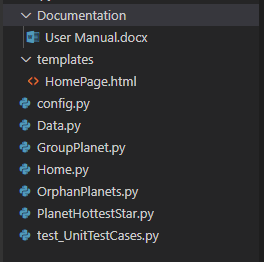
This is an application which downloads a catalogue of exoplanet data and displays the following information:

1. The number of orphan planets (no star).
2. The name (planet identifier) of the planet orbiting the hottest star.
3. A timeline of the number of planets discovered per year grouped by size. Size categories:
   1. “small” is less than 1 Jupiter radii.
   2. “medium” is less than 2 Jupiter radii.
   3. anything bigger is considered “large”.

For example, in 2004 we discovered 2 small planets, 5 medium planets, and 0 large planets.

Assumption made that the application will be run on windows system and instructions are pertaining to it.

# Structure of the application

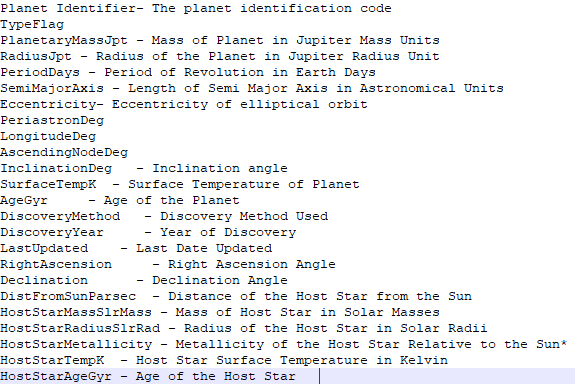


# Data Information

The application is built to handle data in json format. Data is set to load through a link and in JSON format.

For Developer - The link is to be set in config.py file.

Data fields explained below:



# Instructions to run the application(Windows System).

Pre-requisites to run the application.

* 1. Install Python Library and make sure the environment variables are set.

If the system already has Python installed and environment variables set, proceed to install further 2 more libraries to run the application.

Follow below steps for the same.

* 1. Open command prompt
  2. Install Flask library – (pip install flask)
  3. Install Pytest library – (pip install pytest)

Running the application

* Navigate to the directory of the application in command prompt.
* Run the Home.py file – (Home.py)
* The command prompt notifies you that the application is running with a link to access it on a browser.
* He application UI page is self-explanatory, which consists of 3 buttons which display’s the result for each of 3 questions mentioned in section 1.

# Testing the application

The application has unit test cases to validate each function. The file test\_file.py consists of the unit test cases.

To validate/run the test cases, follow the below steps.

* Navigate to the application’s directory in command prompt.
* Run command “python -m pytest”.
* It returns count of total test cases, count of test cases passed and count of test cases if failed with details for the same.

# Questions

* 1. **Why did I make the selected design choices?**

Selected Python as the programming language as it offers pandas which can deal with the logic implementation of queries asked effortlessly.

* 1. **Assumptions made.**

1. Application will be run on Windows System and instructions pertaining to it.
2. For Query 1 planet with no stars, data had 6 fields relating to its star. Hence a combination of these fields being empty together is considered an orphan planet with no star.
   1. **Why did I choose not to do few things?**

I chose not to create an application with C# or JavaScript because of the perks offered by Python pandas in dealing with the data format provided. Flask provides routing but I’ve used AJAX calls to query the data to avoid page reload.