

SI/ISM 224 - 2020

Project Brief

You are expected to produce a high quality *Jupyter notebook* to demonstrate your understanding of the programming concepts covered in this module. There are two sections to this assignment both must be completed for assessment.

Section A – 60 marks	Section B – 40 marks	Total – 100 marks.
----------------------	----------------------	--------------------

Section A: 60 Marks

You are required to create a program in Python of sufficient complexity covering a majority of the programming techniques from this module. Your notebook will be assessed for plagiarism.

Topic: *Movie Look Up Program*

The assignment is to design a Python program, implemented in a Jupyter Notebook, to enable a user to look up and receive information about movies from the *Open Movie Database* [<http://www.omdbapi.com>]. The OMDb API is a RESTful web service to obtain movie information.

The project requires that you investigate and understand how to interact with the OMDb RESTful API. In your notebook you need to explain what a RESTful API is and how interaction takes place between the program and this service.

The program you create should present a user with options and instructions for how interaction should take place. At a minimum, the user should be able to provide a Movie title to your program. It should determine if this movie exists or not and, if it does not, respond accordingly. If a legitimate movie has been provided, at a minimum, the program should provide the user with information about this movie (i.e, name, genre, release date, runtime, director, awards, etc.). In addition to this, the program should write this information to a .txt file in a manner that can later be read in when required. When the program starts, it should display the titles of the movies previously looked up and give the user the option to either look up a new movie or view the information for one of the previously looked up movies from the information stored in the textfile.

In addition to these basic features, you are expected to write a comprehensive program that is robust against potential errors encountered and makes appropriate use of the code structures covered in this module. You need to be able handle unexpected inputs from the user and the service you are calling for instance. How you implement this functionality is up to you. Creativity and outside-of-the-box thinking is rewarded. In addition to employing the concepts covered in class, you need to demonstrate your mastery of the content through not only producing the minimum applications specified, but also through incorporating extra features and components to your program. Extra marks are awarded for use of Object-Oriented design features.

General advice:

The OMDb API is a RESTful web service to obtain movie information that can be accessed at: <http://www.omdbapi.com>

Before working with this API and designing your program, you need to spend some time researching, firstly, what an API is, secondly, how to interact with an API through Python and, thirdly, how to use this API and handle the responses that it provides.

On the page linked above, once you are ready to begin, click on the API key button in the menu bar. Select the free option and enter the required details. You should then receive a verification email within a minute or two. This email will contain:

- *Your API key (which will be needed to make any API requests)*
- *An example of a request with your key*
- *A link to activate your key (you will need to do this first before you can use your key)*

PyPi provides a useful wrapper for accessing the API:

<https://pypi.org/project/omdb/>

It is strongly recommended that you browse this page and understand how to interact with the service. This is all explained at the link, along with examples of how to interact from python to the service. In your program, consider how dictionaries will enable you to work with the data returned

To install a library from PyPi using pip in a Jupyter Notebook run the following code in a cell:

```
import sys
!{sys.executable} -m pip install --upgrade pip
!{sys.executable} -m pip install omdb
```

Section B: 40 Marks

The second section of this project requires you to document your program. This should be conducted within your Jupyter Notebook using *Markdown*. In essence, your Jupyter notebook should read as a complete project, introducing your user to your program, demonstrating how it works, and documenting your key design decisions. How you structure your notebook is up to you. However, it should be clear what code is your final code for Section A, and which code is demonstrating particular features in your documentation.

You are expected to document key aspects of the logic of your solutions. This involves explaining particular sections of code in the program and why things were done in a particular way and how the program works together. Additionally, you need to include a section where you provide an overview of what an API is and inform the reader about key details to do with the OMDBAPI and how it is interacted with.

Your documentation also needs to include a retrospective section, where you reflect on problems encountered and how you went about solving those problems. In addition to this, such a retrospective should reflect on what improvements could be made to your program and what you learned from the experience. Finally, you need to include help documentation

explaining how your program should be used.

Assessment Criteria:

Section A - Implementation:

Your program will be assessed based on the following criteria:

- The program must work. i.e., It must run without crashing or throwing errors, even when unexpected input might be encountered. In this regard, your program will be tested to determine how it handles various input situations.
- Your program must be sufficiently complex.
- A majority of code structures covered in this module should be included in the program.
- Sufficient and appropriate commenting is important.
- It must connect to the API using a Python Library
- All of the required features will be assessed.
- Extra features included over and above the minimum requirements are awarded marks.

Section B - Design documentation:

Your documentation will be assessed based on the following criteria:

- The layout and design must be neat and professional.
- The program description must be complete and thorough.
- The explanation of the code logic must be in-depth and accurate.
- The explanation of the API must be accurate and detailed.
- The retrospective should go into detail about problems and solutions you encountered.
- The help document must cover all possible use and error cases.

Submission Information:

For this project you must upload a .zip file containing a complete Jupyter Notebook saved in the .ipynb format along with any txt files or style files you use. This means that you should not upload a .html file or a PDF file.

All work submitted will be assessed for plagiarism via Turnitin. Submitting incorrectly, incompletely, or after the submission deadline may cause your project assignment to be deemed incomplete, resulting in a grade of zero. No e-mail submissions will be accepted.

Project due date: 25 June 2020: 17h00

A link will be provided on Sunlearn for you to submit your assignments.

This project will require a considerable amount of effort on your part and it is expected that, to complete the assignment, you will need to research and investigate the necessary tools, techniques, and services to be able to construct a solution. Your first port of call in any situations should be the documentation for python, the API, and then stackoverflow.