

Assignment Run Book

Pre-requisites and Installations

The document is prepared with respect to the RedHat Data Engineer Assignment.

Git Hub Link: [ANKITBISANI/Comics-Assignment: Comics Assignment \(github.com\)](https://github.com/ANKITBISANI/Comics-Assignment)

(**Download the branch on to desktop)

Tools/Technologies:

- Anaconda Jupyter Notebook (Python 3.7)
- MySQL (installation with root user on the desktop for localhost)
- Python IDLE (3.7) (as an alternate)

Modules used and installation:

- Requests
- Random
- Json
- PyMySQL
- Sys

If any of the module gives the error “No Module found error: \$module name “after importing “import \$module name”, please install on windows:

Py -m pip install \$module name

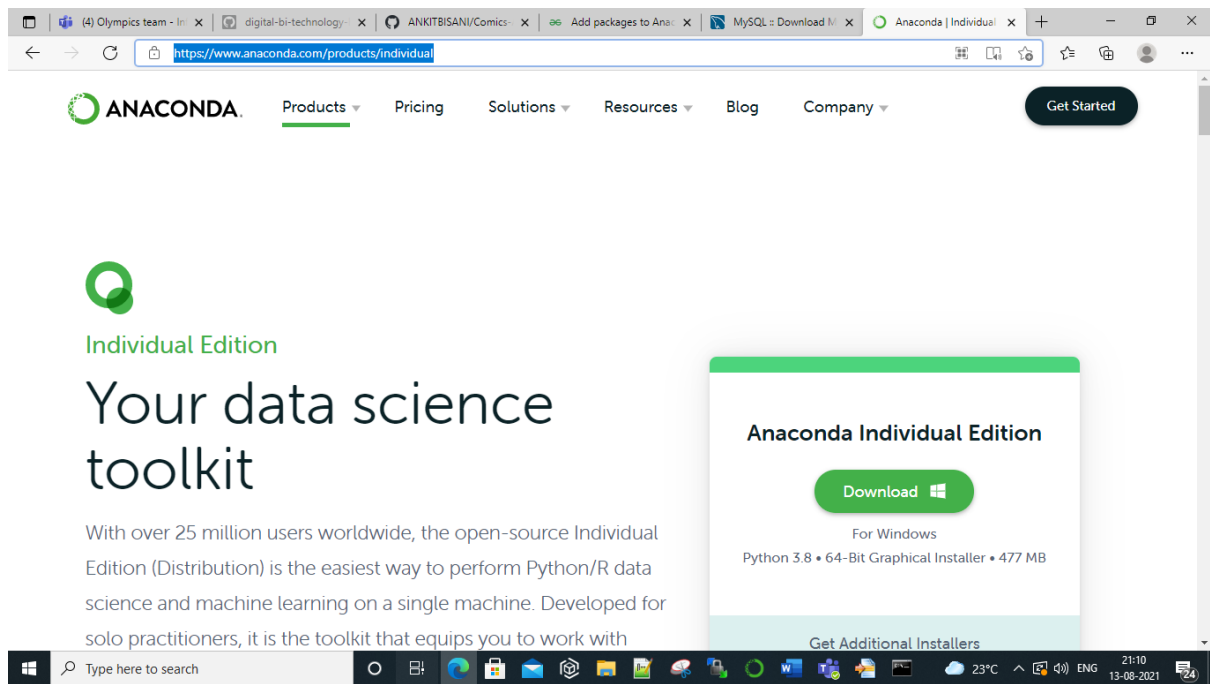
If using Anaconda(recommended):

(Please note MySQL module is not defaulted in Anaconda Jupyter and can be downloaded by below method. Other Modules mentioned are by default)**

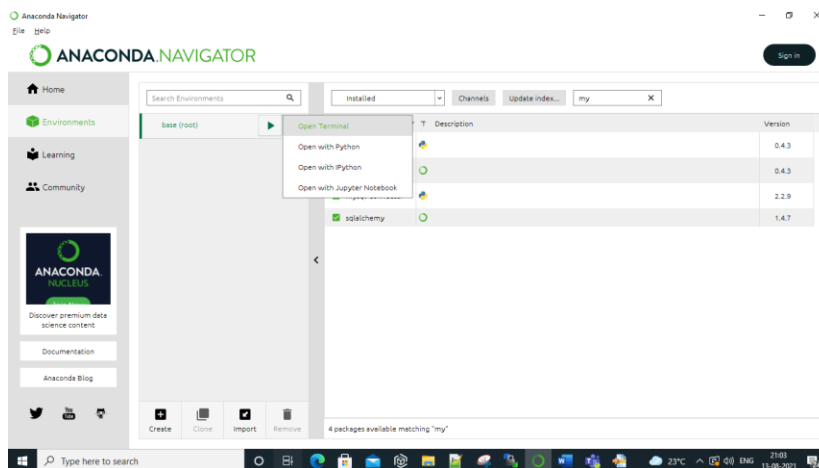
If you are using Anaconda Jupyter Notebook, follow below steps:

Steps 1: Download Anaconda individual addition

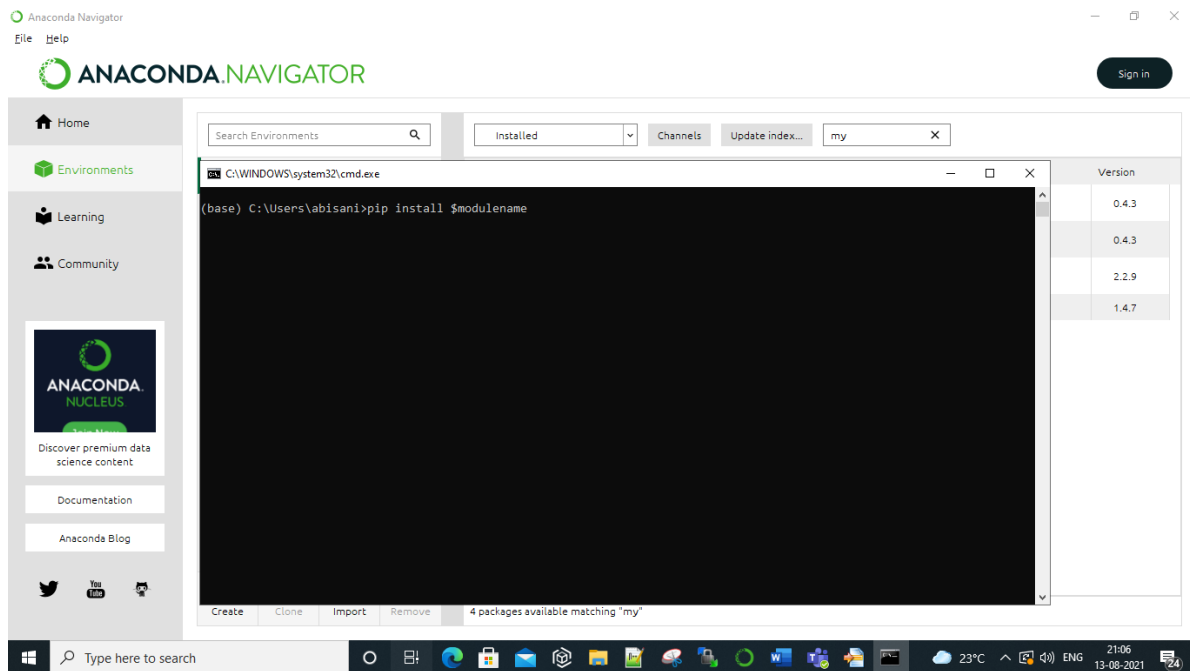
[Anaconda | Individual Edition](#)



Step 2: Go to Environments



Step 3: Install the modules as in screenshot



MySQL Installations:

Please Download: [MySQL :: Download MySQL Installer](#) from the link and follow the installation instructions

Open the MySQL workbench with the root password

****Please remember the root password as that would be used in scripts as per assignments**

Solutions (1 &2)

1. GET 15 random comics and following details in using Python. a. Get the names of the comic b. Get the alt-text of the comic c. Get the number of the comic d. Get the link of the comic e. Get the image of the comics f. Get the image Link of the comics
2. Insert into MySQL - Please include SQL database schema(s) for any table(s) created in your GitHub repo

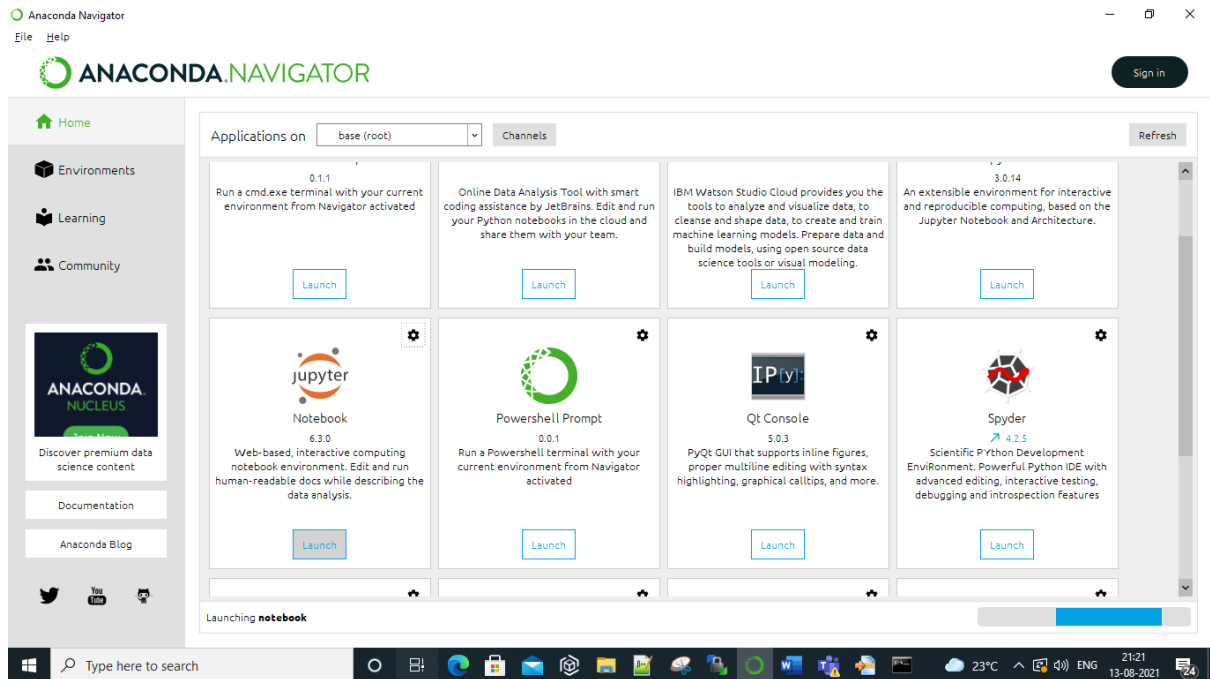
Step 1: create schema and user from root password on MySQL workbench

Create schema comics;

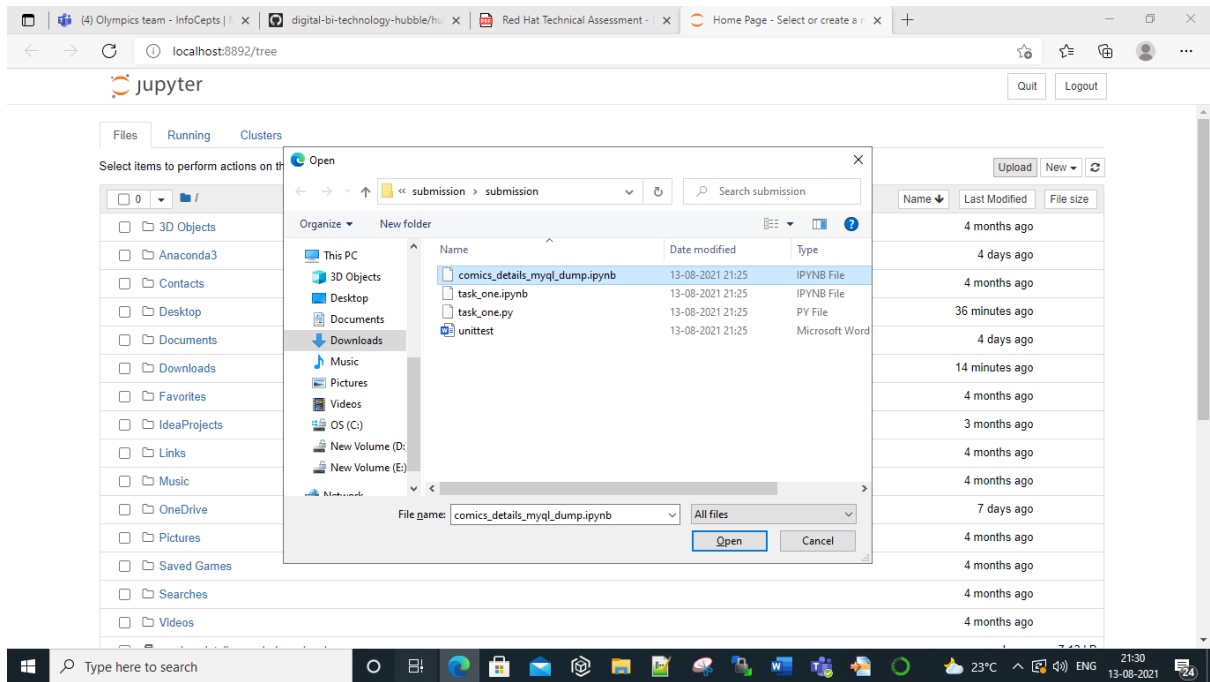
```
CREATE TABLE `comics_details` (
  `names` varchar(500) DEFAULT NULL,
  `alt_text` varchar(500) DEFAULT NULL,
  `number` int DEFAULT NULL,
```

```
`link` varchar(500) DEFAULT NULL,  
`image` varchar(500) DEFAULT NULL,  
`image_link` varchar(500) DEFAULT NULL  
);
```

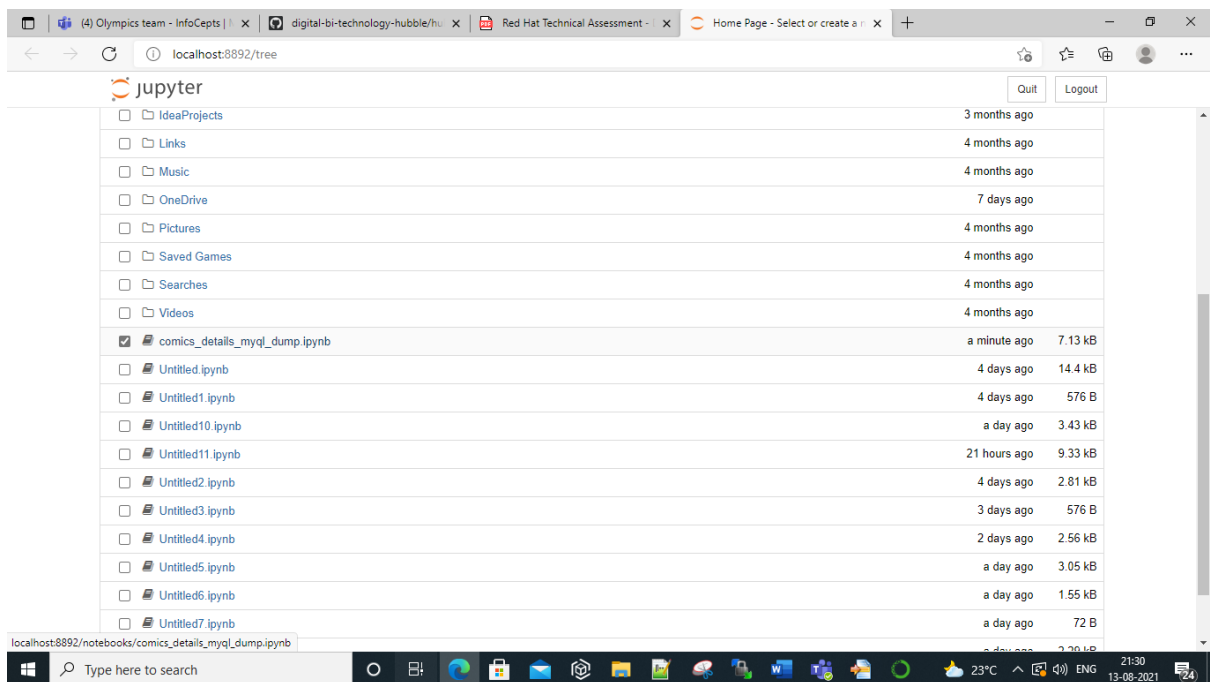
Step 2 : Launch Jupyter Notebook



Step 3 : Upload the downloaded file : comics_details_mysql_dump.ipynb

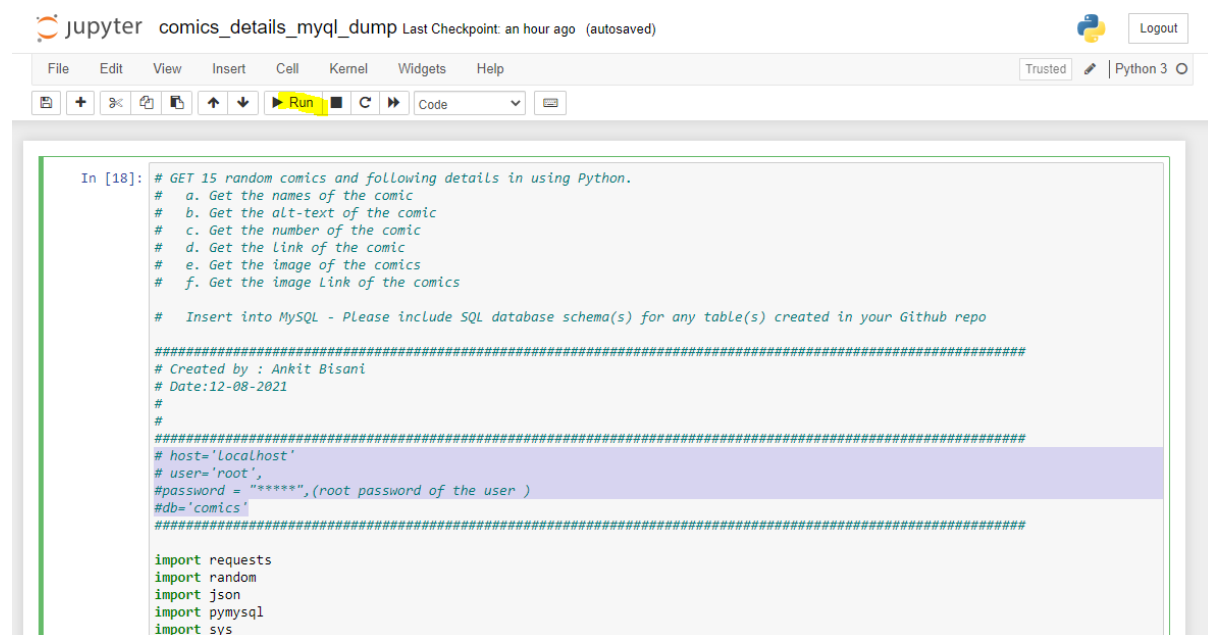


Step 4: Open the file: `comics_details_myql_dump.ipynb`



Step 5 : Replace the password and replace with localhost root password in the file and run the file

```
conn = pymysql.connect(  
    host='localhost',  
    user='root',  
    password = "March@1988", ##Need to replace  
    db='comics',  
)
```



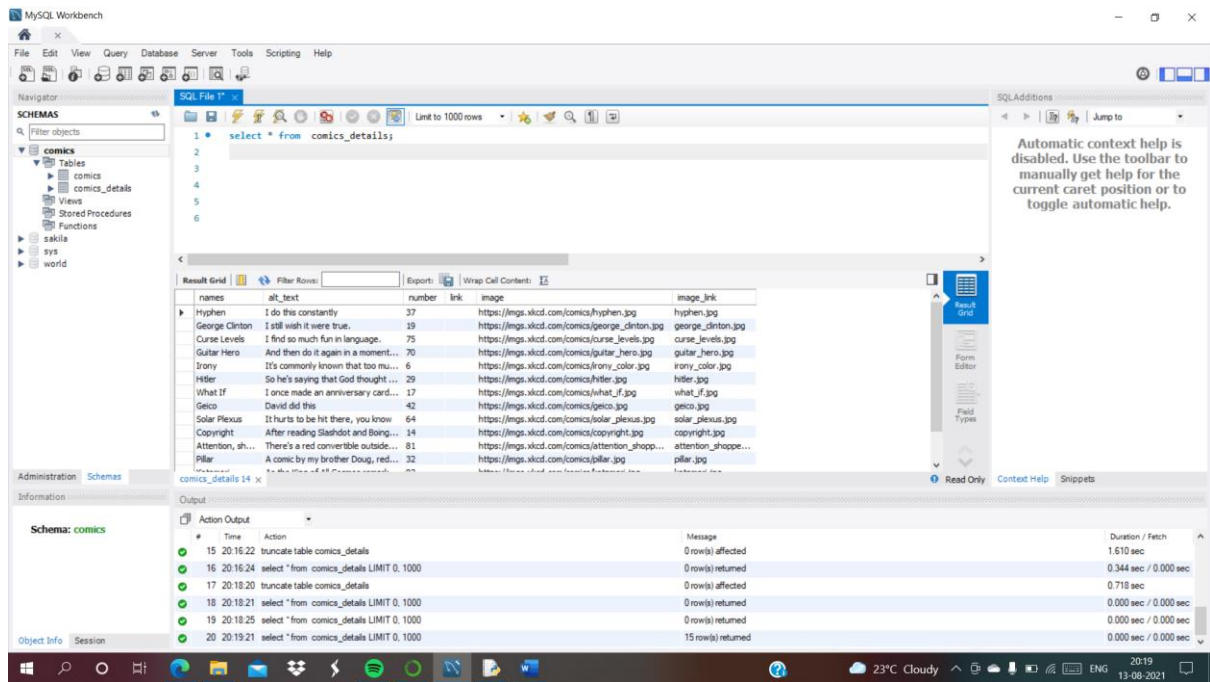
The screenshot shows a Jupyter Notebook titled 'comics_details_mysql_dump' with a last checkpoint 'an hour ago (autosaved)'. The interface includes a top bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help' menus. Below the menu bar is a toolbar with icons for file operations, a 'Run' button (highlighted in yellow), and a 'Code' dropdown. The notebook content area displays a Python script in a cell, labeled 'In [18]:'. The script is a Jupyter-style magic command (%%) that fetches 15 random comic details from a MySQL database. It includes comments for each step: getting names, alt-text, number, link, image, and image link. It also includes a section for inserting into MySQL, with a note to include the SQL database schema(s) for any table(s) created in the Github repo. The script is created by Ankit Bisani on 12-08-2021. The database connection details are: host='localhost', user='root', password = '*****', (root password of the user), db='comics'. The script imports requests, random, json, pymysql, and sys.

```
In [18]: %%GET 15 random comics and following details in using Python.  
# a. Get the names of the comic  
# b. Get the alt-text of the comic  
# c. Get the number of the comic  
# d. Get the Link of the comic  
# e. Get the image of the comics  
# f. Get the image Link of the comics  
  
# Insert into MySQL - Please include SQL database schema(s) for any table(s) created in your Github repo  
  
#####  
# Created by : Ankit Bisani  
# Date:12-08-2021  
#  
#####  
# host='localhost'  
# user='root',  
#password = "*****", (root password of the user )  
#db='comics'  
#####  
  
import requests  
import random  
import json  
import pymysql  
import sys
```

Step 6:

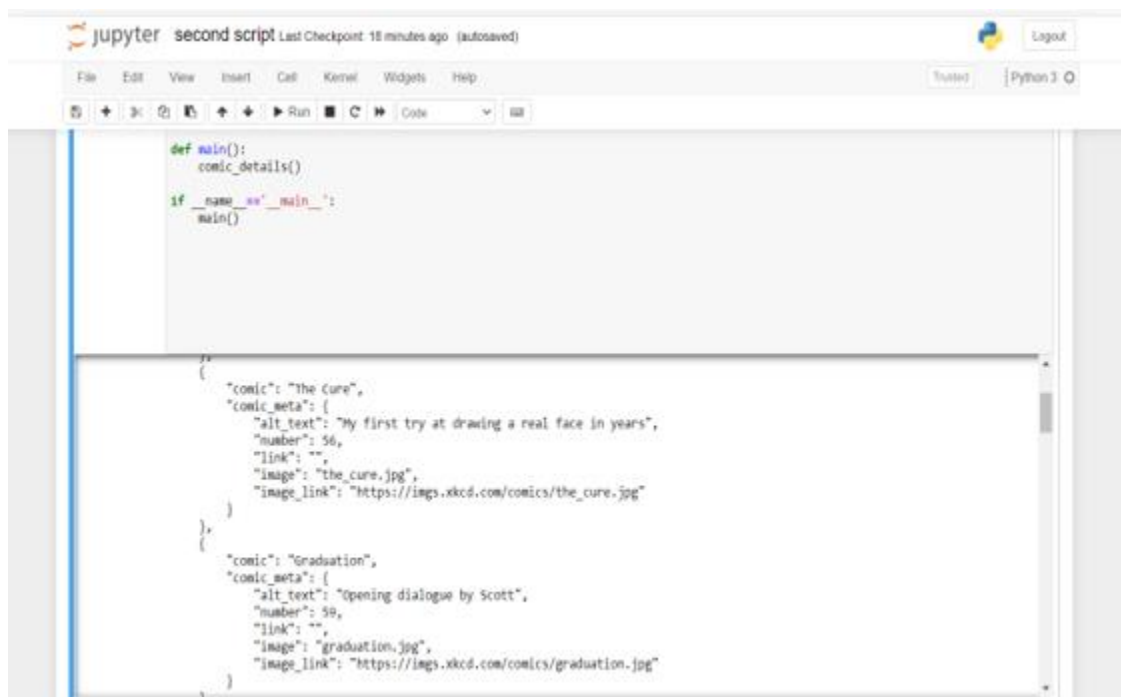
After successful run , go to MySQLworkbench and login with root creds and check the table

select * from comics .comics_details :

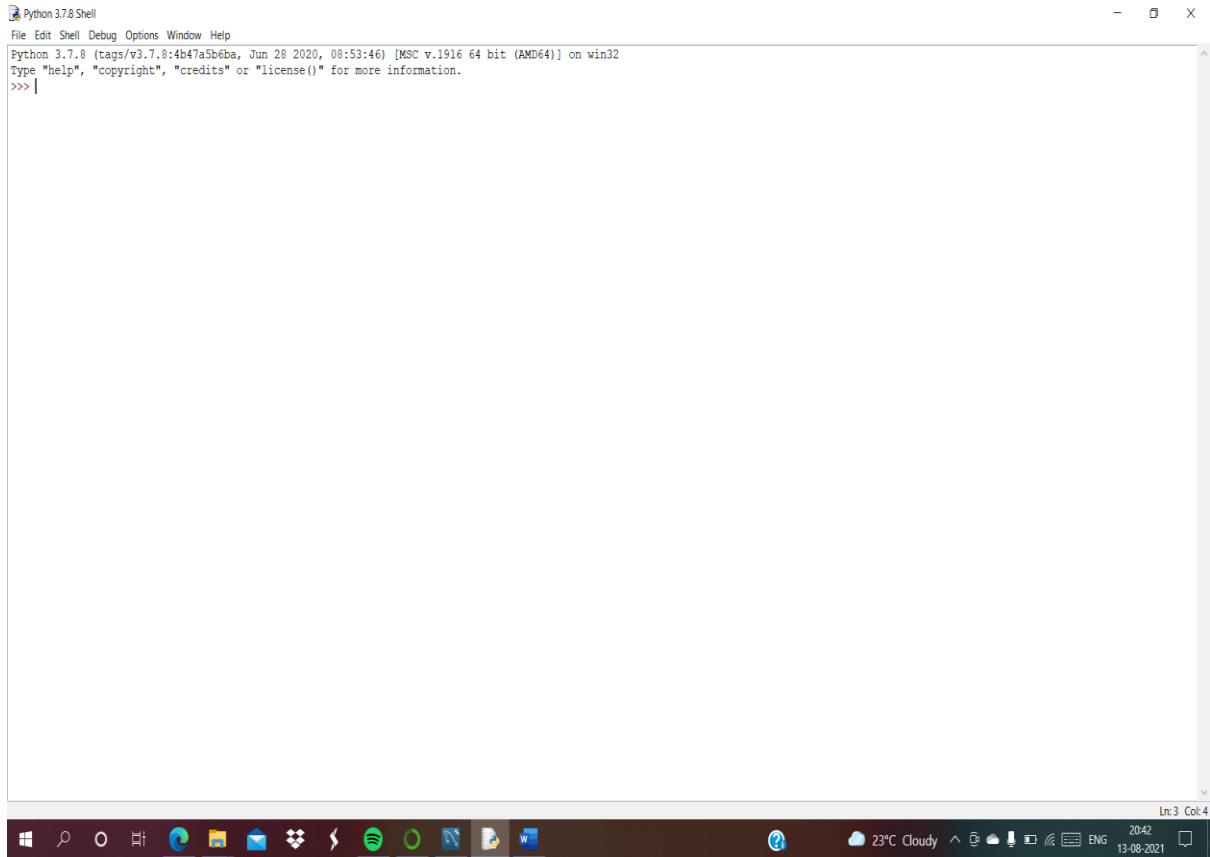


3. Write a script called `task_one.py` that when called will output something like this to the console.
(reference below)

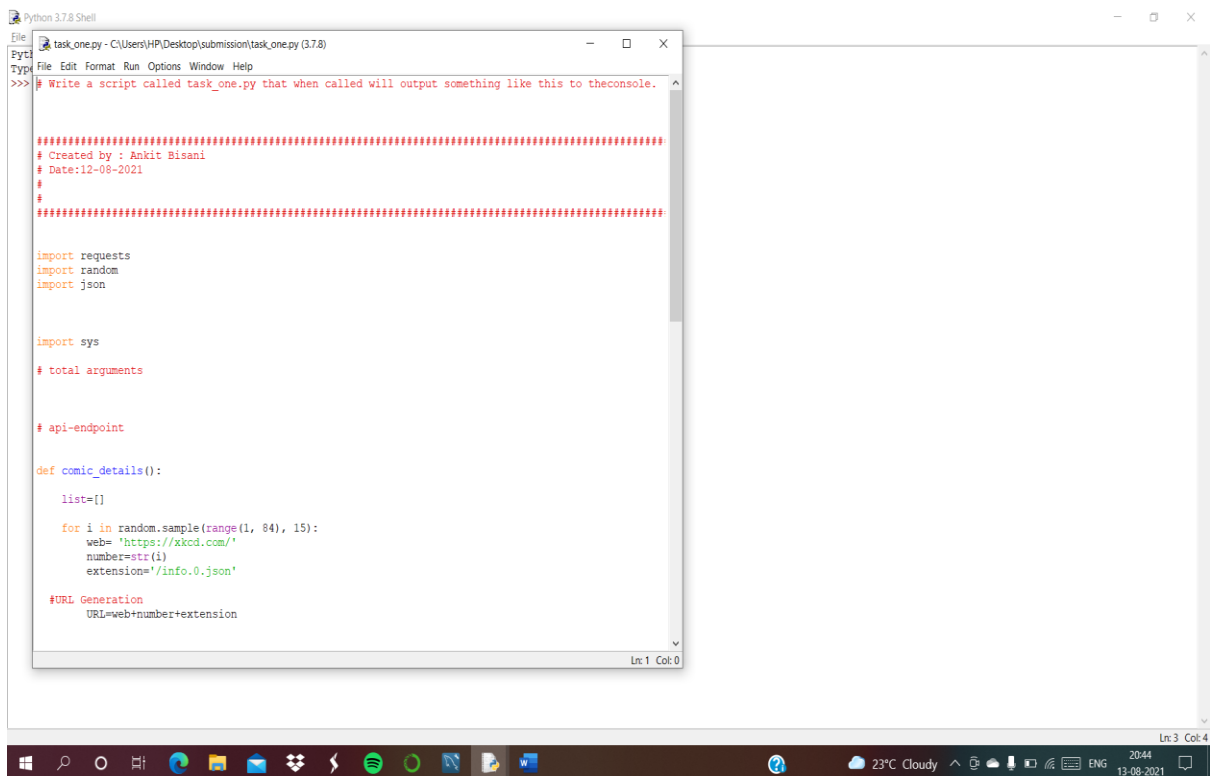
Repeat the above steps 1-5 for `task_one.ipynb` and observe the results:



Alternatively, you can open task_one.py in Python IDLE (3.7) and run the scripts



A screenshot of a Python 3.7.8 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the Python version and build information: 'Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32'. Below this, it says 'Type "help", "copyright", "credits" or "license()" for more information.' and the prompt '>>>' is visible.



A screenshot of a Python 3.7.8 Shell window with a task_one.py script. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The main text area shows the script content: '# Write a script called task_one.py that when called will output something like this to the console.' followed by a red separator line, '# Created by : Ankit Bisani', '# Date:12-08-2021', another red separator line, 'import requests', 'import random', 'import json', 'import sys', '# total arguments', '# api-endpoint', 'def comic_details():', 'list=[]', 'for i in random.sample(range(1, 84), 15):', 'web= \'https://xkcd.com/\'', 'number=str(i)', 'extension= \'/info.0.json\'', '#URL Generation', 'URL=web+number+extension'. The status bar at the bottom shows 'Ln:1 Col:0'.

