TOC Report

The HangMan Game

Submitted to:
Sunakshi Mehra Ma'am
Delhi Technological University

Submitted by:
Anukrity Varshney 2K18/IT/030
Bhuvesh Gautam 2K18/IT/037

Introduction:

This project uses Data science, string matching algorithms(TOC) and database management that makes a hangman game,

In this game we have to guess the movie with all blank spaces, there is input as a movie string with provided domain genre.

Summary to the project:

GUI asking movie genre

Takes genre as input

Search IMBD for movies of that genre

Movie list from that genre is imported in csv file

A random movie name is selected

Hangman Game GUI opens

Displays your chances

Displays the blanks and the keys to fill the word

IMPLEMENTATION DETAILS:

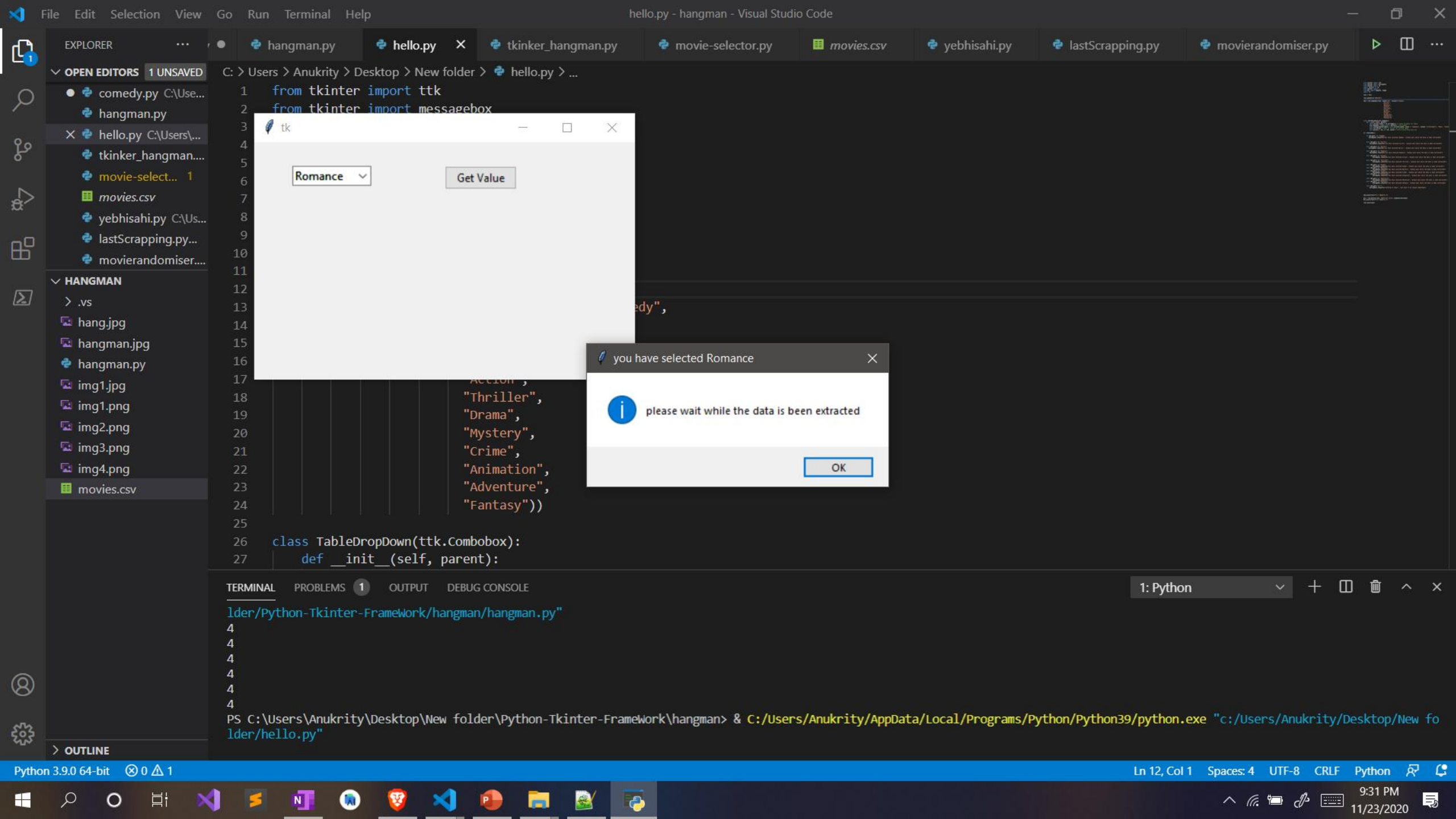
Programming Language	Python
Operating System	Windows
Library Packages	Selenium, BeautifulSoup, Pandas
Interface design	Python Tkinter GUI

Libraries used for web scrapping:

- **Selenium**: Selenium is a web testing library. It is used to automate browser activities.

- **BeautifulSoup**: Beautiful Soup is a Python package for parsing HTML and XML documents. It creates parse trees that is helpful to extract the data easily.

- **Pandas**: Pandas is a library used for data manipulation and analysis. It is used to extract the data and store it in the desired format.


File Edit Selection View Go Run Terminal Help
hello.py - hangman - Visual Studio Code
EXPLORER
hangman.py
hello.py
tkinker_hangman.py
movie-selector.py
movies.csv
yebhisahi.py
lastScrapping.py
movierandomiser.py
OPEN EDITORS 1 UNSAVED
comedy.py C:\Use...
hangman.py
hello.py C:\Users\...
tkinker_hangman....
movie-select... 1
movies.csv
yebhisahi.py C:\Us...
lastScrapping.py...
movierandomiser....
HANGMAN
.vs
hang.jpg
hangman.jpg
hangman.py
img1.jpg
img1.png
img2.png
img3.png
img4.png
movies.csv
OUTLINE
C: > Users > Anukrity > Desktop > New folder > hello.py > ...
1 from tkinter import ttk
2 from tkinter import messagebox
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 "Thriller",
19 "Drama",
20 "Mystery",
21 "Crime",
22 "Animation",
23 "Adventure",
24 "Fantasy"))
25
26 class TableDropDown(ttk.Combobox):
27 def __init__(self, parent):
tk
Romance
Get Value
you have selected Romance
please wait while the data is been extracted
OK
TERMINAL
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
1: Python
lder/Python-Tkinter-FrameWork/hangman/hangman.py"
4
4
4
4
4
4
PS C:\Users\Anukrity\Desktop\New folder\Python-Tkinter-FrameWork\hangman> & C:/Users/Anukrity/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Anukrity/Desktop/New folder/hello.py"
Python 3.9.0 64-bit
Ln 12, Col 1
Spaces: 4
UTF-8
CRLF
Python
9:31 PM
11/23/2020

How we Scraped Data From A Website?

When we run the code for web scraping, a request is sent to the URL that you have mentioned. As a response to the request, the server sends the data and allows you to read the HTML or XML page. The code then, parses the HTML or XML page, finds the data and extracts it.

To extract data using web scraping with python, you need to follow these basic steps:

1. Find the URL that is to be scraped,, in our case we scrapes data of genres through different page URLs: Comedy Sci-fi, Horror, Romance, action, Thriller, Drama Mystery, Crime

2. Inspecting the Page

Find the data we want to extract, in our case we extracted : titles, Years, Time, imdb_ratings, Metascores

Write the code

1. Run the code and extract the data

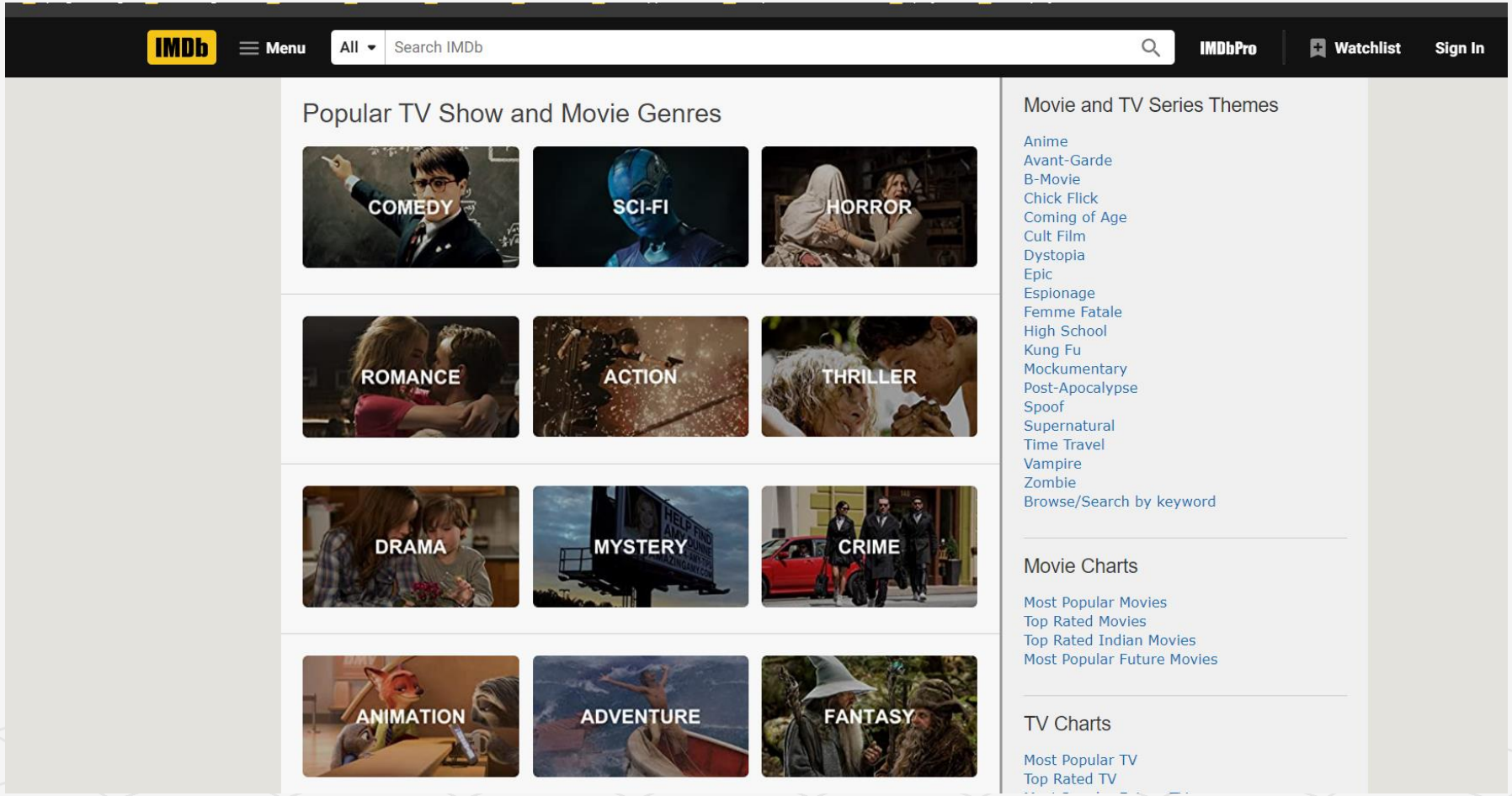
2. Store the data in the required format

Where we found the data and how?

These movie info are taken up from an online site(IMBD) containing all details about the movie, this detail is randomly selected using web scrapping and a random spaced word is generated

Web scrapping IMBD :

Using regex, we are scrapping movie names from IMBD in the list of movie name and genre



New Tab

imdb.com/search/title/?genres=comedy&explore=title_type,genres&pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=3396781f-d87f-4fac-8694-c56ce6f490fe&pf_...

programming | Learning sites | Acads | to do.. | Dreams | fitness | Job Application | Imported from Chr... | projects | TOC project

IMDb | Companies | Search IMDb | Sign In

Top 50 Comedy Movies and TV Shows

1-50 of 1,567,953 titles. | Next »

View Mode: Compact | **Detailed**

Sort by: **Popularity▲** | A-Z | User Rating | Number of Votes | US Box Office | Runtime | Year | Release Date | Date of Your Rating | Your Rating

1. **Schitt's Creek** (2015–2020)

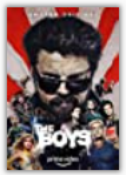
TV-14 | 22 min | Comedy

★ 8.5 ☆ Rate this

When rich video-store magnate Johnny Rose and his family suddenly find themselves broke, they are forced to leave their pampered lives to regroup in Schitt's Creek.

Stars: Eugene Levy, Catherine O'Hara, Dan Levy, Annie Murphy

Votes: 54,519



2. **The Boys** (2019–)

TV-MA | 60 min | Action, Comedy, Crime

★ 8.7 ☆ Rate this

A group of vigilantes sets out to take down corrupt superheroes who abuse their superpowers.

Stars: Karl Urban, Jack Quaid, Antony Starr, Erin Moriarty

Votes: 216,366

3. **Freaky** (2020)

R | 102 min | Comedy, Horror, Thriller

★ 6.9 ☆ Rate this 66 Metascore

After swapping bodies with a deranged serial killer, a young girl in high school discovers she has less than 24 hours before the change becomes permanent.

Director: Christopher Landon | Stars: Vince Vaughn, Kathryn Newton, Celeste

Title Type

TV Episodes (1,203,066)
Short Films (161,575)
Feature Films (95,664)
TV Series (47,498)
Video (21,768)
TV Movies (19,238)
TV Mini-series (7,414)
TV Shorts (5,783)
TV Specials (4,380)
Video Games (1,410)

Genres

Comedy (1,567,953)
Drama (286,540)
Short (223,472)
Family (202,564)
Talk-Show (193,813)
Romance (183,595
Animation (162,217)
Music (107,615)
Adventure (100,792)
Fantasy (91,543)
Action (79,271)
Sci-Fi (56,318)
News (45,137)
Crime (43,536)
Game-Show (40,454)
Mystery (37,321)
Musical (33,390)
Horror (32,005)
Thriller (29,279)
Reality-TV (28,991)
Documentary (21,410)
Sport (20,537)
History (8,553)
Western (4,026)
Biography (3,738)
War (3,198)
Film-Noir (3)

Elements | Console | Sources | Network | 8 2

```
uet("ns");
}
</script>
<div id="f6c4794f-ba5c-423c-a97a-12cab73c5ed4">…</div>
<script type="text/javascript">…</script>
<script type="text/javascript">…</script>
<script>
if (typeof uet == 'function') {
uet("ne");
}
</script>
<div id="wrapper">…</div> == $0
<script>…</script>
<div id="b287cf0c-c991-4c69-873c-7a738ff29f04">…</div>
<script type="text/javascript">…</script>
<script type="text/javascript">…</script>
<script>…</script>
<script>…</script>
<script>…</script>
<script type="text/javascript" src="https://m.media-amazon.com/images/G/01/imdb/js/collections/common-2354271187._CB427941191_.js"></script>
<script type="text/javascript" src="https://m.media-amazon.com/images/G/01/imdb/js/collections/jquery-185131483._CB447141647_.js"></script>
```

html | body#styleguide-v2.fixed | div#wrapper

Styles | Computed | Layout | Event Listeners | DOM Breakpoints | Properties | Accessibility

Filter :hov .cls

```
element.style {
```

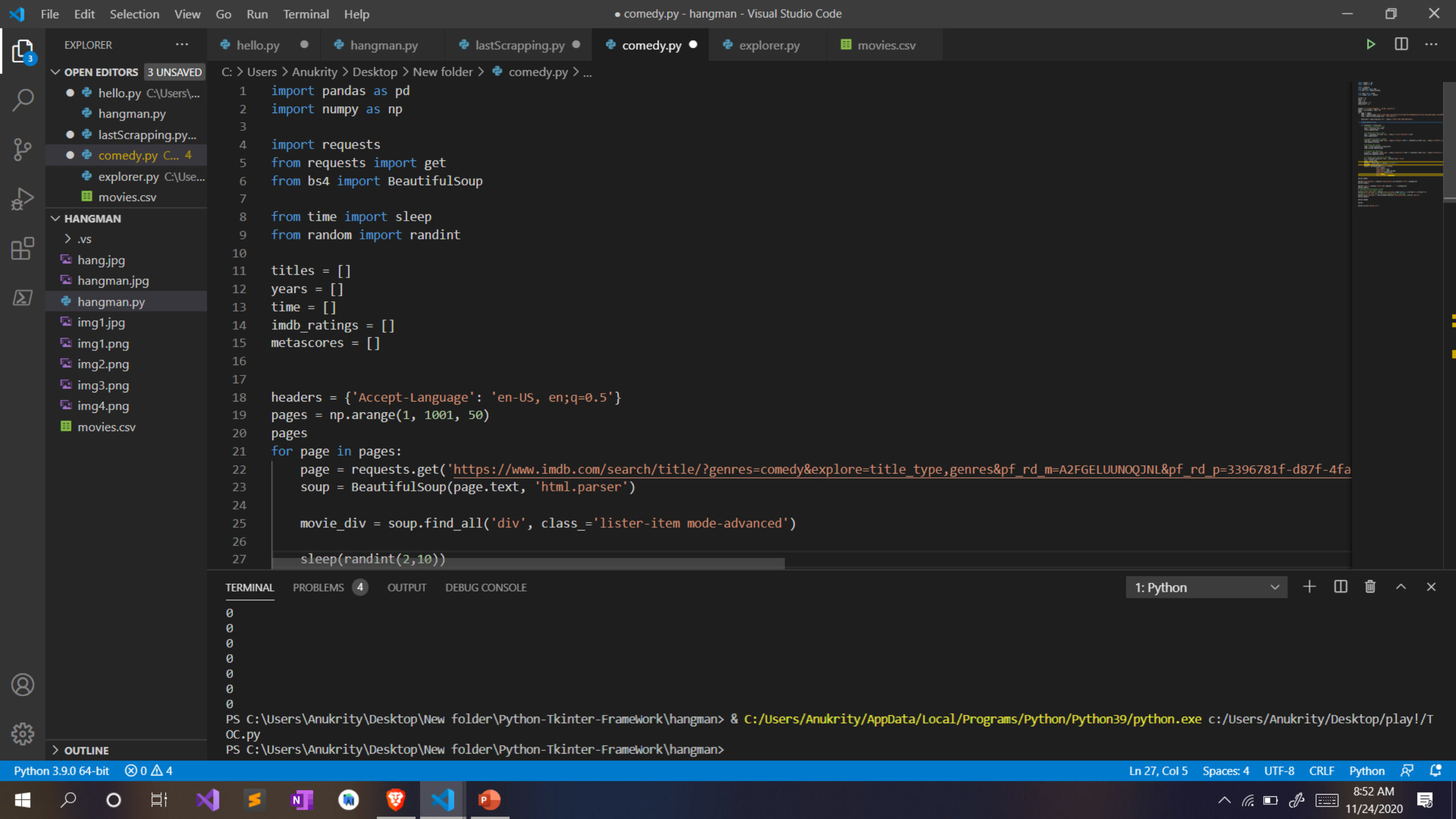
Console | What's New

Highlights from the Chrome 87 update

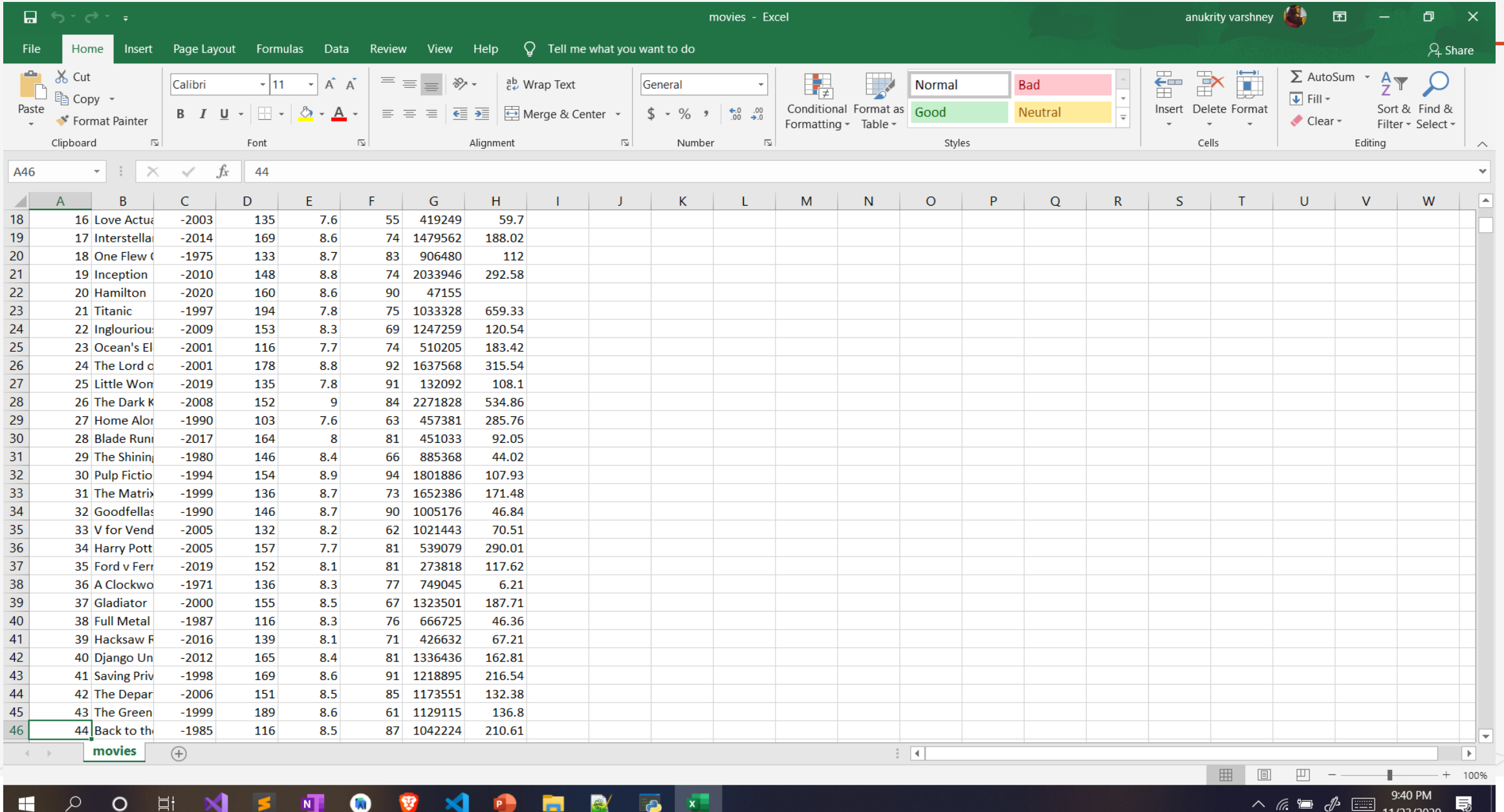
New CSS Grid debugging tools
Debug and inspect CSS Grid with the new CSS Grid debugging tools.

New WebAuthn tab
Emulate authenticators and debug the Web Authentication API with the new WebAuthn tab.

8:54 AM
11/24/2020


File Edit Selection View Go Run Terminal Help
• comedy.py - hangman - Visual Studio Code
EXPLORER
OPEN EDITORS 3 UNSAVED
hello.py C:\Users\...
hangman.py
lastScrapping.py...
comedy.py C... 4
explorer.py C:\Use...
movies.csv
HANGMAN
.vs
hang.jpg
hangman.jpg
hangman.py
img1.jpg
img1.png
img2.png
img3.png
img4.png
movies.csv
OUTLINE
hello.py
hangman.py
lastScrapping.py
comedy.py
explorer.py
movies.csv
C: > Users > Anukrity > Desktop > New folder > comedy.py > ...
1 import pandas as pd
2 import numpy as np
3
4 import requests
5 from requests import get
6 from bs4 import BeautifulSoup
7
8 from time import sleep
9 from random import randint
10
11 titles = []
12 years = []
13 time = []
14 imdb_ratings = []
15 metascores = []
16
17
18 headers = {'Accept-Language': 'en-US, en;q=0.5'}
19 pages = np.arange(1, 1001, 50)
20 pages
21 for page in pages:
22 page = requests.get('https://www.imdb.com/search/title/?genres=comedy&explore=title_type,genres&pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=3396781f-d87f-4fa
23 soup = BeautifulSoup(page.text, 'html.parser')
24
25 movie_div = soup.find_all('div', class_='lister-item mode-advanced')
26
27 sleep(randint(2,10))
TERMINAL
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
1: Python
0
0
0
0
0
0
0
PS C:\Users\Anukrity\Desktop\New folder\Python-Tkinter-FrameWork\hangman> & C:/Users/Anukrity/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Anukrity/Desktop/play!/TOC.py
PS C:\Users\Anukrity\Desktop\New folder\Python-Tkinter-FrameWork\hangman>
Python 3.9.0 64-bit
0 4
Ln 27, Col 5
Spaces: 4
UTF-8
CRLF
Python
8:52 AM
11/24/2020

Movie detail imported to csv file for data manipulation



	A	B	C	D	E	F	G	H
18	16	Love Actua	-2003	135	7.6	55	419249	59.7
19	17	Interstella	-2014	169	8.6	74	1479562	188.02
20	18	One Flew (	-1975	133	8.7	83	906480	112
21	19	Inception	-2010	148	8.8	74	2033946	292.58
22	20	Hamilton	-2020	160	8.6	90	47155	
23	21	Titanic	-1997	194	7.8	75	1033328	659.33
24	22	Inglourious	-2009	153	8.3	69	1247259	120.54
25	23	Ocean's El	-2001	116	7.7	74	510205	183.42
26	24	The Lord o	-2001	178	8.8	92	1637568	315.54
27	25	Little Won	-2019	135	7.8	91	132092	108.1
28	26	The Dark K	-2008	152	9	84	2271828	534.86
29	27	Home Alor	-1990	103	7.6	63	457381	285.76
30	28	Blade Runı	-2017	164	8	81	451033	92.05
31	29	The Shinin	-1980	146	8.4	66	885368	44.02
32	30	Pulp Fictio	-1994	154	8.9	94	1801886	107.93
33	31	The Matrix	-1999	136	8.7	73	1652386	171.48
34	32	Goodfellas	-1990	146	8.7	90	1005176	46.84
35	33	V for Vend	-2005	132	8.2	62	1021443	70.51
36	34	Harry Pott	-2005	157	7.7	81	539079	290.01
37	35	Ford v Ferr	-2019	152	8.1	81	273818	117.62
38	36	A Clockwo	-1971	136	8.3	77	749045	6.21
39	37	Gladiator	-2000	155	8.5	67	1323501	187.71
40	38	Full Metal	-1987	116	8.3	76	666725	46.36
41	39	Hacksaw F	-2016	139	8.1	71	426632	67.21
42	40	Django Un	-2012	165	8.4	81	1336436	162.81
43	41	Saving Priv	-1998	169	8.6	91	1218895	216.54
44	42	The Depar	-2006	151	8.5	85	1173551	132.38
45	43	The Green	-1999	189	8.6	61	1129115	136.8
46	44	Back to th	-1985	116	8.5	87	1042224	210.61

How do the hangman game operates?

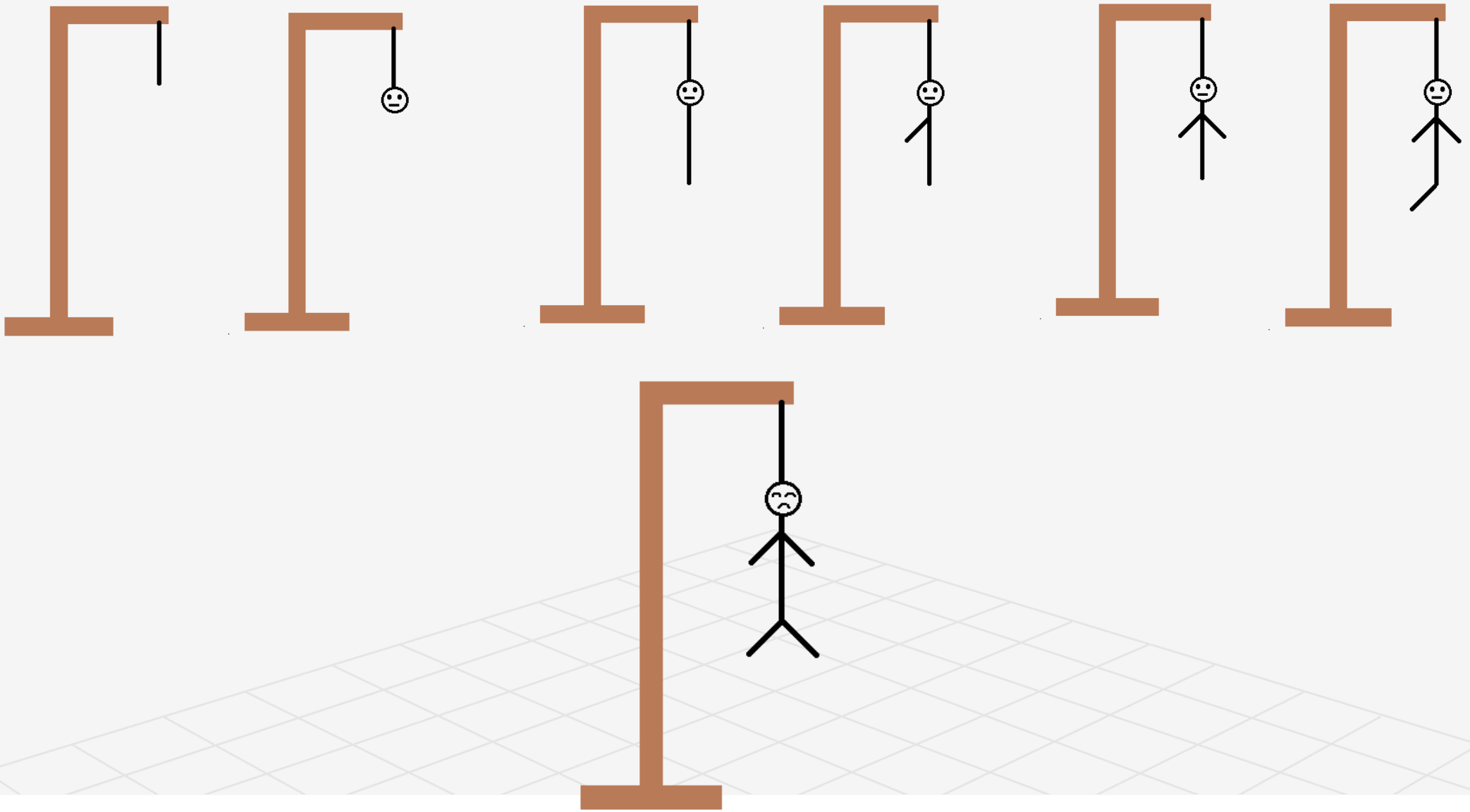
From the data we extracted we select a random row using random function in python, which extracts all the data related to movie in that row, there we extract the movie name and replace it with dashes,

We represent the Hangman with the help of Tkinter GUI which creates an input window, which asks us to guess the word,

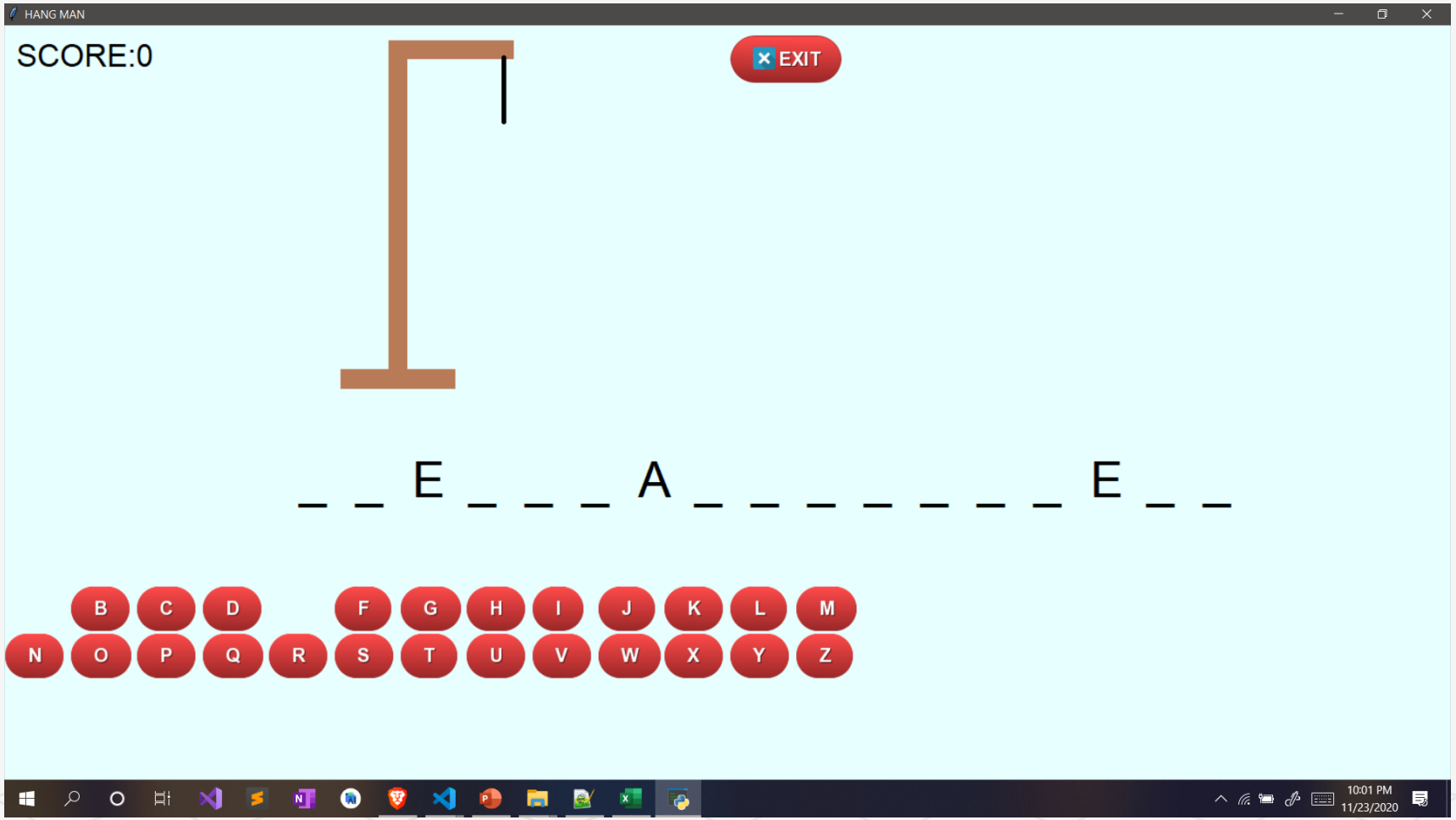
If the alphabetical letter we clicked matches the hidden movie letters, it fills it it in all right places

If you guess the wrong letter, the chances decreases and the image changes.

Images of changing Hangman image:



Data taken from web scrapping for hangman game



Where is Regex used:

- for web scrapping the movie names with specified genre:
  - We will scrap the html for all the movie names with tag of the genre like for comedy, thriller, action, romance, etc
- string matching while we play the game with guessed word:
  - If the guessed word exist in the movie name then the dash gets replaced with the word
  - Else the number of chance gets decreased by one.

Conclusions

The most effective way to learn technical skills is by doing. While this whole project could have been done manually inserting values into Excel, we like to take the long view and think about how the skills learned here will help in the future. The process of learning is more important than the final result, and in this project we were able to see how to use 3 critical skills for data science:

1. Web Scraping: Retrieving online data

2. Regular Expressions: Parsing our data to extract information

3. Visualization: Showcasing all our hard work using GUI in a Hangman game