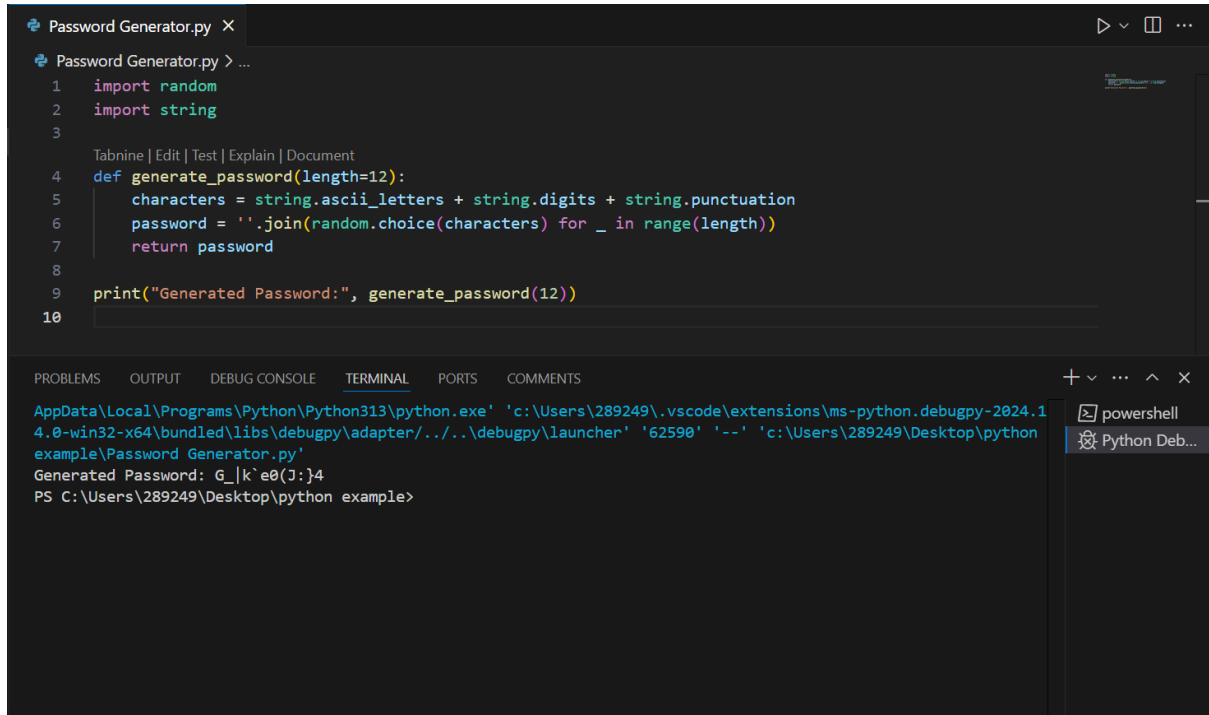


## Python projects:

### 1.Password Generator:



The image shows a Visual Studio Code editor window with a Python file named 'Password Generator.py'. The code defines a function 'generate\_password' that takes a length parameter (default 12) and returns a random password. The password is generated by joining random choices from a set of characters (ASCII letters, digits, and punctuation). The script also includes a print statement to display the generated password.

```
1 import random
2 import string
3
4 def generate_password(length=12):
5     characters = string.ascii_letters + string.digits + string.punctuation
6     password = ''.join(random.choice(characters) for _ in range(length))
7     return password
8
9 print("Generated Password:", generate_password(12))
10
```

The terminal output shows the execution of the script, displaying the generated password: 'G\_|k'e0(J:}4'.

```
AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\289249\.vscode\extensions\ms-python.debugpy-2024.1
4.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '62590' '--' 'c:\Users\289249\Desktop\python
example\Password Generator.py'
Generated Password: G_|k'e0(J:}4
PS C:\Users\289249\Desktop\python example>
```

## 2. To-Do List (CLI):

```
To-Do List (CLI).py x
To-Do List (CLI).py > ...
1  tasks = []
2
3  while True:
4      print("\n1. Add Task\n2. View Tasks\n3. Remove Task\n4. Exit")
5      choice = input("Enter choice: ")
6
7      if choice == "1":
8          task = input("Enter task: ")
9          tasks.append(task)
10         print("Task added!")
11
12         elif choice == "2":
13             print("\nTo-Do List:")
14             for idx, task in enumerate(tasks, 1):
15                 print(f"{idx}. {task}")
16
17         elif choice == "3":
18             task_num = int(input("Enter task number to remove: "))
19             if 0 < task_num <= len(tasks):
20                 tasks.pop(task_num - 1)
21                 print("Task removed!")
22
23         elif choice == "4":
24             break
25
26         else:
27             print("Invalid choice. Try again.")
28
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE
4. Exit
Enter choice: 1
Enter task: gym,swimming
Task added!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 2

To-Do List:
1. gym,swimming

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 3
Enter task number to remove: 1
Task removed!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 4
PS C:\Users\289249\Desktop\python example>
```

### 3. Weather App (API-based):

```
python weather_app.py X
python weather_app.py > ...
3 # Your actual API key
4 API_KEY = "b9f26b57eb51a90e313e3b7def017319" # Your OpenWeather API key
5
6 # Get the city name from the user
7 city = input("Enter city name: ")
8
9 # Create the URL to fetch weather data
10 url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"
11
12 # Make a GET request to the API
13 response = requests.get(url).json()
14
15 # Check if the city was found and return the weather information
16 if response["cod"] == 200:
17     print(f"City: {response['name']}")
18     print(f"Temperature: {response['main']['temp']}°C")
19     print(f"Weather: {response['weather'][0]['description']}")
20 else:
21     print("City not found!")
22
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ... ^ x

/Desktop/python example/python weather_app.py"
Enter city name: chennai
City: Chennai
Temperature: 27.14°C
Enter city name: chennai
City: Chennai
Temperature: 27.14°C
City: Chennai
Temperature: 27.14°C
Weather: clear sky
PS C:\Users\289249\Desktop\python example>

> & C:/Users/289249/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/289249
/Desktop/python example/python weather_app.py"
Enter city name: london
City: London
Temperature: 2.82°C
Weather: overcast clouds
PS C:\Users\289249\Desktop\python example> []
```

## 4. Number Guessing Game:

```
Number Guessing Game.py X
Number Guessing Game.py > ...
1  import random
2
3  number = random.randint(1, 100)
4
5  while True:
6      guess = int(input("Guess the number (1-100): "))
7
8      if guess < number:
9          print("Too low! Try again.")
10     elif guess > number:
11         print("Too high! Try again.")
12     else:
13         print("Congratulations! You guessed it right.")
14         break
15
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\289249\Desktop\python example> & C:/Users/289249/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/289249/Desktop/python example/Number Guessing Game.py"
Guess the number (1-100): 77
Too high! Try again.
Guess the number (1-100): 50
Guess the number (1-100): 50
Too low! Try again.
Guess the number (1-100): 22
Too low! Try again.
Guess the number (1-100): 3
Too low! Try again.
Guess the number (1-100): 9
Too low! Try again.
Guess the number (1-100): 9
Guess the number (1-100): 9
Too low! Try again.
Guess the number (1-100): 99
Too high! Try again.
Guess the number (1-100): 67
Guess the number (1-100): 99
Too high! Try again.
Guess the number (1-100): 99
Guess the number (1-100): 99
Guess the number (1-100): 99
Too high! Try again.
Guess the number (1-100): 99
Too high! Try again.
Guess the number (1-100): 67
Congratulations! You guessed it right.
PS C:\Users\289249\Desktop\python example>
```

## 5. QR Code Generator:

```
QR Code Generator.py X
QR Code Generator.py > ...
1  import qrcode
2
3  data = input("Enter text or URL: ")
4  qr = qrcode.make(data)
5  qr.save("qrcode.png")
6  print("QR Code generated and saved as 'qrcode.png'!")
7  import qrcode
8
9  # Ask the user for input (text or URL)
10 data = input("Enter text or URL: ")
11
12 # Create the QR code using the entered data
13 qr = qrcode.make(data)
14
15 # Save the generated QR code as an image file
16 qr.save("qrcode.png")
17
18 # Print a confirmation message
19 print("QR Code generated and saved as 'qrcode.png'!")
20
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
"c:/Users/289249/Desktop/python example/QR Code Generator.py"
"c:/Users/289249/Desktop/python example/QR Code Generator.py"
Enter text or URL: https://github.com/ssethumadav
Enter text or URL: https://github.com/ssethumadav
QR Code generated and saved as 'qrcode.png'!
Enter text or URL: 
```

