

Python mini projects

Password Generator

```
main.py
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 print("Generated Password:", generate_password(12))
```

input

Generated Password: =4Hu6w\$!OSIj

...Program finished with exit code 0
Press ENTER to exit console.

To-do List (CLI)

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

Python mini projects

Weather App (API-based)

```
main.py
1 import requests
2
3 API_KEY = "bd5e378503939ddaee76f12ad7a97608"
4 city = input("Enter city name: ")
5 url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"
6
7 response = requests.get(url).json()
8
9 if response["cod"] == 200:
10     print(f"City: {response['name']}")
11     print(f"Temperature: {response['main']['temp']}°C")
12     print(f"Weather: {response['weather'][0]['description']}")
13 else:
14     print("City not found!")
```

input

Enter city name: Paris
City: Paris
Temperature: 5.29°C
Weather: overcast clouds

Number Guessing Game

```
main.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     if guess < number:
8         print("Too low! Try again.")
9     elif guess > number:
10        print("Too high! Try again.")
11    else:
12        print("Congratulations! You guessed it right.")
13        break
```

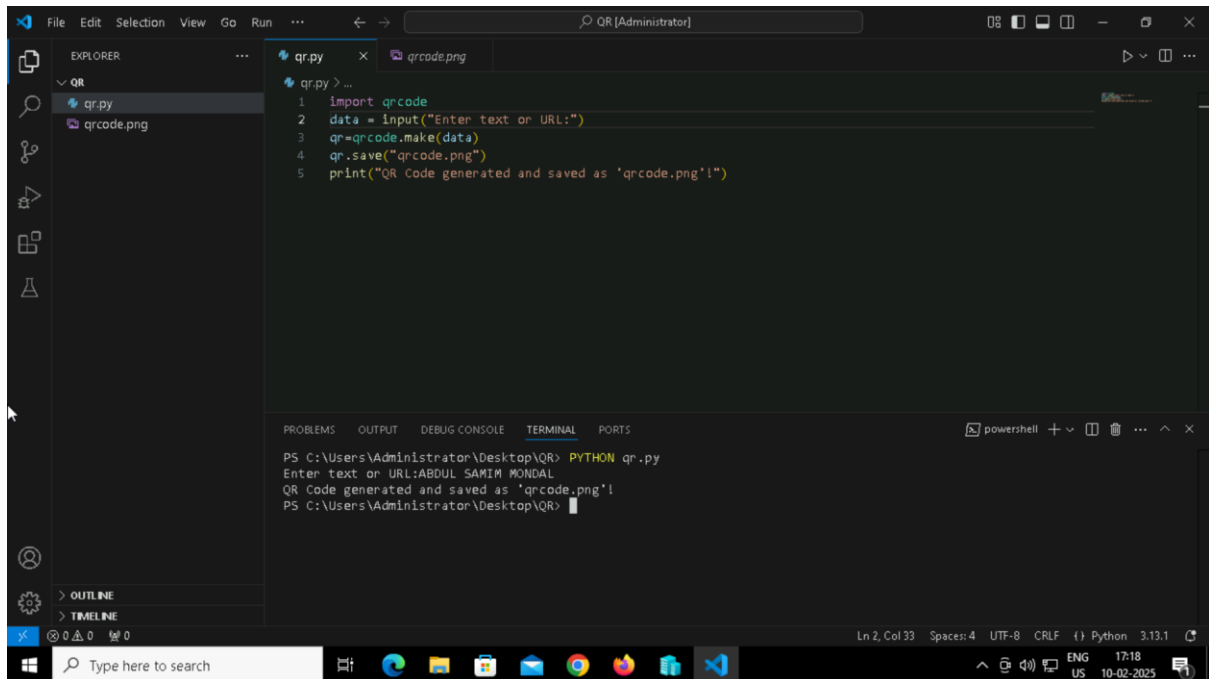
input

Guess the number (1-100): 4
Congratulations! You guessed it right.

...Program finished with exit code 0
Press ENTER to exit console.

Python mini projects

QR CODE GENERATOR



This screenshot shows a Visual Studio Code editor window with a Python script named `qr.py` open. The script uses the `qrcode` library to generate a QR code from user input. The Explorer sidebar on the left shows the project structure with a folder named `QR` containing `qr.py` and `qrcode.png`. The terminal at the bottom shows the command `PYTHON qr.py` being executed, followed by the user input `ABDUL SAMIM MONDAL` and the output message `QR Code generated and saved as 'qrcode.png'!`.

```
1 import qrcode
2 data = input("Enter text or URL:")
3 qr=qrcode.make(data)
4 qr.save("qrcode.png")
5 print("QR Code generated and saved as 'qrcode.png'!")
```

PS C:\Users\Administrator\Desktop\QR> PYTHON qr.py
Enter text or URL:ABDUL SAMIM MONDAL
QR Code generated and saved as 'qrcode.png'!
PS C:\Users\Administrator\Desktop\QR>

