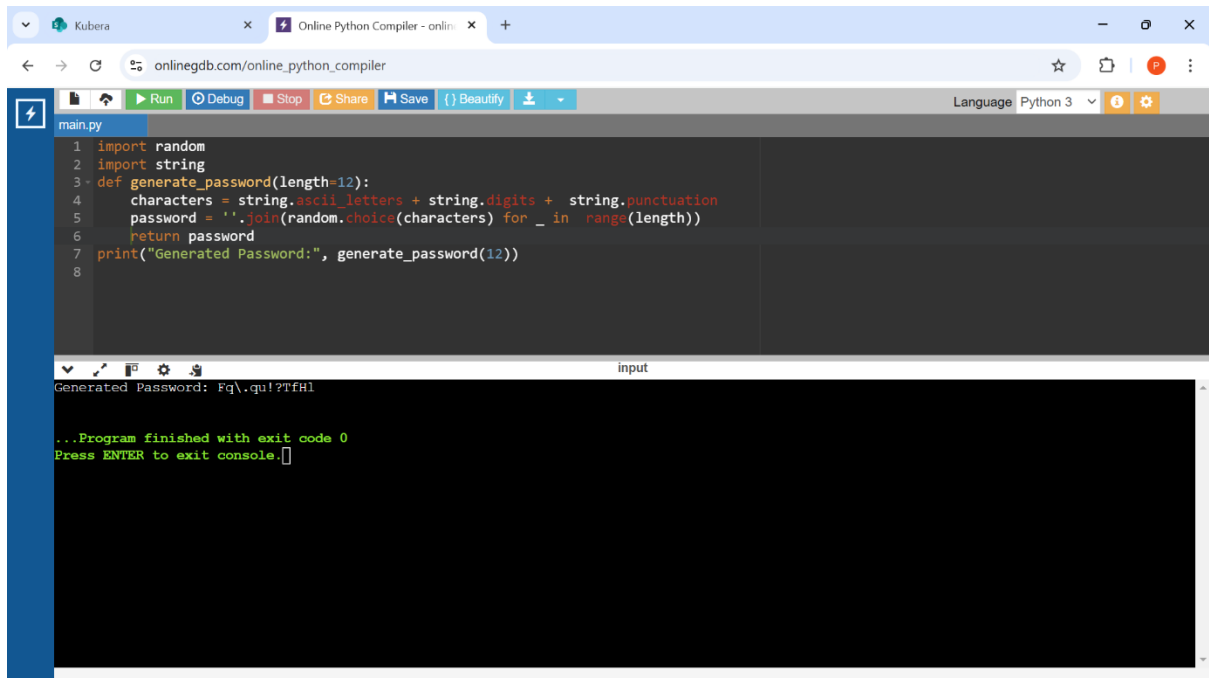


Basic Project

Password Generator



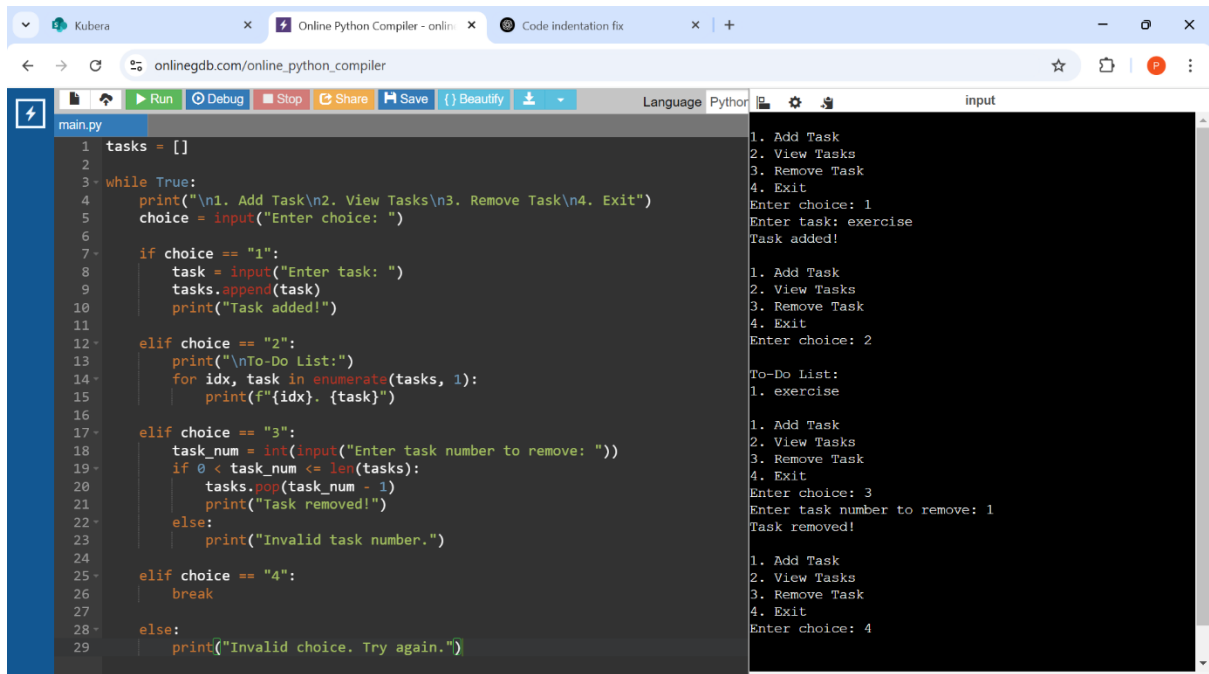
The screenshot shows a web browser with the URL `onlinegdb.com/online_python_compiler`. The code editor contains a Python script named `main.py` that generates a random password of length 12. The script imports `random` and `string`, defines a `generate_password` function, and prints the result. The output console shows the generated password: `Fq\qu!TfH1`. The program finished with exit code 0.

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

Generated Password: Fq\qu!TfH1

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

To-Do List (CLI)



The screenshot shows a web browser with the URL `onlinegdb.com/online_python_compiler`. The code editor contains a Python script named `main.py` that implements a To-Do List CLI. The script uses a `while` loop to handle user input and perform actions like adding, viewing, and removing tasks. The output console shows the user interacting with the CLI, entering choices and tasks, and seeing the resulting list of tasks.

```
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        else:
23            print("Invalid task number.")
24
25    elif choice == "4":
26        break
27
28    else:
29        print("Invalid choice. Try again.")
```

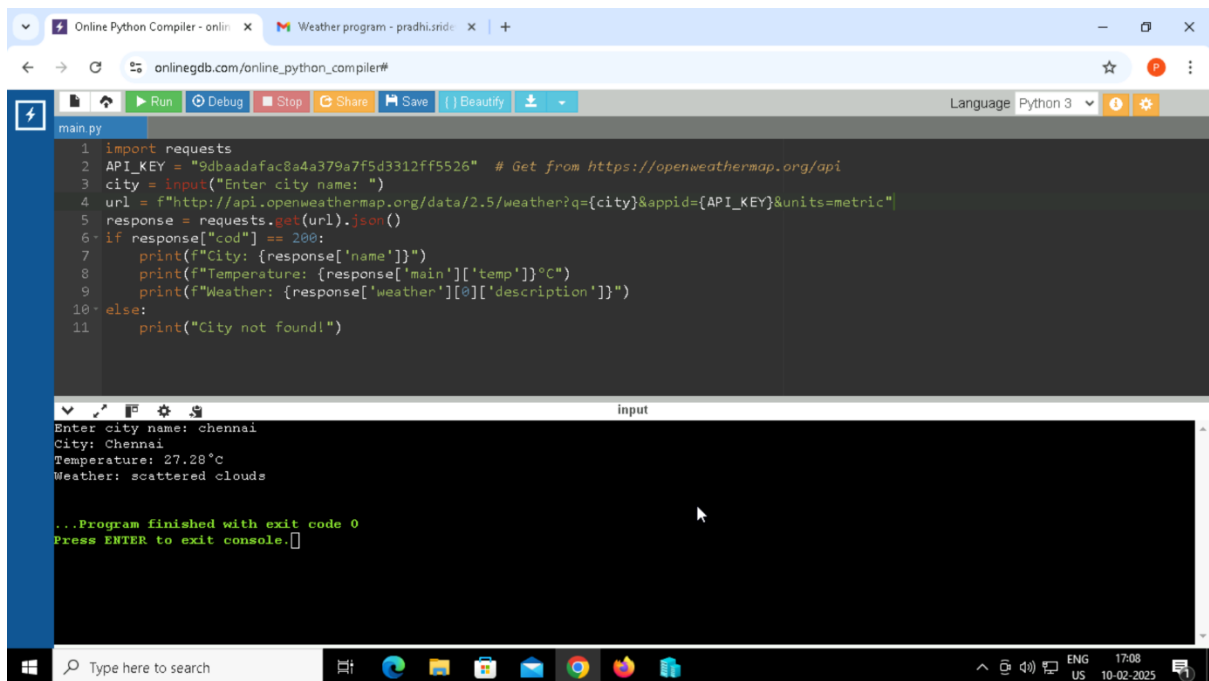
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 1
Enter task: exercise
Task added!

1. Add Task
2. View Tasks
3. Remove Task
4. Exit
Enter choice: 2
To-Do List:
1. exercise

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

Weather App (API-based)



The screenshot shows a web browser with two tabs: "Online Python Compiler - online" and "Weather program - pradhilaxide". The address bar shows "onlinegdb.com/online_python_compiler#". The main editor displays a Python script for a weather app. The script imports the 'requests' library, defines an API key, prompts the user for a city name, constructs a URL to the OpenWeatherMap API, and prints the city name, temperature, and weather description. The output window shows the user entering "chennai", and the program prints "City: Chennai", "Temperature: 27.28°C", and "Weather: scattered clouds". The program finished with exit code 0.

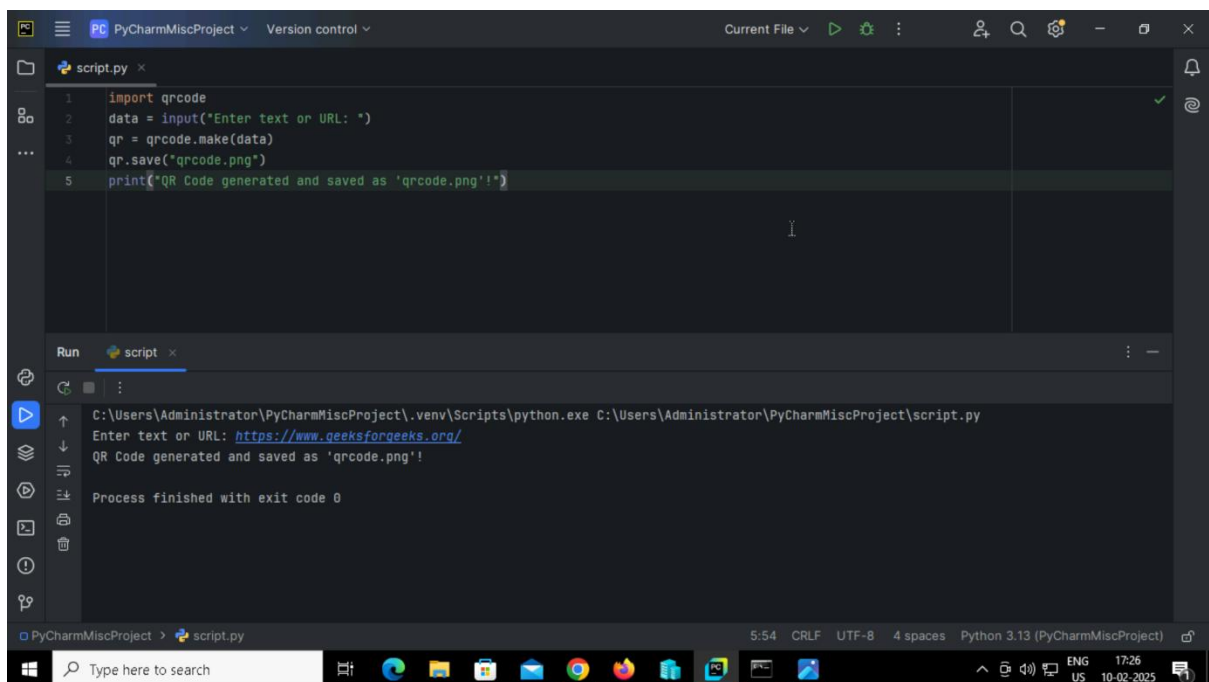
```
main.py
1 import requests
2 API_KEY = "9dbaadafac8a4a379a7f5d3312ff5526" # Get from https://openweathermap.org/api
3 city = input("Enter city name: ")
4 url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"
5 response = requests.get(url).json()
6 if response["cod"] == 200:
7     print(f"City: {response['name']}")
8     print(f"Temperature: {response['main']['temp']}°C")
9     print(f"Weather: {response['weather'][0]['description']}")
10 else:
11     print("City not found!")
```

input

Enter city name: chennai
City: Chennai
Temperature: 27.28°C
Weather: scattered clouds

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

QR Code Generator



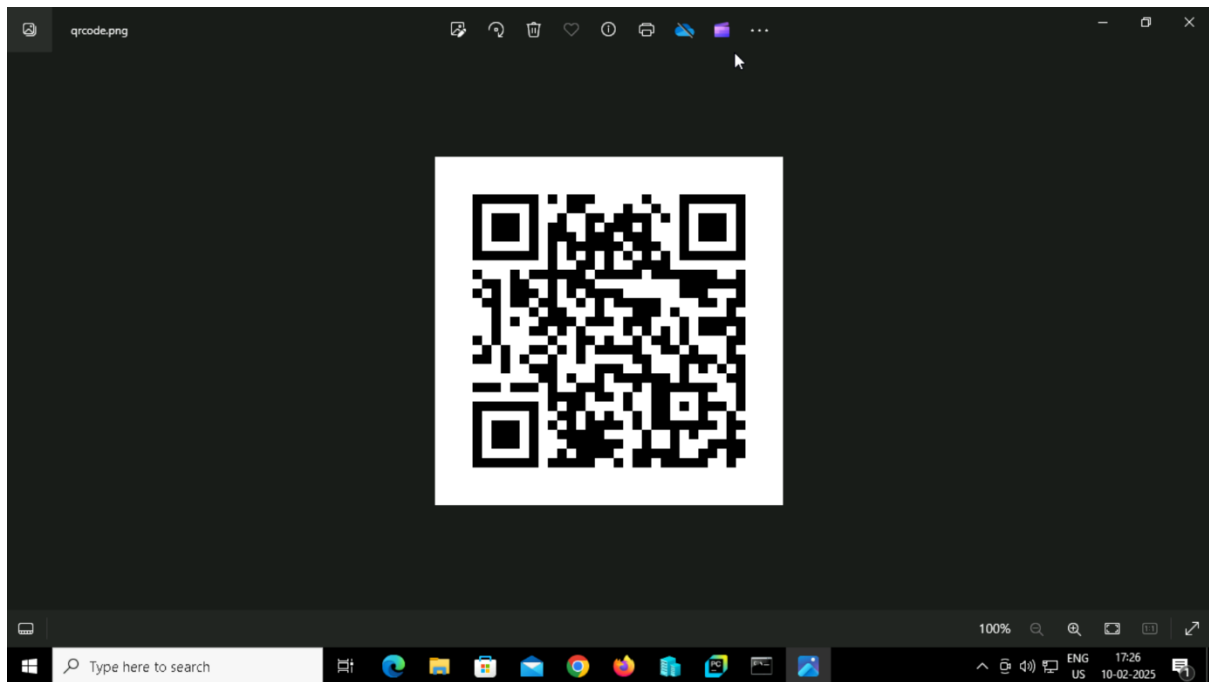
The screenshot shows the PyCharm IDE with a project named "PyCharmMiscProject". The main editor displays a Python script for a QR code generator. The script imports the 'qrcode' library, prompts the user for text or a URL, generates a QR code, saves it as 'qrcode.png', and prints a confirmation message. The Run window shows the command executed: "C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script.py". The output shows the user entering the URL "https://www.geeksforgeeks.org/", the QR code being generated and saved, and the process finishing with exit code 0.

```
script.py
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'!")
```

Run script

C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script.py
Enter text or URL: <https://www.geeksforgeeks.org/>
QR Code generated and saved as 'qrcode.png'!

Process finished with exit code 0



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
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
```

```
input
Guess the number (1-100): 78
Too high! Try again.
Guess the number (1-100): 36
Too low! Try again.
Guess the number (1-100): 55
Too high! Try again.
Guess the number (1-100): 45
Too low! Try again.
Guess the number (1-100): 50
Too low! Try again.
Guess the number (1-100): 51
Too low! Try again.
Guess the number (1-100): 52
Congratulations! You guessed it right.

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