

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Week 3 – Monday Name:

Manideep Yelugam

Hall Ticket No: 2303A510F8

Lab Objectives

- To understand ethical risks involved in AI-generated code.
- To identify issues related to privacy, security, and transparency.
- To analyze the responsibility of developers when using AI tools.
- To promote responsible and ethical AI coding practices.

Lab Outcomes

After completing this lab, students will be able to:

- Identify insecure coding patterns generated by AI tools.
- Analyze privacy and security risks in AI-generated programs.
- Understand the importance of transparency and explainability.
- Recognize the role of human responsibility in ethical AI coding.

Task Description #1: Privacy in API Usage

Objective:

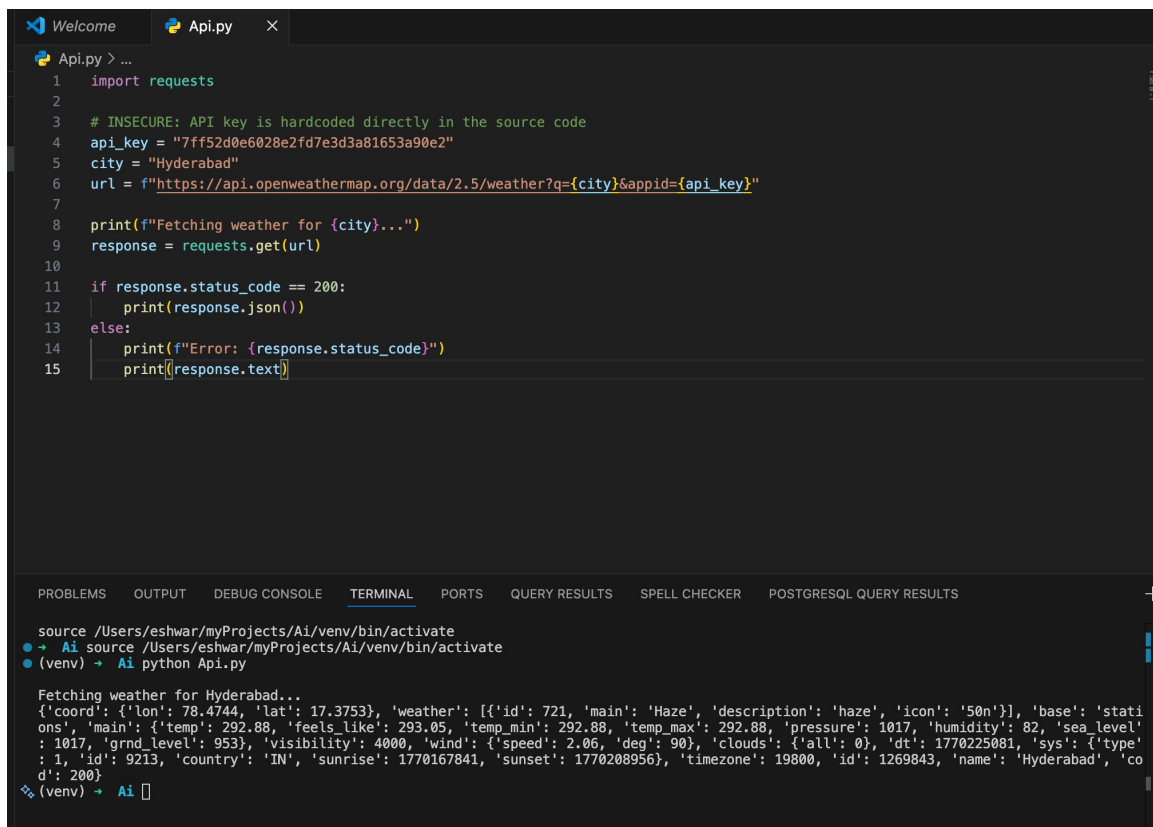
To generate a Python program that fetches weather data securely without exposing API keys.

Risk Analysis:

AI-generated code may hardcode API keys directly in the program. This is unsafe and may lead to security breaches.

Conclusion:

Using environment variables protects sensitive credentials and follows ethical security practices.



```
Api.py > ...
1 import requests
2
3 # INSECURE: API key is hardcoded directly in the source code
4 api_key = "7ff52d0e6028e2fd7e3d3a81653a90e2"
5 city = "Hyderabad"
6 url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"
7
8 print(f"Fetching weather for {city}...")
9 response = requests.get(url)
10
11 if response.status_code == 200:
12     print(response.json())
13 else:
14     print(f"Error: {response.status_code}")
15     print([response.text])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS SPELL CHECKER POSTGRESQL QUERY RESULTS

```
source /Users/eshwar/myProjects/Ai/venv/bin/activate
● → Ai source /Users/eshwar/myProjects/Ai/venv/bin/activate
● (venv) → Ai python Api.py

Fetching weather for Hyderabad...
{'coord': {'lon': 78.4744, 'lat': 17.3753}, 'weather': [{'id': 721, 'main': 'Haze', 'description': 'haze', 'icon': '50n'}], 'base': 'stations', 'main': {'temp': 292.88, 'feels_like': 293.05, 'temp_min': 292.88, 'temp_max': 292.88, 'pressure': 1017, 'humidity': 82, 'sea_level': 1017, 'grnd_level': 953}, 'visibility': 4000, 'wind': {'speed': 2.06, 'deg': 90}, 'clouds': {'all': 0}, 'dt': 1770225081, 'sys': {'type': 1, 'id': 9213, 'country': 'IN', 'sunrise': 1770167841, 'sunset': 1770208956, 'timezone': 19800, 'id': 1269843, 'name': 'Hyderabad', 'code': 200}}
(venv) → Ai
```

Task Description #2: Privacy & Security in File Handling

Objective:

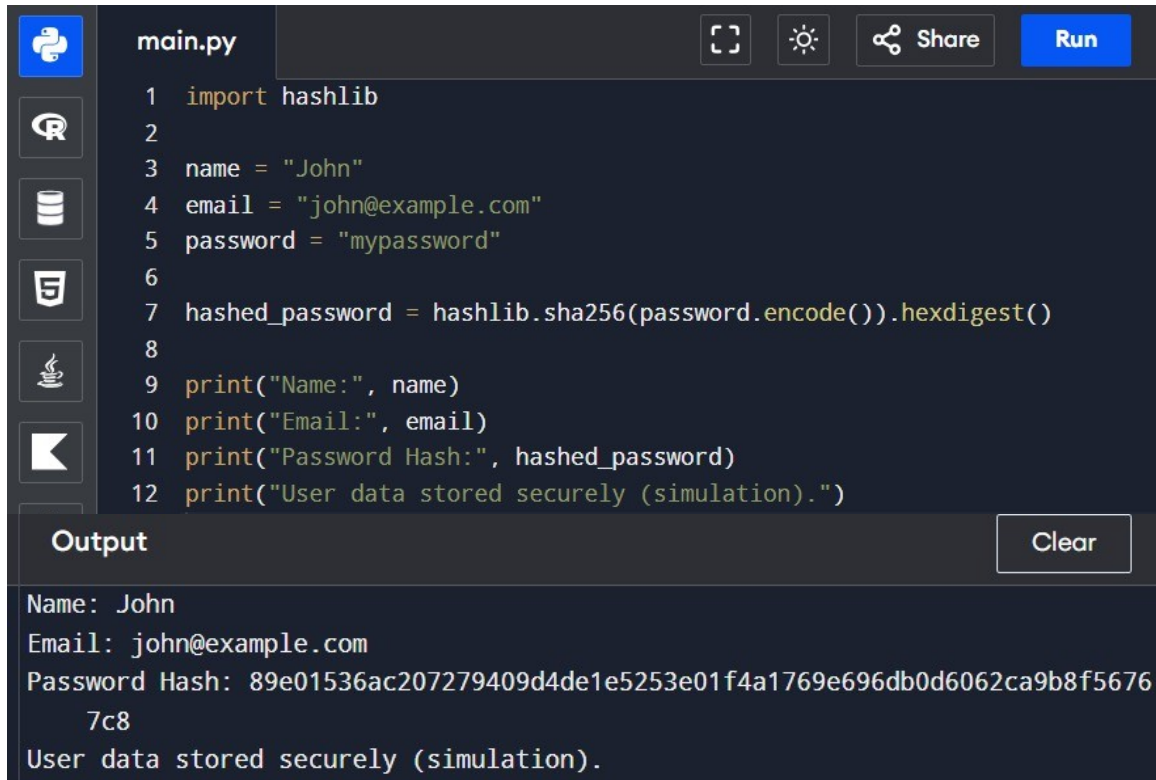
To analyze how AI-generated code stores user data and improve its security.

Privacy Risk Identified:

Storing passwords in plain text can compromise user accounts.

Conclusion:

Hashing passwords ensures data privacy and security.



The image shows a Python code editor interface. On the left, there is a sidebar with icons for Python, R, a database, a document, a terminal, and a back arrow. The main area displays a file named 'main.py' with the following code:

```
1 import hashlib
2
3 name = "John"
4 email = "john@example.com"
5 password = "mypassword"
6
7 hashed_password = hashlib.sha256(password.encode()).hexdigest()
8
9 print("Name:", name)
10 print("Email:", email)
11 print("Password Hash:", hashed_password)
12 print("User data stored securely (simulation).")
```

At the top right of the editor, there are icons for full screen, settings, and a 'Share' button, along with a blue 'Run' button. Below the code editor is an 'Output' panel with a 'Clear' button. The output shows the execution results:

```
Name: John
Email: john@example.com
Password Hash: 89e01536ac207279409d4de1e5253e01f4a1769e696db0d6062ca9b8f56767c8
User data stored securely (simulation).
```

Task Description #3: Transparency in Algorithm Design

Objective:

To create an Armstrong number checking program with clear explanation.

Explanation:








The program checks whether the sum of digits raised to the power of total digits equals the original number.

Conclusion:

The logic is simple, transparent, and easy to understand.

Conclusion:




Explainable AI systems are more ethical and user-friendly.



Programiz

Python Online Compiler

main.py

 Share

Run

```
1 def recommend_product(product, purchase_history):
2     if product in purchase_history:
3         return "Recommended because you purchased similar items
           earlier."
4     else:
5         return "Recommended based on popular customer trends."
6
7
8 history = ["Shoes", "Bag", "Watch"]
9
10 print(recommend_product("Shoes", history))
11 print(recommend_product("Laptop", history))
12
```

Output

```
Recommended because you purchased similar items earlier.
Recommended based on popular customer trends.
```