

## Assignment 5.1

---

**Course Title:** Ethical Foundations – Responsible AI

**Lab Title:** Responsible AI Coding Practices

**Lab No:** 5

**Week:** 3

**Day:** Monday

---

**Student Name:** Pochampalli Akshith

**Roll No:** 2303A54015

---

### Aim

To understand ethical risks in AI-generated code and apply responsible AI coding practices focusing on privacy, security, transparency, and accountability.

---

### Objectives

- Identify insecure coding patterns generated by AI
  - Analyze privacy and security risks
  - Ensure transparency and explainability in algorithms
  - Understand developer responsibility in AI-assisted programming
- 

### Tools Used

- VS Code
- 

### Task-Wise Implementation

---

#### Task 1: Privacy in API Usage

##### Problem Statement

Generate a Python program to fetch weather data securely without exposing API keys.

##### AI Prompt Used

Generate a Python program to fetch weather data from a weather API without hardcoding the API key.

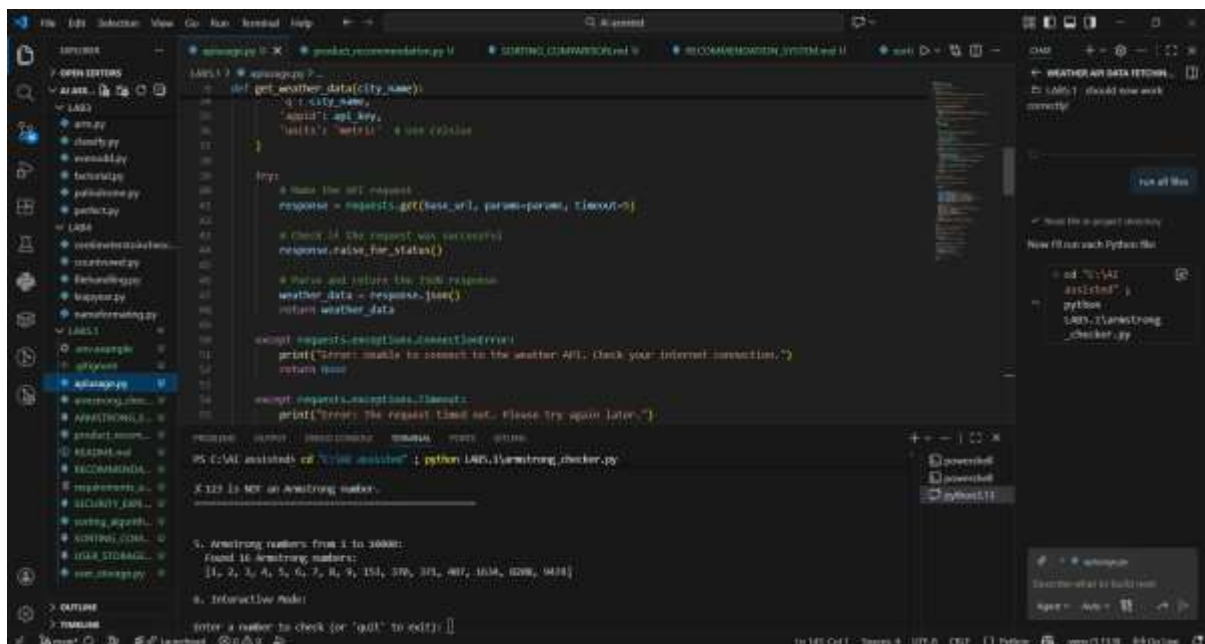
Use environment variables to store and access the API key securely.

Include basic error handling.

### Observation

- AI-generated code avoided hardcoded API keys
- Environment variables were used for security

### Secure Code



```
def get_weather_data(city_name):
    """Fetch weather data for a given city name using an API key stored in an environment variable.

    Args:
        city_name (str): The name of the city to fetch weather data for.

    Returns:
        dict: A dictionary containing weather data for the specified city.

    Raises:
        requests.exceptions.ConnectionError: If there is a problem connecting to the weather API.
        requests.exceptions.Timeout: If the request to the weather API times out.
    """
    # Get the API key from the environment variables
    api_key = os.getenv('API_KEY')

    # Check if the API key is set
    if not api_key:
        raise ValueError("API key is not set in environment variables")

    # Make the API request
    response = requests.get(base_url, params=params, timeout=5)

    # Check if the request was successful
    response.raise_for_status()

    # Parse and return the JSON response
    weather_data = response.json()
    return weather_data

except requests.exceptions.ConnectionError:
    print("Error: Unable to connect to the weather API. Check your internet connection.")
    return None

except requests.exceptions.Timeout:
    print("Error: The request timed out. Please try again later.")
```

### Ethical Analysis

Using environment variables protects sensitive credentials and prevents unauthorized misuse.

## Task 2: Privacy & Security in File Handling

### Problem Statement

Store user data securely without exposing sensitive information.

### AI Prompt Used

Generate a Python program to store user details (name, email, password) securely.

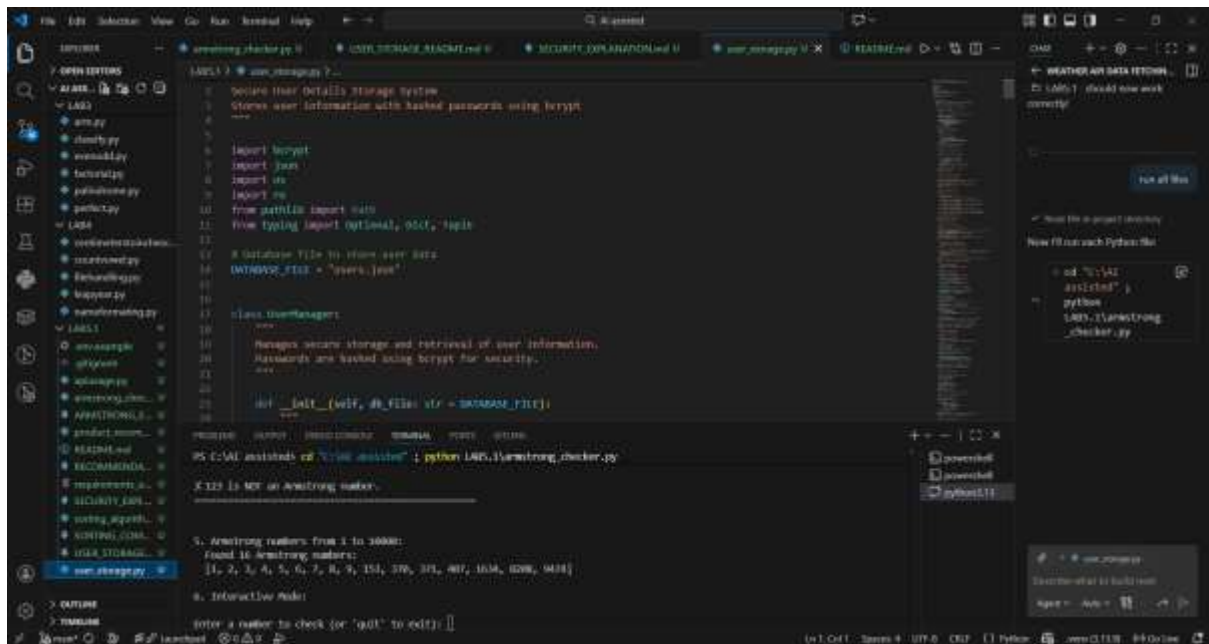
Do not store passwords in plain text.

Use hashing for password storage and explain why this approach is secure.

### Privacy Risk Identified

- Plain-text password storage is insecure

### Secure Code



## Ethical Analysis

Hashing ensures passwords cannot be recovered even if the file is leaked.

## Task 3: Transparency in Algorithm Design

### Problem Statement

Design an Armstrong number checking program with clear explanation.

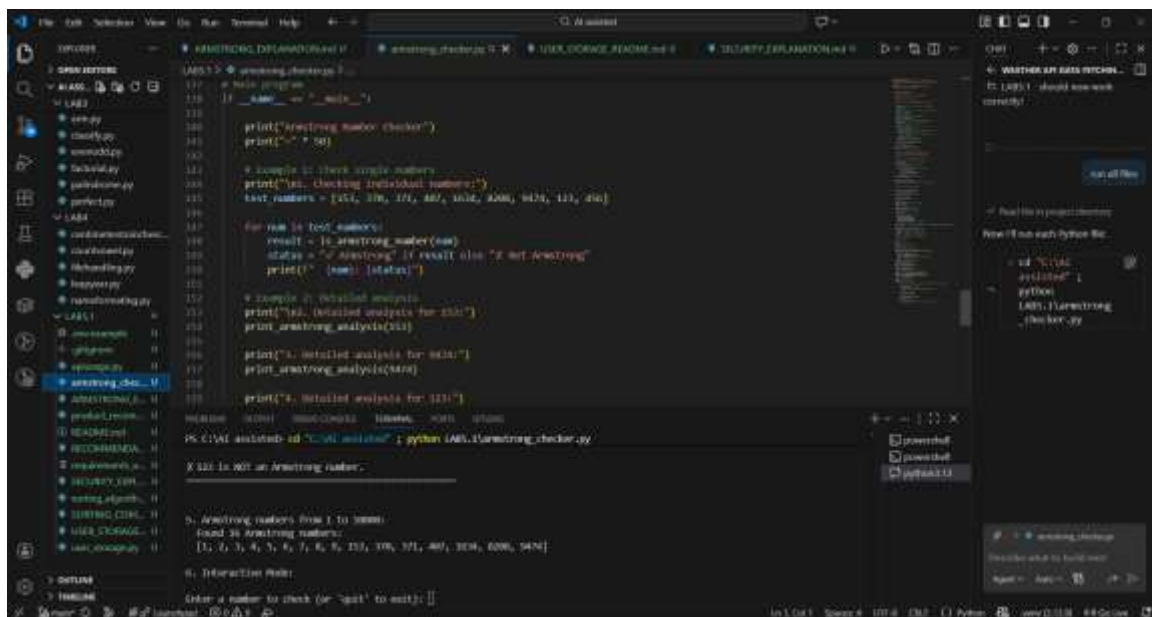
### AI Prompt Used

Generate a Python function to check whether a given number is an Armstrong number.

Add clear comments to every important line of code.

Also explain the code line-by-line in simple terms.

### Implementation



## Transparency Evaluation

The explanation clearly matches the program logic, ensuring understandability.

## Task 4: Transparency in Algorithm Comparison

### Problem Statement

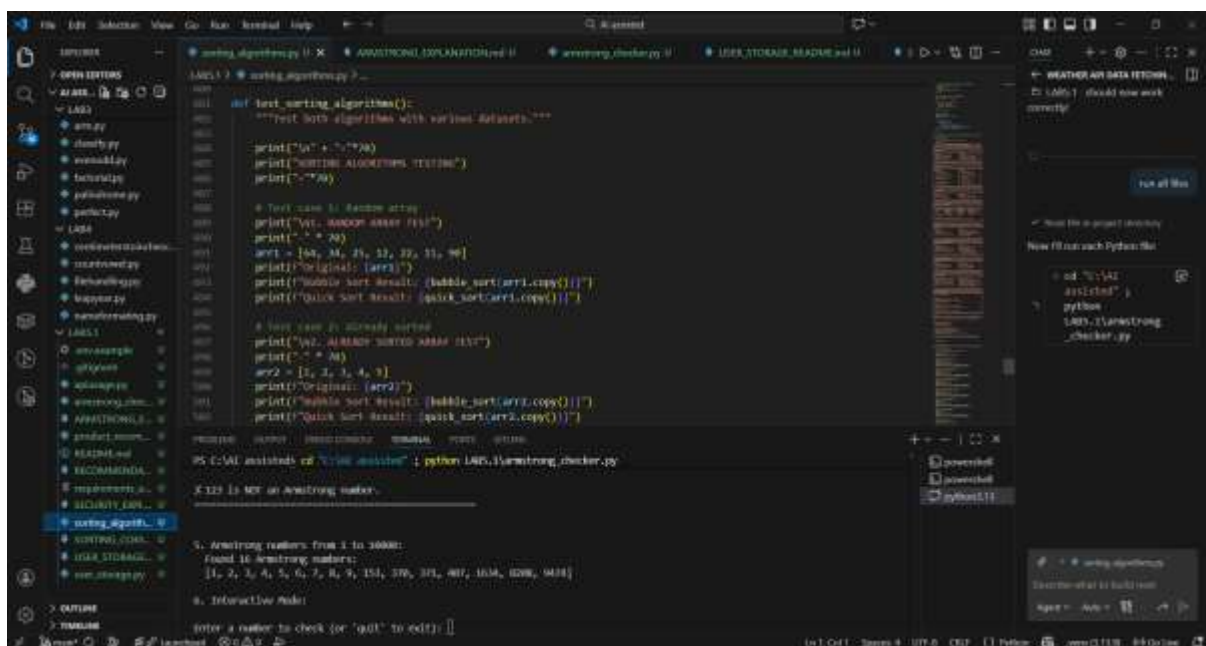
Implement and compare two sorting algorithms.

### AI Prompt Used

Generate Python code for Bubble Sort and Quick Sort.

Include step-by-step comments explaining how each algorithm works.

Compare both algorithms in terms of logic, time complexity, and efficiency.



Comparison

Algorithm    Time Complexity    Efficiency

Bubble Sort     $O(n^2)$     Low

Quick Sort     $O(n \log n)$     High

Task 5: Transparency in AI Recommendations

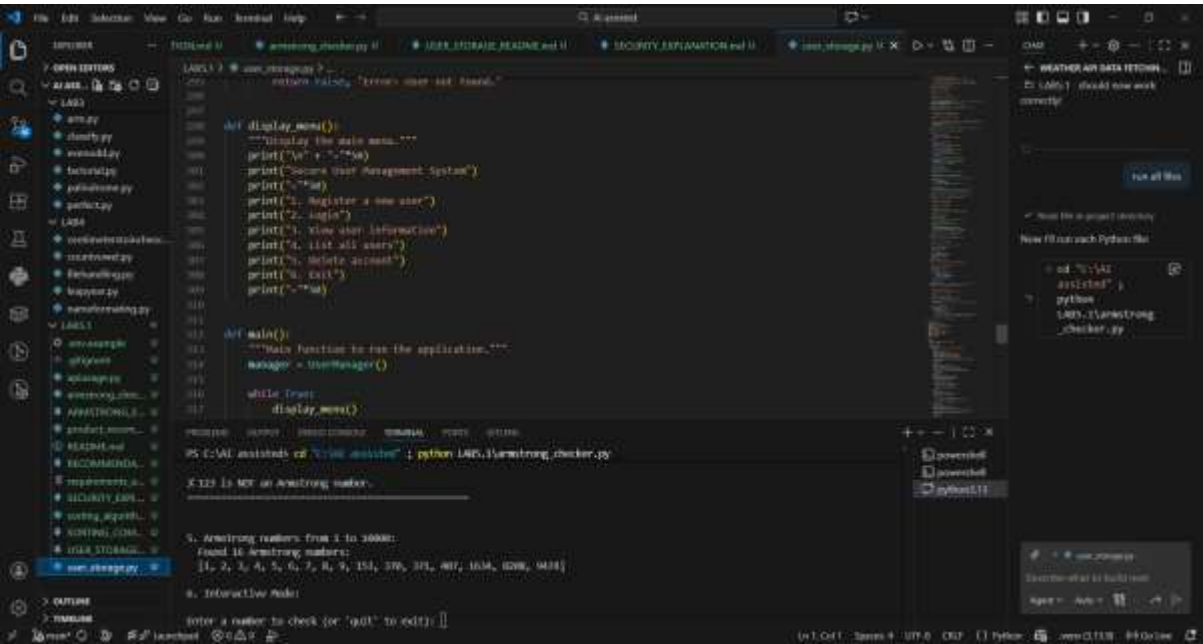
Problem Statement

Create an explainable recommendation system.

AI Prompt Used

Generate a simple product recommendation system in Python.  
For each recommended product, also provide a clear explanation of why it was recommended.  
Ensure the recommendations are explainable and transparent.

Implementation



Transparency Evaluation

Each recommendation includes a reason, making the system explainable.

Result

All tasks were implemented successfully using ethical AI coding practices with proper privacy, security, and transparency.

## **Conclusion**

AI-generated code must be reviewed by developers to ensure ethical compliance. Human accountability is essential in responsible AI usage.