

2303A51760

K. Sai Sathvik

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Task 1: Privacy in API Usage

AI Prompt Used

Generate a Python program that fetches weather data securely without exposing API keys in the code.

Output:

```
import requests

api_key = "YOUR_API_KEY"

url = f"https://api.openweathermap.org/data/2.5/weather?q=London&appid={api_key}"

response = requests.get(url)

print(response.json())
```

Improved Secure Version

```
import os

import requests

# API key is stored securely as an environment variable

api_key = os.getenv("WEATHER_API_KEY")

city = "London"

url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"

response = requests.get(url)
```

```
print(response.json())
```

Task 2: Privacy & Security in File Handling

AI Prompt Used

Generate a Python script that stores user name, email, and password securely.

Ensure the password is not stored in plain text.

Output:

```
name = input("Enter name: ")
email = input("Enter email: ")
password = input("Enter password: ")
with open("users.txt", "a") as file:
    file.write(f"{name},{email},{password}\n")
```

Revised Secure Version

```
import hashlib

name = input("Enter name: ")
email = input("Enter email: ")
password = input("Enter password: ")

# Hashing the password
hashed_password = hashlib.sha256(password.encode()).hexdigest()

with open("users.txt", "a") as file:
    file.write(f"{name},{email},{hashed_password}\n")
```

Task 3: Transparency in Algorithm Design (Armstrong Number)

AI Prompt Used

Generate a Python function to check an Armstrong number and explain the code line by line.

Transparent & Commented Code

```
def is_armstrong(num):  
    # Convert number to string  
    digits = str(num)  
    # Count number of digits  
    power = len(digits)  
    total = 0  
    # Calculate sum of digits raised to power  
    for d in digits:  
        total += int(d) ** power  
    # Check if sum equals original number  
    return total == num
```

Explanation

- Converts number to digits
- Raises each digit to power of total digits
- Sums the values
- Compares with original number

Task 4: Transparency in Algorithm Comparison

AI Prompt Used

Generate Python code for QuickSort and BubbleSort with step-by-step explanation.

Bubble Sort

```
def bubble_sort(arr):  
    n = len(arr)  
    for i in range(n):  
        for j in range(0, n - i - 1):
```

```
if arr[j] > arr[j + 1]:  
    arr[j], arr[j + 1] = arr[j + 1], arr[j]
```

Quick Sort

```
def quick_sort(arr):  
    if len(arr) <= 1:  
        return arr  
    pivot = arr[0]  
    left = [x for x in arr[1:] if x <= pivot]  
    right = [x for x in arr[1:] if x > pivot]  
    return quick_sort(left) + [pivot] + quick_sort(right)
```

Comparison

Algorithm	Time Complexity	Efficiency
-----------	-----------------	------------

Bubble Sort	$O(n^2)$	Slow
-------------	----------	------

Quick Sort	$O(n \log n)$	Fast
------------	---------------	------

Task 5: Transparency in AI Recommendations

AI Prompt Used

Generate a recommendation system that explains why each product is recommended.

Explainable Recommendation System

```
def recommend_products(user_interest):  
    products = {  
        "laptop": "Recommended because you searched for coding devices",  
        "headphones": "Recommended due to interest in music",
```

```
"books": "Recommended because you like learning resources"  
}
```

```
for product, reason in products.items():  
    print(f"{product}: {reason}")
```

```
recommend_products("technology")
```

Evaluation

- Each recommendation has a **clear reason**
- Easy for users to understand decision logic