AI-ASSISTED CODING

ASSIGNMENT-5.1

Ethical Foundations – Responsible AI Coding Practices ---------------------------------------

NAME: GATTU VARSHITH

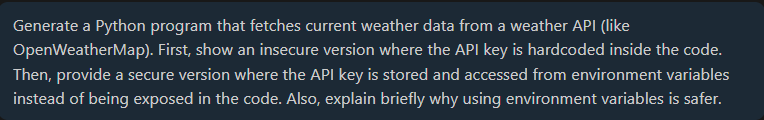
HALL-TICKET: 2403A510E3

BATCH NO: 05

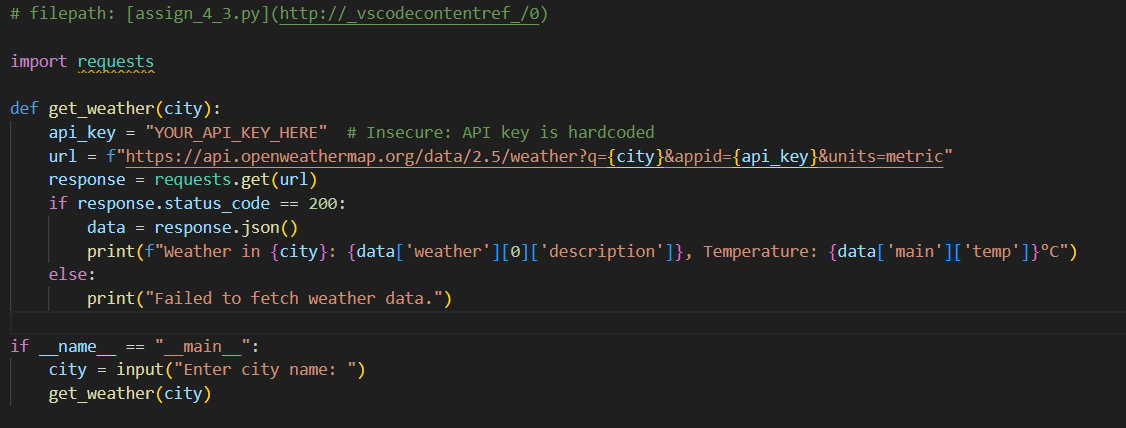
Task 1:   
Use an AI tool to generate a Python program that connects to a weather API.

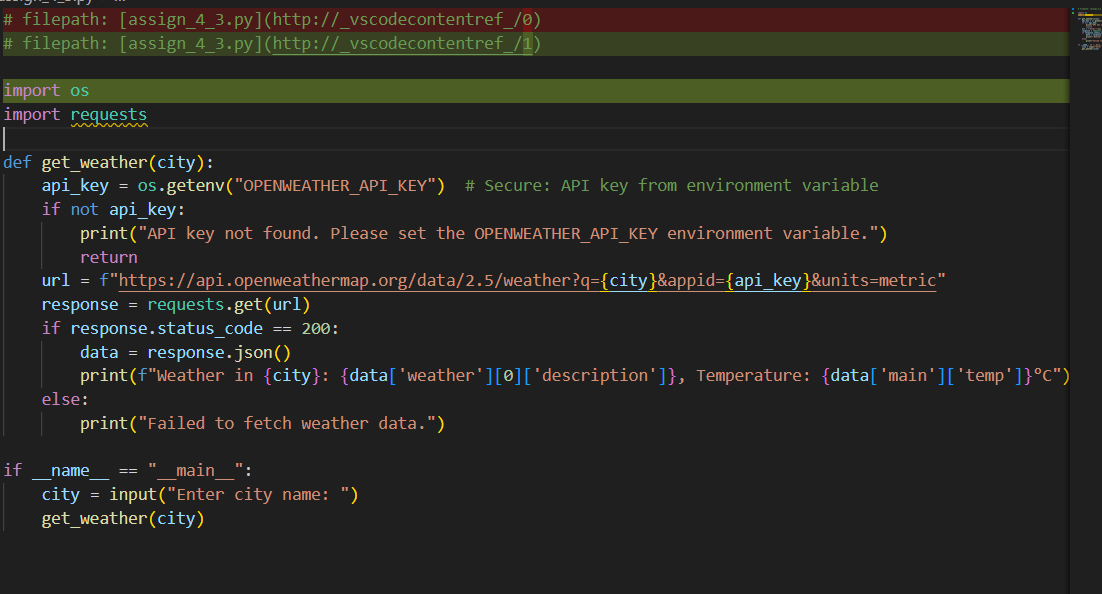
Prompt:  
*"Generate code to fetch weather data securely without exposing API keys in the code."*

Prompt:



CODE SCREENSHOT:

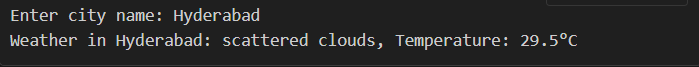


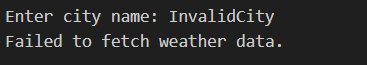






OUTPUT:





--------------------------------------------------------------------------------------------------------------

Task 2:

Use an AI tool to generate a Python script that stores user data (name, email, password) in a file.

Analyze: Check if the AI stores sensitive data in plain text or without encryption.

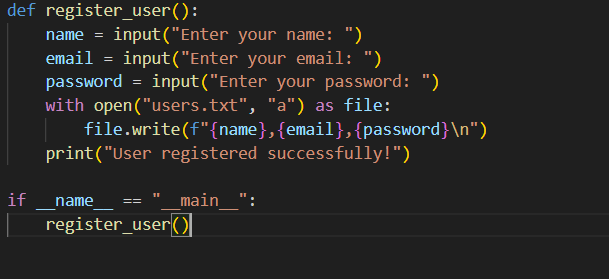
Expected Output:

* Identified privacy risks.
* Revised version with encrypted password storage (e.g., hashing).

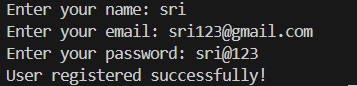
Prompt:



CODE SCREENSHOT:



Output:



**IN the file:**

* **Plain-text password storage**: Passwords are stored exactly as entered.
* **No hashing or encryption**: A data breach would expose user passwords.
* **Vulnerable to insider threats**: Anyone with access to the file can read sensitive data.

----------------------------------------------------------------------------------------------------------

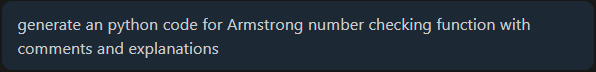
Task 3:

Use AI to generate an Armstrong number checking function with comments and explanations.

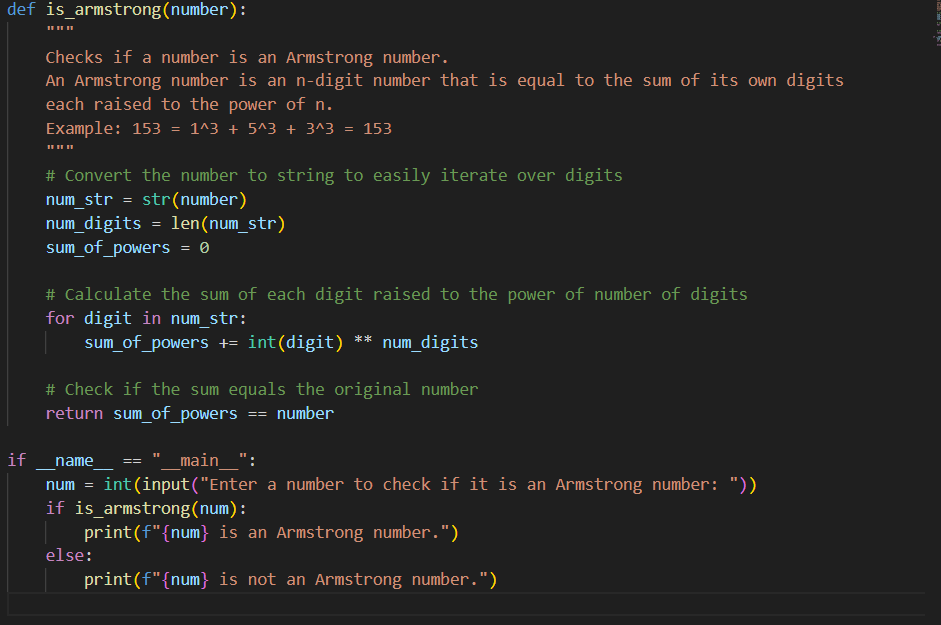
Instructions:

1. Ask AI to explain the code line-by-line.
2. Compare the explanation with code functionality.

PROMPT:



CODE:



OUTPUT:



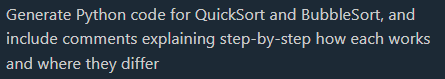
------------------------------------------------------------------------------------------------------------------------------------

TASK 4:

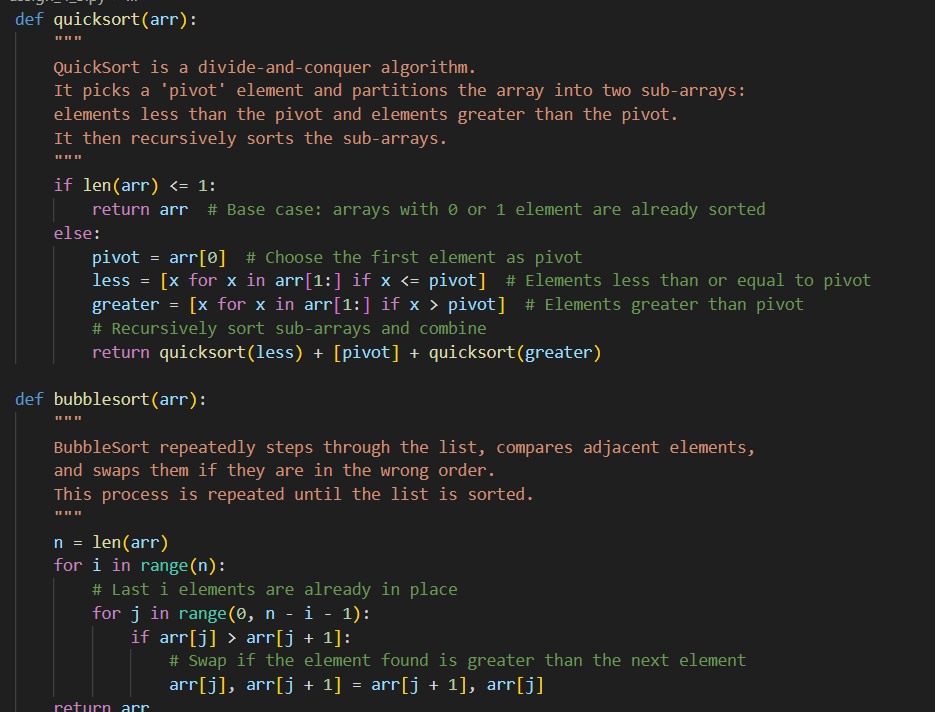
Use AI to implement two sorting algorithms (e.g., QuickSort and BubbleSort).

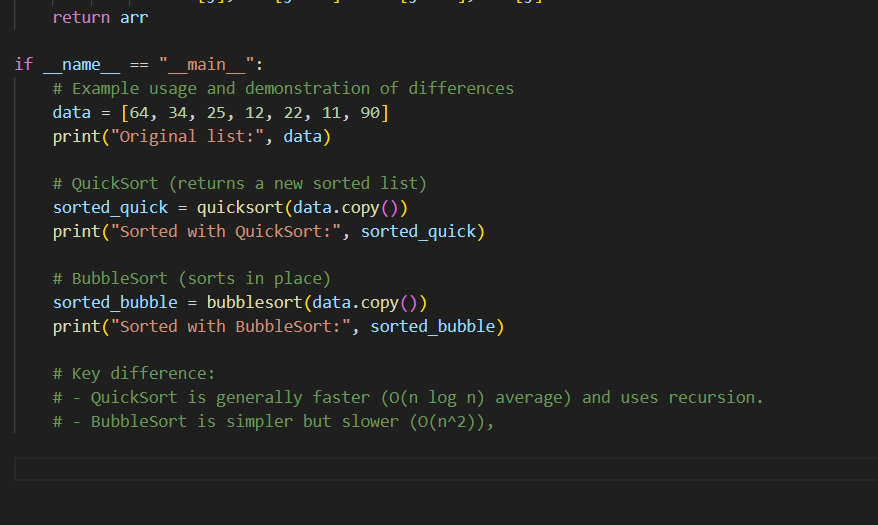
Prompt:  
*"Generate Python code for QuickSort and BubbleSort, and include comments explaining step-by-step how each works and where they differ."*

PROMPT:

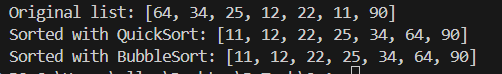


CODE:





OUTPUT:



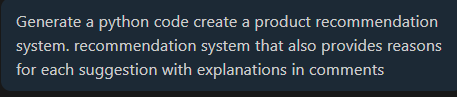
--------------------------------------------------------------------------------------------------------------

TASK 5:

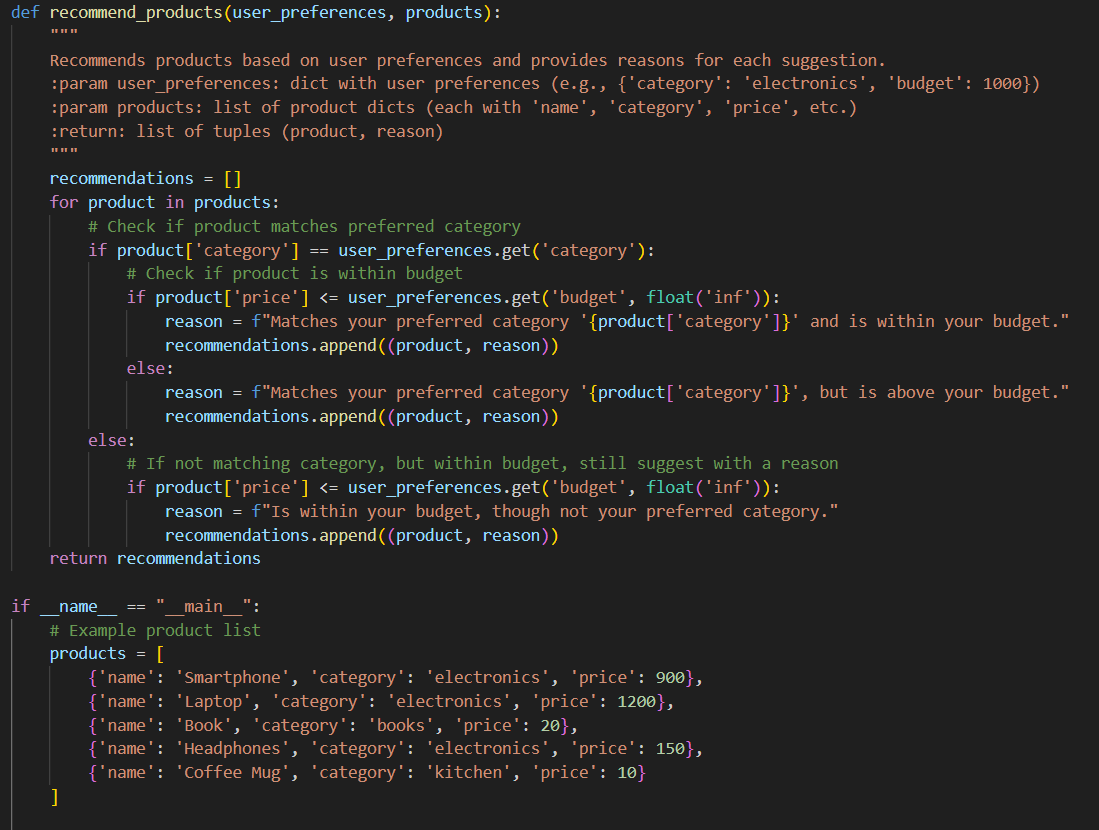
: Use AI to create a product recommendation system.

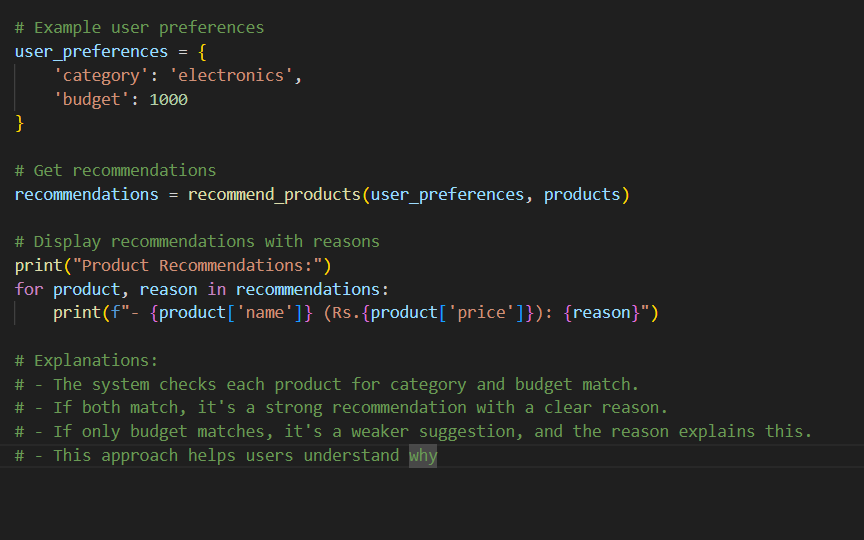
Prompt:  
*"Generate a recommendation system that also provides reasons for each suggestion."*

**Prompt:**

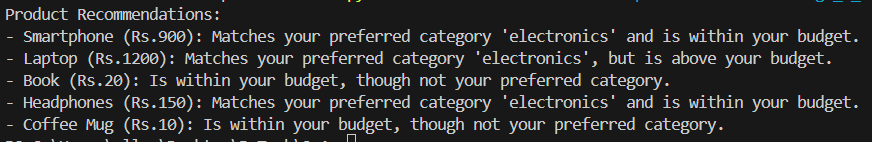
****

CODE :





Output:



TASK 6:

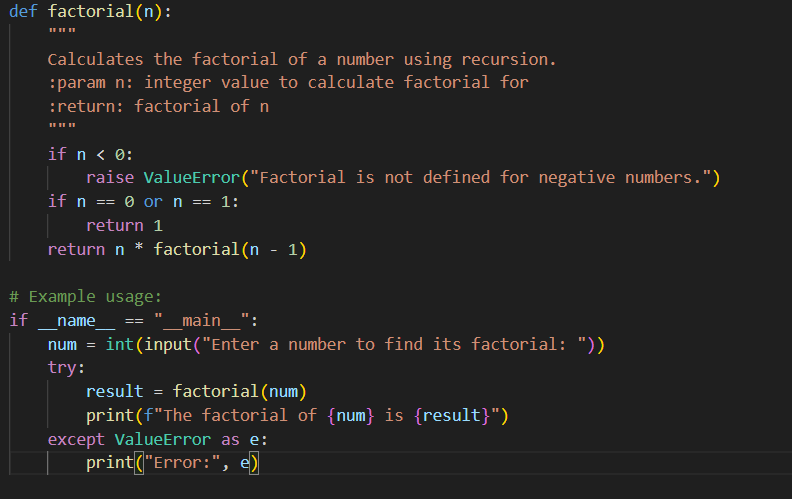
*Ask AI to generate a Python function for calculating factorial using recursion.*

*Prompt:  
"Generate a recursive factorial function with comments that explain each line and a final summary of the algorithm’s flow."*

**Prompt:**



CODE :

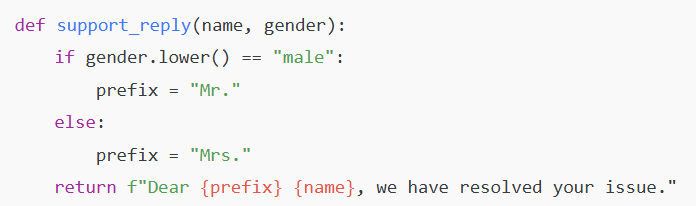


Output:



TASK 7:

*Code Snippet:*

**

*Task:*

*Regenerate the code so that support messages use neutral language (e.g., “Dear {name}”) and optionally accept preferred titles.*

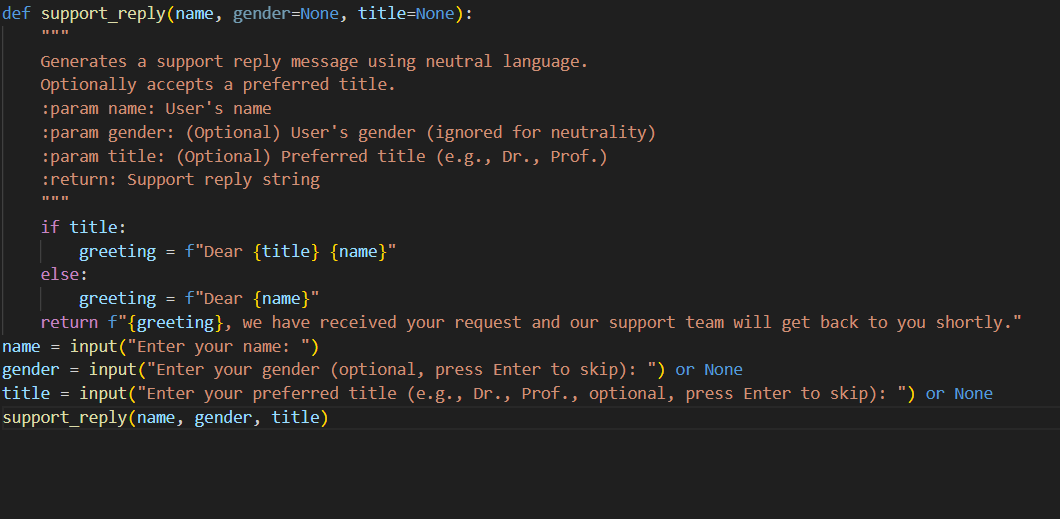
*Expected Output:*

* *Neutral, user-friendly support responses.*

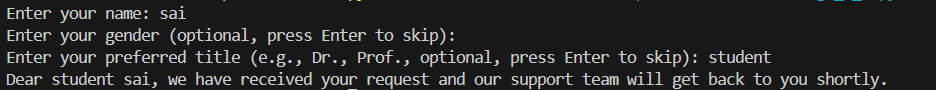
**Prompt:**



CODE :



Output:

****

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots