ASSIGNMENT -6.5

NAME: HABEEBA KHANAM

HALLTICKET NO: 2303A51474

BATCH:29

EXPERIMENT 6: AI-BASED CODE COMPLETION: WORKING WITH SUGGESTIONS FOR CLASSES, LOOPS, CONDITIONALS

TASK DESCRIPTION #1

(AI-Based Code Completion for Conditional Eligibility Check)

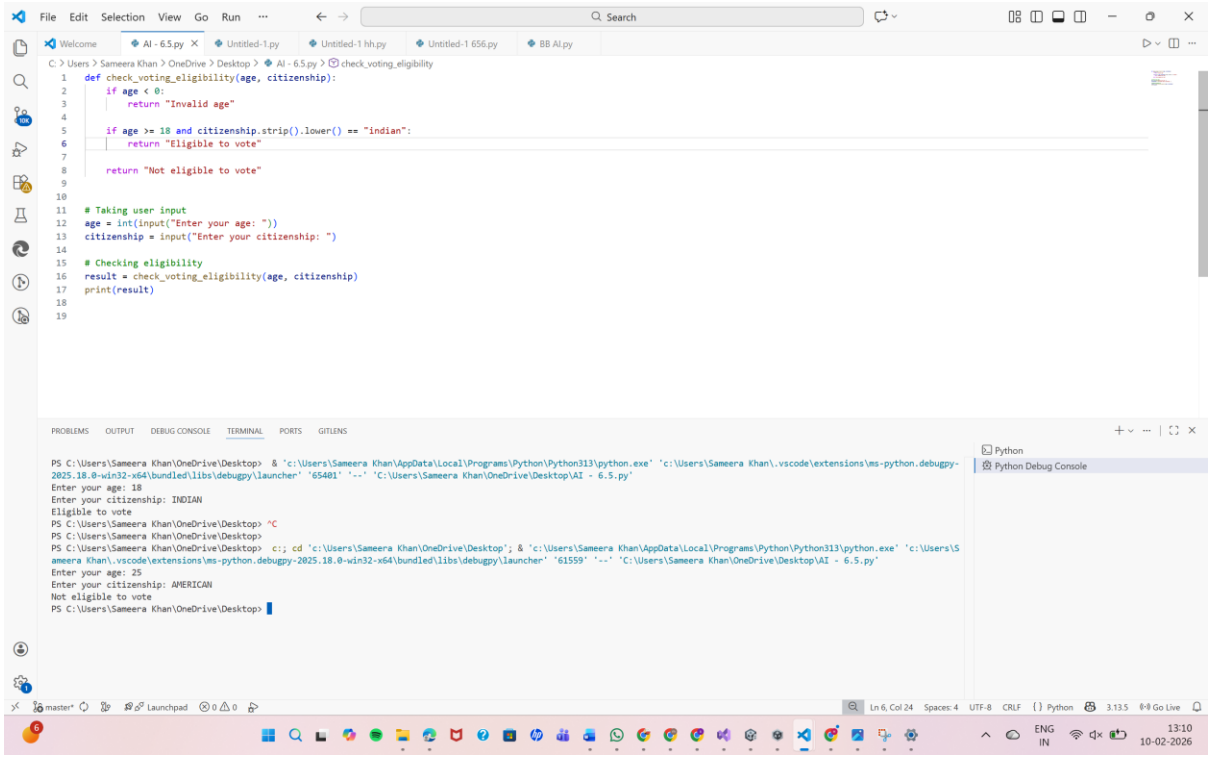
TASK:

Use an AI tool to generate eligibility logic.

PROMPT:

“Generate Python code to check voting eligibility based on age and
citizenship.”

CODE:



OBSERVATION:

- The program ensures that **both conditions must be satisfied** (age ≥ 18 and valid citizenship) before declaring a user eligible to vote.
- Input handling is simple and user-friendly, allowing easy modification for additional eligibility rules.
- The logic flow is clear, readable, and follows standard Python programming practices.
- AI-assisted code completion significantly **reduces development time** while maintaining logical correctness.

TASK DESCRIPTION #2

(AI-Based Code Completion for Loop-Based String Processing)

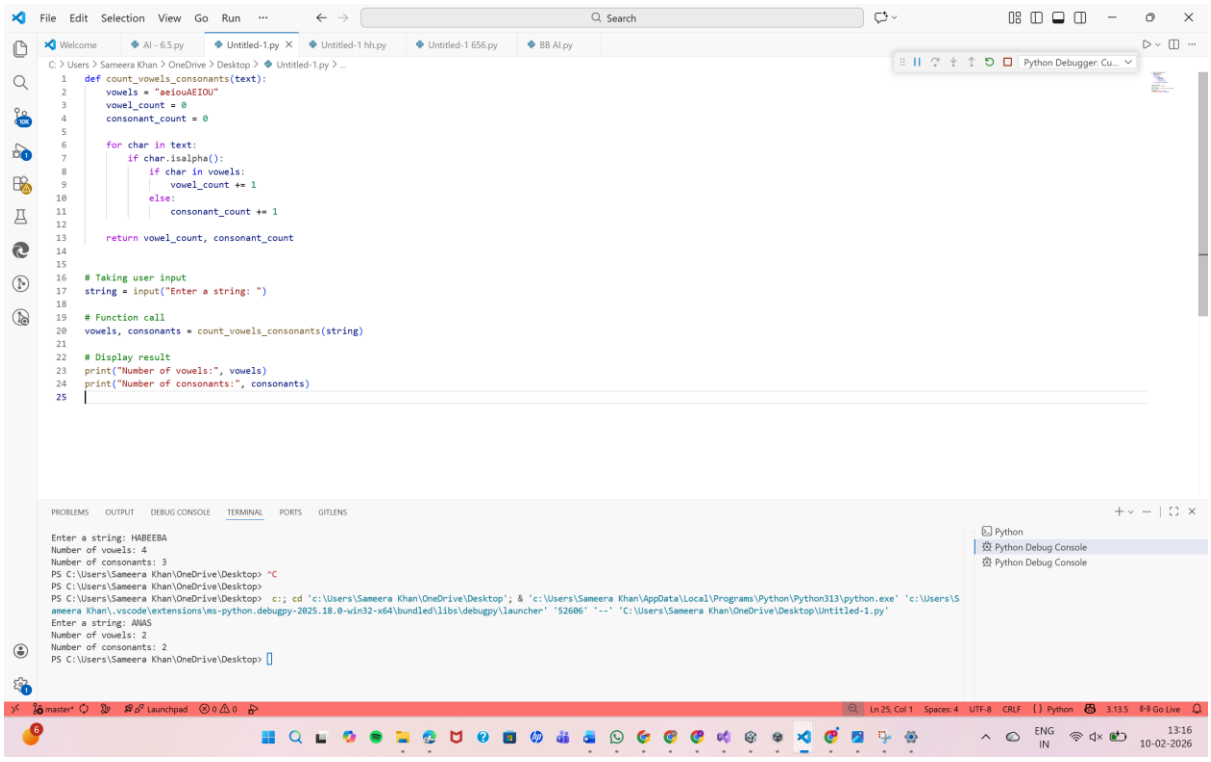
TASK:

Use an AI tool to process strings using loops.

PROMPT:

"Generate Python code to count vowels and consonants in a string using a loop."

CODE:



OBSERBVATION:

- The AI-generated code correctly uses **loops and conditionals** for string processing.
- Non-alphabetic characters are ignored, improving accuracy.
- The logic is efficient, readable, and easy to extend.

TASK DESCRIPTION #3

(AI-Assisted Code Completion Reflection
Task)

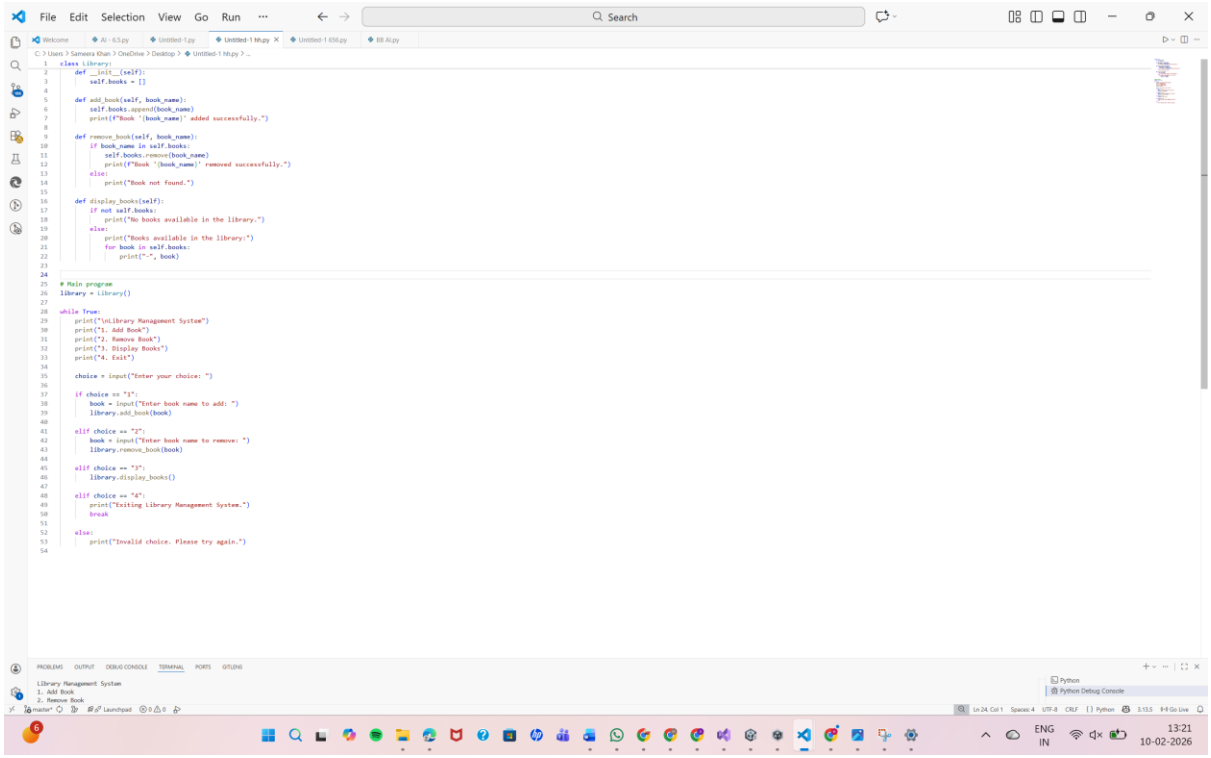
TASK:

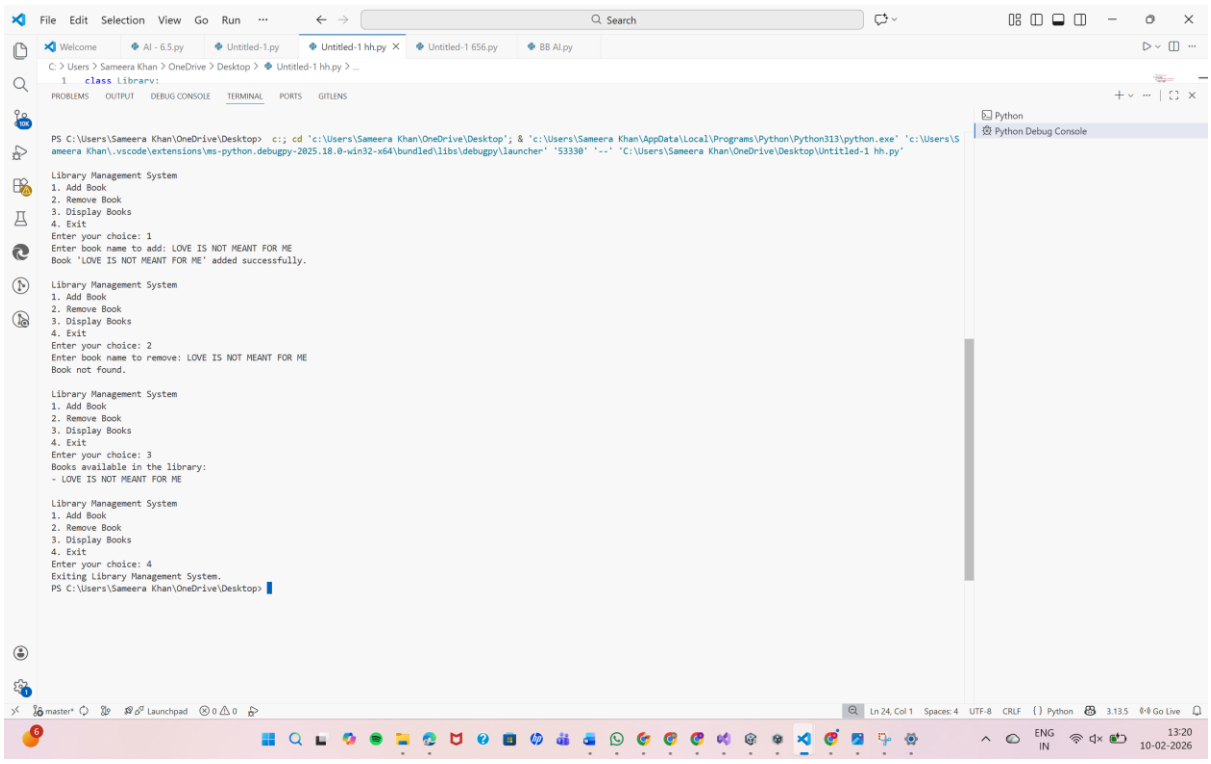
Use an AI tool to generate a complete program using classes,
loops, and conditionals.

PROMPT:

“Generate a Python program for a library management system
using classes, loops, and conditional statements.”

CODE:





OBSERAVTION:

- The AI-generated program correctly integrates **classes, loops, and conditional logic**.
- The menu-driven structure improves usability and clarity.

- The code is readable, modular, and easy to extend with features like book search or file storage.

TASK DESCRIPTION #4

(AI-Assisted Code Completion for Class-Based Attendance System)

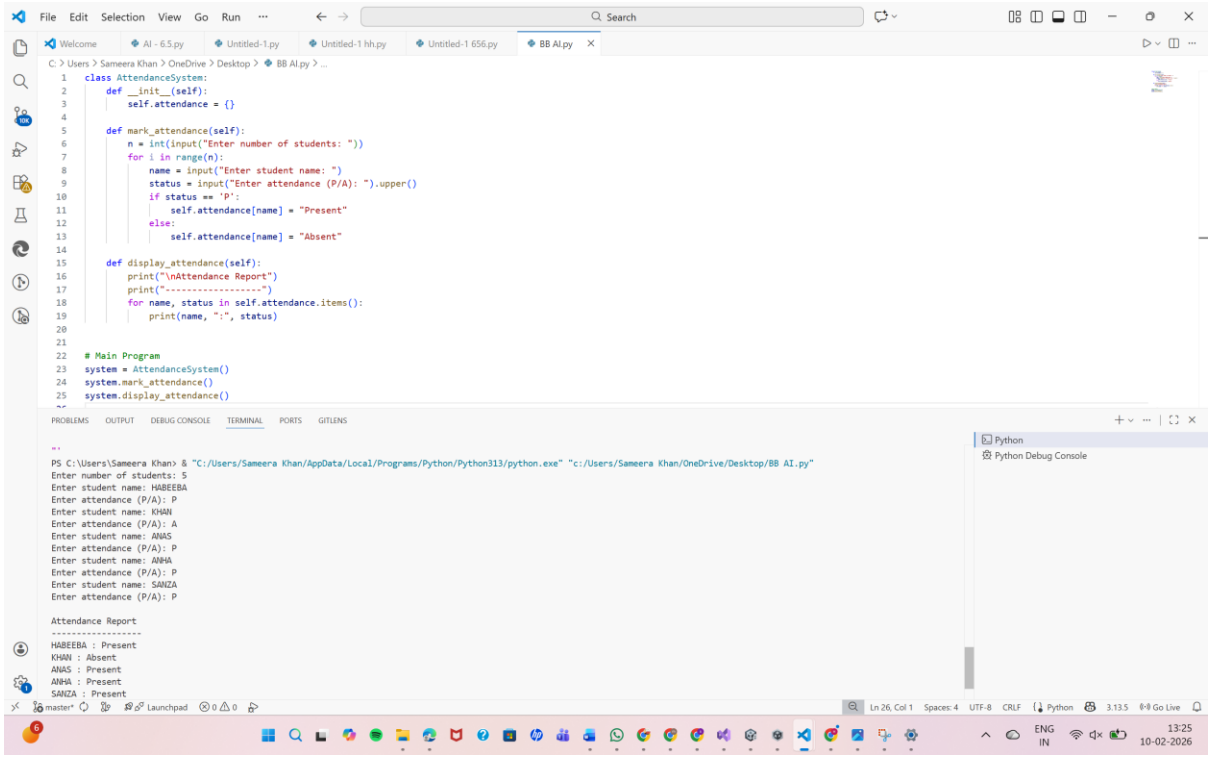
TASK:

Use an AI tool to generate an attendance management class.

PROMPT:

“Generate a Python class to mark and display student
attendance using loops.”

CODE:



OBSERVATION:

- The AI-generated solution correctly uses a **class-based structure** to manage student attendance data.
- **Loops** are effectively applied to record attendance for multiple students and to display stored records.
- **Conditional statements** accurately distinguish between present and absent students.
- The program logic is simple, readable, and modular, making it easy to extend for additional features such as attendance percentage or file storage.

TASK DESCRIPTION #5

(AI-Based Code Completion for Conditional Menu Navigation)

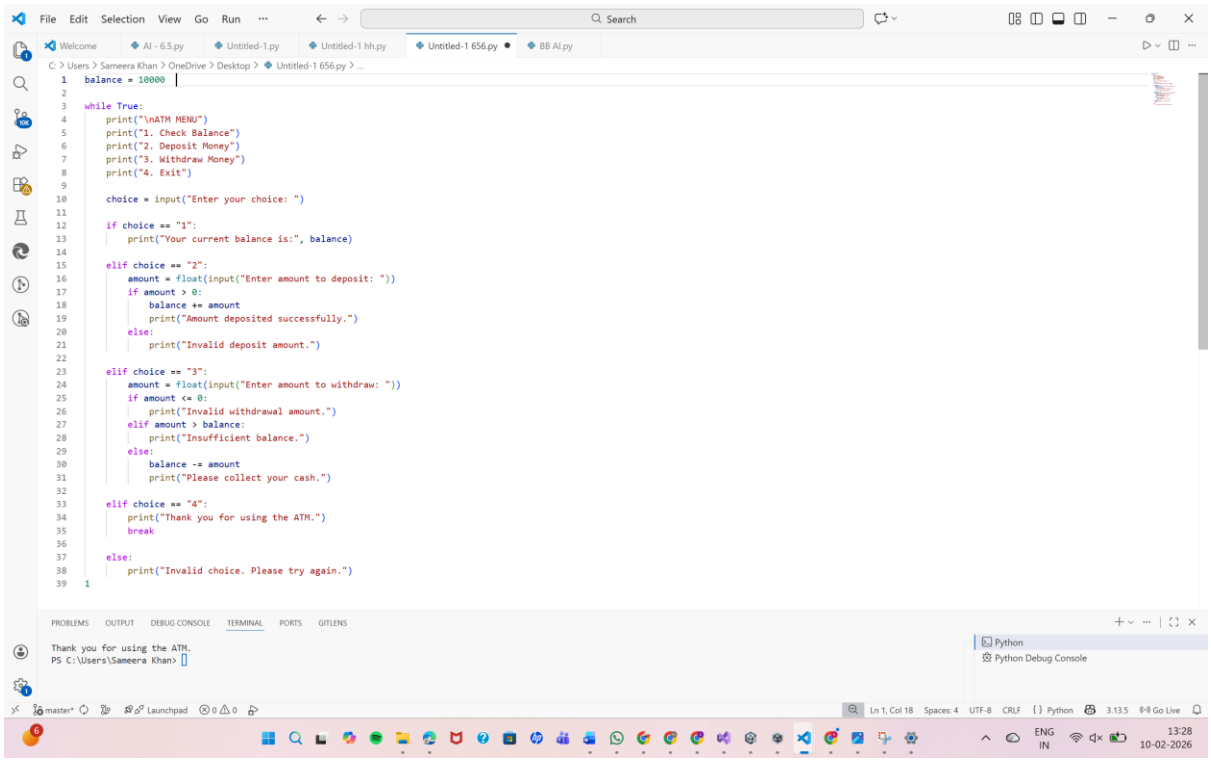
TASK:

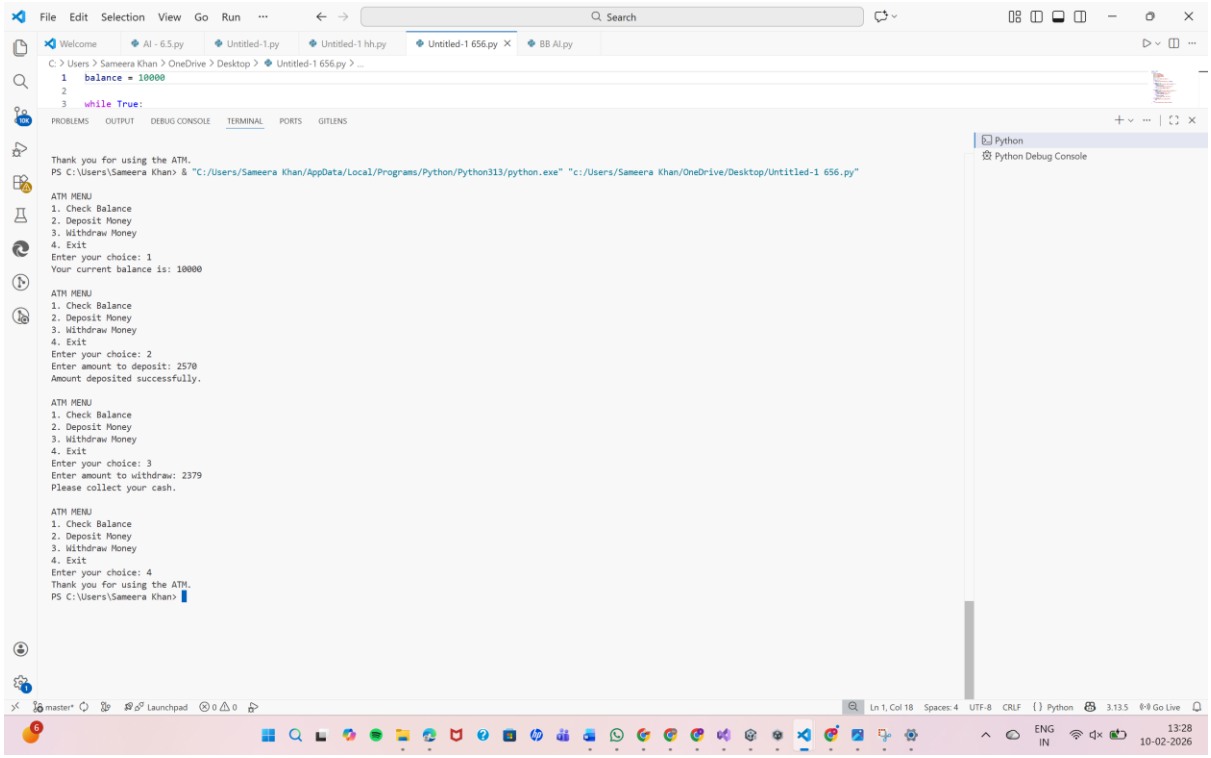
Use an AI tool to complete a navigation menu.

PROMPT:

“Generate a Python program using loops and conditionals
to simulate an ATM menu.”

CODE:





OBSERVATION:

- The AI-generated program correctly implements a **menu-driven system** using loops and conditional statements.
- The **loop structure** allows continuous menu navigation until the user chooses to exit.
- **Conditional logic (if–elif–else)** ensures accurate execution of ATM operations based on user selection.