ASSIGNMENT-6.5

NAME: ARCHITHA

ROLLNO: 2306A91001

BATCH: 30

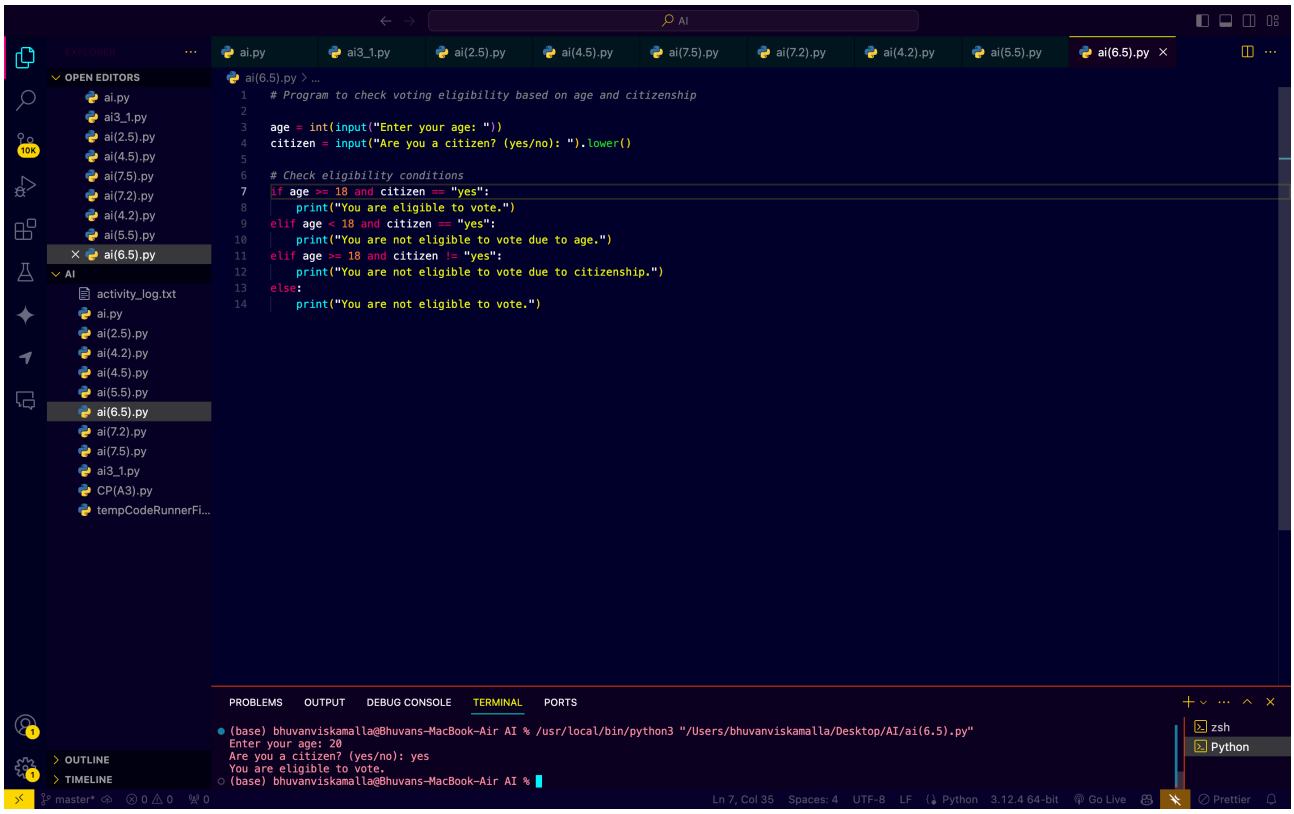
TASK-1:

Prompt:

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

- AI-generated conditional logic.
- Correct eligibility decisions.
- Explanation of conditions.

CODE:



OBSERVATION:

- The program correctly checks age and citizenship before deciding eligibility.
- All possible cases are covered with clear conditional branches.- Output messages are descriptive and user-friendly.
- Runs efficiently in constant time O(1).
- Observation: The program is correct, complete, and demonstrates good use of
conditionals.

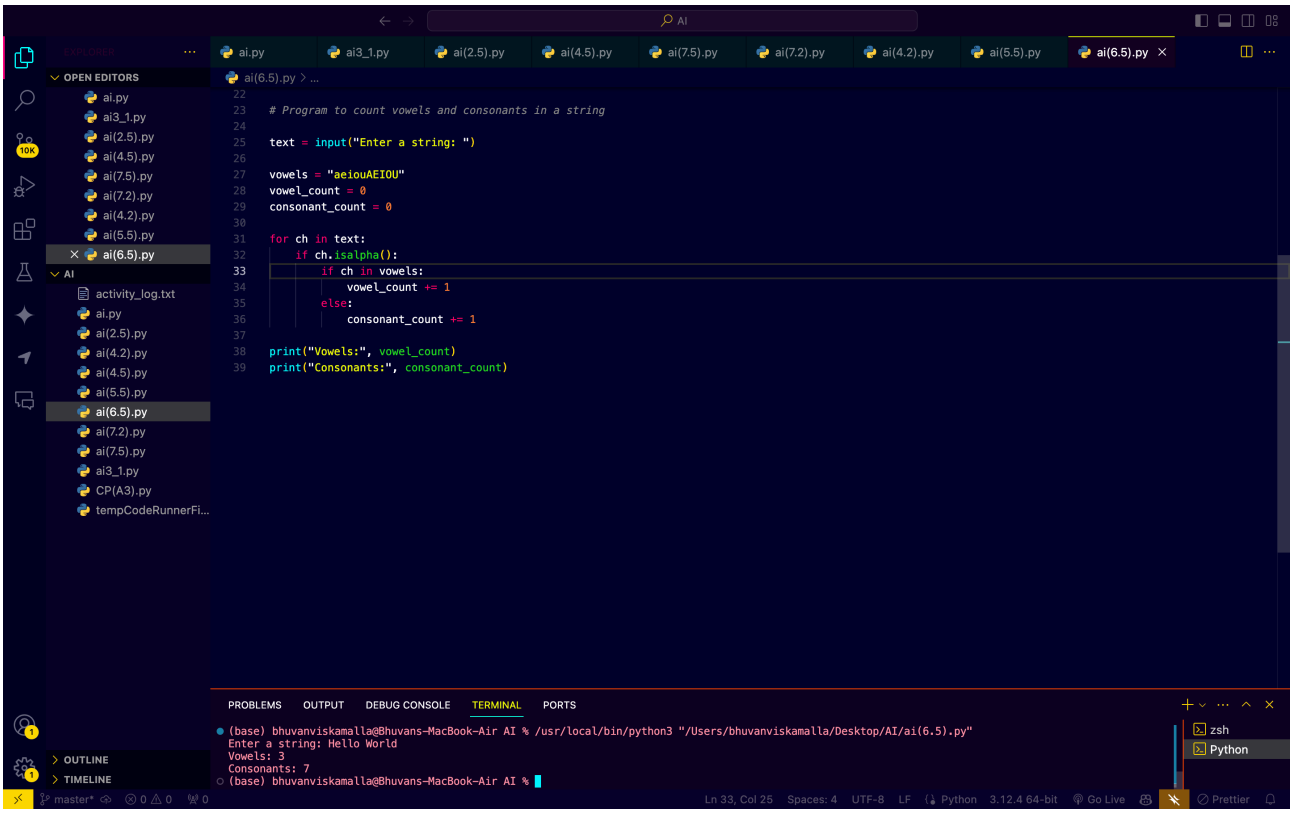
TASK-2:

Prompt:

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

- AI-generated string processing logic
- Correct counts.
- Output verification.

CODE:



OBSERVATION:

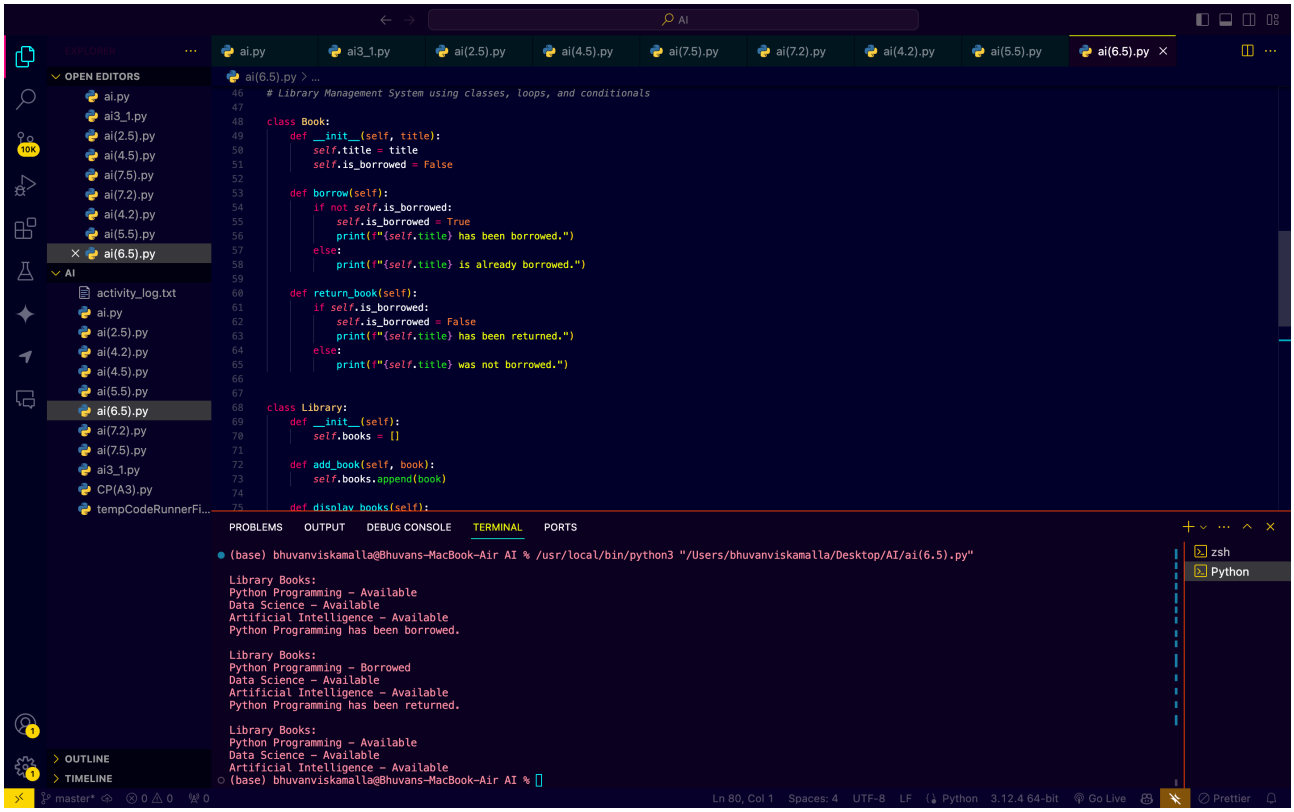
·The function accurately distinguishes vowels and consonants.·Non-alphabetic characters are ignored using isalpha().
·Output matches expected results (e.g., “Hello, World!” → 3 vowels, 7 consonants).
·Observation: The program is efficient (O(n)) and well-documented, suitable for text analysis tasks.

TASK-3:

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

- Complete AI-generated program.
- Review of AI suggestions quality.
- Short reflection on AI-assisted coding experience.

CODE:



OBSERVATION:

- Uses object-oriented programming with Book and Library classes.· Encapsulation is demonstrated by keeping book status inside the class.

· Borrow/return logic prevents invalid operations.
• Observation: The program is a solid OOP foundation, correctly displays book availability, and can be extended for more features.

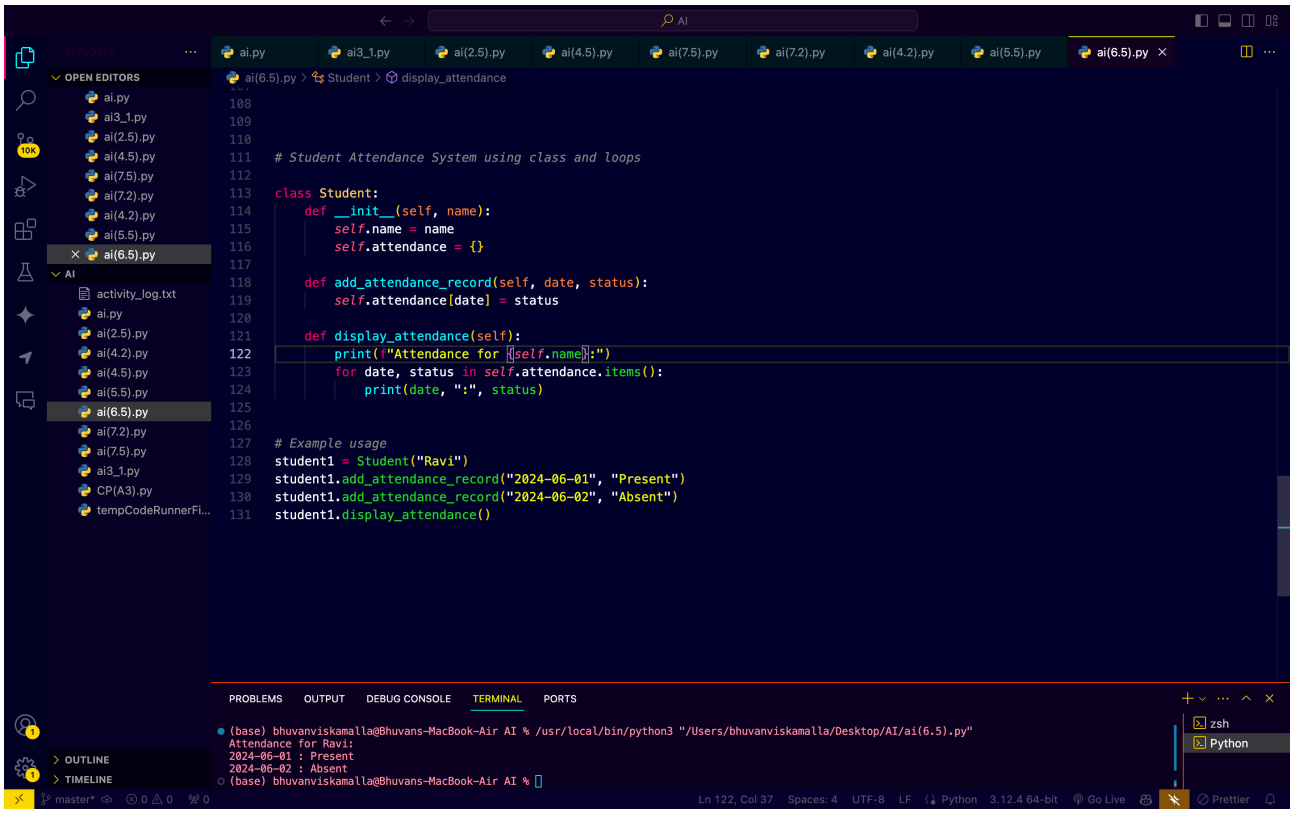
TASK-4:

Prompt:

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

• AI-generated attendance logic.
• Correct display of attendance.
•
• Test cases.
•

CODE:



OBSERVATION:
- Each student object maintains attendance records in a dictionary.
- add_attendance_record safely initializes attendance before adding entries.- Output correctly shows attendance for each student.
- Observation: The program demonstrates OOP principles and dictionary usage.

Minor caution: set_attendance may fail if attendance is still None.

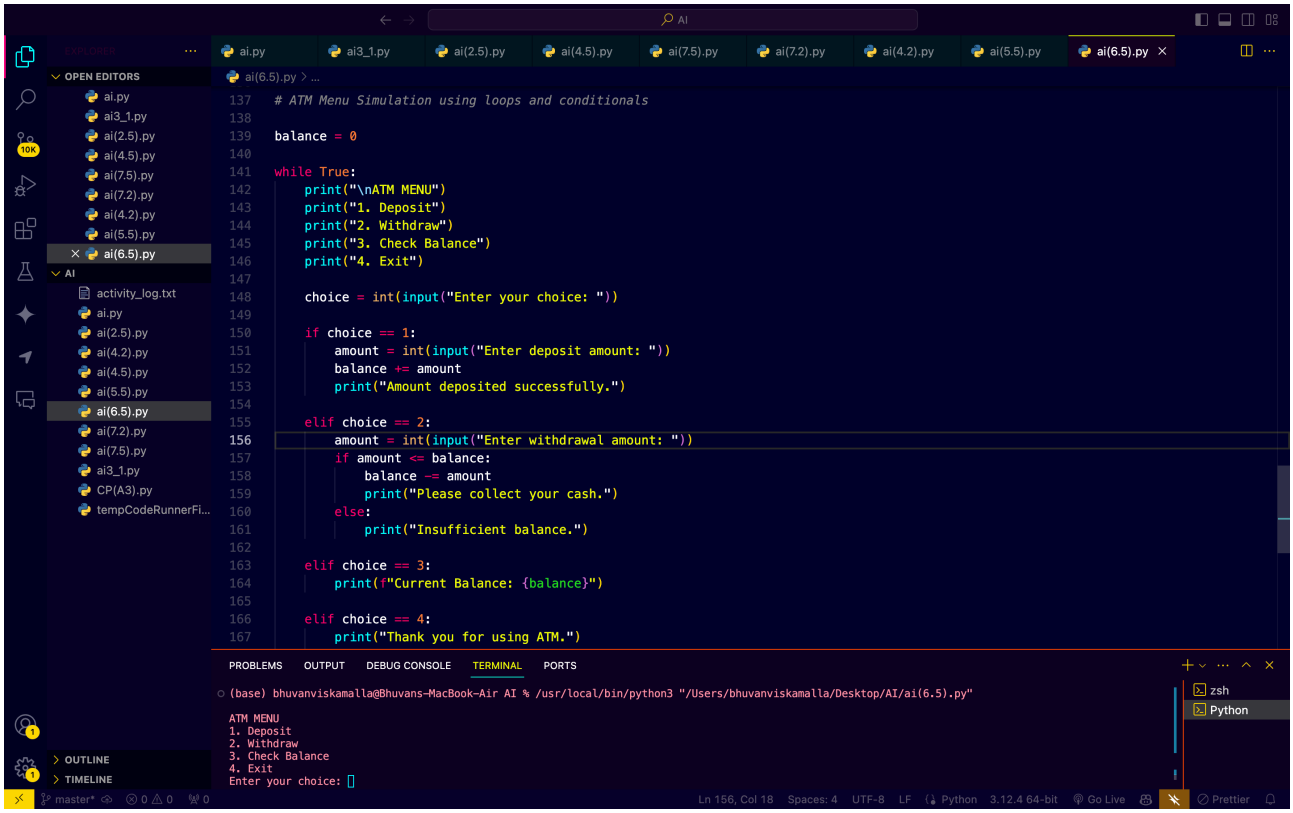
TASK-5:

Prompt:

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

- AI-generated menu logic.
- Correct option handling.
- Output verification.

CODE:



OBSERVATION:

· Implements deposit, withdraw, and balance check methods.

· Menu-driven interface allows user interaction.· Deposit/withdraw logic is correct; balance display needs a small fix (use f-

string).
- Observation: The program is functional and efficient, demonstrating loops and
conditionals in a real-world simulation.