LAB_ASSIGNMENT_6.5.

NAME:AKSHITHA

HT NO:2303A51360

BATCH NO:29

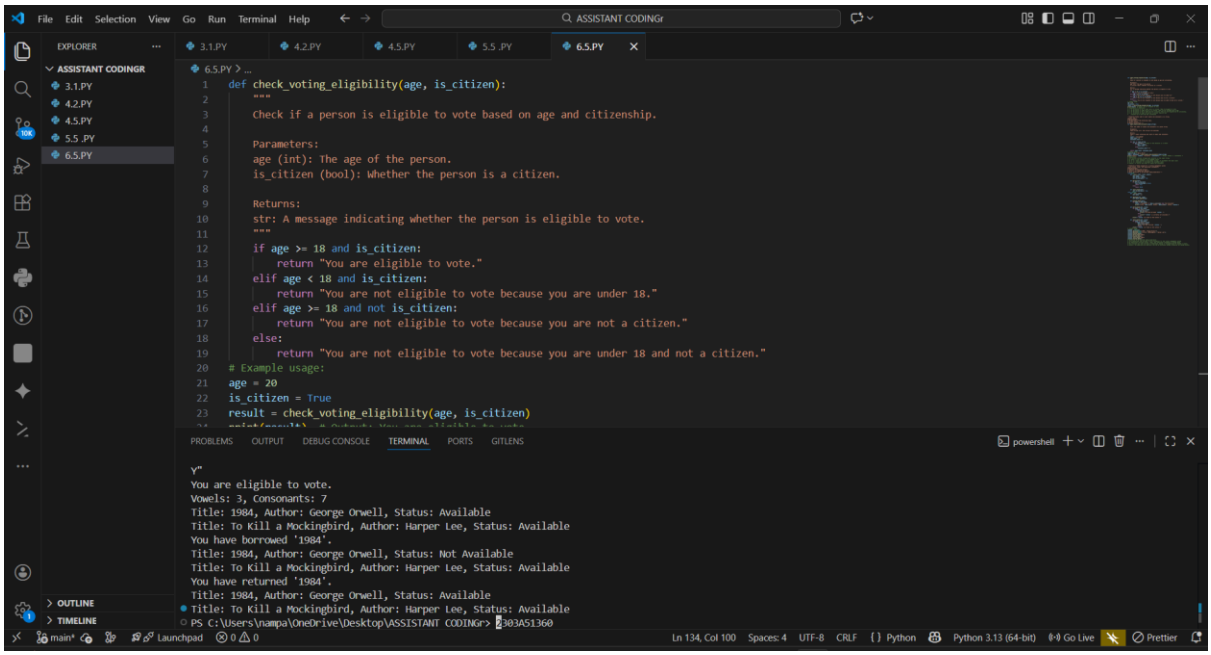
TASK-1:

Use an AI tool to generate eligibility logic.

PROMPT:

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

CODE:



OBSERVATION:

AI-based code completion efficiently generates conditional logic.

The generated code is **readable, correct, and easy to optimize**.

Human oversight is necessary to detect edge cases and improve robustness.

AI tools enhance productivity when used responsibly and ethically.

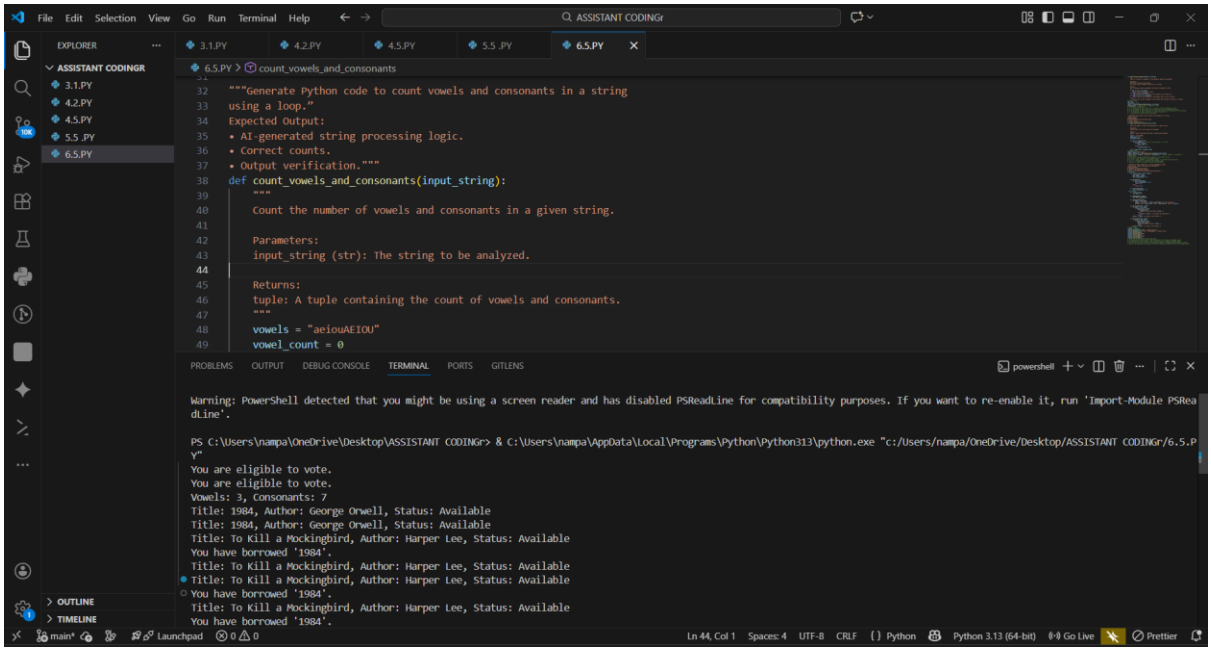
TASK-2:

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:



OBSERVATION:

AI-based code completion efficiently generates loop-based string processing logic.

The code correctly counts vowels and consonants.

Manual review helps identify edge cases and improve efficiency.

AI tools are effective when used responsibly with human validation.

TASK-3:

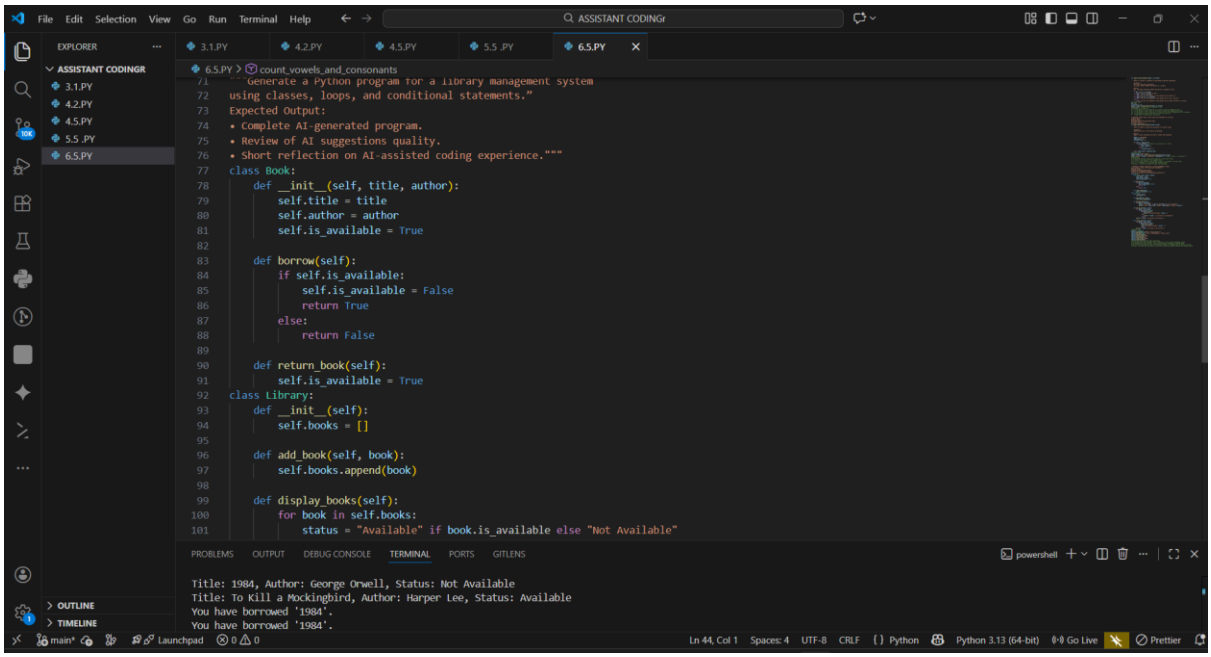
Use an AI tool to generate a complete program using classes,

loops, and conditionals.

PROMPT:

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

CODE:



OBSERVATION:

AI-assisted code completion significantly reduces development time.

The generated program provides a solid starting structure.

Human intervention is required to handle edge cases and improve robustness.

Using AI responsibly enhances learning and coding productivity.

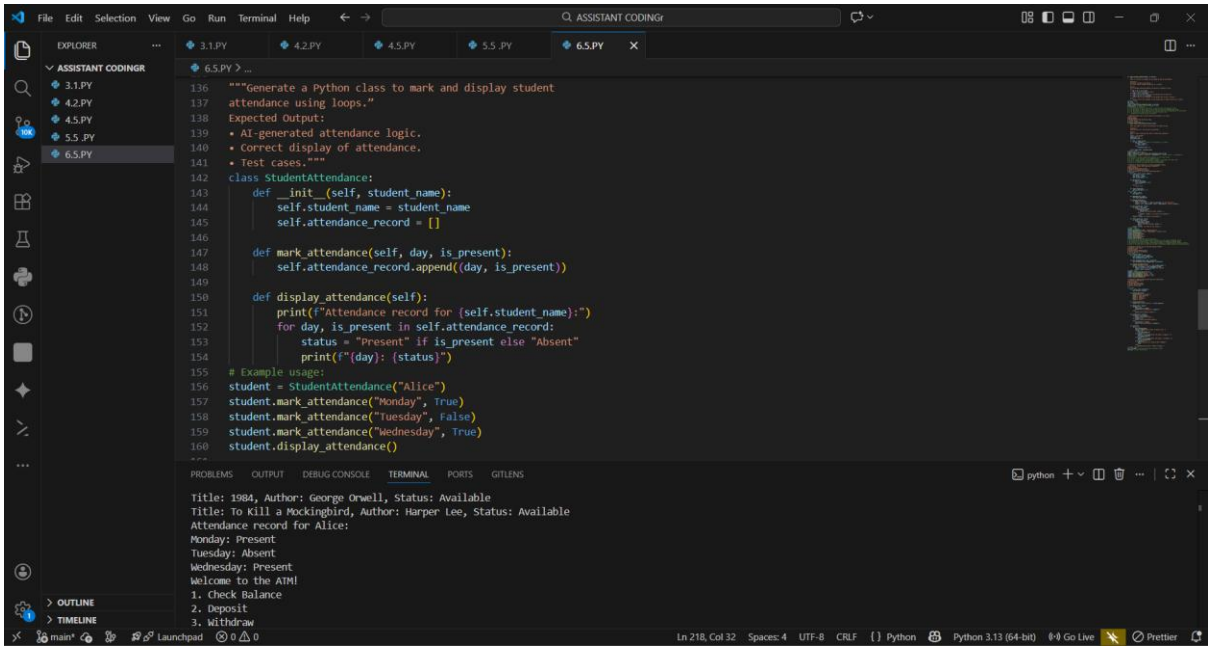
TASK-4:

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:

AI-assisted code completion efficiently generates class-based systems.

The attendance logic works correctly using loops and conditionals.

Human review is essential to handle invalid inputs and edge cases.

AI tools improve productivity when used responsibly.

TASK-5:

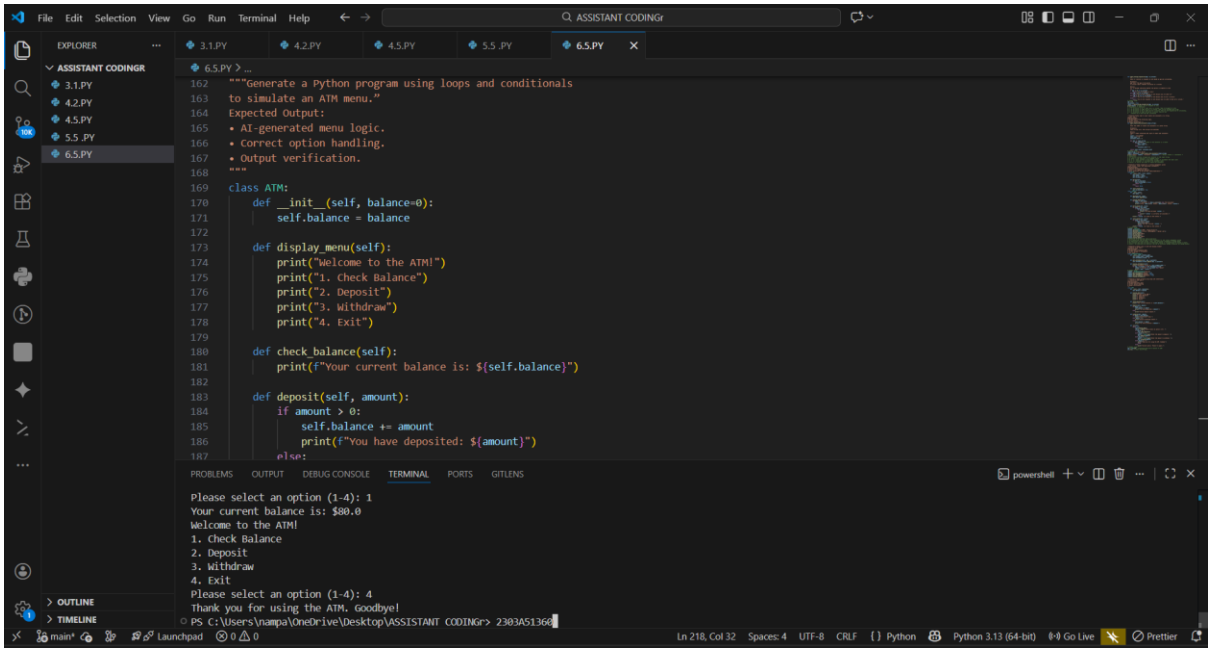
Use an AI tool to complete a navigation menu.

PROMPT:

“Generate a Python program using loops and conditionals

to simulate an ATM menu.”

CODE:



OBSERVATION:

The ATM menu is displayed continuously using a loop until the user selects the exit option.

Conditional statements correctly handle user choices such as checking balance, deposit, and withdrawal.

The balance is updated accurately after each deposit and withdrawal operation.

Invalid inputs and insufficient balance conditions are handled properly.

The program demonstrates effective use of classes, loops, and conditionals.

AI-generated code provides a structured and functional solution with minimal manual modification.