AI ASSISTED CODING

2303A52454

B-34

LAB-6.5

Task Description #1 (AI-Based Code Completion for Conditional Eligibility Check)

Task: Use an AI tool to generate eligibility logic.

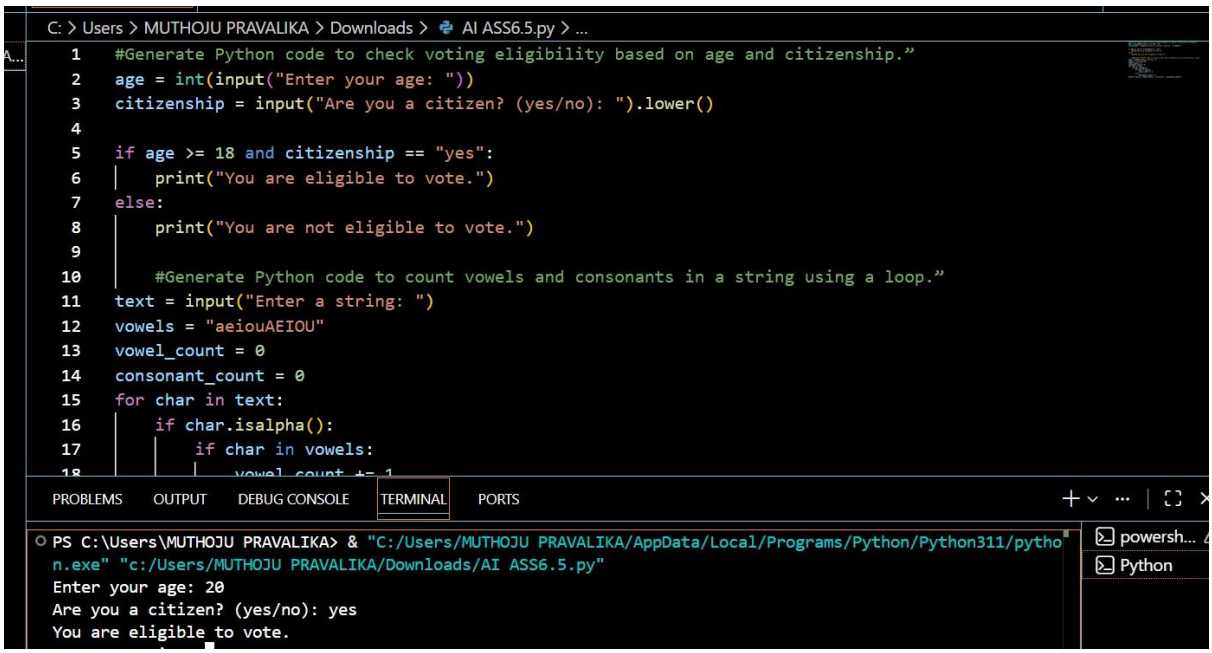
Prompt:

“Generate Python code to check voting eligibility based on

age and citizenship.”

Expected Output:

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



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

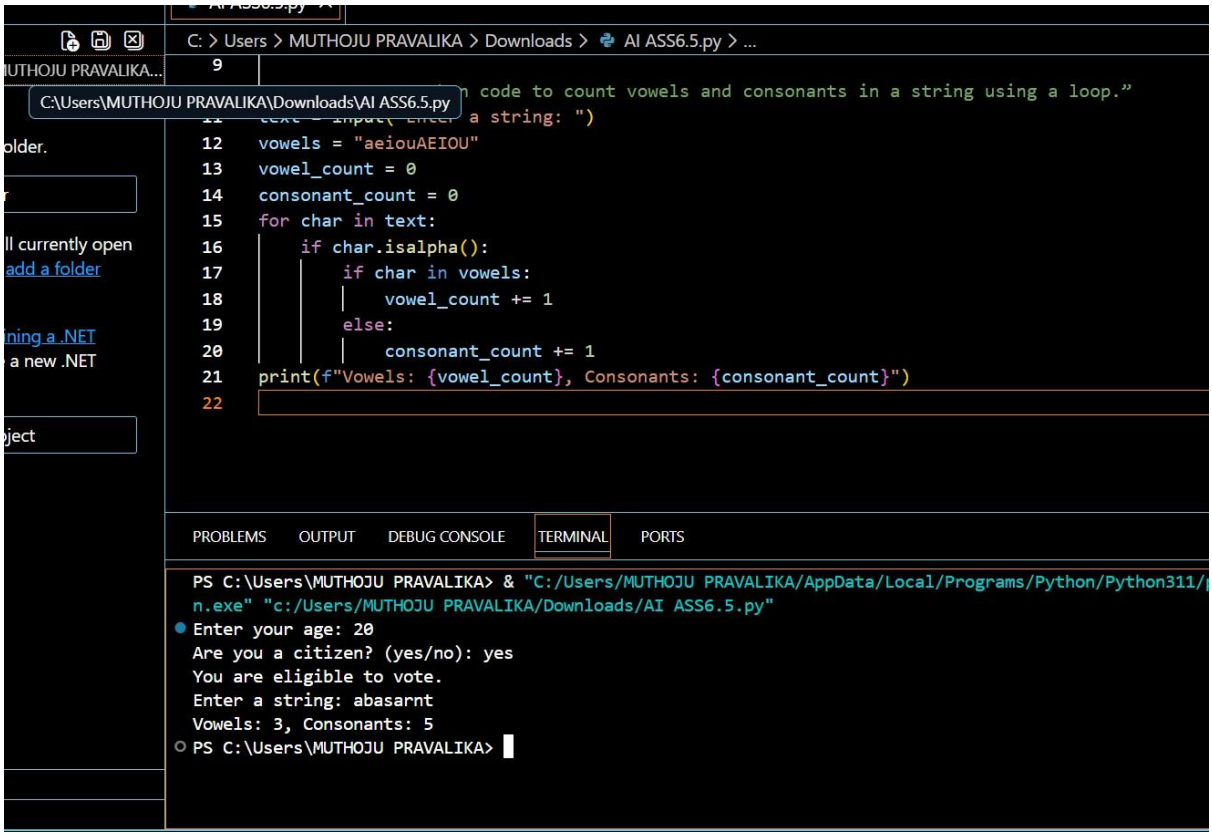
using a loop.”

Expected Output:

• AI-generated string processing logic.

• Correct counts.

• Output verification.



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

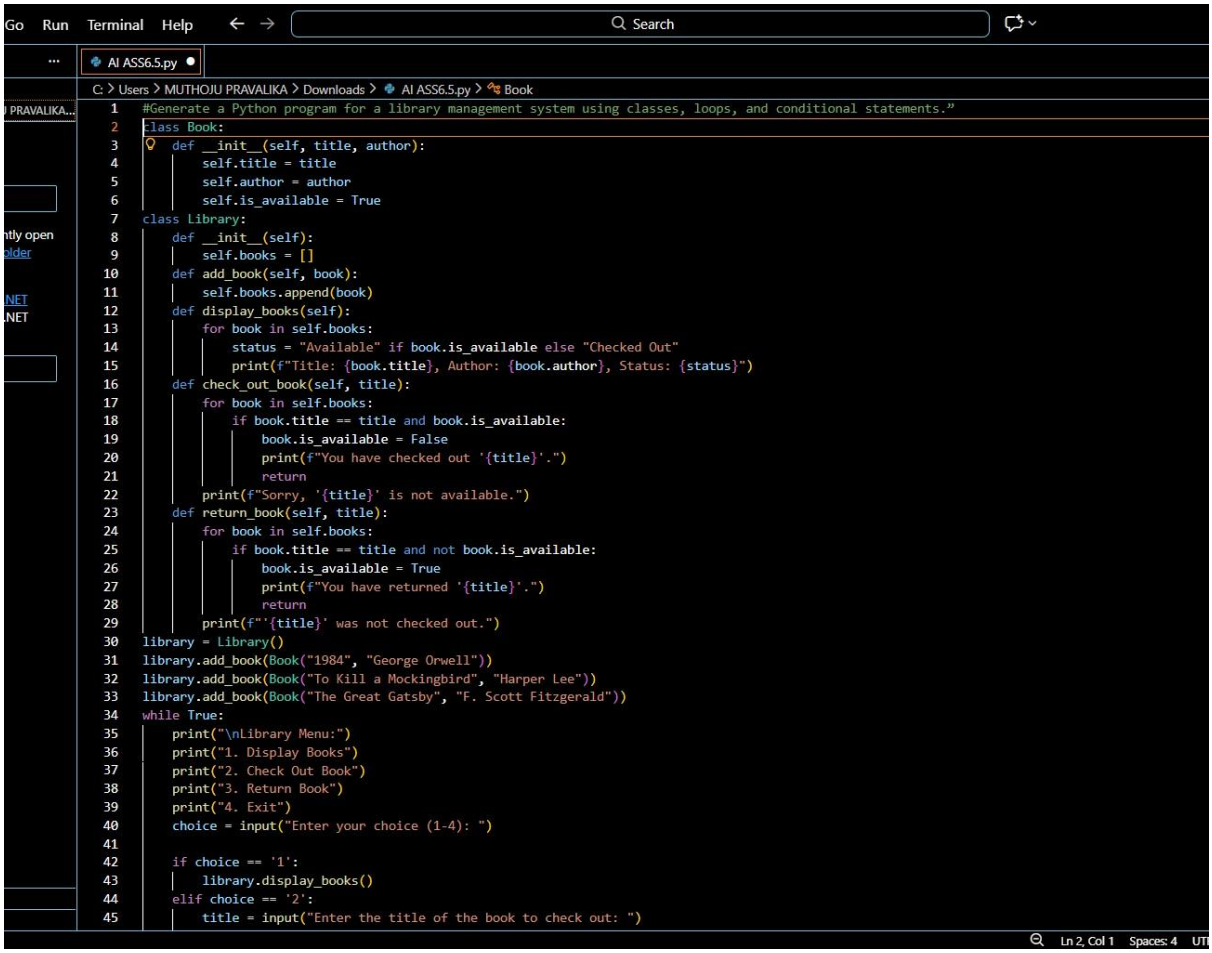
classes, loops, and conditional statements.”

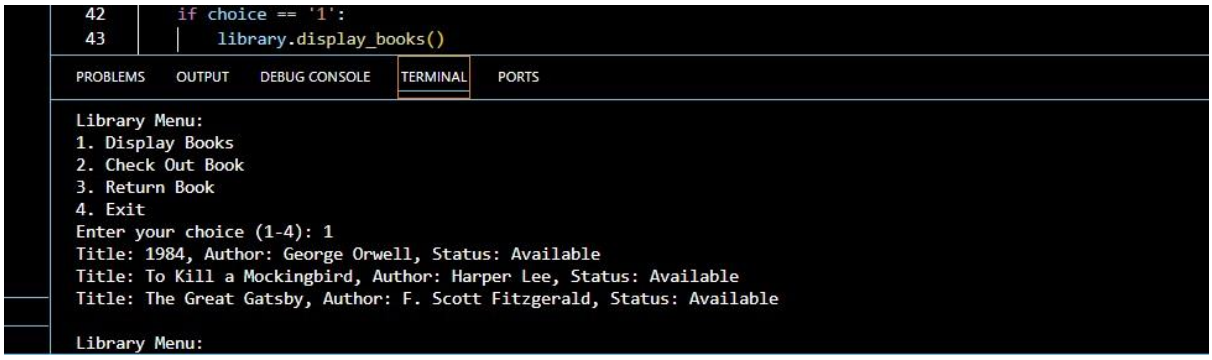
Expected Output:

• Complete AI-generated program.

• Review of AI suggestions quality.

• Short reflection on AI-assisted coding experience.





Task Description #4 (AI-Assisted Code Completion for Class-Based Attendance System)

Task: Use an AI tool to generate an attendance management class.

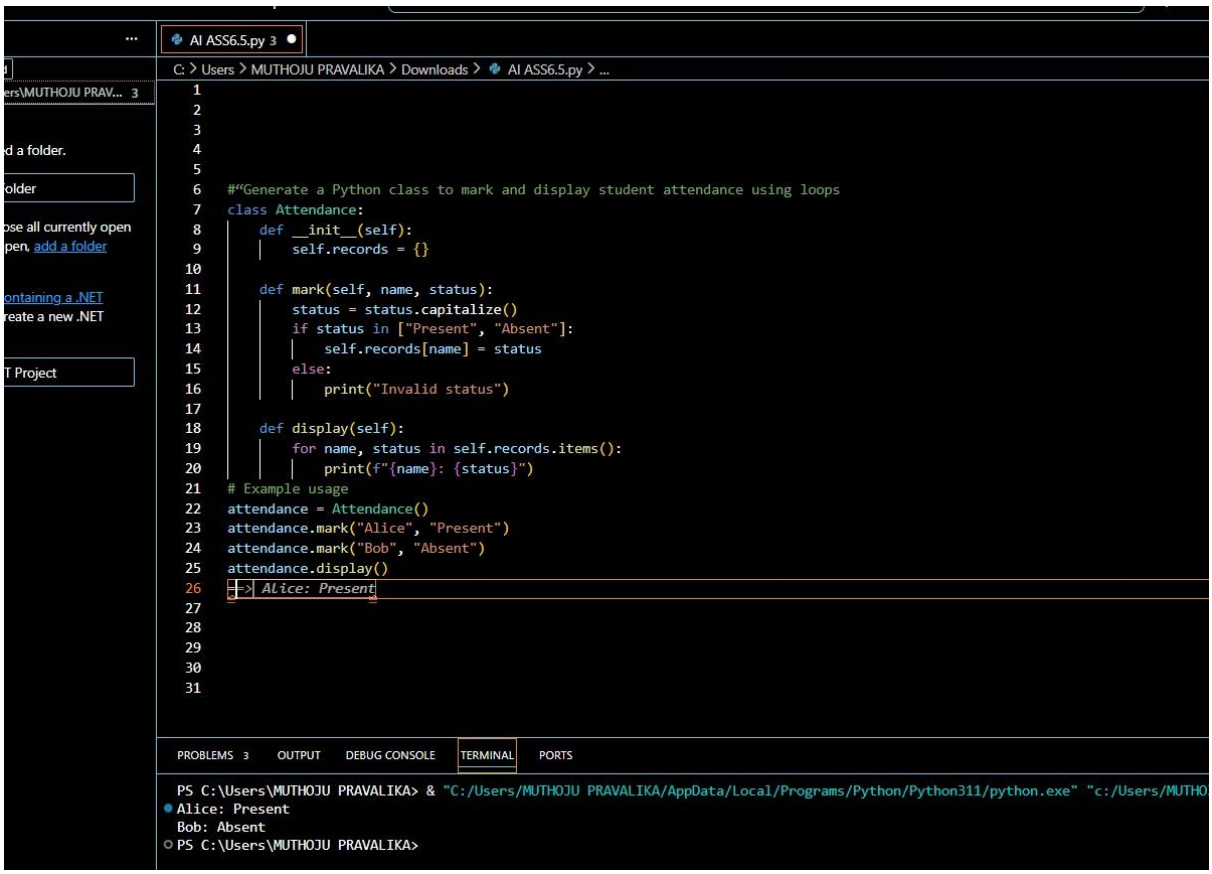
Prompt: "Generate a Python class to mark and display student attendance using loops."

Expected Output:

- AI-generated attendance logic.

• Correct display of attendance.

• Test cases.



Task Description #5 (AI-Based Code Completion for Conditional

Menu Navigation)

Task: Use an AI tool to complete a navigation menu.

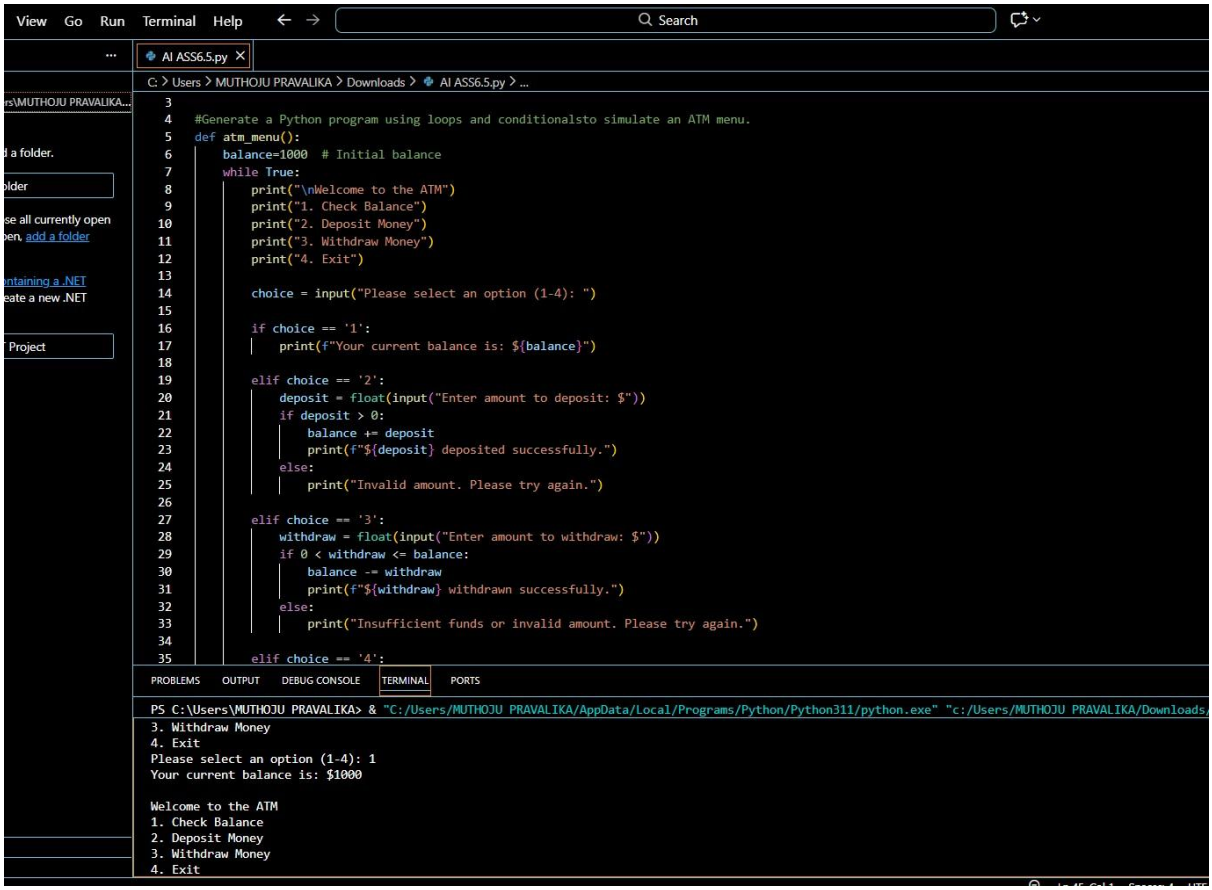
Prompt: “Generate a Python program using loops and conditionals

to simulate an ATM menu.” Expected Output:

• AI-generated menu logic.

• Correct option handling.

• Output verification.


View Go Run Terminal Help
Search
AI ASS6.5.py
C: > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > ...
#Generate a Python program using loops and conditionalsto simulate an ATM menu.
def atm_menu():
balance=1000 # Initial balance
while True:
print("\nWelcome to the ATM")
print("1. Check Balance")
print("2. Deposit Money")
print("3. Withdraw Money")
print("4. Exit")
choice = input("Please select an option (1-4): ")
if choice == '1':
print(f"Your current balance is: ${balance}")
elif choice == '2':
deposit = float(input("Enter amount to deposit: $"))
if deposit > 0:
balance += deposit
print(f"${deposit} deposited successfully.")
else:
print("Invalid amount. Please try again.")
elif choice == '3':
withdraw = float(input("Enter amount to withdraw: $"))
if 0 < withdraw <= balance:
balance -= withdraw
print(f"${withdraw} withdrawn successfully.")
else:
print("Insufficient funds or invalid amount. Please try again.")
elif choice == '4':
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit