Experiment 6.5

AI-Based Code Completion: Working with suggestions for classes, loops, conditionals

Name : M Pravalika

Batch : 34

Hall No : 2303a52347

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

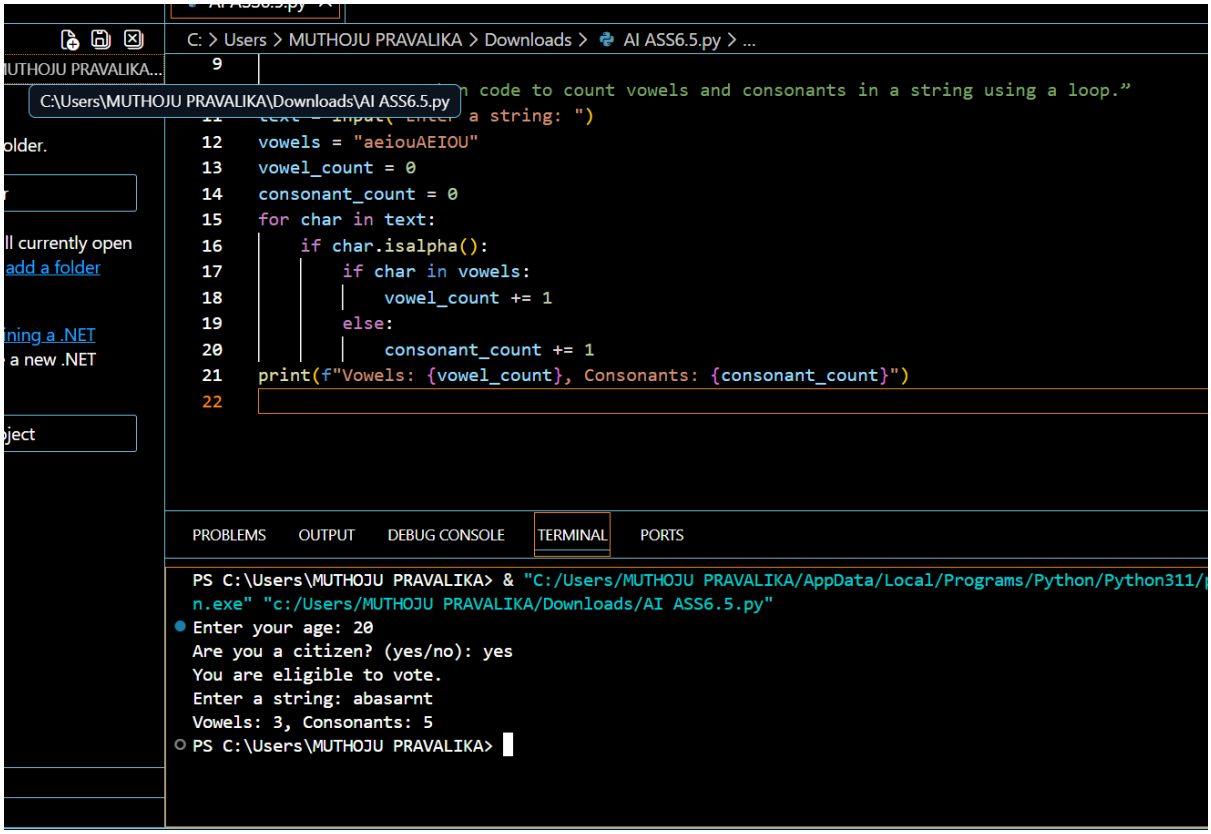
using a loop.”

Expected Output:

• AI-generated string processing logic.

• Correct counts.

• Output verification.



Task Description #3 (AI-Assisted Code Completion Reflection Task)

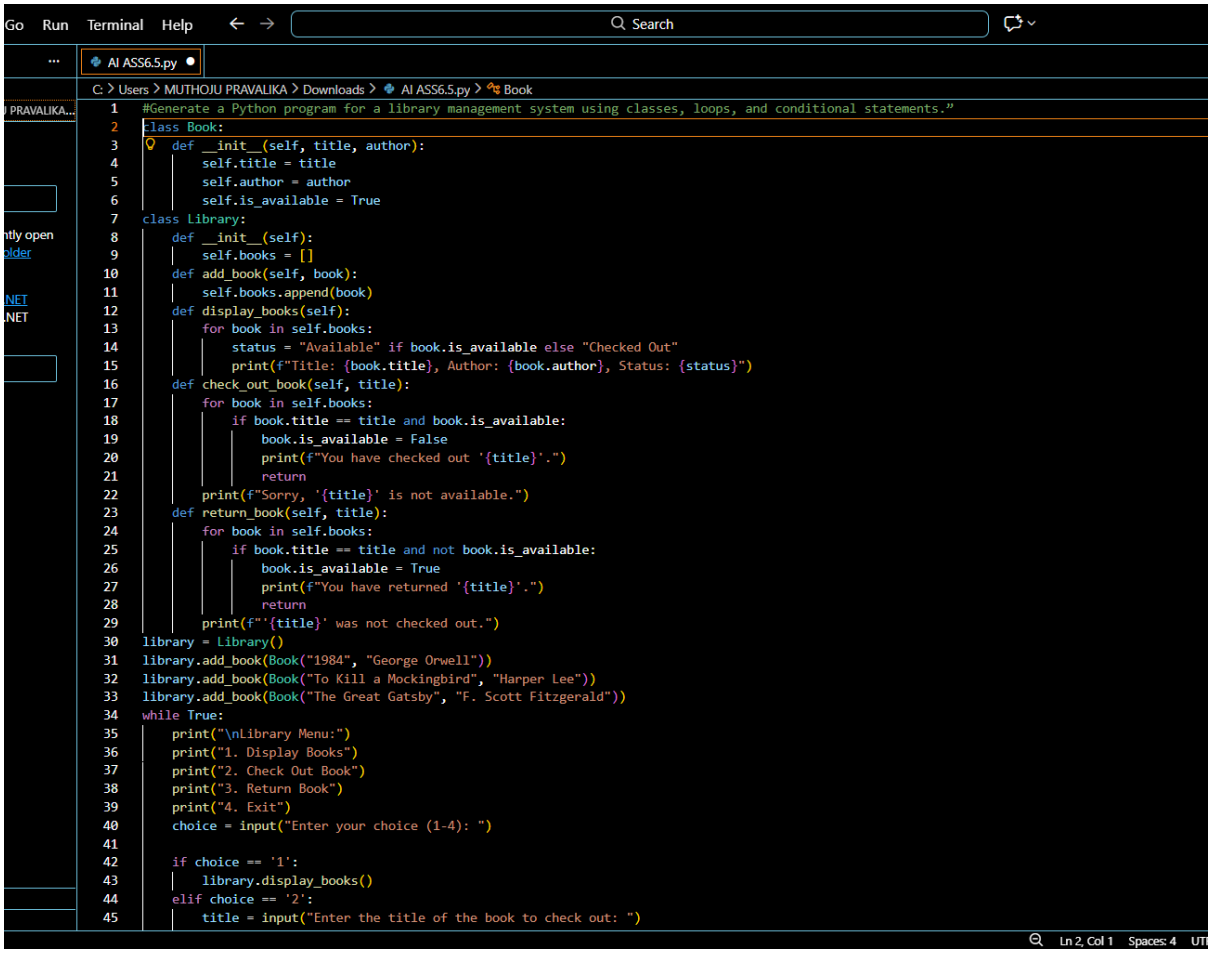
Task: Use an AI tool to generate a complete program using classes, loops, and conditionals.

Prompt:

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

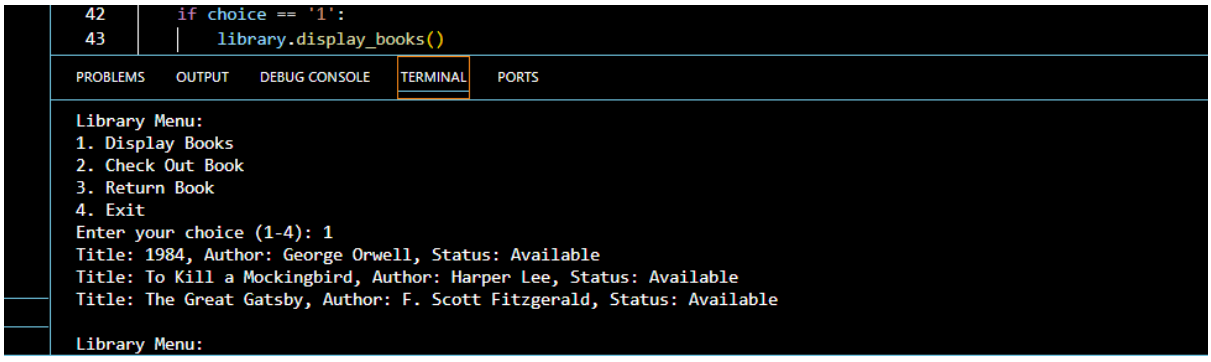
Expected Output:

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



```python
#Generate a Python program for a library management system using classes, loops, and conditional statements."
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = True
class Library:
    def __init__(self):
        self.books = []
    def add_book(self, book):
        self.books.append(book)
    def display_books(self):
        for book in self.books:
            status = "Available" if book.is_available else "Checked Out"
            print(f"Title: {book.title}, Author: {book.author}, Status: {status}")
    def check_out_book(self, title):
        for book in self.books:
            if book.title == title and book.is_available:
                book.is_available = False
                print(f"You have checked out '{title}'.")
                return
        print(f"Sorry, '{title}' is not available.")
    def return_book(self, title):
        for book in self.books:
            if book.title == title and not book.is_available:
                book.is_available = True
                print(f"You have returned '{title}'.")
                return
        print(f"'{title}' was not checked out.")
library = Library()
library.add_book(Book("1984", "George Orwell"))
library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
while True:
    print("\nLibrary Menu:")
    print("1. Display Books")
    print("2. Check Out Book")
    print("3. Return Book")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")

    if choice == '1':
        library.display_books()
    elif choice == '2':
        title = input("Enter the title of the book to check out: ")
```



```
    if choice == '1':
        library.display_books()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Library Menu:
1. Display Books
2. Check Out Book
3. Return Book
4. Exit
Enter your choice (1-4): 1
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
Title: The Great Gatsby, Author: F. Scott Fitzgerald, Status: Available

Library Menu:
```

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

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

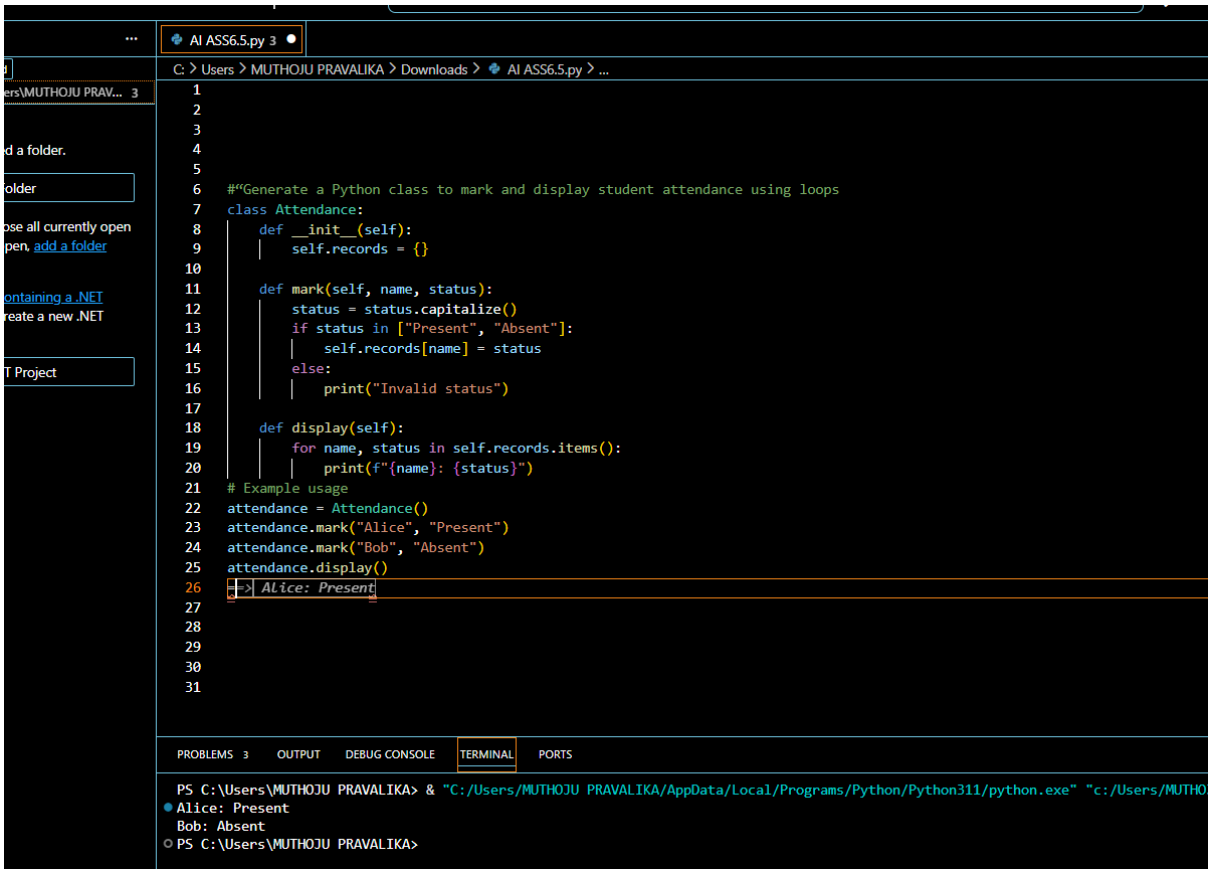
Prompt: "Generate a Python class to mark and display student attendance using loops."

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.

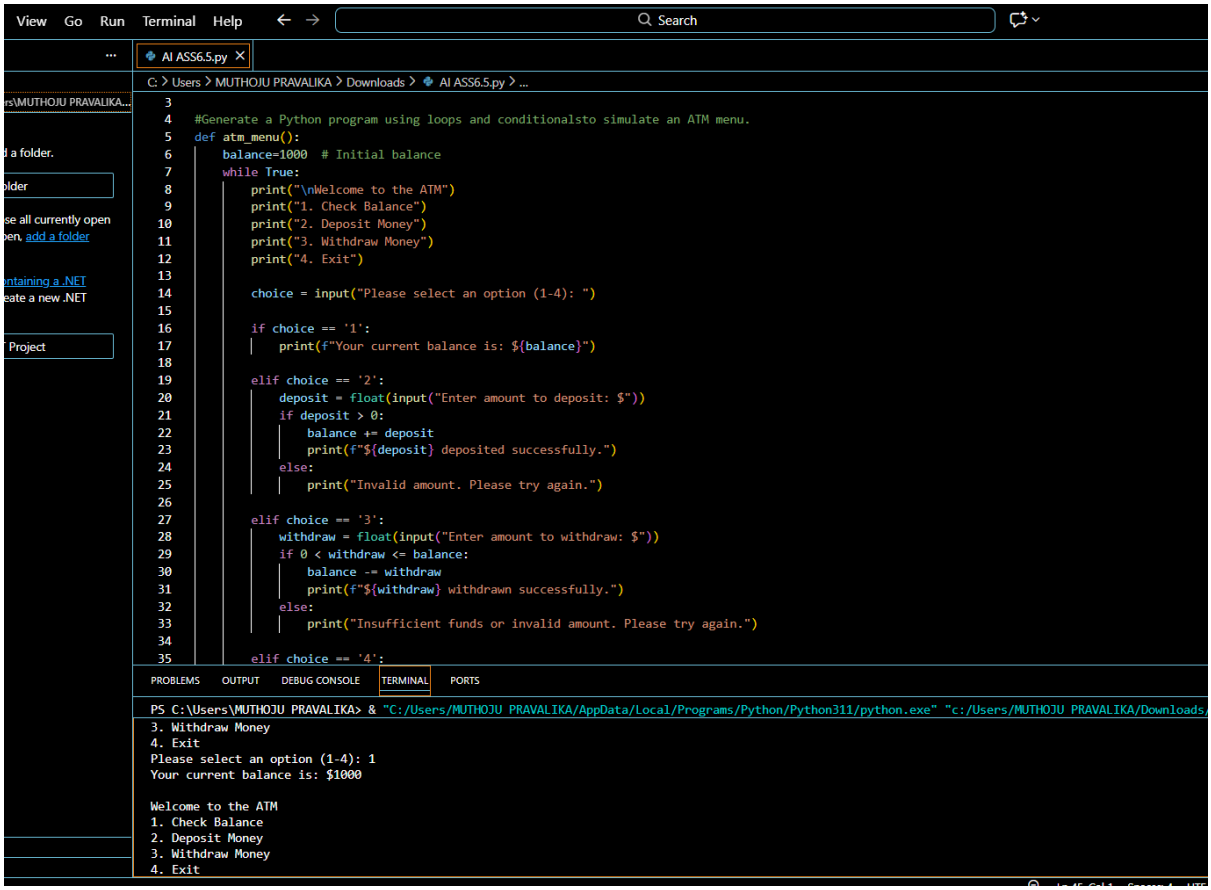
Prompt: "Generate a Python program using loops and conditionals to simulate an ATM menu."

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