AI ASSISTED CODING

LAB-6.5

J.Varshini

2303A51758

Batch-11

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."

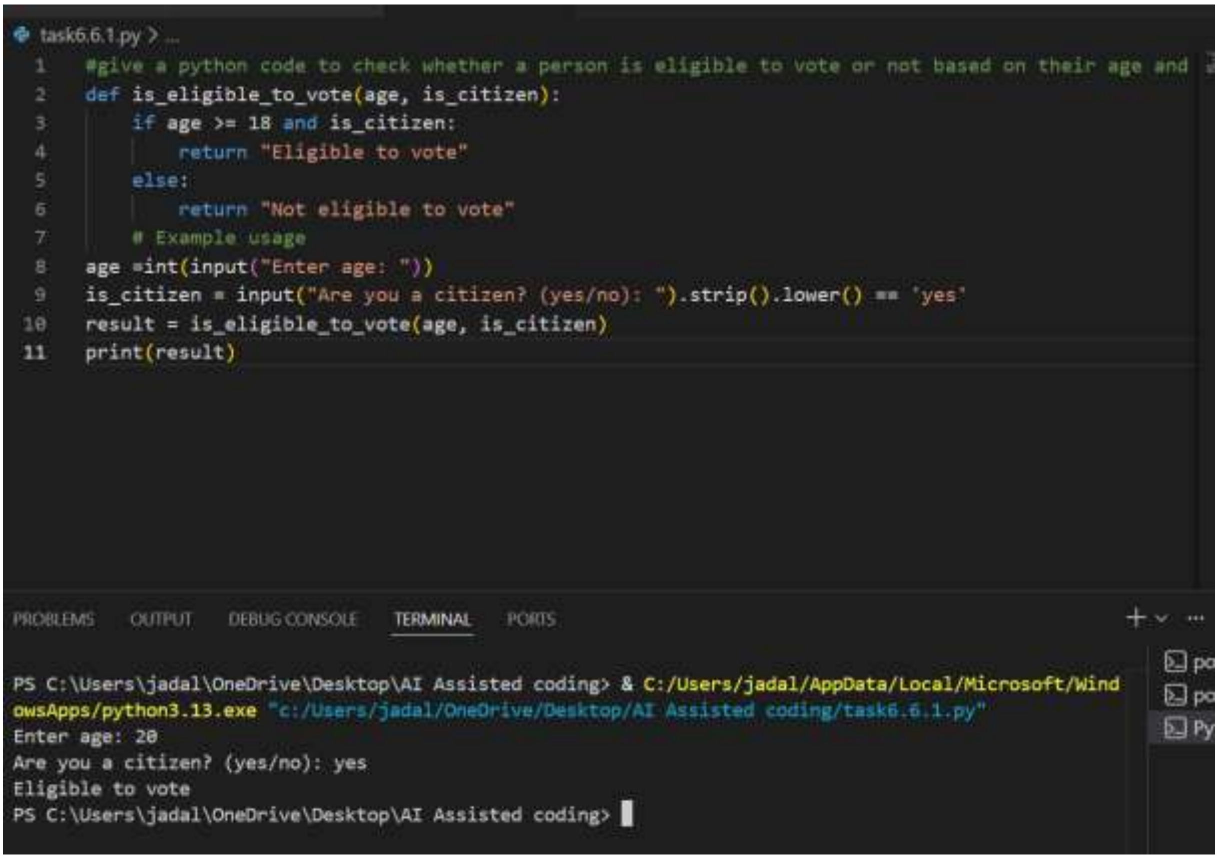
Expected Output:

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

Prompt:

# Give a python code to check whether a person is eligible to vote or not based on their age and citizenship

Given Code and Output:



```python
#give a python code to check whether a person is eligible to vote or not based on their age and
def is_eligible_to_vote(age, is_citizen):
    if age >= 18 and is_citizen:
        return "Eligible to vote"
    else:
        return "Not eligible to vote"
    # Example usage
age =int(input("Enter age: "))
is_citizen = input("Are you a citizen? (yes/no): ").strip().lower() == 'yes'
result = is_eligible_to_vote(age, is_citizen)
print(result)
```

```
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> & C:/Users/jadal/AppData/Local/Microsoft/Wind
owsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task6.6.1.py"
Enter age: 20
Are you a citizen? (yes/no): yes
Eligible to vote
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> 
```

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.

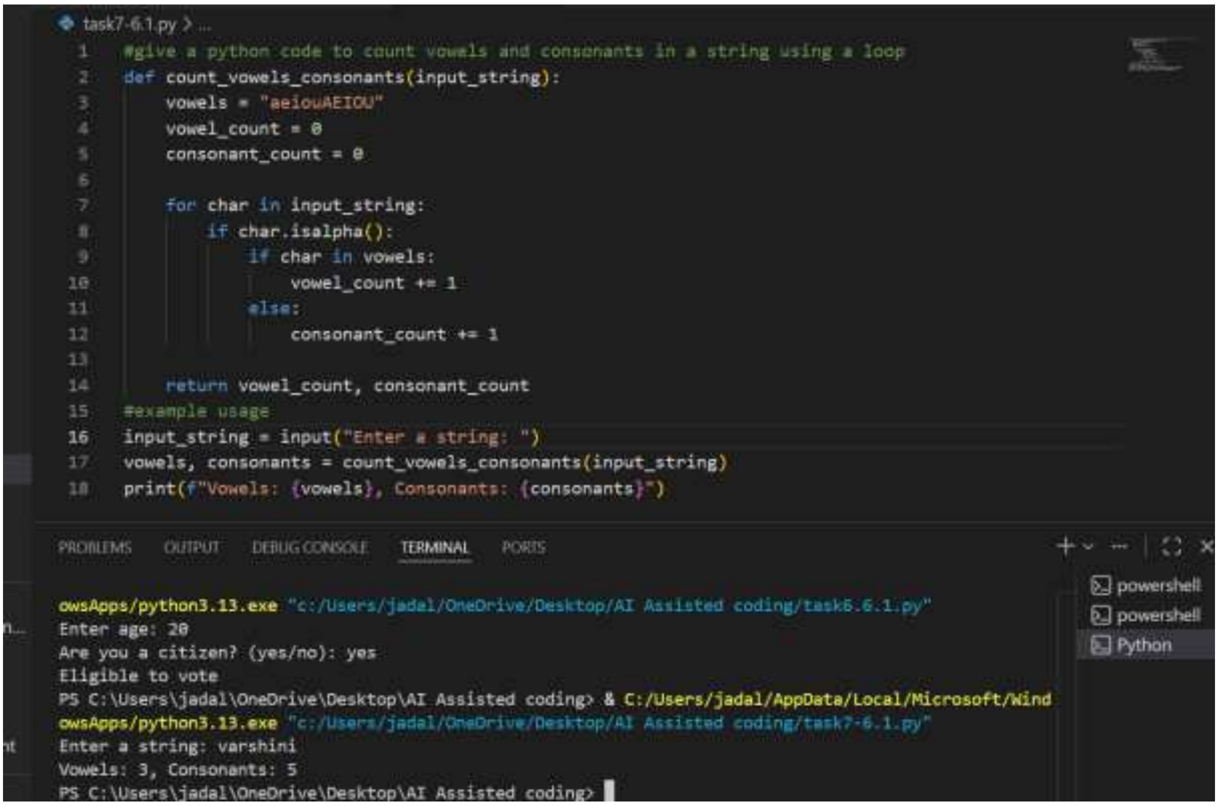
• Correct counts.

• Output verification.

<u>Prompt:</u>

#Given a python code to count vowels and consonants in a string using a loop **<u>Given</u>**

<u>Code and Output:</u>



```python
#give a python code to count vowels and consonants in a string using a loop
def count_vowels_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
        if char.isalpha():
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count
#example usage
input_string = input("Enter a string: ")
vowels, consonants = count_vowels_consonants(input_string)
print(f"Vowels: {vowels}, Consonants: {consonants}")
```

```
owsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task6.6.1.py"
Enter age: 20
Are you a citizen? (yes/no): yes
Eligible to vote
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> & C:/Users/jadal/AppData/Local/Microsoft/Wind
owsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task7-6.1.py"
Enter a string: varshini
Vowels: 3, Consonants: 5
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

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.

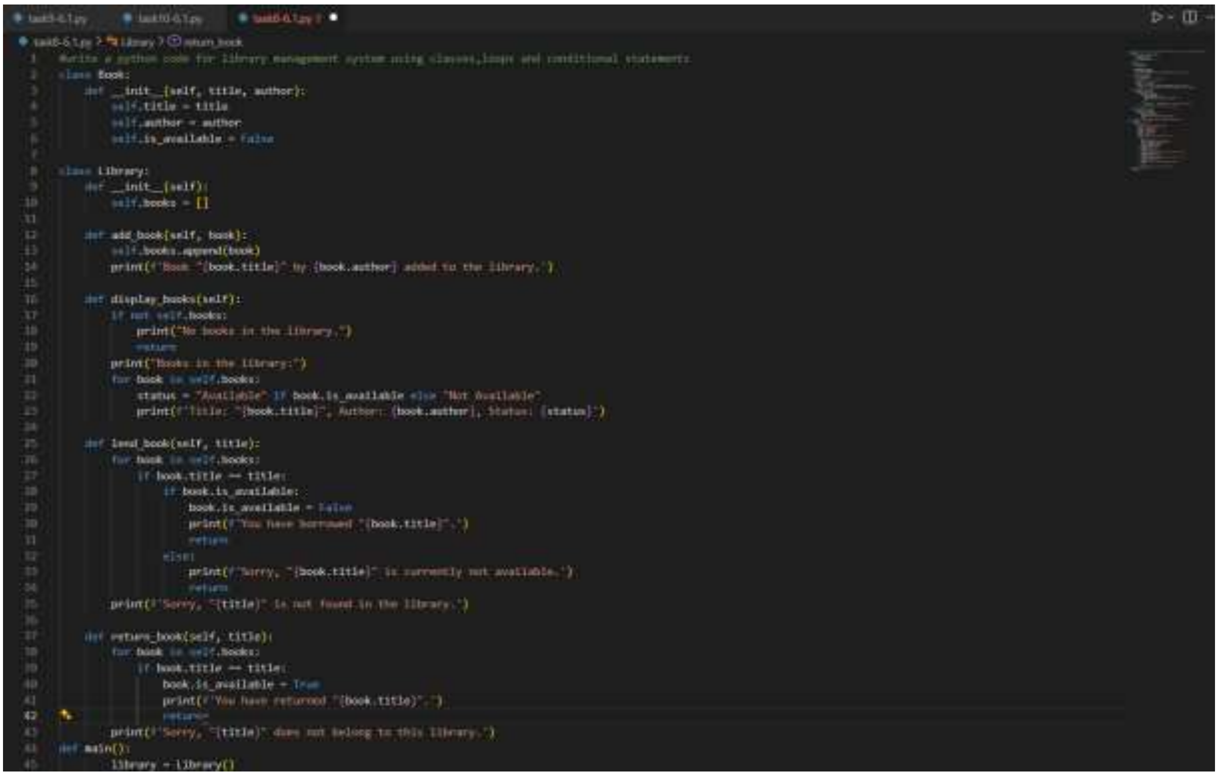
• Review of AI suggestions quality.

• Short reflection on AI-assisted coding experience.

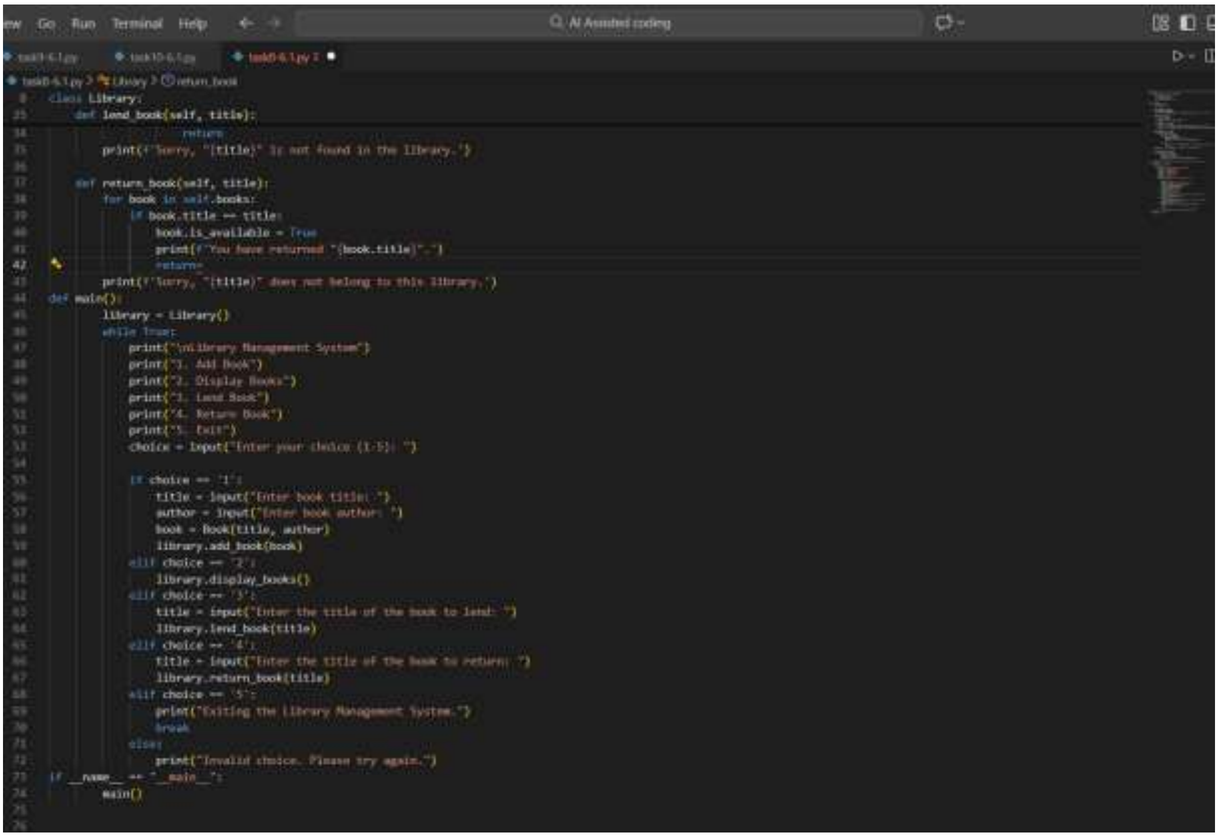
Prompt:

#Give a python code for library management system using classes, loops and conditional statements

Given Code:

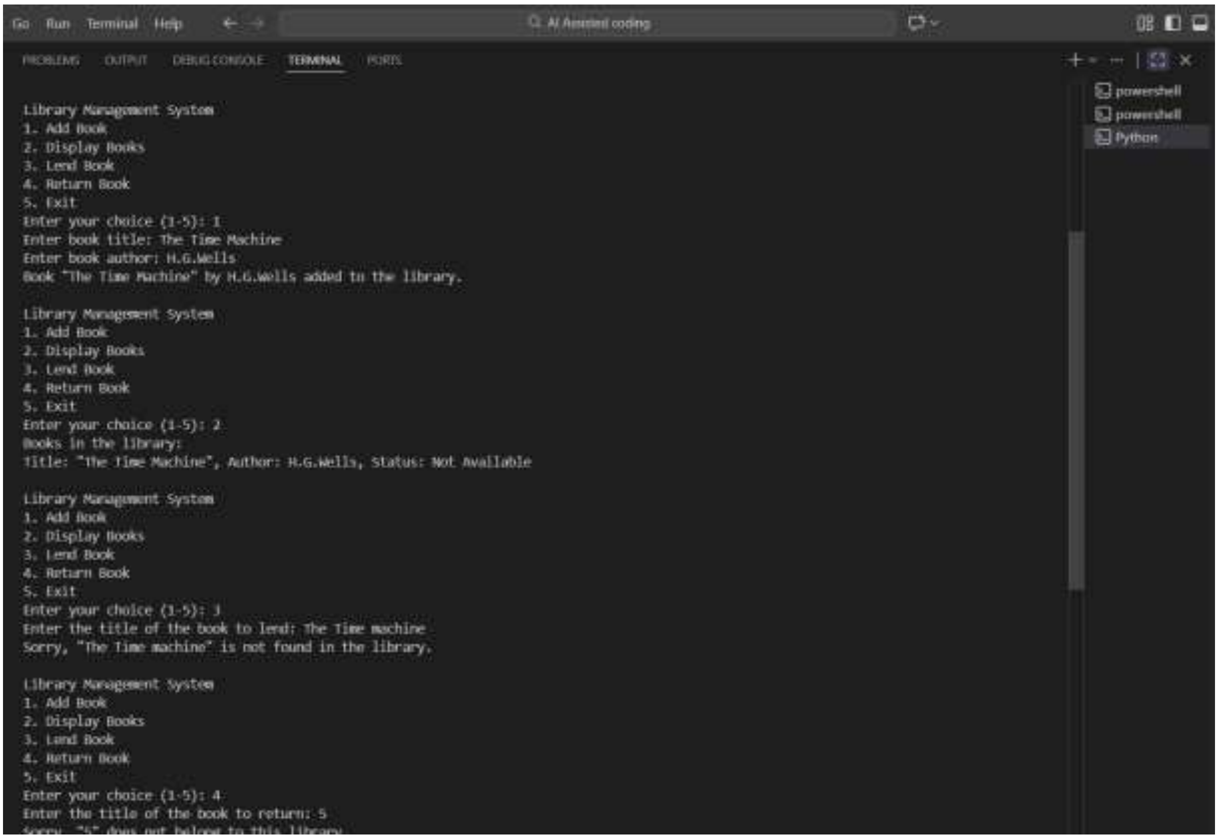


```python
#write a python code for library management system using classes,loops and conditional statements
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = False

class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)
        print(f'Book "{book.title}" by {book.author} added to the library.')

    def display_books(self):
        if not self.books:
            print("No books in the library.")
            return
        print("Books in the library:")
        for book in self.books:
            status = "Available" if book.is_available else "Not Available"
            print(f'Title: "{book.title}", Author: {book.author}, Status: {status}')

    def lend_book(self, title):
        for book in self.books:
            if book.title == title:
                if book.is_available:
                    book.is_available = False
                    print(f'You have borrowed "{book.title}".')
                    return
                else:
                    print(f'Sorry, "{book.title}" is currently not available.')
                    return
        print(f'Sorry, "{title}" is not found in the library.')

    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                book.is_available = True
                print(f'You have returned "{book.title}".')
                return
        print(f'Sorry, "{title}" does not belong to this library.')
def main():
        library = Library()
```



```python
class Library:
    def lend_book(self, title):
                    return
        print(f'Sorry, "{title}" is not found in the library.')

    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                book.is_available = True
                print(f'You have returned "{book.title}".')
                return
        print(f'Sorry, "{title}" does not belong to this library.')
def main():
        library = Library()
        while True:
            print("\nLibrary Management System")
            print("1. Add Book")
            print("2. Display Books")
            print("3. Lend Book")
            print("4. Return Book")
            print("5. Exit")
            choice = input("Enter your choice (1-5): ")

            if choice == '1':
                title = input("Enter book title: ")
                author = input("Enter book author: ")
                book = Book(title, author)
                library.add_book(book)
            elif choice == '2':
                library.display_books()
            elif choice == '3':
                title = input("Enter the title of the book to lend: ")
                library.lend_book(title)
            elif choice == '4':
                title = input("Enter the title of the book to return: ")
                library.return_book(title)
            elif choice == '5':
                print("Exiting the Library Management System.")
                break
            else:
                print("Invalid choice. Please try again.")
if __name__ == "__main__":
        main()
```

Output:



```
Library Management System
1. Add Book
2. Display Books
3. Lend Book
4. Return Book
5. Exit
Enter your choice (1-5): 1
Enter book title: The Time Machine
Enter book author: H.G.Wells
Book "The Time Machine" by H.G.Wells added to the library.

Library Management System
1. Add Book
2. Display Books
3. Lend Book
4. Return Book
5. Exit
Enter your choice (1-5): 2
Books in the library:
Title: "The Time Machine", Author: H.G.Wells, Status: Not Available

Library Management System
1. Add Book
2. Display Books
3. Lend Book
4. Return Book
5. Exit
Enter your choice (1-5): 3
Enter the title of the book to lend: The Time machine
Sorry, "The Time machine" is not found in the library.

Library Management System
1. Add Book
2. Display Books
3. Lend Book
4. Return Book
5. Exit
Enter your choice (1-5): 4
Enter the title of the book to return: 5
Sorry, "5" does not belong to this library.
```

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.”

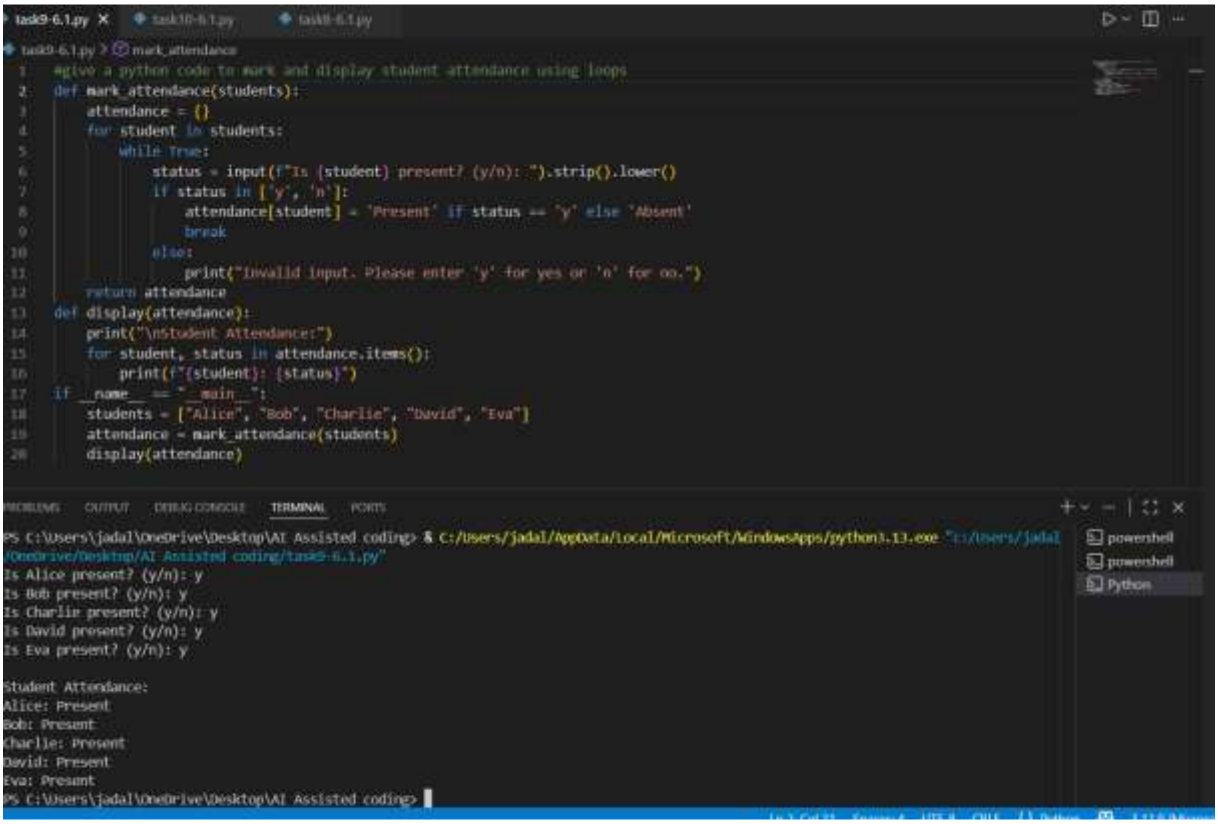
Expected Output:

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

Prompt:

# Give a python code to mark and display student attendance using loops.

Given Code:



```python
#give a python code to mark and display student attendance using loops
def mark_attendance(students):
    attendance = {}
    for student in students:
        while True:
            status = input(f"Is {student} present? (y/n): ").strip().lower()
            if status in ['y', 'n']:
                attendance[student] = 'Present' if status == 'y' else 'Absent'
                break
            else:
                print("Invalid input. Please enter 'y' for yes or 'n' for no.")
    return attendance
def display(attendance):
    print("\nStudent Attendance:")
    for student, status in attendance.items():
        print(f"{student}: {status}")
if __name__ == "__main__":
    students = ["Alice", "Bob", "Charlie", "David", "Eva"]
    attendance = mark_attendance(students)
    display(attendance)
```

```
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task9-6.1.py"
Is Alice present? (y/n): y
Is Bob present? (y/n): y
Is Charlie present? (y/n): y
Is David present? (y/n): y
Is Eva present? (y/n): y

Student Attendance:
Alice: Present
Bob: Present
Charlie: Present
David: Present
Eva: Present
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

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

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

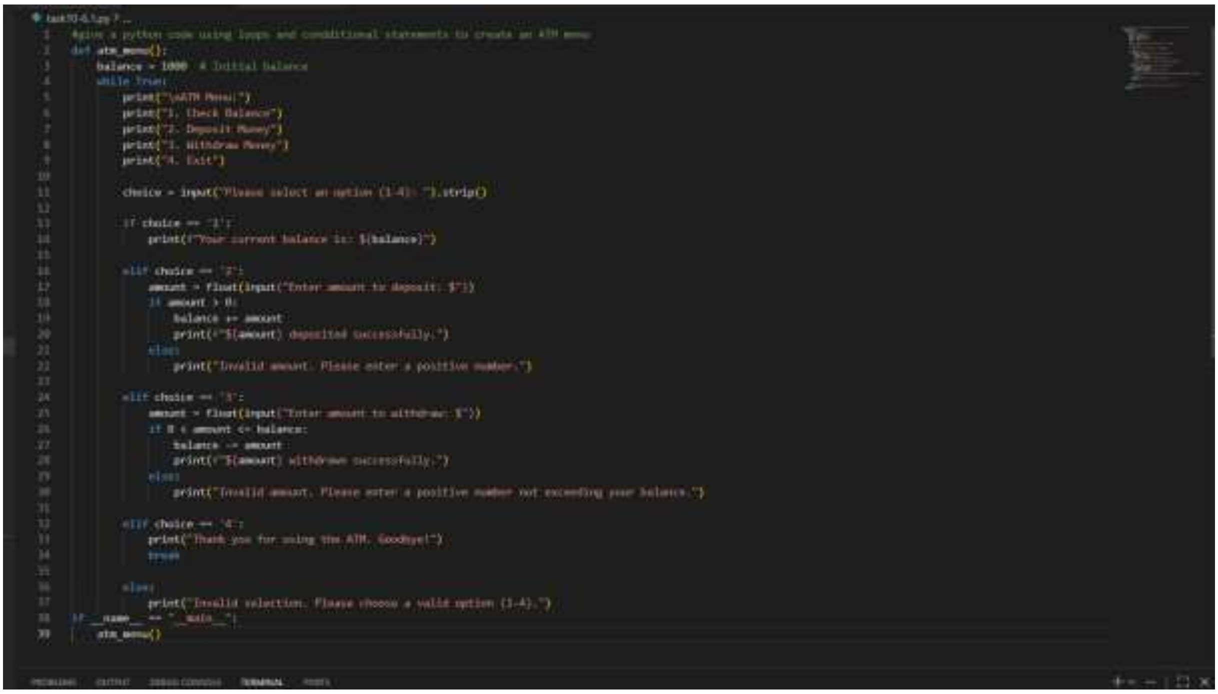
Prompt: "Generate a Python program using loops and conditionals to simulate an ATM menu." Expected Output:

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

Prompt:

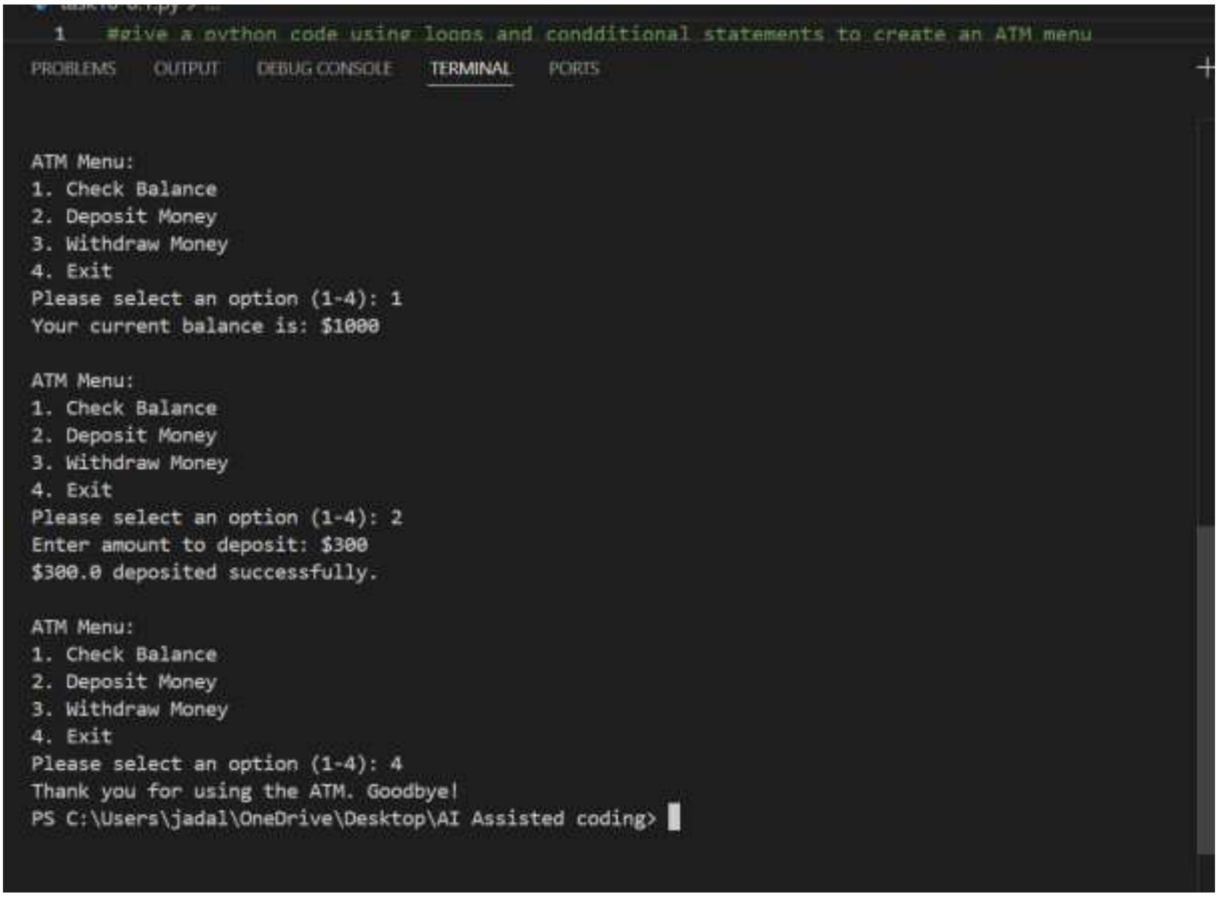
# Give a python code using loops and conditional statements to create an ATM menu **Given**

Code:



```python
#give a python code using loops and condditional statements to create an ATM menu
def atm_menu():
    balance = 1000  # Initial balance
    while True:
        print("\nATM Menu:")
        print("1. Check Balance")
        print("2. Deposit Money")
        print("3. Withdraw Money")
        print("4. Exit")

        choice = input("Please select an option (1-4): ").strip()

        if choice == '1':
            print(f"Your current balance is: ${balance}")

        elif choice == '2':
            amount = float(input("Enter amount to deposit: $"))
            if amount > 0:
                balance += amount
                print(f"${amount} deposited successfully.")
            else:
                print("Invalid amount. Please enter a positive number.")

        elif choice == '3':
            amount = float(input("Enter amount to withdraw: $"))
            if 0 < amount <= balance:
                balance -= amount
                print(f"${amount} withdrawn successfully.")
            else:
                print("Invalid amount. Please enter a positive number not exceeding your balance.")

        elif choice == '4':
            print("Thank you for using the ATM. Goodbye!")
            break

        else:
            print("Invalid selection. Please choose a valid option (1-4).")
if __name__ == "__main__":
    atm_menu()
```

Output:



```
1   #give a python code using loops and condditional statements to create an ATM menu
```

```
ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 2
Enter amount to deposit: $300
$300.0 deposited successfully.

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 4
Thank you for using the ATM. Goodbye!
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> 
```