AI ASSISTED CODING

LAB-6.5

p.v gayathri

2303A51909

Batch-12

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."

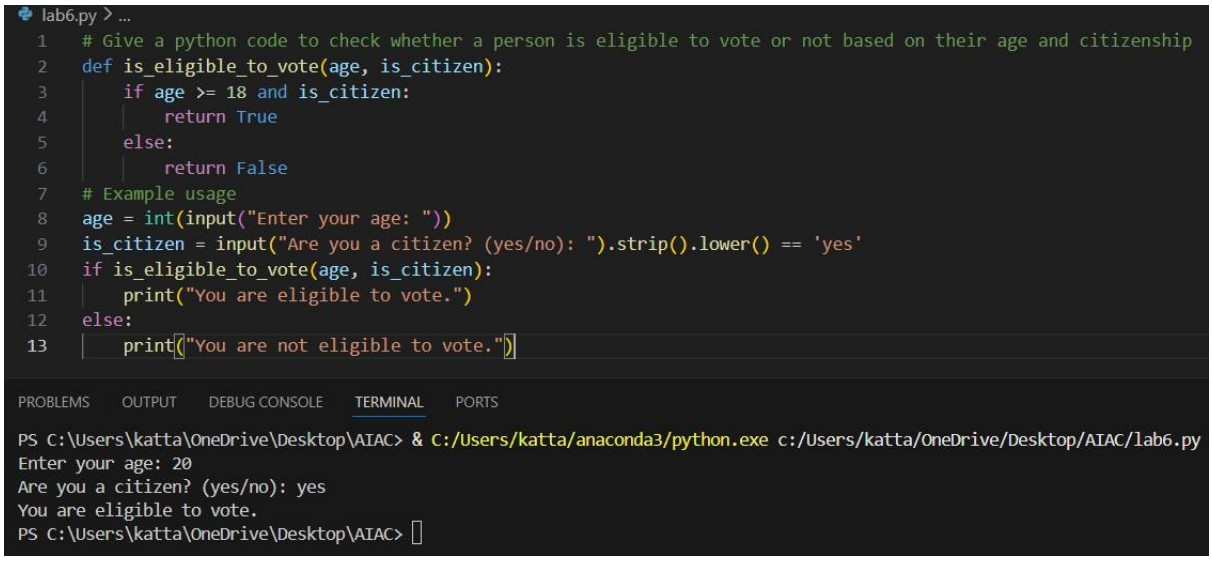
Expected Output:

- **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:



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.**

Prompt:

#Given a python code to count vowels and consonants in a string using a loop **Given**

Code and Output:

lab6.py > ...

```
#Given 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():  # Check if the character is a letter
            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}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/Users/k
Enter a string:Hello World
Vowels: 3, Consonants: 7
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

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
#Give 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 = True
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):
        print("Available books in the library:")
        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 borrow_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 checked out.')
                    return
        print(f'Sorry, "{title}" is not available in the library.')
    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                if not book.is_available:
                    book.is_available = True
                    print(f'You have returned "{book.title}".')
                    return
                else:
                    print(f'"{book.title}" was not checked out.')
                    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. Borrow 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 borrow: ")
            library.borrow_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. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
if __name__ == "__main__":
    main()
```

Output:

```
Library Management System
1. Add Book
2. Display Books
3. Borrow 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. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 2
Available books in the library:
Title: "The Time Machine", Author: H.G. Wells, Status: Available

Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 3
Enter the title of the book to borrow: The Time Machine
You have borrowed "The Time Machine".

Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 4
Enter the title of the book to return: The Time Machine
You have returned "The Time Machine".
```

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:
        status = input(f"Is {student} present? (y/n): ").strip().lower()
        if status == 'y':
            attendance[student] = 'Present'
        else:
            attendance[student] = 'Absent'
    return attendance
def display_attendance(attendance):
    print("\nAttendance Record:")
    for student, status in attendance.items():
        print(f"{student}: {status}")
if __name__ == "__main__":
    student_list = ["Alice", "Bob", "Charlie", "David"]
    attendance_record = mark_attendance(student_list)
    display_attendance(attendance_record)
```

Output:

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Use
Is Alice present? (y/n): y
Is Bob present? (y/n): y
Is Charlie present? (y/n): n
Is David present? (y/n): y

Attendance Record:
Alice: Present
Bob: Present
Charlie: Absent
David: Present
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

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:

```
# Give a python code using loops and conditional statements to create an ATM menu
def atm_menu():
    balance = 1000  # Initial balance
    while True:
        print("\n--- ATM Menu ---")
        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':
            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 check your balance and try again.")

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

        else:
            print("Invalid selection. Please choose a valid option (1-4).")
# Run the ATM menu
atm_menu()
```

Output:

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/ar

--- 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: $10000
$10000.0 deposited successfully.

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 3
Enter amount to withdraw: $100
$100.0 withdrawn 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!
```