SR UNIVERSITY

School of Computer Science and Artificial Intelligence

Lab Assignment # 6.5

Program	**: B. Tech (CSE)**
Specialization	**:**
Course Title	**: AI Assisted coding**
Course Code	**:**
Semester	**: II**
Academic Session	**: 2025-2026**
Name of Student	**: Mohammed Hashir Sayam Hussain**
Enrollment No.	**: 2403A51L27**
Batch No.	**: 51**
Date	**:23-01-2026**

Submission Starts here

Task1: AI-Based Code Completion for **Conditional Eligibility Check** – Voting eligibility based on age and citizenship.

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

Code Screenshot:

```
age = int(input("Enter age: "))
citizen = input("Are you a citizen? (yes/no): ").lower()

if age >= 18 and citizen == "yes":
    print("Eligible to vote")
else:
    print("Not eligible to vote")
```

Output :
Enter age: 20
Are you a citizen? (yes/no): yes
Eligible to vote

Explanation of code:

- The program takes **age** and **citizenship status** as input.
- The `if` condition checks whether age is **18 or above** and citizenship is `"yes"`.
- If both conditions are true, voting eligibility is confirmed.
- Otherwise, the user is declared not eligible.
- This repeats until all 10 numbers are printed.

SR UNIVERSITY

Task 2:AI-Based Code Completion for **Loop-Based String Processing** – Counting vowels and consonants.

Prompt:*# Generate Python code to count vowels and consonants in a string using a loop.*

Code Screenshot:

```
text = input("Enter a string: ").lower()
vowels = 0
consonants = 0

for char in text:
    if char.isalpha():
        if char in "aeiou":
            vowels += 1
        else:
            consonants += 1

print("Vowels:", vowels)
print("Consonants:", consonants)
```

Output :
Enter a string: Hello World
Vowels: 3
Consonants: 7

Explanation of code:

- The string is converted to lowercase for easy comparison.
- A loop checks each character.
- `isalpha()` ensures only letters are counted.
- Vowels are counted separately from consonants.
- Final counts are displayed.

Task 3:AI-Assisted Code Completion Reflection – **Library Management System**.

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

Code Screenshot:

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

    def add_book(self, book):
        self.books.append(book)

    def display_books(self):
        if not self.books:
            print("No books available")
        else:
            for book in self.books:
                print(book)

lib = Library()

while True:
    print("1. Add Book")
    print("2. Display Books")
    print("3. Exit")

    choice = input("Enter choice: ")

    if choice == "1":
        book = input("Enter book name: ")
        lib.add_book(book)
    elif choice == "2":
        lib.display_books()
    elif choice == "3":
        break
    else:
        print("Invalid choice")
```

Output :

1. Add Book
2. Display Books
3. Exit

Enter choice: 1

Enter book name: Python Basics

Explanation of code:

- A `Library` class stores books using a list.
- Loops allow repeated user interaction.
- Conditionals manage menu choices.
- AI-generated code is readable and modular.
- Demonstrates effective AI-assisted programming.

SR UNIVERSITY

Task4:-Assisted Code Completion for **Class-Based Attendance System**.

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

Code Screenshot:

```
class Attendance:
    def __init__(self):
        self.students = {}

    def mark_attendance(self, name, status):
        self.students[name] = status

    def display_attendance(self):
        for name, status in self.students.items():
            print(name, ":", status)

att = Attendance()
att.mark_attendance("Alice", "Present")
att.mark_attendance("Bob", "Absent")
att.display_attendance()
```

Output :

Alice : Present

Bob : Absent

Explanation of code:

- A dictionary stores student names and attendance status.
- Attendance is marked using a method.
- Loop iterates through records for display.
- Shows clean and efficient AI-generated logic.

SR UNIVERSITY

Task5: AI-Based Code Completion for **Conditional Menu Navigation (ATM System)**.

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

Code Screenshot:

```
balance = 5000

while True:
    print("1. Check Balance")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Exit")

    choice = input("Enter choice: ")

    if choice == "1":
        print("Balance:", balance)
    elif choice == "2":
        amount = int(input("Enter deposit amount: "))
        balance += amount
    elif choice == "3":
        amount = int(input("Enter withdraw amount: "))
        if amount <= balance:
            balance -= amount
        else:
            print("Insufficient balance")
    elif choice == "4":
        break
    else:
        print("Invalid option")
```

Output :

1. Check Balance
2. Deposit
3. Withdraw
4. Exit

Enter choice: 1

Balance: 5000

Explanation of code:

- Uses a loop for continuous menu display.
- Conditional statements handle ATM operations.
- Balance is updated securely.
- Demonstrates real-world AI-assisted menu logic.

SR UNIVERSITY