AIAC ASSIGNMENT-6.5

Name:K.Lokesh **Batch - 21**

H.No:2303A51201

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.”

```
def check_voting_eligibility():
    try:
        # Take age input from the user and convert it to an integer
        age = int(input("Enter your age: "))

        # Take citizenship input from the user and convert it to lowercase
        citizenship = input("Enter your citizenship: ").strip().lower()

        # Check if the user is underage
        if age < 18:
            print("Underage - Not eligible to vote")

        # Check if the user is not an Indian citizen
        elif citizenship != "indian":
            print("Not eligible to vote due to citizenship")

        # If the user is 18 or older and an Indian citizen
        else:
            print("Eligible to vote")

    except ValueError:
        # Handle the case where age input is not a valid integer
        print("Invalid input for age. Please enter a valid integer.")
# Call the function to check voting eligibility
check_voting_eligibility()
```

Output:

```
Enter your age: 19
Enter your citizenship: indian
Eligible to vote
```

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.”

```python
def count_vowels_consonants():
    try:
        # Take input from the user
        user_input = input("Please enter a string (no numbers allowed): ")

        # Check if the input contains any digits
        if any(char.isdigit() for char in user_input):
            raise ValueError("Input should not contain numbers.")

        # Convert the input string to lowercase
        user_input = user_input.lower()

        # Initialize counters for vowels and consonants
        vowel_count = 0
        consonant_count = 0

        # Define a set of vowels for easy checking
        vowels = set("aeiou")

        # Loop through each character in the string
        for char in user_input:
            # Check if the character is an alphabet letter
            if char.isalpha():
                # Check if the character is a vowel
                if char in vowels:
                    vowel_count += 1  # Increment vowel count
                else:
                    consonant_count += 1  # Increment consonant count

        # Display the results
        print(f"Number of vowels: {vowel_count}")
        print(f"Number of consonants: {consonant_count}")

    except ValueError as ve:
        print(ve)  # Print the error message if input is invalid
    except Exception as e:
        print(f"An error occurred: {e}")  # Catch any other exceptions
# Call the function to execute the program
count_vowels_consonants()
```

OUTPUT:

```
Please enter a string (no numbers allowed): Saisharan suhas
Number of vowels: 6
Number of consonants: 8
```

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
class Library:
    def __init__(self):
        # Initialize an empty list to store book names
        self.books = []

    def add_book(self, book_name):
        # Add a book to the library
        self.books.append(book_name)
        print(f'Book "{book_name}" added to the library.')

    def remove_book(self, book_name):
        # Remove a book from the library if it exists
        if book_name in self.books:
            self.books.remove(book_name)
            print(f'Book "{book_name}" removed from the library.')
        else:
            print(f'Book "{book_name}" does not exist in the library.')

    def display_books(self):
        # Display all books in the library
        if self.books:
            print("Books in the library:")
            for book in self.books:
                print(f'- {book}')
        else:
            print("No books in the library.")
def main():
    library = Library()

    while True:
        # Display the menu options
        print("\nLibrary Management System")
        print("1. Add book")
        print("2. Remove book")
        print("3. Display books")
        print("4. Exit")

        try:
            # Get user choice and validate it as an integer
            choice = int(input("Enter your choice (1-4): "))
```

```
        try:
            # Get user choice and validate it as an integer
            choice = int(input("Enter your choice (1-4): "))

            if choice == 1:
                # Add a book
                book_name = input("Enter the name of the book to add: ")
                library.add_book(book_name)
            elif choice == 2:
                # Remove a book
                book_name = input("Enter the name of the book to remove: ")
                library.remove_book(book_name)
            elif choice == 3:
                # Display all books
                library.display_books()
            elif choice == 4:
                # Exit the program
                print("Exiting the Library Management System. Goodbye!")
                break
            else:
                print("Invalid choice. Please enter a number between 1 and 4.")
        except ValueError:
            # Handle invalid input that cannot be converted to an integer
            print("Invalid input. Please enter a valid number.")
if __name__ == "__main__":
    main()
```

OUTPUT:

```
Library Management System
1. Add book
2. Remove book
3. Display books
4. Exit
Enter your choice (1-4): 1
Enter the name of the book to add: Suhas
Book "Suhas" added to the library.

Library Management System
1. Add book
2. Remove book
3. Display books
4. Exit
Enter your choice (1-4): 2
Enter the name of the book to remove: Suhas
Book "Suhas" removed from the library.

Library Management System
1. Add book
2. Remove book
3. Display books
4. Exit
Enter your choice (1-4): 3
Books in the library:
- sharan
- Sherlock

Library Management System
1. Add book
2. Remove book
3. Display books
4. Exit
Enter your choice (1-4): 4
Exiting the Library Management System. Goodbye!
```

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

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

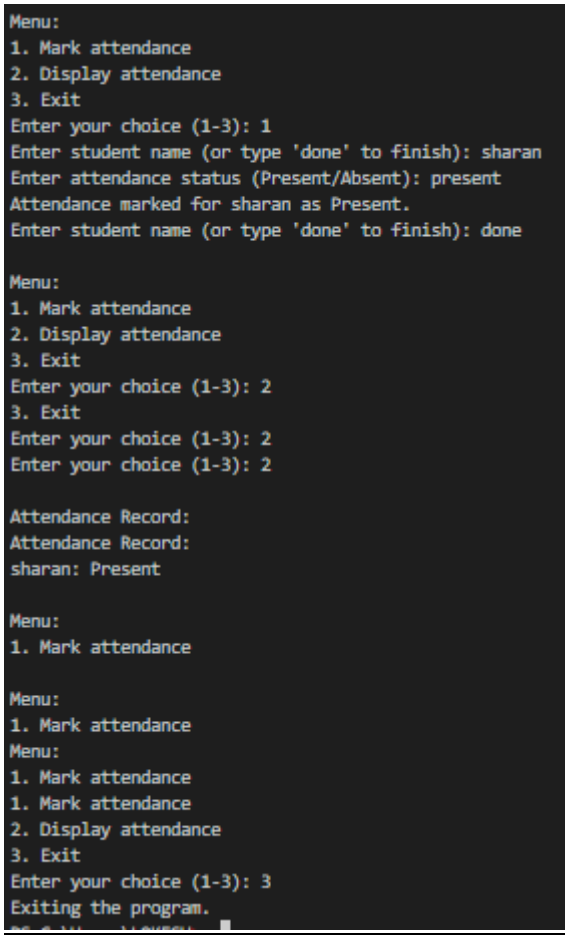
Prompt: “Generate a Python class to mark and display student attendance using loops.”

Expected Output:

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

```python
class Attendance:
    def __init__(self):
        # Initialize an empty dictionary to store student attendance
        self.attendance_record = {}

    def mark_attendance(self):
        # Loop to mark attendance for multiple students
        while True:
            student_name = input("Enter student name (or type 'done' to finish): ")

            if student_name.lower() == 'done':
                break

            # Ensure the student name is a valid string (not empty)
            if isinstance(student_name, str) and student_name.strip():
                status = input("Enter attendance status (Present/Absent): ").capitalize()

                # Validate attendance status
                if status == "Present" or status == "Absent":
                    self.attendance_record[student_name] = status
                    print(f"Attendance marked for {student_name} as {status}.")
                else:
                    print("Invalid status. Please enter 'Present' or 'Absent'.")
            else:
                print("Invalid name. Please enter a valid student name.")

    def display_attendance(self):
        # Display the attendance record
        if not self.attendance_record:
            print("No attendance records available.")
        else:
            print("\nAttendance Record:")
            for student, status in self.attendance_record.items():
                print(f"{student}: {status}")


def run():
    attendance_system = Attendance()

    while True:
        print("\nMenu:")
        print("1. Mark attendance")
        print("2. Display attendance")
        print("3. Exit")

        try:
            choice = int(input("Enter your choice (1-3): "))

            if choice == 1:
                attendance_system.mark_attendance()
            elif choice == 2:
                attendance_system.display_attendance()
            elif choice == 3:
                print("Exiting the program.")
                break
            else:
                print("Invalid choice. Please enter a number between 1 and 3.")

        except ValueError:
            print("Invalid input. Please enter a valid integer.")


if __name__ == "__main__":
    run()
```

OUTPUT:



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**

```python
balance = 0.0  # initial balance (float)

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

    try:
        choice = int(input("Enter your choice (1-4): "))

        if choice == 1:
            print("Current Balance: ₹", balance)

        elif choice == 2:
            amount = float(input("Enter amount to deposit: "))
            if amount > 0:
                balance += amount
                print("Amount deposited successfully.")
            else:
                print("Deposit amount must be positive.")

        elif choice == 3:
            amount = float(input("Enter amount to withdraw: "))
            if amount <= 0:
                print("Withdrawal amount must be positive.")
            elif amount > balance:
                print("Insufficient balance.")
            else:
                balance -= amount
                print("Please collect your cash.")

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

        else:
            print("Invalid choice. Please enter between 1 and 4.")

    except ValueError:
        print("Invalid input! Please enter numbers only.")
```

Output:

```
ATM Menu
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 1
Current Balance: ₹ 0.0

ATM Menu
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 2
Enter amount to deposit: 2000
Amount deposited successfully.

ATM Menu
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 3
Enter amount to withdraw: 1000
Please collect your cash.

ATM Menu
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 4
Thank you for using the ATM. Exiting...
```