Lab Assignment-6.5

NAME: P.sathwik
BATCH : 12
ROLLNO: 2303A51780

TASK 1: (AI-BASED CODE COMPLETION FOR CONDITIONAL ELIGIBILITY CHECK)

PROMPT USED:
Generate Python code to check voting eligibility based on age and citizenship.

Code:

```python
#generate python code to check voting eligibility based on age and citizenship.
def is_eligible_to_vote(age, is_citizen):
    if age >= 18 and is_citizen:
        return True
    return False
try:
    age = int(input("Enter your age: "))
    citizenship_input = input("Are you a citizen? (yes/no): ").strip().lower()
    is_citizen = citizenship_input == "yes"

    if is_eligible_to_vote(age, is_citizen):
        print("You are eligible to vote.")
    else:
        print("You are not eligible to vote.")
except ValueError:
    print("Please enter a valid age.")
```

Output:

```
Enter your age: 23
Are you a citizen? (yes/no): yes
You are eligible to vote.
```

Explanation:
This program checks voting eligibility using age and citizenship.
If the person is 18 or older and is a citizen, they are allowed to vote.
If the age is invalid or the conditions are not met, the program shows an appropriate message.

Task 2:(AI-Based Code Completion for loop based string processing)

Prompt used:

```
#generate python code to count vowels and consonants in a string using loops.
# #Optimize the code using list comprehensions.
```

Code:

```
def count_vowels_and_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
user_input = input("Enter a string: ")
vowels, consonants = count_vowels_and_consonants(user_input)
print(f"Vowels: {vowels}, Consonants: {consonants}")
#Optimize the code using list comprehensions.
def count_vowels_and_consonants_optimized(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = sum(1 for char in input_string if char.isalpha() and char in vowels)
    consonant_count = sum(1 for char in input_string if char.isalpha() and char not in vowels)
    return vowel_count, consonant_count
user_input = input("Enter a string (optimized): ")
vowels, consonants = count_vowels_and_consonants_optimized(user_input)
print(f"Vowels: {vowels}, Consonants: {consonants}")
```

Output:

```
Enter a string: A E i o U
Vowels: 5, Consonants: 0
Enter a string (optimized): e b g J l
Vowels: 1, Consonants: 4
```

Explanation:

This program counts the number of vowels and consonants in a given string.
It first uses a loop to check each character and then an optimized method using list comprehensions.
Only alphabet letters are considered for counting.

Task 3: (AI-Based Code Completion Reflection task)

Prompt used:

```
#generate a python program or a library management system using loops,classes
and conditional statements.
# #Optimize the library management system by adding search functionality.
```

Code:

```
# #Optimize the library management system by adding search functionality.
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = True

    def borrow_book(self):
        if self.is_available:
            self.is_available = False
            return True
        return False

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

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

    def display_books(self):
        print("\nAvailable Books:")
        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 borrow_book(self, title):
        for book in self.books:
            if book.title == title:
                if book.borrow_book():
                    print(f"You have borrowed '{title}'.")
                else:
                    print(f"Sorry, '{title}' is not available.")
                return
        print(f"Book '{title}' not found in the library.")

    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                book.return_book()
                print(f"You have returned '{title}'.")
                return
        print(f"Book '{title}' not found in the library.")
```

```
    def search_books(self, keyword):
        print(f"\nSearch Results for '{keyword}':")
        found = False
        for book in self.books:
            if keyword.lower() in book.title.lower() or keyword.lower() in book.author.lower():
                status = "Available" if book.is_available else "Not Available"
                print(f"Title: {book.title}, Author: {book.author}, Status: {status}")
                found = True
        if not found:
            print("No books found matching your search criteria.")
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"))
library.add_book(Book("Rich dad poor dad", "Robert Kiyosaki"))
while True:
    print("\nLibrary Menu:")
    print("1. Display Books")
    print("2. Borrow Book")
    print("3. Return Book")
    print("4. Search Books")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == "1":
        library.display_books()
    elif choice == "2":
        book_title = input("Enter the title of the book to borrow: ")
        library.borrow_book(book_title)
    elif choice == "3":
        book_title = input("Enter the title of the book to return: ")
        library.return_book(book_title)
    elif choice == "4":
        search_keyword = input("Enter a keyword to search for books: ")
        library.search_books(search_keyword)
    elif choice == "5":
        print("Exiting the library system. Goodbye!")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output:

```
Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Search Books
5. Exit
Enter your choice (1-5): 4
Enter a keyword to search for books: Rich dad poor dad

Search Results for 'Rich dad poor dad':
Title: Rich dad poor dad, Author: Robert Kiyosaki, Status: Available
```

Task 4:(AI-Based Code Completion for class based Attendance system)

Prompt used:

```
#generate a python code class to mark and display students attendence using loops.
# #Optimize the attendance system by adding percentage calculation.
```

Code:

```python
#generate a python code class to mark and display students attendence using loops.
# #Optimize the attendance system by adding percentage calculation.
class Student:
    def __init__(self, name):
        self.name = name
        self.attendance_record = []

    def mark_attendance(self, present):
        self.attendance_record.append(present)

    def display_attendance(self):
        total_classes = len(self.attendance_record)
        attended_classes = sum(self.attendance_record)
        attendance_percentage = (attended_classes / total_classes) * 100 if total_classes > 0 else 0

        print(f"\nAttendance Record for {self.name}:")
        for i, present in enumerate(self.attendance_record, 1):
            status = "Present" if present else "Absent"
            print(f"Class {i}: {status}")
        print(f"Total Classes: {total_classes}, Attended: {attended_classes}, Attendance Percentage: {attendance_percentage:.2f}%")
student_name = input("Enter student name: ")
student = Student(student_name)
num_classes = int(input("Enter number of classes to mark attendance for: "))
for i in range(num_classes):
    attendance_input = input(f"Was {student_name} present for class {i + 1}? (yes/no): ").strip().lower()
    is_present = attendance_input == "yes"
    student.mark_attendance(is_present)
student.display_attendance()
```

Output:

```
Enter student name: Sushanth
Enter number of classes to mark attendance for: 5
Was Sushanth present for class 1? (yes/no): yes
Was Sushanth present for class 2? (yes/no): yes
Was Sushanth present for class 3? (yes/no): no
Was Sushanth present for class 4? (yes/no): yes
Was Sushanth present for class 5? (yes/no): no

Attendance Record for Sushanth:
Class 1: Present
Class 2: Present
Class 3: Absent
Class 4: Present
Class 5: Absent
Total Classes: 5, Attended: 3, Attendance Percentage: 60.00%
```

Explanation:
This program creates a student class to record attendance for each class.
It uses loops to mark students as present or absent.
And it calculates and displays the attendance percentage.

Task-5:(AI-Based Code Completion for cconditional menu Navigation)

Prompt used:

```
#generate a python code using loops and conditionals to simulate an ATM menu.
# #Optimize the ATM simulation by adding transaction history feature.
```

Code:

```
#generate a python code using loops and conditionals to simulate an ATM menu.
# #Optimize the ATM simulation by adding transaction history feature.
class ATM:
    def __init__(self, initial_balance=0):
        self.balance = initial_balance
        self.transaction_history = []

    def deposit(self, amount):
        self.balance += amount
        self.transaction_history.append(f"Deposited: Rs.{amount:.2f}")

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance.")
        else:
            self.balance -= amount
            self.transaction_history.append(f"Withdrew: Rs.{amount:.2f}")

    def check_balance(self):
        print(f"Current Balance: Rs.{self.balance:.2f}")

    def display_transaction_history(self):
        print("\nTransaction History:")
        if not self.transaction_history:
            print("No transactions yet.")
        else:
            for transaction in self.transaction_history:
                print(transaction)
atm = ATM(1000)  # Initial balance of Rs.1000
while True:
    print("\nATM Menu:")
    print("1. Deposit")
    print("2. Withdraw")
    print("3. Check Balance")
    print("4. Transaction History")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")
```

```python
if choice == "1":
    amount = float(input("Enter amount to deposit: Rs."))
    atm.deposit(amount)
    print(f"Rs.{amount:.2f} deposited successfully.")
elif choice == "2":
    amount = float(input("Enter amount to withdraw: Rs."))
    atm.withdraw(amount)
elif choice == "3":
    atm.check_balance()
elif choice == "4":
    atm.display_transaction_history()
elif choice == "5":
    print("Exiting ATM. Thank you!")
    break
else:
    print("Invalid choice. Please try again.")
```

OUTPUT:

```
ATM Menu:
1. Deposit
2. Withdraw
3. Check Balance
4. Transaction History
5. Exit
Enter your choice (1-5): 1
Enter amount to deposit: Rs.5000
Rs.5000.00 deposited successfully.
```

Explanation:

This program simulates an ATM system using a menu.
Users can deposit money, withdraw money, check balance, and view transaction history.
The program runs continuously until the user chooses to exit.