AI ASSISTED CODING

ASSIGNMENT - 6.5

Name: G. Likhitha Rao

Roll no : 2303A52487

Batch : 35

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.”

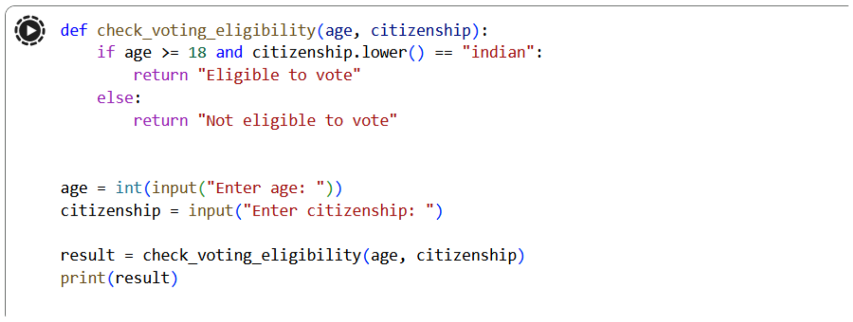
Expected Output:

• AI-generated conditional logic.

• Correct eligibility decisions.

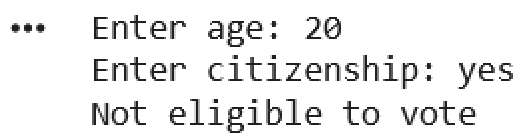
• Explanation of conditions.

CODE:



```
def check_voting_eligibility(age, citizenship):
    if age >= 18 and citizenship.lower() == "indian":
        return "Eligible to vote"
    else:
        return "Not eligible to vote"


age = int(input("Enter age: "))
citizenship = input("Enter citizenship: ")

result = check_voting_eligibility(age, citizenship)
print(result)
```

OUTPUT:



```
•••  Enter age: 20
     Enter citizenship: yes
     Not 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."

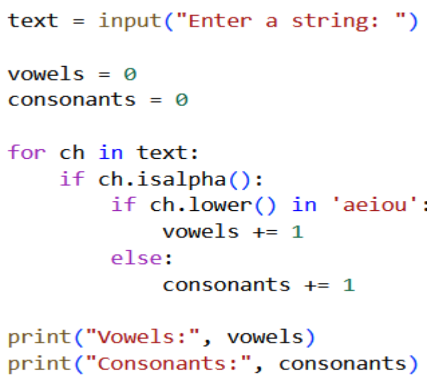
Expected Output:

• AI-generated string processing logic.

• Correct counts.

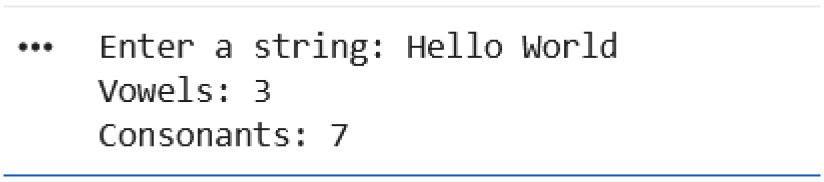
• Output verification.

CODE:



```python
text = input("Enter a string: ")

vowels = 0
consonants = 0

for ch in text:
    if ch.isalpha():
        if ch.lower() in 'aeiou':
            vowels += 1
        else:
            consonants += 1

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

OUTPUT:



```
•••  Enter a string: Hello World
     Vowels: 3
     Consonants: 7
```

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.”

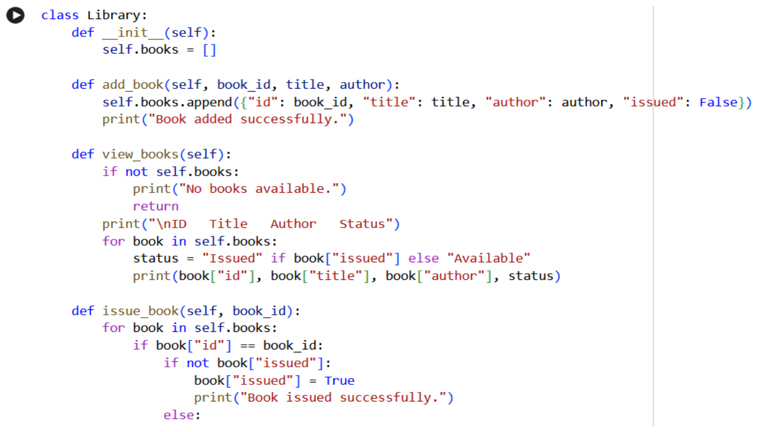
Expected Output:

• Complete AI-generated program.

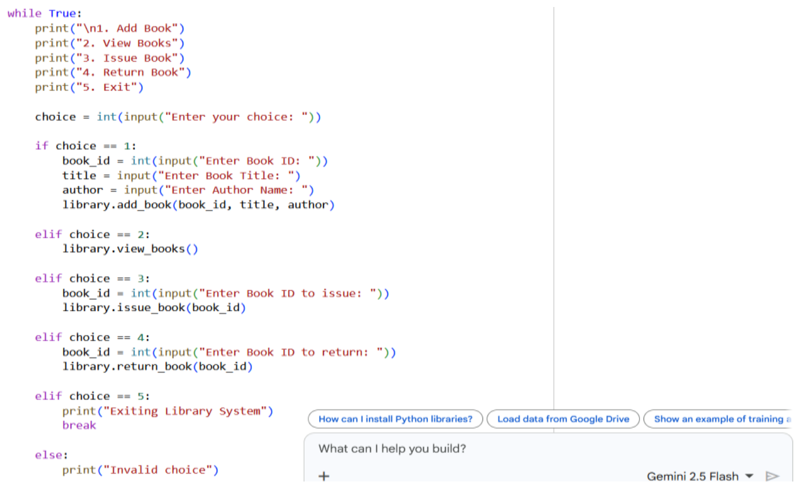
• Review of AI suggestions quality.

• Short reflection on AI-assisted coding experience

CODE:



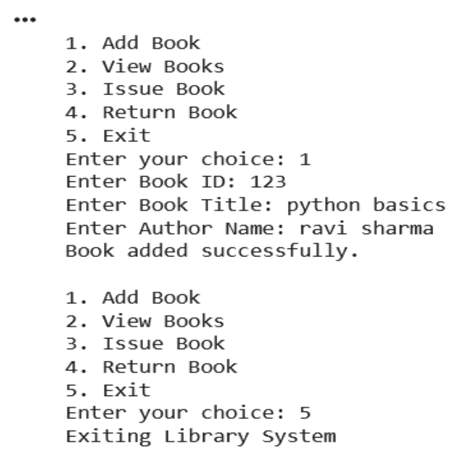
```python
class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book_id, title, author):
        self.books.append({"id": book_id, "title": title, "author": author, "issued": False})
        print("Book added successfully.")

    def view_books(self):
        if not self.books:
            print("No books available.")
            return
        print("\nID   Title   Author   Status")
        for book in self.books:
            status = "Issued" if book["issued"] else "Available"
            print(book["id"], book["title"], book["author"], status)

    def issue_book(self, book_id):
        for book in self.books:
            if book["id"] == book_id:
                if not book["issued"]:
                    book["issued"] = True
                    print("Book issued successfully.")
                else:
```



```python
while True:
    print("\n1. Add Book")
    print("2. View Books")
    print("3. Issue Book")
    print("4. Return Book")
    print("5. Exit")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        book_id = int(input("Enter Book ID: "))
        title = input("Enter Book Title: ")
        author = input("Enter Author Name: ")
        library.add_book(book_id, title, author)

    elif choice == 2:
        library.view_books()

    elif choice == 3:
        book_id = int(input("Enter Book ID to issue: "))
        library.issue_book(book_id)

    elif choice == 4:
        book_id = int(input("Enter Book ID to return: "))
        library.return_book(book_id)

    elif choice == 5:
        print("Exiting Library System")
        break

    else:
        print("Invalid choice")
```

How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
+
Gemini 2.5 Flash

OUTPUT:



```
...
    1. Add Book
    2. View Books
    3. Issue Book
    4. Return Book
    5. Exit
    Enter your choice: 1
    Enter Book ID: 123
    Enter Book Title: python basics
    Enter Author Name: ravi sharma
    Book added successfully.

    1. Add Book
    2. View Books
    3. Issue Book
    4. Return Book
    5. Exit
    Enter your choice: 5
    Exiting Library System
```

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.”

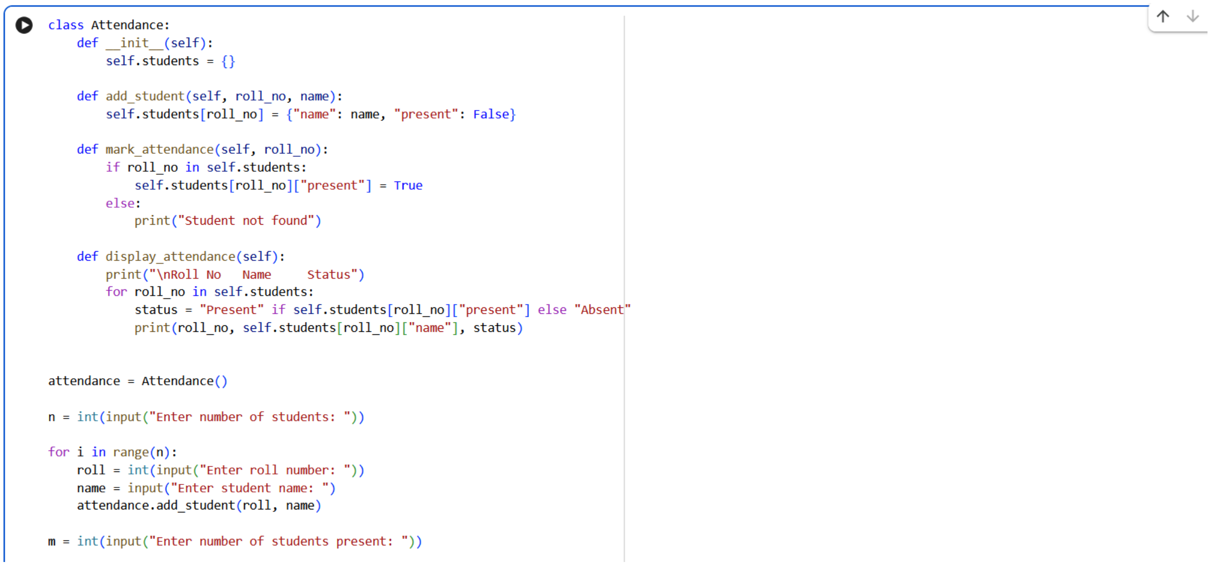
Expected Output:

• AI-generated attendance logic.

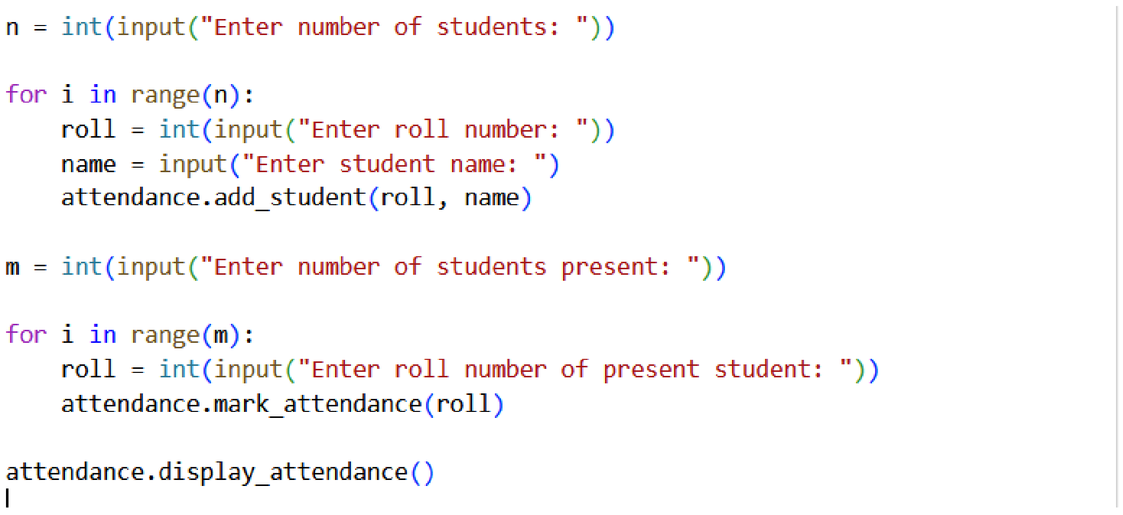
• Correct display of attendance.

• Test cases.

CODE:



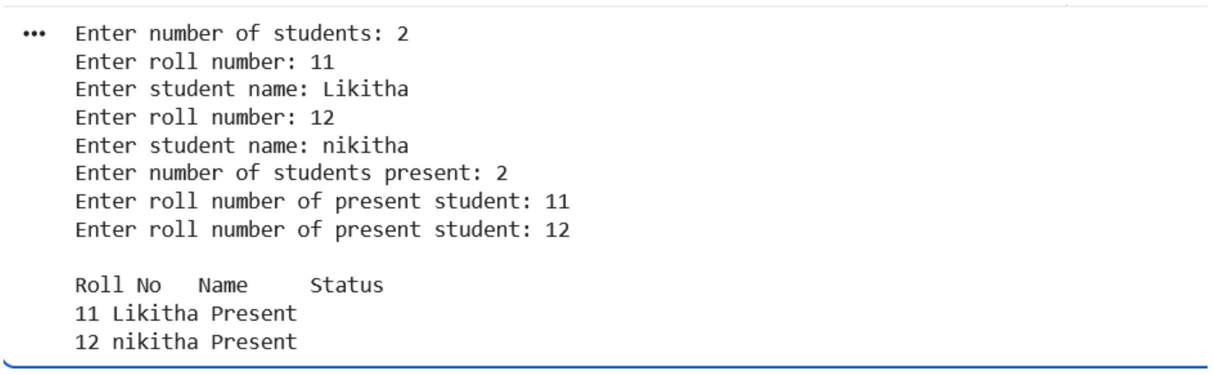
```python
class Attendance:
    def __init__(self):
        self.students = {}

    def add_student(self, roll_no, name):
        self.students[roll_no] = {"name": name, "present": False}

    def mark_attendance(self, roll_no):
        if roll_no in self.students:
            self.students[roll_no]["present"] = True
        else:
            print("Student not found")

    def display_attendance(self):
        print("\nRoll No   Name     Status")
        for roll_no in self.students:
            status = "Present" if self.students[roll_no]["present"] else "Absent"
            print(roll_no, self.students[roll_no]["name"], status)


attendance = Attendance()

n = int(input("Enter number of students: "))

for i in range(n):
    roll = int(input("Enter roll number: "))
    name = input("Enter student name: ")
    attendance.add_student(roll, name)

m = int(input("Enter number of students present: "))
```



```python
n = int(input("Enter number of students: "))

for i in range(n):
    roll = int(input("Enter roll number: "))
    name = input("Enter student name: ")
    attendance.add_student(roll, name)

m = int(input("Enter number of students present: "))
```

```python
for i in range(m):
    roll = int(input("Enter roll number of present student: "))
    attendance.mark_attendance(roll)

attendance.display_attendance()
```

OUTPUT:



```
Enter number of students: 2
Enter roll number: 11
Enter student name: Likitha
Enter roll number: 12
Enter student name: nikitha
Enter number of students present: 2
Enter roll number of present student: 11
Enter roll number of present student: 12

Roll No    Name      Status
11 Likitha Present
12 nikitha Present
```

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.”

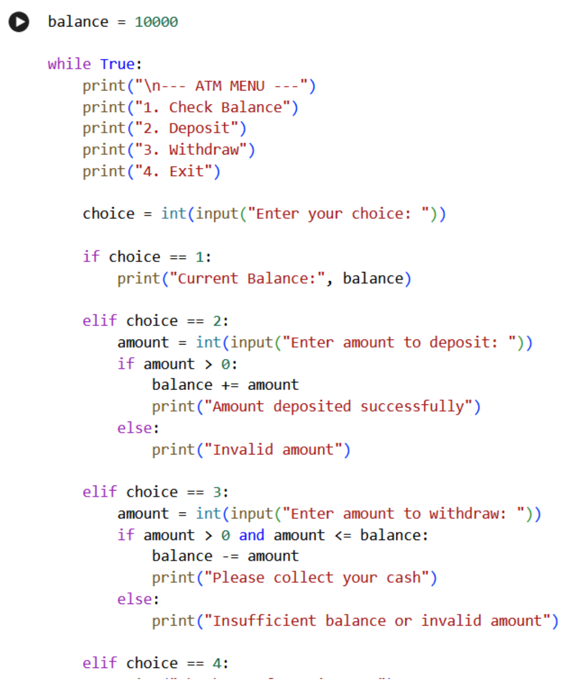
Expected Output:

• AI-generated menu logic.

• Correct option handling.

• Output verification.

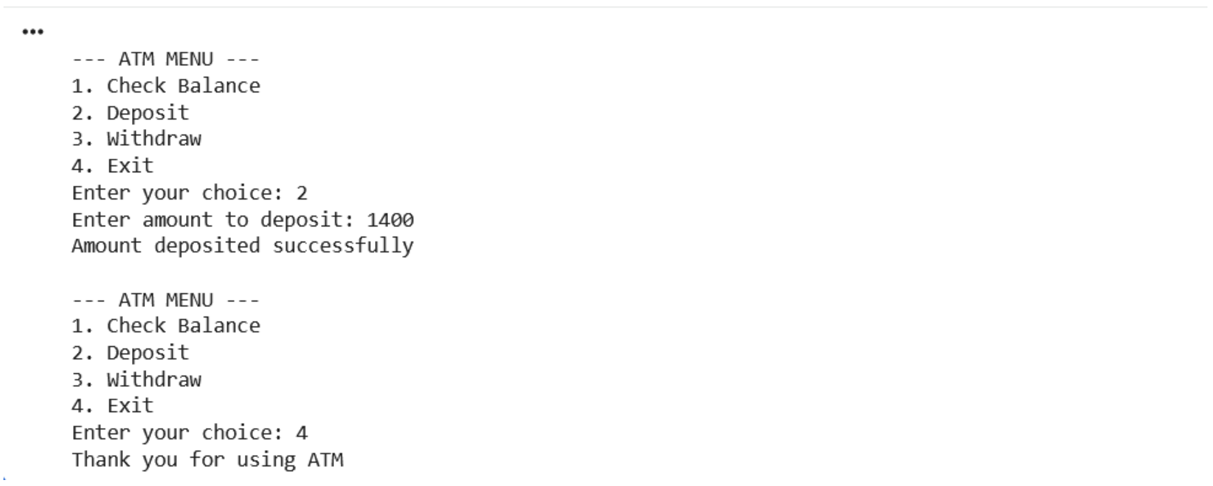
CODE:



```python
balance = 10000

while True:
    print("\n--- ATM MENU ---")
    print("1. Check Balance")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Exit")

    choice = int(input("Enter your choice: "))

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

    elif choice == 2:
        amount = int(input("Enter amount to deposit: "))
        if amount > 0:
            balance += amount
            print("Amount deposited successfully")
        else:
            print("Invalid amount")

    elif choice == 3:
        amount = int(input("Enter amount to withdraw: "))
        if amount > 0 and amount <= balance:
            balance -= amount
            print("Please collect your cash")
        else:
            print("Insufficient balance or invalid amount")

    elif choice == 4:
```

```python
    elif choice == 3:
        amount = int(input("Enter amount to withdraw: "))
        if amount > 0 and amount <= balance:
            balance -= amount
            print("Please collect your cash")
        else:
            print("Insufficient balance or invalid amount")

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

    else:
        print("Invalid choice")
```

OUTPUT:



```
--- ATM MENU ---
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 2
Enter amount to deposit: 1400
Amount deposited successfully

--- ATM MENU ---
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 4
Thank you for using ATM
```