School of Computer Science and Artificial Intelligence

Lab Assignment # 6.5

Program : B. Tech (CSE)

Specialization : AIML

Course Title : AI Assisted
Coding Course Code:

23CS002PC304

Semester : VI

Academic Session : 2025-2026

Name of Student : R. Deekshith Goud
Enrollment No. :
2303A52104 Batch No. : 33

Date : 30/01/26

Learning Objectives

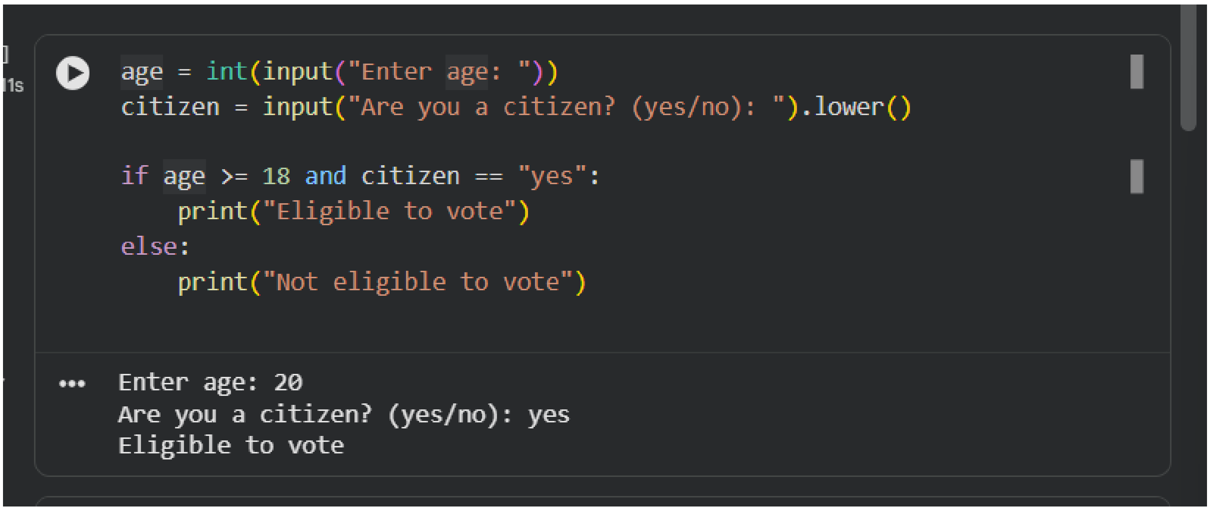
- **LO1:** Use AI-based code completion tools to generate Python code involving classes, loops, and conditionals.
- **LO2:** Interpret and explain AI-generated code line-by-line.
- **LO3:** Identify errors, inefficiencies, or logical flaws in AI-suggested implementations.
- **LO4:** Optimize AI-generated code for better readability and performance.
- **LO5:** Demonstrate ethical and responsible use of AI tools in coding tasks.

Task Description #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
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")
```

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

Explanation

- age >= 18 → checks minimum voting age
- citizen == "yes" → ensures citizenship
- and → both conditions must be true
- Output correctly displays eligibility

Verification

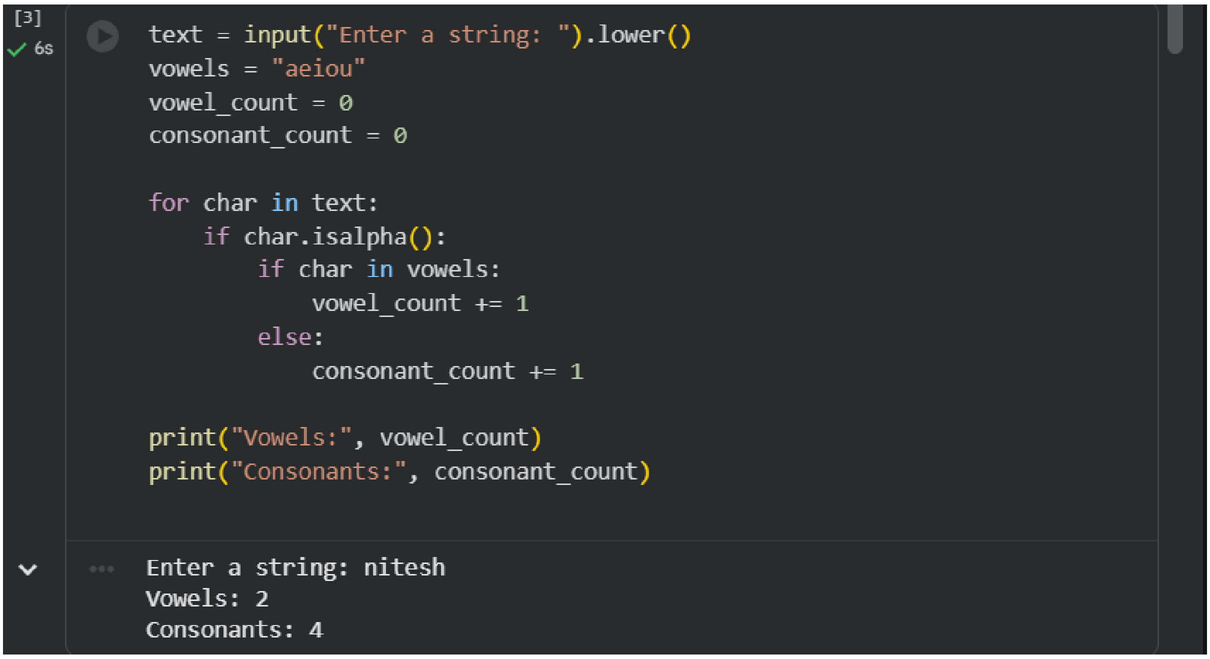
Age	Citizenship	Output
20	yes	Eligible
16	yes	Not Eligible
25	no	Not Eligible

Task Description #2: AI-Based Code Completion for Loop-Based String Processing

Prompt Used

“Generate Python code to count vowels and consonants in a string using a loop.”

CODE:



```
text = input("Enter a string: ").lower()
vowels = "aeiou"
vowel_count = 0
consonant_count = 0

for char in text:
    if char.isalpha():
        if char in vowels:
            vowel_count += 1
        else:
            consonant_count += 1

print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
```

```
Enter a string: nitesh
Vowels: 2
Consonants: 4
```

Explanation

- for char in text → loops through characters
- isalpha() → ignores spaces and symbols
- Counts vowels and consonants separately

Output Verification

Input: Hello World

Output:

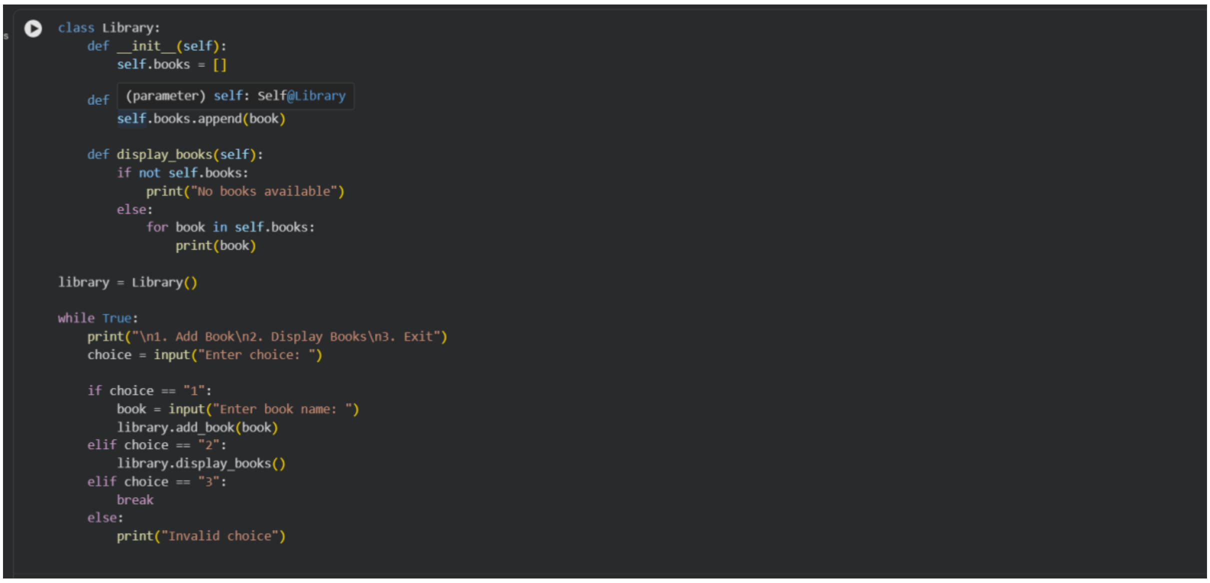
Vowels: 3
Consonants: 7

Task Description #3: AI-Assisted Code Completion Reflection Task

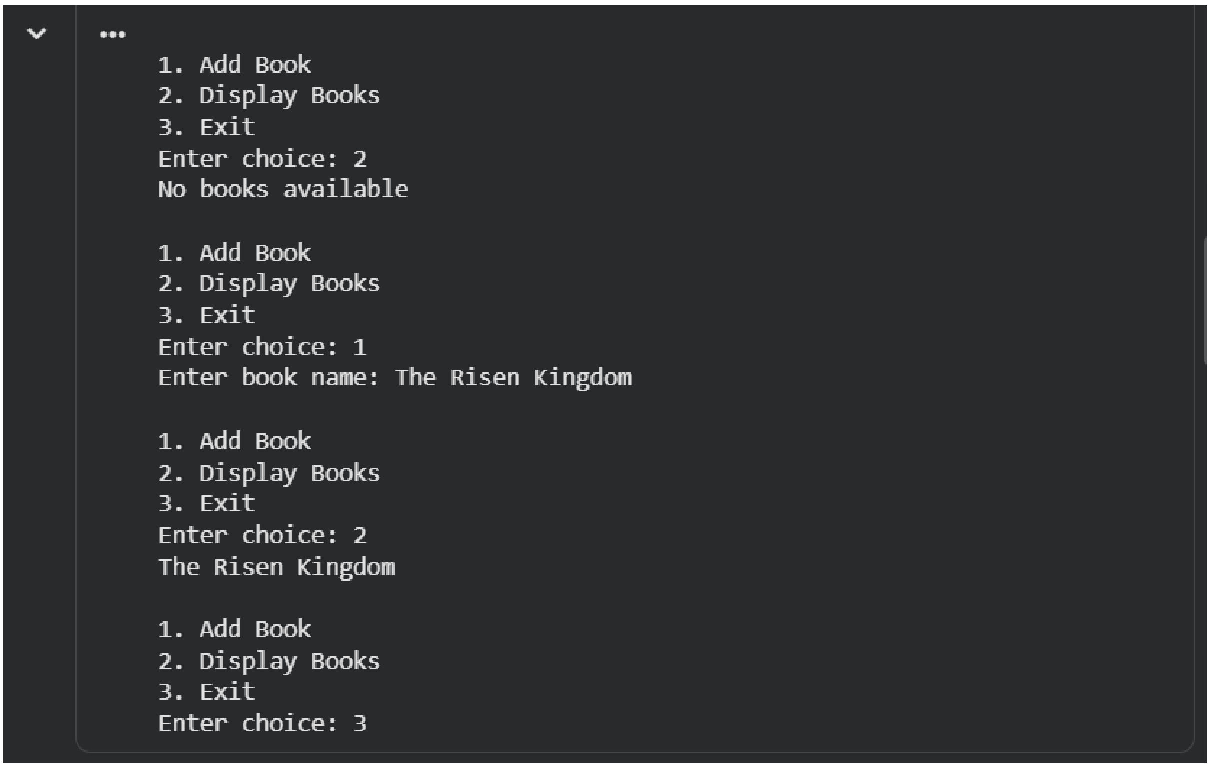
Prompt Used

"Generate a Python program for a library management system using classes, loops, and conditional statements."

CODE:



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

    def (parameter) self: Self@Library
        self.books.append(book)

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

library = Library()

while True:
    print("\n1. Add Book\n2. Display Books\n3. Exit")
    choice = input("Enter choice: ")

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



```
1. Add Book
2. Display Books
3. Exit
Enter choice: 2
No books available

1. Add Book
2. Display Books
3. Exit
Enter choice: 1
Enter book name: The Risen Kingdom

1. Add Book
2. Display Books
3. Exit
Enter choice: 2
The Risen Kingdom

1. Add Book
2. Display Books
3. Exit
Enter choice: 3
```

Review of AI Suggestions

- Correct use of **class**
- Uses **loop & conditionals**
- Easy to understand
- No delete/search feature (can be improved)

Reflection on AI-Assisted Coding

AI-based code completion helps in quickly generating working code structures. However, the programmer must review logic, handle edge cases, and optimize performance. AI should be used as a **support tool**, not a replacement for logical thinking.

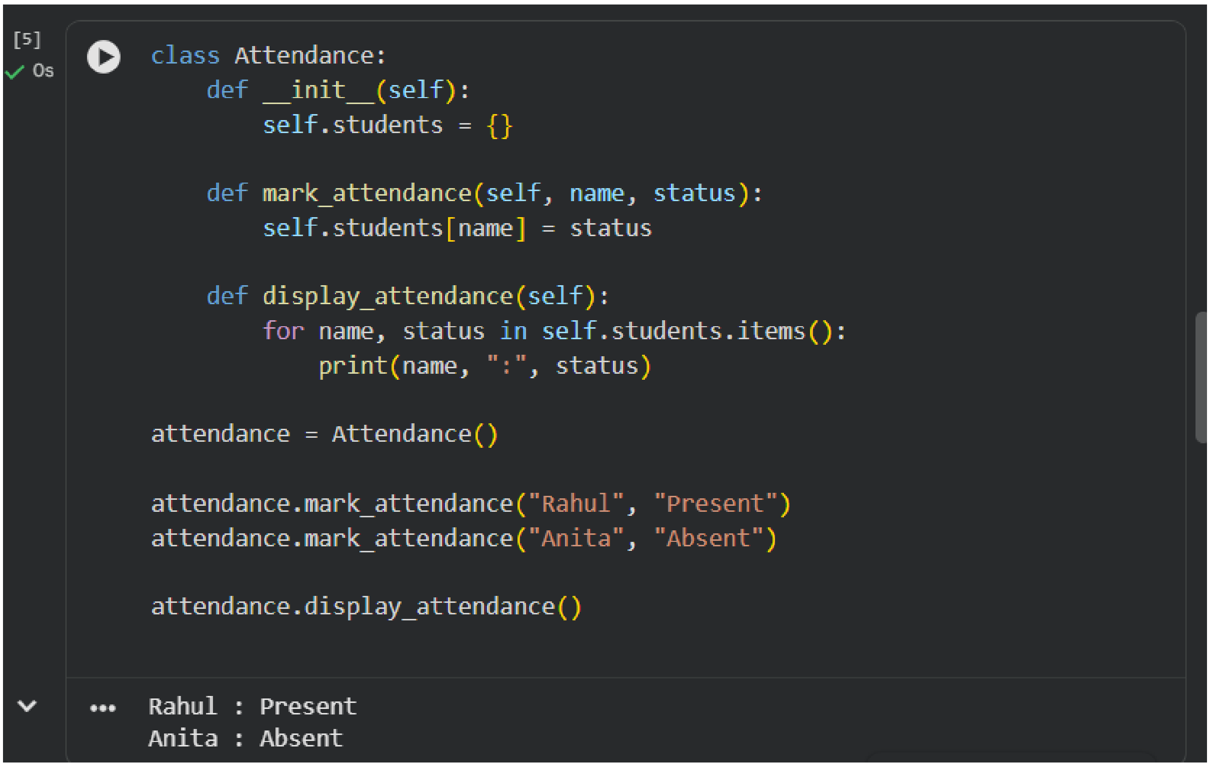
Task Description #4: AI-Assisted Code Completion for Class-Based

Attendance System

Prompt Used

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

CODE:



```
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)

attendance = Attendance()

attendance.mark_attendance("Rahul", "Present")
attendance.mark_attendance("Anita", "Absent")

attendance.display_attendance()
```

```
Rahul : Present
Anita : Absent
```

Test Case

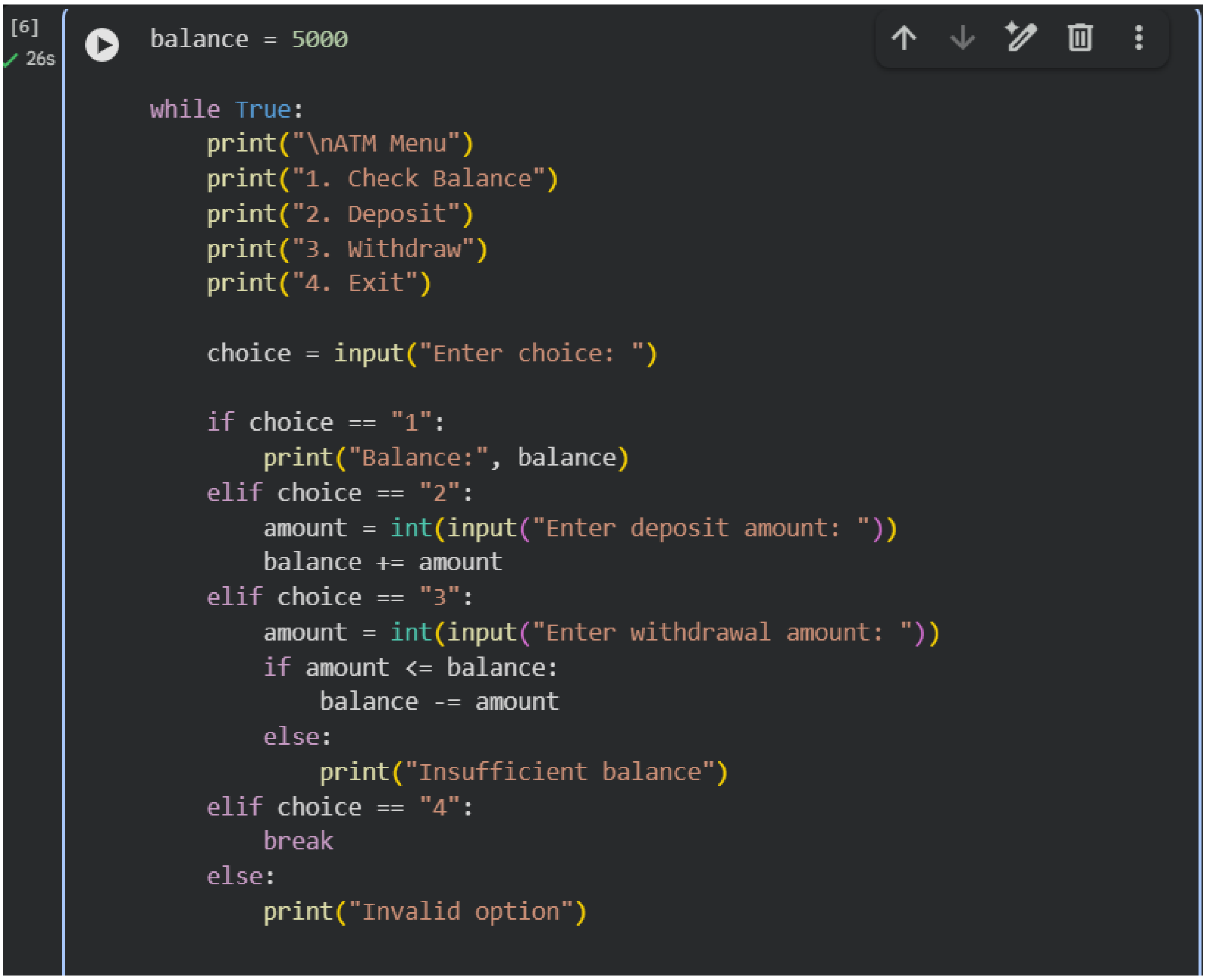
- Adding new student → ✓ Works
- Display attendance → ✓ Correct

Task Description #5: AI-Based Code Completion for Conditional Menu Navigation

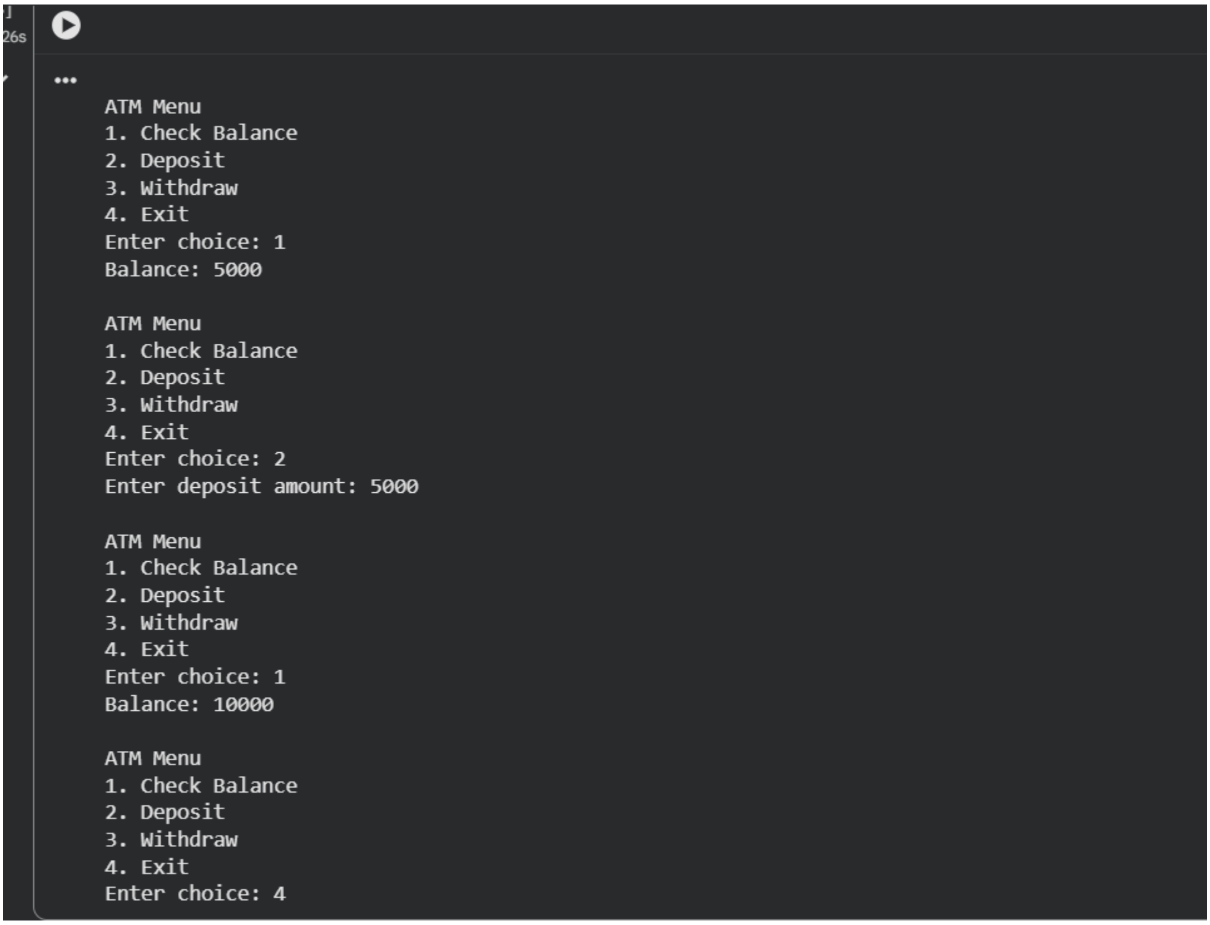
Prompt Used

“Generate a Python program using loops and conditionals to simulate an ATM menu.”

CODE:



```python
balance = 5000

while True:
    print("\nATM Menu")
    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 withdrawal amount: "))
        if amount <= balance:
            balance -= amount
        else:
            print("Insufficient balance")
    elif choice == "4":
        break
    else:
        print("Invalid option")
```



```
ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 1
Balance: 5000

ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 2
Enter deposit amount: 5000

ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 1
Balance: 10000

ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 4
```

Output Verification

- Balance check → ✓
- Deposit → ✓
- Withdraw with limit → ✓

Ethical & Responsible Use of AI (LO5)

- AI-generated code was **reviewed and understood**
- Logic errors and improvements were identified
- AI used as a **learning aid**, not for blind copying
- Code explanations ensured conceptual clarity

Conclusion

This experiment demonstrated the effective use of **AI-based code completion** for Python programming involving **classes, loops, and conditionals**. AI tools improve productivity but must be used responsibly with human judgment.