

AI ASSISTED CODING LAB - 6.5

2303A52197

G.Rishika

Batch – 35

TASK 1:

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:

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

  if age >= 18 and citizenship == "yes":
      print("You are eligible to vote.")
  else:
      print("You are not eligible to vote.")
```

OUTPUT:

```
Enter your age: 18
Are you a citizen? (yes/no): yes
You are eligible to vote.
```

TASK - 2

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:

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

vowels = 0
consonants = 0

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

print("Number of vowels:", vowels)
print("Number of consonants:", consonants)
```

OUTPUT:

```
print("Number of vowels:", vowels)
print("Number of consonants:", consonants)

... Enter a string: ALPHABETS
Number of vowels: 3
Number of consonants: 6
```

TASK – 3

Task Description #3 (AI-Assisted Code Completion Reflection

Task)

Task: Use an AI tool to generate a complete program using classes, loops, and conditionals.

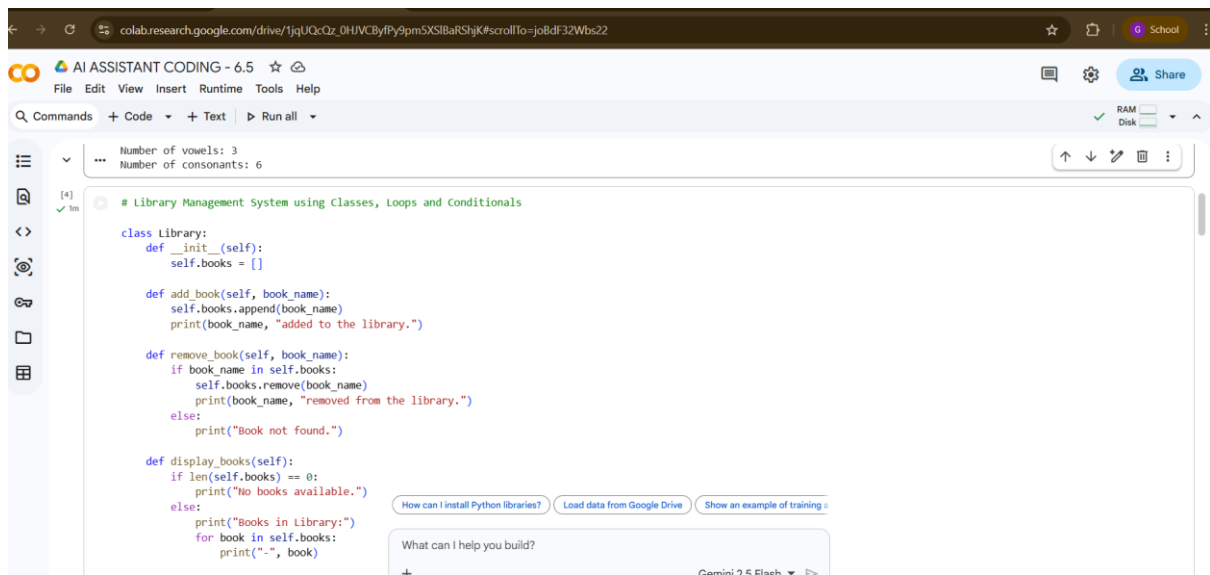
Prompt:

“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:



The screenshot shows the Google Colab AI Assistant Coding interface. The top bar includes the Google Assistant logo, the text "AI ASSISTANT CODING - 6.5", and a "Share" button. Below the bar is a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The main area displays a Python class named "Library" with methods for adding, removing, and displaying books. The class is defined as follows:

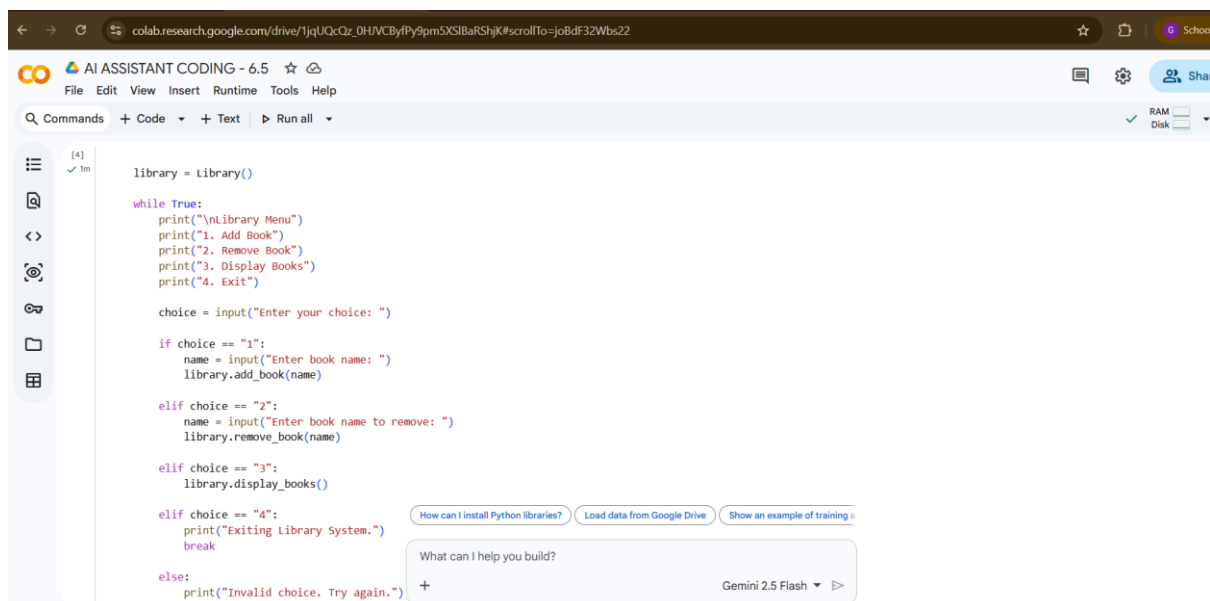
```
class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book_name):
        self.books.append(book_name)
        print(book_name, "added to the library.")

    def remove_book(self, book_name):
        if book_name in self.books:
            self.books.remove(book_name)
            print(book_name, "removed from the library.")
        else:
            print("Book not found.")

    def display_books(self):
        if len(self.books) == 0:
            print("No books available.")
        else:
            print("Books in Library:")
            for book in self.books:
                print("-", book)
```

Below the code, there are three buttons: "How can I install Python libraries?", "Load data from Google Drive", and "Show an example of training". A text input field with the placeholder "What can I help you build?" is also visible, along with a "Gemini 2.5 Flash" dropdown menu.



The screenshot shows the Google Colab AI Assistant Coding interface. The top bar includes the Google Assistant logo, the text "AI ASSISTANT CODING - 6.5", and a "Share" button. Below the bar is a menu with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The main area displays a Python script that creates a "Library" object and runs a menu loop. The script is as follows:

```
library = Library()

while True:
    print("\nLibrary Menu")
    print("1. Add Book")
    print("2. Remove Book")
    print("3. Display Books")
    print("4. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        name = input("Enter book name: ")
        library.add_book(name)

    elif choice == "2":
        name = input("Enter book name to remove: ")
        library.remove_book(name)

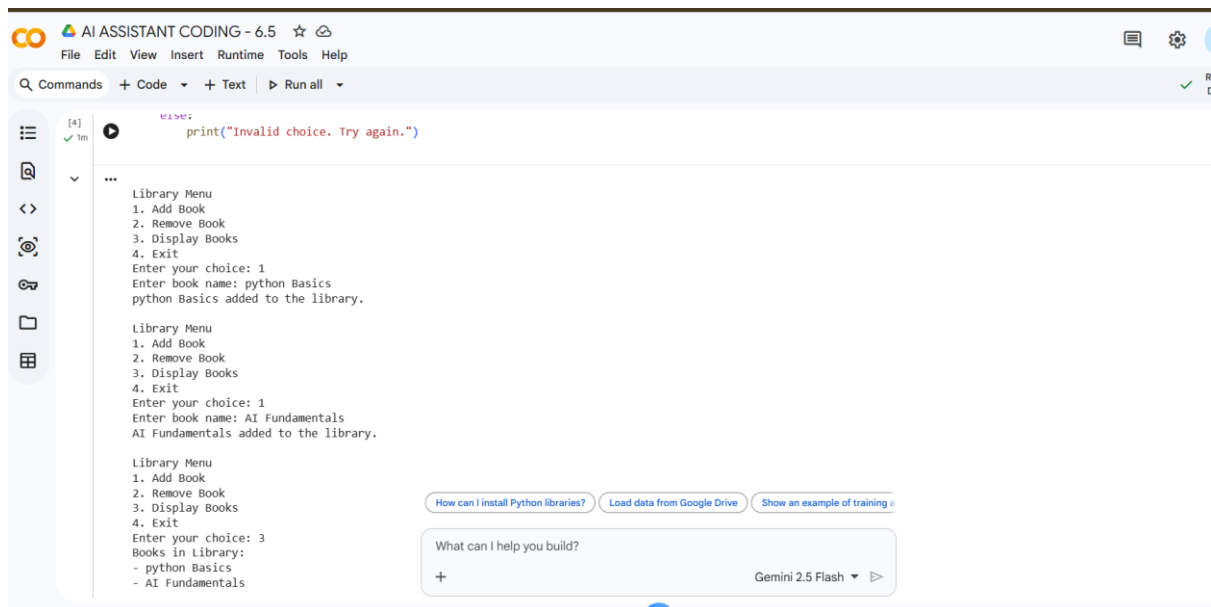
    elif choice == "3":
        library.display_books()

    elif choice == "4":
        print("Exiting Library System.")
        break

    else:
        print("Invalid choice. Try again.")
```

Below the code, there are three buttons: "How can I install Python libraries?", "Load data from Google Drive", and "Show an example of training". A text input field with the placeholder "What can I help you build?" is also visible, along with a "Gemini 2.5 Flash" dropdown menu.

OUTPUT:



The screenshot shows the AI Assistant Coding interface. The top bar includes the logo, version (6.5), and menu items (File, Edit, View, Insert, Runtime, Tools, Help). Below the bar is a search bar and tabs for Commands, Code, Text, and Run all. The main area displays a Python script with a loop that prompts the user to enter a choice. The output shows the script running and displaying the library menu, adding books, and handling invalid choices. The interface also includes a sidebar with icons for file operations and a bottom bar with a chat window and a Gemini 2.5 Flash model selector.

```
[4] ✓ 1m
python: print("Invalid choice. Try again.")

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 1
Enter book name: python Basics
python Basics added to the library.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 1
Enter book name: AI Fundamentals
AI Fundamentals added to the library.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 3
Books in Library:
- python Basics
- AI Fundamentals

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: python Basics
Invalid choice. Try again.

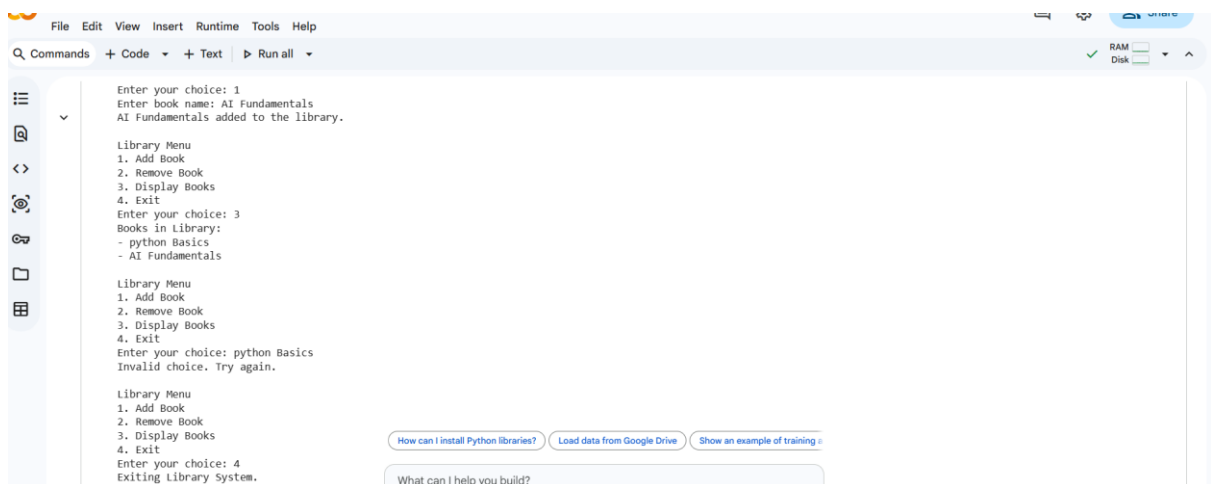
Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 4
Exiting Library System.
```

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 ▶



The screenshot shows the AI Assistant Coding interface. The top bar includes the logo, version (6.5), and menu items (File, Edit, View, Insert, Runtime, Tools, Help). Below the bar is a search bar and tabs for Commands, Code, Text, and Run all. The main area displays a Python script with a loop that prompts the user to enter a choice. The output shows the script running and displaying the library menu, adding books, and handling invalid choices. The interface also includes a sidebar with icons for file operations and a bottom bar with a chat window and a Gemini 2.5 Flash model selector.

```
Enter your choice: 1
Enter book name: AI Fundamentals
AI Fundamentals added to the library.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 3
Books in Library:
- python Basics
- AI Fundamentals

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: python Basics
Invalid choice. Try again.

Library Menu
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice: 4
Exiting Library System.
```

How can I install Python libraries? Load data from Google Drive Show an example of training

What can I help you build?

TASK – 4

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:



The screenshot shows the AI Assistant Coding interface. The top bar includes the logo, 'AI ASSISTANT CODING - 6.5', and icons for file, edit, view, insert, runtime, tools, and help. Below the bar is a search bar with 'Commands', '+ Code', '+ Text', and 'Run all'. The main area displays a Python class named 'AttendanceSystem'. The class has three methods: 'mark_attendance' which takes a number of students and marks their attendance, 'display_attendance' which shows the current attendance, and a 'while True' loop at the bottom for testing. The interface also features a sidebar with icons for file explorer, search, and other tools. At the bottom, there is a chat area with a prompt 'What can I help you build?' and a 'Gemini 2.5 Flash' model selector.

```
[s]
✓ 1m
class AttendanceSystem:
    def __init__(self):
        self.attendance = {} # Stores student name and status

    def mark_attendance(self):
        n = int(input("Enter number of students to mark attendance: "))
        for i in range(n):
            name = input(f"Enter name of student {i+1}: ")
            status = input(f"Is {name} present? (yes/no): ").lower()
            if status == "yes":
                self.attendance[name] = "Present"
            else:
                self.attendance[name] = "Absent"
        print("\nAttendance marked successfully!\n")

    def display_attendance(self):
        if not self.attendance:
            print("No attendance data available.")
        else:
            print("Student Attendance:")
            for student, status in self.attendance.items():
                print(f"{student}: {status}")

# Create object of AttendanceSystem
attendance_system = AttendanceSystem()

while True:
```

```
colab.research.google.com/drive/1jqUQzOz_0HVCByfPy9pm5XSIbaRShjKf?scrollTo=joBdf32Wbs22

AI ASSISTANT CODING - 6.5
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

# Create object of AttendanceSystem
attendance_system = AttendanceSystem()

while True:
    print("\nAttendance Menu")
    print("1. Mark Attendance")
    print("2. Display Attendance")
    print("3. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        attendance_system.mark_attendance()
    elif choice == "2":
        attendance_system.display_attendance()
    elif choice == "3":
        print("Exiting Attendance System.")
        break
    else:
        print("Invalid choice. Try again.")
```

OUTPUT:

```
AI ASSISTANT CODING - 6.5
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

break
else:
    print("Invalid choice. Try again.")

Attendance Menu
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 1
Enter number of students to mark attendance: 3
Enter name of student 1: priya
Is priya present? (yes/no): no
Enter name of student 2: riya
Is riya present? (yes/no): yes
Enter name of student 3: gita
Is gita present? (yes/no): yes

Attendance marked successfully!

Attendance Menu
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 2
Student Attendance:
priya: Absent
riya: Present
gita: Present
```

```
AI ASSISTANT CODING - 6.5
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

Attendance Menu
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 2
Student Attendance:
priya: Absent
riya: Present
gita: Present

Attendance Menu
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 3
Exiting Attendance System.
```

TASK – 5

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

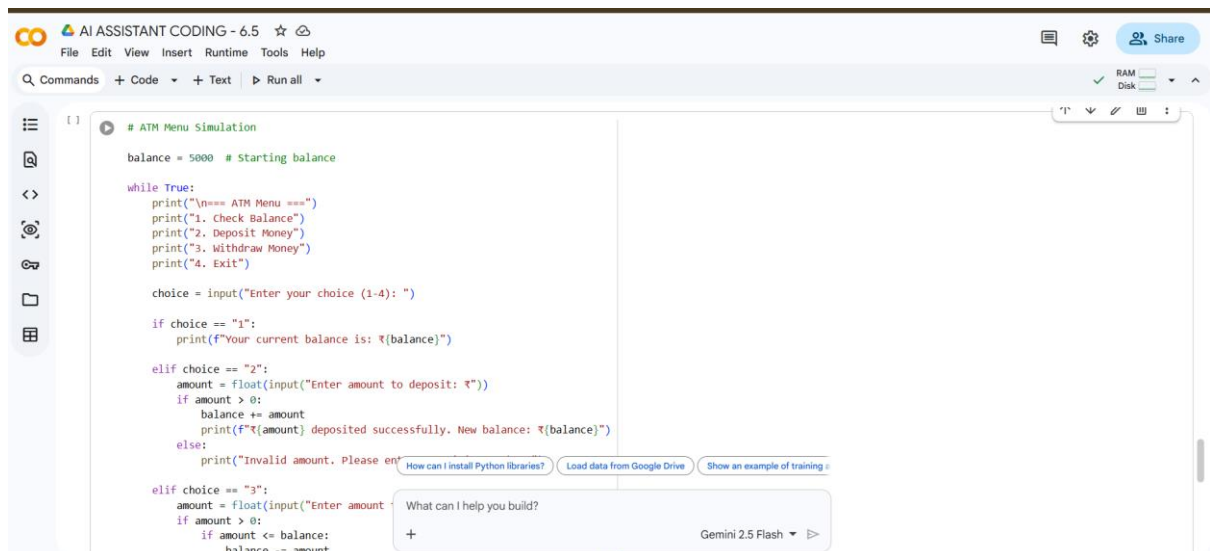
Task: Use an AI tool to complete a navigation menu.

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:



The screenshot displays the AI Assistant Coding interface (version 6.5) with a Python program for an ATM menu simulation. The code is as follows:

```
# ATM Menu Simulation

balance = 5000 # Starting balance

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

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

    if choice == "1":
        print(f"Your current balance is: ₹{balance}")

    elif choice == "2":
        amount = float(input("Enter amount to deposit: ₹"))
        if amount > 0:
            balance += amount
            print(f"₹{amount} deposited successfully. New balance: ₹{balance}")
        else:
            print("Invalid amount. Please enter a positive value.")

    elif choice == "3":
        amount = float(input("Enter amount to withdraw: ₹"))
        if amount > 0:
            if amount <= balance:
                balance -= amount
                print(f"₹{amount} withdrawn successfully. New balance: ₹{balance}")
            else:
                print("Insufficient balance. Please enter an amount less than or equal to your current balance.")
        else:
            print("Invalid amount. Please enter a positive value.")

    elif choice == "4":
        print("Exiting the ATM menu. Goodbye!")
        break

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

The interface includes a sidebar with icons for file explorer, search, and other tools. The top bar shows the AI Assistant Coding logo, version 6.5, and a share button. The bottom bar displays the prompt "What can I help you build?" and the Gemini 2.5 Flash model name.


```
colab.research.google.com/drive/1jqUQcQz_0HjVCByfPy9pm5XSIBaRShjK#scrollTo=QSBiDrRNd10

AI ASSISTANT CODING - 6.5
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[ ]
balance -= amount
print(f"₹{amount} withdrawn successfully. New balance: ₹{balance}")
else:
    print("Insufficient balance.")
else:
    print("Invalid amount. Please enter a positive number.")

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

else:
    print("Invalid choice. Please select a number between 1 and 4.")
```

OUTPUT:

```
colab.research.google.com/drive/1jqUQcQz_0HjVCByfPy9pm5XSIBaRShjK#scrollTo=H1nqWlUSe-DK

AI ASSISTANT CODING - 6.5
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

RAM
Disk

...
=== ATM Menu ===
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 1
Your current balance is: ₹5000

=== ATM Menu ===
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 2
Enter amount to deposit: ₹5000
₹5000.0 deposited successfully. New balance: ₹10000.0

=== ATM Menu ===
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice (1-4): 3
Enter amount to withdraw: ₹3000
₹3000.0 withdrawn successfully. New balance: ₹7000.0

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
```

3. Withdraw Money

4. Exit

Enter your choice (1-4): 4

Thank you for using the ATM. Goodbye!

