AI ASSIGNMENT-6.5

Name : B.Harshini

Hall No. : 2303A52242

Batch : 36

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

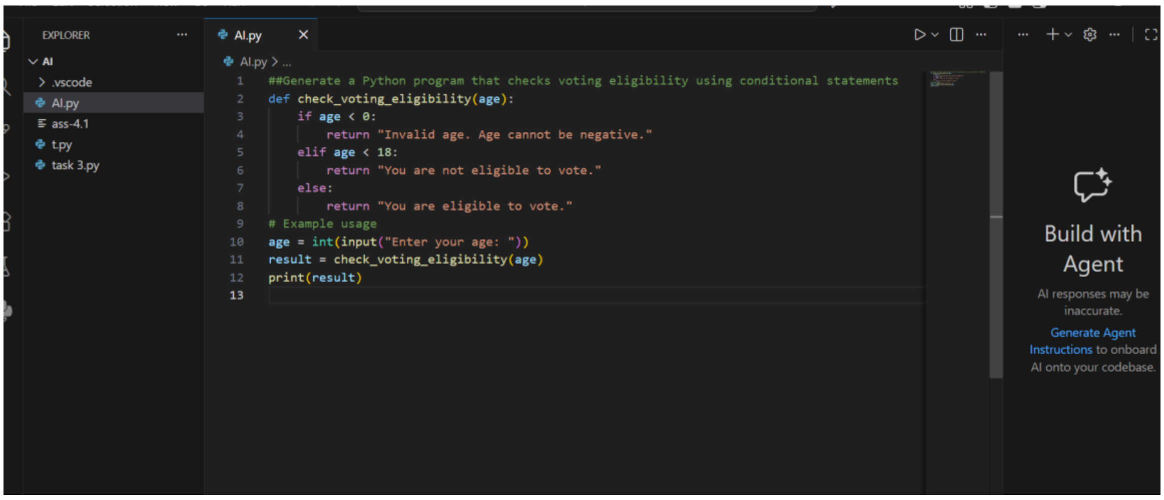
citizenship.”

Expected Output:

• AI-generated conditional logic.

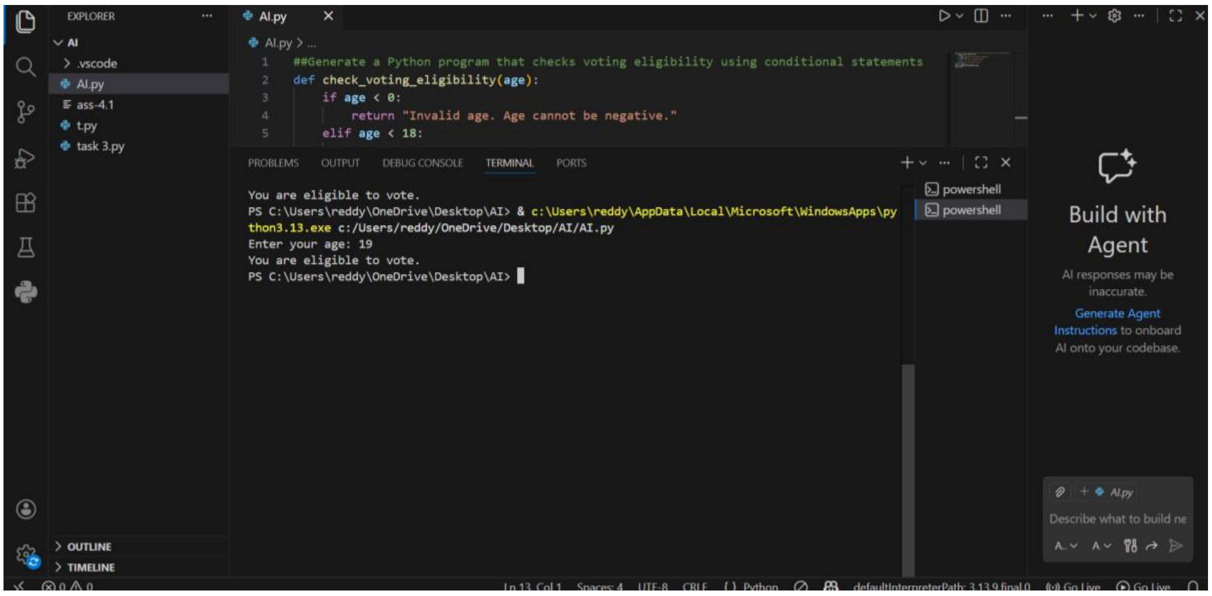
• Correct eligibility decisions.

• Explanation of conditions.


EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
AI.py
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

```python
##Generate a Python program that checks voting eligibility using conditional statements
def check_voting_eligibility(age):
    if age < 0:
        return "Invalid age. Age cannot be negative."
    elif age < 18:
        return "You are not eligible to vote."
    else:
        return "You are eligible to vote."
# Example usage
age = int(input("Enter your age: "))
result = check_voting_eligibility(age)
print(result)
```

Output:



```python
##Generate a Python program that checks voting eligibility using conditional statements
def check_voting_eligibility(age):
    if age < 0:
        return "Invalid age. Age cannot be negative."
    elif age < 18:
```

```
You are eligible to vote.
PS C:\Users\reddy\OneDrive\Desktop\AI> & c:\Users\reddy\AppData\Local\Microsoft\WindowsApps\py
thon3.13.exe c:/Users/reddy/OneDrive/Desktop/AI/AI.py
Enter your age: 19
You are eligible to vote.
PS C:\Users\reddy\OneDrive\Desktop\AI>
```

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

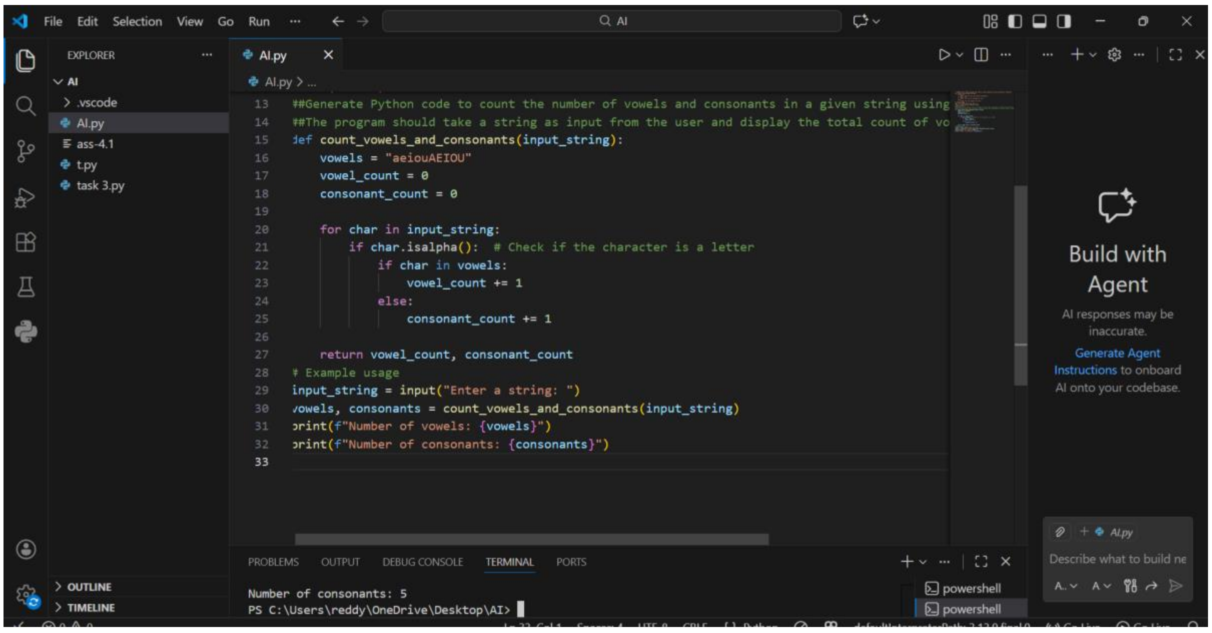
using a loop.”

Expected Output:

• AI-generated string processing logic.

• Correct counts.

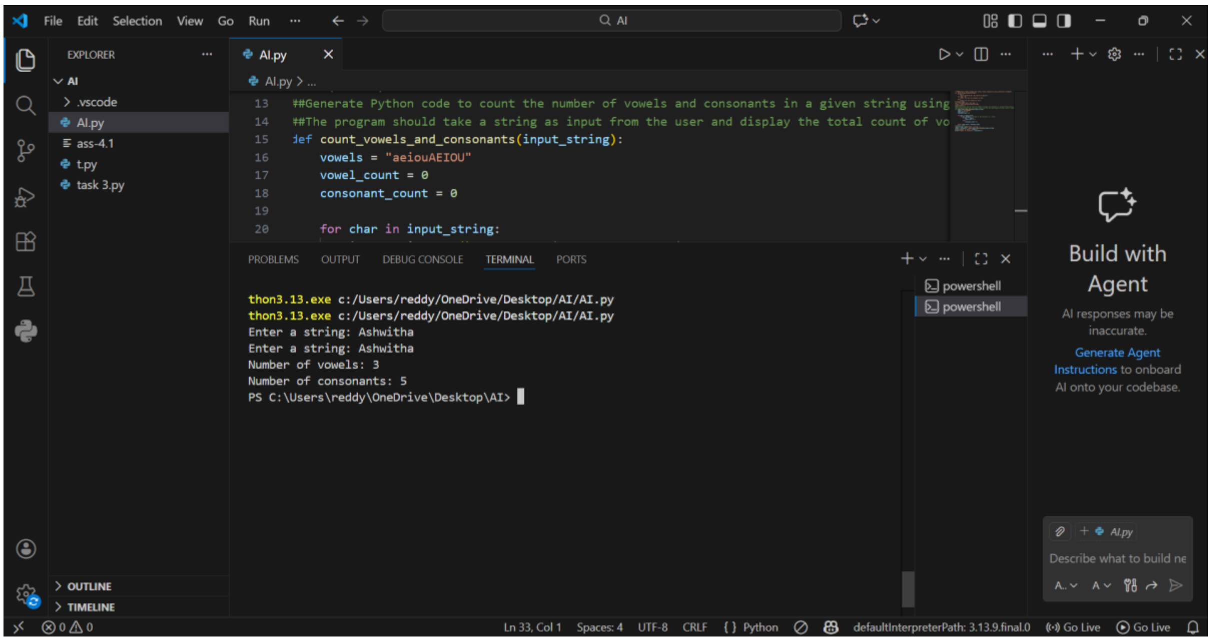
• Output verification.



```python
##Generate Python code to count the number of vowels and consonants in a given string using
##The program should take a string as input from the user and display the total count of vo
def count_vowels_and_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
        if char.isalpha():  # Check if the character is a letter
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count
# Example usage
input_string = input("Enter a string: ")
vowels, consonants = count_vowels_and_consonants(input_string)
print(f"Number of vowels: {vowels}")
print(f"Number of consonants: {consonants}")
```

```
Number of consonants: 5
PS C:\Users\reddy\OneDrive\Desktop\AI>
```

Output:



```
##Generate Python code to count the number of vowels and consonants in a given string using
##The program should take a string as input from the user and display the total count of vo
def count_vowels_and_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
```

```
thon3.13.exe c:/Users/reddy/OneDrive/Desktop/AI/AI.py
thon3.13.exe c:/Users/reddy/OneDrive/Desktop/AI/AI.py
Enter a string: Ashwitha
Enter a string: Ashwitha
Number of vowels: 3
Number of consonants: 5
PS C:\Users\reddy\OneDrive\Desktop\AI>
```

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

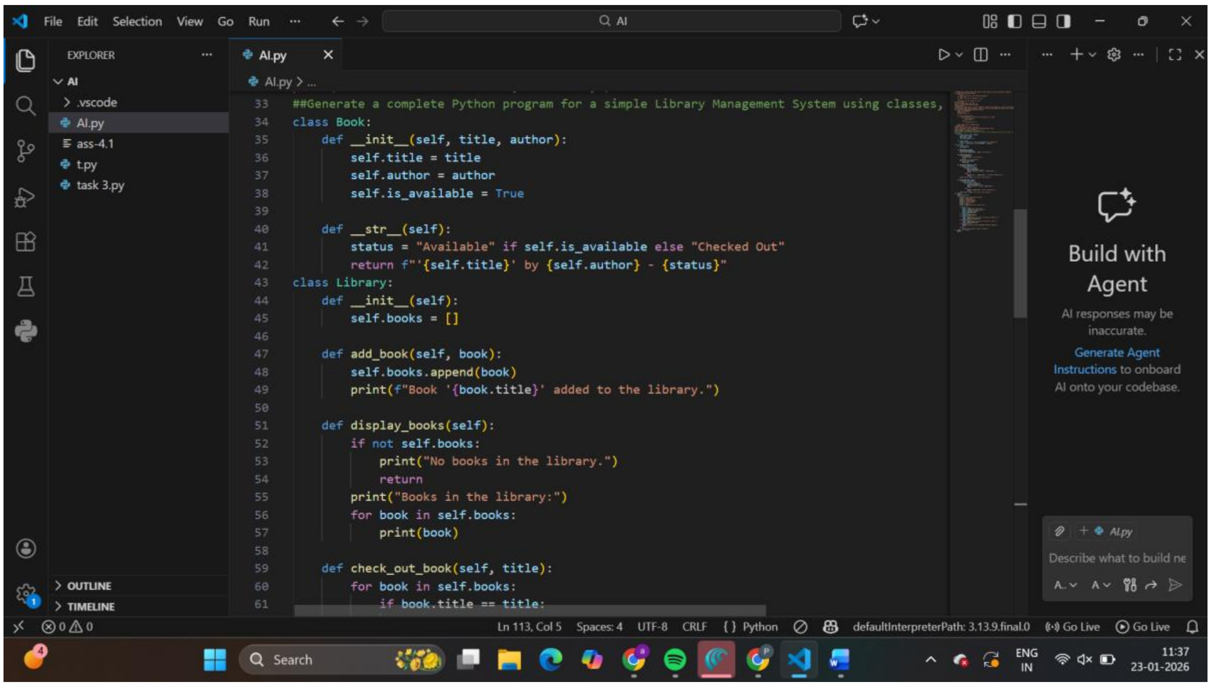
using classes, loops, and conditional statements.”

Expected Output:

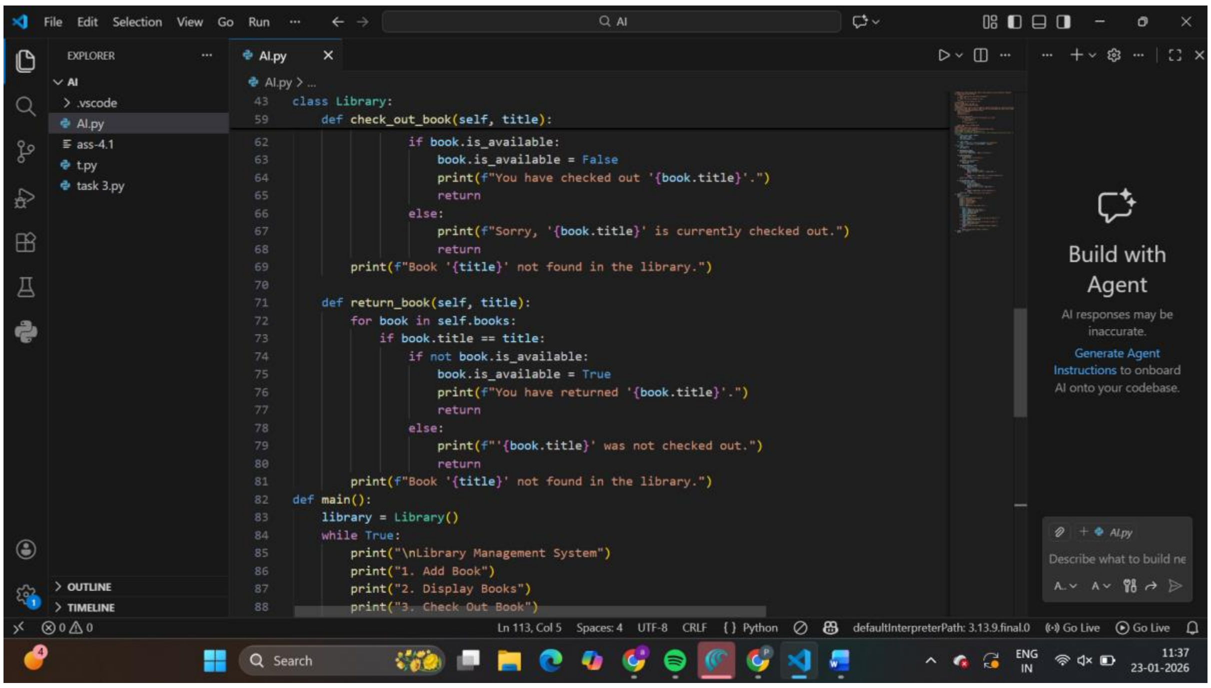
• Complete AI-generated program.

• Review of AI suggestions quality.

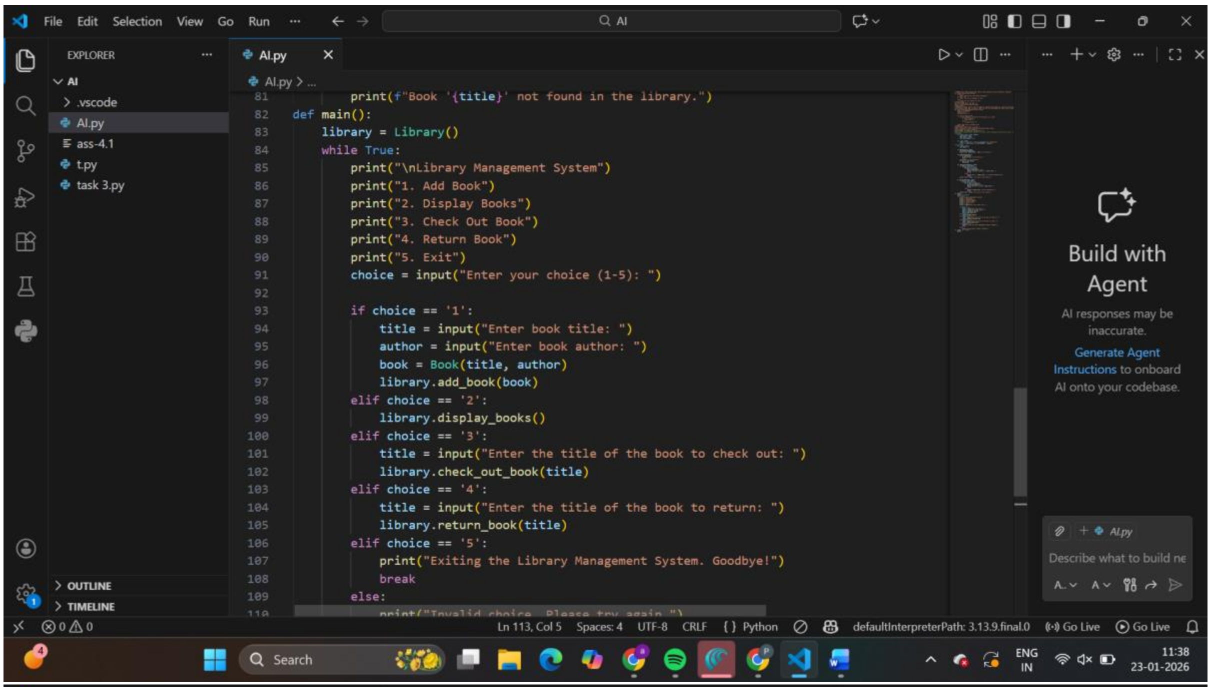
• Short reflection on AI-assisted coding experience.



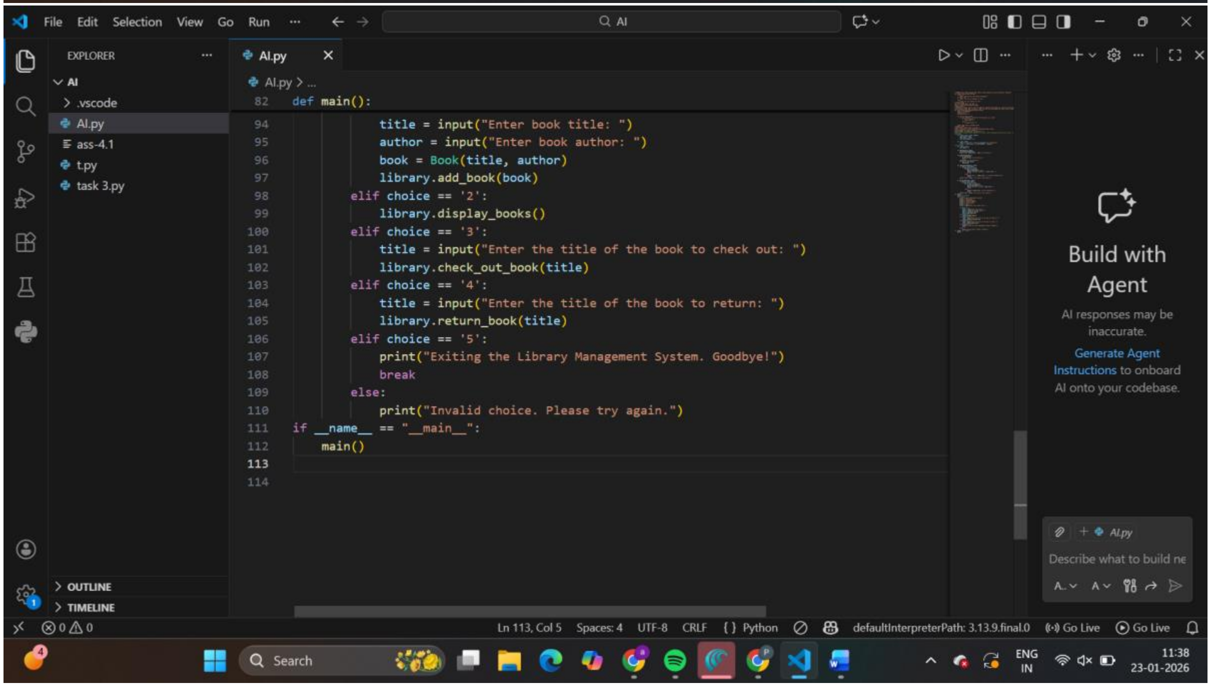
```python
##Generate a complete Python program for a simple Library Management System using classes,
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = True

    def __str__(self):
        status = "Available" if self.is_available else "Checked Out"
        return f"'{self.title}' by {self.author} - {status}"
class Library:
    def __init__(self):
        self.books = []

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

    def display_books(self):
        if not self.books:
            print("No books in the library.")
            return
        print("Books in the library:")
        for book in self.books:
            print(book)

    def check_out_book(self, title):
        for book in self.books:
            if book.title == title:
```



```python
class Library:
    def check_out_book(self, title):
                if book.is_available:
                    book.is_available = False
                    print(f"You have checked out '{book.title}'.")
                    return
                else:
                    print(f"Sorry, '{book.title}' is currently checked out.")
                    return
        print(f"Book '{title}' not found in the library.")

    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                if not book.is_available:
                    book.is_available = True
                    print(f"You have returned '{book.title}'.")
                    return
                else:
                    print(f"'{book.title}' was not checked out.")
                    return
        print(f"Book '{title}' not found in the library.")
def main():
    library = Library()
    while True:
        print("\nLibrary Management System")
        print("1. Add Book")
        print("2. Display Books")
        print("3. Check Out Book")
```

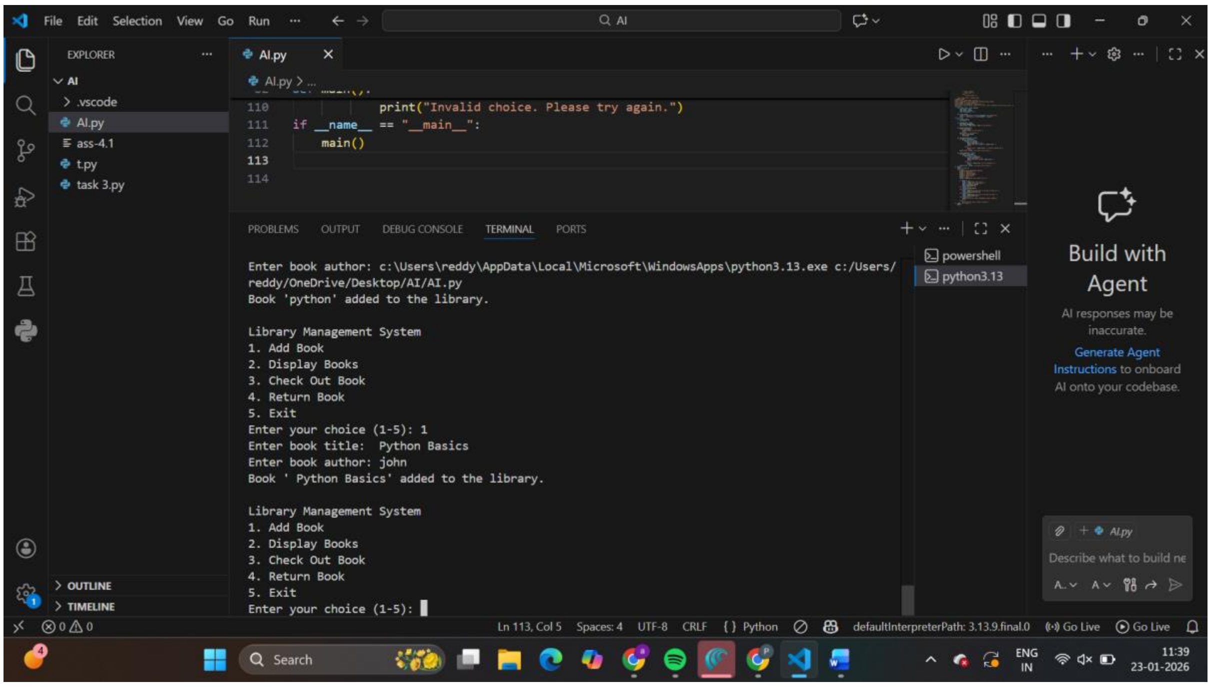

File Edit Selection View Go Run
EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 113, Col 5 Spaces: 4 UTF-8 CRLF {} Python defaultInterpreterPath: 3.13.9.final.0 Go Live

```python
        print(f"Book '{title}' not found in the library.")
def main():
    library = Library()
    while True:
        print("\nLibrary Management System")
        print("1. Add Book")
        print("2. Display Books")
        print("3. Check Out Book")
        print("4. Return Book")
        print("5. Exit")
        choice = input("Enter your choice (1-5): ")

        if choice == '1':
            title = input("Enter book title: ")
            author = input("Enter book author: ")
            book = Book(title, author)
            library.add_book(book)
        elif choice == '2':
            library.display_books()
        elif choice == '3':
            title = input("Enter the title of the book to check out: ")
            library.check_out_book(title)
        elif choice == '4':
            title = input("Enter the title of the book to return: ")
            library.return_book(title)
        elif choice == '5':
            print("Exiting the Library Management System. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
```


File Edit Selection View Go Run
EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 113, Col 5 Spaces: 4 UTF-8 CRLF {} Python defaultInterpreterPath: 3.13.9.final.0 Go Live

```python
def main():
            title = input("Enter book title: ")
            author = input("Enter book author: ")
            book = Book(title, author)
            library.add_book(book)
        elif choice == '2':
            library.display_books()
        elif choice == '3':
            title = input("Enter the title of the book to check out: ")
            library.check_out_book(title)
        elif choice == '4':
            title = input("Enter the title of the book to return: ")
            library.return_book(title)
        elif choice == '5':
            print("Exiting the Library Management System. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
if __name__ == "__main__":
    main()
```

Output:



```
        print("Invalid choice. Please try again.")
if __name__ == "__main__":
    main()
```

```
Enter book author: c:\Users\reddy\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/Users/
reddy/OneDrive/Desktop/AI/AI.py
Book 'python' added to the library.

Library Management System
1. Add Book
2. Display Books
3. Check Out Book
4. Return Book
5. Exit
Enter your choice (1-5): 1
Enter book title:  Python Basics
Enter book author: john
Book ' Python Basics' added to the library.

Library Management System
1. Add Book
2. Display Books
3. Check Out Book
4. Return Book
5. Exit
Enter your choice (1-5): 
```

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

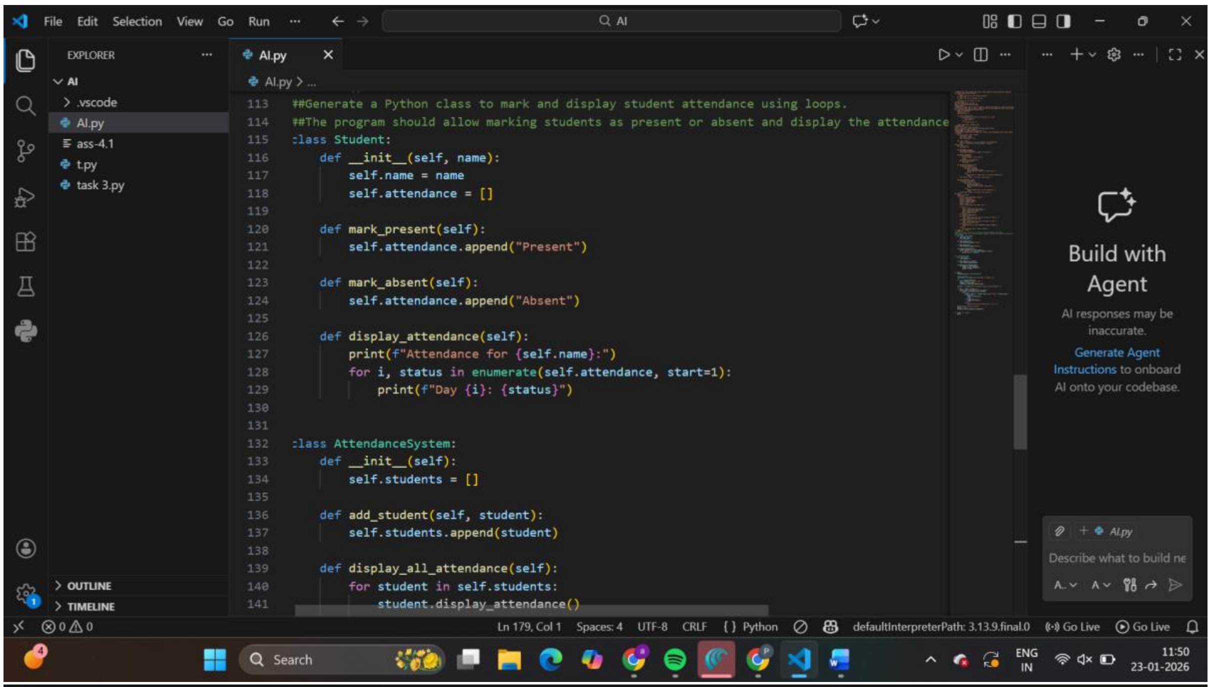
Task: Use an AI tool to generate an attendance management class.

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

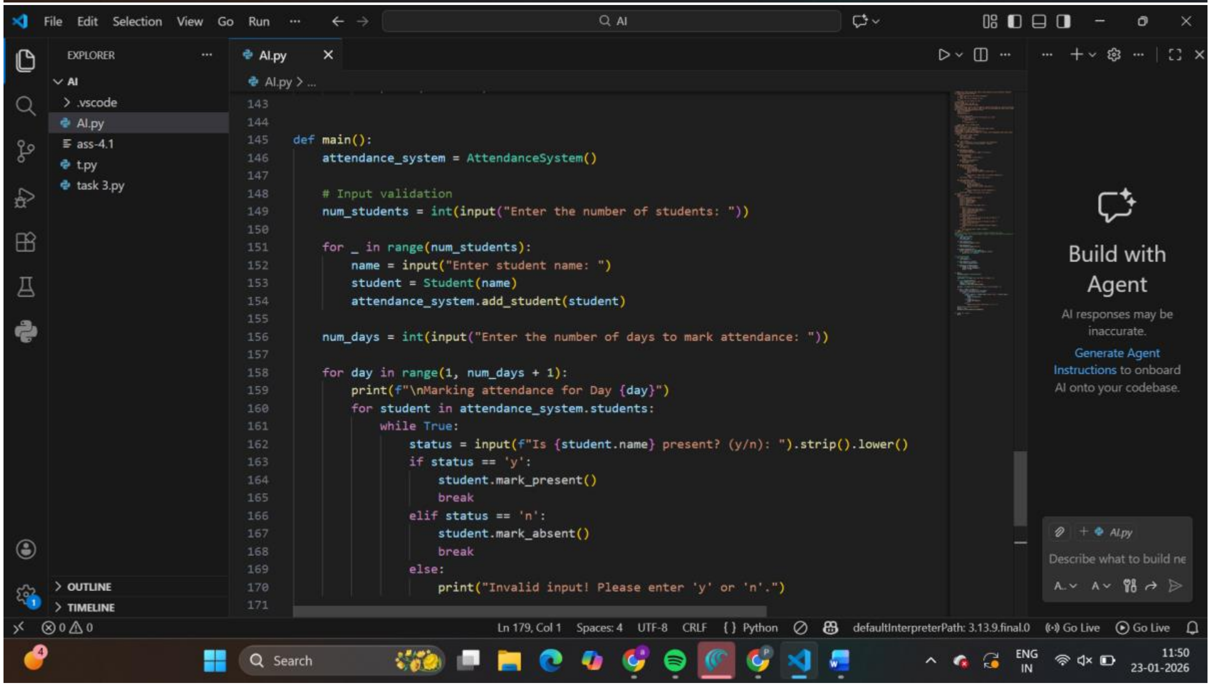
Expected Output:

• AI-generated attendance logic.

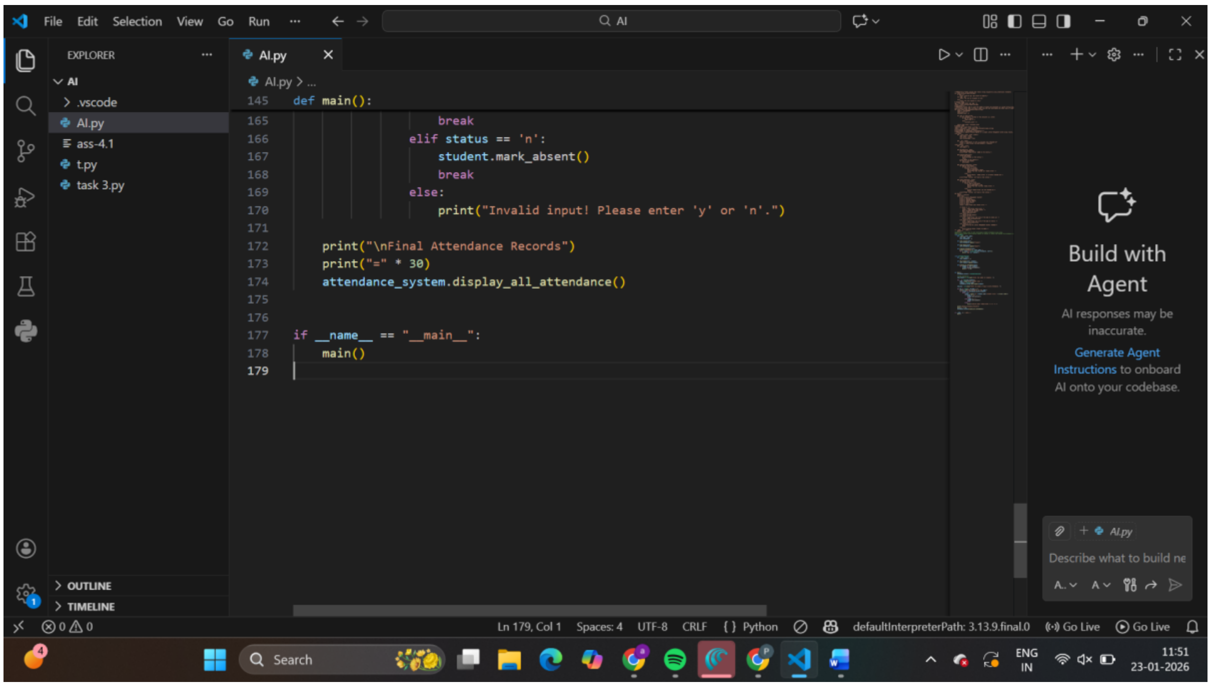
• Correct display of attendance.



```python
##Generate a Python class to mark and display student attendance using loops.
##The program should allow marking students as present or absent and display the attendance
:lass Student:
    def __init__(self, name):
        self.name = name
        self.attendance = []

    def mark_present(self):
        self.attendance.append("Present")

    def mark_absent(self):
        self.attendance.append("Absent")

    def display_attendance(self):
        print(f"Attendance for {self.name}:")
        for i, status in enumerate(self.attendance, start=1):
            print(f"Day {i}: {status}")


:lass AttendanceSystem:
    def __init__(self):
        self.students = []

    def add_student(self, student):
        self.students.append(student)

    def display_all_attendance(self):
        for student in self.students:
            student.display_attendance()
```

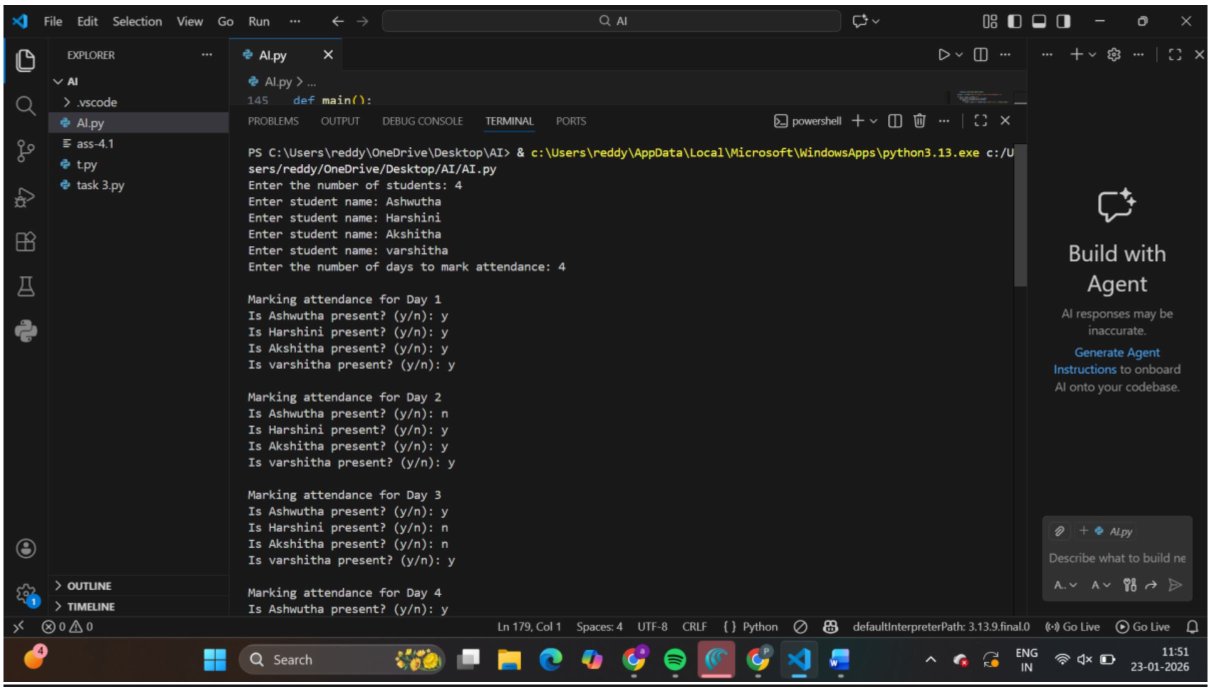


```python


def main():
    attendance_system = AttendanceSystem()

    # Input validation
    num_students = int(input("Enter the number of students: "))

    for _ in range(num_students):
        name = input("Enter student name: ")
        student = Student(name)
        attendance_system.add_student(student)

    num_days = int(input("Enter the number of days to mark attendance: "))

    for day in range(1, num_days + 1):
        print(f"\nMarking attendance for Day {day}")
        for student in attendance_system.students:
            while True:
                status = input(f"Is {student.name} present? (y/n): ").strip().lower()
                if status == 'y':
                    student.mark_present()
                    break
                elif status == 'n':
                    student.mark_absent()
                    break
                else:
                    print("Invalid input! Please enter 'y' or 'n'.")

```


EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 179, Col 1
Spaces: 4
UTF-8
CRLF
Python
defaultInterpreterPath: 3.13.9.final.0
Go Live
Search
ENG IN
11:51
23-01-2026

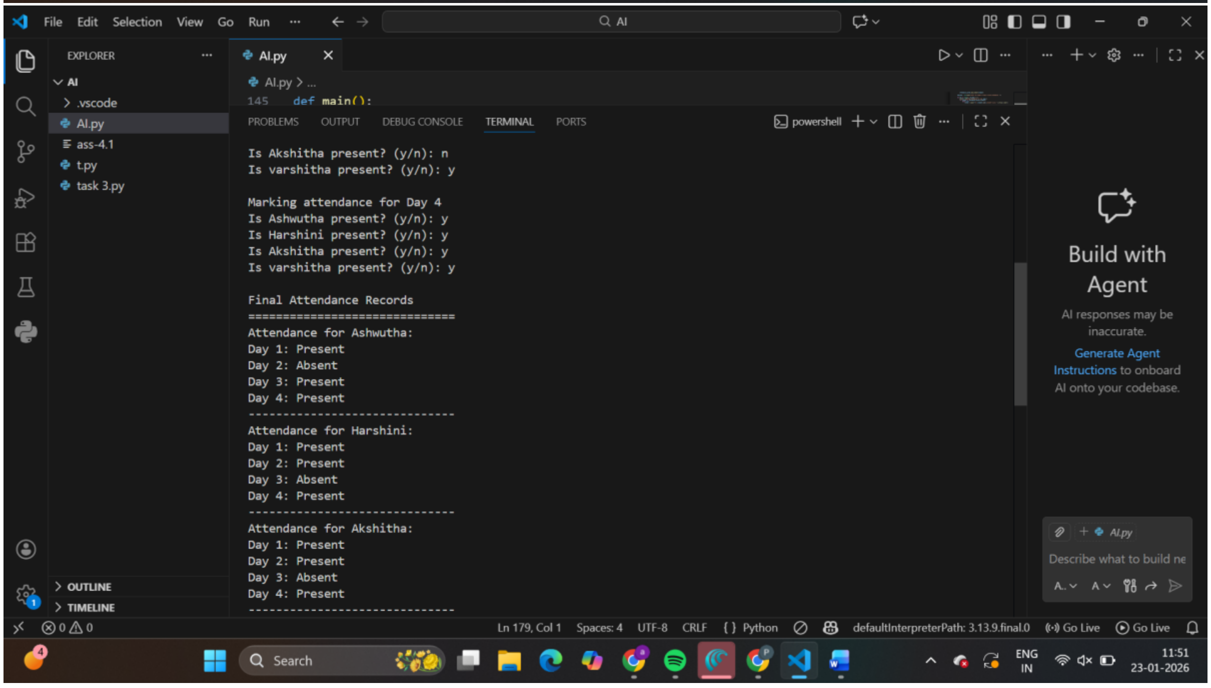
```python
def main():
                    break
                elif status == 'n':
                    student.mark_absent()
                    break
                else:
                    print("Invalid input! Please enter 'y' or 'n'.")

    print("\nFinal Attendance Records")
    print("=" * 30)
    attendance_system.display_all_attendance()


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

Output:

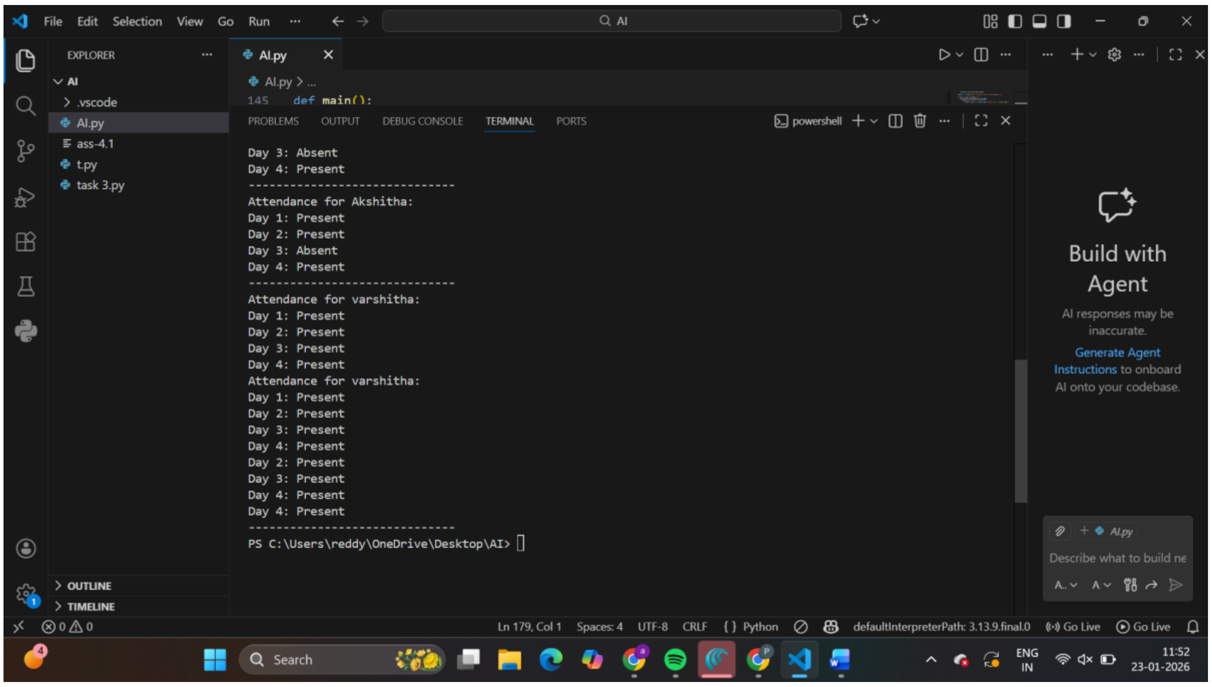

AI.py
EXPLORER
.vscode
ass-4.1
t.py
task 3.py
TERMINAL
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

```
def main():
```

```
PS C:\Users\reddy\OneDrive\Desktop\AI> & c:\Users\reddy\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/U
sers/reddy/OneDrive/Desktop/AI/AI.py
Enter the number of students: 4
Enter student name: Ashwutha
Enter student name: Harshini
Enter student name: Akshitha
Enter student name: varshitha
Enter the number of days to mark attendance: 4

Marking attendance for Day 1
Is Ashwutha present? (y/n): y
Is Harshini present? (y/n): y
Is Akshitha present? (y/n): y
Is varshitha present? (y/n): y

Marking attendance for Day 2
Is Ashwutha present? (y/n): n
Is Harshini present? (y/n): y
Is Akshitha present? (y/n): y
Is varshitha present? (y/n): y

Marking attendance for Day 3
Is Ashwutha present? (y/n): y
Is Harshini present? (y/n): n
Is Akshitha present? (y/n): n
Is varshitha present? (y/n): y

Marking attendance for Day 4
Is Ashwutha present? (y/n): y
```



```
Is Akshitha present? (y/n): n
Is varshitha present? (y/n): y

Marking attendance for Day 4
Is Ashwutha present? (y/n): y
Is Harshini present? (y/n): y
Is Akshitha present? (y/n): y
Is varshitha present? (y/n): y

Final Attendance Records
==============================
Attendance for Ashwutha:
Day 1: Present
Day 2: Absent
Day 3: Present
Day 4: Present
------------------------------
Attendance for Harshini:
Day 1: Present
Day 2: Present
Day 3: Absent
Day 4: Present
------------------------------
Attendance for Akshitha:
Day 1: Present
Day 2: Present
Day 3: Absent
Day 4: Present
------------------------------
```



```
Day 3: Absent
Day 4: Present
------------------------------
Attendance for Akshitha:
Day 1: Present
Day 2: Present
Day 3: Absent
Day 4: Present
------------------------------
Attendance for varshitha:
Day 1: Present
Day 2: Present
Day 3: Present
Day 4: Present
Attendance for varshitha:
Day 1: Present
Day 2: Present
Day 3: Present
Day 4: Present
Day 2: Present
Day 3: Present
Day 4: Present
Day 4: Present
------------------------------
PS C:\Users\reddy\OneDrive\Desktop\AI>
```

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

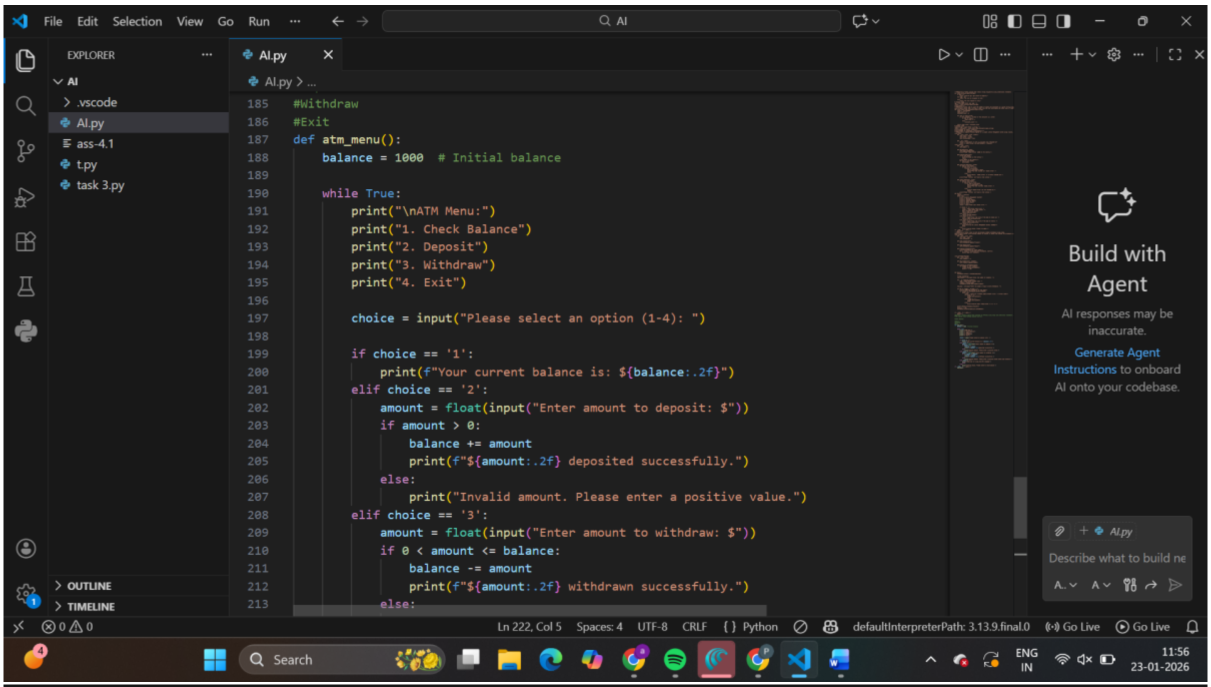
to simulate an ATM menu.”

Expected Output:

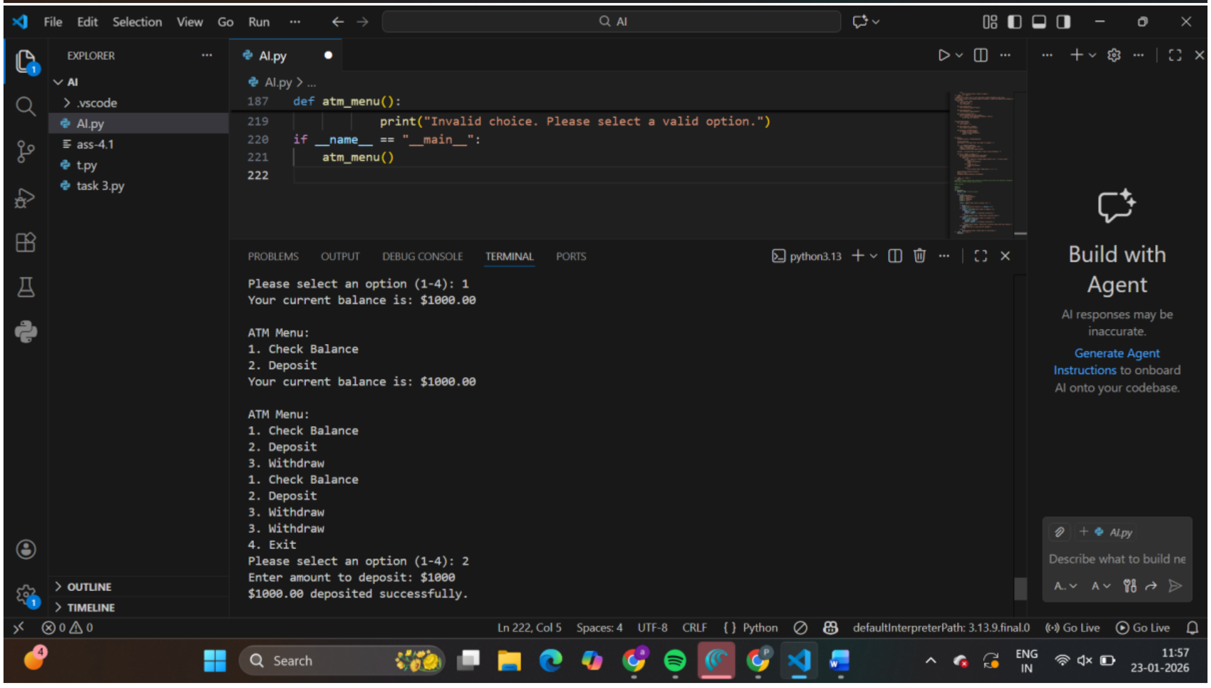
• AI-generated menu logic.

• Correct option handling.

• Output verification.

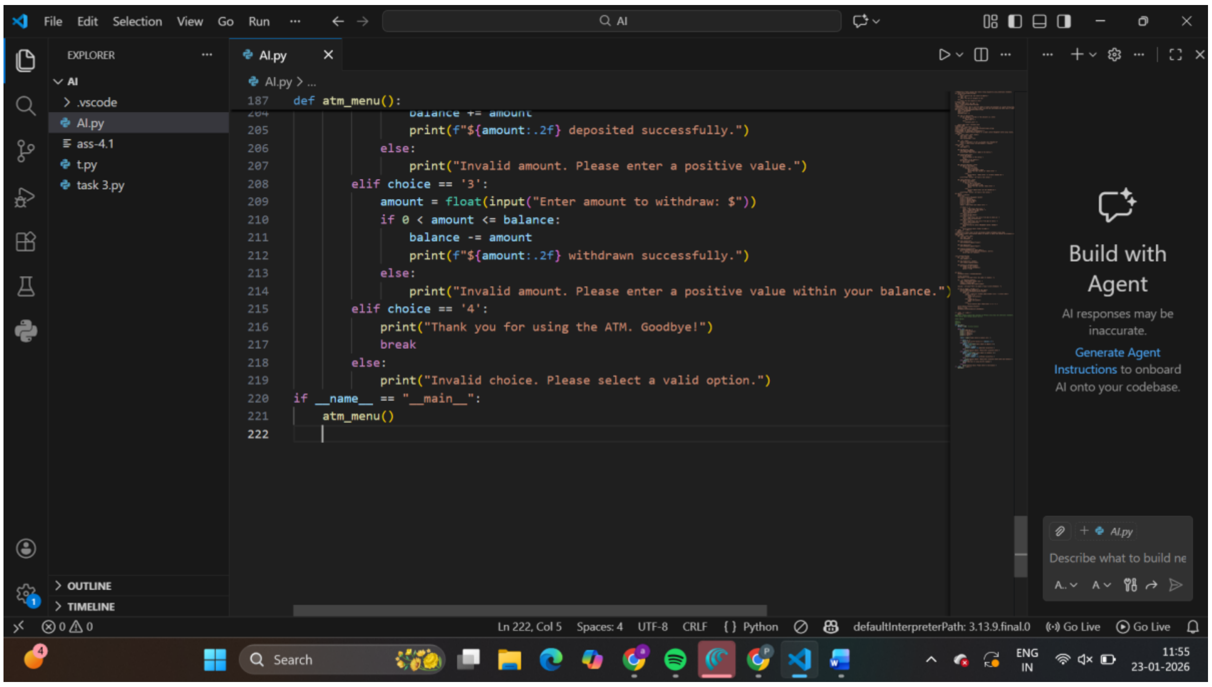

File Edit Selection View Go Run
EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 222, Col 5 Spaces: 4 UTF-8 CRLF { } Python defaultInterpreterPath: 3.13.9.final.0 Go Live
11:56
23-01-2026

```python
#Withdraw
#Exit
def atm_menu():
    balance = 1000  # Initial balance

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

        choice = input("Please select an option (1-4): ")

        if choice == '1':
            print(f"Your current balance is: ${balance:.2f}")
        elif choice == '2':
            amount = float(input("Enter amount to deposit: $"))
            if amount > 0:
                balance += amount
                print(f"${amount:.2f} deposited successfully.")
            else:
                print("Invalid amount. Please enter a positive value.")
        elif choice == '3':
            amount = float(input("Enter amount to withdraw: $"))
            if 0 < amount <= balance:
                balance -= amount
                print(f"${amount:.2f} withdrawn successfully.")
            else:
```


File Edit Selection View Go Run
EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python3.13
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 222, Col 5 Spaces: 4 UTF-8 CRLF { } Python defaultInterpreterPath: 3.13.9.final.0 Go Live
11:57
23-01-2026

```python
def atm_menu():
                print("Invalid choice. Please select a valid option.")
if __name__ == "__main__":
    atm_menu()
```

```
Please select an option (1-4): 1
Your current balance is: $1000.00

ATM Menu:
1. Check Balance
2. Deposit
Your current balance is: $1000.00

ATM Menu:
1. Check Balance
2. Deposit
3. Withdraw
1. Check Balance
2. Deposit
3. Withdraw
3. Withdraw
4. Exit
Please select an option (1-4): 2
Enter amount to deposit: $1000
$1000.00 deposited successfully.
```


File Edit Selection View Go Run ···
AI
EXPLORER
AI
.vscode
AI.py
ass-4.1
t.py
task 3.py
OUTLINE
TIMELINE
AI.py
AI.py > ...
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build ne
Ln 222, Col 5 Spaces: 4 UTF-8 CRLF { } Python defaultInterpreterPath: 3.13.9.final.0 Go Live Go Live
Search
ENG IN
11:55
23-01-2026

```python
def atm_menu():
                balance += amount
                print(f"${amount:.2f} deposited successfully.")
            else:
                print("Invalid amount. Please enter a positive value.")
        elif choice == '3':
            amount = float(input("Enter amount to withdraw: $"))
            if 0 < amount <= balance:
                balance -= amount
                print(f"${amount:.2f} withdrawn successfully.")
            else:
                print("Invalid amount. Please enter a positive value within your balance.")
        elif choice == '4':
            print("Thank you for using the ATM. Goodbye!")
            break
        else:
            print("Invalid choice. Please select a valid option.")
if __name__ == "__main__":
    atm_menu()
    
```