# AI ASSISTED CODING

# LAB 6.5

**2303A51733**

**T.PRANATHI**

**BATCH-11**

## 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.

**PROMPT:**

#Give me a python code to check whether the person is eligible to vote or not based on their citizenship

**CODE:**

```python
#Giive me a python code to check whether the person is eligible to vote or not based on their
def is_eligible_to_vote(age, citizenship):
    if age >= 18 and citizenship.lower() == "yes":
        return True
    else:
        return False
# Example usage
age = int(input("Enter your age: "))
citizenship = input("Are you a citizen? (yes/no): ")
if is_eligible_to_vote(age, citizenship):
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")

```

**OUTPUT:**

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop/AIAC/lab5.py
Enter your age: 18
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop/AIAC/lab5.py
Enter your age: 17
Are you a citizen? (yes/no): no
You are not eligible to vote.
PS C:\Users\thota\OneDrive\Desktop\AIAC>

powershell
Python
powershell
Python
Python
Python
Python
Python
Python
Python
Python

Ln 7, Col 16  Spaces: 4  UTF-
```

# **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.

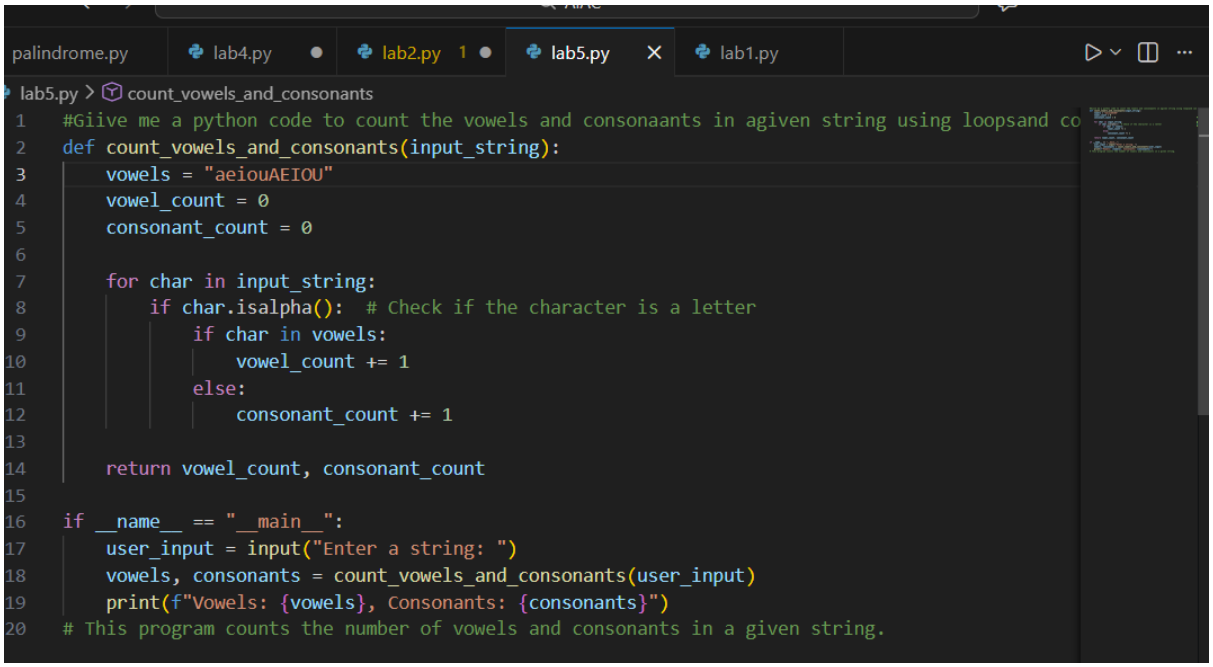
• Correct counts.

• Output verification.

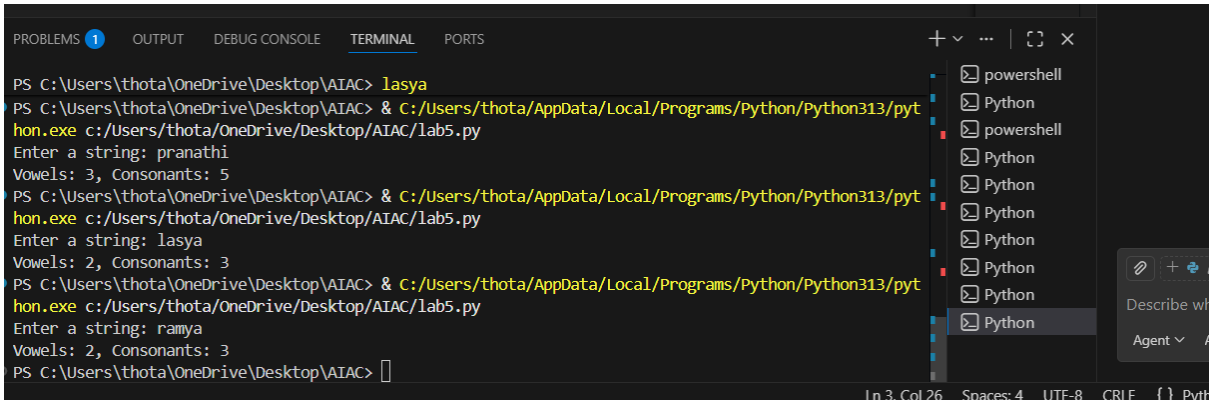
**PROMPT:**

#Give me a python code to count the vowels and consonaants in agiven string using loopsand conditional statements with using inputb by user

**CODE:**




**OUTPUT:**




# **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.

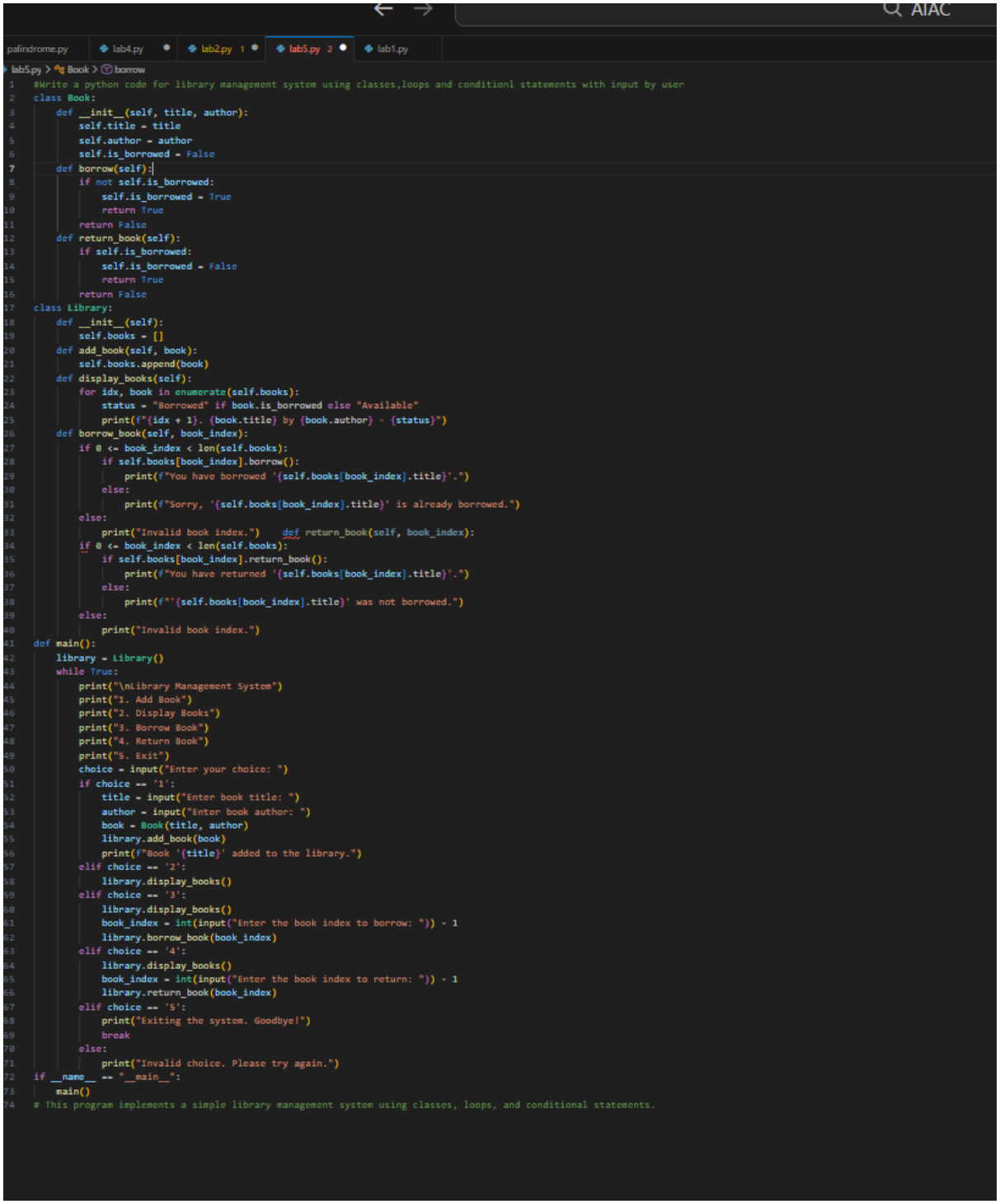
• Review of AI suggestions quality.

• Short reflection on AI-assisted coding experience.

**PROMPT:**

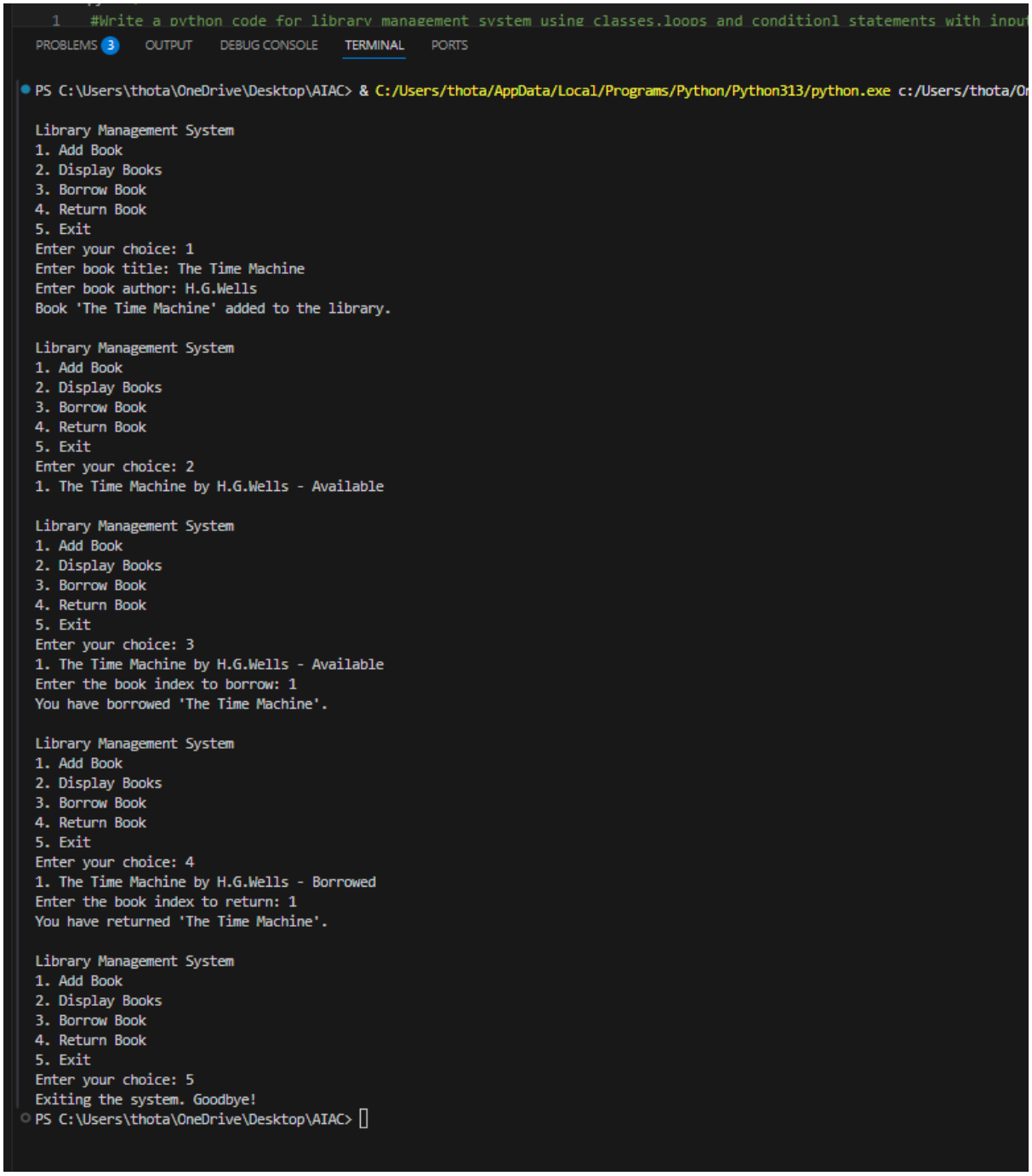
**#**Write a python code for library management system using classes,loops and conditionl statements with input by user

**CODE:**



```python
#Write a python code for library management system using classes,loops and conditionl statements with input by user
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_borrowed = False
    def borrow(self):
        if not self.is_borrowed:
            self.is_borrowed = True
            return True
        return False
    def return_book(self):
        if self.is_borrowed:
            self.is_borrowed = False
            return True
        return False
class Library:
    def __init__(self):
        self.books = []
    def add_book(self, book):
        self.books.append(book)
    def display_books(self):
        for idx, book in enumerate(self.books):
            status = "Borrowed" if book.is_borrowed else "Available"
            print(f"{idx + 1}. {book.title} by {book.author} - {status}")
    def borrow_book(self, book_index):
        if 0 <= book_index < len(self.books):
            if self.books[book_index].borrow():
                print(f"You have borrowed '{self.books[book_index].title}'.")
            else:
                print(f"Sorry, '{self.books[book_index].title}' is already borrowed.")
        else:
            print("Invalid book index.")    def return_book(self, book_index):
        if 0 <= book_index < len(self.books):
            if self.books[book_index].return_book():
                print(f"You have returned '{self.books[book_index].title}'.")
            else:
                print(f"'{self.books[book_index].title}' was not borrowed.")
        else:
            print("Invalid book index.")
def main():
    library = Library()
    while True:
        print("\nLibrary Management System")
        print("1. Add Book")
        print("2. Display Books")
        print("3. Borrow Book")
        print("4. Return Book")
        print("5. Exit")
        choice = input("Enter your choice: ")
        if choice == '1':
            title = input("Enter book title: ")
            author = input("Enter book author: ")
            book = Book(title, author)
            library.add_book(book)
            print(f"Book '{title}' added to the library.")
        elif choice == '2':
            library.display_books()
        elif choice == '3':
            library.display_books()
            book_index = int(input("Enter the book index to borrow: ")) - 1
            library.borrow_book(book_index)
        elif choice == '4':
            library.display_books()
            book_index = int(input("Enter the book index to return: ")) - 1
            library.return_book(book_index)
        elif choice == '5':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
if __name__ == "__main__":
    main()
# This program implements a simple library management system using classes, loops, and conditional statements.
```

**OUTPUT:**




**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.

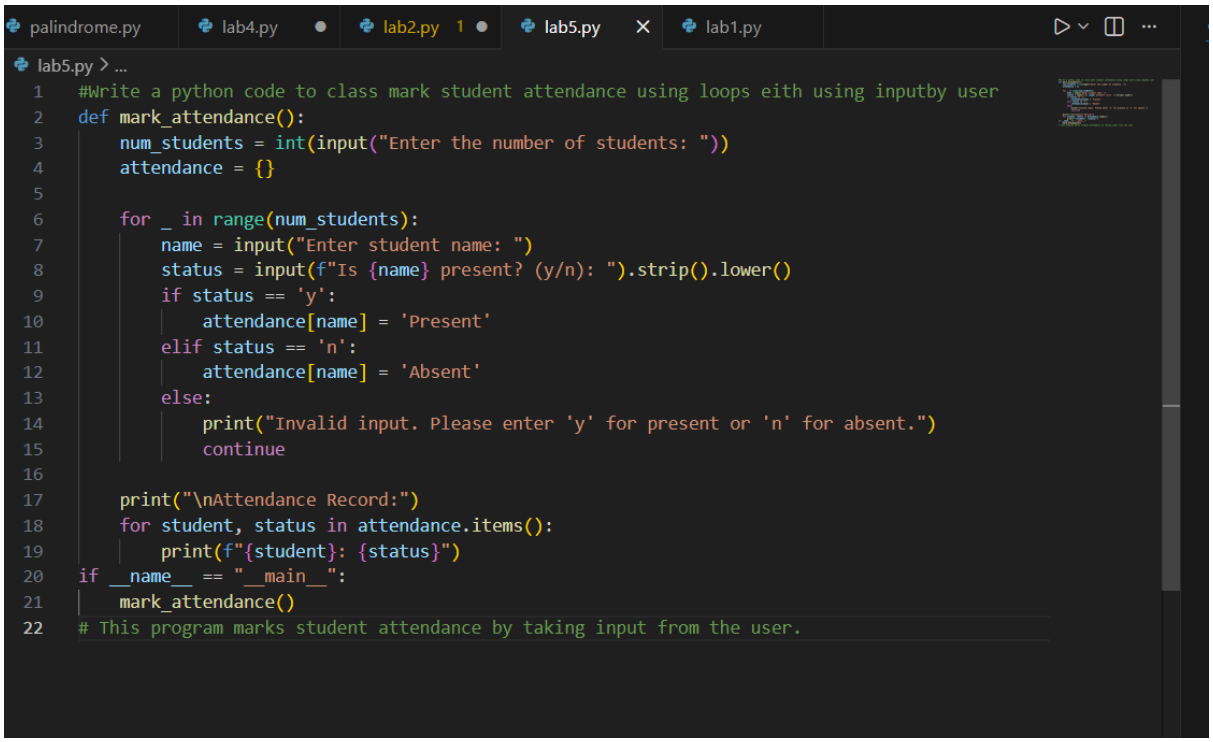
• Correct display of attendance.

• Test cases.

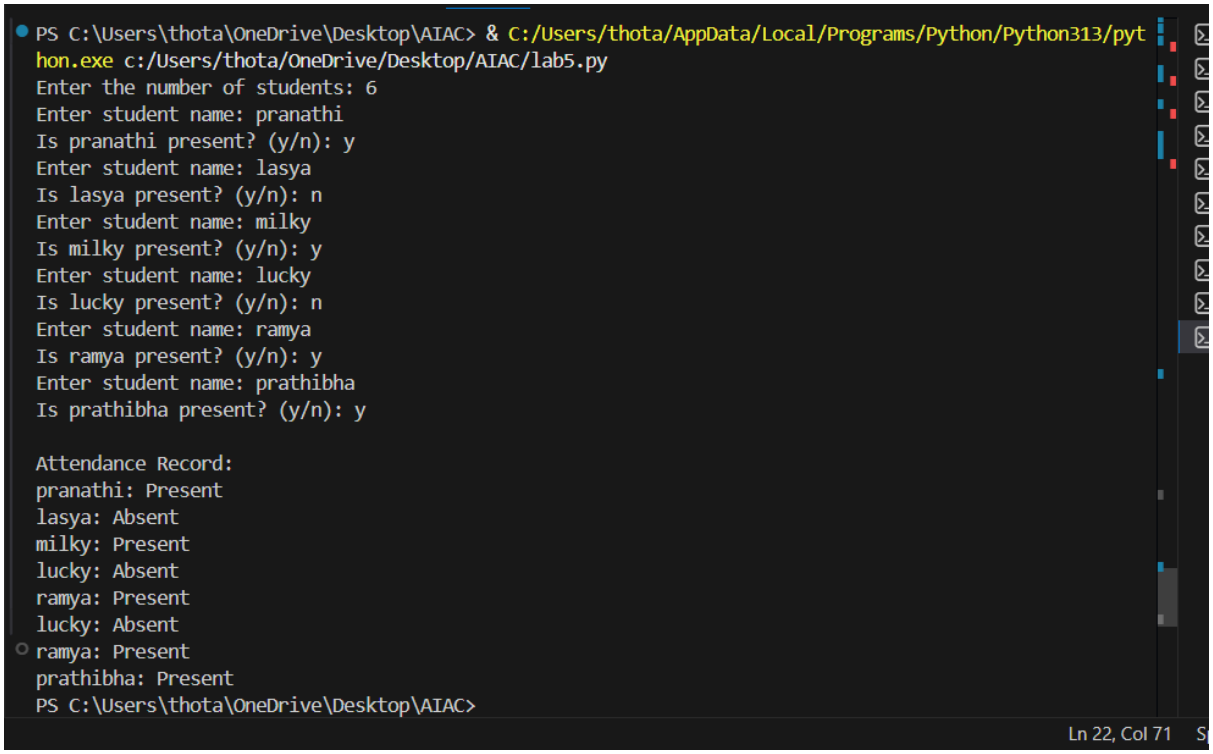
**PROMPT:**

#Write a python code to class mark student attendance using loops eith using inputby user

**CODE:**




**OUTPUT:**




# **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.”

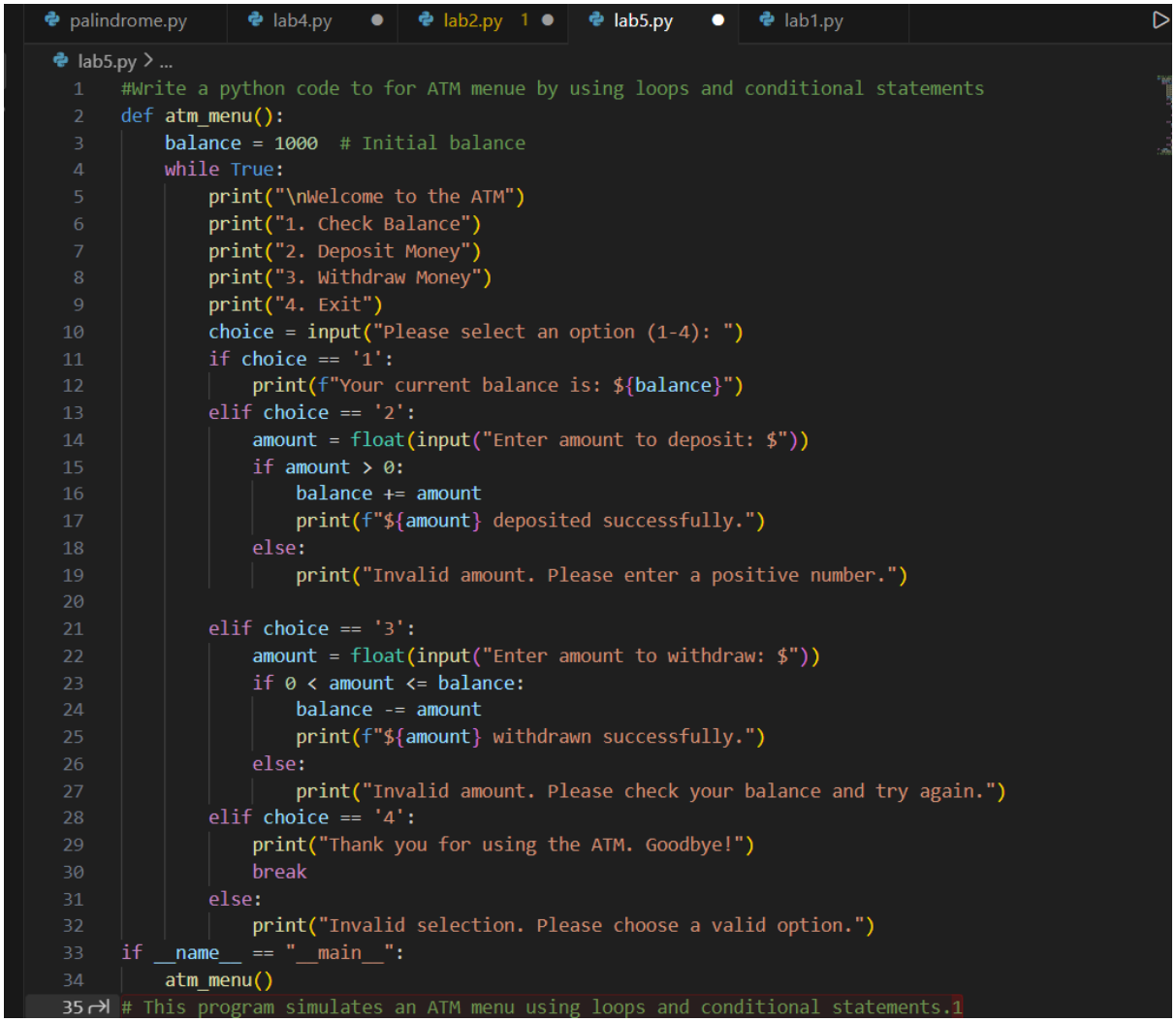
Expected Output:

- AI-generated menu logic.
- Correct option handling.
- Output verification.

**PROMPT:**

#Write a python code to for ATM menue by using loops and conditional statements

## CODE:



```python
#Write a python code to for ATM menue by using loops and conditional statements
def atm_menu():
    balance = 1000  # Initial balance
    while True:
        print("\nWelcome to the ATM")
        print("1. Check Balance")
        print("2. Deposit Money")
        print("3. Withdraw Money")
        print("4. Exit")
        choice = input("Please select an option (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.")
            else:
                print("Invalid amount. Please enter a positive number.")

        elif choice == '3':
            amount = float(input("Enter amount to withdraw: $"))
            if 0 < amount <= balance:
                balance -= amount
                print(f"${amount} withdrawn successfully.")
            else:
                print("Invalid amount. Please check your balance and try again.")
        elif choice == '4':
            print("Thank you for using the ATM. Goodbye!")
            break
        else:
            print("Invalid selection. Please choose a valid option.")
if __name__ == "__main__":
    atm_menu()
# This program simulates an ATM menu using loops and conditional statements.1
```

## OUTPUT:


PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\thota\OneDrive\Desktop\AIAC>
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 2
Enter amount to deposit: $1000
$1000.0 deposited successfully.
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $2000.0
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 3
Enter amount to withdraw: $1000
$1000.0 withdrawn successfully.
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000.0
Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 4
3. Withdraw Money
4. Exit
Please select an option (1-4): 4
4. Exit
Please select an option (1-4): 4
Thank you for using the ATM. Goodbye!
PS C:\Users\thota\OneDrive\Desktop\AIAC>