ASSIGNMENT-6.5

BATCH-29

ROLL-NO:2303A51600

TASK-1

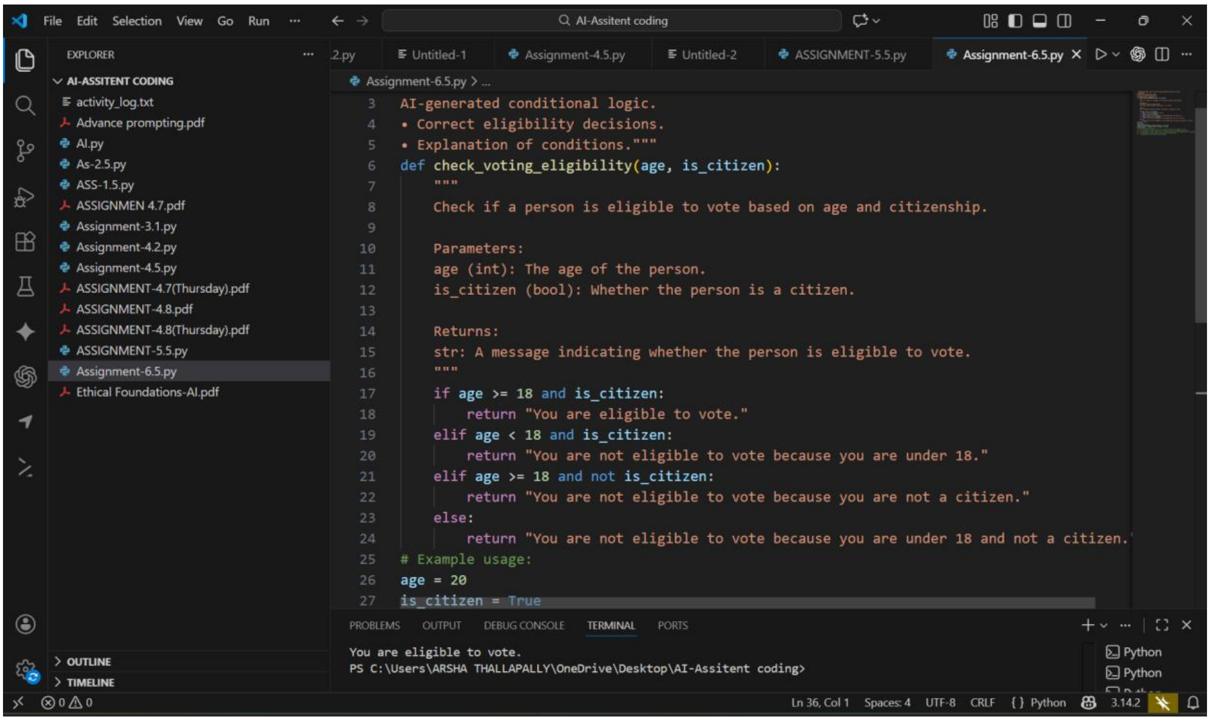
PROMPT:

Generate Python code to check voting eligibility based on age and citizenship.

AI-generated conditional logic.

- Correct eligibility decisions.
- Explanation of conditions.

CODE:



```python
AI-generated conditional logic.
• Correct eligibility decisions.
• Explanation of conditions."""
def check_voting_eligibility(age, is_citizen):
    """
    Check if a person is eligible to vote based on age and citizenship.

    Parameters:
    age (int): The age of the person.
    is_citizen (bool): Whether the person is a citizen.

    Returns:
    str: A message indicating whether the person is eligible to vote.
    """
    if age >= 18 and is_citizen:
        return "You are eligible to vote."
    elif age < 18 and is_citizen:
        return "You are not eligible to vote because you are under 18."
    elif age >= 18 and not is_citizen:
        return "You are not eligible to vote because you are not a citizen."
    else:
        return "You are not eligible to vote because you are under 18 and not a citizen."
# Example usage:
age = 20
is_citizen = True
```

```
You are eligible to vote.
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

OBSERVATIONS:

When the prompt given to explain about the condition – it specified

Logic was correctly given

Specifications given in order of the prompt

TASK-2

PROMPT:

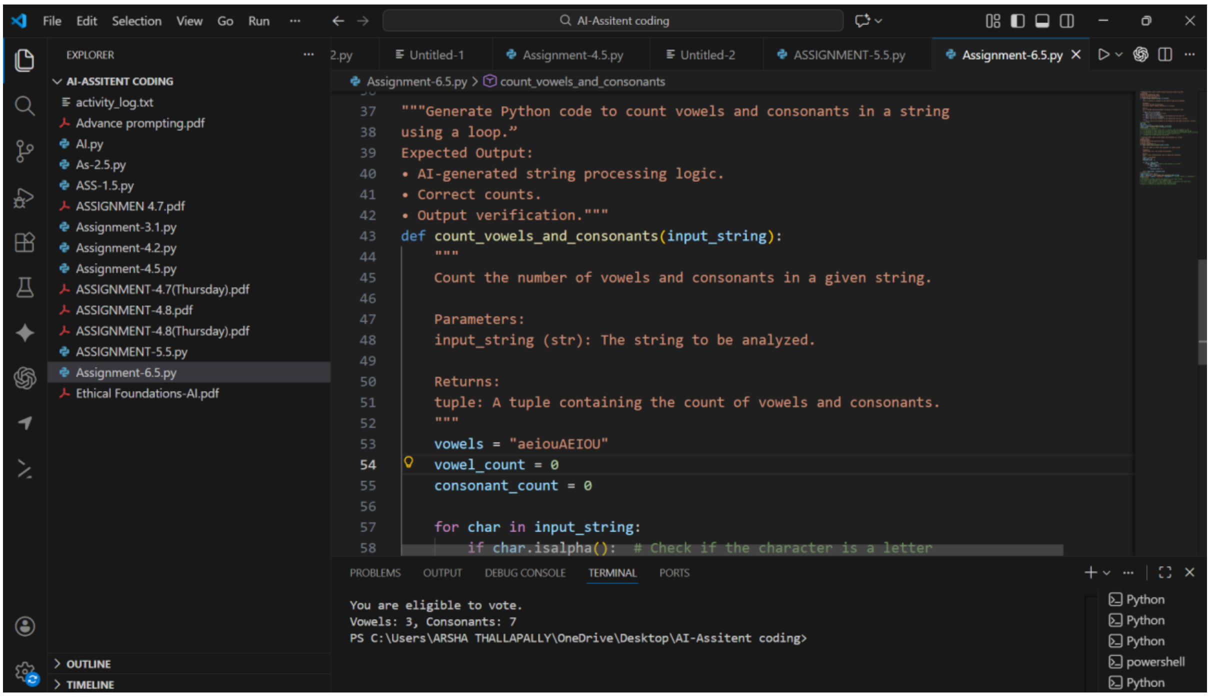
Generate Python code to count vowels and consonants in a string

using a loop.”

Expected Output:

- AI-generated string processing logic.
- Correct counts.
- Output verification

CODE:



```python
"""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."""
def count_vowels_and_consonants(input_string):
    """
    Count the number of vowels and consonants in a given string.

    Parameters:
    input_string (str): The string to be analyzed.

    Returns:
    tuple: A tuple containing the count of vowels and consonants.
    """
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
        if char.isalpha():  # Check if the character is a letter
```

```
You are eligible to vote.
Vowels: 3, Consonants: 7
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

OBSERVATIONS:

Checking output verification-correct

No explanation as not mentioned in the prompt

Few Test considered like checking alphabets

TASK-3

PROMPT:

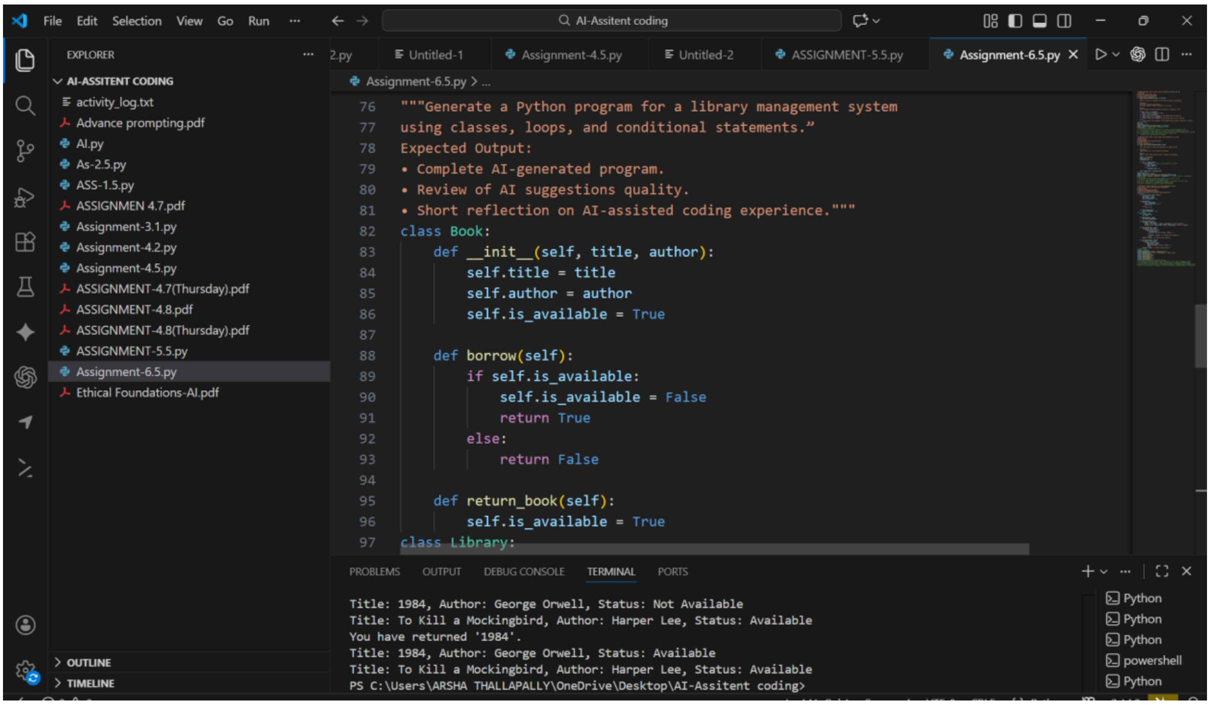
Generate a Python program for a library management system

using classes, loops, and conditional statements.”

Expected Output:

- Complete AI-generated program.

CODE:



```python
"""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."""
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = True

    def borrow(self):
        if self.is_available:
            self.is_available = False
            return True
        else:
            return False

    def return_book(self):
        self.is_available = True
class Library:
```

```
Title: 1984, Author: George Orwell, Status: Not Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
You have returned '1984'.
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

OBSERVATIONS:

Output given with respect to the promping stating loop

Class and conditions

Output is clear and neat

Correct and solves few test cases

TASK-4

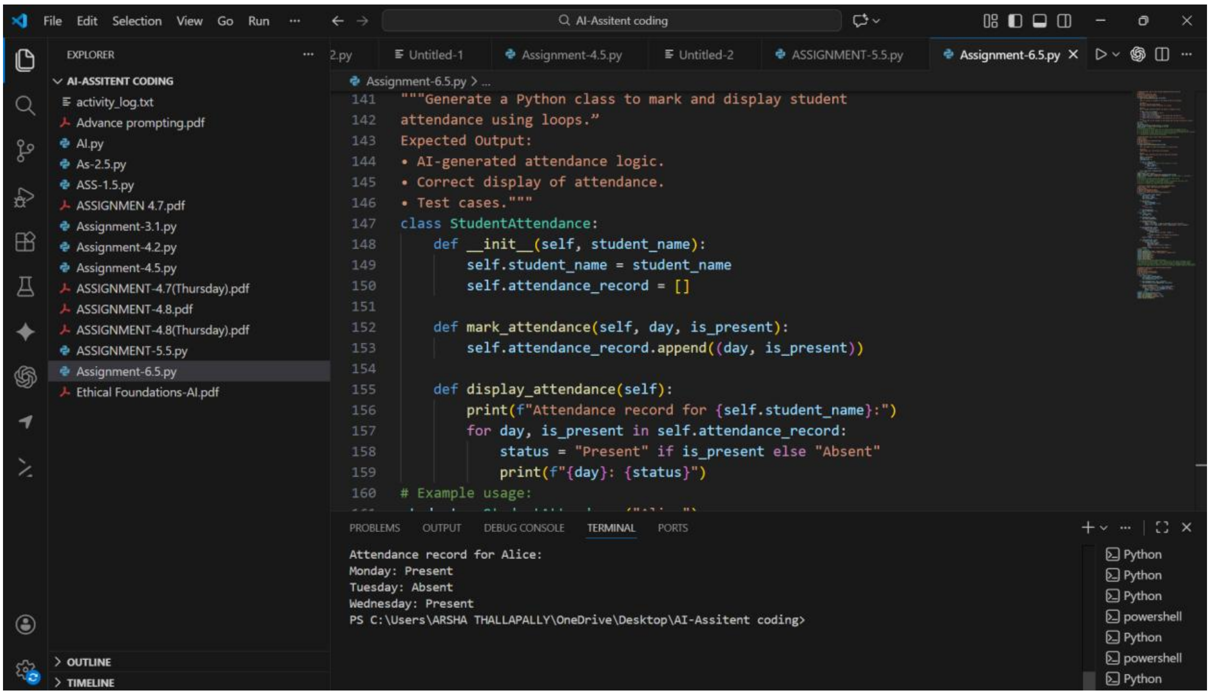
PROMPT:

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

• AI-generated attendance logic.

• Correct display of attendance.

CODE:



```python
"""Generate a Python class to mark and display student
attendance using loops.”
Expected Output:
• AI-generated attendance logic.
• Correct display of attendance.
• Test cases."""
class StudentAttendance:
    def __init__(self, student_name):
        self.student_name = student_name
        self.attendance_record = []

    def mark_attendance(self, day, is_present):
        self.attendance_record.append((day, is_present))

    def display_attendance(self):
        print(f"Attendance record for {self.student_name}:")
        for day, is_present in self.attendance_record:
            status = "Present" if is_present else "Absent"
            print(f"{day}: {status}")
# Example usage:
```

```
Attendance record for Alice:
Monday: Present
Tuesday: Absent
Wednesday: Present
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

OBSERVATIONS:

AI-generated attendance logic-correct

Correct display of attendance-done

Example usage also given

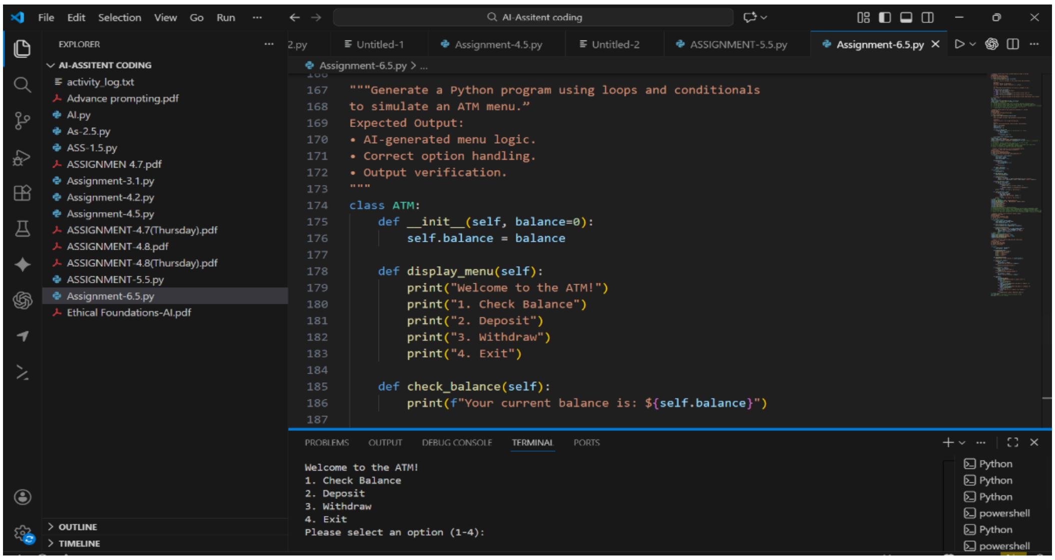
Many variables are considered to solve

Task-5:

PROMPT:

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

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



```python
"""Generate a Python program using loops and conditionals
to simulate an ATM menu.”
Expected Output:
• AI-generated menu logic.
• Correct option handling.
• Output verification.
"""
class ATM:
    def __init__(self, balance=0):
        self.balance = balance

    def display_menu(self):
        print("Welcome to the ATM!")
        print("1. Check Balance")
        print("2. Deposit")
        print("3. Withdraw")
        print("4. Exit")

    def check_balance(self):
        print(f"Your current balance is: ${self.balance}")

```

```
Welcome to the ATM!
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Please select an option (1-4):
```

CO

OBSERVATION:

Logic is optimized one

Option handling is on point

Output is correct

Goes well with the prompt

No violations of prompt seen