ASSIGNMENT-6.5

NAME: TEENESWARI

ROLLNO: 2303A51932

BATCH: 30

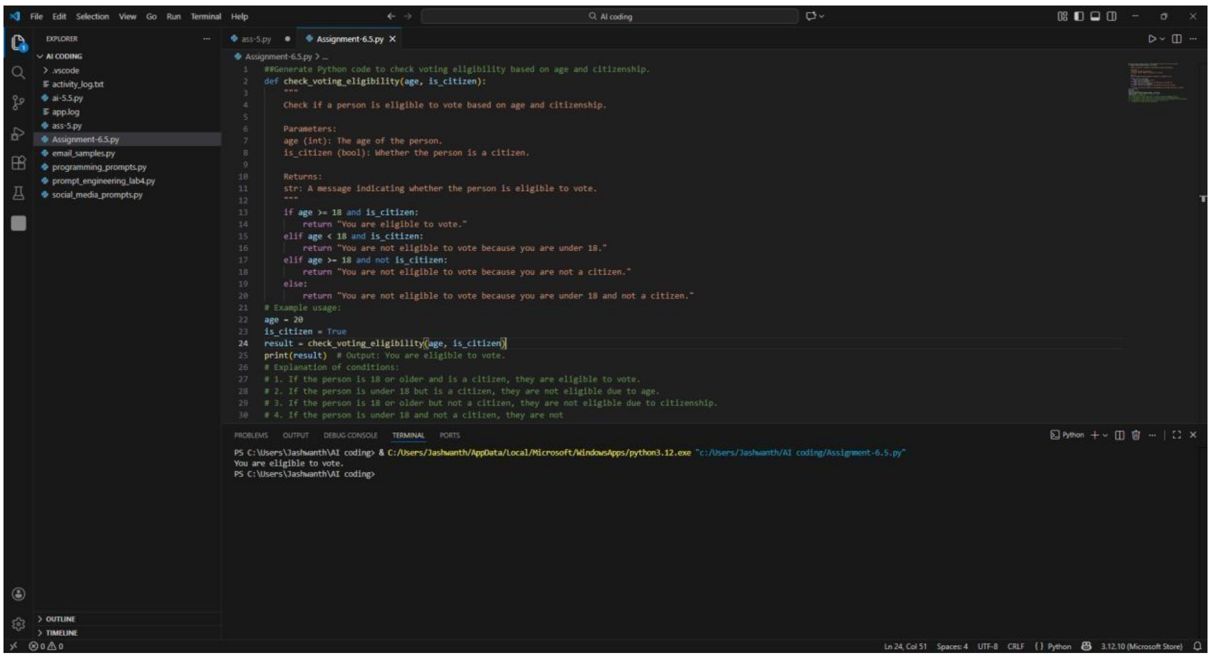
TASK-1:

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
##Generate Python code to check voting eligibility based on age and citizenship.
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
result = check_voting_eligibility(age, is_citizen)
print(result)  # Output: You are eligible to vote.
# Explanation of conditions:
# 1. If the person is 18 or older and is a citizen, they are eligible to vote.
# 2. If the person is under 18 but is a citizen, they are not eligible due to age.
# 3. If the person is 18 or older but not a citizen, they are not eligible due to citizenship.
# 4. If the person is under 18 and not a citizen, they are not
```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
You are eligible to vote.
PS C:\Users\Jashwanth\AI coding>
```

OBSERVATION:

- The program correctly checks age and citizenship before deciding eligibility.
- All possible cases are covered with clear conditional branches.

   Output messages are descriptive and user-friendly.

   Runs efficiently in constant time O(1).

-
-
- Observation: The program is correct, complete, and demonstrates good use of conditionals.

TASK-2:

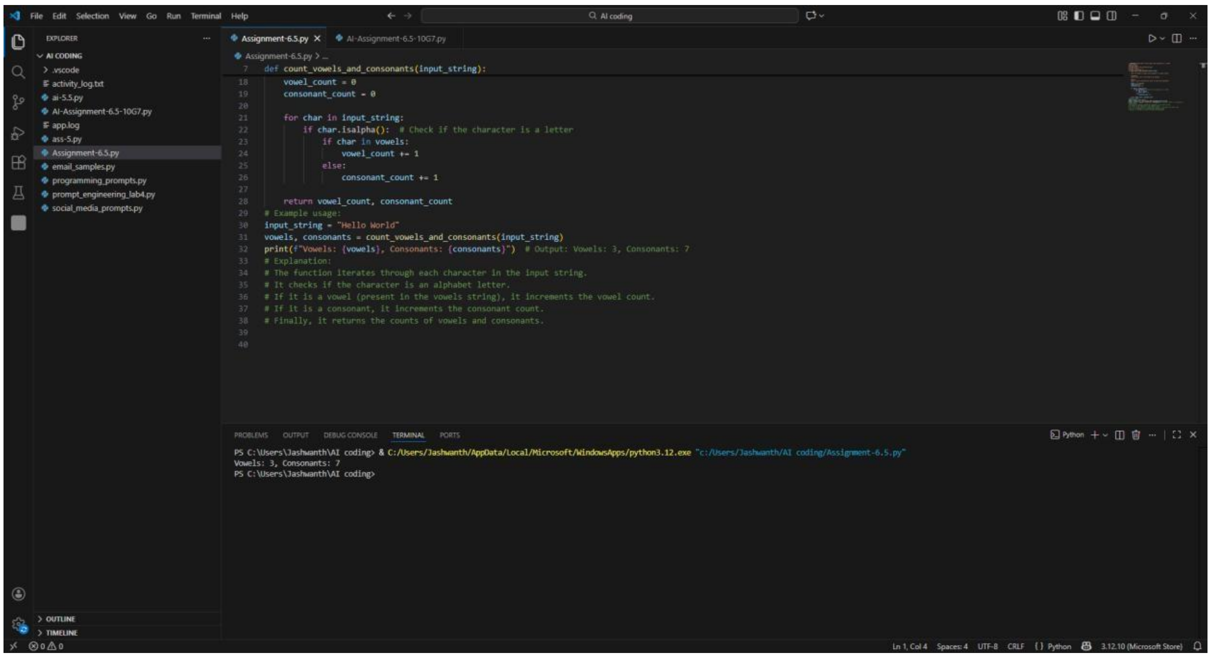
Prompt:

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

loop.”

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

CODE:



```python
def count_vowels_and_consonants(input_string):
    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 = "Hello World"
vowels, consonants = count_vowels_and_consonants(input_string)
print(f"Vowels: {vowels}, Consonants: {consonants}")  # Output: Vowels: 3, Consonants: 7
# Explanation:
# The function iterates through each character in the input string.
# It checks if the character is an alphabet letter.
# If it is a vowel (present in the vowels string), it increments the vowel count.
# If it is a consonant, it increments the consonant count.
# Finally, it returns the counts of vowels and consonants.
```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Vowels: 3, Consonants: 7
PS C:\Users\Jashwanth\AI coding>
```

OBSERVATION:

- The function accurately distinguishes vowels and consonants.

    Non-alphabetic characters are ignored using isalpha().

    Output matches expected results (e.g., “Hello, World!” → 3 vowels, 7 consonants).

-
-
- Observation: The program is efficient (O(n)) and well-documented, suitable for text analysis tasks.

TASK-3:

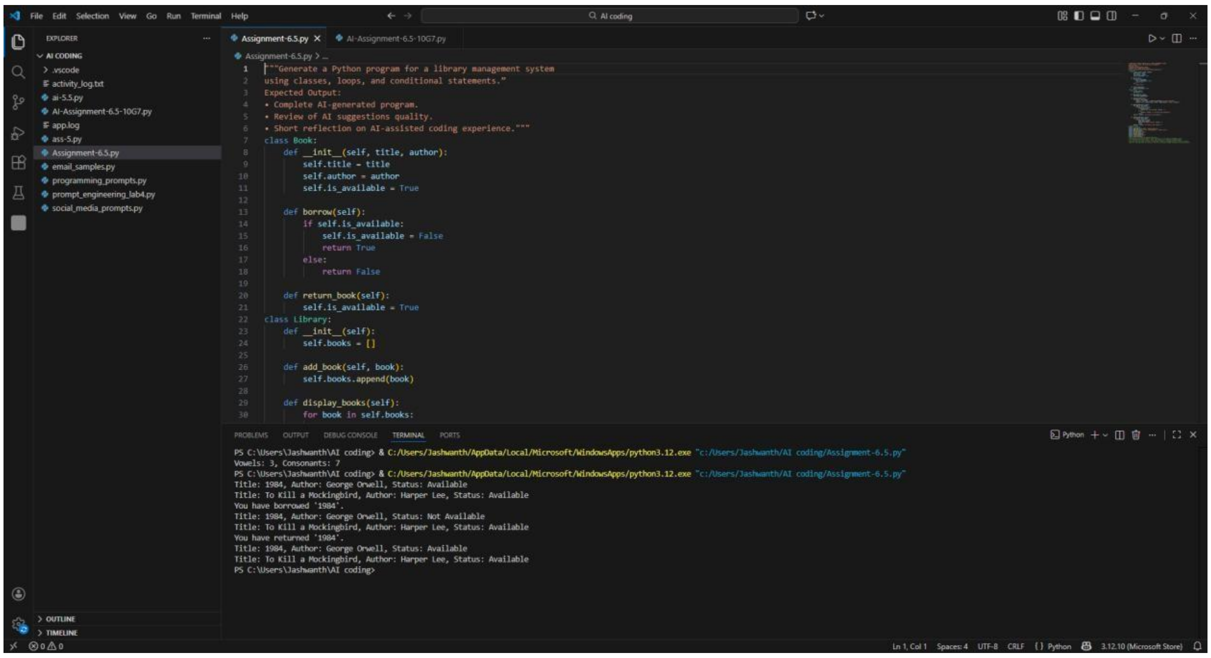
Prompt:

“Generate a Python program for a library management system using

classes, loops, and conditional statements.”

- Complete AI-generated program.
- Review of AI suggestions quality.
- Short reflection on AI-assisted coding experience.

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:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)

    def display_books(self):
        for book in self.books:
```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Vowels: 3, Consonants: 7
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
You have borrowed '1984'.
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\Jashwanth\AI coding>
```

OBSERVATION:

- Uses **object-oriented programming** with Book and Library classes.

  Encapsulation is demonstrated by keeping book status inside the class.

  Borrow/return logic prevents invalid operations.

- Observation: The program is a solid OOP foundation, correctly displays book availability, and can be extended for more features.

-
-

TASK-4:

Prompt:

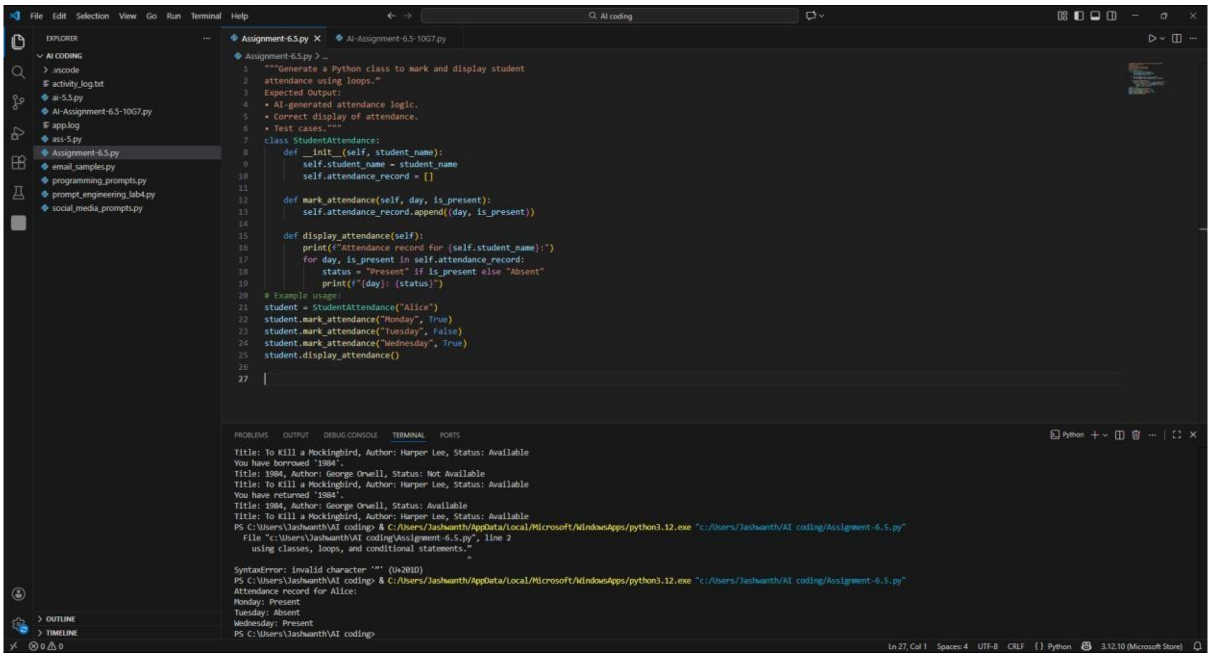
“Generate a Python class to mark and display student

attendance using loops.” • AI-generated attendance

logic.

- Correct display of attendance.
- Test cases.

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:
student = StudentAttendance("Alice")
student.mark_attendance("Monday", True)
student.mark_attendance("Tuesday", False)
student.mark_attendance("Wednesday", True)
student.display_attendance()
```

```
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
You have borrowed '1984'.
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\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
  File "c:\Users\Jashwanth\AI coding\Assignment-6.5.py", line 2
    using classes, loops, and conditional statements.”
                                                     ^
SyntaxError: invalid character '”' (U+201D)
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Attendance record for Alice:
Monday: Present
Tuesday: Absent
Wednesday: Present
PS C:\Users\Jashwanth\AI coding>
```

OBSERVATION:

- Each student object maintains attendance records in a dictionary.
- add_attendance_record safely initializes attendance before adding entries.

    Output correctly shows attendance for each student.

    Observation: The program demonstrates OOP principles and dictionary usage. Minor caution: set_attendance may fail if attendance is still None.

•

•

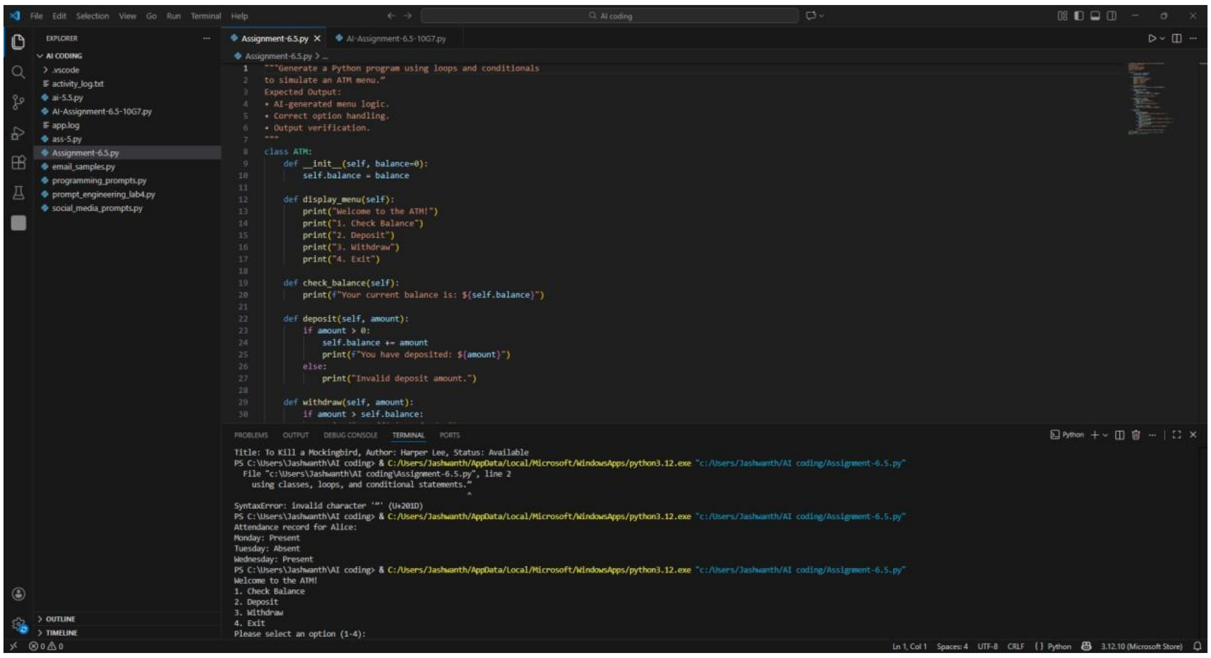
TASK-5:

Prompt:

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

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

CODE:



```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}")

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"You have deposited: ${amount}")
        else:
            print("Invalid deposit amount.")

    def withdraw(self, amount):
        if amount > self.balance:
```

```
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
  File "c:\Users\Jashwanth\AI coding\Assignment-6.5.py", line 2
    using classes, loops, and conditional statements.”
                                                     ^
SyntaxError: invalid character '”' (U+201D)
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Attendance record for Alice:
Monday: Present
Tuesday: Absent
Wednesday: Present
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/Assignment-6.5.py"
Welcome to the ATM!
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Please select an option (1-4):
```

OBSERVATION:

- Implements deposit, withdraw, and balance check methods.
- Menu-driven interface allows user interaction.

- Deposit/withdraw logic is correct; balance display needs a small fix (use fstring).

- Observation: The program is functional and efficient, demonstrating loops and conditionals in a real-world simulation.