

School of Computer Science and Artificial Intelligence

Lab Assignment # 6.5

Program : B. Tech (CSE)
Specialization : -
Course Title : AI Assisted Coding
Course Code : 23CS002PC304
Semester II
Academic Session : 2025-2026
Name of Student : P.Eshwar
Enrollment No. : 2403A51L26
Batch No. 51
Date : 06/02/26

Submission Starts here

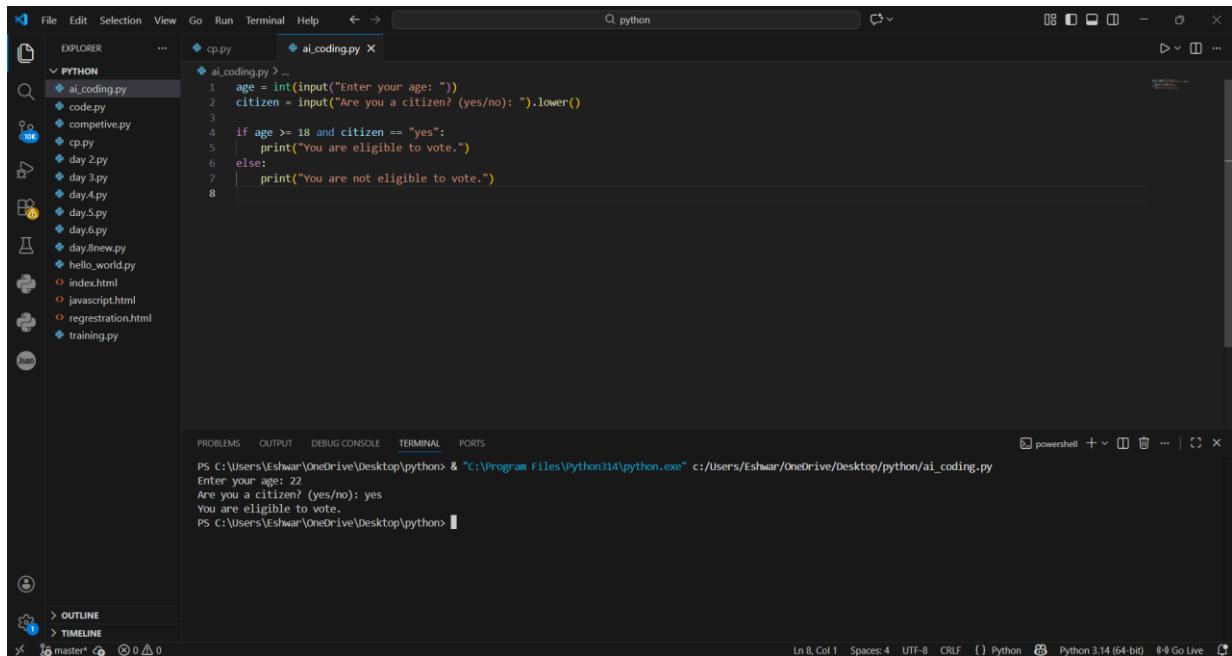
Screenshots:

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.”



The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing various Python files and other file types like HTML and JSON. The main editor area contains the following Python code:

```
ai_coding.py > ...
1 age = int(input("Enter your age: "))
2 citizen = input("Are you a citizen? (yes/no): ").lower()
3
4 if age >= 18 and citizen == "yes":
5     print("You are eligible to vote.")
6 else:
7     print("You are not eligible to vote.")
```

Below the editor is a terminal window titled "python" showing the execution of the code. The terminal output is:

```
PS C:\Users\Eshwar\OneDrive\Desktop> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
Enter your age: 22
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS C:\Users\Eshwar\OneDrive\Desktop>
```

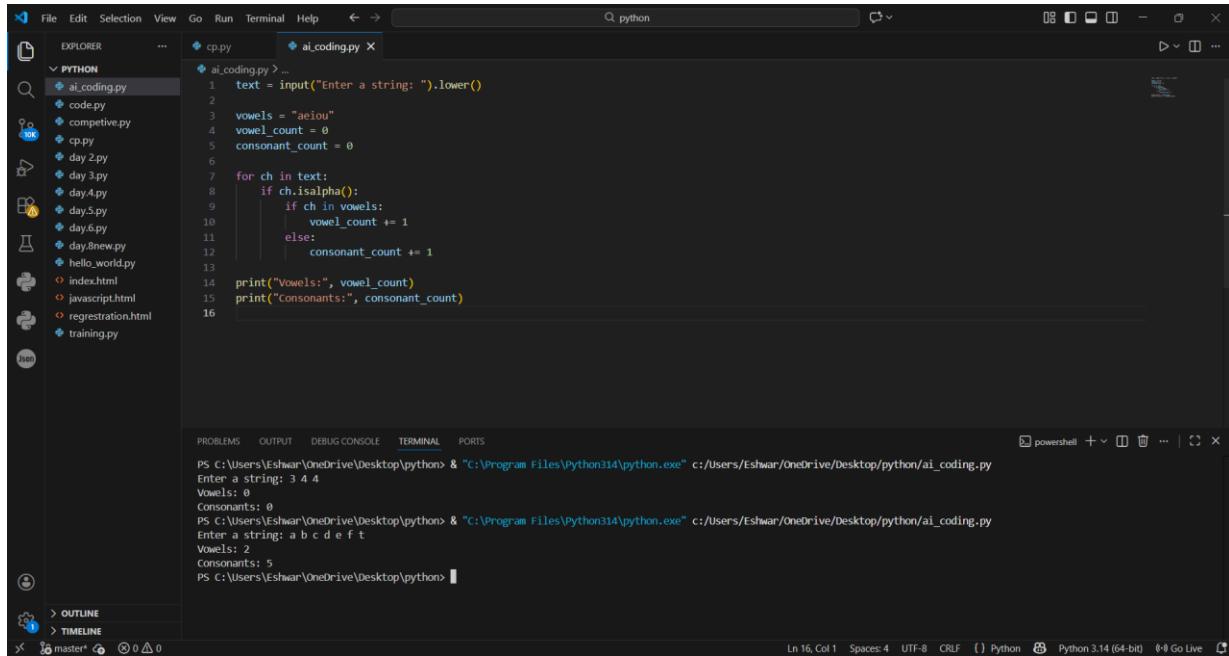
At the bottom of the interface, there are status bars for "Ln 8, Col 1", "Spaces: 4", "UTF-8", "CRLF", "Python", "Python 3.14 (64-bit)", and "Go Live".

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.”



```

File Edit Selection View Go Run Terminal Help ← → Q python
EXPLORER ai_coding.py
PYTHON ai_coding.py ...
ai_coding.py > ...
1 text = input("Enter a string: ").lower()
2
3 vowels = "aeiou"
4 vowel_count = 0
5 consonant_count = 0
6
7 for ch in text:
8     if ch.isalpha():
9         if ch in vowels:
10             vowel_count += 1
11         else:
12             consonant_count += 1
13
14 print("Vowels:", vowel_count)
15 print("Consonants:", consonant_count)
16

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Eshwar\OneDrive\Desktop> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
Enter a string: 3 4 4
Vowels: 0
Consonants: 0
PS C:\Users\Eshwar\OneDrive\Desktop> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
Enter a string: a b c d e f t
Vowels: 2
Consonants: 5
PS C:\Users\Eshwar\OneDrive\Desktop>

Ln 16, Col 1 Spaces:4 UTF-8 CRLF () Python ⚙ Python 3.14 (64-bit) ⓘ Go Live

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.”

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows a file tree with a folder named "PYTHON" containing files: ai_coding.py, cp.py, competitive.py, day 2.py, day 3.py, day 4.py, day 5.py, day 6.py, day 8new.py, hello_world.py, index.html, javascript.html, registration.html, and training.py.
- Editor (Center):** Displays the content of the file "ai_coding.py". The code defines a class `Library` with methods for adding, removing, and displaying books. It also includes a main loop for a library menu.

```
File Edit Selection View Go Run Terminal Help <- > Q python ... EXPLORE PYTHON ai_coding.py cp.py ai_coding.py > ... 1 class Library: 2     def __init__(self): 3         self.books = [] 4 5     def add_book(self, book_name): 6         self.books.append(book_name) 7         print(f'{book_name} added to the library.') 8 9     def remove_book(self, book_name): 10        if book_name in self.books: 11            self.books.remove(book_name) 12            print(f'{book_name} removed from the library.') 13        else: 14            print("Book not found.") 15 16     def display_books(self): 17        if self.books: 18            print("Available books:") 19            for book in self.books: 20                print("-", book) 21        else: 22            print("No books available.") 23 24 25 library = Library() 26 27 while True: 28     print("\nLibrary Menu") 29     print("1. Add Book") 30     print("2. Remove Book") 31     print("3. Display Books") 32     print("4. Exit") 33 34     choice = int(input("Enter your choice: ")) 35 36     if choice == 1: 37         book = input("Enter book name: ")
```

- Search Bar (Top):** Contains the text "python".
- Bottom Status Bar:** Shows "Ln 53, Col 1 Spaces: 4 UTF-8 CRLF () Python Python 3.14 (64-bit)" and a "Go Live" button.

Ouput:

The screenshot shows a Visual Studio Code (VS Code) interface. The left sidebar (Explorer) lists files in a 'PYTHON' folder: ai_coding.py, code.py, competitive.py, cp.py, day 2.py, day 3.py, day 4.py, day 5.py, day 6.py, day 8new.py, hello_world.py, index.html, javascript.html, registration.html, and training.py. The 'ai_coding.py' file is open in the main editor area, displaying Python code for a library system. The code includes functions for adding books, removing books, displaying books, and exiting the system. The status bar at the bottom shows the command prompt PS C:\Users\Eshwar\OneDrive\Desktop\python & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py, and the Python version Python 3.14 (64-bit). The bottom right corner has a 'Go Live' button.

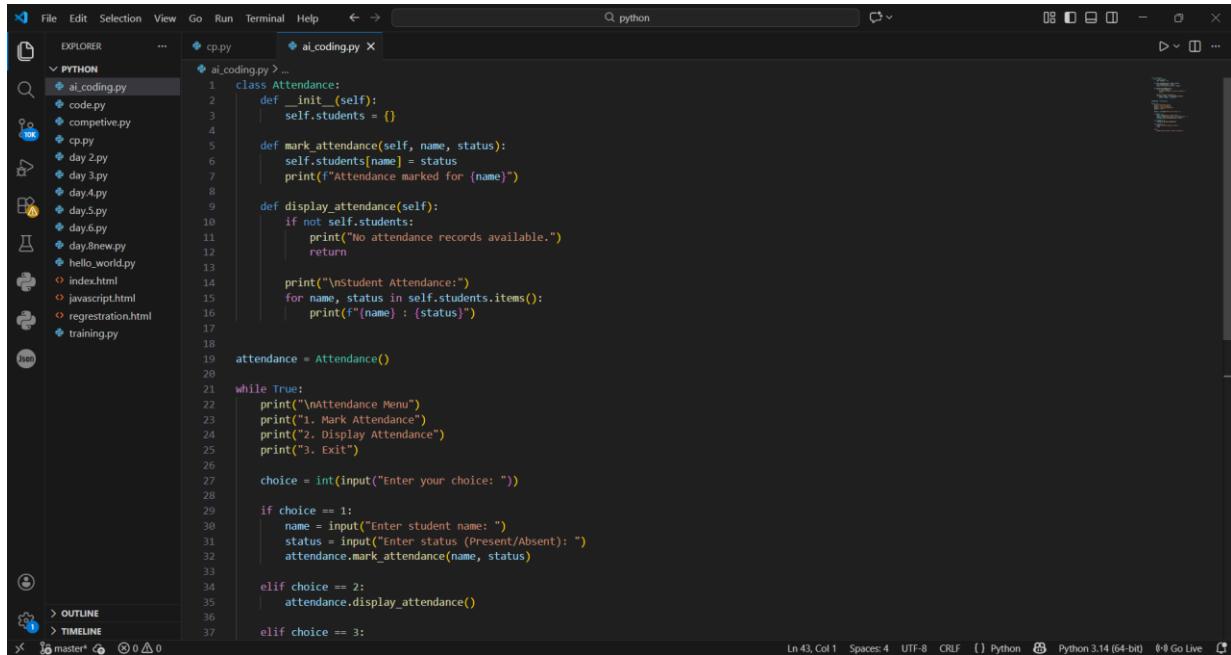
```
38     library.add_book(book)
39
40     elif choice == 2:
41         book = input("Enter book name to remove: ")
42         library.remove_book(book)
43
44     elif choice == 3:
45         library.display_books()
46
47     elif choice == 4:
48         print("Exiting Library System.")
49         break
50
51     else:
52         print("Invalid choice. Please try again.")
```

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.”

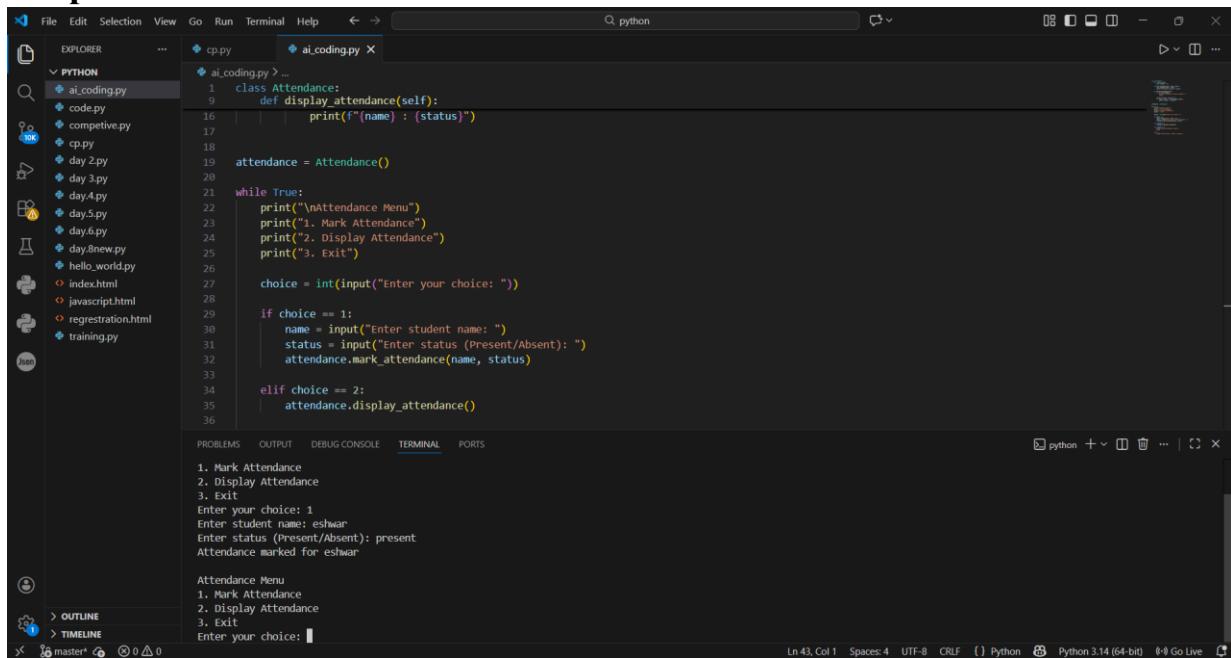


```

File Edit Selection View Go Run Terminal Help ← → 🔍 python
EXPLORER ... ai_coding.py X
ai_coding.py > ...
1 class Attendance:
2     def __init__(self):
3         self.students = {}
4
5     def mark_attendance(self, name, status):
6         self.students[name] = status
7         print(f"Attendance marked for {name}")
8
9     def display_attendance(self):
10        if not self.students:
11            print("No attendance records available.")
12            return
13
14        print("\nStudent Attendance:")
15        for name, status in self.students.items():
16            print(f"{name} : {status}")
17
18    attendance = Attendance()
19
20    while True:
21        print("\nAttendance Menu")
22        print("1. Mark Attendance")
23        print("2. Display Attendance")
24        print("3. Exit")
25
26        choice = int(input("Enter your choice: "))
27
28        if choice == 1:
29            name = input("Enter student name: ")
30            status = input("Enter status (Present/Absent): ")
31            attendance.mark_attendance(name, status)
32
33        elif choice == 2:
34            attendance.display_attendance()
35
36        elif choice == 3:
37
Ln 43, Col 1  Spaces: 4  CRLF  {} Python  Python 3.14 (64-bit)  Go Live

```

Output:



```

File Edit Selection View Go Run Terminal Help ← → 🔍 python
EXPLORER ... ai_coding.py X
ai_coding.py > ...
1 class Attendance:
2     def __init__(self):
3         self.students = {}
4
5     def mark_attendance(self, name, status):
6         self.students[name] = status
7         print(f"{name} : {status}")
8
9     def display_attendance(self):
10        if not self.students:
11            print("No attendance records available.")
12            return
13
14        print("\nStudent Attendance:")
15        for name, status in self.students.items():
16            print(f"{name} : {status}")
17
18    attendance = Attendance()
19
20    while True:
21        print("\nAttendance Menu")
22        print("1. Mark Attendance")
23        print("2. Display Attendance")
24        print("3. Exit")
25
26        choice = int(input("Enter your choice: "))
27
28        if choice == 1:
29            name = input("Enter student name: ")
30            status = input("Enter status (Present/Absent): ")
31            attendance.mark_attendance(name, status)
32
33        elif choice == 2:
34            attendance.display_attendance()
35
36
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice: 1
Enter student name: eshwar
Enter status (Present/Absent): present
Attendance marked for eshwar

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

```

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.”



The screenshot shows a Python code editor interface with the following details:

- File Explorer:** On the left, it lists files under the "PYTHON" folder, including "ai_coding.py", "code.py", "competitive.py", "cp.py", "day 2.py", "day 3.py", "day 4.py", "day 5.py", "day 6.py", "day 8new.py", "hello_world.py", "index.html", "javascript.html", and "registration.html".
- Code Editor:** The main area displays the content of "ai_coding.py". The code is a script for an ATM menu. It starts by defining a variable "balance" and then enters a loop where it prints the menu options (1. Check Balance, 2. Deposit Money, 3. Withdraw Money, 4. Exit) and takes user input. Based on the choice, it performs the corresponding action: depositing or withdrawing money from the balance, or exiting the loop.
- Status Bar:** At the bottom, it shows "Ln 37, Col 1" and other standard status bar information.

```
File Edit Selection View Go Run Terminal Help < - > python
EXPLORER PYTHON ai_coding.py cp.py ai_coding.py x
ai_coding.py > ...
1     balance = 10000
2
3     while True:
4         print("\nATM Menu")
5         print("1. Check Balance")
6         print("2. Deposit Money")
7         print("3. Withdraw Money")
8         print("4. Exit")
9
10        choice = int(input("Enter your choice: "))
11
12        if choice == 1:
13            print("Current Balance: ₹", balance)
14
15        elif choice == 2:
16            amount = int(input("Enter amount to deposit: "))
17            if amount > 0:
18                balance += amount
19                print("Deposit successful.")
20            else:
21                print("Invalid deposit amount.")
22
23        elif choice == 3:
24            amount = int(input("Enter amount to withdraw: "))
25            if amount > 0 and amount <= balance:
26                balance -= amount
27                print("Please collect your cash.")
28            else:
29                print("Insufficient balance or invalid amount.")
30
31        elif choice == 4:
32            print("Thank you for using the ATM.")
33            break
34
35        else:
36            print("Invalid choice. Please try again.")
```

Output:

The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows a folder structure under "PYTHON" containing files like "ai_coding.py", "codepy", "competitive.py", "day 2.py", "day 3.py", "day 4.py", "day 5.py", "day 6.py", "day 8new.py", "hello_world.py", "index.html", "javascript.html", "registration.html", and "training.py".
- Search Bar:** Displays "python".
- Code Editor:** The active file is "ai_coding.py" which contains a script for an ATM system. The code uses a loop to present a menu with options 1 through 4. It handles depositing money by adding it to a balance variable if the amount is greater than zero.
- Terminal:** Below the code editor, the terminal window shows the execution of the script. It prints the menu, asks for a choice (2), asks for a deposit amount (5000), and then prints "Deposit successful."
- Bottom Status Bar:** Shows "Ln 37, Col 1" and other system information.