

AI ASSISTED CODING

2303A52454

B-34

LAB-6.5

Task Description #1 (AI-Based Code Completion for Conditional Eligibility Check)

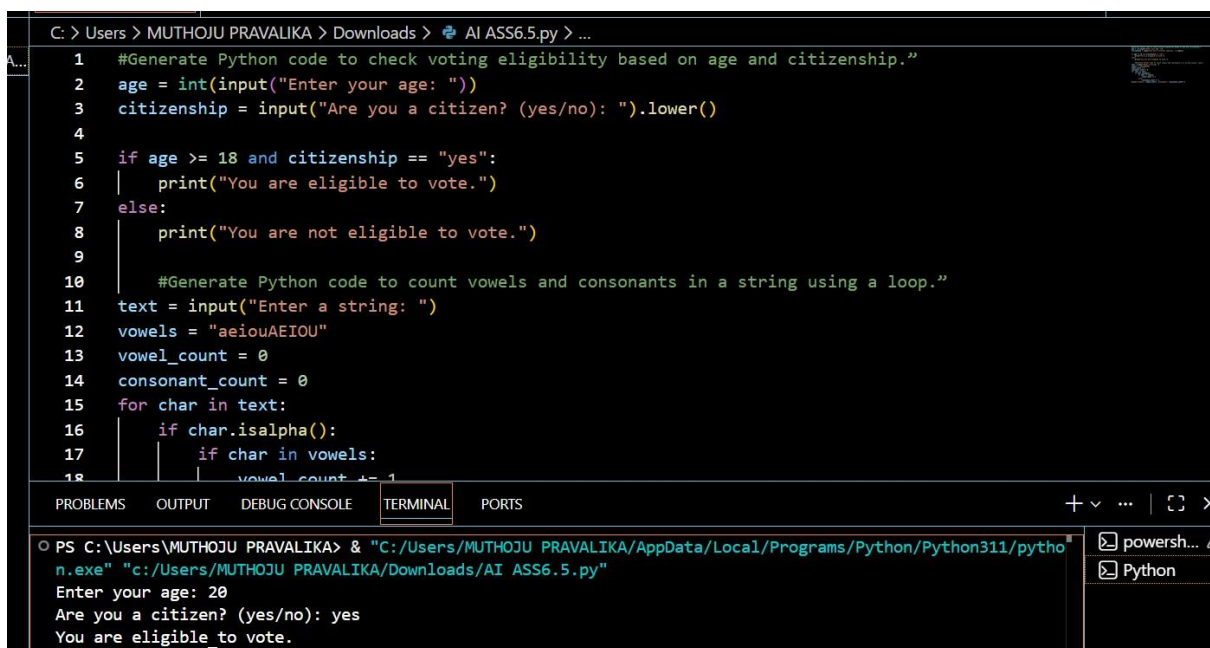
Task: Use an AI tool to generate eligibility logic.

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.



The screenshot shows a Python IDE with a dark theme. The main editor displays two Python scripts. The first script checks voting eligibility based on age and citizenship. The second script counts vowels and consonants in a string using a loop. Below the editor, there is a terminal window showing the execution of the first script. The terminal output shows the user entering '20' for age and 'yes' for citizenship, resulting in the message 'You are eligible to vote.'.

```
C: > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > ...  
1 #Generate Python code to check voting eligibility based on age and citizenship."  
2 age = int(input("Enter your age: "))  
3 citizenship = input("Are you a citizen? (yes/no): ").lower()  
4  
5 if age >= 18 and citizenship == "yes":  
6     print("You are eligible to vote.")  
7 else:  
8     print("You are not eligible to vote.")  
9  
10 #Generate Python code to count vowels and consonants in a string using a loop."  
11 text = input("Enter a string: ")  
12 vowels = "aeiouAEIOU"  
13 vowel_count = 0  
14 consonant_count = 0  
15 for char in text:  
16     if char.isalpha():  
17         if char in vowels:  
18             vowel_count += 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/pytho
n.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"
Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.

Task Description #2(AI-Based Code Completion for Loop-Based String Processing)

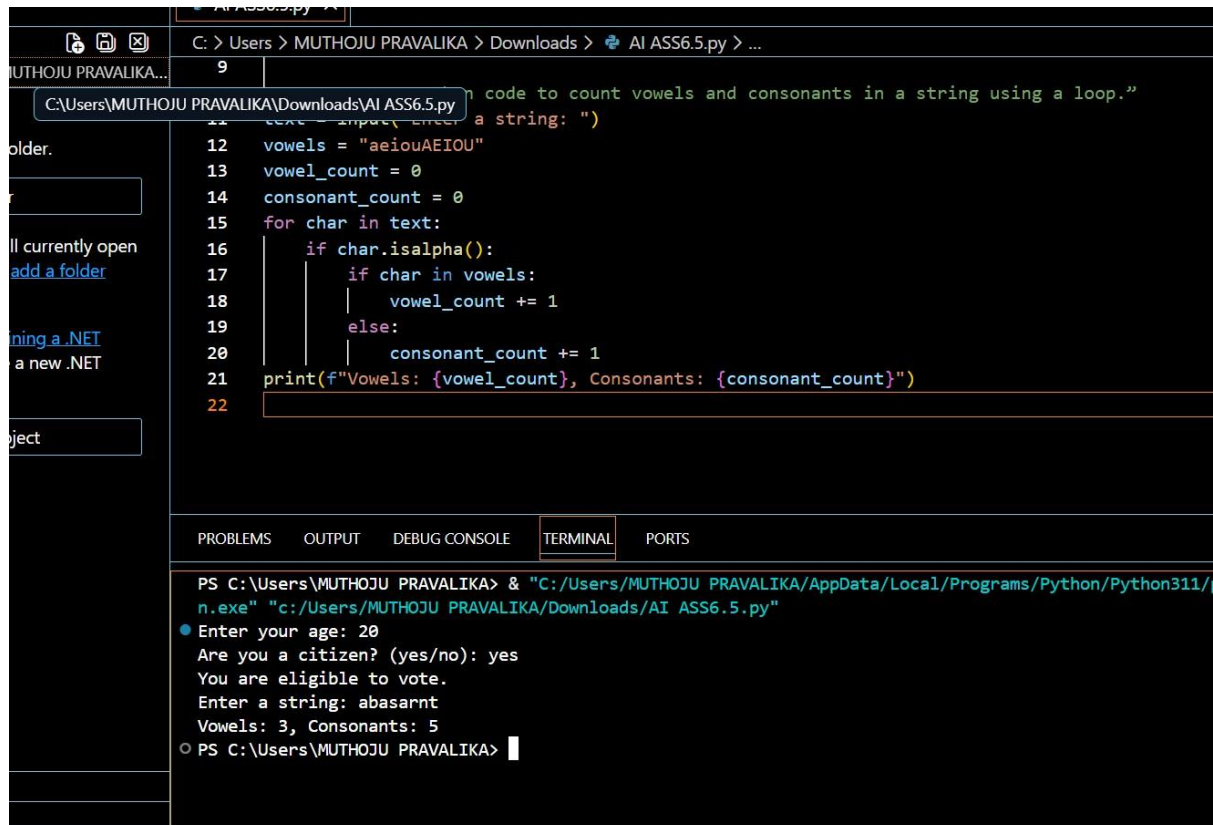
Task: Use an AI tool to process strings using loops.

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.



The screenshot shows a Visual Studio Code editor window. The top pane displays a Python script named `AI ASS6.5.py` located at `C:\Users\MUTHOJU PRAVALIKA\Downloads\AI ASS6.5.py`. The script is designed to count vowels and consonants in a string using a loop. The code is as follows:

```
11 # Prompt user to enter a string: ")
12 vowels = "aeiouAEIOU"
13 vowel_count = 0
14 consonant_count = 0
15 for char in text:
16     if char.isalpha():
17         if char in vowels:
18             vowel_count += 1
19         else:
20             consonant_count += 1
21 print(f"Vowels: {vowel_count}, Consonants: {consonant_count}")
22
```

The bottom pane shows the terminal output of the script. The command prompt is `PS C:\Users\MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"`. The output is as follows:

```
● Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.
Enter a string: abasarnt
Vowels: 3, Consonants: 5
○ PS C:\Users\MUTHOJU PRAVALIKA>
```

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.

```
Go Run Terminal Help ← → Q Search
... AI ASS6.5.py
C:\Users\MUTHOJU PRAVALIKA\Downloads> AI ASS6.5.py > Book
1 #Generate a Python program for a library management system using classes, loops, and conditional statements."
2 class Book:
3     def __init__(self, title, author):
4         self.title = title
5         self.author = author
6         self.is_available = True
7 class Library:
8     def __init__(self):
9         self.books = []
10    def add_book(self, book):
11        self.books.append(book)
12    def display_books(self):
13        for book in self.books:
14            status = "Available" if book.is_available else "Checked Out"
15            print(f>Title: {book.title}, Author: {book.author}, Status: {status}")
16    def check_out_book(self, title):
17        for book in self.books:
18            if book.title == title and book.is_available:
19                book.is_available = False
20                print(f"You have checked out '{title}'.")
21                return
22            print(f"Sorry, '{title}' is not available.")
23    def return_book(self, title):
24        for book in self.books:
25            if book.title == title and not book.is_available:
26                book.is_available = True
27                print(f"You have returned '{title}'.")
28                return
29            print(f"'{title}' was not checked out.")
30    library = Library()
31    library.add_book(Book("1984", "George Orwell"))
32    library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
33    library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
34    while True:
35        print("\nLibrary Menu:")
36        print("1. Display Books")
37        print("2. Check Out Book")
38        print("3. Return Book")
39        print("4. Exit")
40        choice = input("Enter your choice (1-4): ")
41
42        if choice == '1':
43            library.display_books()
44        elif choice == '2':
45            title = input("Enter the title of the book to check out: ")
```

```
42     if choice == '1':
43         library.display_books()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Library Menu:
1. Display Books
2. Check Out Book
3. Return Book
4. Exit
Enter your choice (1-4): 1
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
Title: The Great Gatsby, Author: F. Scott Fitzgerald, Status: Available
Library Menu:
```

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.

The screenshot shows a code editor with a file named 'AI ASS6.5.py 3'. The code defines an 'Attendance' class with methods for marking attendance and displaying it. The terminal output shows the program running successfully, displaying 'Alice: Present' and 'Bob: Absent'.

```

1
2
3
4
5
6 #Generate a Python class to mark and display student attendance using loops
7 class Attendance:
8     def __init__(self):
9         self.records = {}
10
11     def mark(self, name, status):
12         status = status.capitalize()
13         if status in ["Present", "Absent"]:
14             self.records[name] = status
15         else:
16             print("Invalid status")
17
18     def display(self):
19         for name, status in self.records.items():
20             print(f"{name}: {status}")
21 # Example usage
22 attendance = Attendance()
23 attendance.mark("Alice", "Present")
24 attendance.mark("Bob", "Absent")
25 attendance.display()
26 Alice: Present
27
28
29
30
31

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MUTHOJU PRAVALIKA> "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"
Alice: Present
Bob: Absent
PS C:\Users\MUTHOJU PRAVALIKA>

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.

ViewGoRunTerminalHelp←→Q Search

...AI ASS6.5.py X

C:\Users\MUTHOJU PRAVALIKA> Downloads > AI ASS6.5.py > ...

es\MUTHOJU PRAVALIKA...

3

4 #Generate a Python program using loops and conditionalsto simulate an ATM menu.

5 def atm_menu():

6 balance=1000 # Initial balance

7 while True:

8 print("\nWelcome to the ATM")

9 print("1. Check Balance")

10 print("2. Deposit Money")

11 print("3. Withdraw Money")

12 print("4. Exit")

13

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

15

16 if choice == '1':

17 | print(f"Your current balance is: \${balance}")

18

19 elif choice == '2':

20 deposit = float(input("Enter amount to deposit: \$"))

21 if deposit > 0:

22 | balance += deposit

23 | print(f"\${deposit} deposited successfully.")

24 else:

25 | print("Invalid amount. Please try again.")

26

27 elif choice == '3':

28 withdraw = float(input("Enter amount to withdraw: \$"))

29 if 0 < withdraw <= balance:

30 | balance -= withdraw

31 | print(f"\${withdraw} withdrawn successfully.")

32 else:

33 | print("Insufficient funds or invalid amount. Please try again.")

34

35 elif choice == '4':

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTS

PS C:\Users\MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads...

3. Withdraw Money

4. Exit

Please select an option (1-4): 1

Your current balance is: \$1000

Wellcome to the ATM

1. Check Balance

2. Deposit Money

3. Withdraw Money

4. Exit