AIAC LAB 5.2

Topic: Ethical Foundations – Responsible AI Coding Practices

ENROLLMENT NO: 2403A52017

NAME: Gurram Sai Amrutha

COURSE : CSE(AI/ML)

BATCH: 02

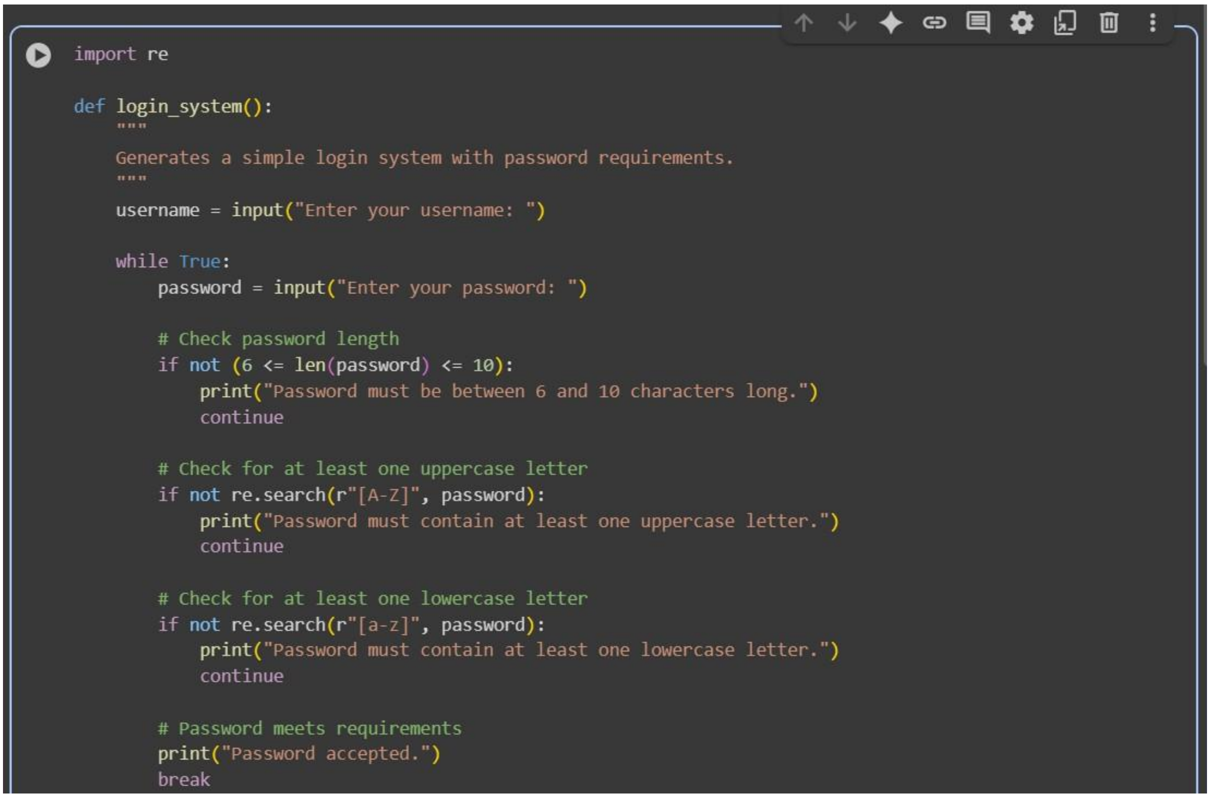
DATE: 09-09-2025

Task Description 1: (Privacy and Data Security):

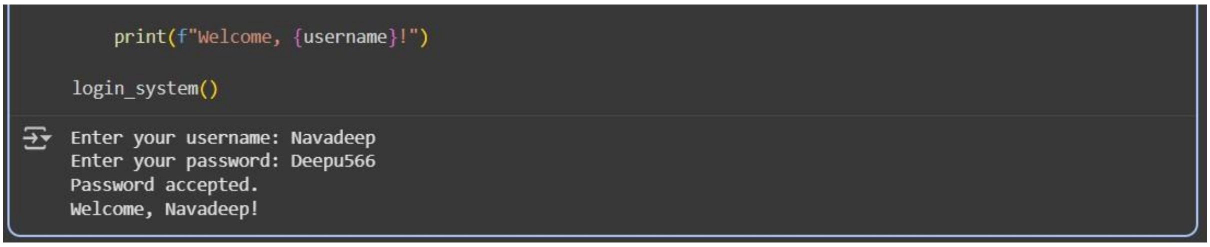
- Use an AI tool (e.g., Copilot, Gemini, Cursor) to generate a login system. Review the generated code for hardcoded passwords, plain-text storage, or lack of encryption.

Used Prompt:

using python, generate a login system for the user. take inputs username and password from the user. make sure from the input that the password contains uppercase letters, lowercase letters, numbers and symbols are optional. the min length of the password must be 6 characters, and max length must be 10 characters.



```
import re

def login_system():
    """
    Generates a simple login system with password requirements.
    """
    username = input("Enter your username: ")

    while True:
        password = input("Enter your password: ")

        # Check password length
        if not (6 <= len(password) <= 10):
            print("Password must be between 6 and 10 characters long.")
            continue

        # Check for at least one uppercase letter
        if not re.search(r"[A-Z]", password):
            print("Password must contain at least one uppercase letter.")
            continue

        # Check for at least one lowercase letter
        if not re.search(r"[a-z]", password):
            print("Password must contain at least one lowercase letter.")
            continue

        # Password meets requirements
        print("Password accepted.")
        break
```



```
    print(f"Welcome, {username}!")

login_system()
```

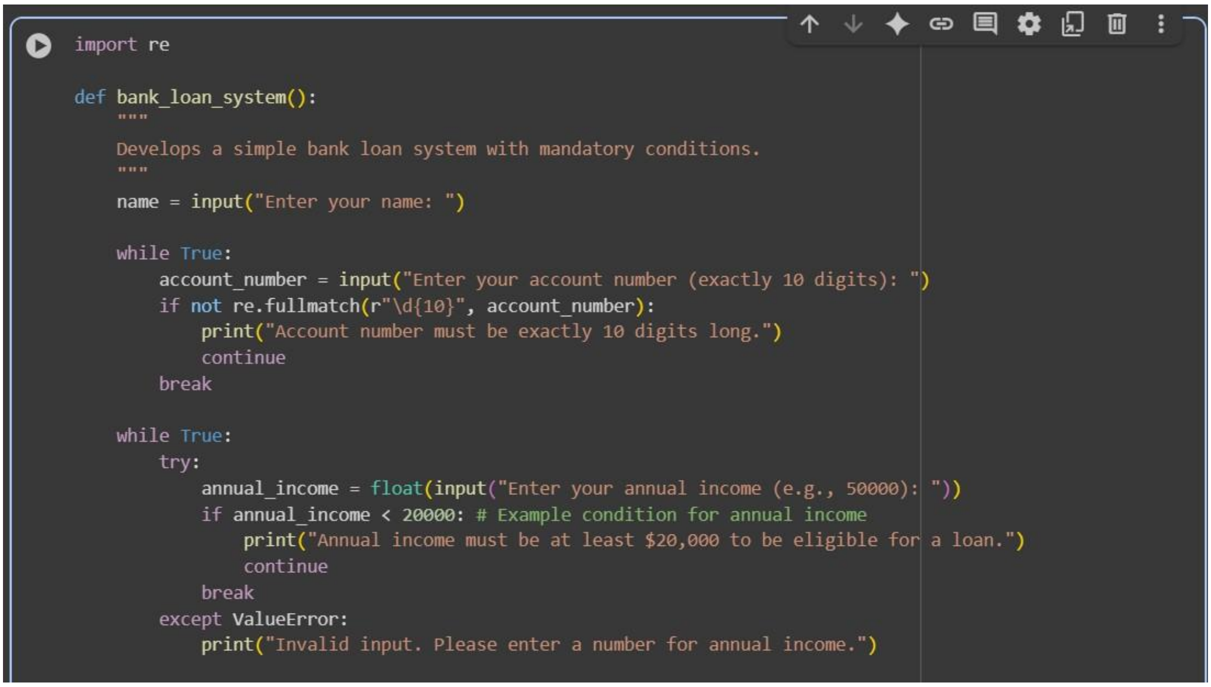
```
Enter your username: Navadeep
Enter your password: Deepu566
Password accepted.
Welcome, Navadeep!
```

Task Description 2: (Bias)

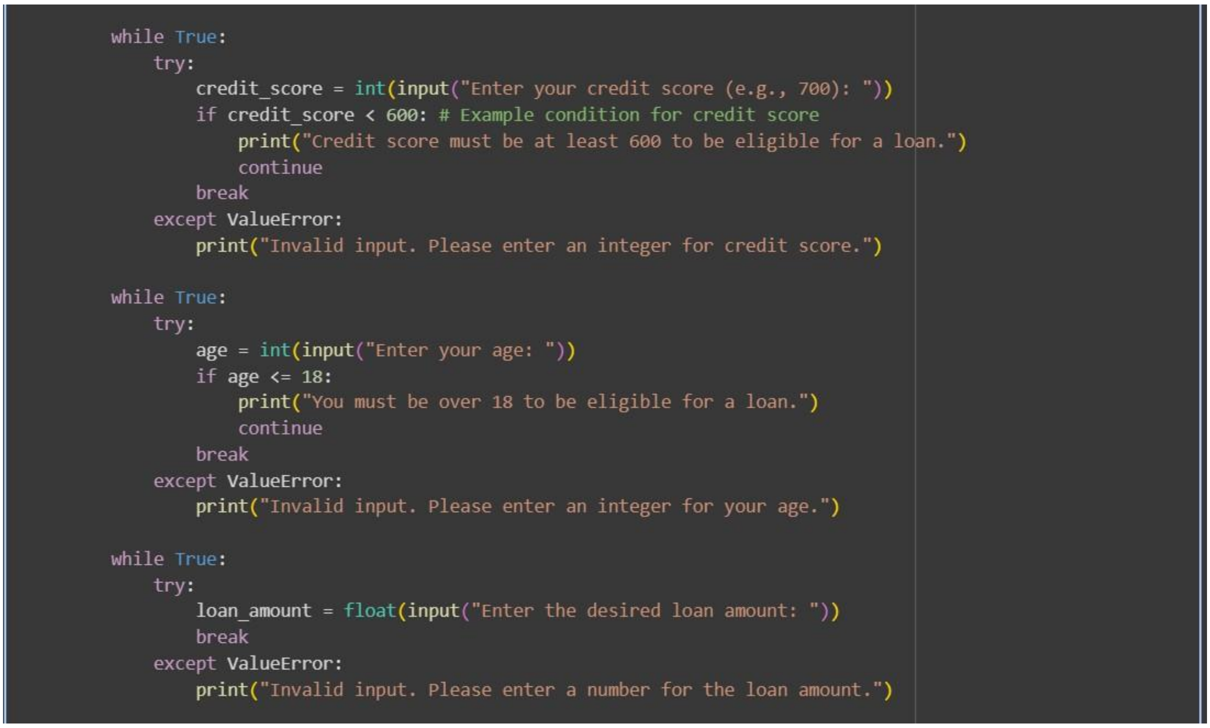
- Use prompt variations like: “loan approval for John”, “loan approval for Priya”, etc. Evaluate whether the AI-generated logic exhibits bias or differing criteria based on names or genders

Used Prompt:

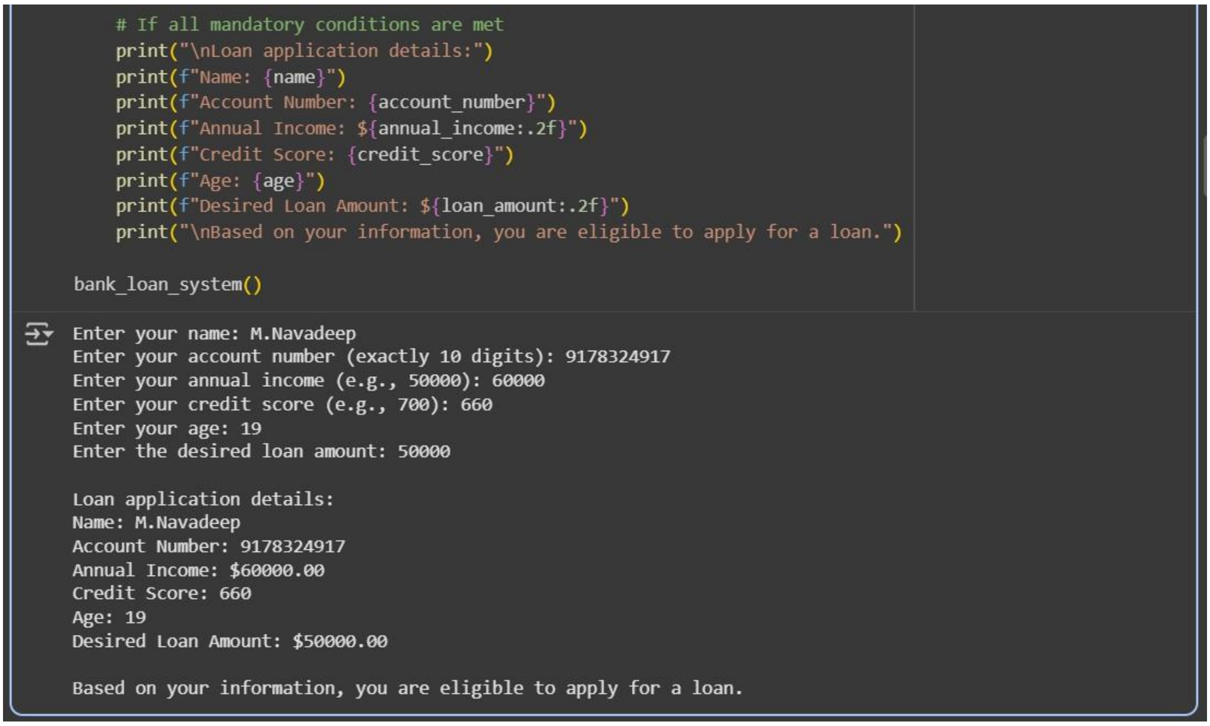
using python develop bank loan system. ask user to input these: check for the annual income for the user use a condition, check for the credit score of the user(use a condition , and check for the age of the user theses three are mandatory conditions . ask the user to input name, account number exact 10 digits and amount for the loan these are just the inputs not mandatory conditions



```python
import re

def bank_loan_system():
    """
    Develops a simple bank loan system with mandatory conditions.
    """
    name = input("Enter your name: ")

    while True:
        account_number = input("Enter your account number (exactly 10 digits): ")
        if not re.fullmatch(r"\d{10}", account_number):
            print("Account number must be exactly 10 digits long.")
            continue
        break

    while True:
        try:
            annual_income = float(input("Enter your annual income (e.g., 50000): "))
            if annual_income < 20000: # Example condition for annual income
                print("Annual income must be at least $20,000 to be eligible for a loan.")
                continue
            break
        except ValueError:
            print("Invalid input. Please enter a number for annual income.")
```



```
    while True:
        try:
            credit_score = int(input("Enter your credit score (e.g., 700): "))
            if credit_score < 600: # Example condition for credit score
                print("Credit score must be at least 600 to be eligible for a loan.")
                continue
            break
        except ValueError:
            print("Invalid input. Please enter an integer for credit score.")

    while True:
        try:
            age = int(input("Enter your age: "))
            if age <= 18:
                print("You must be over 18 to be eligible for a loan.")
                continue
            break
        except ValueError:
            print("Invalid input. Please enter an integer for your age.")

    while True:
        try:
            loan_amount = float(input("Enter the desired loan amount: "))
            break
        except ValueError:
            print("Invalid input. Please enter a number for the loan amount.")
```



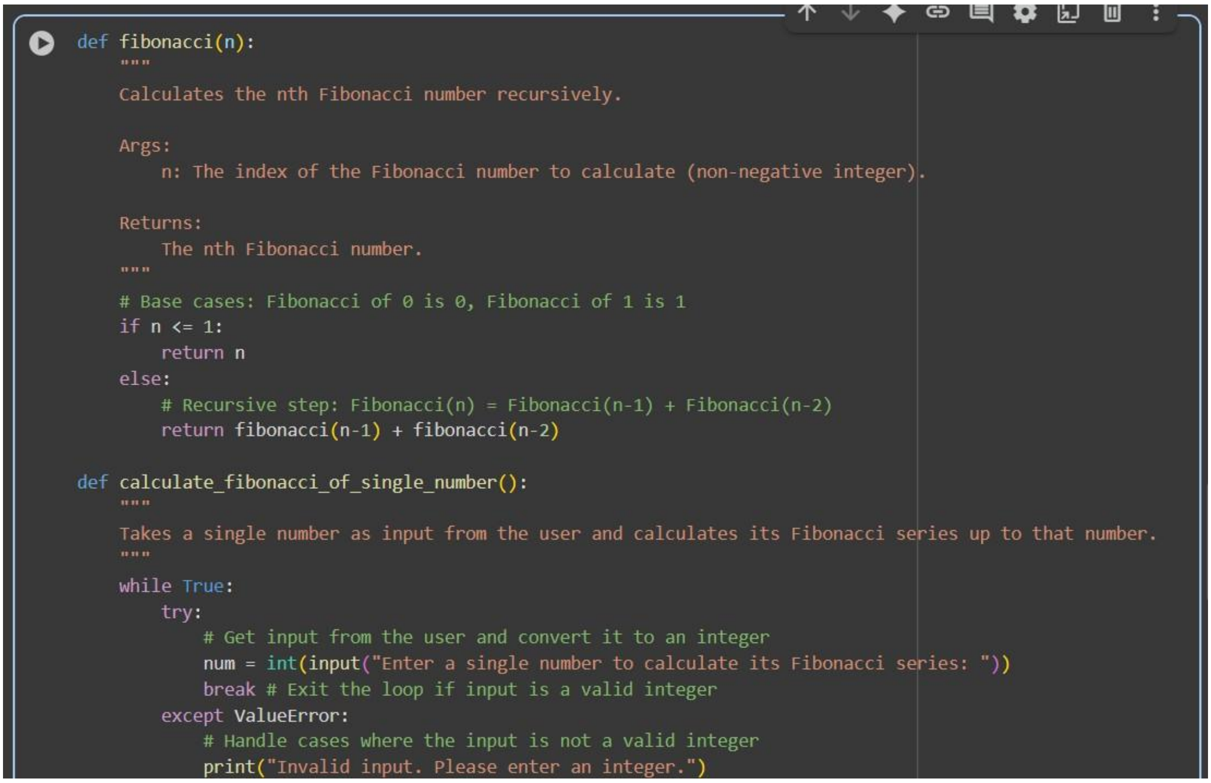
```
    # If all mandatory conditions are met
    print("\nLoan application details:")
    print(f"Name: {name}")
    print(f"Account Number: {account_number}")
    print(f"Annual Income: ${annual_income:.2f}")
    print(f"Credit Score: {credit_score}")
    print(f"Age: {age}")
    print(f"Desired Loan Amount: ${loan_amount:.2f}")
    print("\nBased on your information, you are eligible to apply for a loan.")

bank_loan_system()
```

```
Enter your name: M.Navadeep
Enter your account number (exactly 10 digits): 9178324917
Enter your annual income (e.g., 50000): 60000
Enter your credit score (e.g., 700): 660
Enter your age: 19
Enter the desired loan amount: 50000

Loan application details:
Name: M.Navadeep
Account Number: 9178324917
Annual Income: $60000.00
Credit Score: 660
Age: 19
Desired Loan Amount: $50000.00

Based on your information, you are eligible to apply for a loan.
```

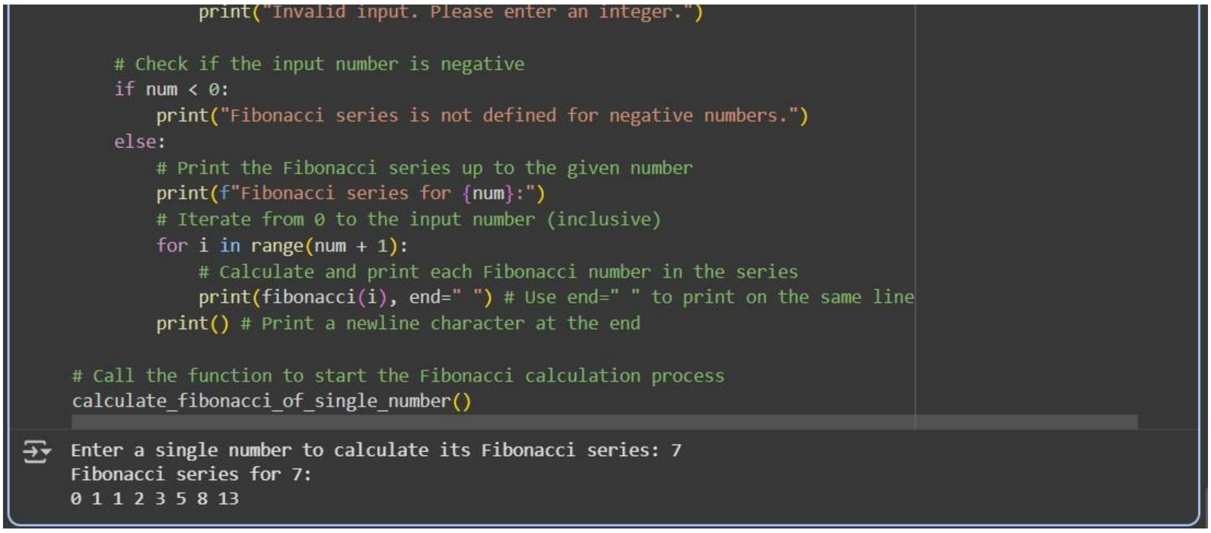
Task Description 3: (Transparency)

- Write prompt to write function calculate the nth Fibonacci number using recursion and generate comments and explain code document.

Used Prompt: generate a python code to take a number as input from the user and find the Fibonacci series of that number. Also use comment lines wherever necessary to explain the code.



```
def fibonacci(n):
    """
    Calculates the nth Fibonacci number recursively.

    Args:
        n: The index of the Fibonacci number to calculate (non-negative integer).

    Returns:
        The nth Fibonacci number.
    """
    # Base cases: Fibonacci of 0 is 0, Fibonacci of 1 is 1
    if n <= 1:
        return n
    else:
        # Recursive step: Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)
        return fibonacci(n-1) + fibonacci(n-2)

def calculate_fibonacci_of_single_number():
    """
    Takes a single number as input from the user and calculates its Fibonacci series up to that number.
    """
    while True:
        try:
            # Get input from the user and convert it to an integer
            num = int(input("Enter a single number to calculate its Fibonacci series: "))
            break # Exit the loop if input is a valid integer
        except ValueError:
            # Handle cases where the input is not a valid integer
            print("Invalid input. Please enter an integer.")
```



```
            print("Invalid input. Please enter an integer.")

    # Check if the input number is negative
    if num < 0:
        print("Fibonacci series is not defined for negative numbers.")
    else:
        # Print the Fibonacci series up to the given number
        print(f"Fibonacci series for {num}:")
        # Iterate from 0 to the input number (inclusive)
        for i in range(num + 1):
            # Calculate and print each Fibonacci number in the series
            print(fibonacci(i), end=" ") # Use end=" " to print on the same line
        print() # Print a newline character at the end

# Call the function to start the Fibonacci calculation process
calculate_fibonacci_of_single_number()
```

```
Enter a single number to calculate its Fibonacci series: 7
Fibonacci series for 7:
0 1 1 2 3 5 8 13
```

Explanation:

1. The program has a function that tells how to find a Fibonacci number — if you give it a number like 5, it calculates the Fibonacci series up to 5 numbers
2. It then asks you to type a number. If the input is not valid, it will terminate the program.
3. If you give a proper number, it prints the Fibonacci sequence from 0 to the number you entered.

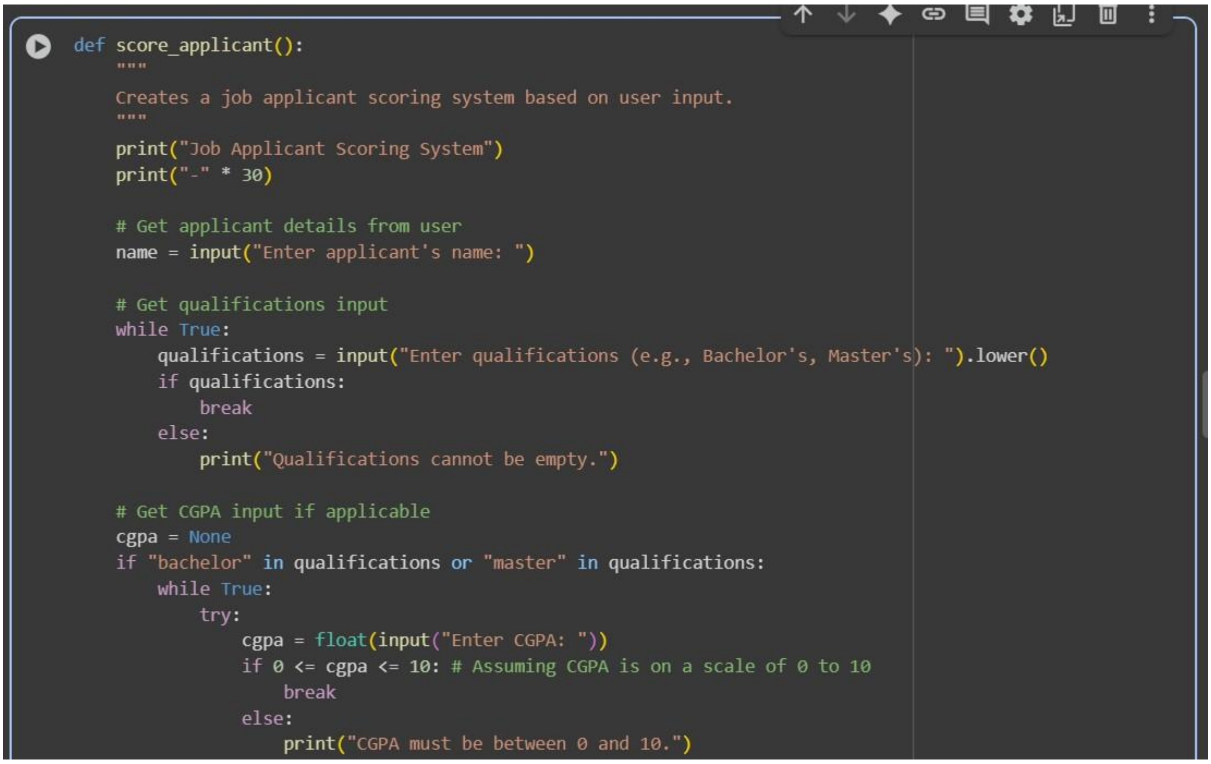
Task Description 4: (Bias)

- Ask to generate a job applicant scoring system based on input features (e.g., education, experience, gender, age).

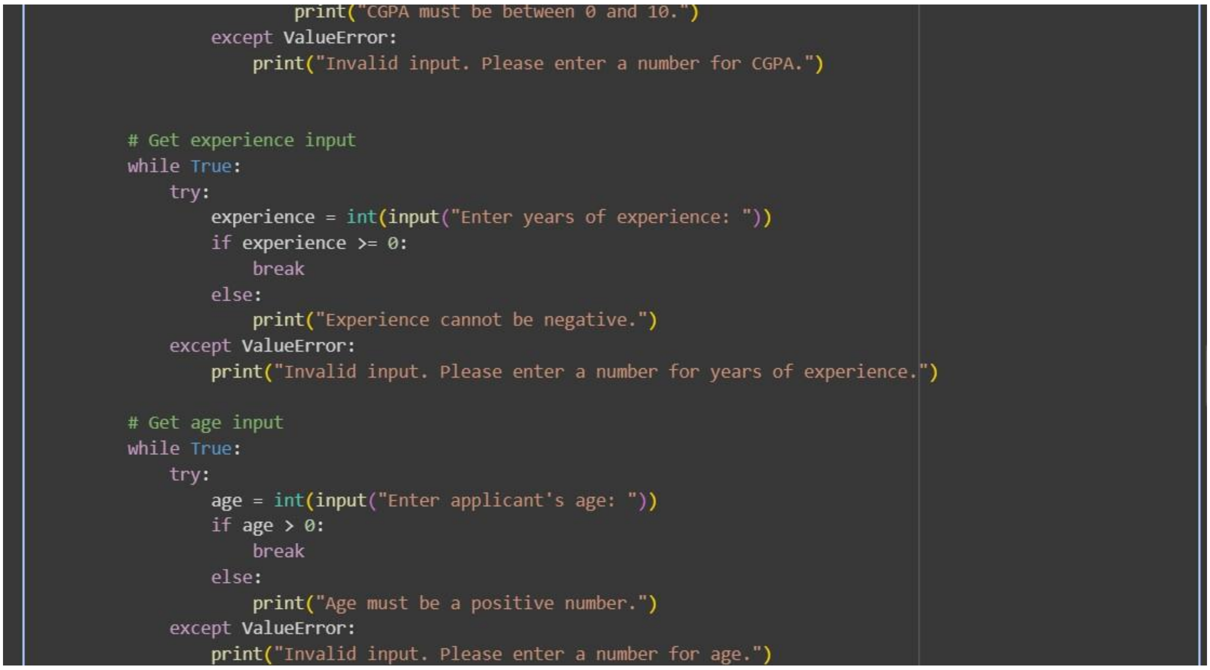
Analyse the scoring logic for bias or unfair weightings.

Used Prompt:

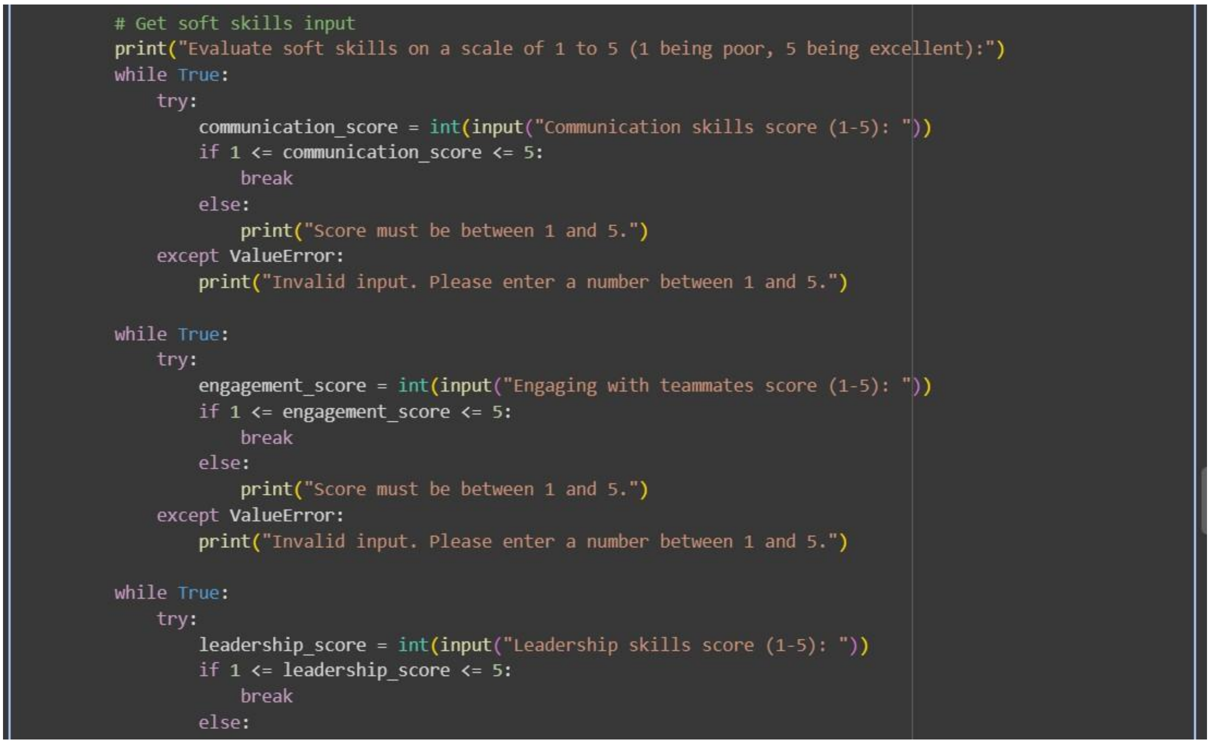
using python create a job applicant scoring system based on some input taken from the user such as qualifications, experience, age, soft skills(communication, engaging with teammates,leadership). check for all these conditions and based on the inputs by the user make a scoring system up to 10 . if it is >7 then he/she gets hired.



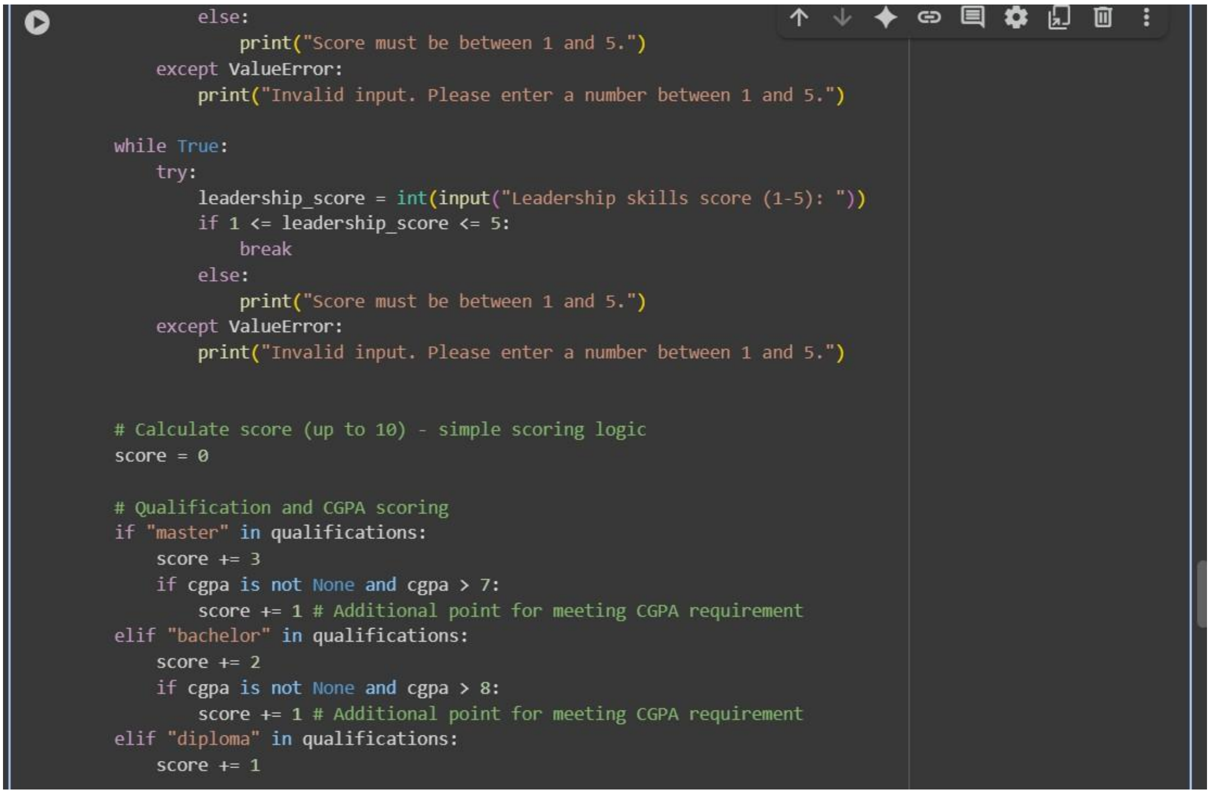
```
def score_applicant():
    """
    Creates a job applicant scoring system based on user input.
    """
    print("Job Applicant Scoring System")
    print("-" * 30)

    # Get applicant details from user
    name = input("Enter applicant's name: ")

    # Get qualifications input
    while True:
        qualifications = input("Enter qualifications (e.g., Bachelor's, Master's): ").lower()
        if qualifications:
            break
        else:
            print("Qualifications cannot be empty.")

    # Get CGPA input if applicable
    cgpa = None
    if "bachelor" in qualifications or "master" in qualifications:
        while True:
            try:
                cgpa = float(input("Enter CGPA: "))
                if 0 <= cgpa <= 10: # Assuming CGPA is on a scale of 0 to 10
                    break
                else:
                    print("CGPA must be between 0 and 10.")
```



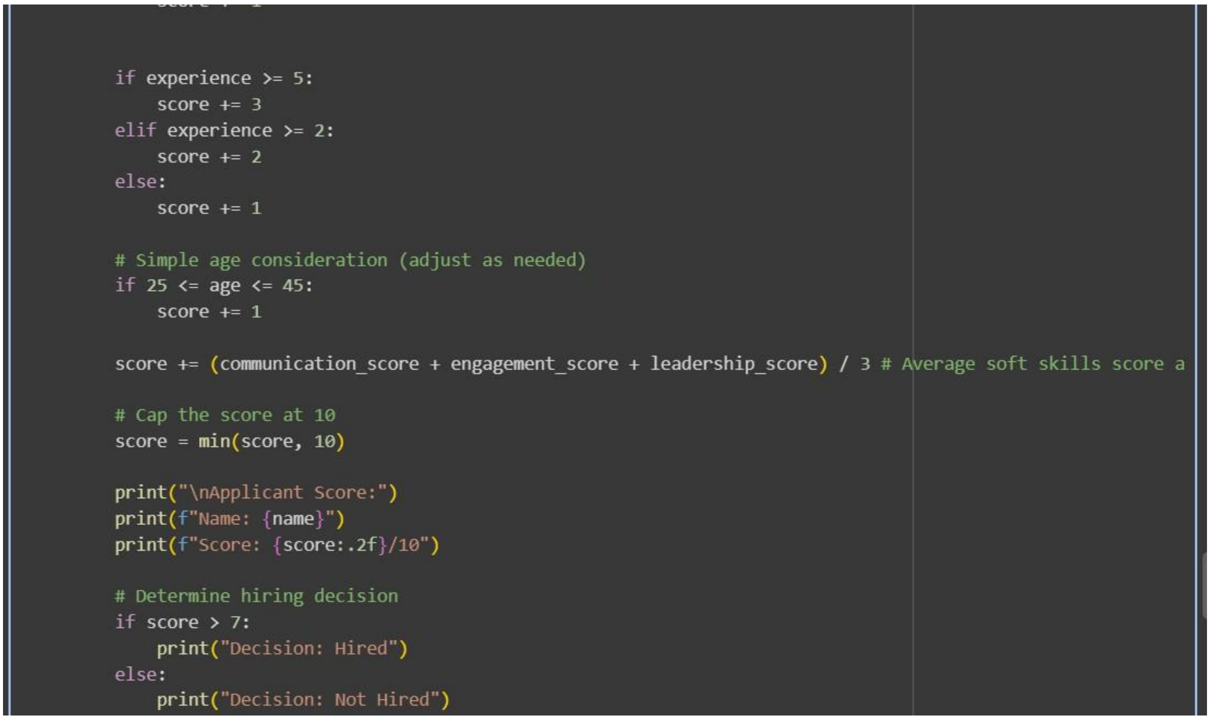
```
                    print("CGPA must be between 0 and 10.")
            except ValueError:
                print("Invalid input. Please enter a number for CGPA.")


    # Get experience input
    while True:
        try:
            experience = int(input("Enter years of experience: "))
            if experience >= 0:
                break
            else:
                print("Experience cannot be negative.")
        except ValueError:
            print("Invalid input. Please enter a number for years of experience.")

    # Get age input
    while True:
        try:
            age = int(input("Enter applicant's age: "))
            if age > 0:
                break
            else:
                print("Age must be a positive number.")
        except ValueError:
            print("Invalid input. Please enter a number for age.")
```



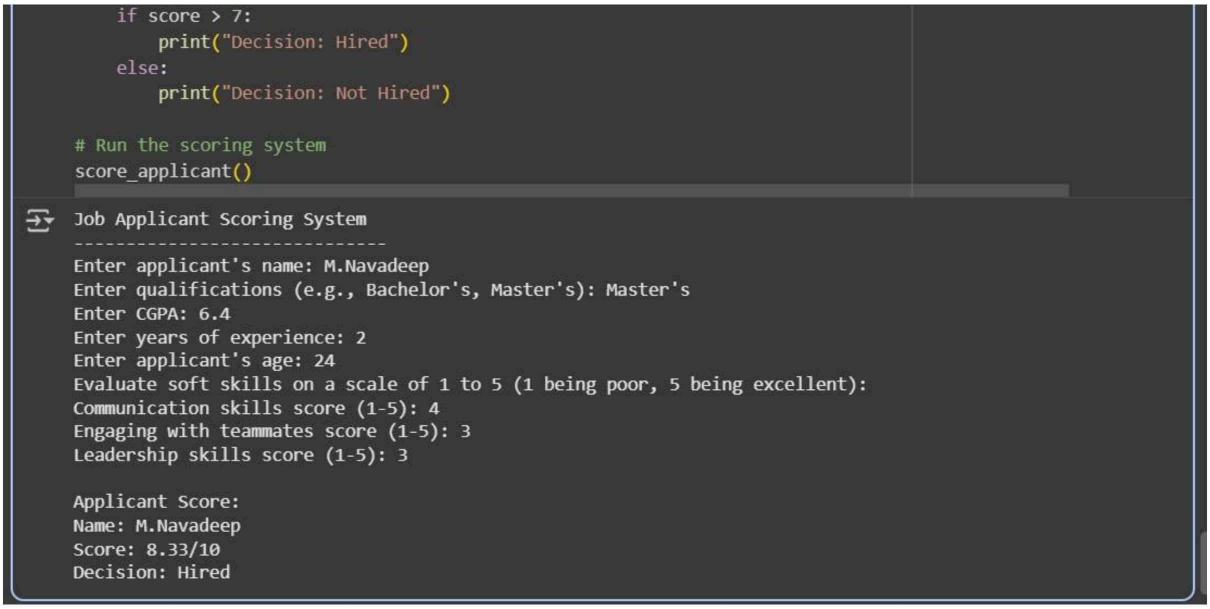
```
    # Get soft skills input
    print("Evaluate soft skills on a scale of 1 to 5 (1 being poor, 5 being excellent):")
    while True:
        try:
            communication_score = int(input("Communication skills score (1-5): "))
            if 1 <= communication_score <= 5:
                break
            else:
                print("Score must be between 1 and 5.")
        except ValueError:
            print("Invalid input. Please enter a number between 1 and 5.")

    while True:
        try:
            engagement_score = int(input("Engaging with teammates score (1-5): "))
            if 1 <= engagement_score <= 5:
                break
            else:
                print("Score must be between 1 and 5.")
        except ValueError:
            print("Invalid input. Please enter a number between 1 and 5.")

    while True:
        try:
            leadership_score = int(input("Leadership skills score (1-5): "))
            if 1 <= leadership_score <= 5:
                break
            else:
```



```
        else:
            print("Score must be between 1 and 5.")
    except ValueError:
        print("Invalid input. Please enter a number between 1 and 5.")

while True:
    try:
        leadership_score = int(input("Leadership skills score (1-5): "))
        if 1 <= leadership_score <= 5:
            break
        else:
            print("Score must be between 1 and 5.")
    except ValueError:
        print("Invalid input. Please enter a number between 1 and 5.")


# Calculate score (up to 10) - simple scoring logic
score = 0

# Qualification and CGPA scoring
if "master" in qualifications:
    score += 3
    if cgpa is not None and cgpa > 7:
        score += 1 # Additional point for meeting CGPA requirement
elif "bachelor" in qualifications:
    score += 2
    if cgpa is not None and cgpa > 8:
        score += 1 # Additional point for meeting CGPA requirement
elif "diploma" in qualifications:
    score += 1
```



```
if experience >= 5:
    score += 3
elif experience >= 2:
    score += 2
else:
    score += 1

# Simple age consideration (adjust as needed)
if 25 <= age <= 45:
    score += 1

score += (communication_score + engagement_score + leadership_score) / 3 # Average soft skills score a

# Cap the score at 10
score = min(score, 10)

print("\nApplicant Score:")
print(f"Name: {name}")
print(f"Score: {score:.2f}/10")

# Determine hiring decision
if score > 7:
    print("Decision: Hired")
else:
    print("Decision: Not Hired")
```



```
    if score > 7:
        print("Decision: Hired")
    else:
        print("Decision: Not Hired")

# Run the scoring system
score_applicant()
```

```
Job Applicant Scoring System
------------------------------
Enter applicant's name: M.Navadeep
Enter qualifications (e.g., Bachelor's, Master's): Master's
Enter CGPA: 6.4
Enter years of experience: 2
Enter applicant's age: 24
Evaluate soft skills on a scale of 1 to 5 (1 being poor, 5 being excellent):
Communication skills score (1-5): 4
Engaging with teammates score (1-5): 3
Leadership skills score (1-5): 3

Applicant Score:
Name: M.Navadeep
Score: 8.33/10
Decision: Hired
```

Task Description 5: (Inclusiveness)

def greet_user(name,gender): if gender.lower() == "male"; title = "Mr." else :

title = "Mrs." return (f"hello,

{title} {name} ! Welcome").

Regenerate code that includes gender-neutral also

Used Prompt :

def greet_user(name,gender): if

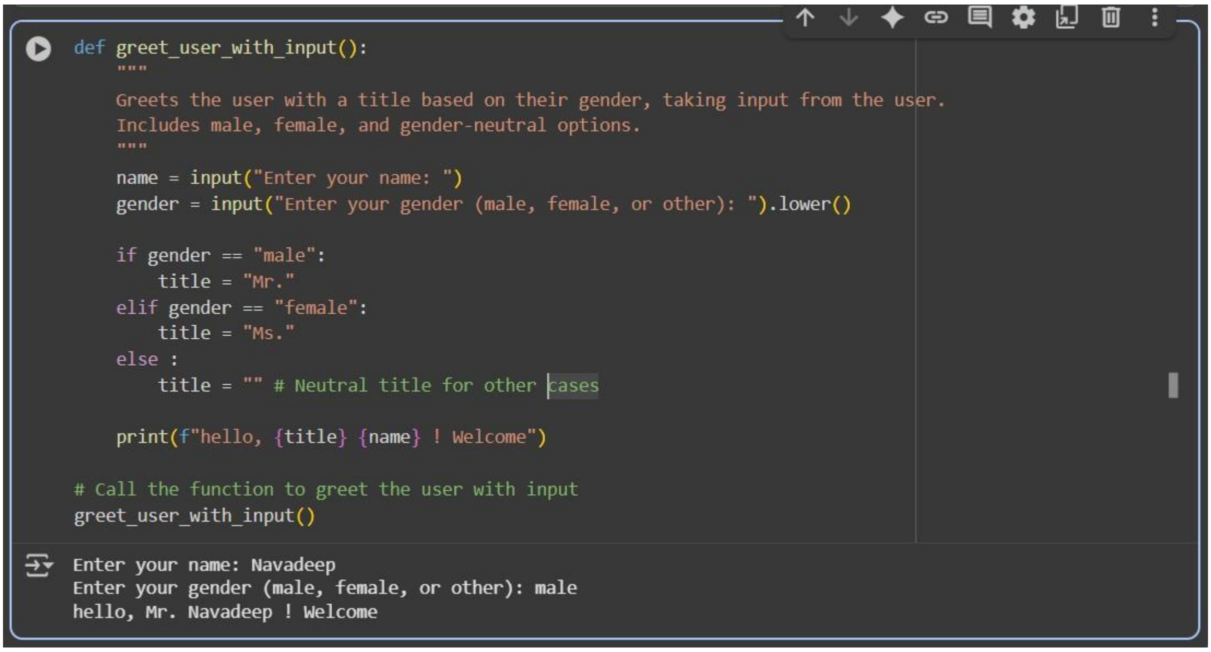
gender.lower() == "male"; title =

"Mr." else :

title = "Mrs." return (f"hello, {title} {name} !

Welcome").

In the code add a gender neutral category and print the output code.



```
def greet_user_with_input():
    """
    Greets the user with a title based on their gender, taking input from the user.
    Includes male, female, and gender-neutral options.
    """
    name = input("Enter your name: ")
    gender = input("Enter your gender (male, female, or other): ").lower()

    if gender == "male":
        title = "Mr."
    elif gender == "female":
        title = "Ms."
    else :
        title = "" # Neutral title for other cases

    print(f"hello, {title} {name} ! Welcome")

# Call the function to greet the user with input
greet_user_with_input()
```

```
Enter your name: Navadeep
Enter your gender (male, female, or other): male
hello, Mr. Navadeep ! Welcome
```