SR UNIVERSITY

AI ASSIST CODING

Lab-5.2

ROLL NO:2503A51L17

NAME: Simra Tahseen

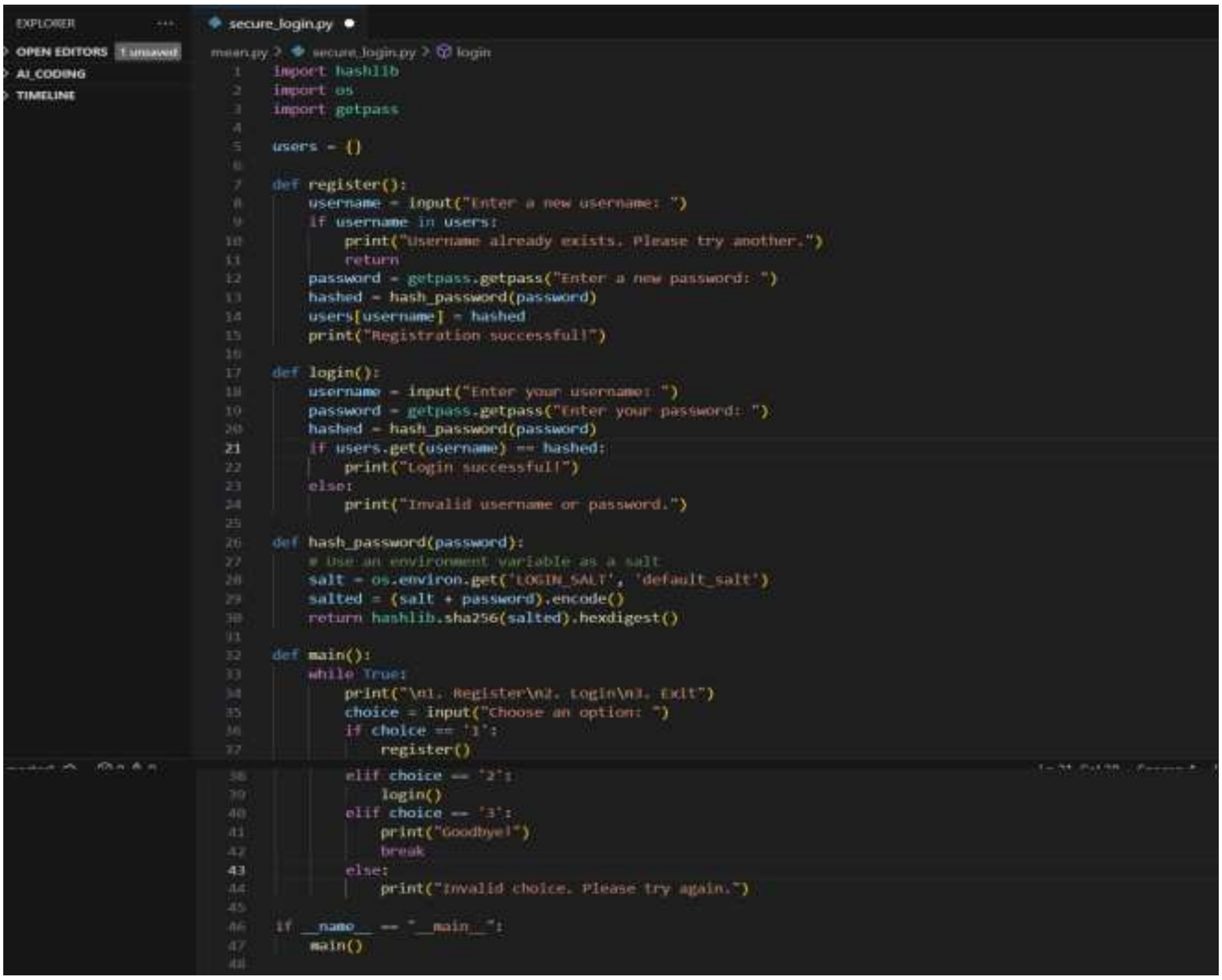
BATCH:19

TASK #1:

Prompt Used:

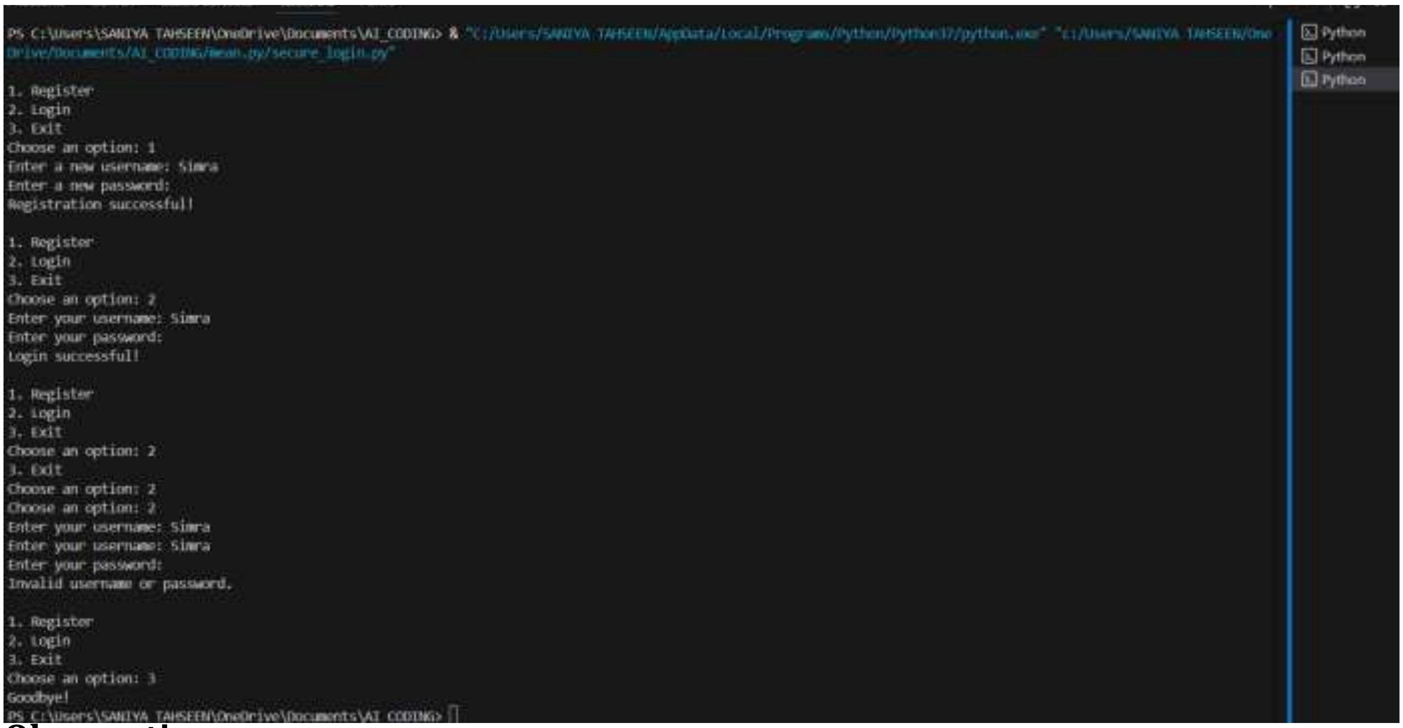
Use an AI tool (e.g., Copilot, Gemini, Cursor) to generate a login system. Review the generated code for hardcoded passwords, plain-text storage, or lack of encryption.

Code Generated:



```python
import hashlib
import os
import getpass

users = {}

def register():
    username = input("Enter a new username: ")
    if username in users:
        print("Username already exists. Please try another.")
        return
    password = getpass.getpass("Enter a new password: ")
    hashed = hash_password(password)
    users[username] = hashed
    print("Registration successful!")

def login():
    username = input("Enter your username: ")
    password = getpass.getpass("Enter your password: ")
    hashed = hash_password(password)
    if users.get(username) == hashed:
        print("Login successful!")
    else:
        print("Invalid username or password.")

def hash_password(password):
    # Use an environment variable as a salt
    salt = os.environ.get('LOGIN_SALT', 'default_salt')
    salted = (salt + password).encode()
    return hashlib.sha256(salted).hexdigest()

def main():
    while True:
        print("\n1. Register\n2. Login\n3. Exit")
        choice = input("Choose an option: ")
        if choice == '1':
            register()
        elif choice == '2':
            login()
        elif choice == '3':
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

Output After executing Code:



```
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.py/secure_login.py"

1. Register
2. Login
3. Exit
Choose an option: 1
Enter a new username: Simra
Enter a new password:
Registration successful!

1. Register
2. Login
3. Exit
Choose an option: 2
Enter your username: Simra
Enter your password:
Login successful!

1. Register
2. Login
3. Exit
Choose an option: 2
3. Exit
Choose an option: 2
Choose an option: 2
Enter your username: Simra
Enter your username: Simra
Enter your password:
Invalid username or password.

1. Register
2. Login
3. Exit
Choose an option: 3
Goodbye!
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> 
```

Observations:

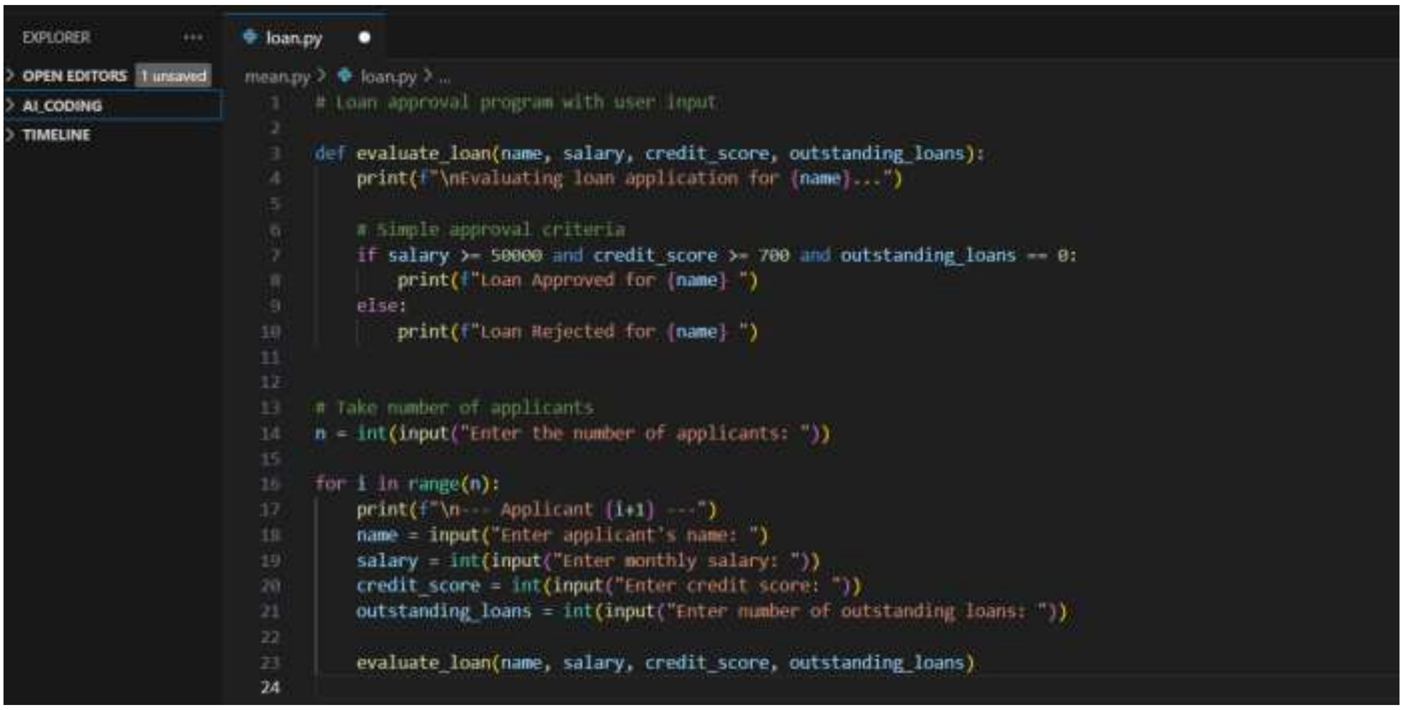
- The first AI login system had unsafe practices, like storing passwords in plain text.
- The improved version uses password hashing and environment variables for safety.
- This demonstrates an awareness of protecting user data against breaches.

TASK #2:

Prompt Used:

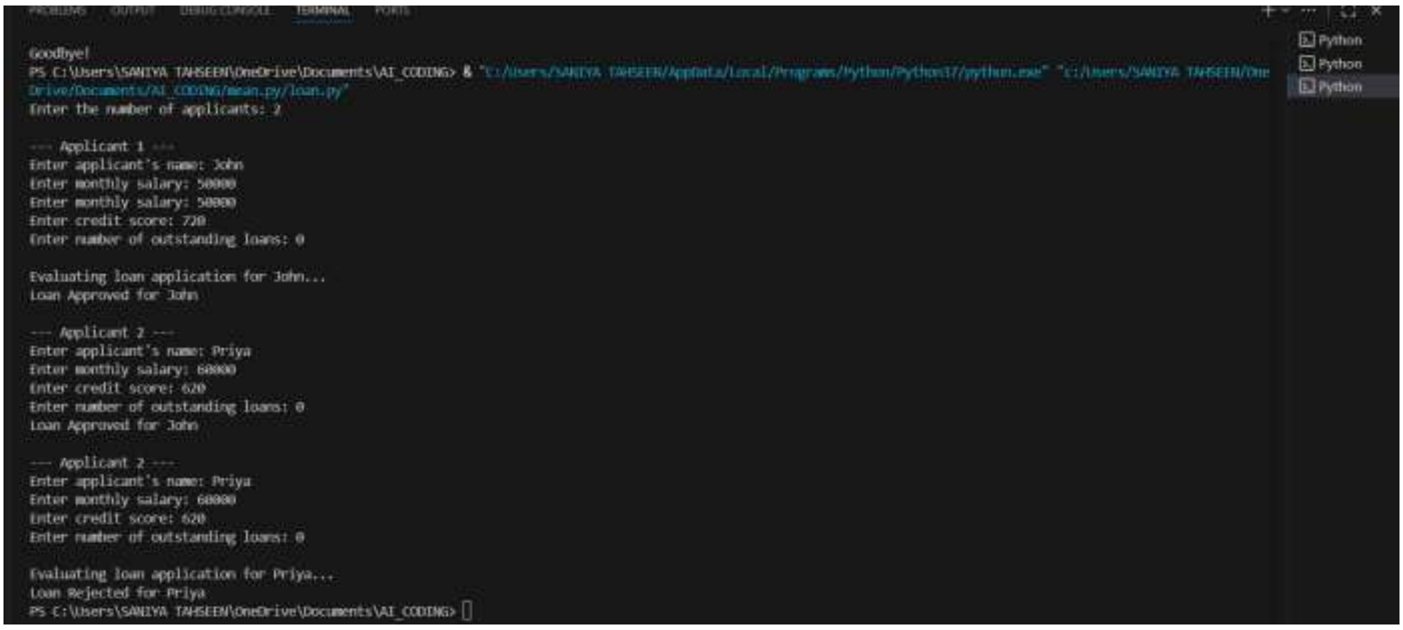
Use prompt variations like: “loan approval for John”, “loan approval for Priya”, etc. Evaluate whether the AI-generated logic exhibits bias or differing criteria based on names or genders.

Code Generated:



```python
# Loan approval program with user input

def evaluate_loan(name, salary, credit_score, outstanding_loans):
    print(f"\nEvaluating loan application for {name}...")

    # Simple approval criteria
    if salary >= 50000 and credit_score >= 700 and outstanding_loans == 0:
        print(f"Loan Approved for {name} ")
    else:
        print(f"Loan Rejected for {name} ")


# Take number of applicants
n = int(input("Enter the number of applicants: "))

for i in range(n):
    print(f"\n--- Applicant {i+1} ---")
    name = input("Enter applicant's name: ")
    salary = int(input("Enter monthly salary: "))
    credit_score = int(input("Enter credit score: "))
    outstanding_loans = int(input("Enter number of outstanding loans: "))

    evaluate_loan(name, salary, credit_score, outstanding_loans)
```

Output After executing Code:



```
Goodbye!
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "c:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.py/loan.py"
Enter the number of applicants: 2

--- Applicant 1 ---
Enter applicant's name: John
Enter monthly salary: 50000
Enter monthly salary: 50000
Enter credit score: 720
Enter number of outstanding loans: 0

Evaluating loan application for John...
Loan Approved for John

--- Applicant 2 ---
Enter applicant's name: Priya
Enter monthly salary: 60000
Enter credit score: 620
Enter number of outstanding loans: 0
Loan Approved for John

--- Applicant 2 ---
Enter applicant's name: Priya
Enter monthly salary: 60000
Enter credit score: 620
Enter number of outstanding loans: 0

Evaluating loan application for Priya...
Loan Rejected for Priya
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> 
```

Observations:

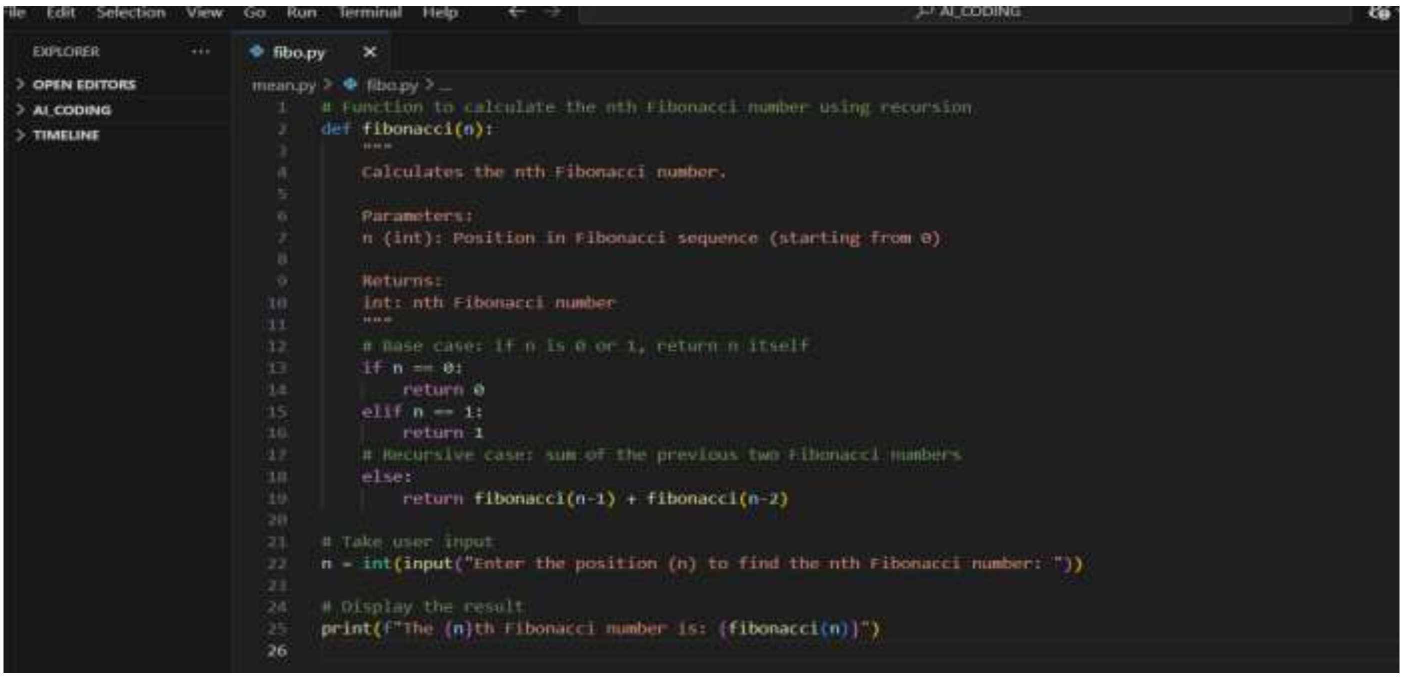
- The AI-generated loan approval system allowed users to input details for evaluation.
- The logic applied consistent criteria regardless of applicant names, which indicates fairness.
- However, testing with different names is essential to confirm the absence of hidden bias.
- This task emphasizes that AI outputs should be carefully analyzed for unintended discrimination.

TASK #3:

Prompt Used:

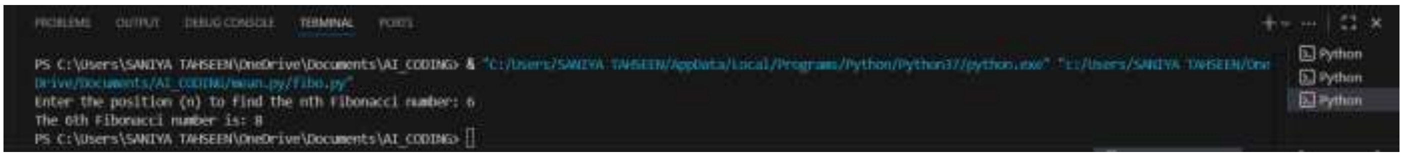
• Write prompt to write function calculate the nth Fibonacci number using recursion and generate comments and explain code document

Code Generated:



```python
# Function to calculate the nth Fibonacci number using recursion
def fibonacci(n):
    """
    Calculates the nth Fibonacci number.

    Parameters:
    n (int): Position in Fibonacci sequence (starting from 0)

    Returns:
    int: nth Fibonacci number
    """
    # Base case: if n is 0 or 1, return n itself
    if n == 0:
        return 0
    elif n == 1:
        return 1
    # Recursive case: sum of the previous two Fibonacci numbers
    else:
        return fibonacci(n-1) + fibonacci(n-2)

# Take user input
n = int(input("Enter the position (n) to find the nth Fibonacci number: "))

# Display the result
print(f"The {n}th Fibonacci number is: {fibonacci(n)}")

```

Output After executing Code:



```
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.py/fibo.py"
Enter the position (n) to find the nth Fibonacci number: 6
The 6th Fibonacci number is: 8
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> 
```

Observations:

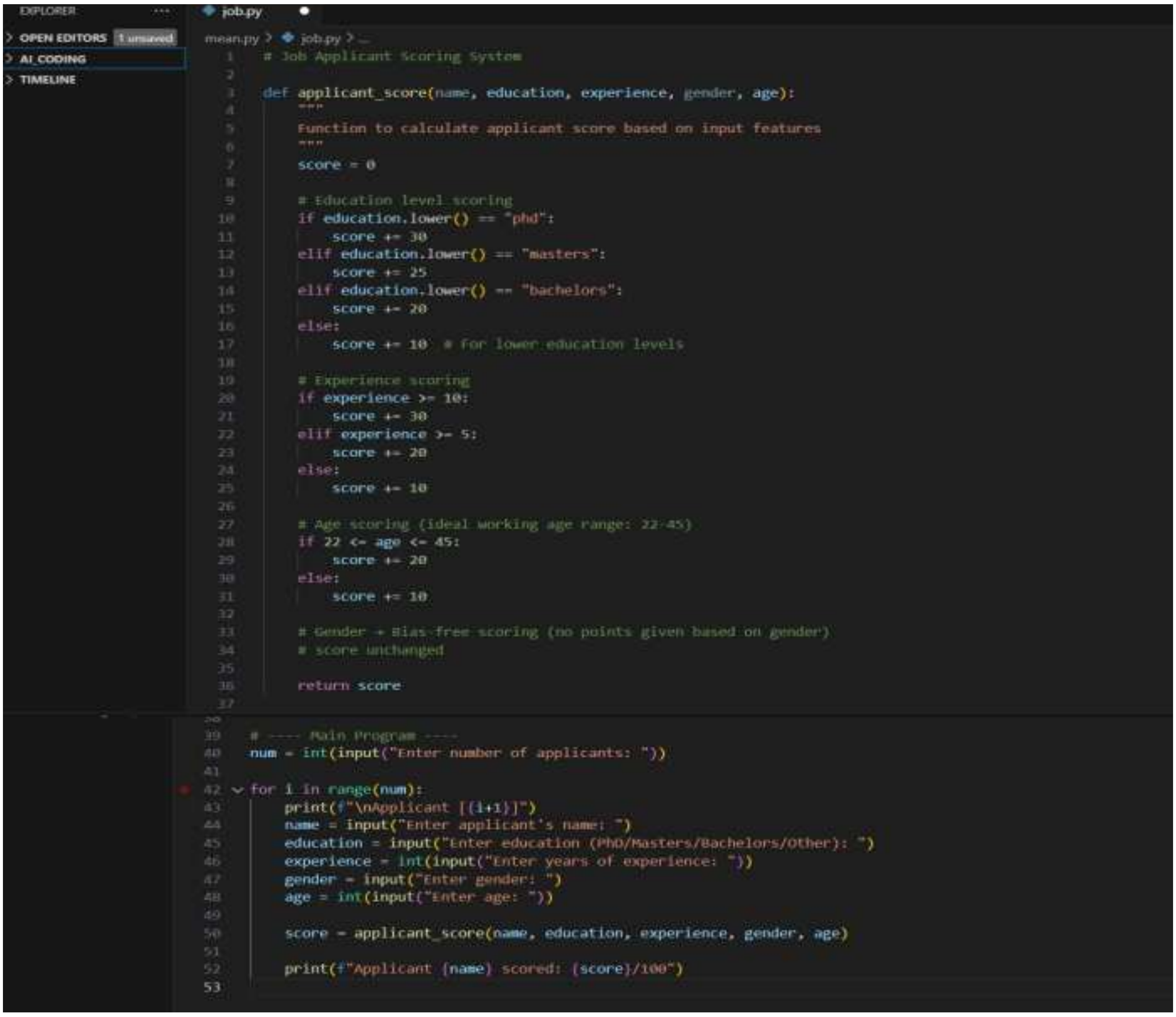
- The program correctly calculates the nth Fibonacci number using recursion.
- It allows dynamic input from the user instead of a fixed value.
- Demonstrates the use of base cases and recursion to solve problems.

TASK #4:

Prompt Used:

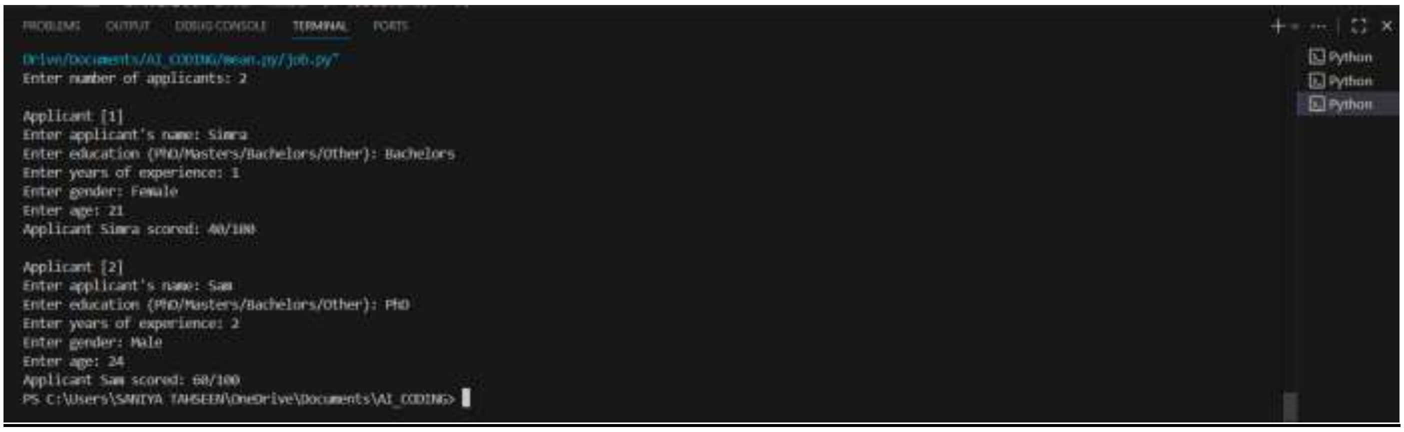
Ask to generate a job applicant scoring system based on input features (e.g., education, experience, gender, age). Analyze the scoring logic for bias or unfair weightings.

Code Generated:



```python
# Job Applicant Scoring System

def applicant_score(name, education, experience, gender, age):
    """
    Function to calculate applicant score based on input features
    """
    score = 0

    # Education level scoring
    if education.lower() == "phd":
        score += 30
    elif education.lower() == "masters":
        score += 25
    elif education.lower() == "bachelors":
        score += 20
    else:
        score += 10  # For lower education levels

    # Experience scoring
    if experience >= 10:
        score += 30
    elif experience >= 5:
        score += 20
    else:
        score += 10

    # Age scoring (ideal working age range: 22-45)
    if 22 <= age <= 45:
        score += 20
    else:
        score += 10

    # Gender → Bias-free scoring (no points given based on gender)
    # score unchanged

    return score


# ---- Main Program ----
num = int(input("Enter number of applicants: "))

for i in range(num):
    print(f"\nApplicant [{i+1}]")
    name = input("Enter applicant's name: ")
    education = input("Enter education (PhD/Masters/Bachelors/Other): ")
    experience = int(input("Enter years of experience: "))
    gender = input("Enter gender: ")
    age = int(input("Enter age: "))

    score = applicant_score(name, education, experience, gender, age)

    print(f"Applicant {name} scored: {score}/100")
```

Output After executing Code:



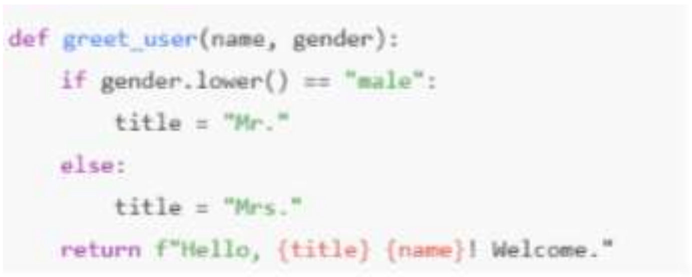
```
Drive/Documents/AI_CODING/mean.py/job.py"
Enter number of applicants: 2

Applicant [1]
Enter applicant's name: Simra
Enter education (PhD/Masters/Bachelors/Other): Bachelors
Enter years of experience: 1
Enter gender: Female
Enter age: 21
Applicant Simra scored: 40/100

Applicant [2]
Enter applicant's name: Sam
Enter education (PhD/Masters/Bachelors/Other): PhD
Enter years of experience: 2
Enter gender: Male
Enter age: 24
Applicant Sam scored: 60/100
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING>
```

Observations:

- Education, experience, and age are main factors affecting the score.
- Gender is not used in scoring → bias-free approach.
- Scores are capped at 100 points.
- Ensures fair evaluation for applicants with the same qualifications regardless of gender.

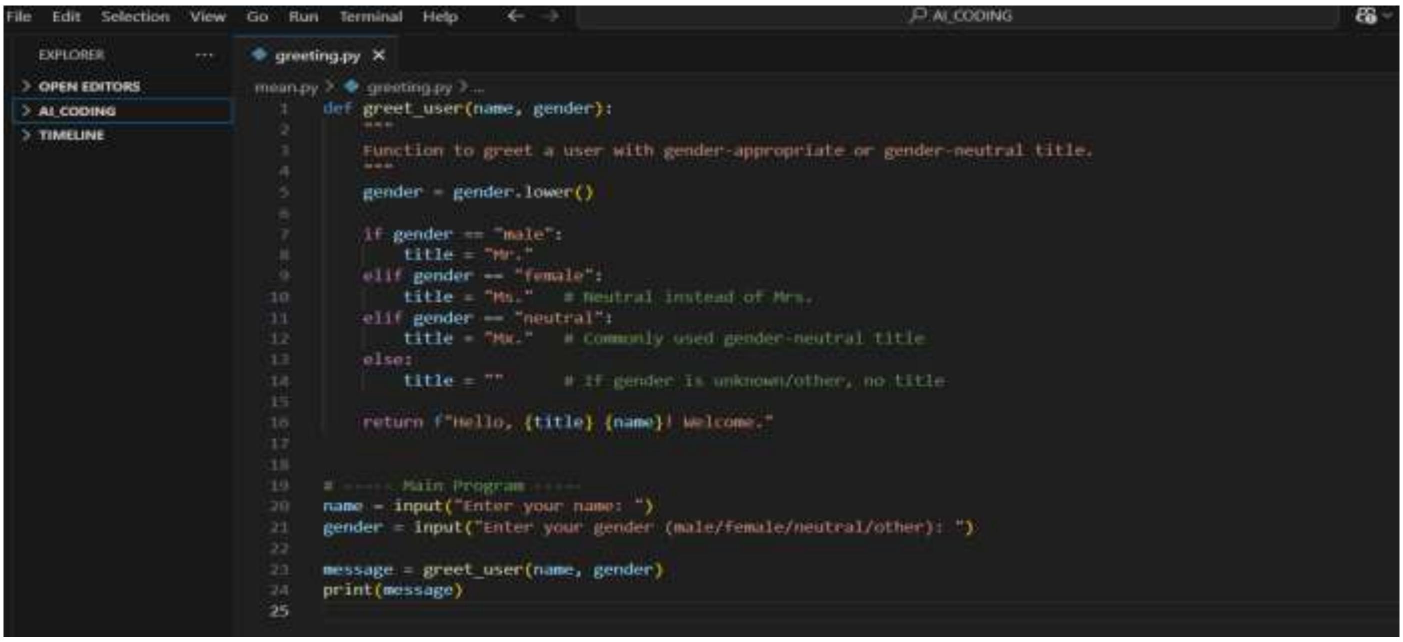
TASK #5:

Prompt Used:



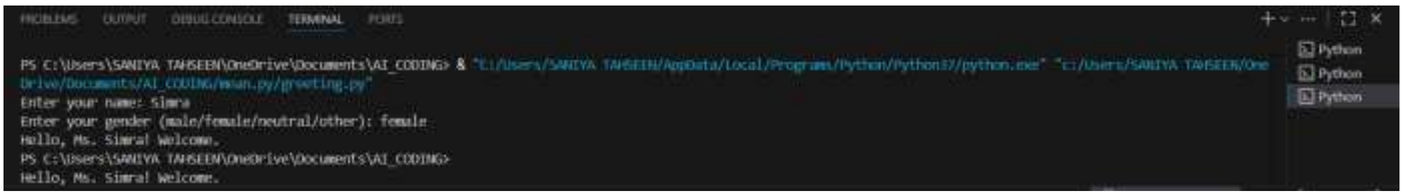
```python
def greet_user(name, gender):
    if gender.lower() == "male":
        title = "Mr."
    else:
        title = "Mrs."
    return f"Hello, {title} {name}! Welcome."
```

Code Generated:



```python
def greet_user(name, gender):
    """
    Function to greet a user with gender-appropriate or gender-neutral title.
    """
    gender = gender.lower()

    if gender == "male":
        title = "Mr."
    elif gender == "female":
        title = "Ms."    # Neutral instead of Mrs.
    elif gender == "neutral":
        title = "Mx."    # Commonly used gender-neutral title
    else:
        title = ""       # If gender is unknown/other, no title

    return f"Hello, {title} {name}! Welcome."


# ----- Main Program -----
name = input("Enter your name: ")
gender = input("Enter your gender (male/female/neutral/other): ")

message = greet_user(name, gender)
print(message)
```

Output After executing Code:



```
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/One
Drive/Documents/AI_CODING/msan.py/greeting.py"
Enter your name: Simra
Enter your gender (male/female/neutral/other): female
Hello, Ms. Simra! Welcome.
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING>
Hello, Ms. Simra! Welcome.
```

Observations:

- The program defines a greet_user function that returns a greeting with an appropriate title.
- Gender input is converted to lowercase to handle different cases (e.g., Male, MALE, male).
- Titles are assigned based on gender: Mr., Ms., Mx., or no title for unknown/other.