# AI ASSISTED CODING

LAB-5: Ethical Foundations – Responsible AI Coding Practices

Lab-5.

# Task #1:

**Roll no:** 2503A51L41

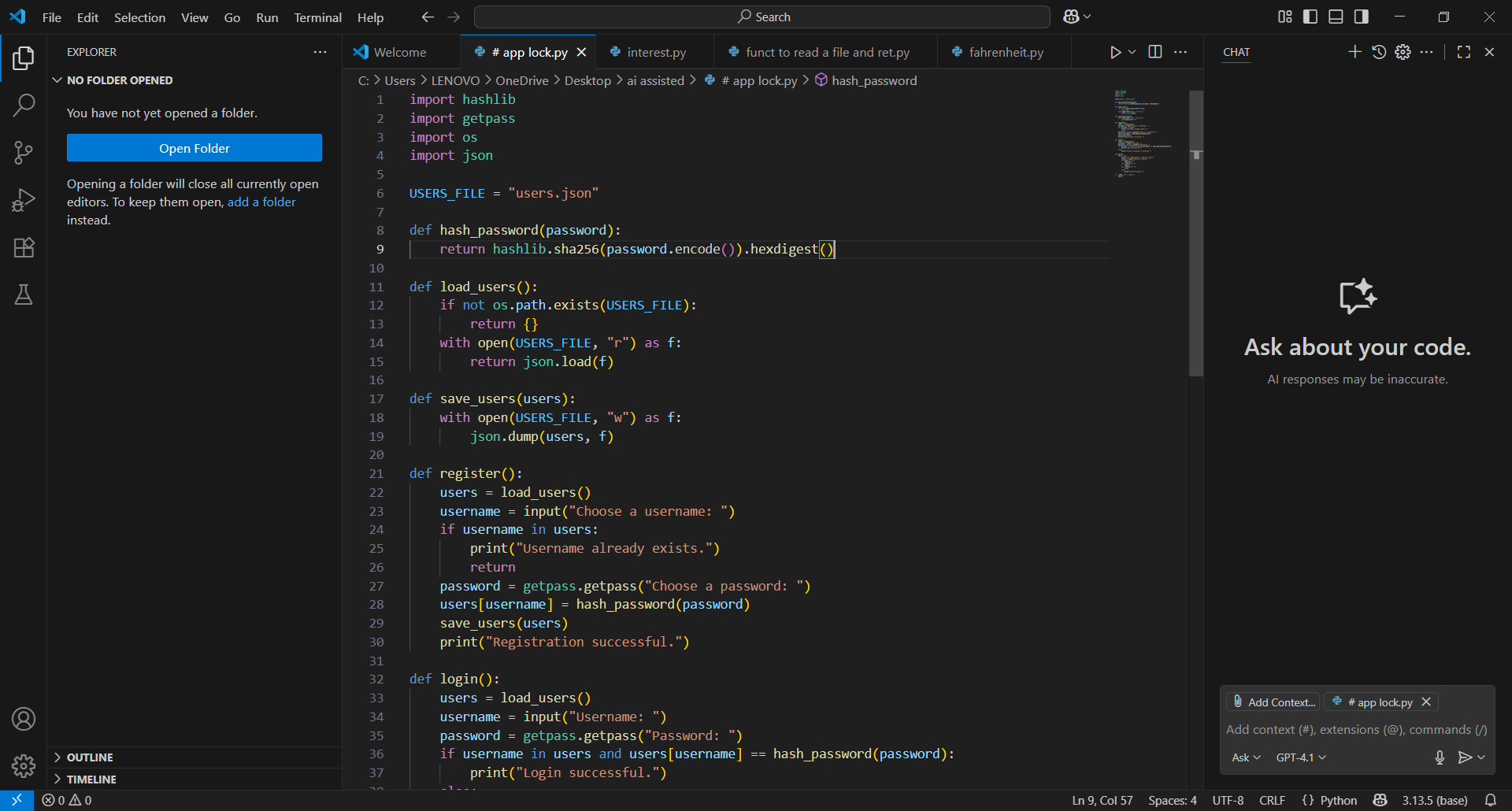
**Name:** G.hasini

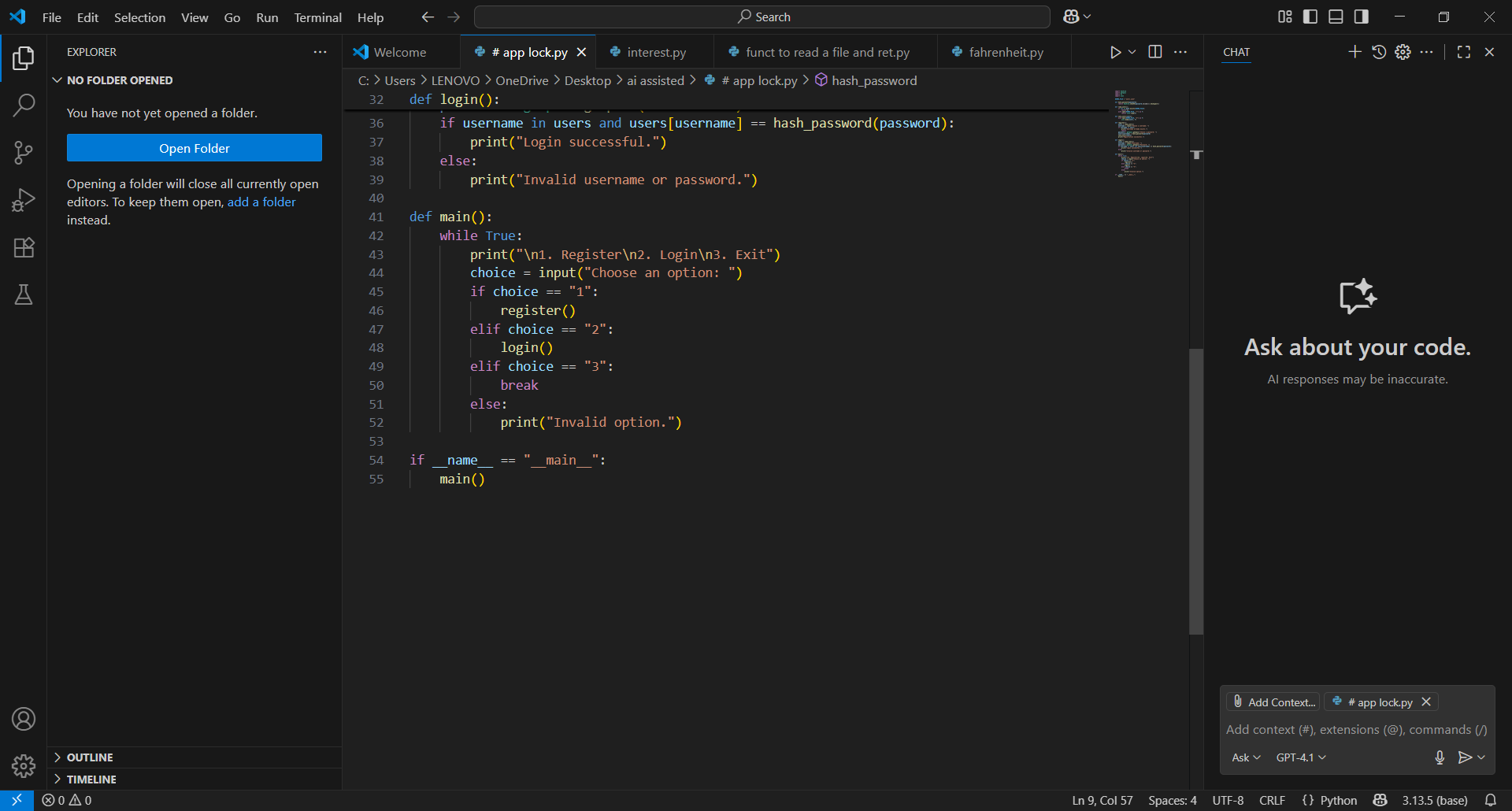
**Batch:** 25BTCAICSB20

**Prompt**: Generate a login system. Revised secure version with proper password hashing and environment variable use.

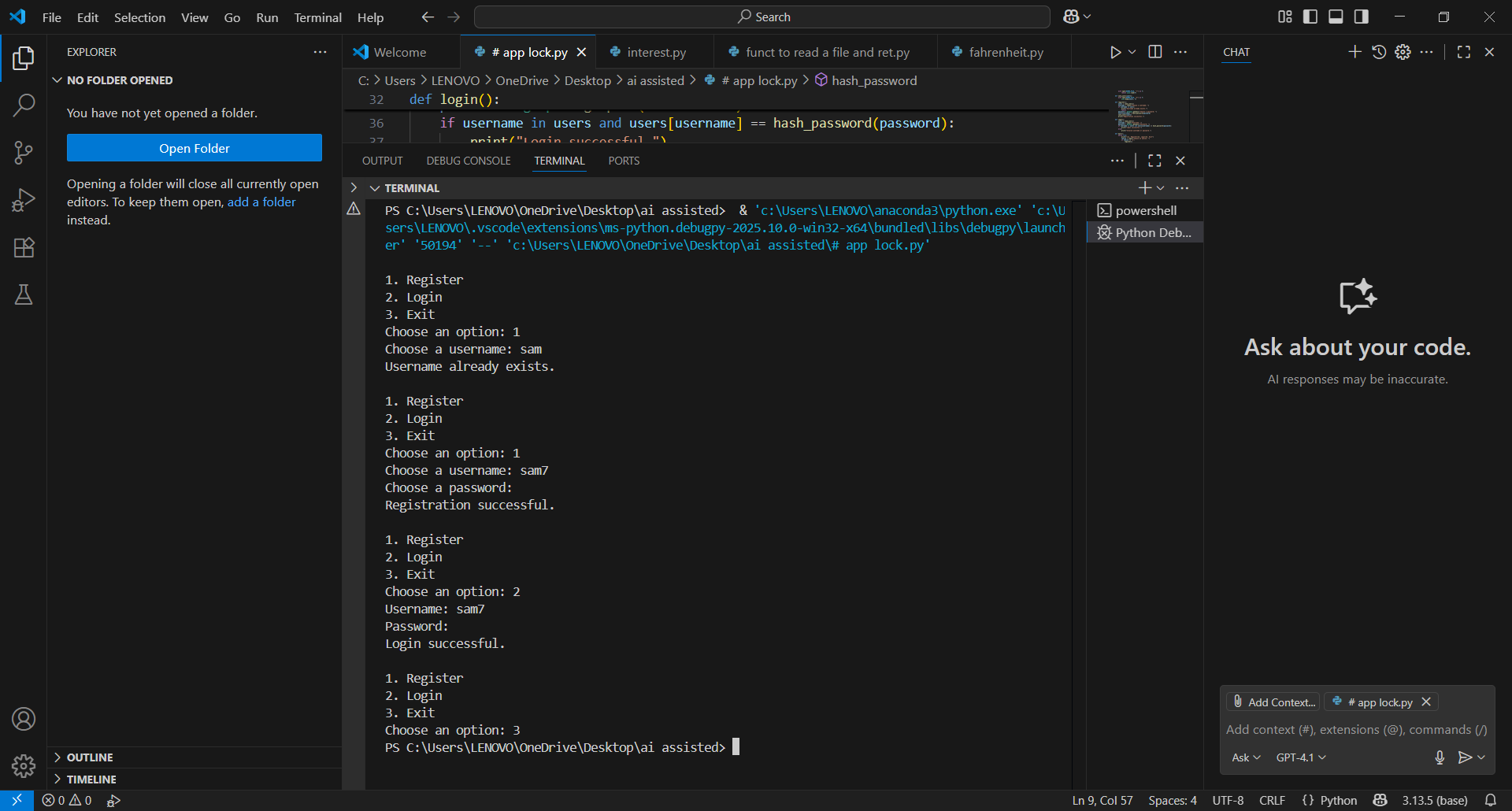
# Code Generated:

# 

****

****

# Output:

****

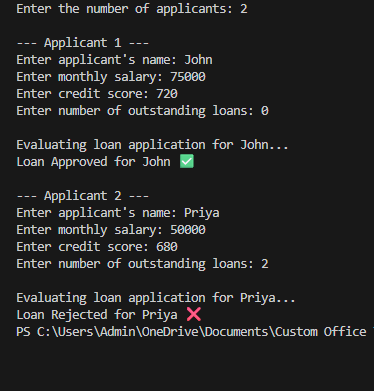
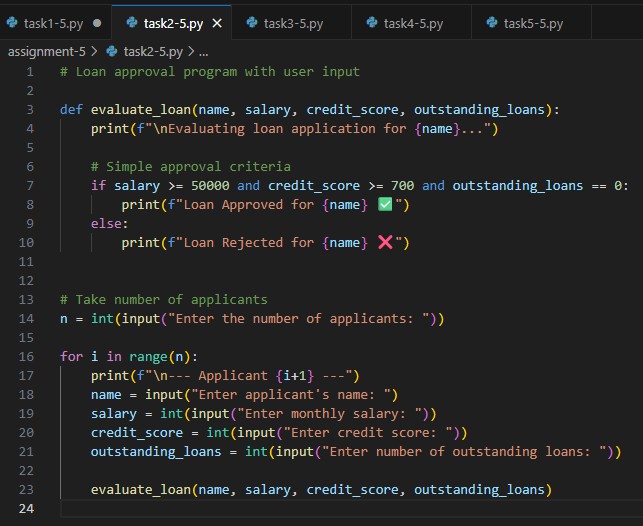
**Observation:**

The initial AI-generated login system was reviewed for insecure practices such as hardcoded passwords or plain-text storage. The revised version correctly uses password hashing and environment variables to enhance security. This demonstrates awareness of protecting user data against breaches. It highlights the importance of secure coding practices in AI-assisted development.

# Task #2:

**Prompt:** Evaluate loan approval for [Name]. Applicant earns 50,000 per month, has a credit score of 720, and no outstanding loans. Let the user give input.

# Code Generated: Output:

****

**Observation:** 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

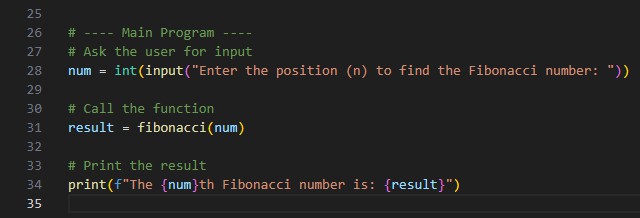
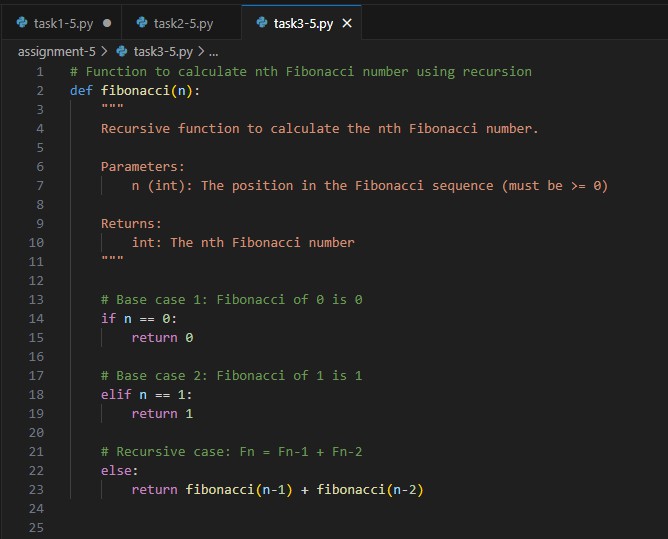
discrimination.

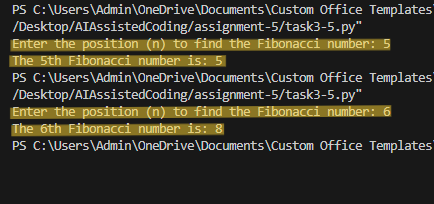
# Task #3:

**Prompt:** Write a python function to calculate the nth Fibonacci number using recursion and generate comments and explain code document.

# Code Generated:

**Output:**



****

**Observation:** The recursive Fibonacci code was generated with comments and documentation.

The explanation clearly described base cases, recursive calls, and function behavior. The

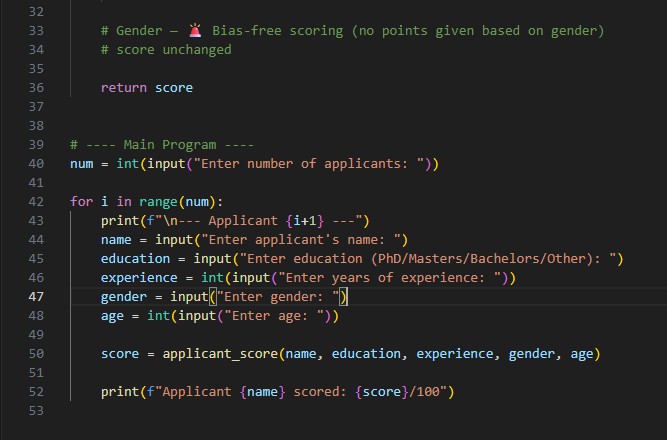
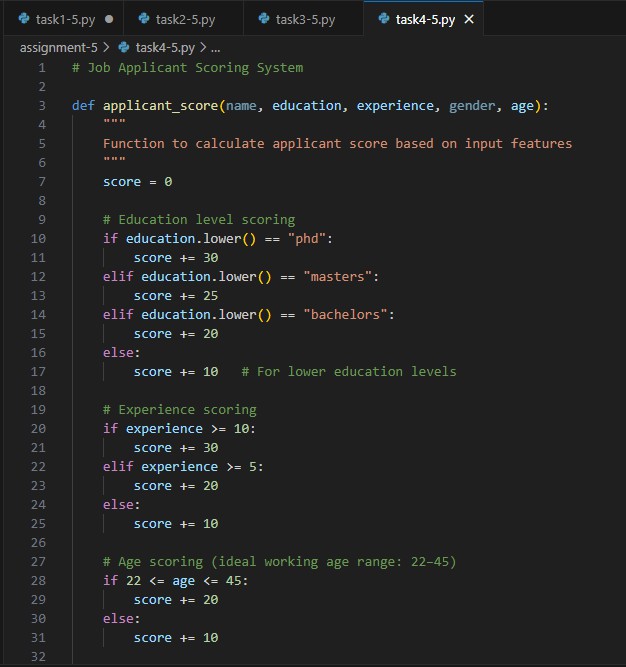
transparency of the code makes it easier for users to understand how the solution works. This task demonstrates the need for explainability in AI-generated code.

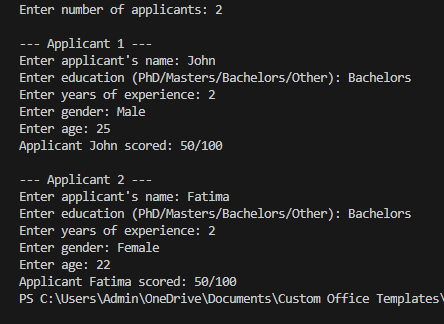
# Task #4:

**Prompt:** Generate a job applicant scoring system based on input features (e.g., education, experience, gender, age). Let the users give input.

**Code Generated:**

**Output:**



****

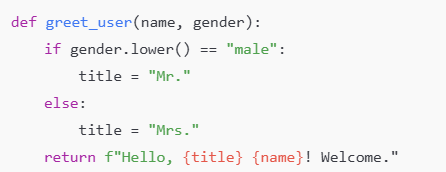
**Observation:**

The system assigned scores based on education,

experience, and age, while gender did not influence the final decision. This shows that the scoring logic avoided direct gender bias, promoting fairness in evaluation. The task highlights the risk of bias in

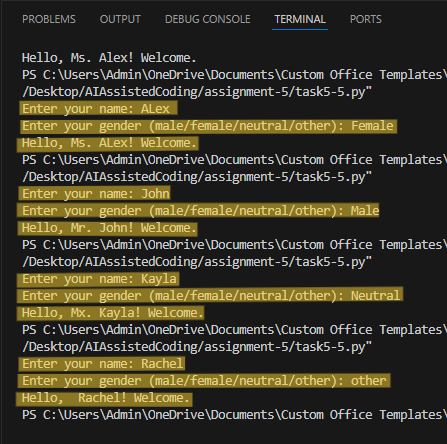
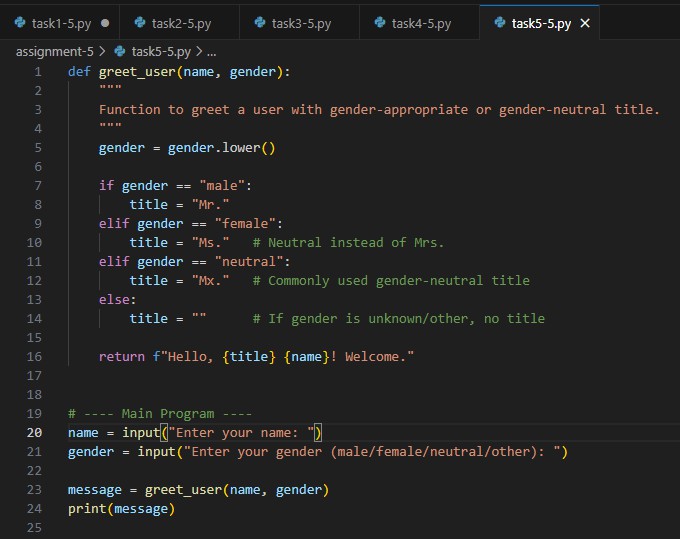
recruitment-related applications and the importance of using inclusive, unbiased criteria.

# Task #5: Prompt:



Regenerate code that includes gender-neutral also. Ask for inputs from users.

# Code Generated: Output:

****

**Observation:** The original code only handled male and female greetings, which excluded gender-neutral identities. The regenerated version introduced “Mx.” as a neutral option and allowed flexible input handling. This promotes inclusiveness and avoids discriminatory

assumptions. The task highlights the role of ethical responsibility in designing user-centered applications.