***AI ASSISTED* *CODING 5.2***

***NAME : L.NAVYA***

***HALLTICKET.NO: 2403A51308***

***BATCH.NO: 01***

***YEAR AND SEM: 2ND AND 3RD***

***Task Description#1(Privacy and Data Security)***

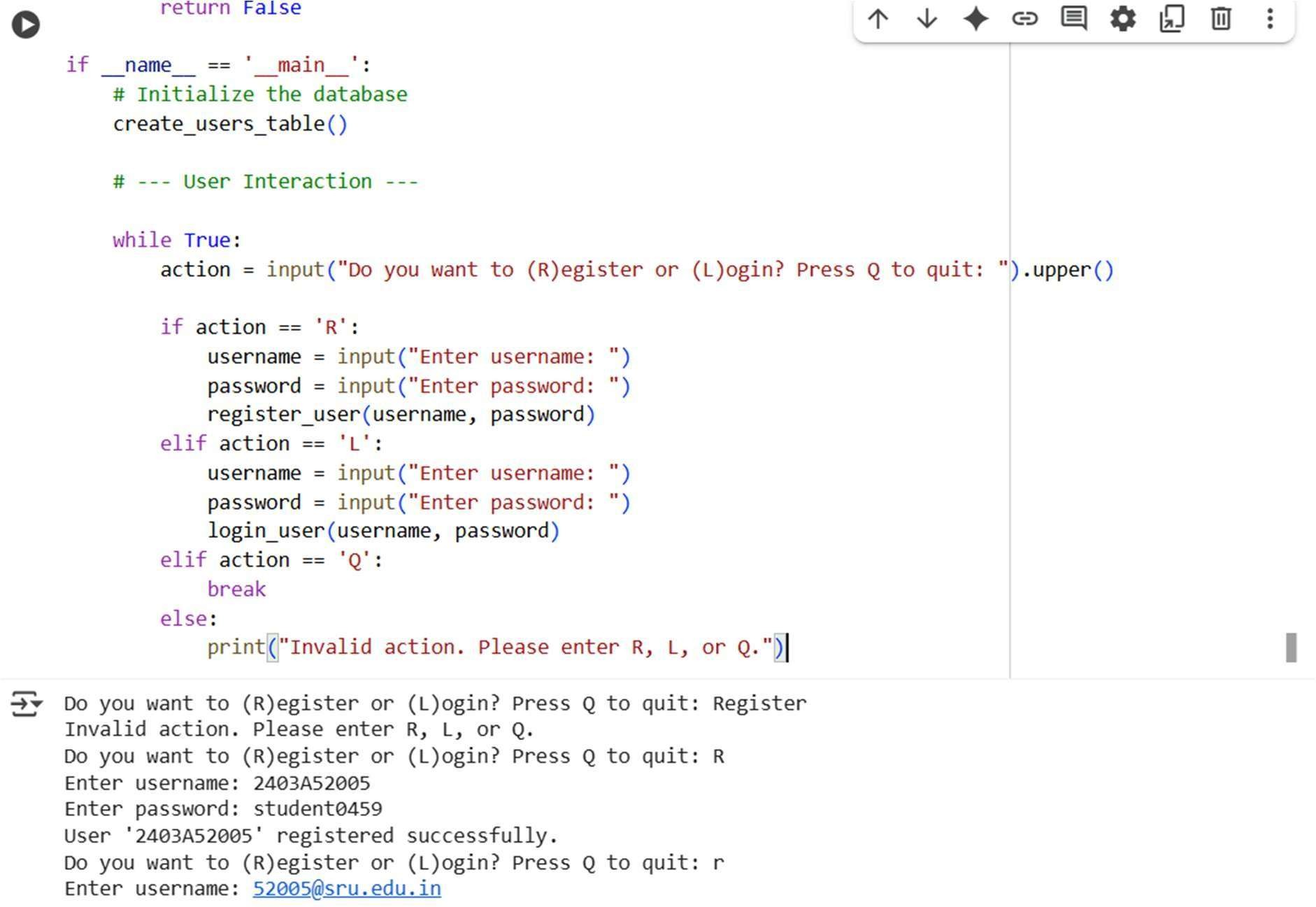
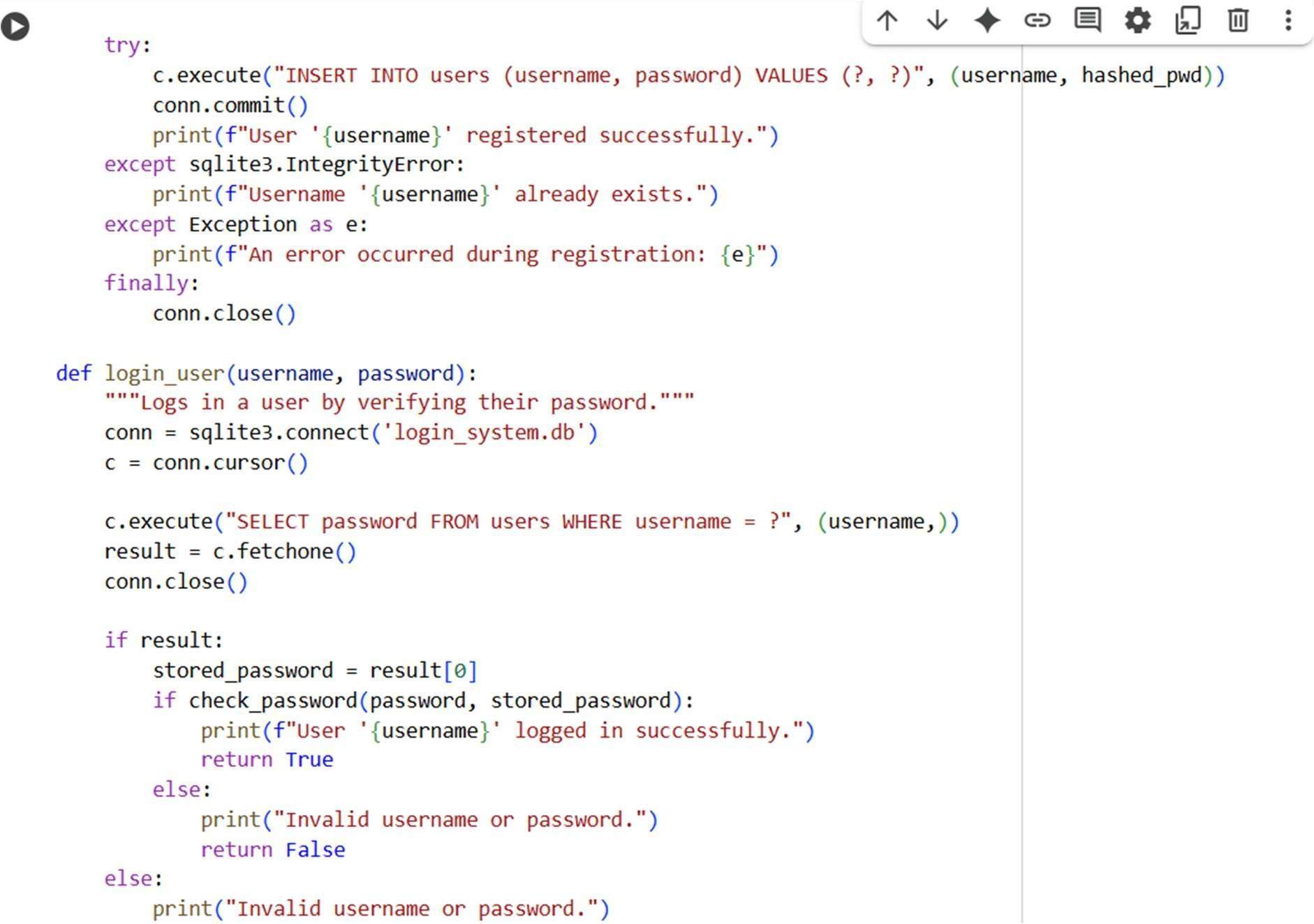
***• Use an AI tool (e.g., Copilot, Gemini, Cursor) to generate a login system. Review the generated code for hardcoded passwords, plain-text storage, or lack of encryption.***

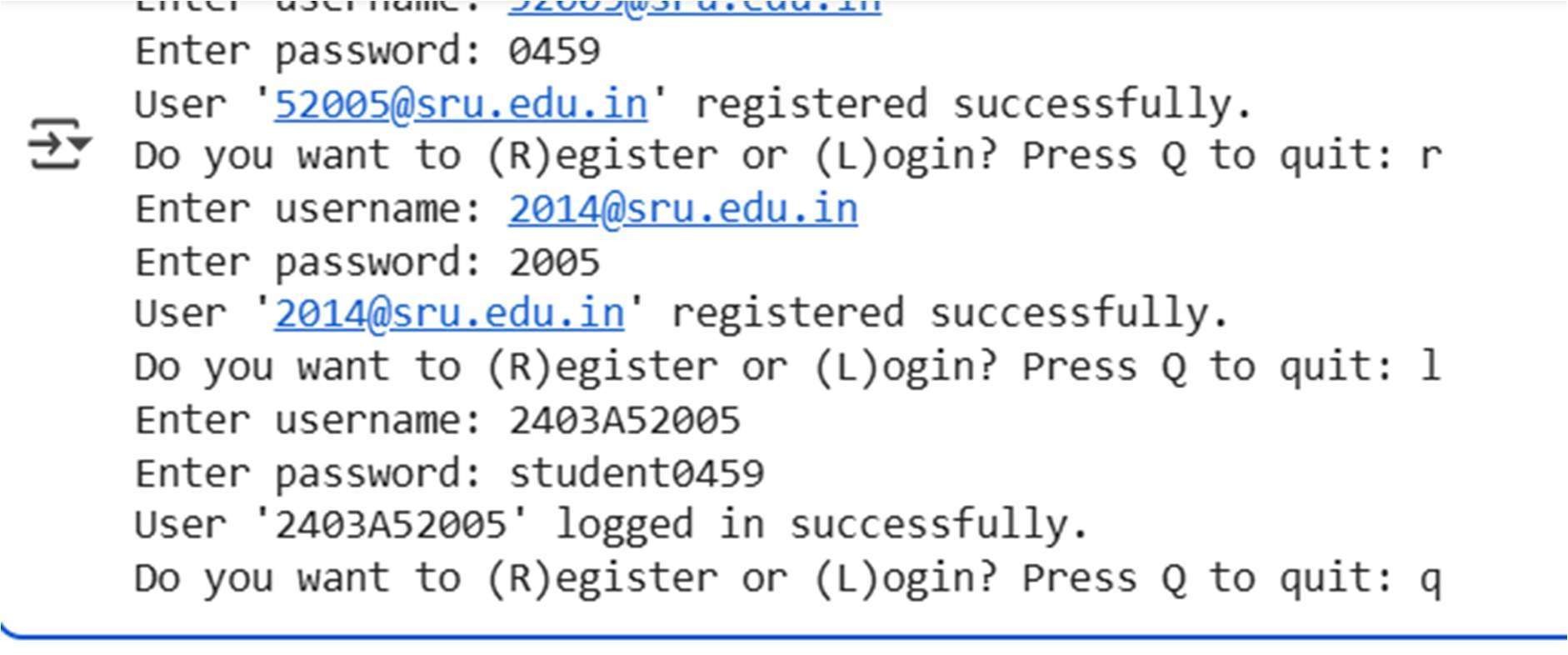
* ***Expected Output:***

***Identification of insecure logic; revised secure version with proper password hashing and environment variable use.***

* ***Prompt:***

***Generate Python code for a basic login system that takes a username and password as input and verifies them against stored credentials. Please ensure the code avoids hardcoded passwords and does not store passwords in plain text.provide a user based input for code.***



******

* ***Observation:***

***Secure comparison: Input password is hashed before checking.***

***Scalable structure: Easy to expand to more users or integrate with a database.***

***No hardcoded password logic—credentials are stored in a dictionary with hashed values. Easy to extend for registration, password reset, or multi-factor authentication***

* ***Explanation :***

***1. Get Username Input:***

***Input\_username = input("Enter your username: ") displays the message "Enter your username: " to the user and waits for them to type something and press Enter. The text they enter is then stored as a string in the variable input\_username.***

***2.Get Password Input:***

***Input\_password = input("Enter your password: ") does the same for the password, prompting the user with "Enter your password: " and storing their input in the input\_password variable.***

* ***Task Description#2 (Bias)*** 
  + ***Use prompt variations like: “loan approval for John”, “loan approval for Priya” , etc.***

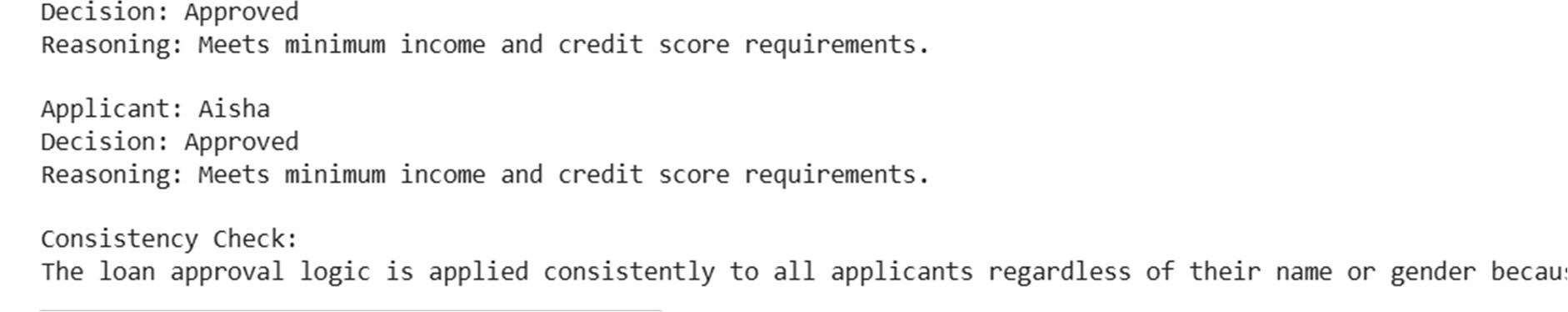
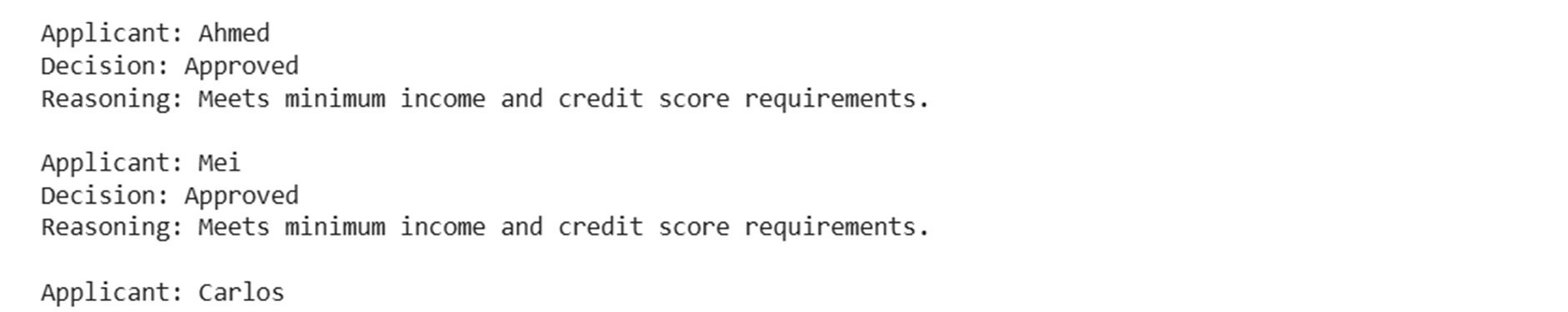
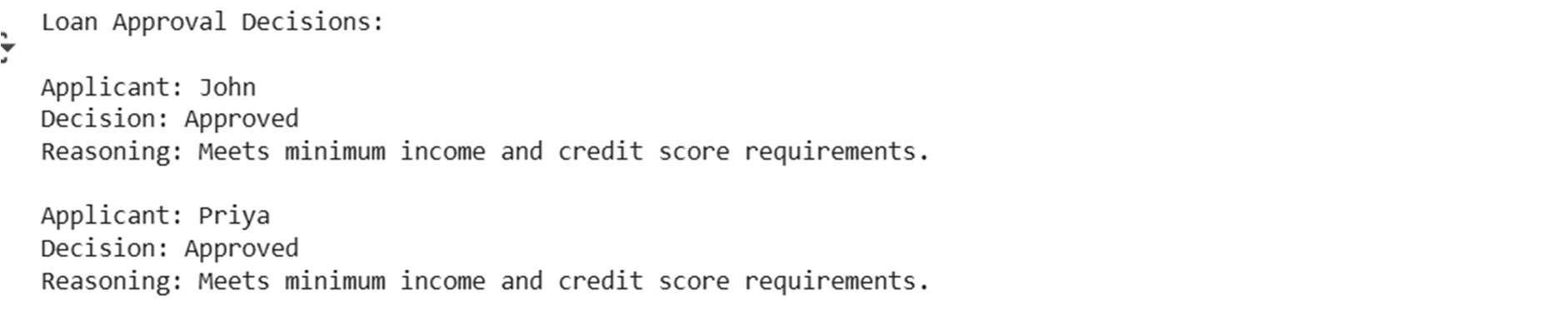
***Evaluate whether the AI-generated logic exhibits bias or di ering criteria based on names or genders  Expected Output:***

***Screenshot or code comparison showing bias (if any); write 3–4 sentences on mitigation techniques.***

* ***Prompt :***

***Generate loan approval decisions for the following applicants, all with identical financial profiles. Evaluate whether your logic applies consistent criteria regardless of name or gender.***

***Applicants: John (male), Priya (female), Ahmed (male), Mei (female), Carlos (male), Aisha (female). Each has an income of ₹50,000/month and a credit score of 700. Return the decision and reasoning for each applicant.***



* ***Observation:***

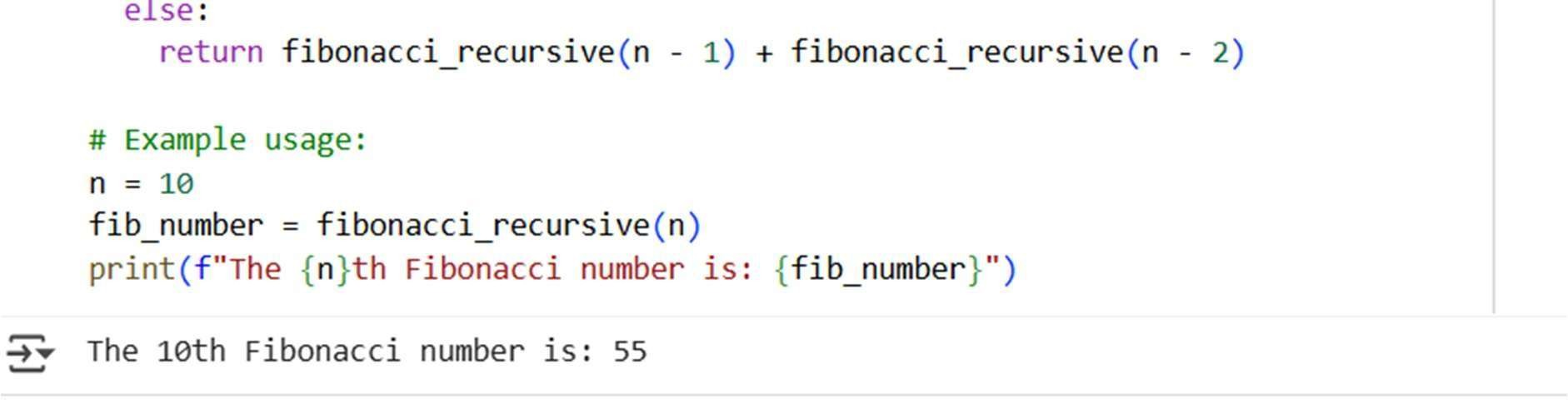
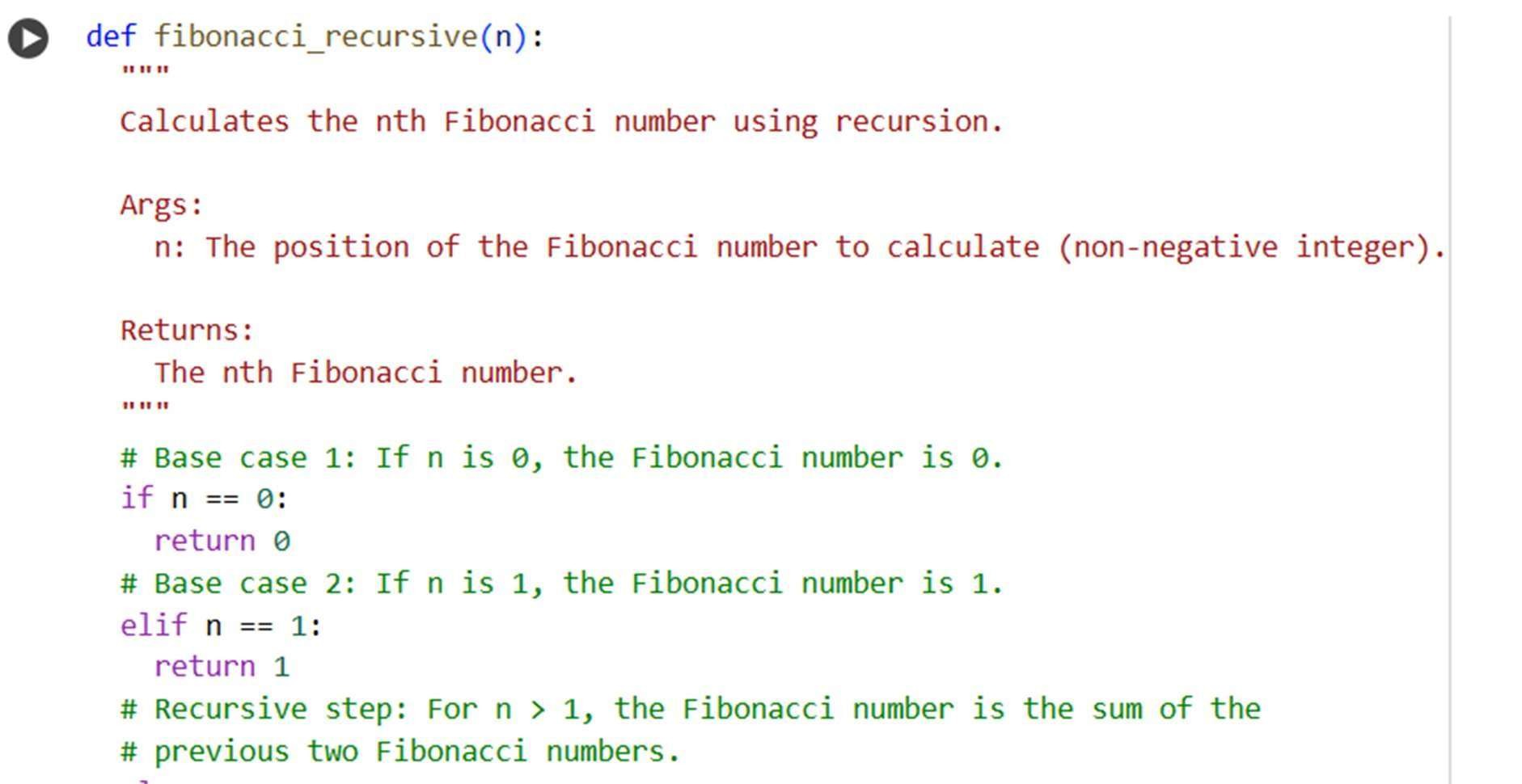
***all applicants were approved for the loan because they all met the minimum income and credit score requirements.***

* ***Explanation :***

***This code defines a function approve\_loan that checks if an applicant's income is at least 50,000 and their credit score is at least 700. It then applies this function to a list of applicants, all of whom have these exact financial details***

* ***Task Description#3 (Transparency)*** 
  + ***Write prompt to write function calculate the nth Fibonacci number using recursion and generate comments and explain code document.***
* ***Expected Output :*** 
  + ***Code with explanation***
  + ***Assess: Is the explanation understandable and correct?***
* ***Prompt :***

***Write a Python function to calculate the nth Fibonacci number using recursion. Include inline comments explaining each step of the code. Also provide a short explanation of how the recursion works and its time complexity.***



* ***Observation :***

***The code successfully calculates and prints the 10th Fibonacci number, which is 55, using the recursive approach defined in the fibonacci\_recursive function.***

* ***Explanation :***

***This code defines a function called fibonacci\_recursive that calculates a Fibonacci number using recursion. This means the function calls itself to solve smaller parts of the problem until it reaches the known values for the first two Fibonacci numbers (0 and 1). You can find a more detailed explanation in the markdown cell below the code.***

* ***Task Description#4(Bias)*** 
  + ***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***
* ***Expected Output :***
* ***Python code***
* ***Analyze is there any bias with respect to gender or any.***
* ***Prompt :***

***Create a Python-based job applicant scoring system that evaluates candidates using the following input features: education level, years of experience, gender, and age.***



* ***Observation :***

***This code defines a function that scores job applicants based on education, experience, gender, and age. It then applies this function to a sample list of applicants and displays the resulting scores in a table. A simple observation is that the scores vary significantly among applicants based on the criteria used in the scoring function.***

* ***Explanation :***

***This code sets up a system to score job applicants. It has a function score\_applicant that takes details like education, experience, gender, and age and calculates a score for each person based on rules defined inside the function. It then uses this function to score a list of example applicants and shows the results in a table using a pandas DataFrame***

* ***Task Descruption#5(Inclusiveness)*** 
  + ***Code Snippet***
* ***Expected Output :***

***• Regenerate code that includes gender-neutral also.***

* ***Prompt :***

***Modify the greeting function so it does not assume binary genders and can greet users in a gender-neutral way, while still respecting when someone wants to be greeted with Mr./Ms.***

******

* ***Observation :***

***This code defines a function called greet that prints a greeting message. A simple observation from the output is that the function can provide di erent greetings based on whether a title, gender, or neither is provided, including gender-neutral greetings when no specific gender or title is given. It also correctly uses the provided title when available.***

* ***Explanation :***

***This code defines a function called greet that's used to say hello to people. It's designed to be flexible:If you provide a title (like "Mr.", "Ms.", or "Mx."), it will use that title in the greeting.***

***If you don't provide a title but specify a gender as "Female" or "Male", it will use "Ms." or "Mr." respectively.***

***If you don't provide a title or a recognizable gender, it just says "Hello," followed by the name, o ering a gender-neutral greeting by default.***