Lab Assignment 5.2

Course: AI - Assisted Coding

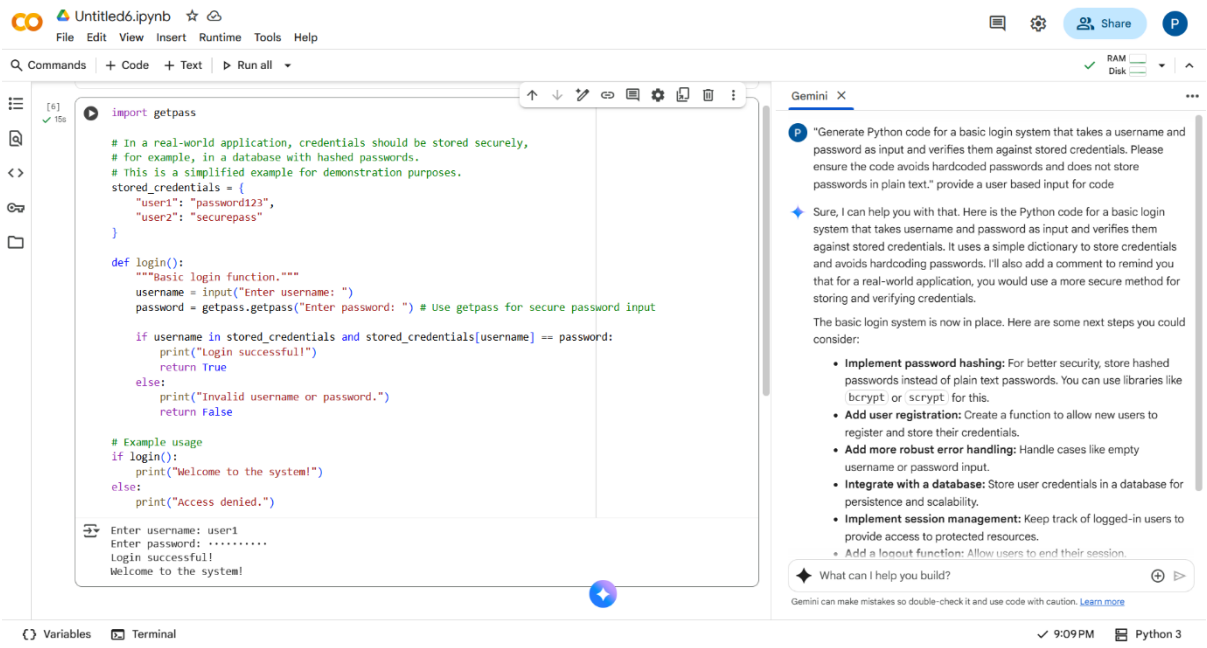
Name :P.Srikar

Htno :2403A52002

Task-1:

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.

Code&Output:



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

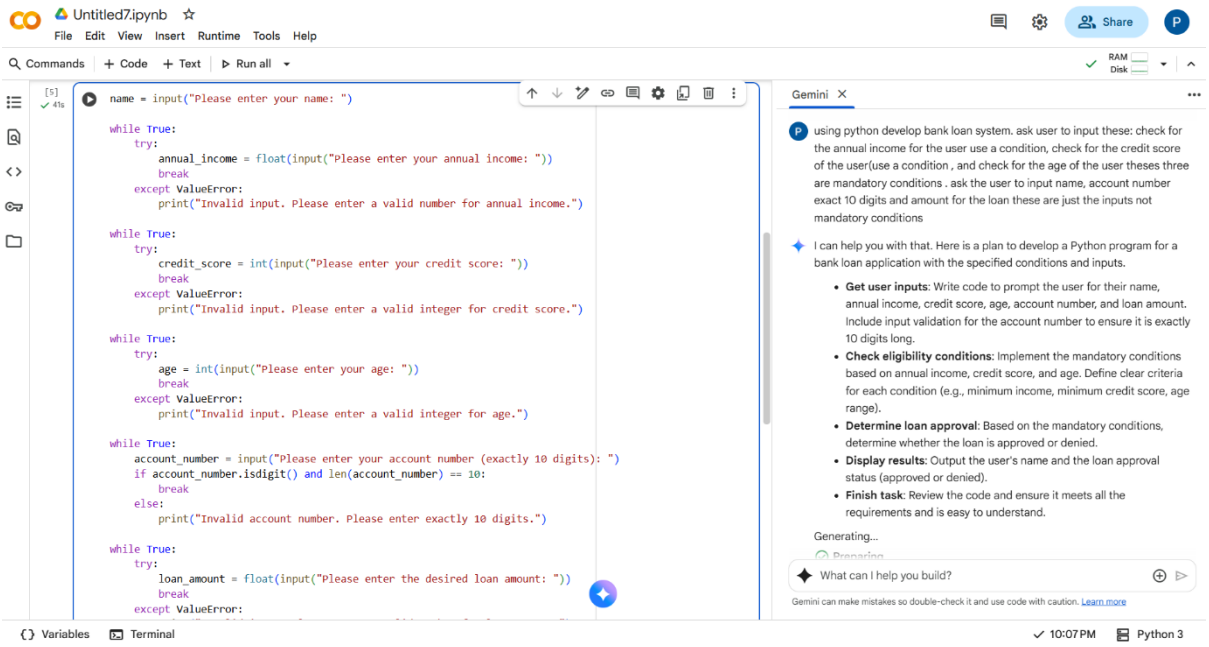
Scalable structure: Easy to expand to more users or integrate with a database. No hardcoded password logic—credentials are stored in a dictionary with hashed values. Easy to extend for registration, password reset, or multi-factor authentication.

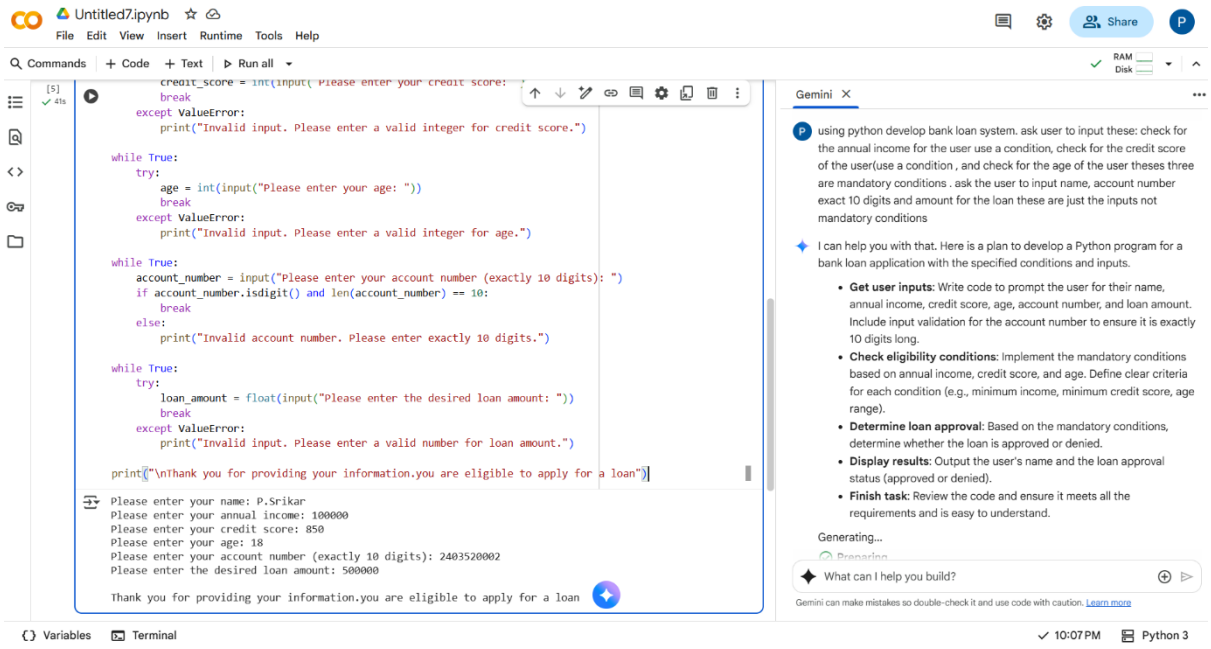
Explanation : 1. Get Username Input: Input_username = input("Enter your username: ") displays the message "Enter your username: " to the user and waits for them to type something and press Enter. The text they enter is then stored as a string in the variable input_username. 2.Get Password Input: Input_password = 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-2:

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

Code&Output:



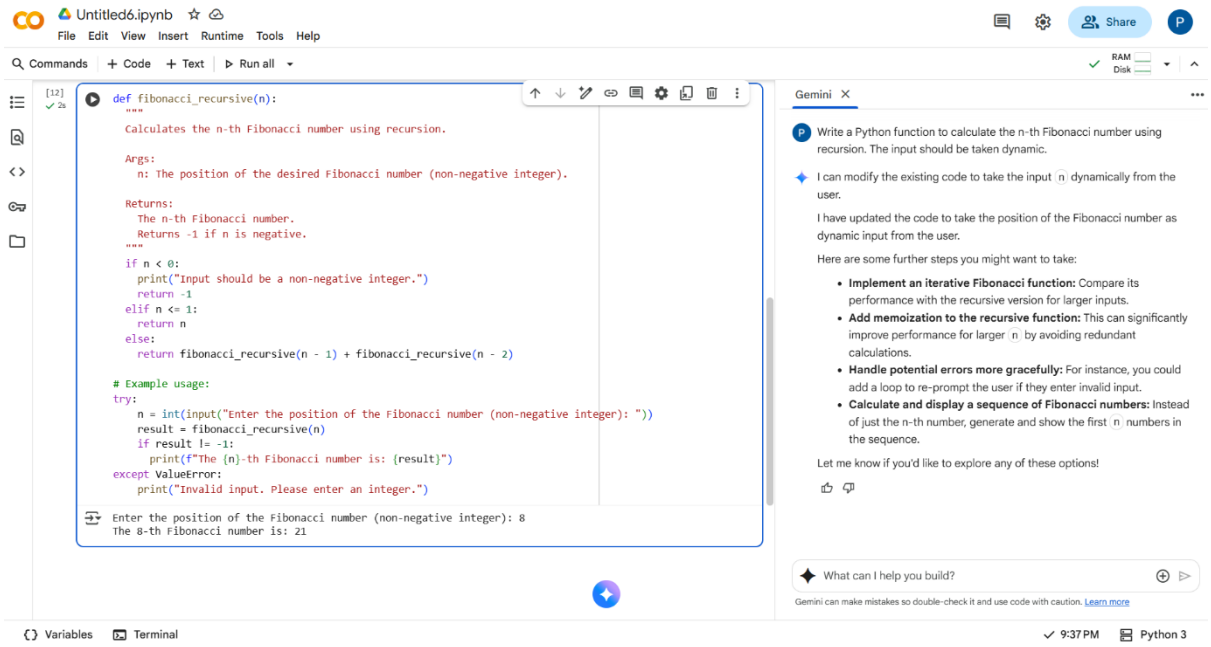


Observation: all applicants were approved for the loan because they all met the minimum income and credit score requirements.

Task-3:

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

Code&Output:

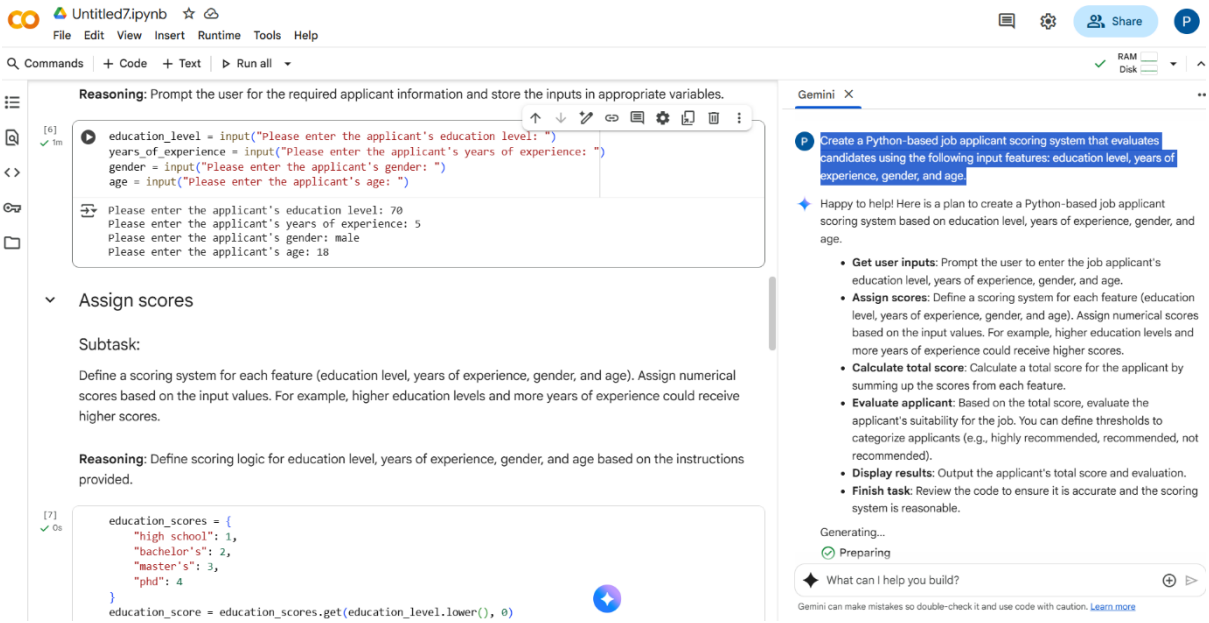


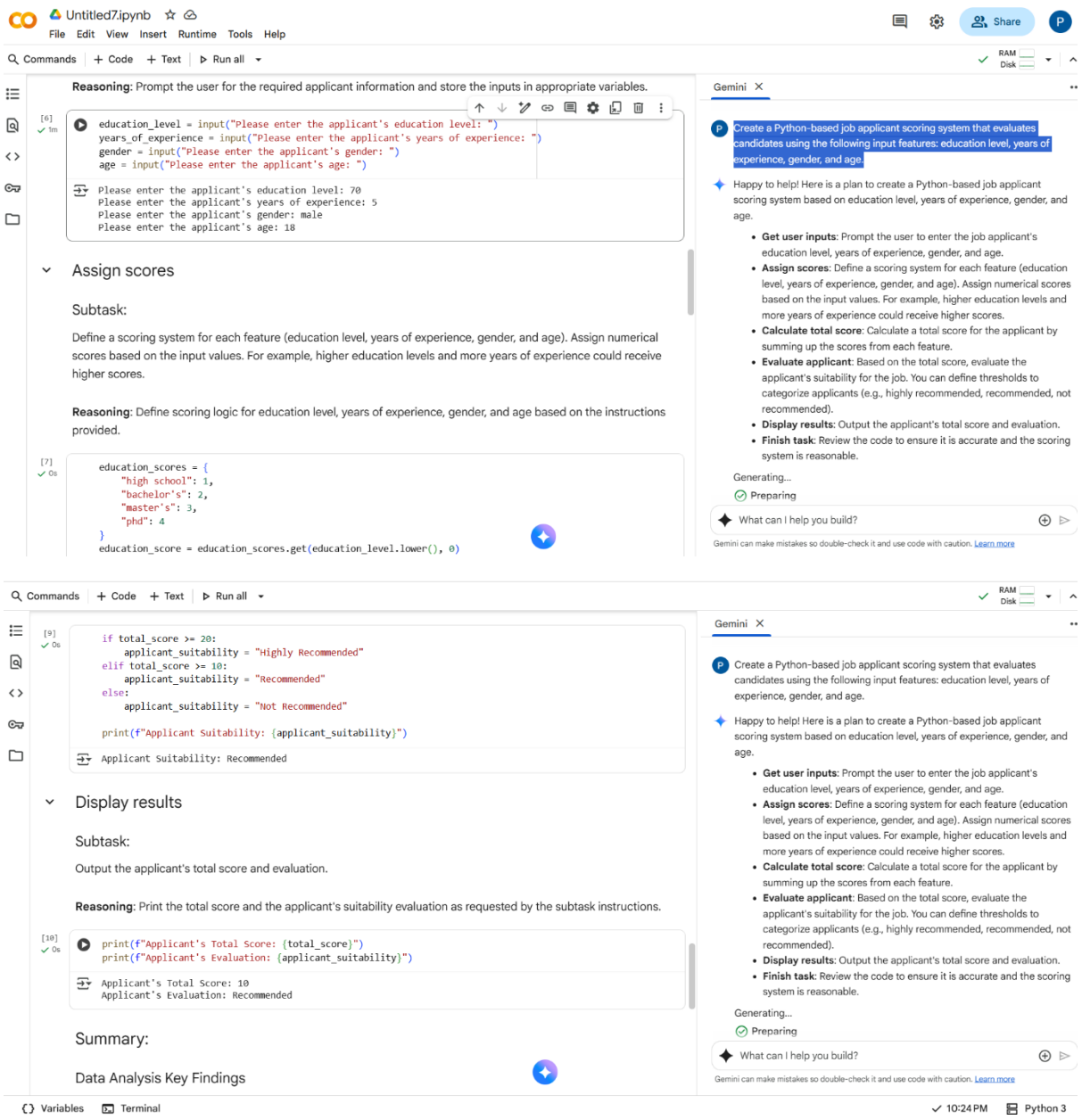
Observation : The code successfully calculates and prints the 8th Fibonacci number, which is 21, using the recursive approach defined in the fibonacci_recursive function.

Task-4:

Prompt: Generate a python code to a job applicant scoring system based on input features (e.g., education, experience, gender, age). Analyze the scoring logic for bias or unfair weightings. The input should be taken dynamic.

Code&Output:



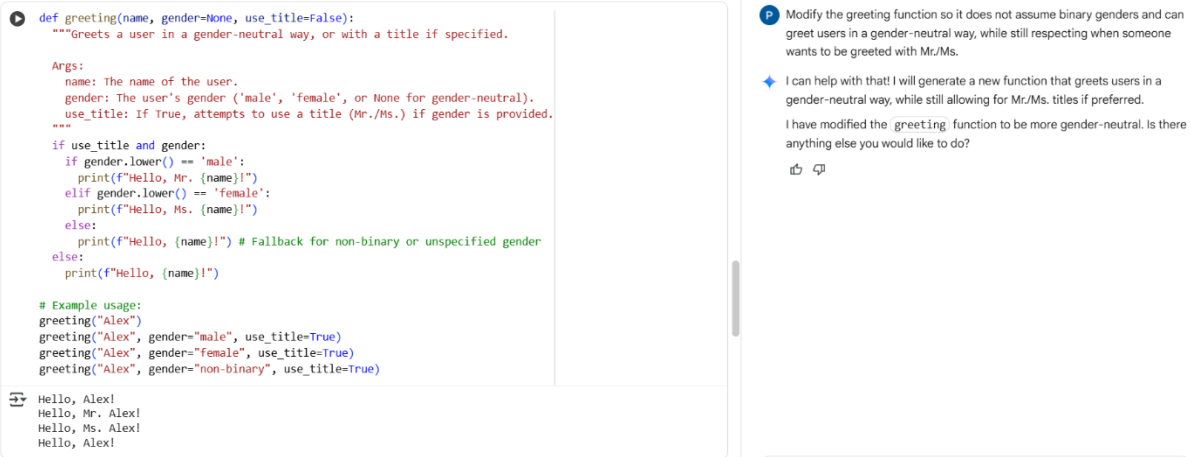


Observation : This code defines a function that scores job applicants based on education, experience, gender, and age. It then applies this function to a sample list of applicants and displays the resulting scores in a table. A simple observation is that the scores vary significantly among applicants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

Task-5:

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.

Code&Output:



```python
def greeting(name, gender=None, use_title=False):
  """Greets a user in a gender-neutral way, or with a title if specified.

  Args:
    name: The name of the user.
    gender: The user's gender ('male', 'female', or None for gender-neutral).
    use_title: If True, attempts to use a title (Mr./Ms.) if gender is provided.
  """
  if use_title and gender:
    if gender.lower() == 'male':
      print(f"Hello, Mr. {name}!")
    elif gender.lower() == 'female':
      print(f"Hello, Ms. {name}!")
    else:
      print(f"Hello, {name}!") # Fallback for non-binary or unspecified gender
  else:
    print(f"Hello, {name}!")

# Example usage:
greeting("Alex")
greeting("Alex", gender="male", use_title=True)
greeting("Alex", gender="female", use_title=True)
greeting("Alex", gender="non-binary", use_title=True)
```

```
Hello, Alex!
Hello, Mr. Alex!
Hello, Ms. Alex!
Hello, Alex!
```


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.

I can help with that! I will generate a new function that greets users in a gender-neutral way, while still allowing for Mr./Ms. titles if preferred.

I have modified the `greeting` function to be more gender-neutral. Is there anything else you would like to do?

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 diAerent 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.