**ASSIGNMENT-03**

**NAME:** M.Shiva Shankar

**HALL TICKET:**2303A51294

**BATCH:**05

**Q) Task 1: AI-Generated Logic for Reading Consumer Details**

**Scenario**

**An electricity billing system must collect accurate consumer data.**

**Task Description**

**Use an AI tool (GitHub Copilot / Gemini) to generate a Python program that:**

* **Reads: o Previous Units (PU) o Current Units (CU) o Type of Customer**
* **Calculates units consumed**
* **Implements logic directly in the main program (no functions) Expected Output**
* **Correct input reading**
* **Units consumed calculation**
* **Screenshot showing AI-generated code**
* **Sample input and output**

**PROMPT:**

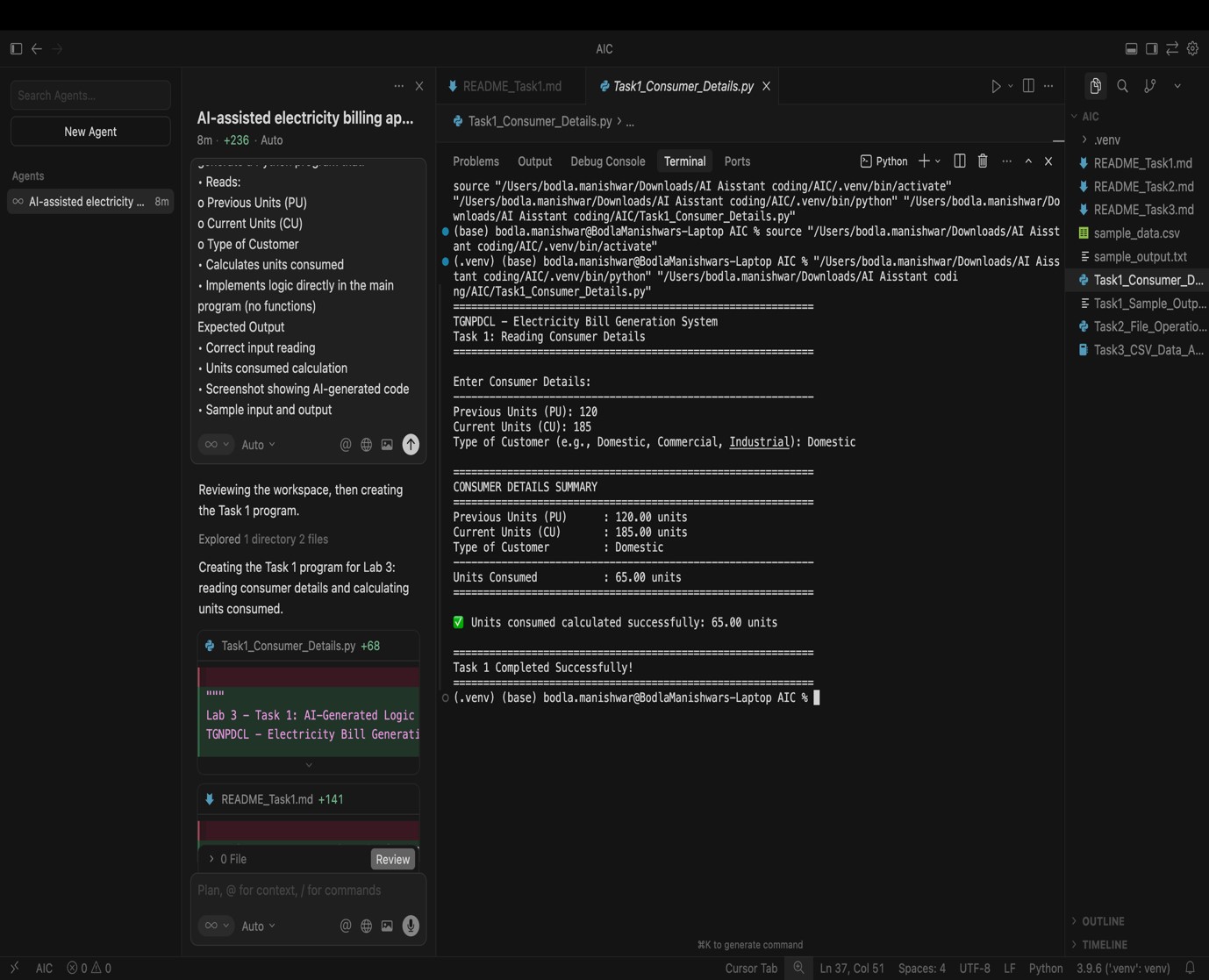
Create a Python program that reads previous units, current units, and customer type, calculates units consumed, and displays the result. Write all logic in the main program without using functions.

**CODE:**

A screenshot of a computer

AI-generated content may be incorrect.

**OUTPUT:**



**CODE EXPLANATION:**

This program collects consumer details such as previous meter units, current meter units, and customer type, then calculates the electricity units consumed by subtracting previous units from current units. It includes basic input validation to handle incorrect values and ensures accurate calculations. The logic is written directly in the main program without using functions, making it simple, clear, and suitable for a lab-based electricity billing application.

**Q) Task 2:** Energy Charges Calculation Based on Units Consumed

Scenario

Energy charges depend on the number of units consumed and customer type.

Task Description

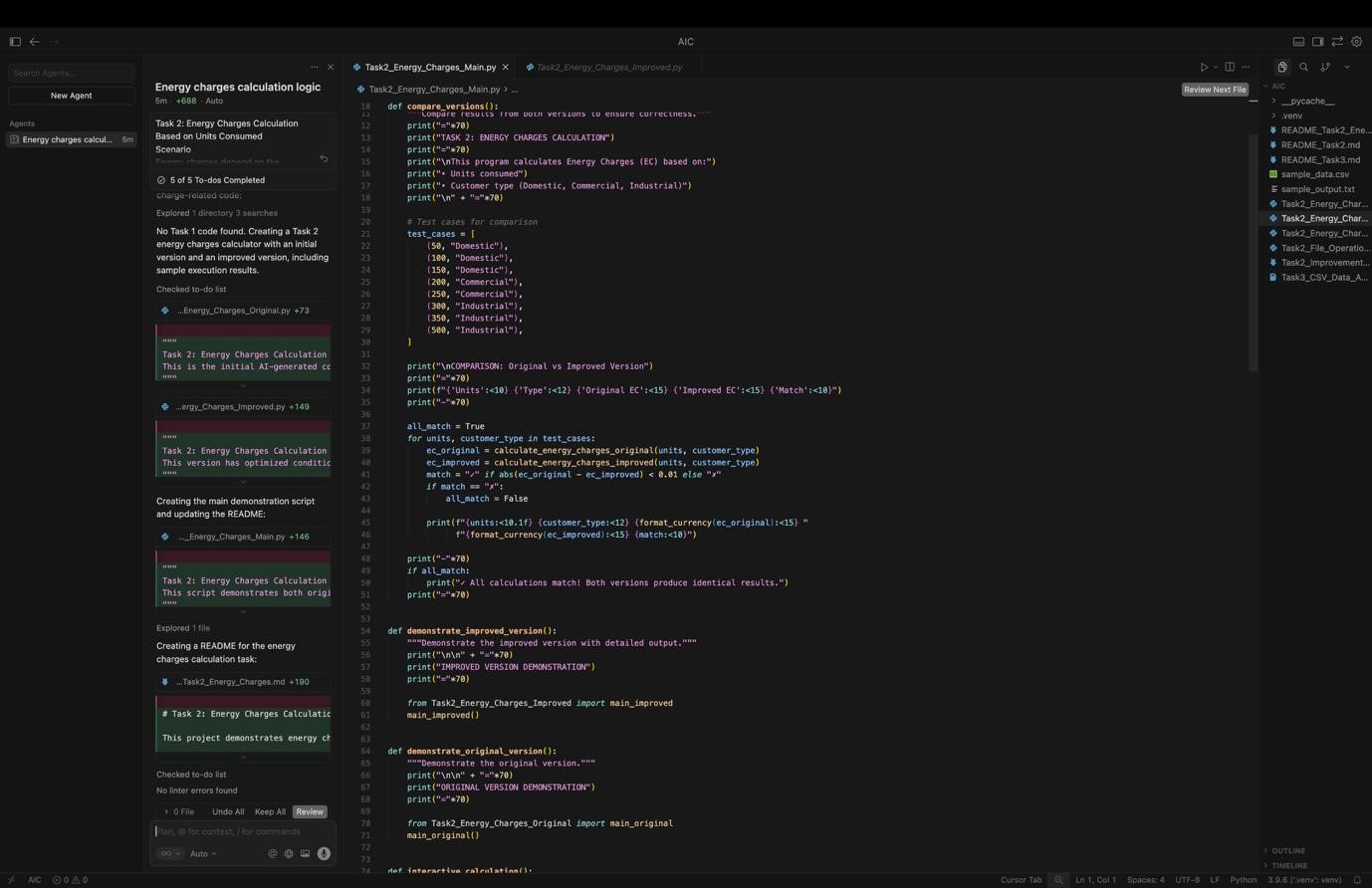
Review the AI-generated code from Task 1 and extend it to:

* Calculate Energy Charges (EC)
* Use conditional statements based on: o Domestic o Commercial o Industrial consumers
* Improve readability using AI prompts such as: o “Simplify energy charge calculation logic” o “Optimize conditional statements” Expected Output
* Correct EC calculation
* Clear conditional logic
* Original and improved versions (optional)
* Sample execution results

**PROMPT:**

Modify the Python billing code to calculate energy charges using conditional statements for Domestic, Commercial, and Industrial consumers and display the result clearly.

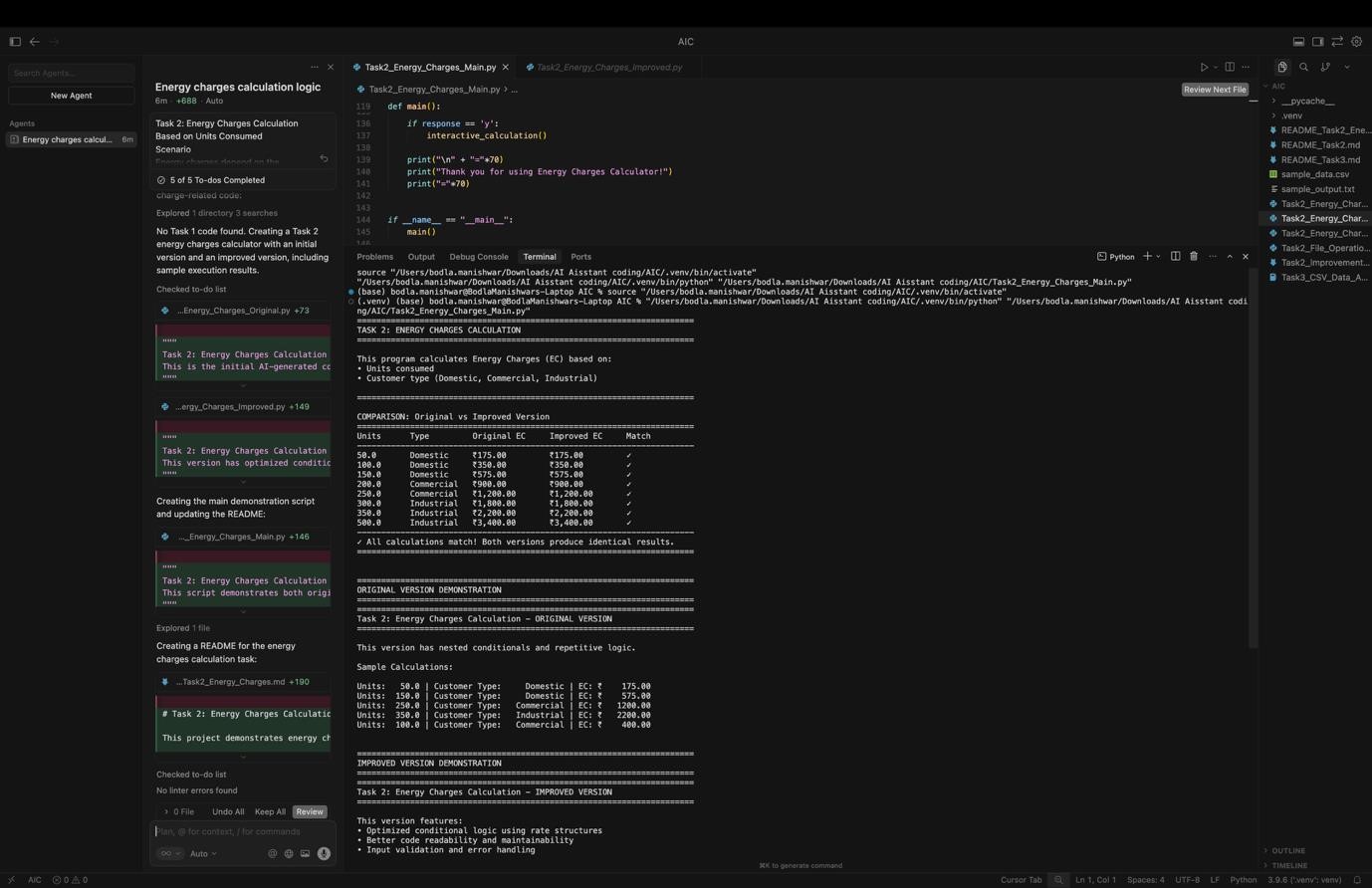
**CODE:**



A screenshot of a computer

AI-generated content may be incorrect.

**OUTPUT:**



A screenshot of a computer

AI-generated content may be incorrect.

**CODE EXPLANATION:**

This script demonstrates the calculation of Energy Charges (EC) based on units consumed and customer type. It compares two AI-generated implementations—an original version and an improved, optimized version—to ensure both produce the same results. The program uses conditional logic for Domestic, Commercial, and Industrial consumers, displays comparison results in a tabular format, and allows interactive user input for real-time calculation. This task highlights the use of AI tools to optimize conditional statements while maintaining correct billing logic.

**Q)** Task 3: Modular Design Using AI Assistance (Using Functions)

Scenario

Billing logic must be reusable for multiple consumers.

Task Description

Use AI assistance to generate a Python program that:

• Uses user-defined functions to:

o Calculate Energy Charges o Calculate Fixed Charges

* Returns calculated values
* Includes meaningful comments Expected Output
* Function-based Python program
* Correct EC and FC values
* Screenshots of AI-assisted function generation
* Test cases with outputs

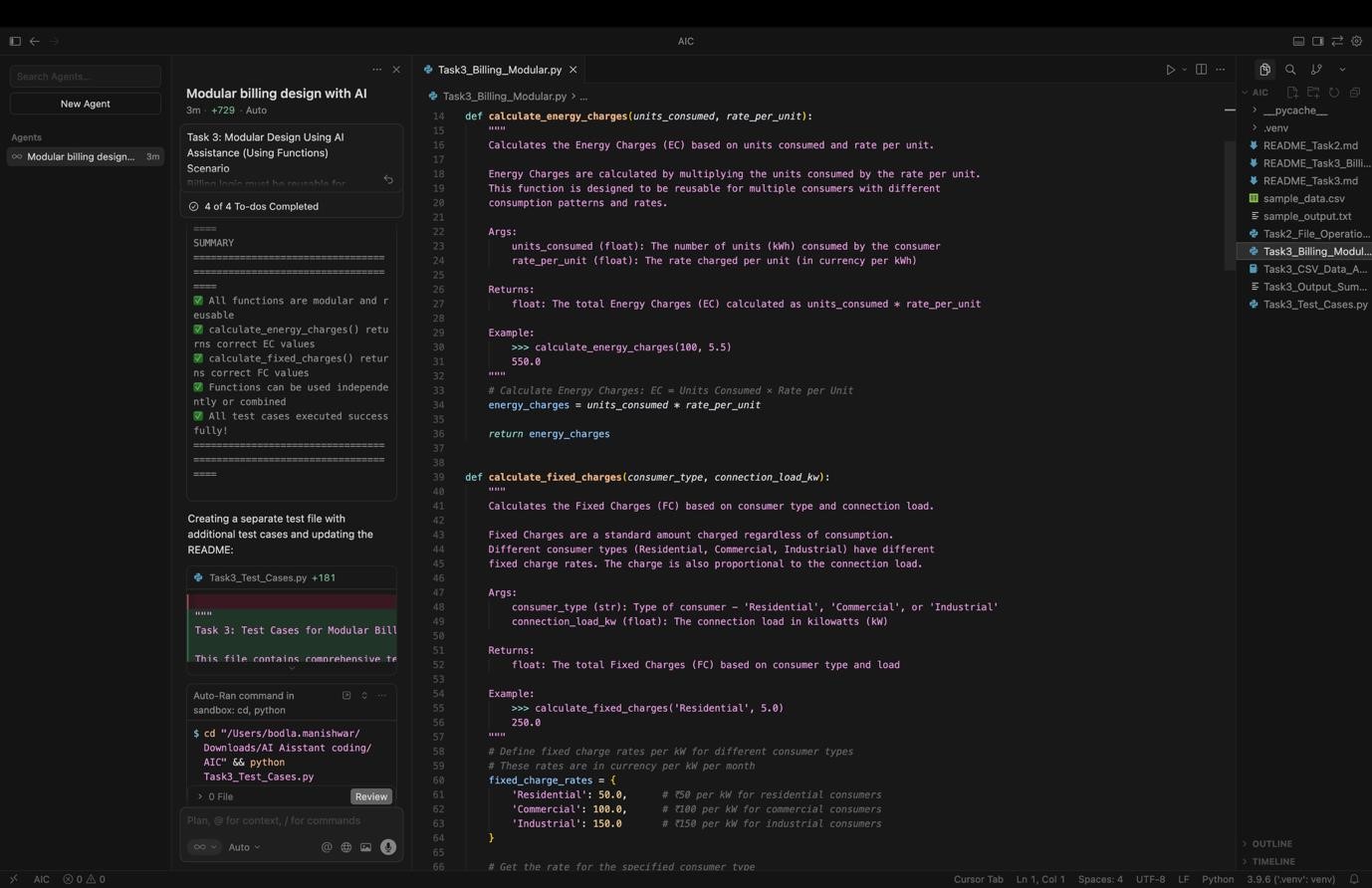
**PROMPT:**

Create a Python billing program using functions to calculate and return Energy Charges and Fixed Charges with proper comments and sample outputs.

**CODE:**

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

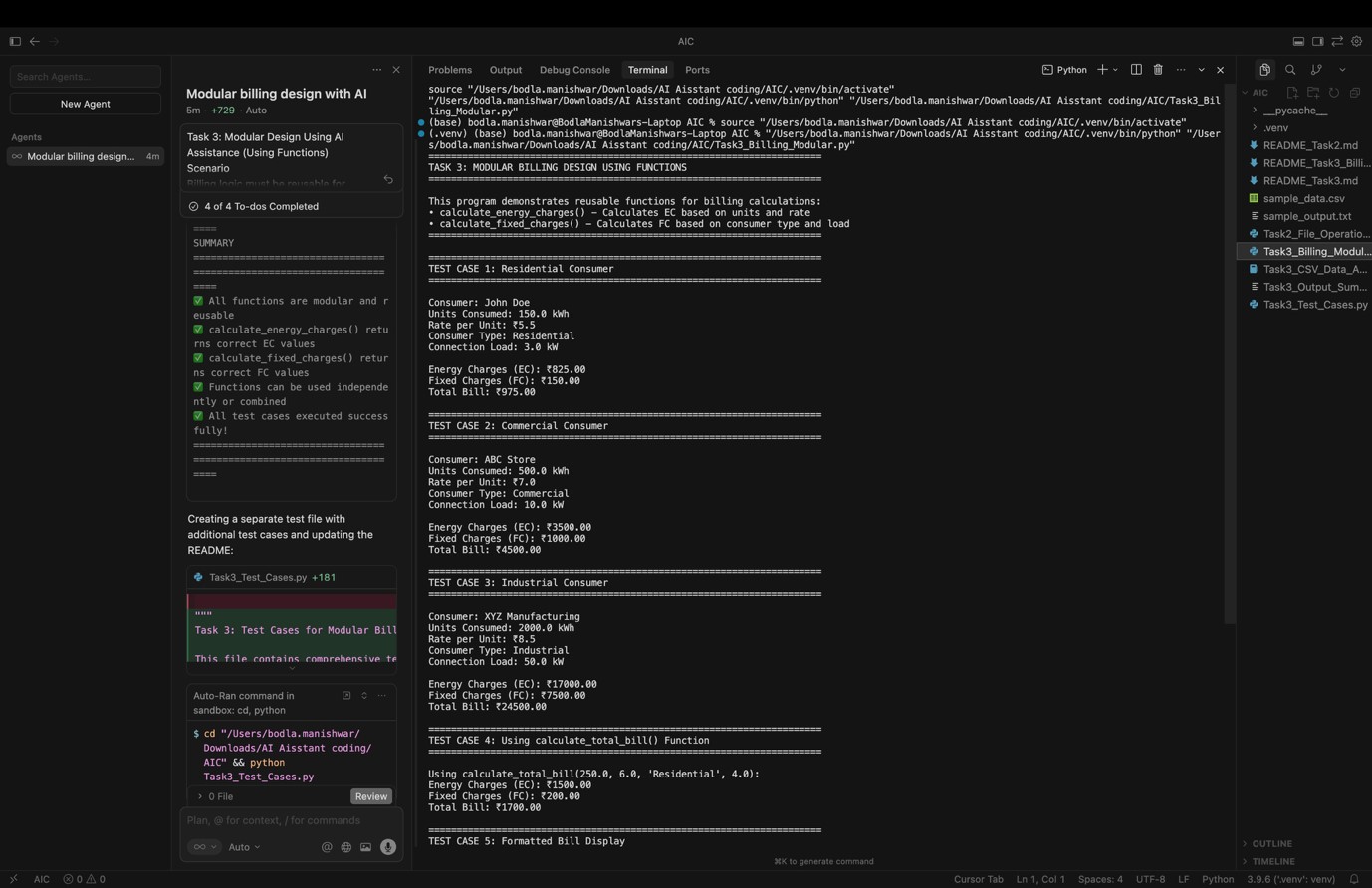
A screenshot of a computer

AI-generated content may be incorrect.

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.



**CODE EXPLANATION:**

This program demonstrates a modular electricity billing system using user-defined functions. Separate reusable functions are created to calculate Energy Charges (EC) based on units consumed and rate per unit, and Fixed Charges (FC) based on consumer type and connection load. These functions return calculated values, allowing them to be reused for multiple consumers. The program includes multiple test cases to verify correctness and shows how modular design improves readability, reusability, and maintainability of billing logic using AI assistance.

**Q) Task 4:** Calculation of Additional Charges

Scenario

Electricity bills include multiple additional charges.

Task Description

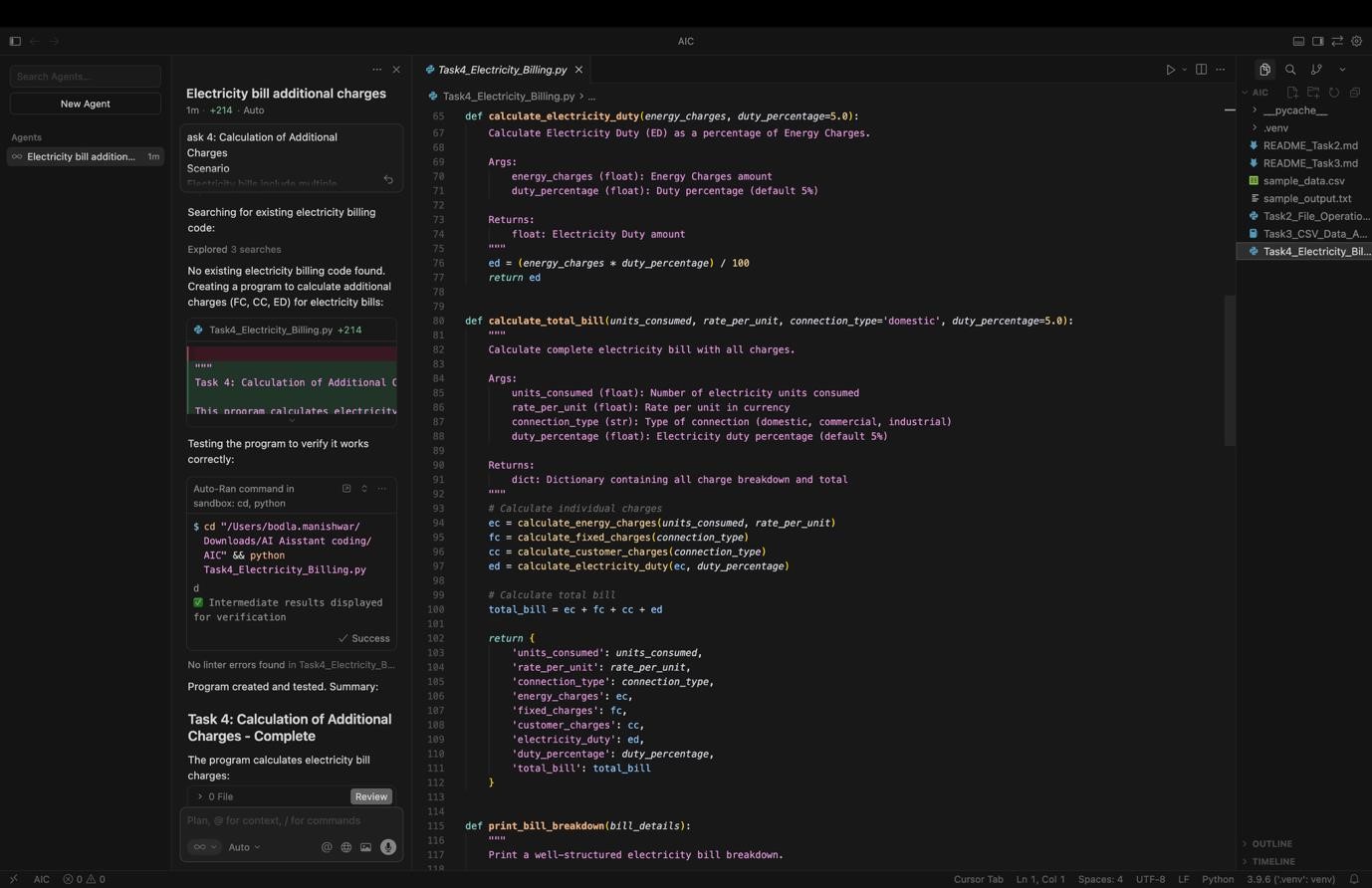
Extend the program to calculate:

* FC – Fixed Charges
* CC – Customer Charges
* ED – Electricity Duty (percentage of EC) Use AI prompts like:
* “Add electricity duty calculation”
* “Improve billing accuracy” Expected Output
* Individual charge values printed
* Correct duty calculation
* Well-structured output
* Verified intermediate results

**PROMPT:**

Modify the Python billing program to calculate FC, CC, and Electricity Duty (percentage of EC) and display all charges clearly.

**CODE:**



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**CODE EXPLANATION:** This program extends the electricity billing system to calculate additional charges such as Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED). Energy Charges are calculated based on units consumed, while FC and CC depend on the connection type. Electricity Duty is computed as a percentage of Energy Charges to ensure accurate billing. The program displays each charge separately with a well-structured bill format and shows intermediate calculations for verification, demonstrating improved billing accuracy using modular functions and AI assistance.

**Q) Task 5:** Final Bill Generation and Output Analysis

Scenario

The final electricity bill must present all values clearly.

Task Description

Develop the final Python application to:

* Calculate total bill:
* Total Bill = EC + FC + CC + ED
* Display: o Energy Charges (EC) o Fixed Charges (FC) o Customer Charges (CC) o Electricity Duty (ED) o Total Bill Amount
* Analyze the program based on: o Accuracy o Readability

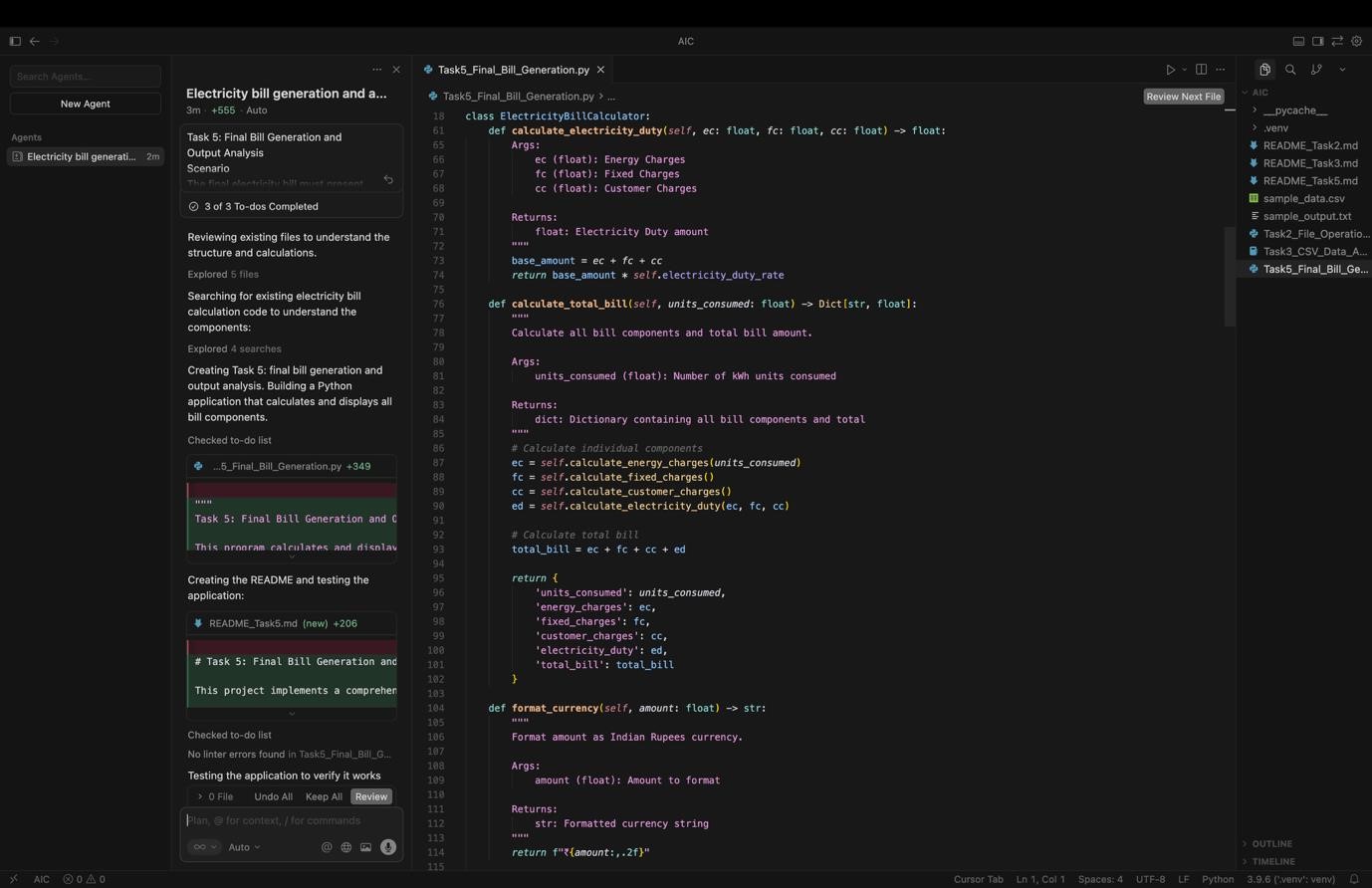
o Real-world applicability Expected Output

* Complete electricity bill output
* Neatly formatted display
* Sample input/output
* Short analysis paragraph

**PROMPT:**

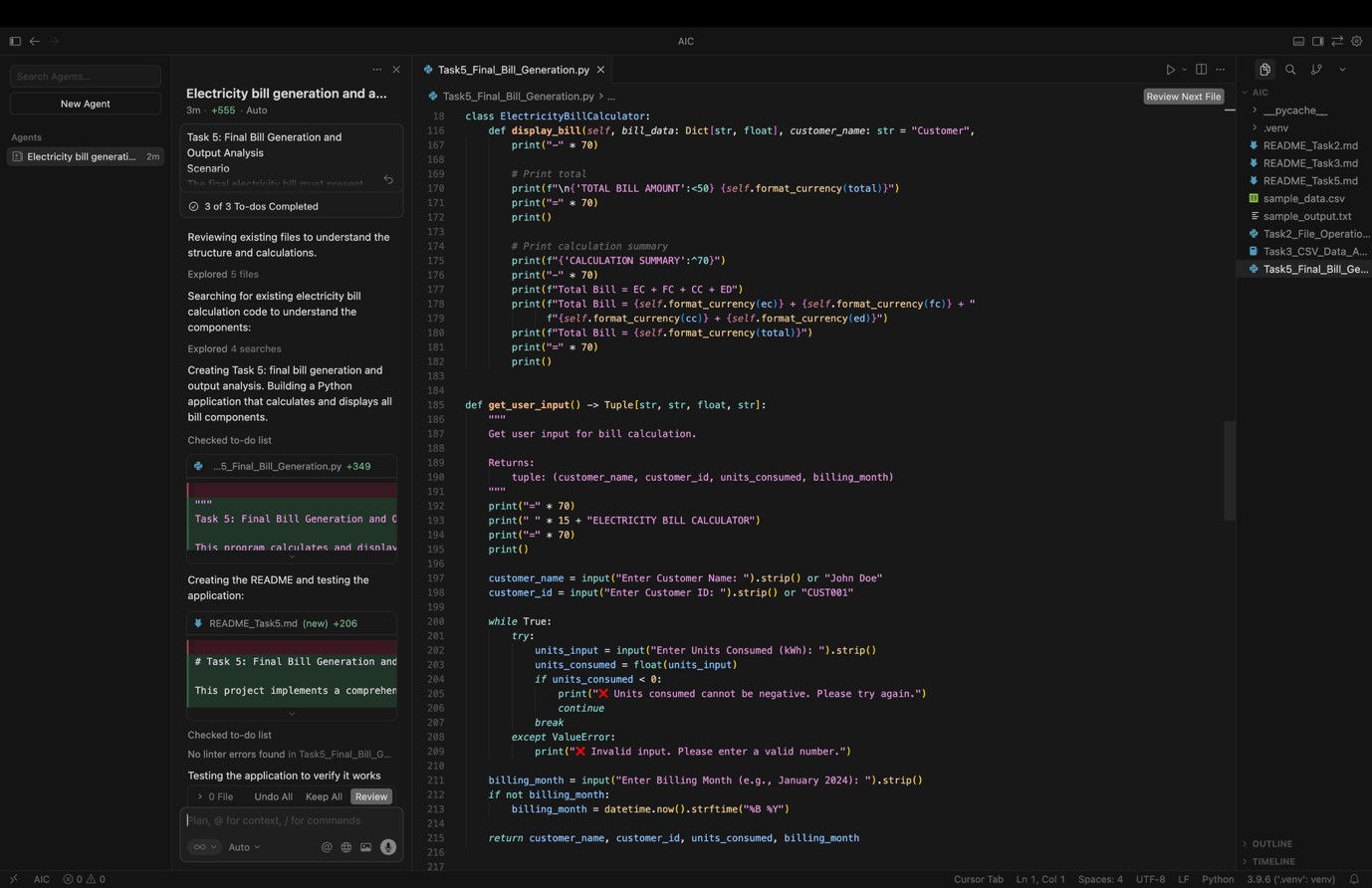
Create a Python program to generate a final electricity bill showing EC, FC, CC, ED, and total amount, with formatted output and a short analysis.

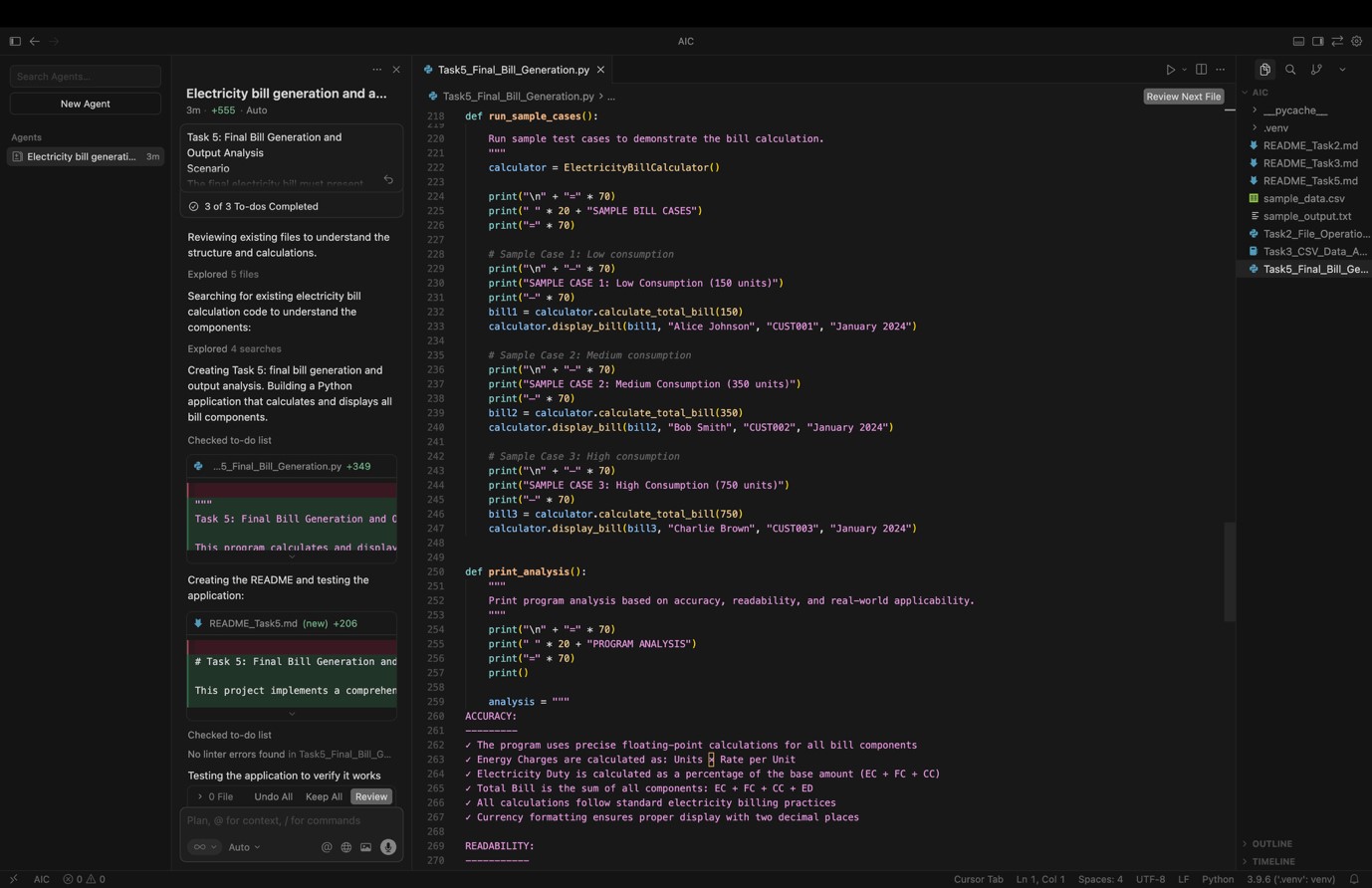
**CODE:**

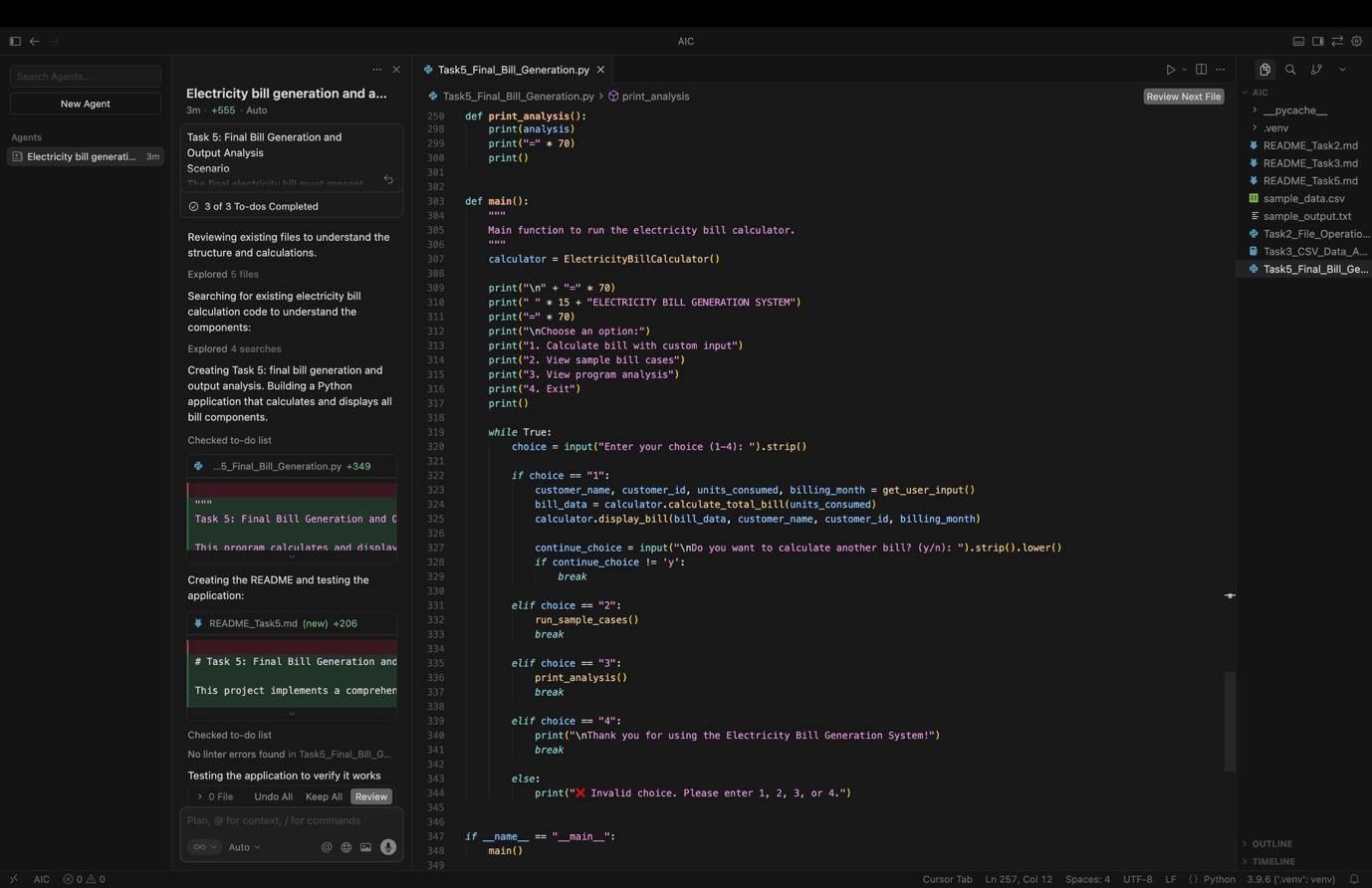


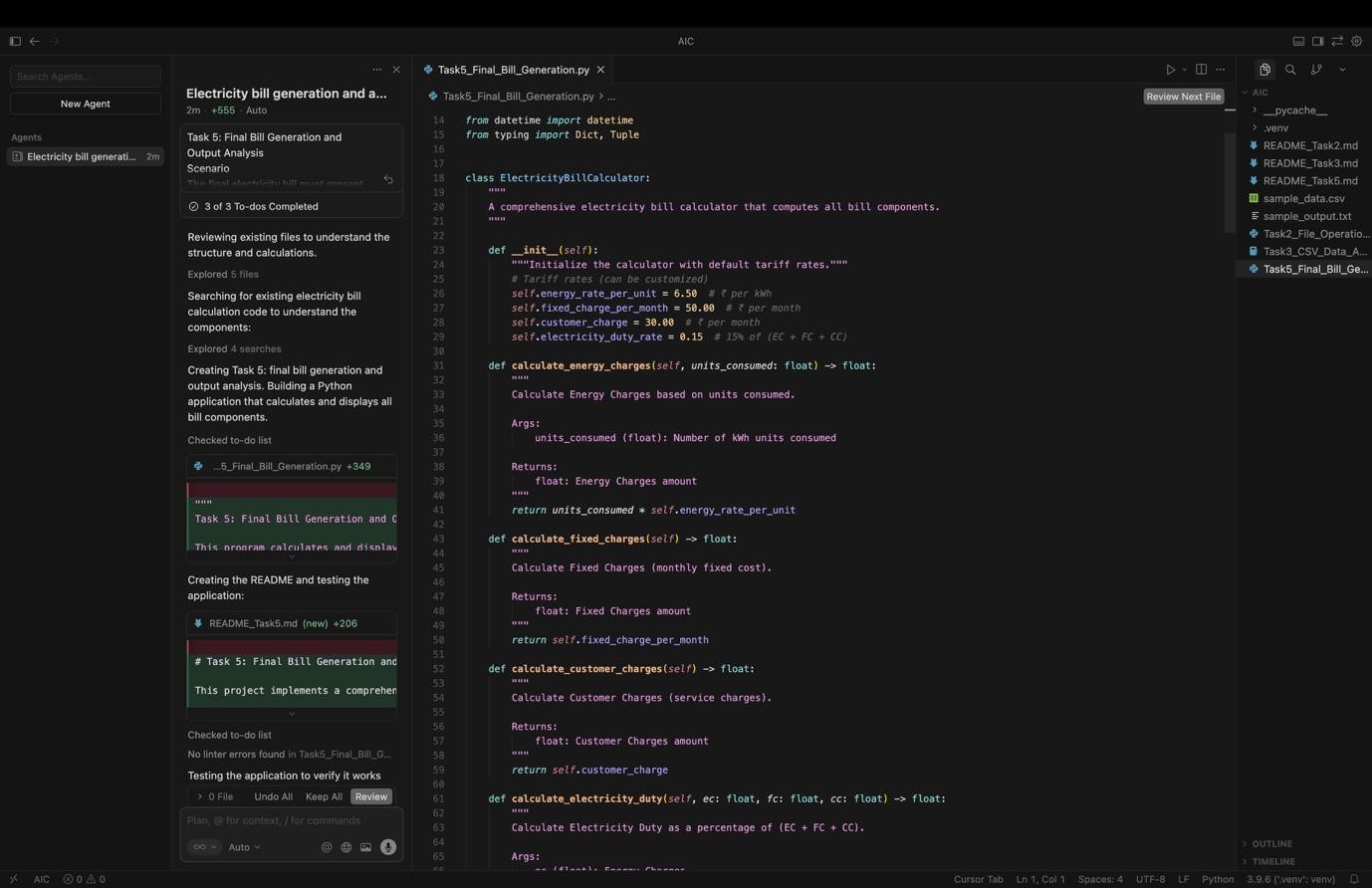
A screenshot of a computer

AI-generated content may be incorrect.

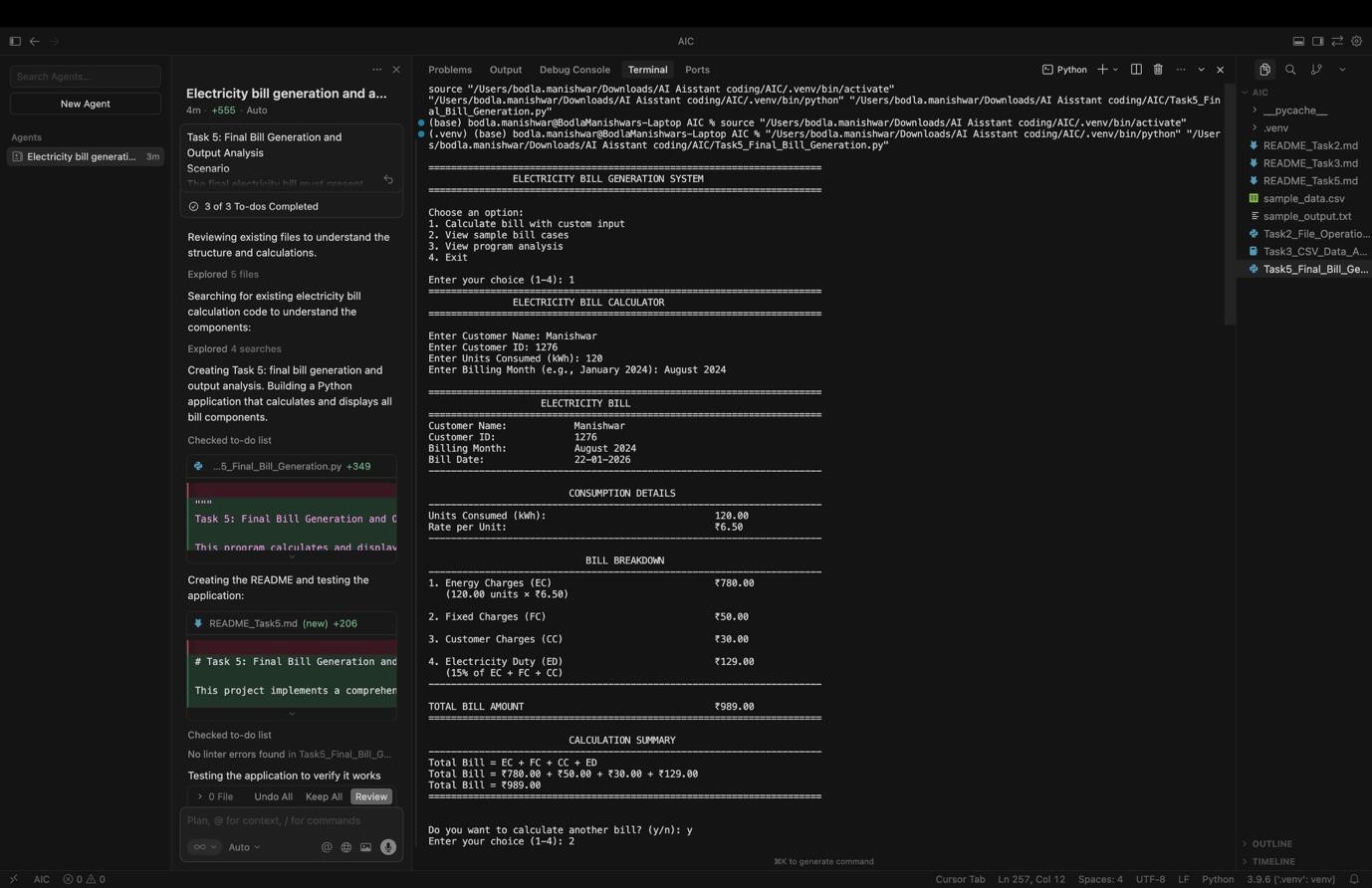


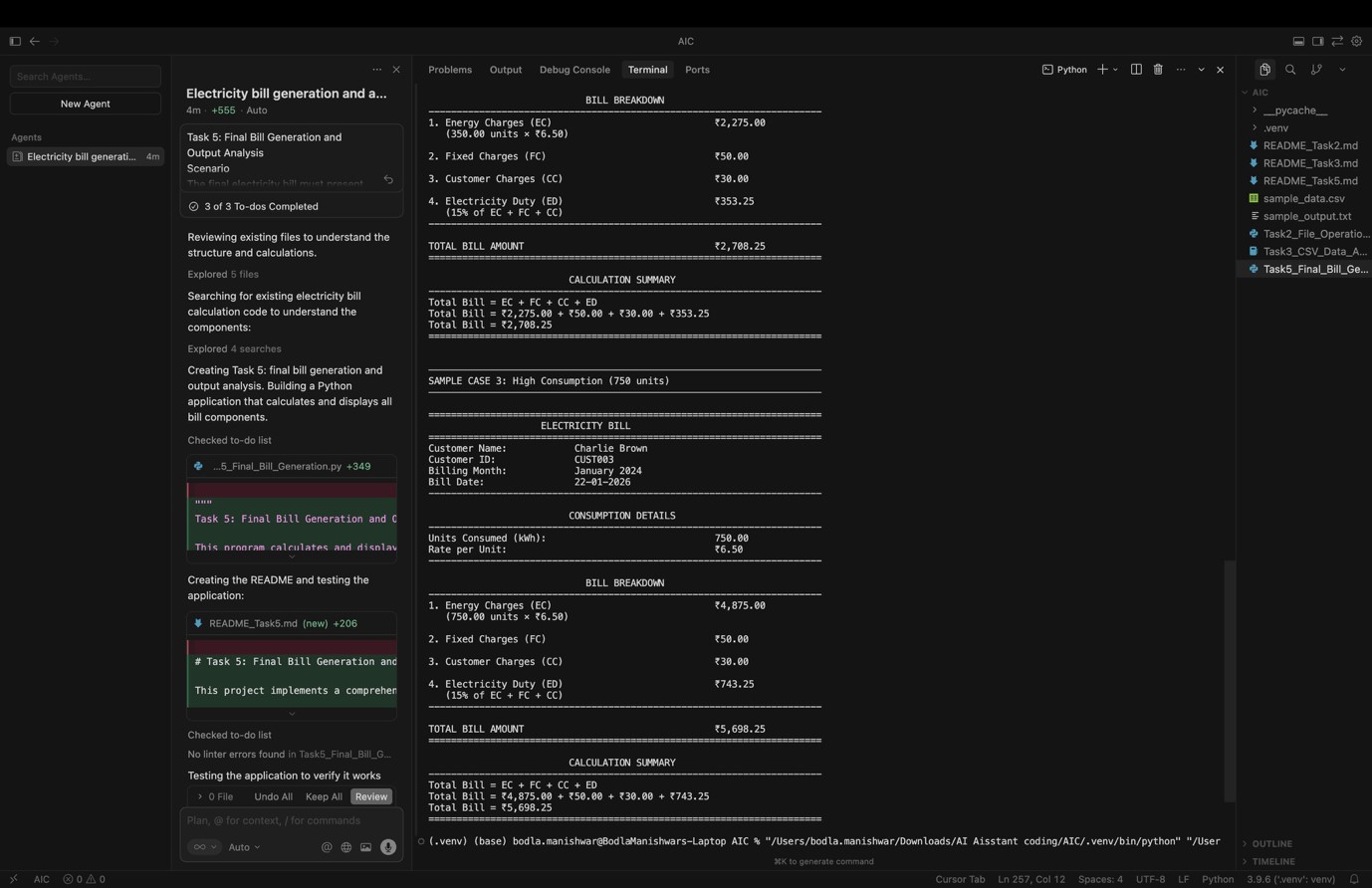


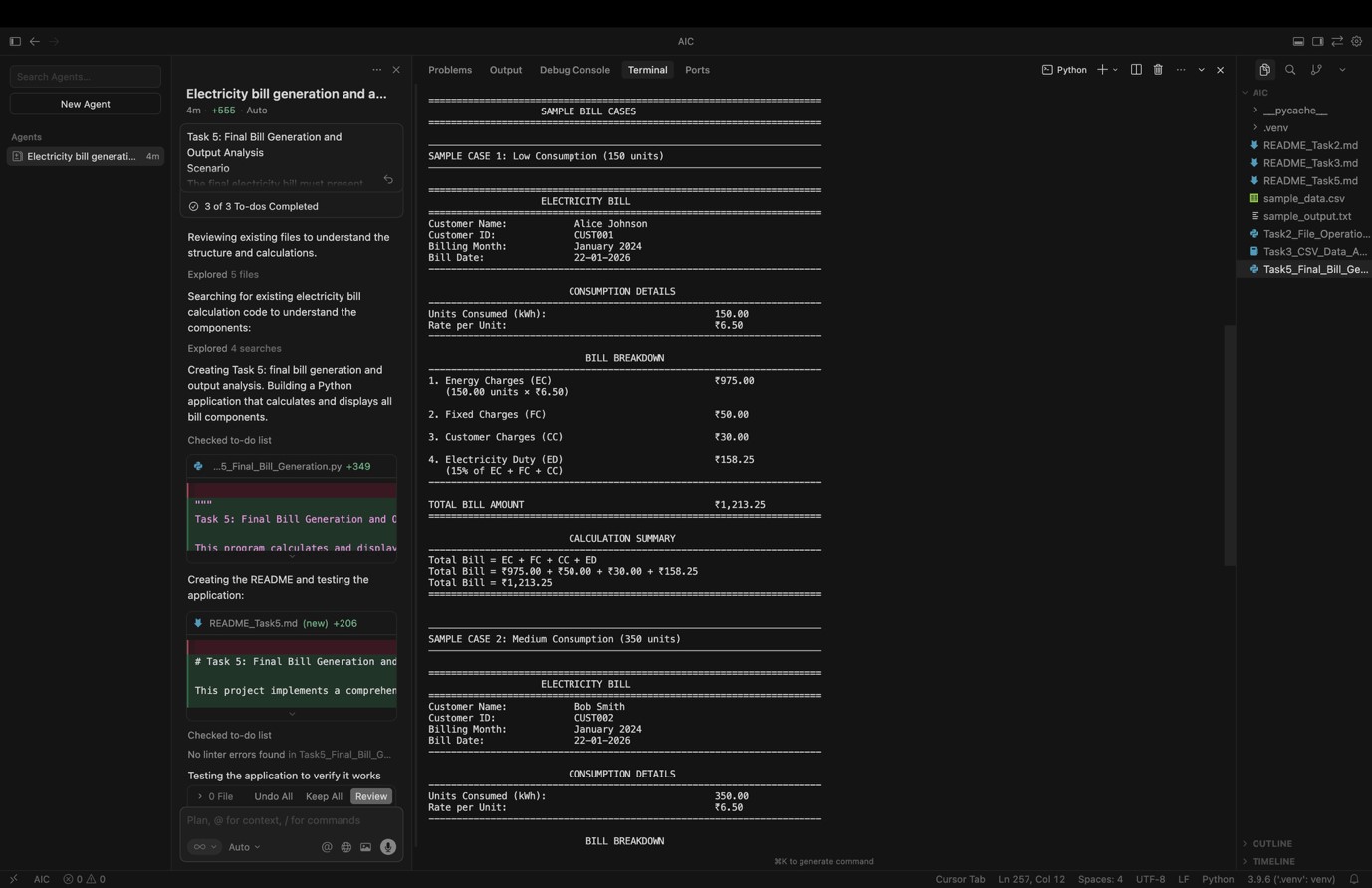




**OUTPUT:**







**CODE EXPLANATION:**

This program generates a complete electricity bill by accurately calculating Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED), and then computing the total bill as the sum of all components. It presents the bill in a neatly formatted, professional layout with clear sectioning for consumption details and charge breakdown.