SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name	Dr. Rishabh Mittal		
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju		
Course Code	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/I	**Regulation**	R23
Date and Day of Assignment	Week 2 - Wednesday	**Time(s)**	23CSBTB01 To 23CSBTB52
Duration	2 Hours	**Applicable to Batches**	All batches
Assignment Number: 3.3(Present assignment number)/**24**(Total number of assignments)			

	Question	***Expected Time to complete***
1	**Lab 3: Application for TGNPDCL – Electricity Bill Generation Using Python & AI Tools** **Lab Objectives** • To design a real-world electricity billing application using Python • To use AI-assisted coding tools for logic generation and optimization • To understand conditional logic and arithmetic operations	Week2 - Wednesday

- To generate structured billing output similar to utility bills

Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Read and validate user input in Python
- Apply conditional logic for tariff-based billing
- Use AI tools to assist in program development
- Calculate and display electricity bill components
- Build a complete real-time application

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:

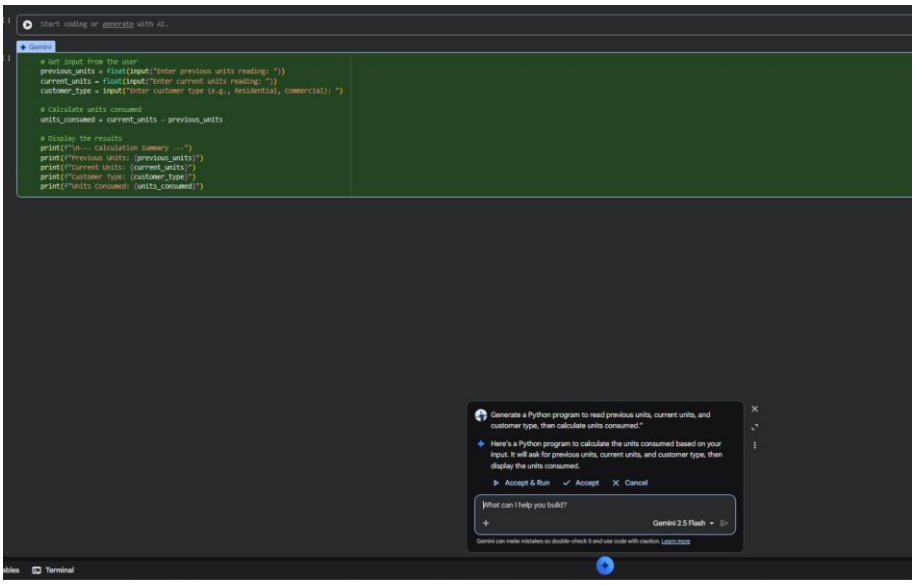
- Reads:
  - Previous Units (PU)
  - Current Units (CU)
  - Type of Customer
- Calculates units consumed
- Implements logic directly in the main program (no functions)

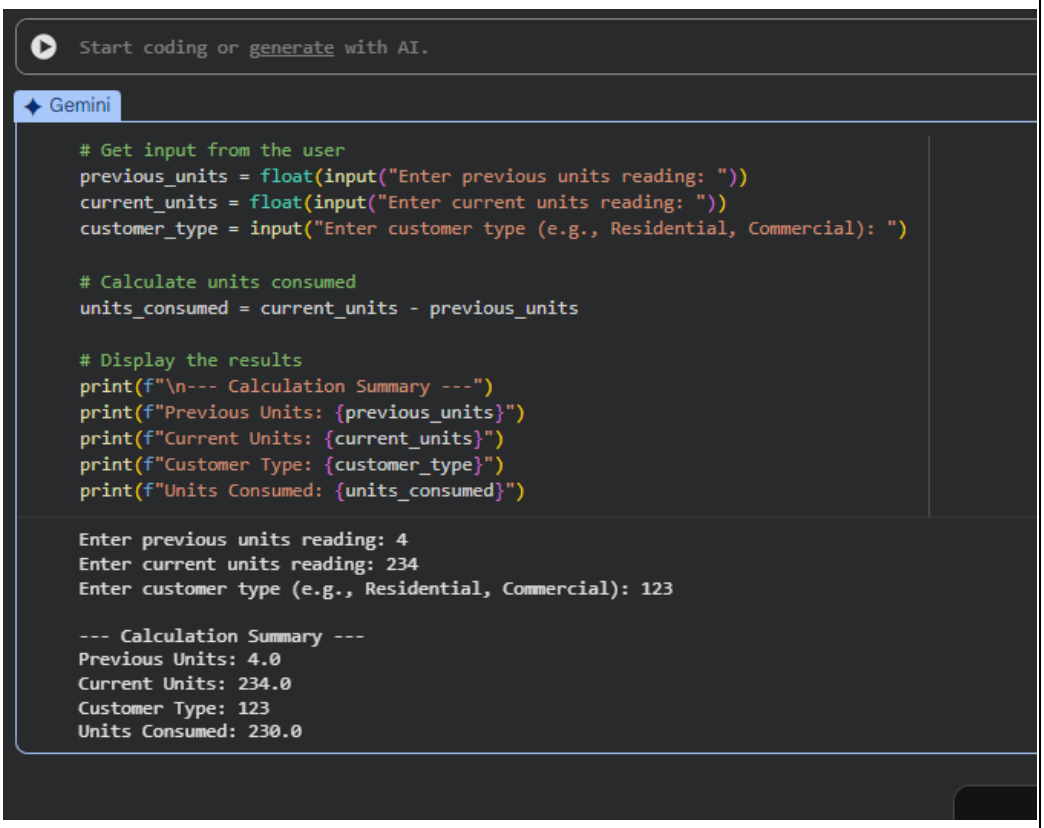
Expected Output

- Correct input reading
- Units consumed calculation
- Screenshot showing AI-generated code
- Sample input and output

Prompt

Generate a Python program to read previous units, current units, and customer type, then calculate units consumed."





Description

This task collects electricity consumer details such as previous meter reading, current meter reading, and customer type.
The program calculates the number of units consumed by subtracting previous units from current units.
All logic is written directly in the main program.

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:

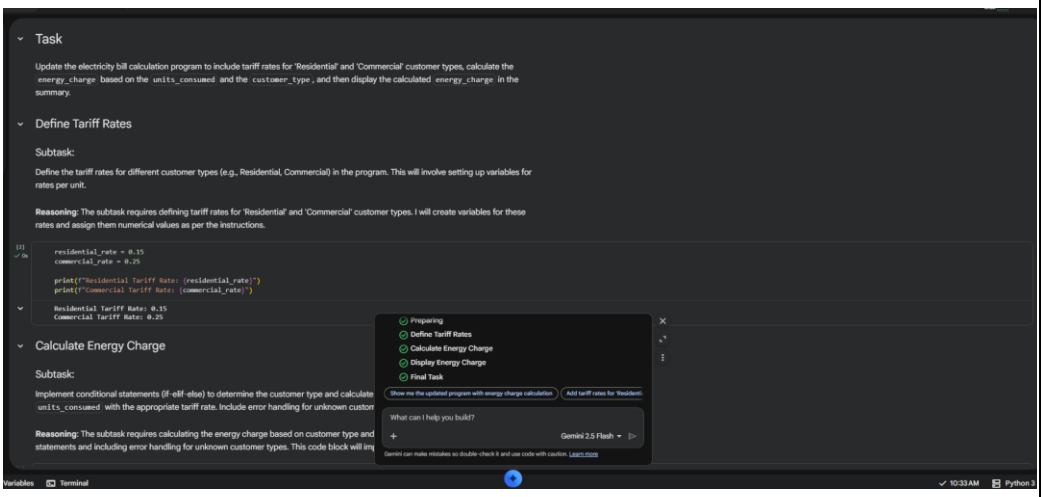
- Calculate **Energy Charges (EC)**
- Use conditional statements based on:
  - Domestic
  - Commercial
  - Industrial consumers
- Improve readability using AI prompts such as:
  - *"Simplify energy charge calculation logic"*
  - *"Optimize conditional statements"*

Expected Output

- Correct EC calculation
- Clear conditional logic
- Original and improved versions (optional)
- Sample execution results

AI Prompt

"Extend the program to calculate energy charges using conditional statements based on customer type and optimize the logic."



Description

Energy charges are calculated based on the number of units consumed and the type of customer.

Different tariffs are applied for Domestic, Commercial, and Industrial consumers using conditional statements.

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:
  - Calculate Energy Charges

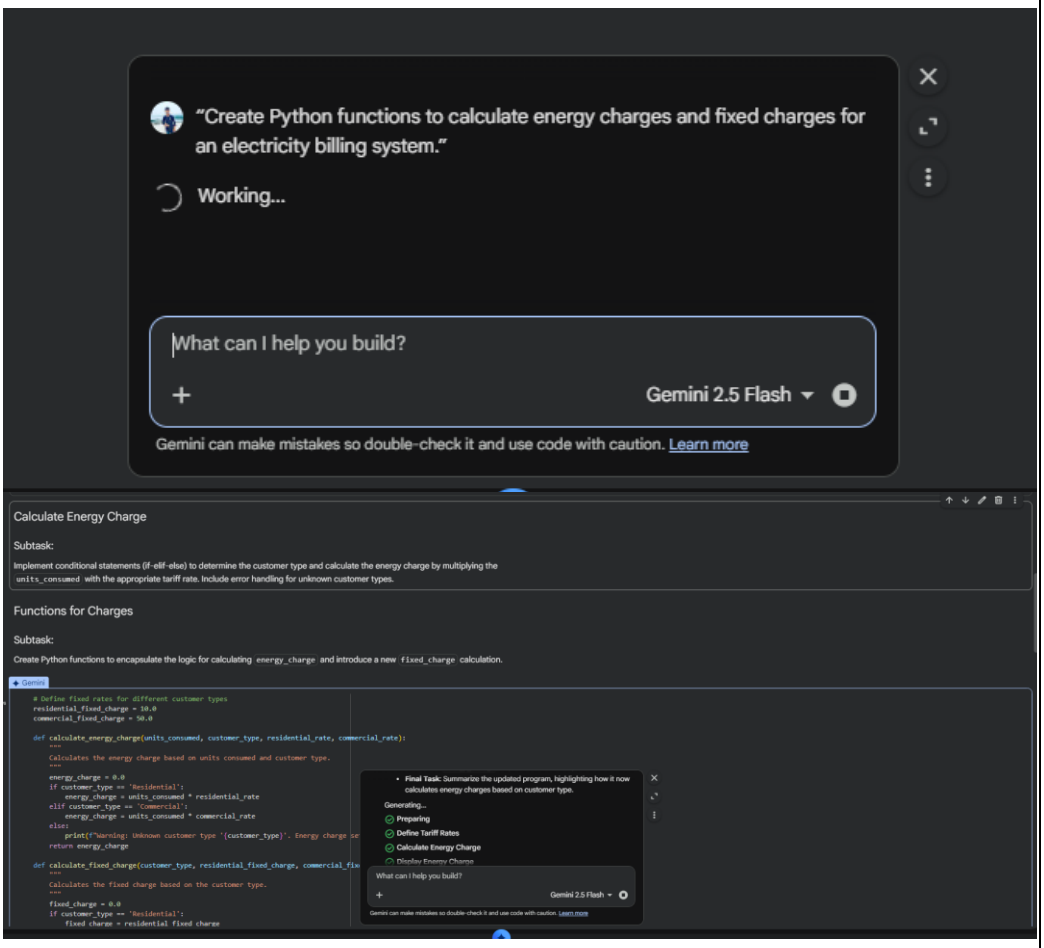
- Calculate Fixed Charges
- Returns calculated values
- Includes meaningful comments

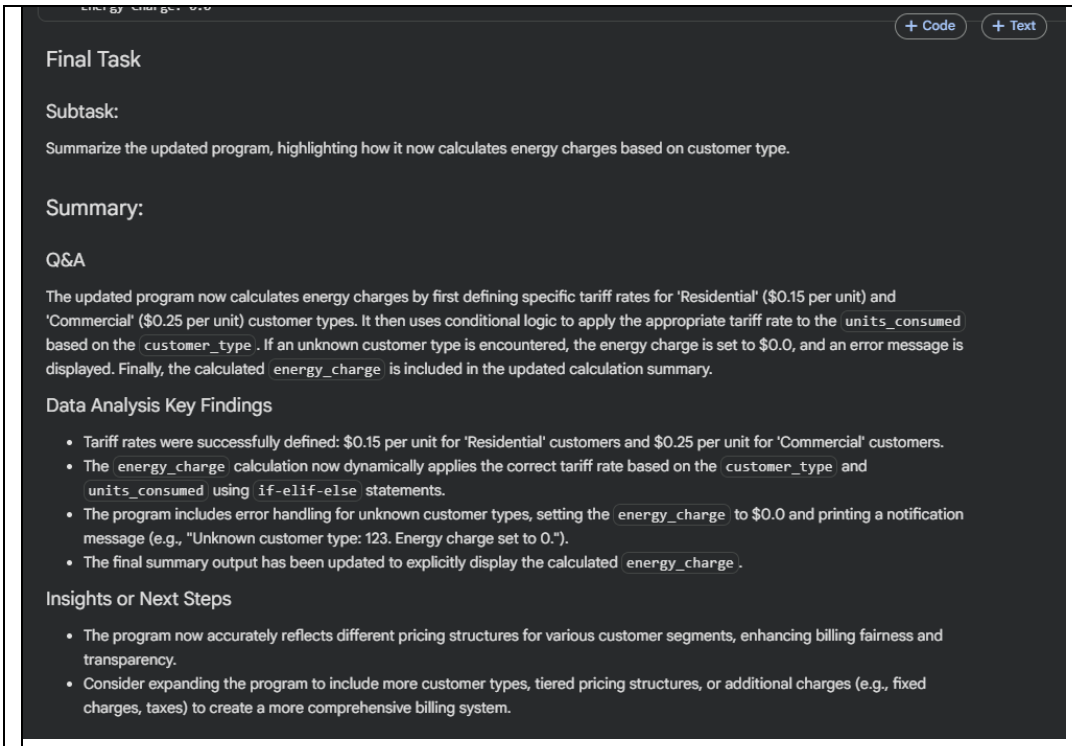
Expected Output

- Function-based Python program
- Correct EC and FC values
- Screenshots of AI-assisted function generation
- Test cases with outputs

AI Prompt

"Create Python functions to calculate energy charges and fixed charges for an electricity billing system."





To improve reusability and readability, the billing logic is divided into user-defined functions.
Separate functions are created for calculating energy charges and fixed charges.

Description

Additional charges such as Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED) are calculated.
Electricity Duty is calculated as 5% of Energy Charges.

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

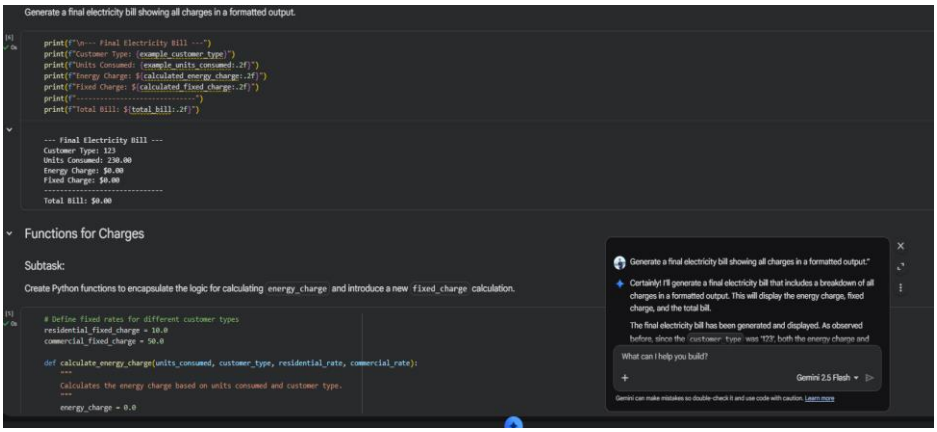
- Correct duty calculation
- Well-structured output
- Verified intermediate results

AI Prompt

"Generate a final electricity bill showing all charges in a formatted output."

Description

The final bill amount is calculated by summing Energy Charges, Fixed Charges, Customer Charges, and Electricity Duty.
The output is displayed in a structured format similar to a real electricity bill.



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:
  - Energy Charges (EC)
  - Fixed Charges (FC)
  - Customer Charges (CC)
  - Electricity Duty (ED)
  - Total Bill Amount
- Analyze the program based on:
  - Accuracy
  - Readability
  - Real-world applicability

Expected Output

- Complete electricity bill output
- Neatly formatted display
- Sample input/output

- Short analysis paragraph

Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments, & code explanation, and output and if required, screenshots.

Program Analysis

- **Accuracy**:
  The program correctly calculates the total electricity bill using all required components and follows the specified formula.
- **Readability**:
  Clear variable names and structured output make the program easy to understand and maintain.
- **Real-World Applicability**:
  The bill format and calculations closely resemble real electricity billing systems such as TGNPDCL.