ASSINGMENT-3.3

2303A51964

U.VIGNESH

BATCH-24

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

Code:

```
#Application for TGNPDCL-electricity bill generation take 3 user inputs ,Previous Units,Current Units,
# Type of Customer(Domestic/Commercial/Industrial consumers) and calculate units consumed .
previous_units = float(input("Enter previous units: "))
current_units = float(input("Enter current units: "))
customer_type = input("Enter customer type (Domestic/Commercial/Industrial): ")
if current_units < previous_units:
    print("Error: Current units cannot be less than previous units.")
    exit()
units_consumed = current_units - previous_units
print(f"Units consumed: {units_consumed}")
```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/OneDrive/Desktop/AI ASSESTE
NT CODING/Assingment-3.3.py"
Enter previous units: 20
Enter current units: 30
Enter customer type (Domestic/Commercial/Industrial): domestic
Units consumed: 10.0
```

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

Code:

```python
#write a function for Energy charges calculation based on customer type and units consumed,optimal cod
def calculate_energy_charges(units, customer_type):
    energy_charges = 0
    if customer_type.lower() == 'domestic':
        if units <= 100:
            energy_charges = units * 1.5
        elif units <= 300:
            energy_charges = (100 * 1.5) + ((units - 100) * 2.5)
        else:
            energy_charges = (100 * 1.5) + (200 * 2.5) + ((units - 300) * 4.0)
    elif customer_type.lower() == 'commercial':
        if units <= 100:
            energy_charges = units * 2.0
        elif units <= 300:
            energy_charges = (100 * 2.0) + ((units - 100) * 3.5)
        else:
            energy_charges = (100 * 2.0) + (200 * 3.5) + ((units - 300) * 5.0)
    elif customer_type.lower() == 'industrial':
        if units <= 100:
            energy_charges = units * 3.0
        elif units <= 300:
            energy_charges = (100 * 3.0) + ((units - 100) * 4.5)
        else:
            energy_charges = (100 * 3.0) + (200 * 4.5) + ((units - 300) * 6.0)
    else:
        print("Invalid customer type")
    return energy_charges
energy_charges = calculate_energy_charges(units_consumed, customer_type)
print(f"Energy charges for {customer_type} customer: {energy_charges}")
```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/O
neDrive/Desktop/AI ASSESTENT CODING/Assingment-3.3.py"
Enter previous units: 20
Enter previous units: 20
Enter current units: 30
Enter customer type (Domestic/Commercial/Industrial): domestic
Units consumed: 10.0
Energy charges for domestic customer: 15.0
```

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**

Code:

```python
#give python optimal code to calculate fixed charges based on units consumed and customer type.
def calculate_fixed_charges(units, customer_type):
    fixed_charges = 0
    if customer_type.lower() == 'domestic':
        if units <= 100:
            fixed_charges = 50
        elif units <= 300:
            fixed_charges = 100
        else:
            fixed_charges = 150
    elif customer_type.lower() == 'commercial':
        if units <= 100:
            fixed_charges = 100
        elif units <= 300:
            fixed_charges = 200
        else:
            fixed_charges = 300
    elif customer_type.lower() == 'industrial':
        if units <= 100:
            fixed_charges = 250
        elif units <= 300:
            fixed_charges = 500
        else:
            fixed_charges = 750
    return fixed_charges

fixed_charges = calculate_fixed_charges(units_consumed, customer_type)
print(f"Fixed charges for {customer_type} customer: {fixed_charges}")
```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/O
neDrive/Desktop/AI ASSESTENT CODING/Assingment-3.3.py"
Enter previous units: 10
Enter current units: 20
Enter customer type (Domestic/Commercial/Industrial): domestic
Units consumed: 10.0
Energy charges for domestic customer: 15.0
Fixed charges for domestic customer: 50
```

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**

Code:

```
#extend the program to calculate customer charges and add electricity duty calculation to improve billing accuracy without
# creating functions.
customer_charges = 20  # Flat customer charge
electricity_duty_rate = 0.05  # 5% electricity duty
total_customer_charges = energy_charges + fixed_charges + customer_charges
electricity_duty = total_customer_charges * electricity_duty_rate
print(f"Total customer charges: {total_customer_charges}")
print(f"Electricity duty: {electricity_duty}")
```

Output:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/O
neDrive/Desktop/AI ASSESTENT CODING/Assingment-3.3.py"
Enter previous units: 10
Enter current units: 20
Enter customer type (Domestic/Commercial/Industrial): domestic
Units consumed: 10.0
Energy charges for domestic customer: 15.0
Fixed charges for domestic customer: 50
Total customer charges: 85.0
Electricity duty: 4.25
```

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

Code:

```
#calculate total bill amount by adding ec+fc+cc+ed.
total_bill_amount = total_customer_charges + electricity_duty + energy_charges + fixed_charges
#display energy charges, fixed charges, customer charges, electricity duty, total bill amount.
print("----- Bill Details -----")
print(f"Energy Charges: {energy_charges}")
print(f"Fixed Charges: {fixed_charges}")
print(f"Customer Charges: {customer_charges}")
print(f"Electricity Duty: {electricity_duty}")
print(f"Total Bill Amount: {total_bill_amount}")
```

Output:

```
----- Bill Details -----
Energy Charges: 15.0
Fixed Charges: 50
Customer Charges: 20
Electricity Duty: 4.25
Total Bill Amount: 154.25
```