# AI Assisted Coding Lab – 3.1

**Name:** Gattu Varshith

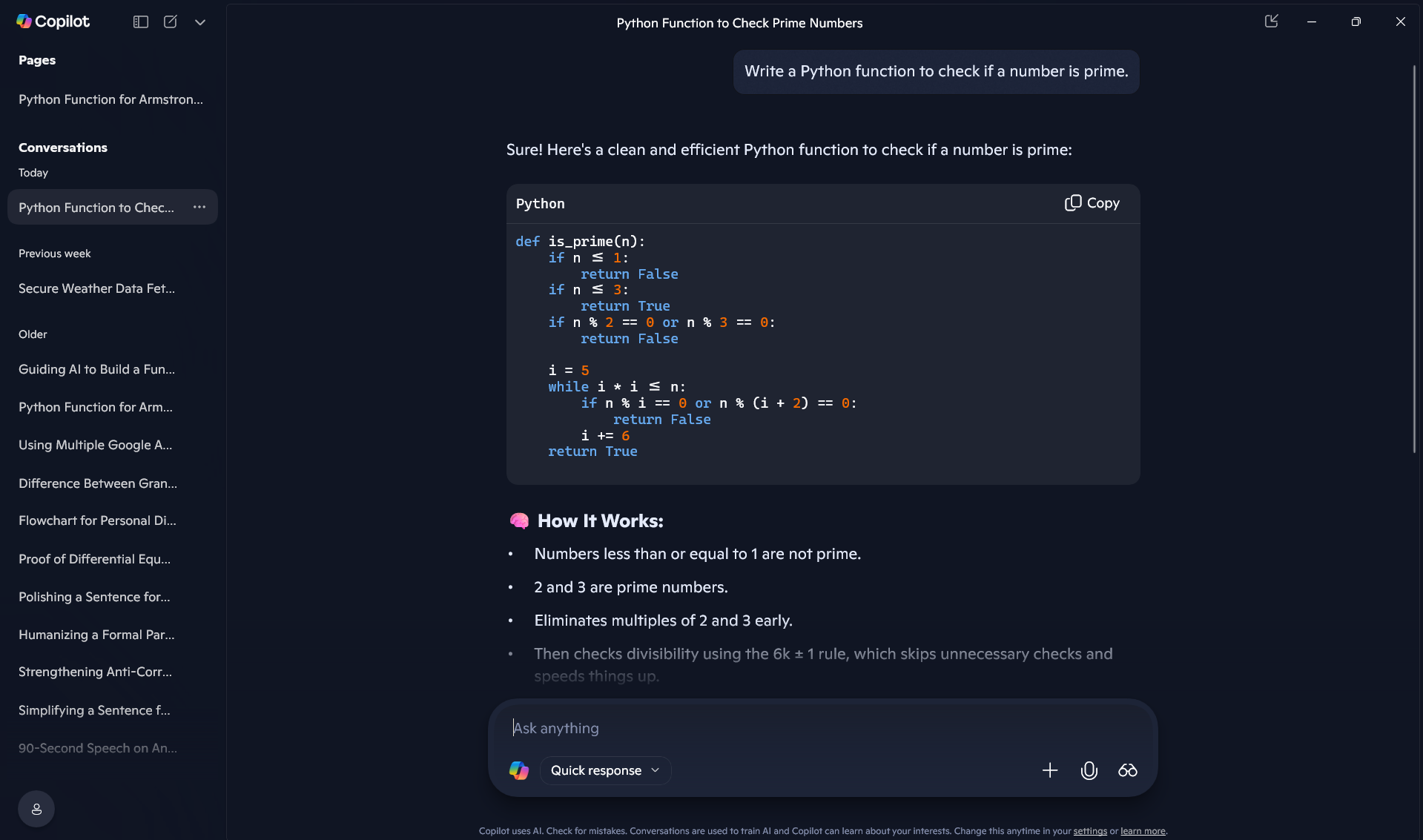
**Roll Number:** 2403A510E3

**Batch:** 24BTCAICSB05

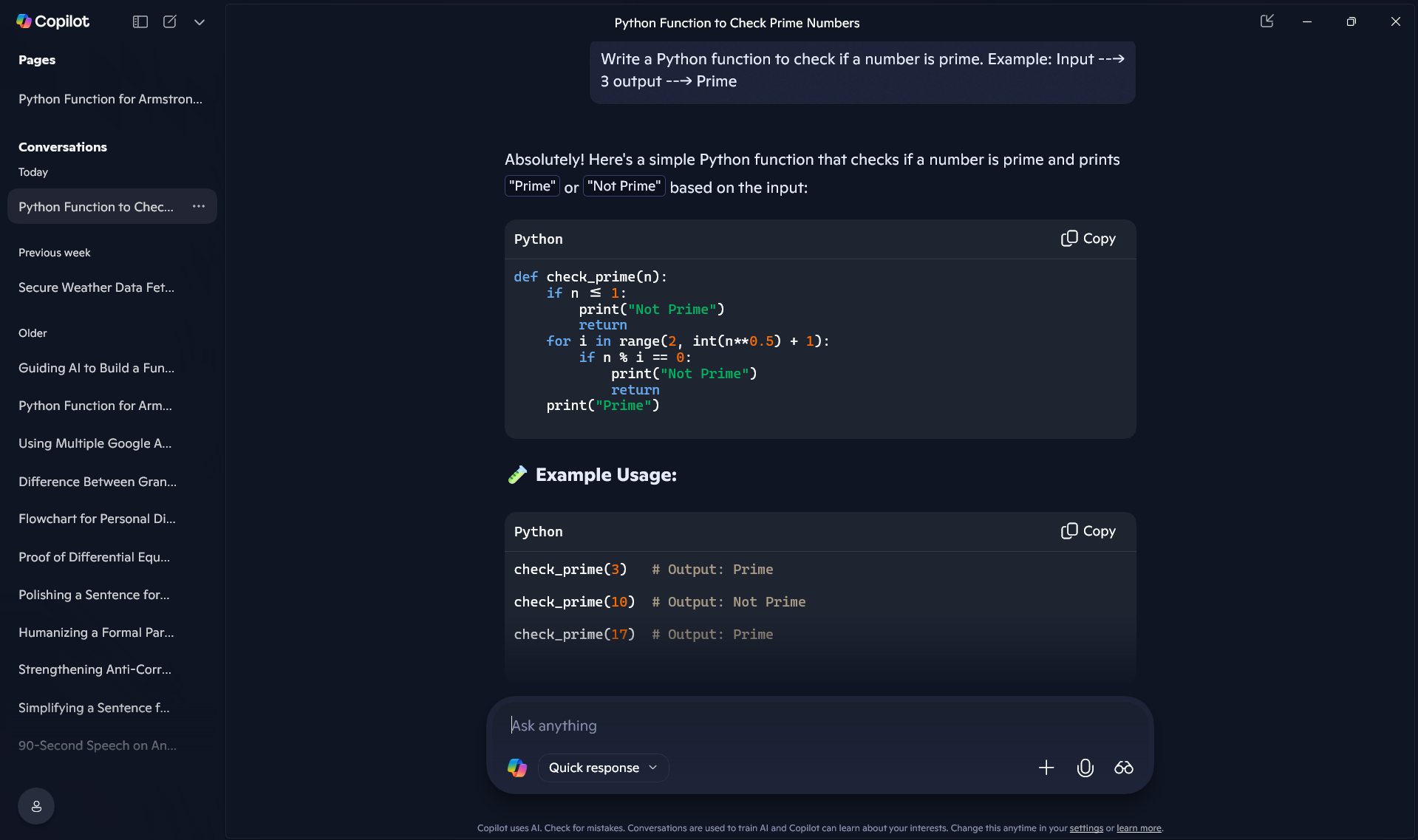
## Task 1: Prompt Engineering – Improving Prompts and Context Management

Objective: To explore how prompt design and context influence AI-generated outputs and to learn techniques to improve AI responses.  
  
Task: Use different prompting strategies to generate a Python function that checks if a number is prime. Compare the results for Zero-Shot, One-Shot, Few-Shot, and Context-Managed prompts. Then request AI to optimize the logic.

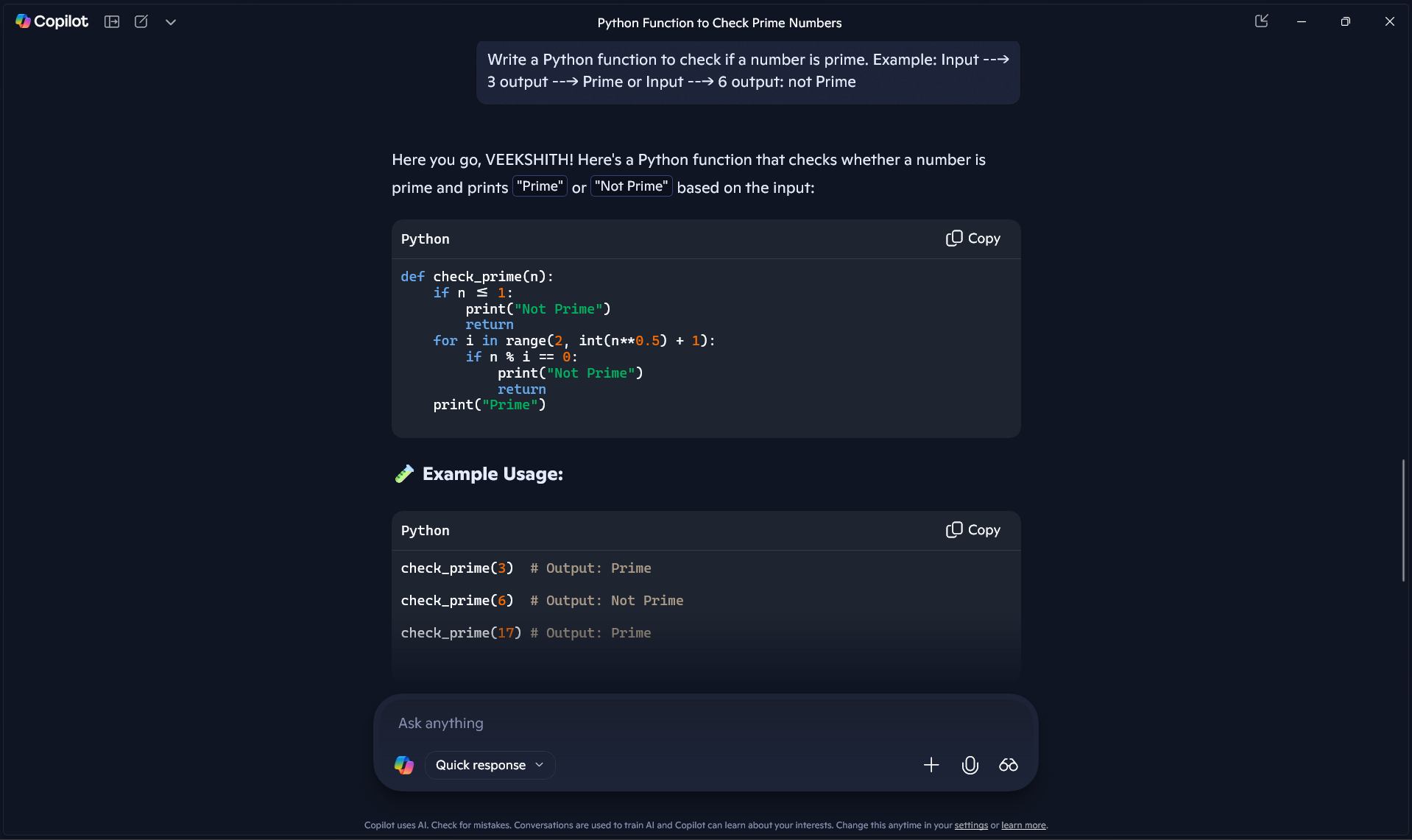
• Zero-Shot Prompt: Directly asked AI to write a function. Output worked but was simple.



• One-Shot Prompt: Gave one example. AI wrote a correct function with example-based logic.



• Few-Shot Prompt: Provided multiple examples. AI generated more accurate and safer code.



## Task 2: Mobile Data Usage Billing Application

Objective: Use Python programming and AI-assisted coding tools to create a mobile data billing system. It should take inputs for Data Consumed (in GB), Plan Type, and Additional Services, then calculate Data Charges (DC), Value-added Charges (VC), Tax, and Total bill amount. Finally, it should display an itemized bill.

• Used AI (Copilot) to generate initial code for billing system.

• Inputs taken: Data Consumed, Plan Type, Additional Services.

• Charges calculated for Data, Services, and Tax, then displayed in an itemized format.

**Copilot’s code:**

def calculate\_bill():

print("📱 Welcome to Telecom Billing System 📱")

# Input

data\_gb = float(input("Enter data consumed (in GB): "))

plan\_type = input("Enter plan type (Prepaid/Postpaid): ").strip().lower()

services = input("Enter additional services used (comma-separated): ").strip().lower().split(',')

# Rates

data\_rate = 10 if plan\_type == "prepaid" else 8 # ₹ per GB

service\_rates = {

"caller tune": 30,

"ott subscription": 100,

"international roaming": 200,

"cloud storage": 50

}

tax\_rate = 0.18 # 18%

# Calculations

dc = data\_gb \* data\_rate

vc = sum(service\_rates.get(service.strip(), 0) for service in services if service.strip())

subtotal = dc + vc

tax = subtotal \* tax\_rate

total = subtotal + tax

# Output

print("\n📄 Itemized Bill 📄")

print(f"Plan Type: {plan\_type.capitalize()}")

print(f"Data Used: {data\_gb} GB")

print(f"Data Charges (DC): ₹{dc:.2f}")

print(f"Value-added Services: {', '.join([s.strip().capitalize() for s in services if s.strip()]) or 'None'}")

print(f"Value-added Charges (VC): ₹{vc:.2f}")

print(f"Tax (18%): ₹{tax:.2f}")

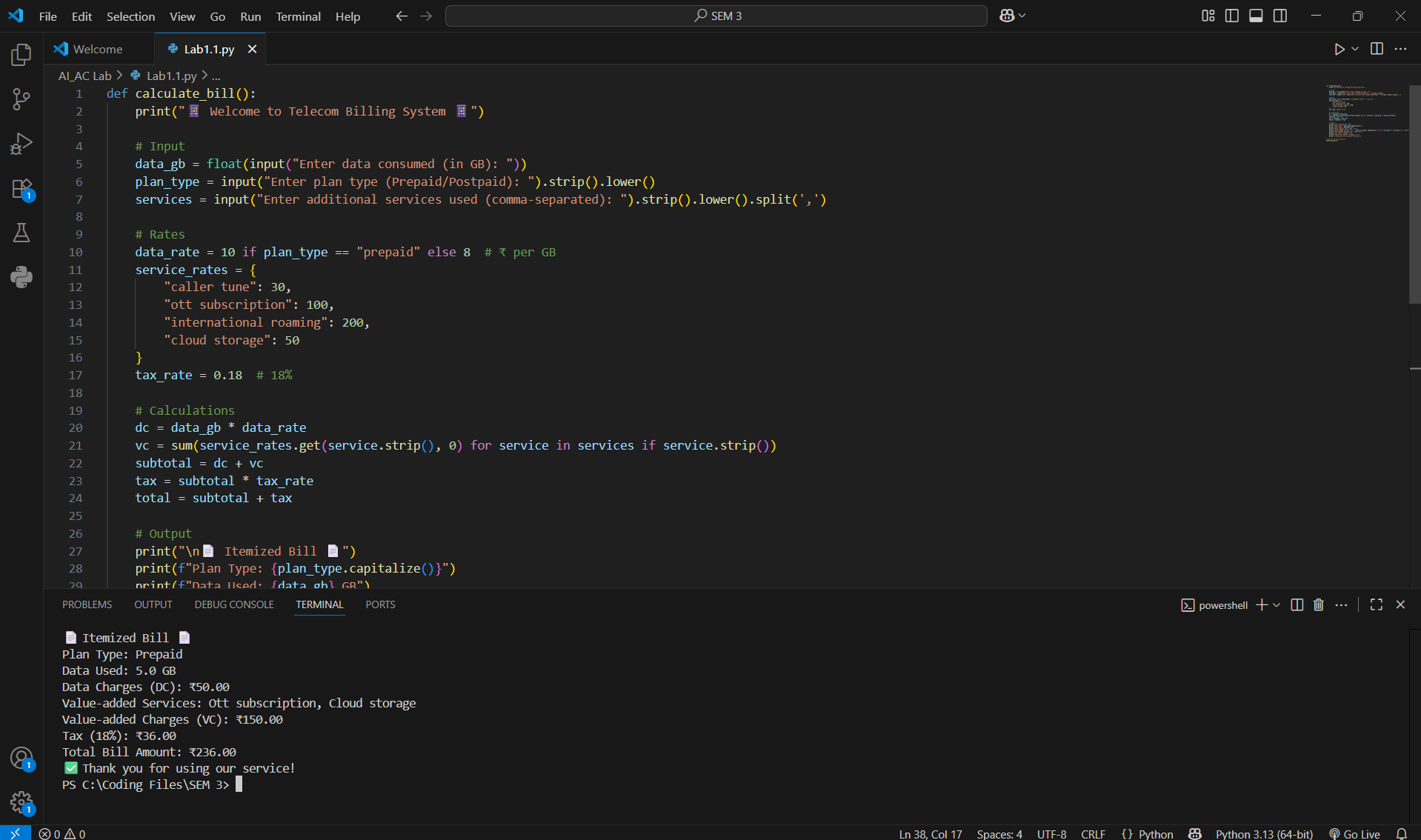
print(f"Total Bill Amount: ₹{total:.2f}")

print("✅ Thank you for using our service!")

# Run the billing function

calculate\_bill()

**Output screenshot:**



## Task 3: LPG Billing System

Objective: Build a Python program to calculate LPG bills using AI-assisted coding tools. Inputs include Cylinder Type, Number of Cylinders, Subsidy (for domestic), and Delivery charges. Program should display itemized bill with all details.

• AI was prompted to generate a program based on given cylinder prices and billing rules.

• Program reads user inputs and applies the billing formula:

Bill Amount = (Price per Cylinder × Quantity) - Subsidy + Delivery Charges

• Output displays Cylinder Type, Quantity, Base Amount, Subsidy, Delivery Charges, and Total.

• A line-by-line explanation of AI-generated code was also documented in simple terms.

**ChatGpt’s code:**

# LPG Billing System

def calculate\_lpg\_bill(cylinder\_type, quantity, subsidy, delivery\_charges):

# Price list

prices = {

"Domestic 14.2 kg": 905.00,

"Domestic 5 kg": 335.50,

"Commercial 19 kg": 1886.50,

"Commercial 47.5 kg": 4712.00

}

# Get base price per cylinder

if cylinder\_type not in prices:

return None

price\_per\_cylinder = prices[cylinder\_type]

base\_amount = price\_per\_cylinder \* quantity

# Subsidy only for domestic cylinders

if "Domestic" in cylinder\_type:

total = base\_amount - subsidy + delivery\_charges

else:

total = base\_amount + delivery\_charges

# Itemized bill

bill = {

"Cylinder Type": cylinder\_type,

"Quantity": quantity,

"Price per Cylinder": price\_per\_cylinder,

"Base Amount": base\_amount,

"Subsidy": subsidy if "Domestic" in cylinder\_type else 0,

"Delivery Charges": delivery\_charges,

"Total Bill": total

}

return bill

# --- User Inputs ---

print("LPG Billing System\n")

print("Choose Cylinder Type:")

print("1. Domestic 14.2 kg")

print("2. Domestic 5 kg")

print("3. Commercial 19 kg")

print("4. Commercial 47.5 kg")

choice = int(input("Enter your choice (1-4): "))

cylinder\_map = {

1: "Domestic 14.2 kg",

2: "Domestic 5 kg",

3: "Commercial 19 kg",

4: "Commercial 47.5 kg"

}

cylinder\_type = cylinder\_map.get(choice, None)

quantity = int(input("Enter number of cylinders: "))

subsidy = 0

# Subsidy only if Domestic

if "Domestic" in cylinder\_type:

subsidy = float(input("Enter subsidy amount: "))

delivery\_charges = float(input("Enter delivery charges (₹10 - ₹50): "))

# --- Calculation ---

bill = calculate\_lpg\_bill(cylinder\_type, quantity, subsidy, delivery\_charges)

# --- Output ---

if bill:

print("\n------- LPG Bill -------")

print(f"Cylinder Type : {bill['Cylinder Type']}")

print(f"Quantity : {bill['Quantity']}")

print(f"Price per Cylinder: ₹{bill['Price per Cylinder']}")

print(f"Base Amount : ₹{bill['Base Amount']}")

print(f"Subsidy : ₹{bill['Subsidy']}")

print(f"Delivery Charges : ₹{bill['Delivery Charges']}")

print(f"Total Bill Amount : ₹{bill['Total Bill']}")

print("------------------------")

else:

print("Invalid Cylinder Type Selected.")

**Output Screenshot:**

