

# Supermarket Billing System Code

```
import requests
import smtplib
from email.message import EmailMessage
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas

def fetch_supermarket_data():
    """Fetch supermarket data from the API."""
    try:
        url = "http://demo3278802.mockable.io/Super_Market_Data"
        response = requests.get(url)

        if response.status_code == 200:
            return response.json()
        else:
            print("Failed to retrieve data. Status code:", response.status_code)
            return None
    except Exception as e:
        print(f"Error fetching data: {e}")
        return None

def calculate_bill(price, quantity):
    """Calculate total price and GST."""
    total_price = price * quantity
    gst = (total_price * 5) / 100
    final_price = total_price + gst
    return total_price, gst, final_price

def send_email_bill(pdf_filename, email):
    """Send the bill PDF via email."""
    try:
        sender_email = "your_email@gmail.com"
        receiver_email = email
        password = "your_password"

        msg = EmailMessage()
        msg["Subject"] = "Supermarket Bill Invoice"
        msg["From"] = sender_email
        msg["To"] = receiver_email
        msg.set_content("Please find your bill attached.")

        with open(pdf_filename, "rb") as f:
            msg.add_attachment(f.read(), maintype="application", subtype="pdf", filename=pdf_filename)

        with smtplib.SMTP_SSL("smtp.gmail.com", 465) as server:
            server.login(sender_email, password)
            server.send_message(msg)

        print("Email sent successfully!")
    except Exception as e:
        print(f"Error sending email: {e}")

def print_bill_to_file(total_price, gst, final_price):
    """Save the bill to a file and print it."""
    try:
        with open("customer_bill.csv", "w") as file:
            file.write(f"Total Price,GST,Final Price\n{total_price},{gst},{final_price}")

        print("Bill saved to customer_bill.csv")
        with open("customer_bill.csv", "r") as file:
            print(file.read())
    except Exception as e:
        print(f"Error handling the bill file: {e}")

def generate_pdf_bill(total_price, gst, final_price):
    """Generate a PDF bill."""
    pdf_file = "customer_bill.pdf"
    c = canvas.Canvas(pdf_file, pagesize=letter)
    c.setFont("Helvetica-Bold", 14)
```

```

c.drawString(200, 750, "Supermarket Bill Invoice")

c.setFont("Helvetica", 12)
c.drawString(100, 700, f"Total Price: {total_price}")
c.drawString(100, 680, f"GST: {gst}")
c.drawString(100, 660, f"Final Price: {final_price}")

c.save()
print("Bill saved as PDF:", pdf_file)

def supermarket():
    """Main function for the supermarket billing system."""
    try:
        data = fetch_supermarket_data()
        if not data:
            return

        fruit_name = data["Fruitname"]
        price = data["Price"]

        quantity = int(input(f"How much quantity of {fruit_name} do you want? "))
        total_price, gst, final_price = calculate_bill(price, quantity)

        print(f"Total Price: {total_price}, GST: {gst}, Final Price: {final_price}")

        bill_method = int(input("Do you want an email bill (press 1), printed bill (press 2), or PDF bill (press 3)?"))

        if bill_method == 1:
            email = input("Enter your email: ")
            send_email_bill("customer_bill.pdf", email)
        elif bill_method == 2:
            print_bill_to_file(total_price, gst, final_price)
        elif bill_method == 3:
            generate_pdf_bill(total_price, gst, final_price)
        else:
            print("Invalid input")

    except Exception as e:
        print(f"Error: {e}")

supermarket()

```