**1. Title of Project:**

Food Ordering System

Jonathan Gomes(9900)

Shwen Coutinho(9881)

Mark Lopes(9913)

**2. Database Used-Type:**

Database Type: SQL (PostgreSQL)

**3. Tables with Structure and RDBMS Concepts Used:**

**Tables**:-

customer table:

customer\_id (Primary Key)

name

email

password

address

product table:

product\_id (Primary Key)

name

price

orders table:

order\_id (Primary Key)

customer\_id (Foreign Key referencing customer.customer\_id)

order\_date

order\_items table:

order\_item\_id (Primary Key)

order\_id (Foreign Key referencing orders.order\_id)

product\_id (Foreign Key referencing product.product\_id)

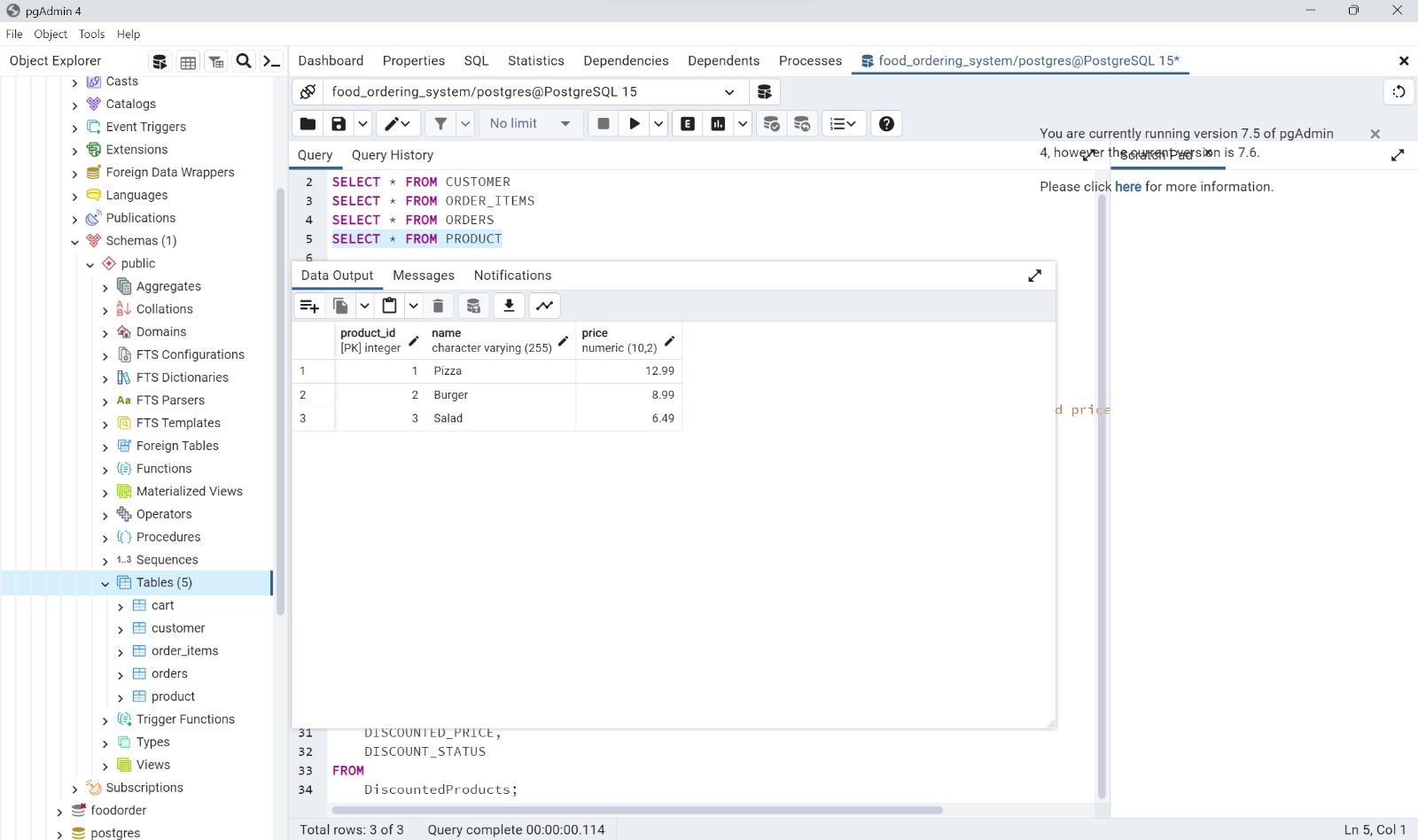
quantity

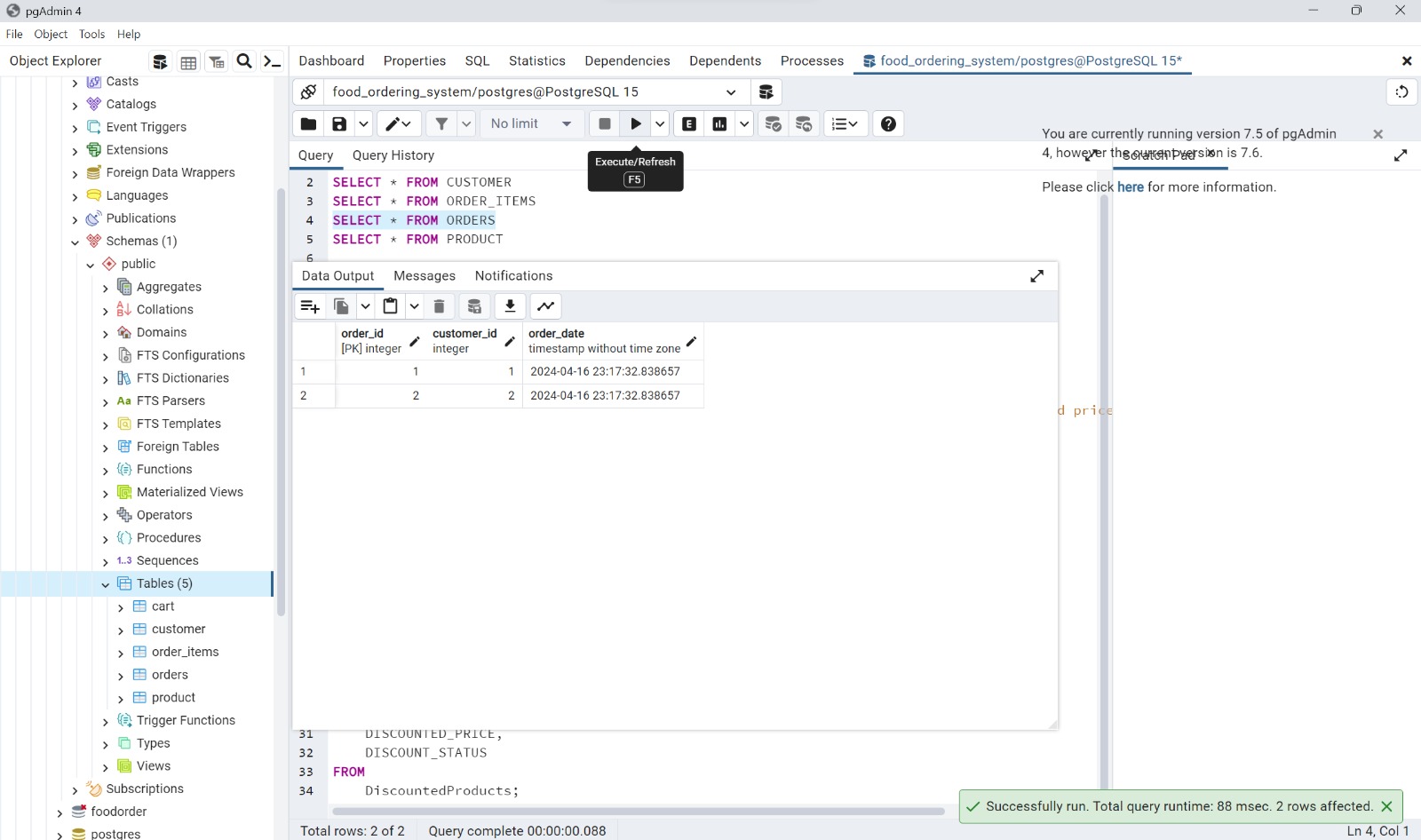
**RDBMS Concepts Used**:-

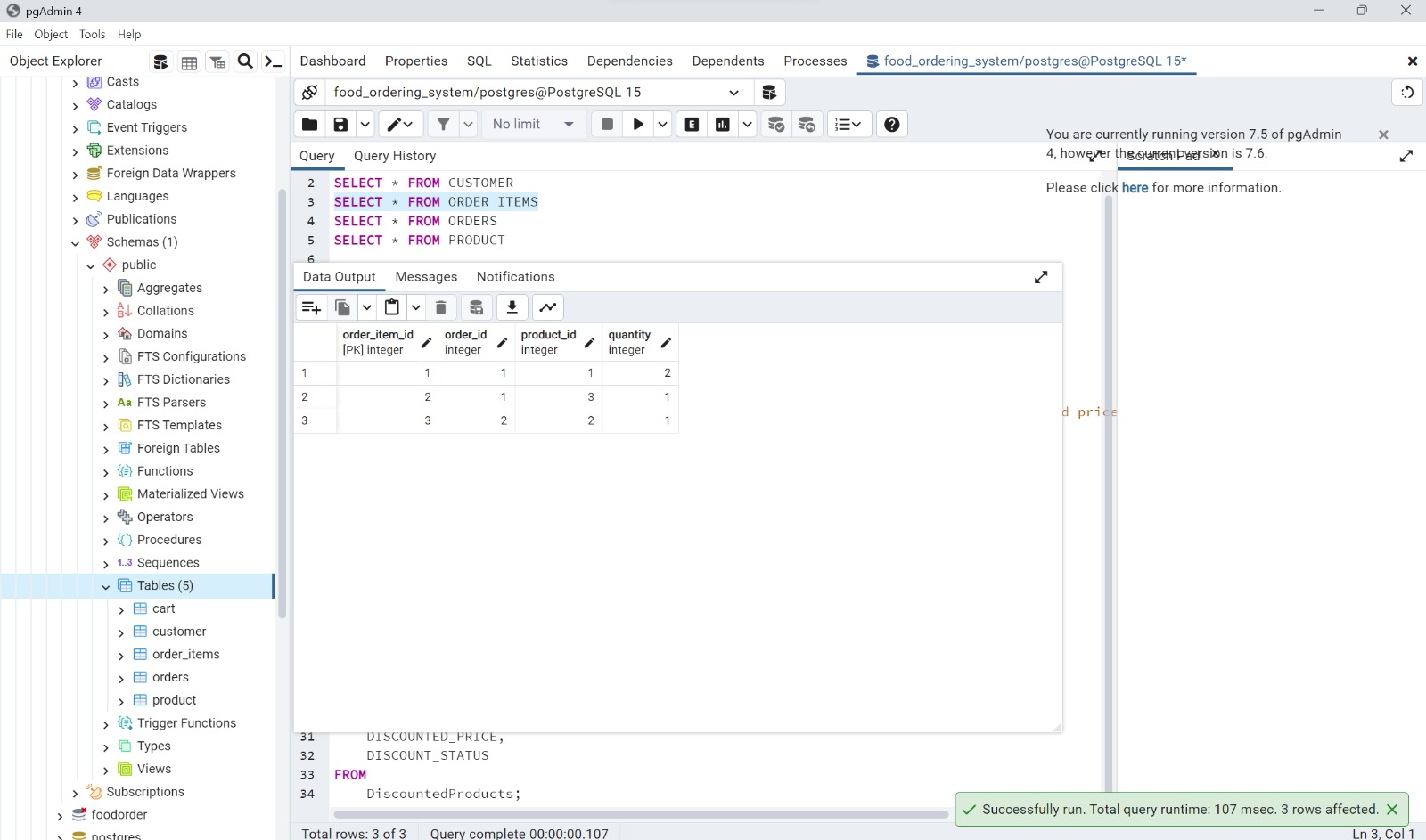
Primary Key: Unique identifier for each record in a table.

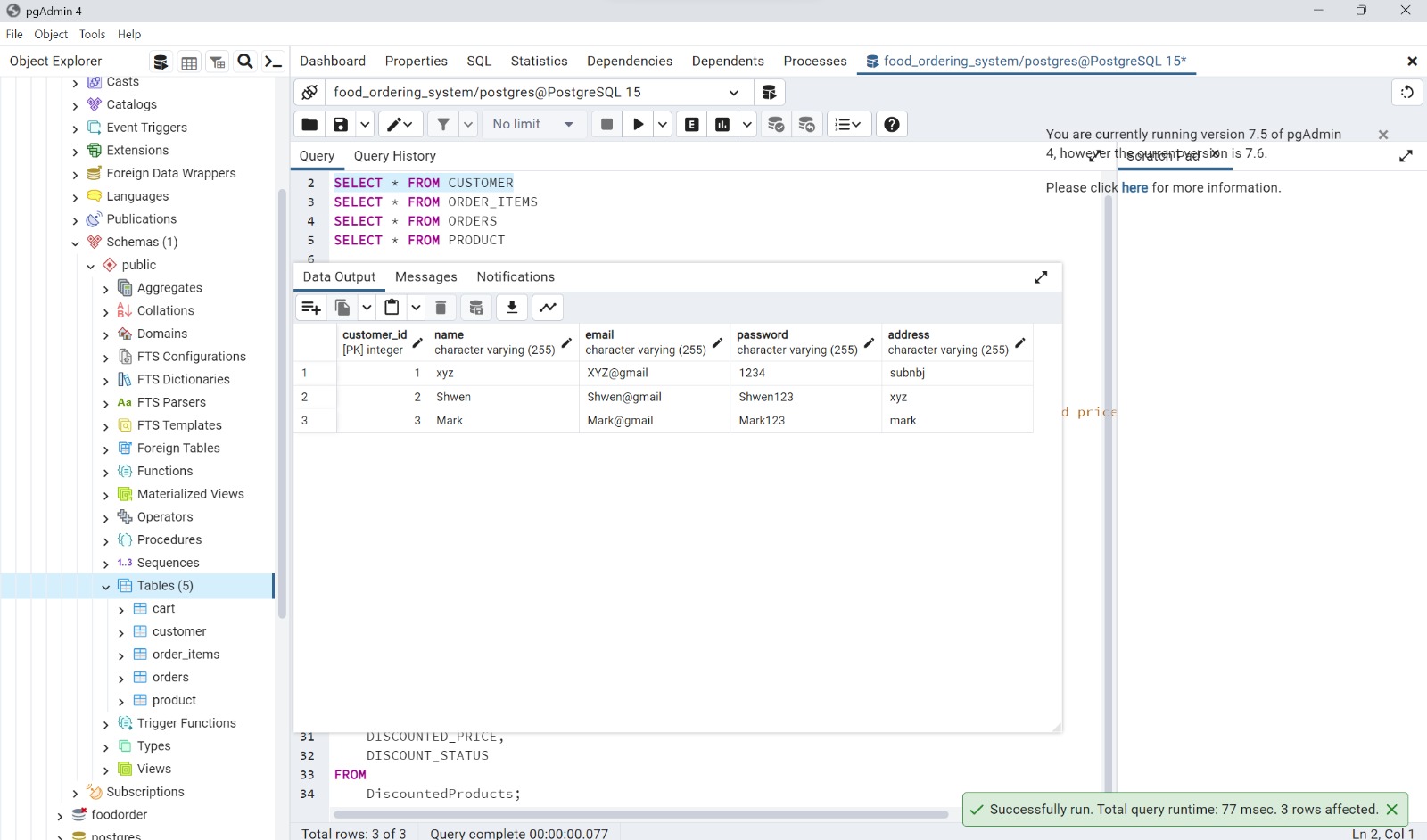
Foreign Key: Establishes relationships between tables (orders.customer\_id referencing customer.customer\_id and order\_items.order\_id referencing orders.order\_id).

**4. CRUD operations screenshots:**

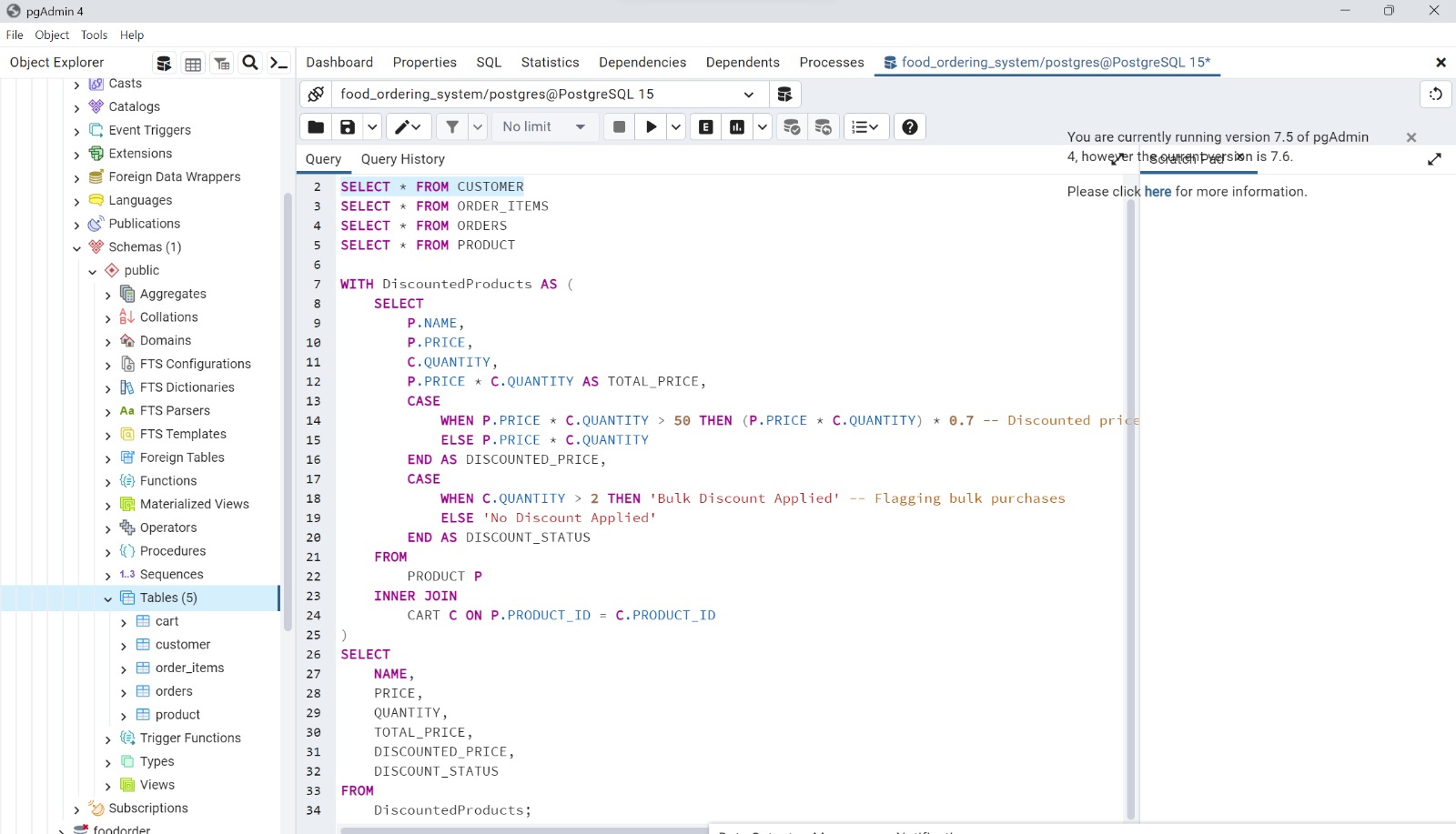








**5. Use of Advanced SQL (Code):**



**6. GUI-DB Connectivity Code (Flask):**

from flask import Flask, render\_template, request, redirect, url\_for

import psycopg2

from psycopg2 import Error

app = Flask(\_\_name\_\_)

# PostgreSQL connection parameters

DB\_HOST = "localhost"

DB\_NAME = "food\_ordering\_system"

DB\_USER = "postgres"

DB\_PASSWORD = "Mphasis123"

def connect\_to\_db():

    try:

        connection = psycopg2.connect(

            user=DB\_USER,

            password=DB\_PASSWORD,

            host=DB\_HOST,

            database=DB\_NAME

        )

        return connection

    except Error as e:

        print(f"Error connecting to PostgreSQL database: {e}")

        return None

@app.route('/', methods=['GET', 'POST'])

def login():

    if request.method == 'POST':

        email = request.form['email']

        password = request.form['password']

        # Connect to PostgreSQL database

        connection = connect\_to\_db()

        if connection:

            try:

                cursor = connection.cursor()

                # Check if user with given email and password exists

                cursor.execute("SELECT customer\_id, name FROM customer WHERE email = %s AND password = %s", (email, password))

                user = cursor.fetchone()

                if user:

                    # Store user\_id in session (or use Flask-Login for session management)

                    customer\_id = user[0]

                    return redirect(url\_for('select\_items', customer\_id=customer\_id))

                else:

                    return "Invalid credentials. Please try again."

                cursor.close()

                connection.close()

            except Error as e:

                print(f"Error querying database: {e}")

                return f"Error: {e}"

        else:

            return "Failed to connect to database. Please try again later."

    return render\_template('login.html')

@app.route('/select\_items/<int:customer\_id>', methods=['GET', 'POST'])

def select\_items(customer\_id):

    if request.method == 'POST':

        product\_id = int(request.form['product\_id'])

        quantity = int(request.form['quantity'])

        # Connect to PostgreSQL database

        connection = connect\_to\_db()

        if connection:

            try:

                cursor = connection.cursor()

                # Add item to cart

                cursor.execute("INSERT INTO cart (customer\_id, product\_id, quantity) VALUES (%s, %s, %s)", (customer\_id, product\_id, quantity))

                connection.commit()

                cursor.close()

                connection.close()

                return "Item added to cart!"

            except Error as e:

                print(f"Error inserting data into cart: {e}")

                return f"Error: {e}"

        else:

            return "Failed to connect to database. Please try again later."

    # Fetch products from database to display in the form

    connection = connect\_to\_db()

    if connection:

        try:

            cursor = connection.cursor()

            # Retrieve products from product table

            cursor.execute("SELECT product\_id, name, price FROM product")

            products = cursor.fetchall()

            cursor.close()

            connection.close()

            return render\_template('select\_items.html', products=products, customer\_id=customer\_id)

        except Error as e:

            print(f"Error retrieving products: {e}")

            return f"Error: {e}"

    else:

        return "Failed to connect to database. Please try again later."

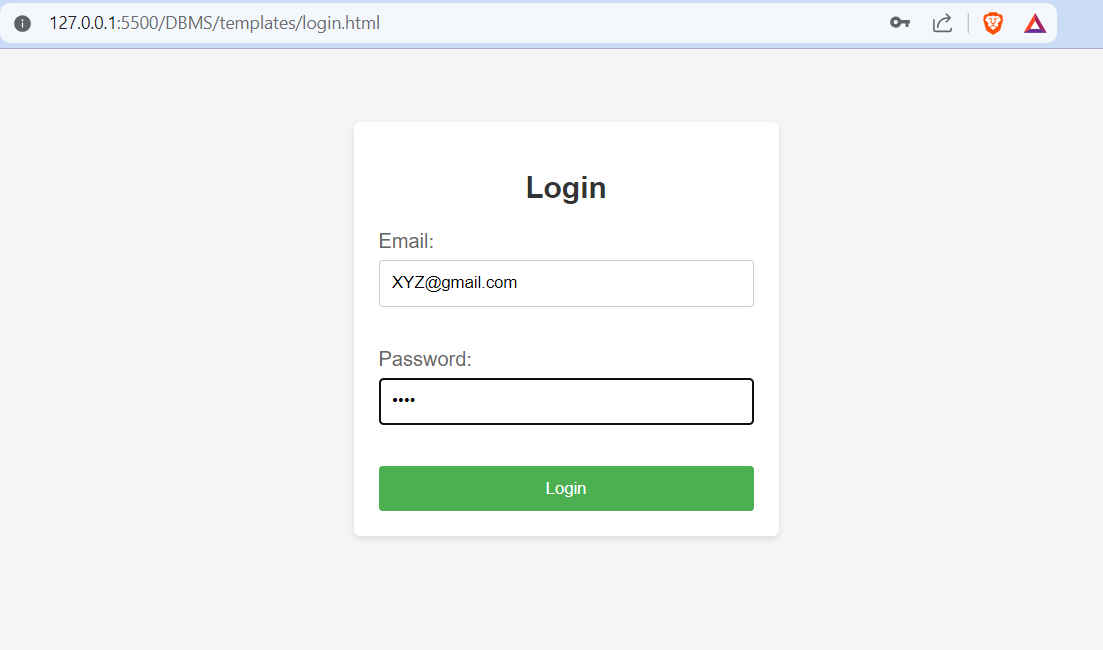
if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**7. Screenshots Demonstration:**

Login page:-

Here we can login using the emails and the corresponding passwords in the customer table



Select Order page:-

A screenshot of a computer screen

Description automatically generatedSql query:-

Here the customer can select their order and add it to cart, the order\_items table will be simultaneously changed in sql.

After this we can run our sql query to get us the price of all the orders. If the price is above 50 a 30% discount will be applied to the total cost

