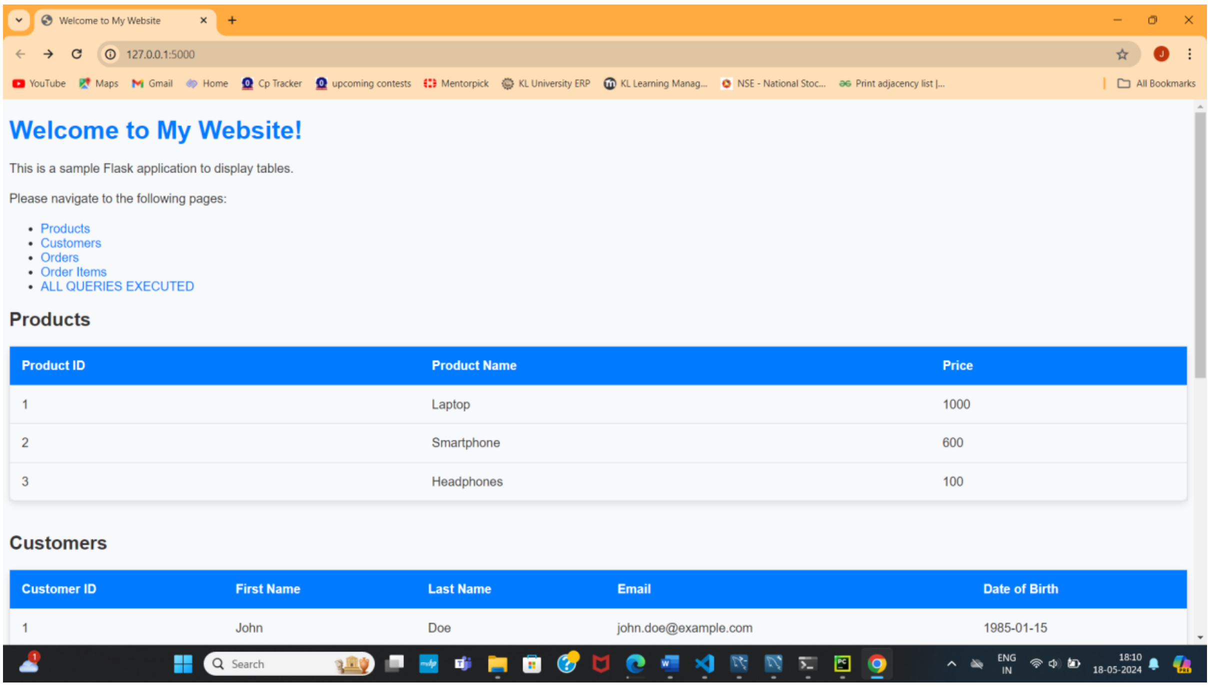
SQL BACK END TASK :

Outputs of Backend and Frontend:

FRONT END OUTPUT :



Welcome to My Website!

This is a sample Flask application to display tables.

Please navigate to the following pages:

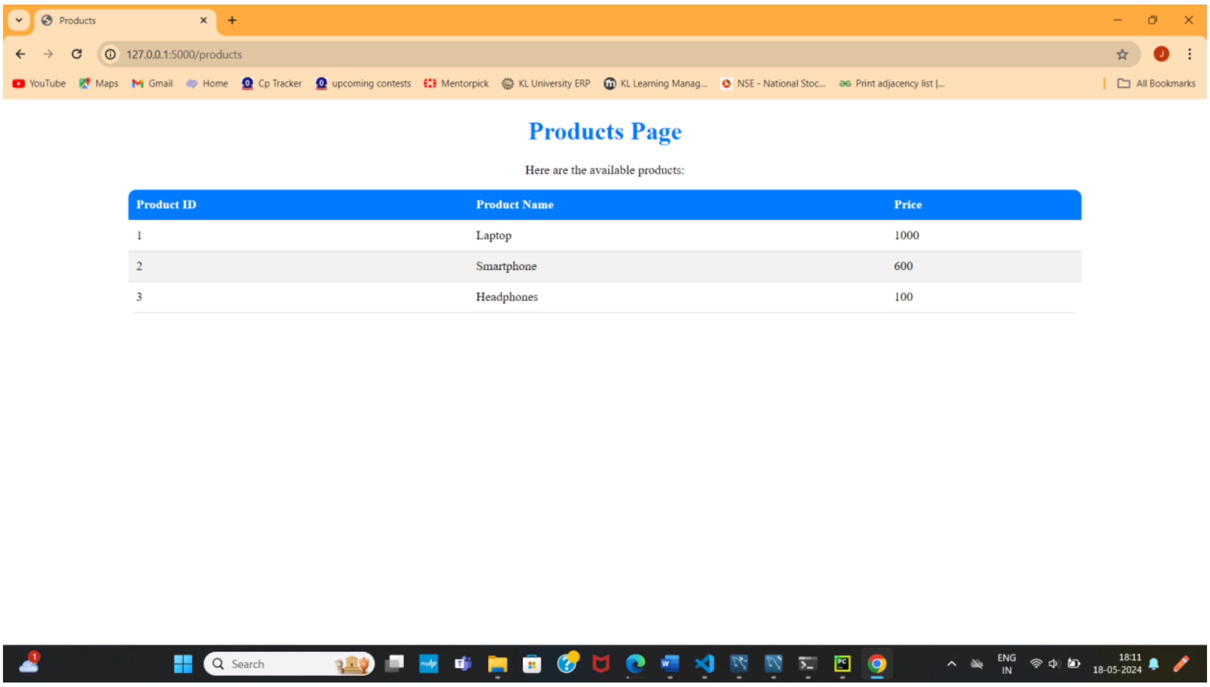
- Products
- Customers
- Orders
- Order Items
- ALL QUERIES EXECUTED

Products

Product ID	Product Name	Price
1	Laptop	1000
2	Smartphone	600
3	Headphones	100

Customers

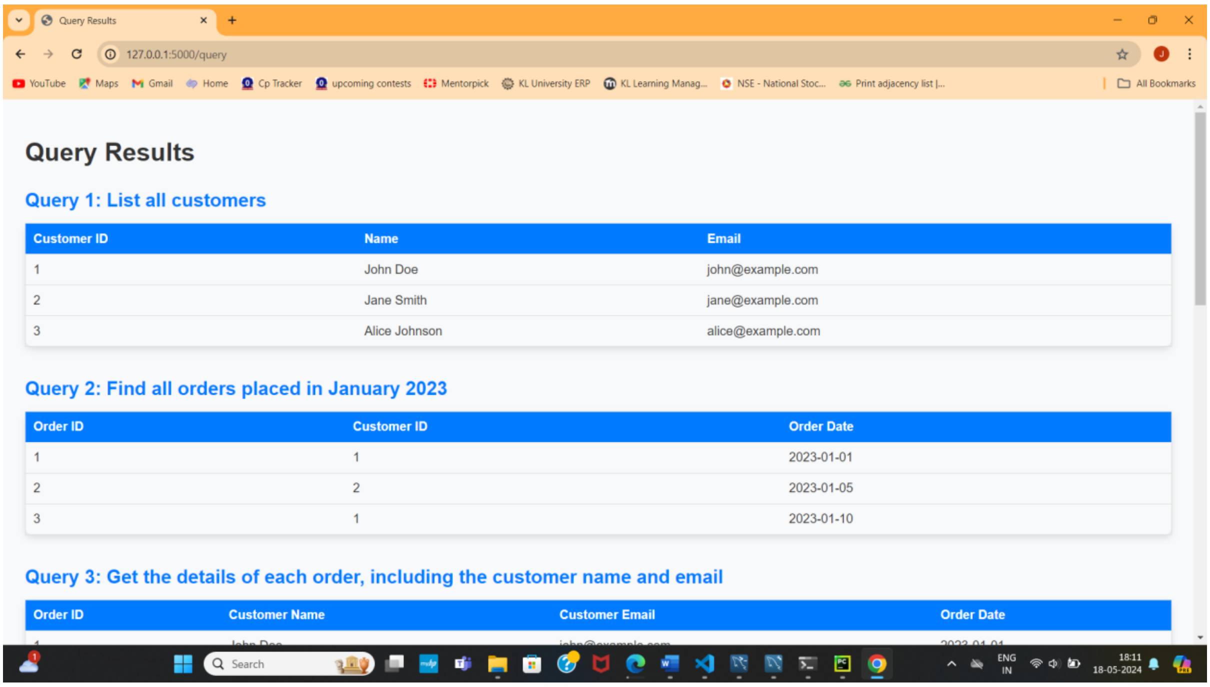
Customer ID	First Name	Last Name	Email	Date of Birth
1	John	Doe	john.doe@example.com	1985-01-15



Products Page

Here are the available products:

Product ID	Product Name	Price
1	Laptop	1000
2	Smartphone	600
3	Headphones	100



Query Results

Query 1: List all customers

Customer ID	Name	Email
1	John Doe	john@example.com
2	Jane Smith	jane@example.com
3	Alice Johnson	alice@example.com

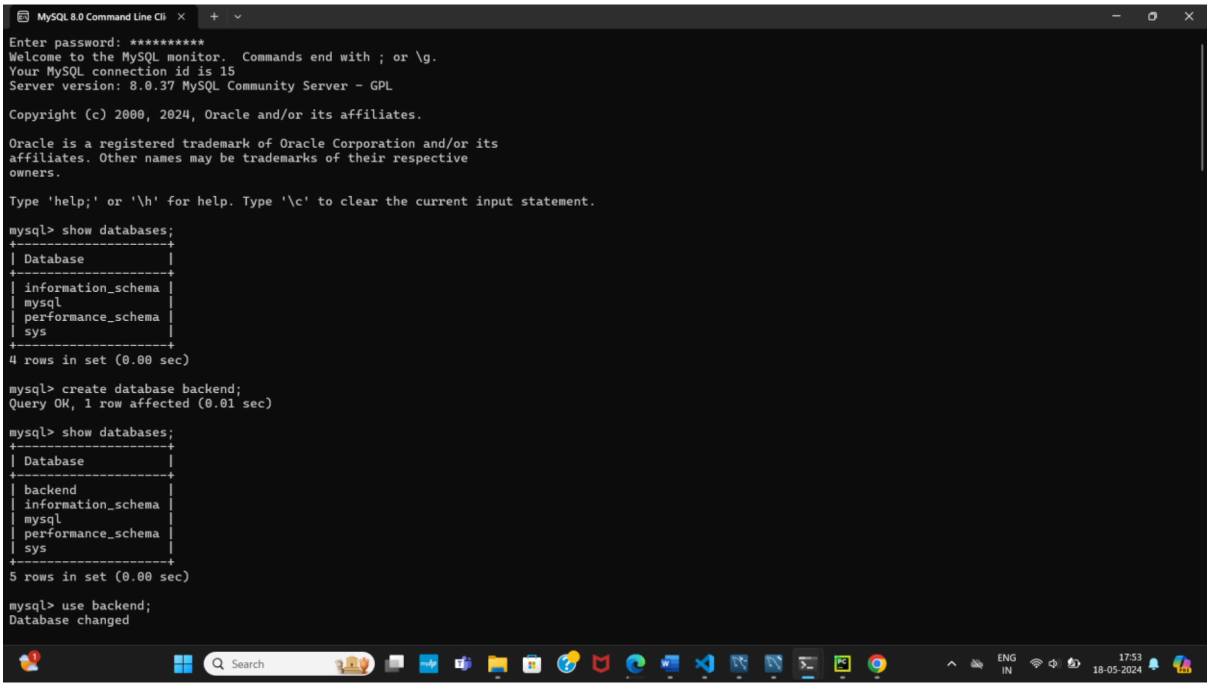
Query 2: Find all orders placed in January 2023

Order ID	Customer ID	Order Date
1	1	2023-01-01
2	2	2023-01-05
3	1	2023-01-10

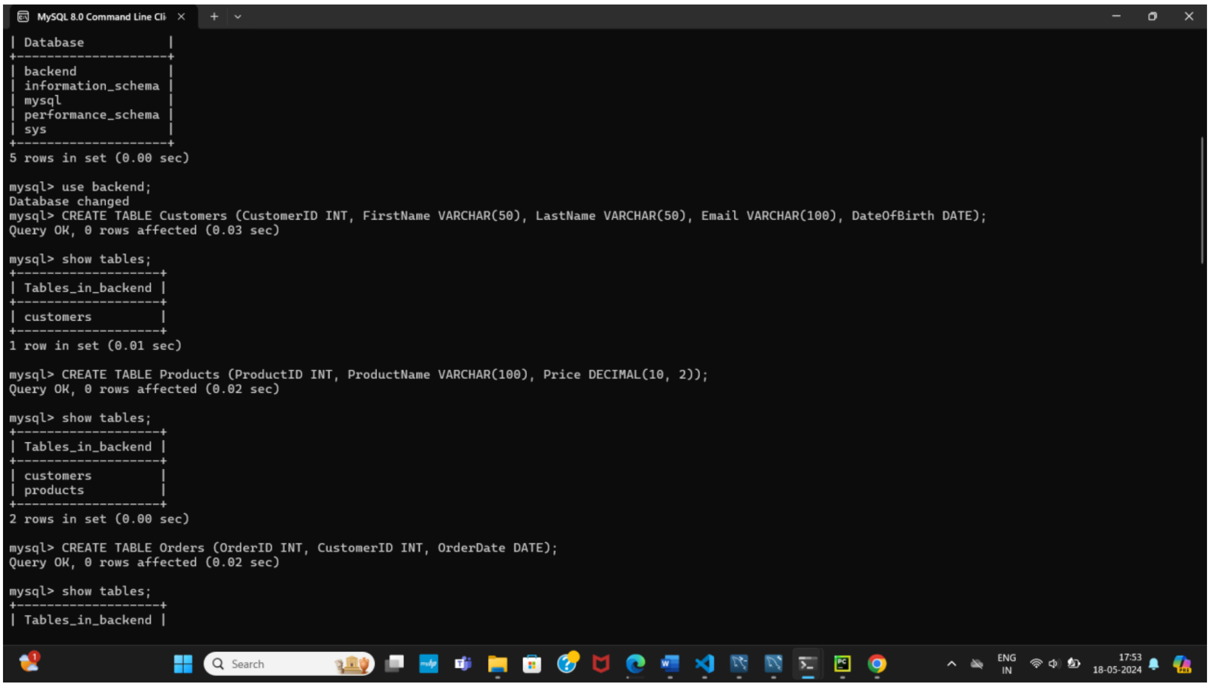
Query 3: Get the details of each order, including the customer name and email

Order ID	Customer Name	Customer Email	Order Date
1	John Doe	john@example.com	2023-01-01

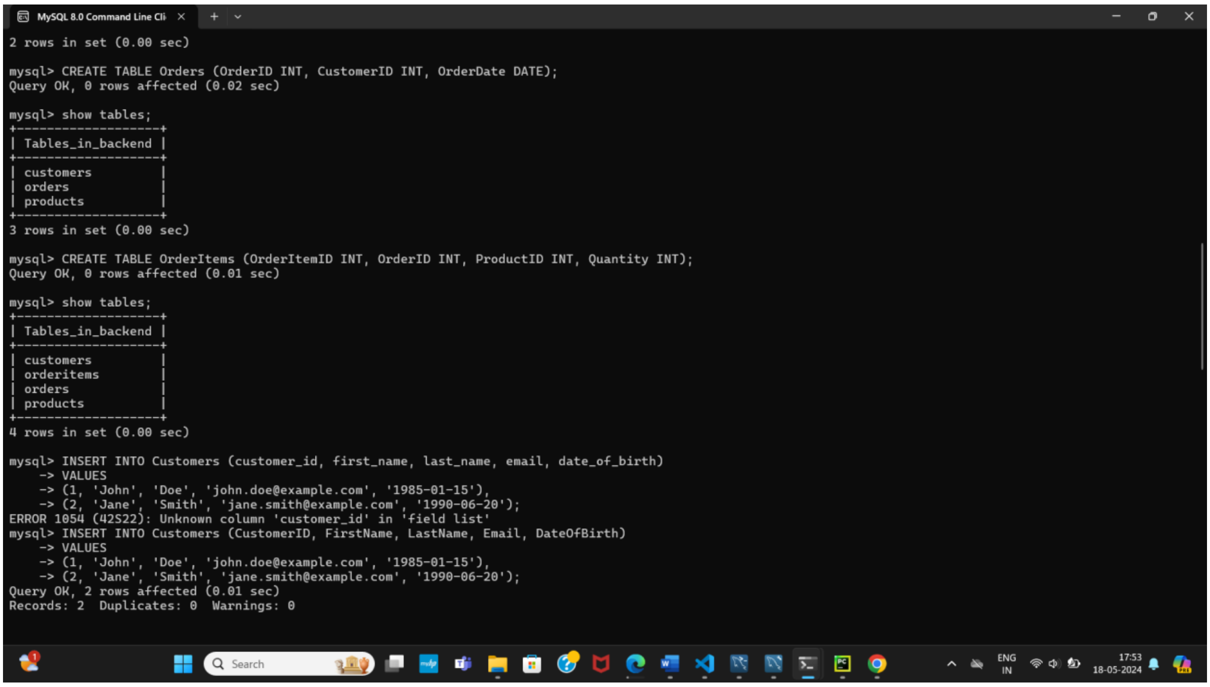
SQL BACKEND OUTPUTS:



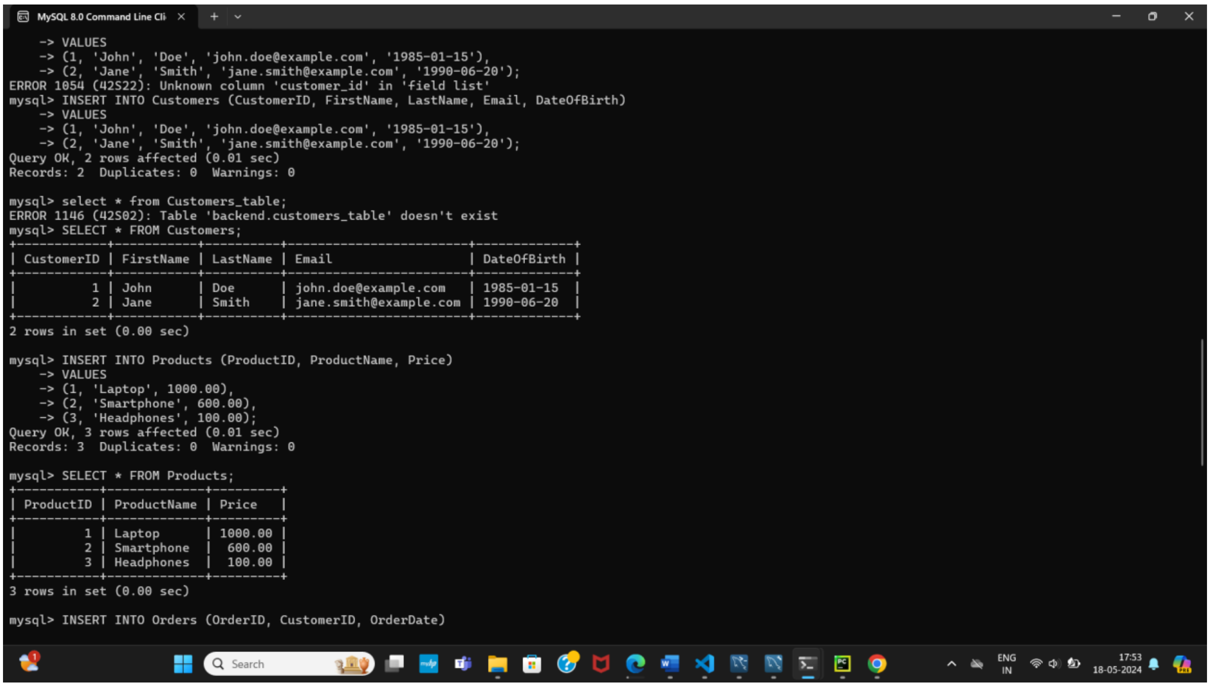
```
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> create database backend;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| backend            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use backend;
Database changed
```



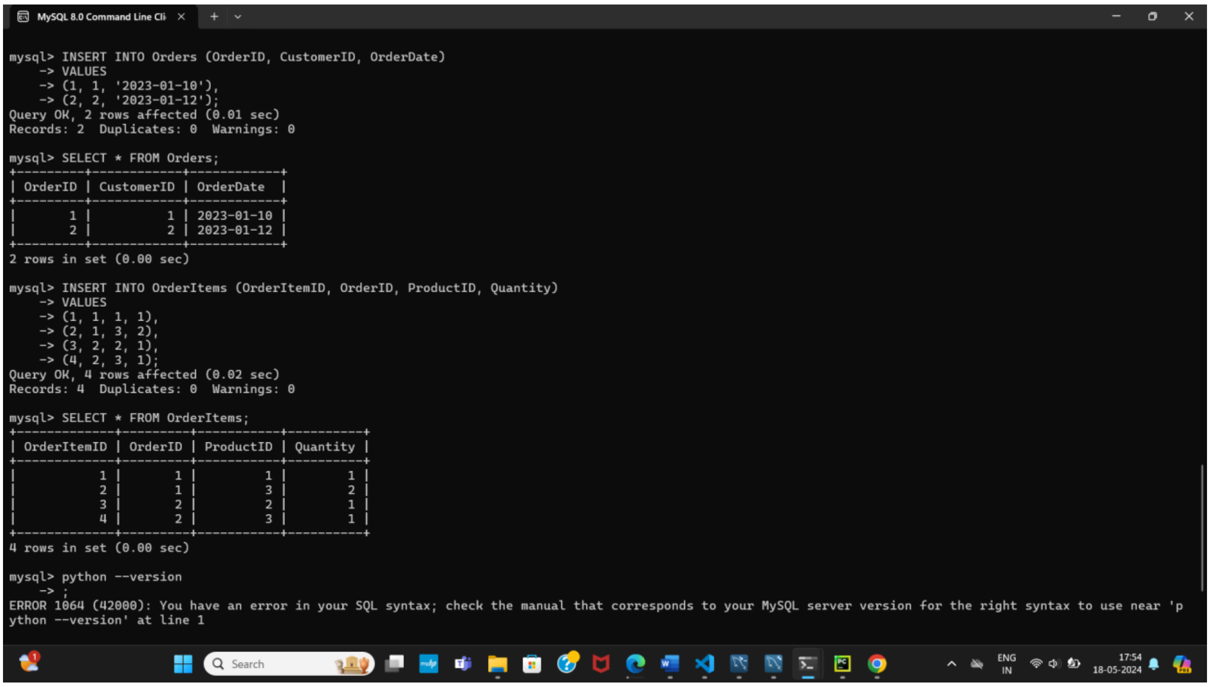
```
| Database           |
+--------------------+
| backend            |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use backend;
Database changed
mysql> CREATE TABLE Customers (CustomerID INT, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), DateOfBirth DATE);
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-------------------+
| Tables_in_backend |
+-------------------+
| customers         |
+-------------------+
1 row in set (0.01 sec)

mysql> CREATE TABLE Products (ProductID INT, ProductName VARCHAR(100), Price DECIMAL(10, 2));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-------------------+
| Tables_in_backend |
+-------------------+
| customers         |
| products          |
+-------------------+
2 rows in set (0.00 sec)

mysql> CREATE TABLE Orders (OrderID INT, CustomerID INT, OrderDate DATE);
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-------------------+
| Tables_in_backend |
```



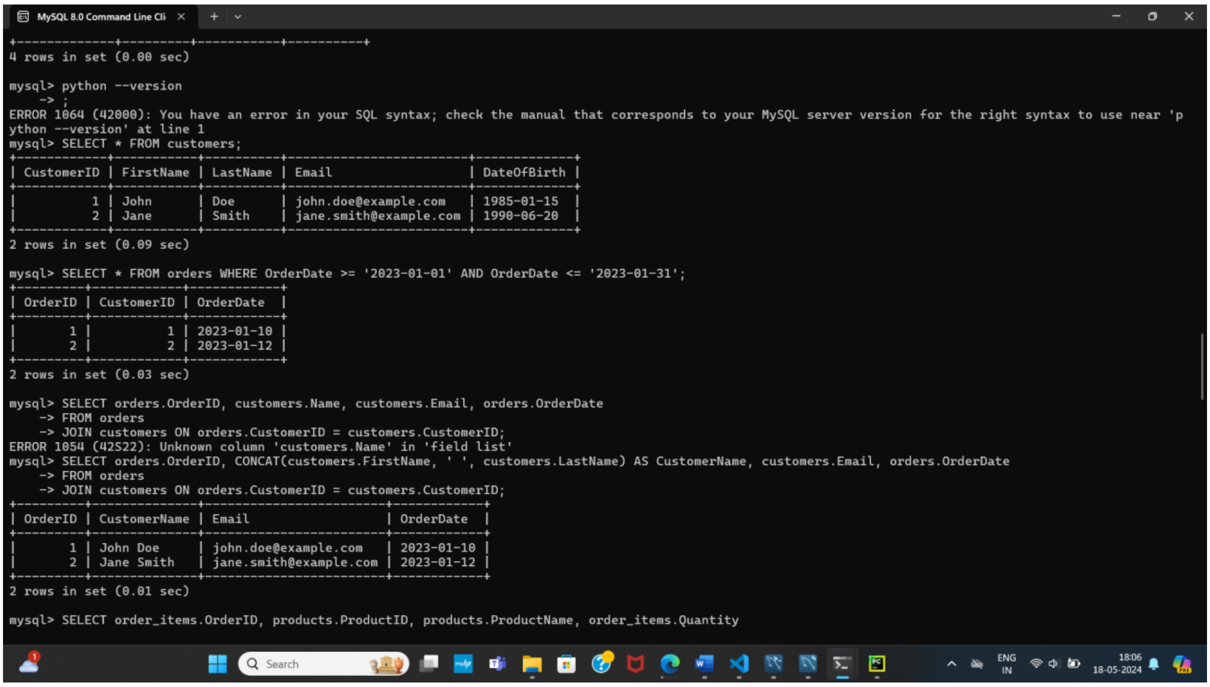
```
2 rows in set (0.00 sec)

mysql> CREATE TABLE Orders (OrderID INT, CustomerID INT, OrderDate DATE);
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-------------------+
| Tables_in_backend |
+-------------------+
| customers         |
| orders            |
| products          |
+-------------------+
3 rows in set (0.00 sec)

mysql> CREATE TABLE OrderItems (OrderItemID INT, OrderID INT, ProductID INT, Quantity INT);
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-------------------+
| Tables_in_backend |
+-------------------+
| customers         |
| orderitems        |
| orders            |
| products          |
+-------------------+
4 rows in set (0.00 sec)

mysql> INSERT INTO Customers (customer_id, first_name, last_name, email, date_of_birth)
    -> VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
ERROR 1054 (42S22): Unknown column 'customer_id' in 'field list'
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth)
    -> VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```



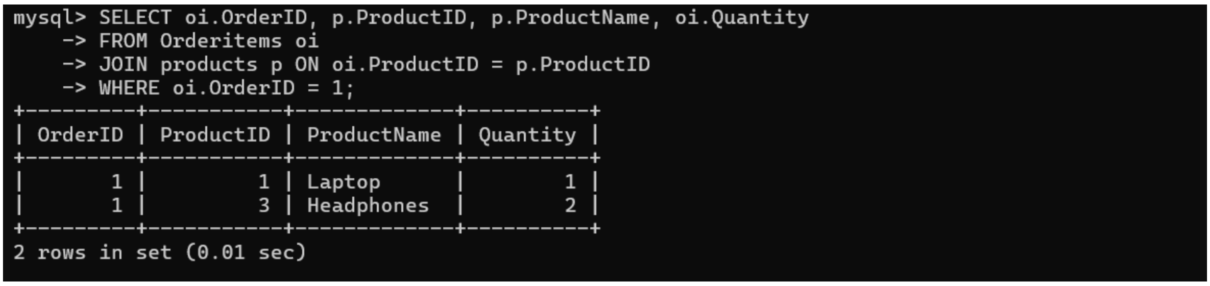
```
    -> VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
ERROR 1054 (42S22): Unknown column 'customer_id' in 'field list'
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth)
    -> VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from Customers_table;
ERROR 1146 (42S02): Table 'backend.customers_table' doesn't exist
mysql> SELECT * FROM Customers;
+------------+-----------+----------+------------------------+-------------+
| CustomerID | FirstName | LastName | Email                  | DateOfBirth |
+------------+-----------+----------+------------------------+-------------+
|          1 | John      | Doe      | john.doe@example.com   | 1985-01-15  |
|          2 | Jane      | Smith    | jane.smith@example.com | 1990-06-20  |
+------------+-----------+----------+------------------------+-------------+
2 rows in set (0.00 sec)

mysql> INSERT INTO Products (ProductID, ProductName, Price)
    -> VALUES
    -> (1, 'Laptop', 1000.00),
    -> (2, 'Smartphone', 600.00),
    -> (3, 'Headphones', 100.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Products;
+-----------+-------------+---------+
| ProductID | ProductName | Price   |
+-----------+-------------+---------+
|         1 | Laptop      | 1000.00 |
|         2 | Smartphone  |  600.00 |
|         3 | Headphones  |  100.00 |
+-----------+-------------+---------+
3 rows in set (0.00 sec)

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate)
```



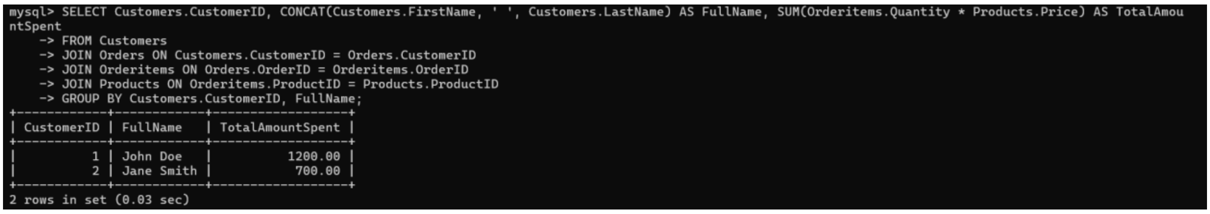
```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate)
    -> VALUES
    -> (1, 1, '2023-01-10'),
    -> (2, 2, '2023-01-12');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Orders;
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-10 |
|       2 |          2 | 2023-01-12 |
+---------+------------+------------+
2 rows in set (0.00 sec)

mysql> INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity)
    -> VALUES
    -> (1, 1, 1, 1),
    -> (2, 1, 3, 2),
    -> (3, 2, 2, 1),
    -> (4, 2, 3, 1);
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM OrderItems;
+-------------+---------+-----------+----------+
| OrderItemID | OrderID | ProductID | Quantity |
+-------------+---------+-----------+----------+
|           1 |       1 |         1 |        1 |
|           2 |       1 |         3 |        2 |
|           3 |       2 |         2 |        1 |
|           4 |       2 |         3 |        1 |
+-------------+---------+-----------+----------+
4 rows in set (0.00 sec)

mysql> python --version
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'p
ython --version' at line 1
```



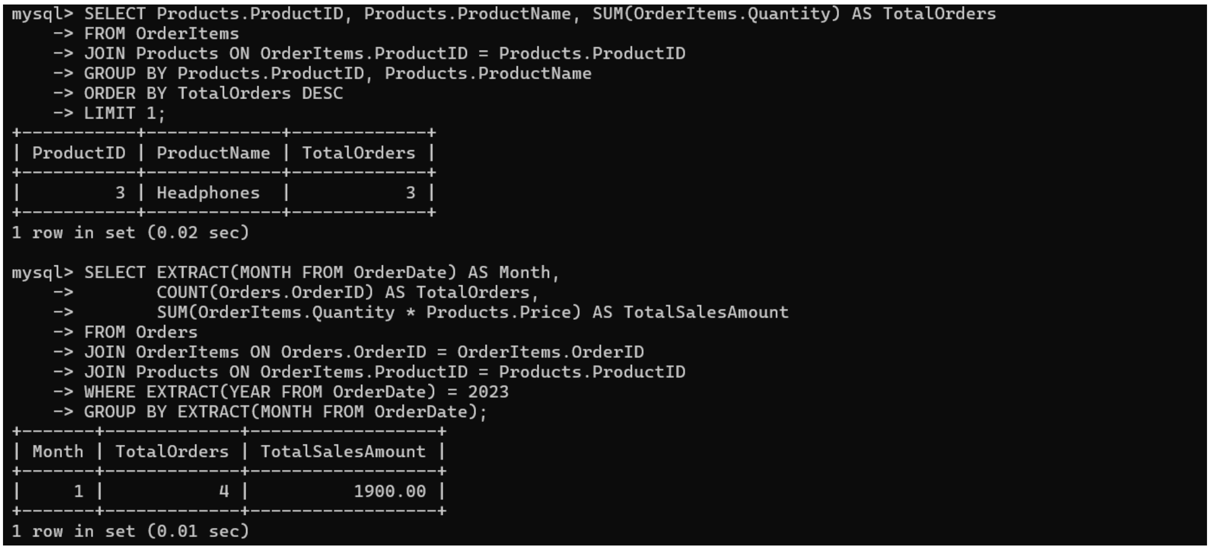
```
+-------------+---------+-----------+----------+
4 rows in set (0.00 sec)

mysql> python --version
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'p
ython --version' at line 1
mysql> SELECT * FROM customers;
+------------+-----------+----------+------------------------+-------------+
| CustomerID | FirstName | LastName | Email                  | DateOfBirth |
+------------+-----------+----------+------------------------+-------------+
|          1 | John      | Doe      | john.doe@example.com   | 1985-01-15  |
|          2 | Jane      | Smith    | jane.smith@example.com | 1990-06-20  |
+------------+-----------+----------+------------------------+-------------+
2 rows in set (0.09 sec)

mysql> SELECT * FROM orders WHERE OrderDate >= '2023-01-01' AND OrderDate <= '2023-01-31';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-10 |
|       2 |          2 | 2023-01-12 |
+---------+------------+------------+
2 rows in set (0.03 sec)

mysql> SELECT orders.OrderID, customers.Name, customers.Email, orders.OrderDate
    -> FROM orders
    -> JOIN customers ON orders.CustomerID = customers.CustomerID;
ERROR 1054 (42S22): Unknown column 'customers.Name' in 'field list'
mysql> SELECT orders.OrderID, CONCAT(customers.FirstName, ' ', customers.LastName) AS CustomerName, customers.Email, orders.OrderDate
    -> FROM orders
    -> JOIN customers ON orders.CustomerID = customers.CustomerID;
+---------+--------------+------------------------+------------+
| OrderID | CustomerName | Email                  | OrderDate  |
+---------+--------------+------------------------+------------+
|       1 | John Doe     | john.doe@example.com   | 2023-01-10 |
|       2 | Jane Smith   | jane.smith@example.com | 2023-01-12 |
+---------+--------------+------------------------+------------+
2 rows in set (0.01 sec)

mysql> SELECT order_items.OrderID, products.ProductID, products.ProductName, order_items.Quantity
```



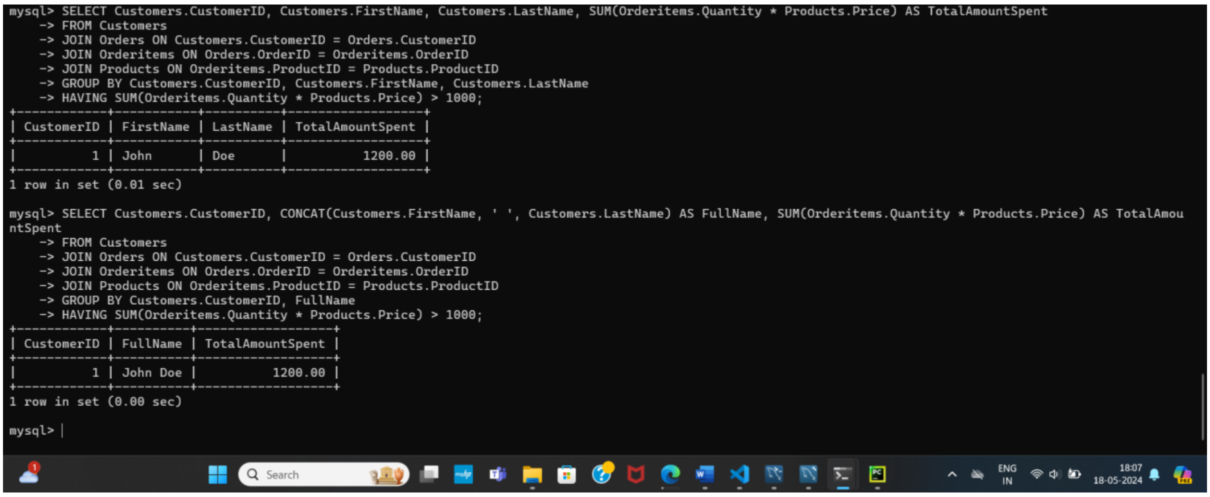
```
mysql> SELECT oi.OrderID, p.ProductID, p.ProductName, oi.Quantity
    -> FROM Orderitems oi
    -> JOIN products p ON oi.ProductID = p.ProductID
    -> WHERE oi.OrderID = 1;
+---------+-----------+-------------+----------+
| OrderID | ProductID | ProductName | Quantity |
+---------+-----------+-------------+----------+
|       1 |         1 | Laptop      |        1 |
|       1 |         3 | Headphones  |        2 |
+---------+-----------+-------------+----------+
2 rows in set (0.01 sec)
```



```
mysql> SELECT Customers.CustomerID, CONCAT(Customers.FirstName, ' ', Customers.LastName) AS FullName, SUM(Orderitems.Quantity * Products.Price) AS TotalAmou
ntSpent
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN Orderitems ON Orders.OrderID = Orderitems.OrderID
    -> JOIN Products ON Orderitems.ProductID = Products.ProductID
    -> GROUP BY Customers.CustomerID, FullName;
+------------+------------+------------------+
| CustomerID | FullName   | TotalAmountSpent |
+------------+------------+------------------+
|          1 | John Doe   |          1200.00 |
|          2 | Jane Smith |           700.00 |
+------------+------------+------------------+
2 rows in set (0.03 sec)
```



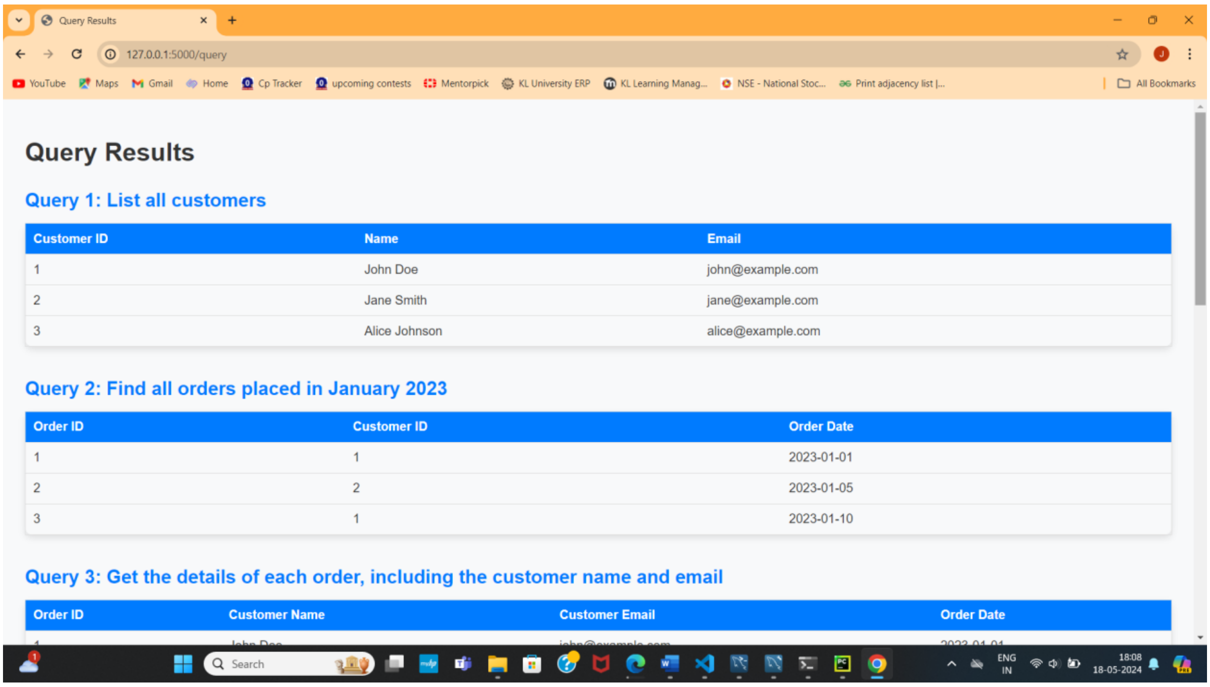
```
mysql> SELECT Products.ProductID, Products.ProductName, SUM(OrderItems.Quantity) AS TotalOrders
    -> FROM OrderItems
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> GROUP BY Products.ProductID, Products.ProductName
    -> ORDER BY TotalOrders DESC
    -> LIMIT 1;
+-----------+-------------+-------------+
| ProductID | ProductName | TotalOrders |
+-----------+-------------+-------------+
|         3 | Headphones  |           3 |
+-----------+-------------+-------------+
1 row in set (0.02 sec)

mysql> SELECT EXTRACT(MONTH FROM OrderDate) AS Month,
    ->        COUNT(Orders.OrderID) AS TotalOrders,
    ->        SUM(OrderItems.Quantity * Products.Price) AS TotalSalesAmount
    -> FROM Orders
    -> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> WHERE EXTRACT(YEAR FROM OrderDate) = 2023
    -> GROUP BY EXTRACT(MONTH FROM OrderDate);
+-------+-------------+------------------+
| Month | TotalOrders | TotalSalesAmount |
+-------+-------------+------------------+
|     1 |           4 |          1900.00 |
+-------+-------------+------------------+
1 row in set (0.01 sec)
```



```
mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Orderitems.Quantity * Products.Price) AS TotalAmountSpent
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN Orderitems ON Orders.OrderID = Orderitems.OrderID
    -> JOIN Products ON Orderitems.ProductID = Products.ProductID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> HAVING SUM(Orderitems.Quantity * Products.Price) > 1000;
+------------+-----------+----------+------------------+
| CustomerID | FirstName | LastName | TotalAmountSpent |
+------------+-----------+----------+------------------+
|          1 | John      | Doe      |          1200.00 |
+------------+-----------+----------+------------------+
1 row in set (0.01 sec)

mysql> SELECT Customers.CustomerID, CONCAT(Customers.FirstName, ' ', Customers.LastName) AS FullName, SUM(Orderitems.Quantity * Products.Price) AS TotalAmou
ntSpent
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN Orderitems ON Orders.OrderID = Orderitems.OrderID
    -> JOIN Products ON Orderitems.ProductID = Products.ProductID
    -> GROUP BY Customers.CustomerID, FullName
    -> HAVING SUM(Orderitems.Quantity * Products.Price) > 1000;
+------------+----------+------------------+
| CustomerID | FullName | TotalAmountSpent |
+------------+----------+------------------+
|          1 | John Doe |          1200.00 |
+------------+----------+------------------+
1 row in set (0.00 sec)

mysql> |
```

1. **List all customers**

2. Find all orders placed in January 2023.



Query Results

Query 1: List all customers

Customer ID	Name	Email
1	John Doe	john@example.com
2	Jane Smith	jane@example.com
3	Alice Johnson	alice@example.com

Query 2: Find all orders placed in January 2023

Order ID	Customer ID	Order Date
1	1	2023-01-01
2	2	2023-01-05
3	1	2023-01-10

Query 3: Get the details of each order, including the customer name and email

Order ID	Customer Name	Customer Email	Order Date
1	John Doe	john@example.com	2023-01-01

2. Find all orders placed in January 2023.

3. Get the details of each order, including the customer name and email.

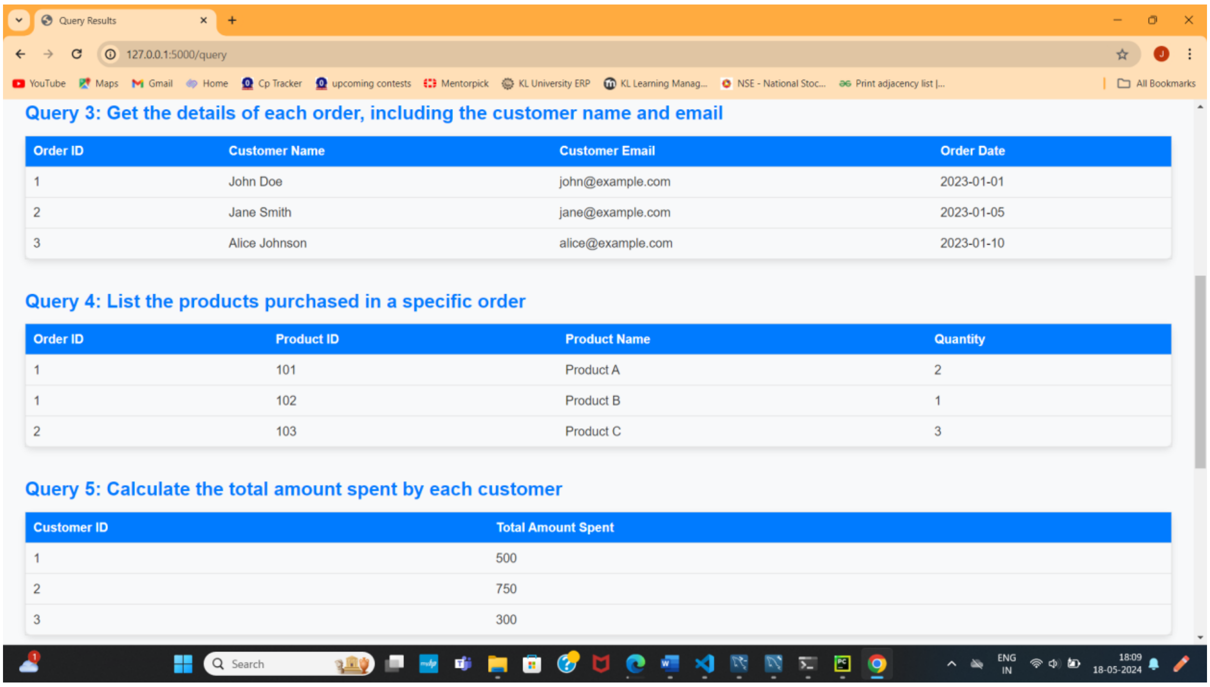
4. List the products purchased in a specific order (e.g., OrderID = 1).

5. Calculate the total amount spent by each customer.

6. Find the most popular product (the one that has been ordered the most).

7. Get the total number of orders and the total sales amount for each month in 2023.

8.Find customers who have spent more than $1000.



Query 3: Get the details of each order, including the customer name and email

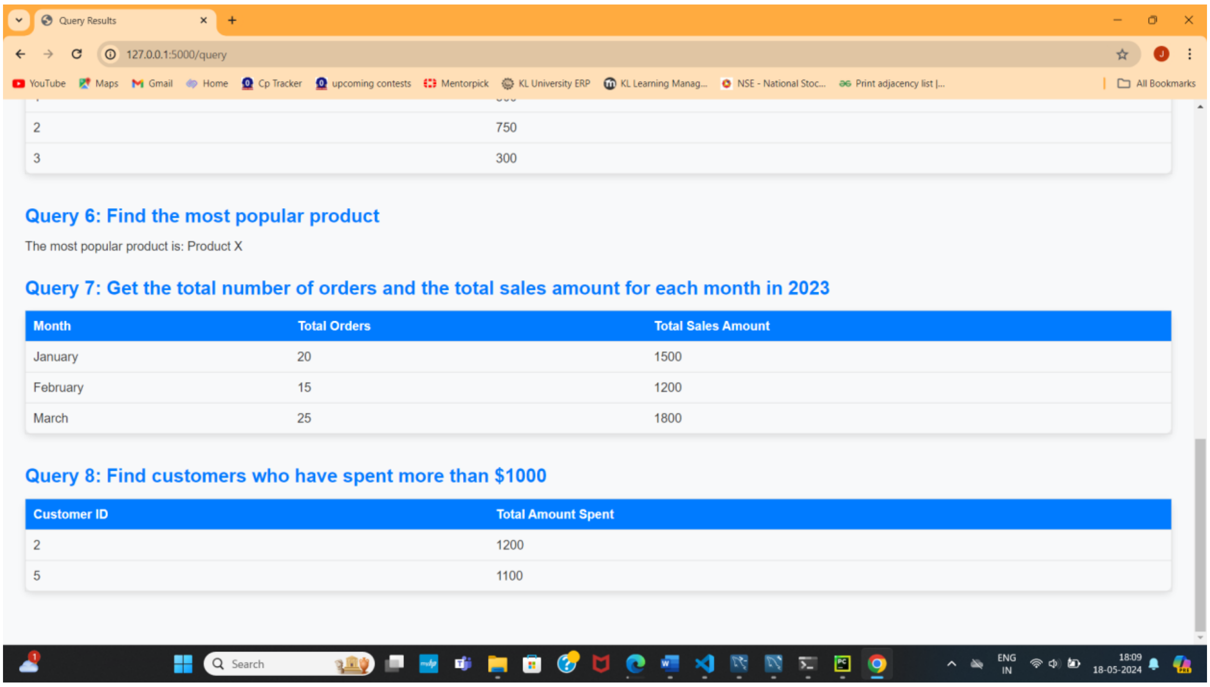
Order ID	Customer Name	Customer Email	Order Date
1	John Doe	john@example.com	2023-01-01
2	Jane Smith	jane@example.com	2023-01-05
3	Alice Johnson	alice@example.com	2023-01-10

Query 4: List the products purchased in a specific order

Order ID	Product ID	Product Name	Quantity
1	101	Product A	2
1	102	Product B	1
2	103	Product C	3

Query 5: Calculate the total amount spent by each customer

Customer ID	Total Amount Spent
1	500
2	750
3	300



2	750
3	300

Query 6: Find the most popular product

The most popular product is: Product X

Query 7: Get the total number of orders and the total sales amount for each month in 2023

Month	Total Orders	Total Sales Amount
January	20	1500
February	15	1200
March	25	1800

Query 8: Find customers who have spent more than $1000

Customer ID	Total Amount Spent
2	1200
5	1100