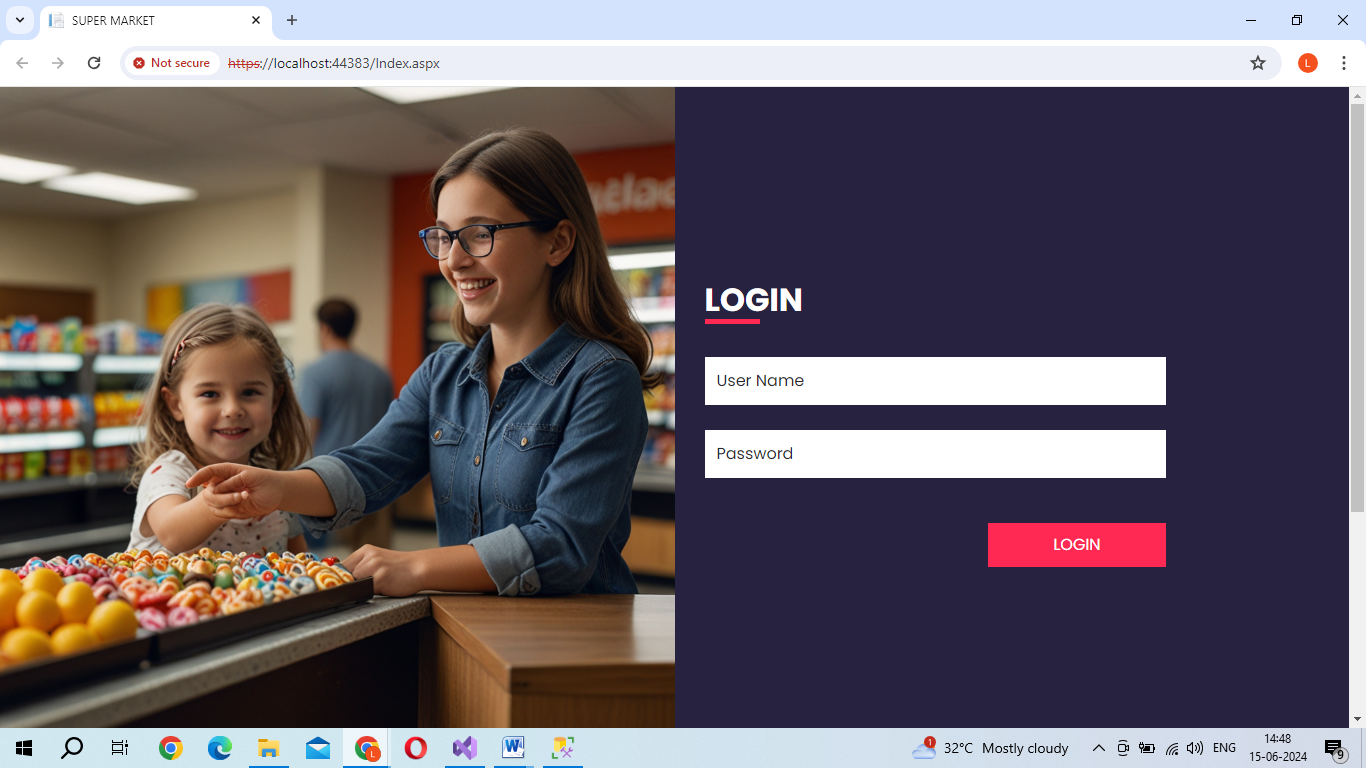
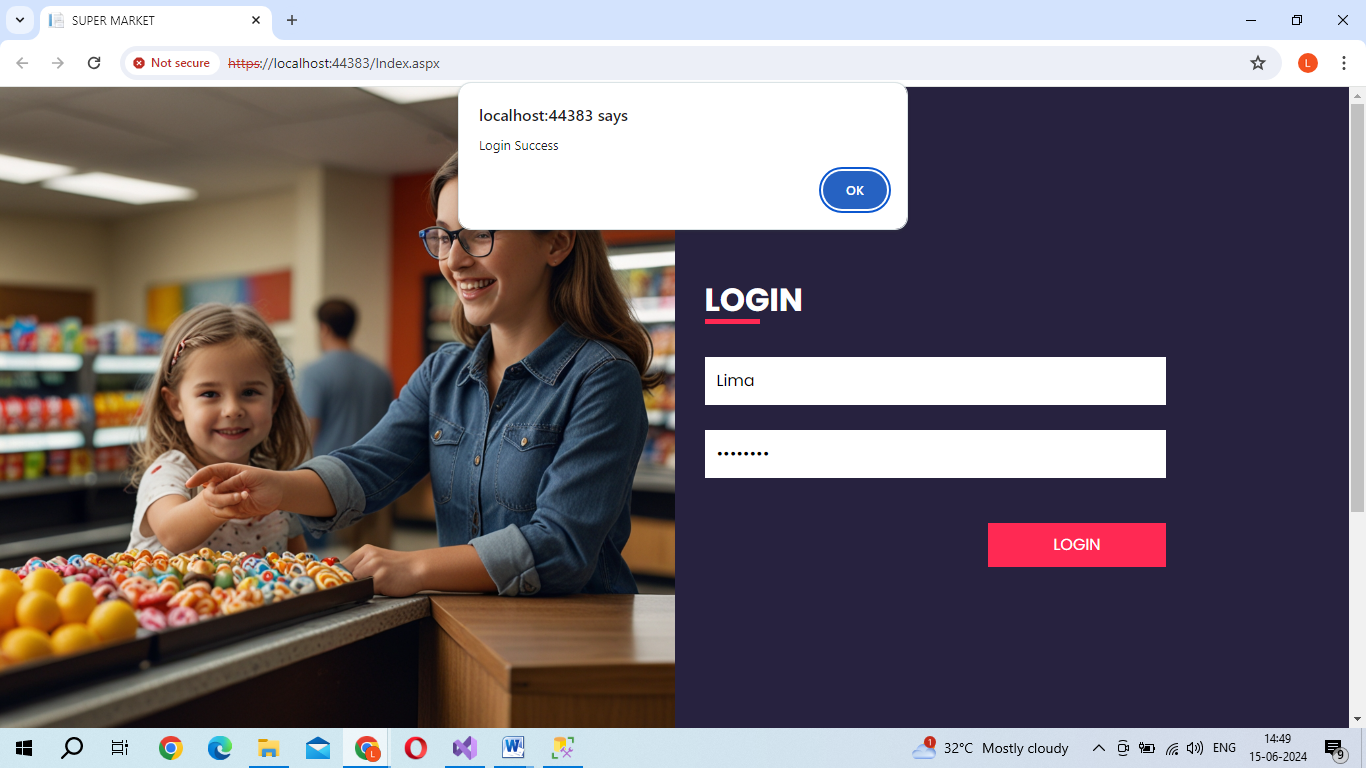
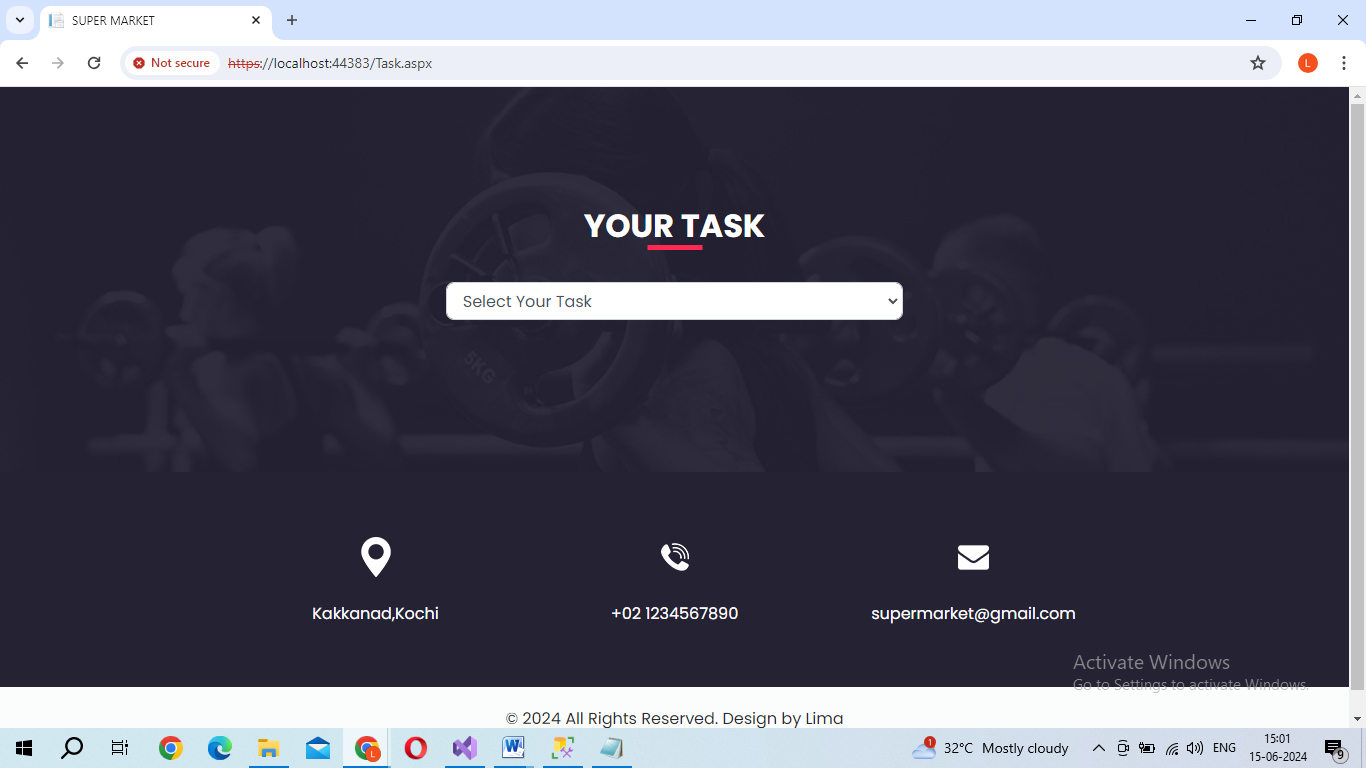
First we need to Login by using UserName and Password

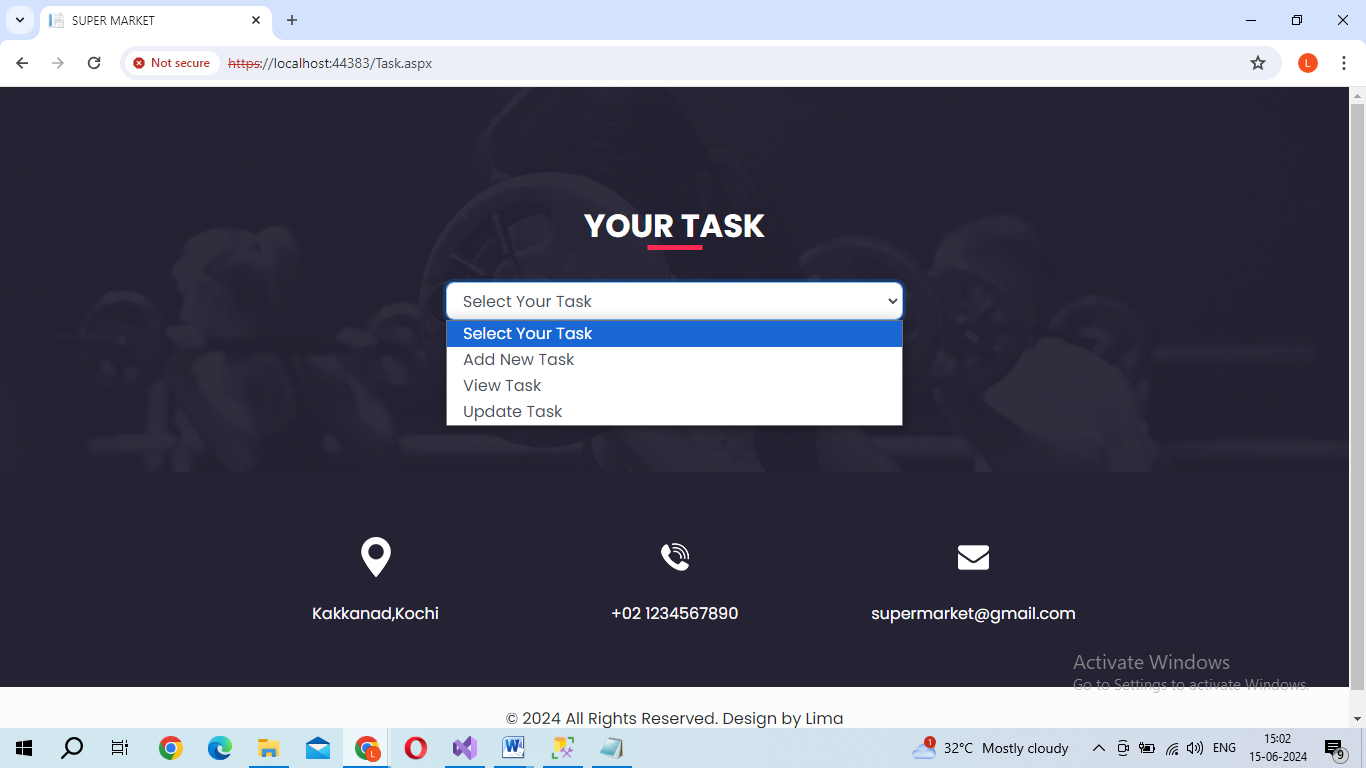


If the password and UserName is correct we are redirect to our task updation page.

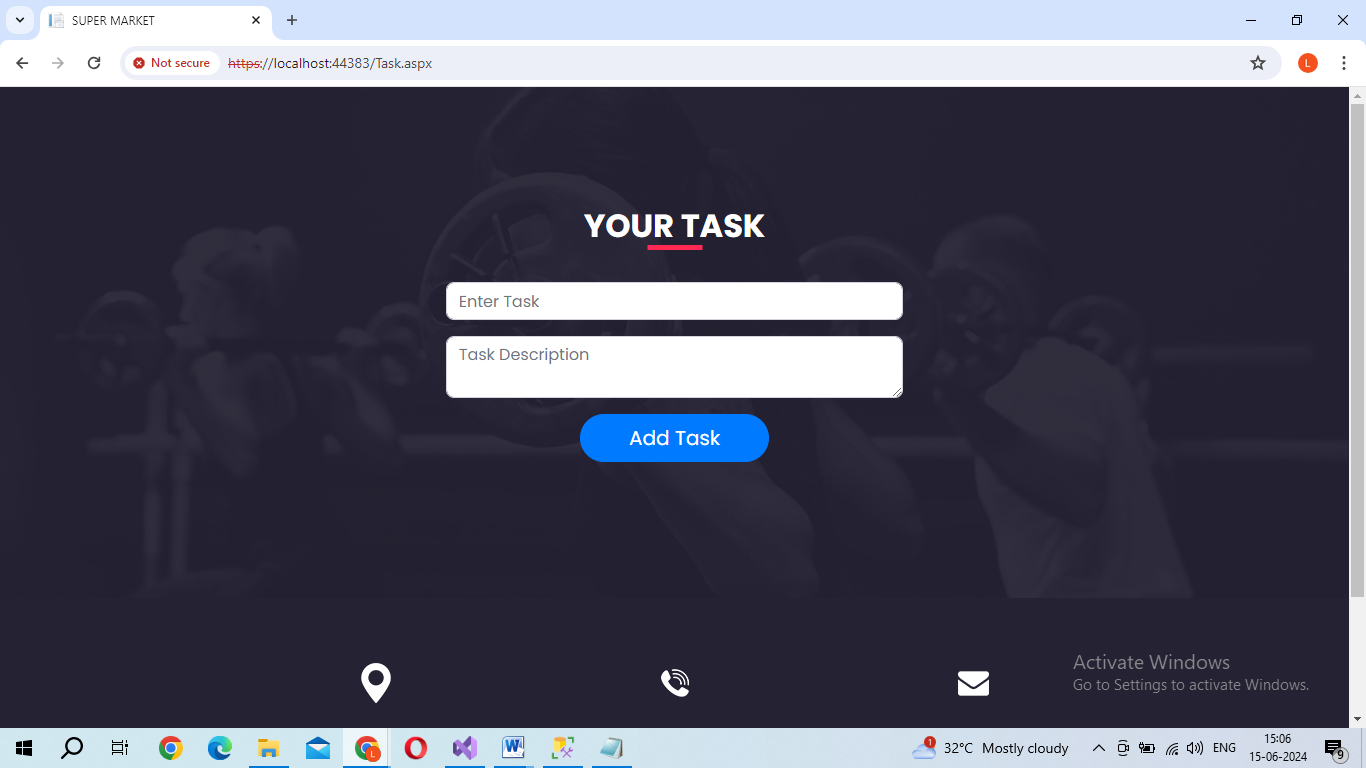


After the Login we have enter the task updation page. If we select “Add new Task” then we have an option to add new task





After the Task Name and Discription you should click “Add Button ” for adding the task.



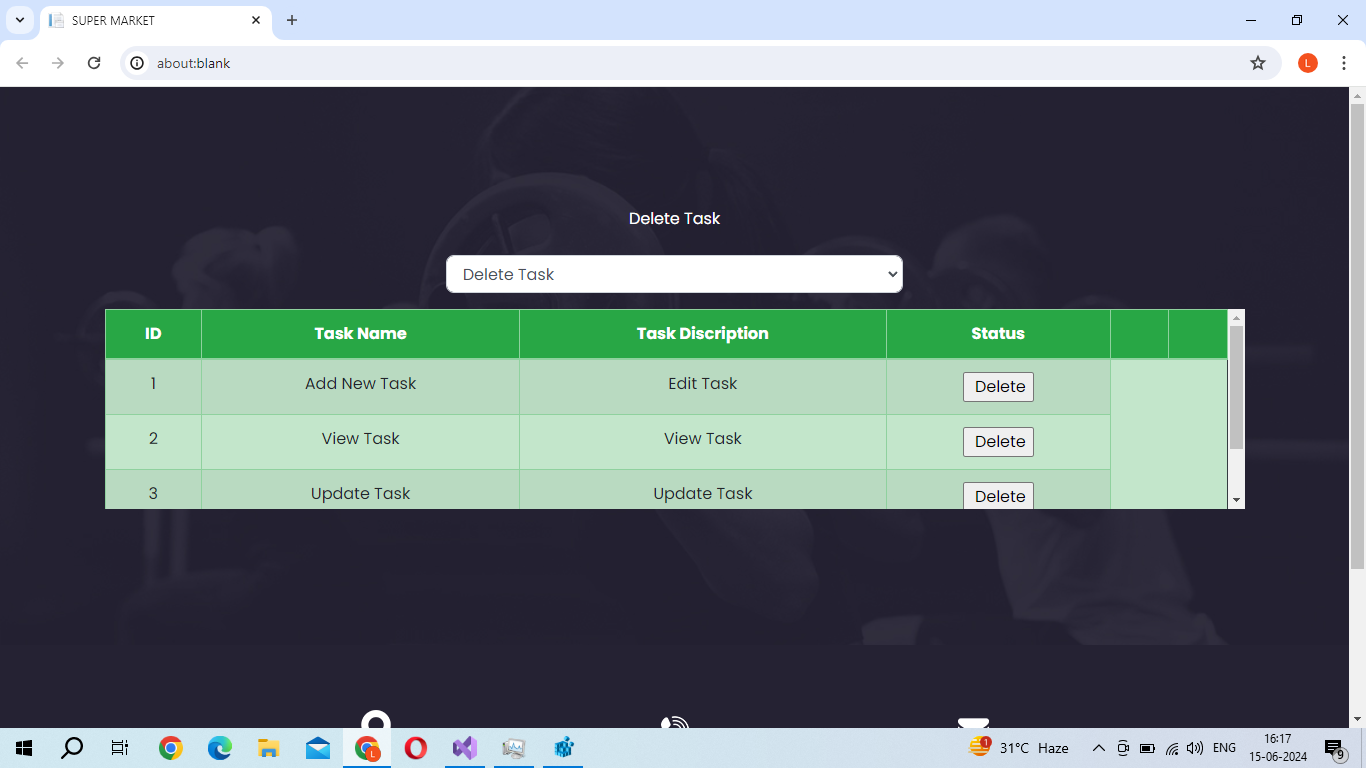
When we select View task we get the table and also we get the task completed or not



When we select the Update Task we also get the table for update the priority. We can choose the priority and click the submit button we completed the task updation.



When need to the delete the task you must select the “Delete Task” in list



CREATE TABLE Customers (

customer\_id int IDENTITY(1, 1) NOT NULL PRIMARY KEY,

customer\_name varchar(25),

email varchar(25),

phone int,

address varchar(50),

city varchar(25),

country varchar(25),

);

CREATE TABLE Product\_Categories (

category\_id int IDENTITY(1, 1) NOT NULL PRIMARY KEY,

category\_name varchar(25)

);

CREATE TABLE Products (

product\_id int IDENTITY(1, 1) NOT NULL PRIMARY KEY,

product\_name varchar(25),

category\_id int FOREIGN KEY REFERENCES Product\_Categories(category\_id) ,

unit\_price int,

cost\_price int,

description varchar(25),

country varchar(25),

);

CREATE TABLE Orders (

order\_id int IDENTITY(1, 1) NOT NULL PRIMARY KEY,

customer\_id int FOREIGN KEY REFERENCES Customers(customer\_id),

order\_date date,

total\_amount float,

region varchar(50)

);

CREATE TABLE Orders (

order\_detail\_id int IDENTITY(1, 1) NOT NULL PRIMARY KEY,

order\_id int FOREIGN KEY REFERENCES Orders(order\_id),

product\_id int FOREIGN KEY REFERENCES Products(product\_id),

quantity int,

unit\_price float,

discoun int

);

Total Sales by Product Category

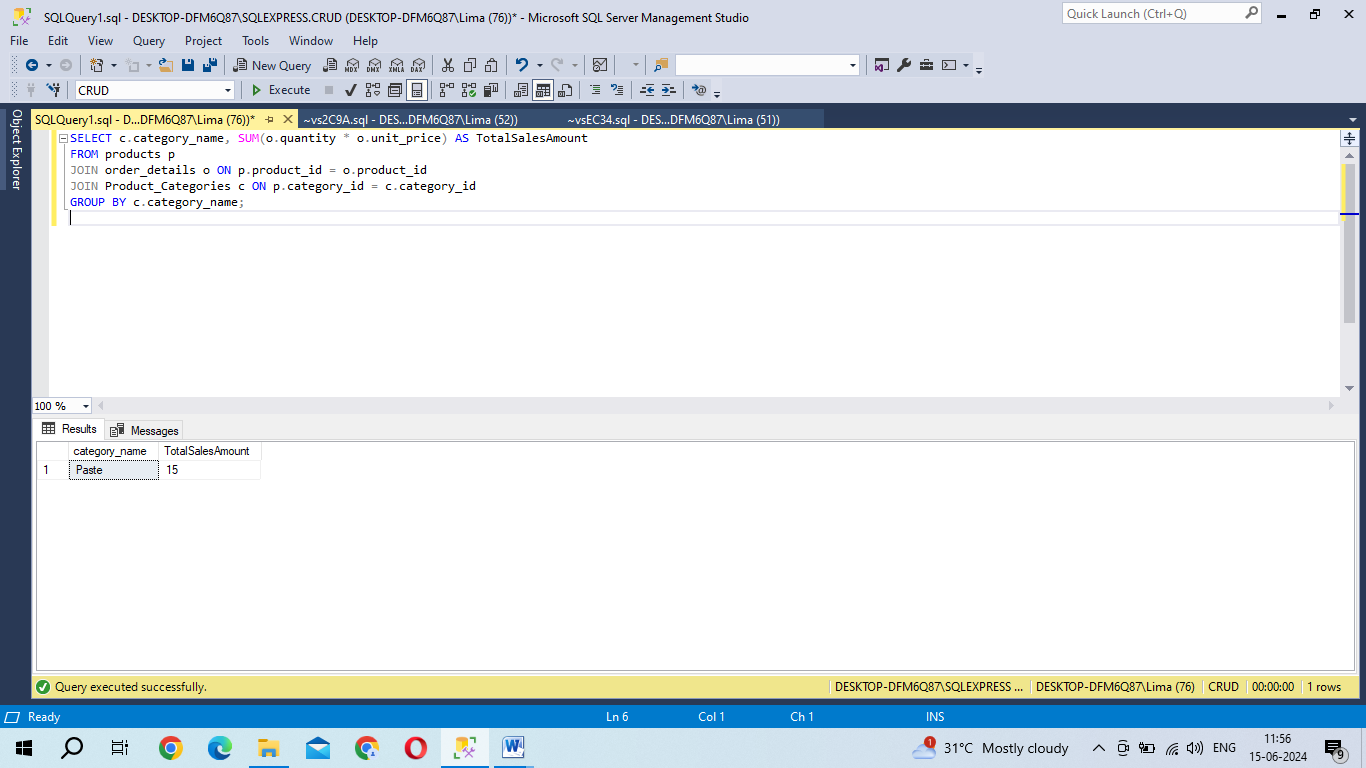
SELECT c.category\_name, SUM(o.quantity \* o.unit\_price) AS TotalSalesAmount

FROM products p

JOIN order\_details o ON p.product\_id = o.product\_id

JOIN categories c ON p.category\_id = c.category\_id

GROUP BY c.category\_name;



Product-wise Profit Margin

SELECT p.product\_name,

SUM(o.quantity \* o.unit\_price) AS SalesAmount,

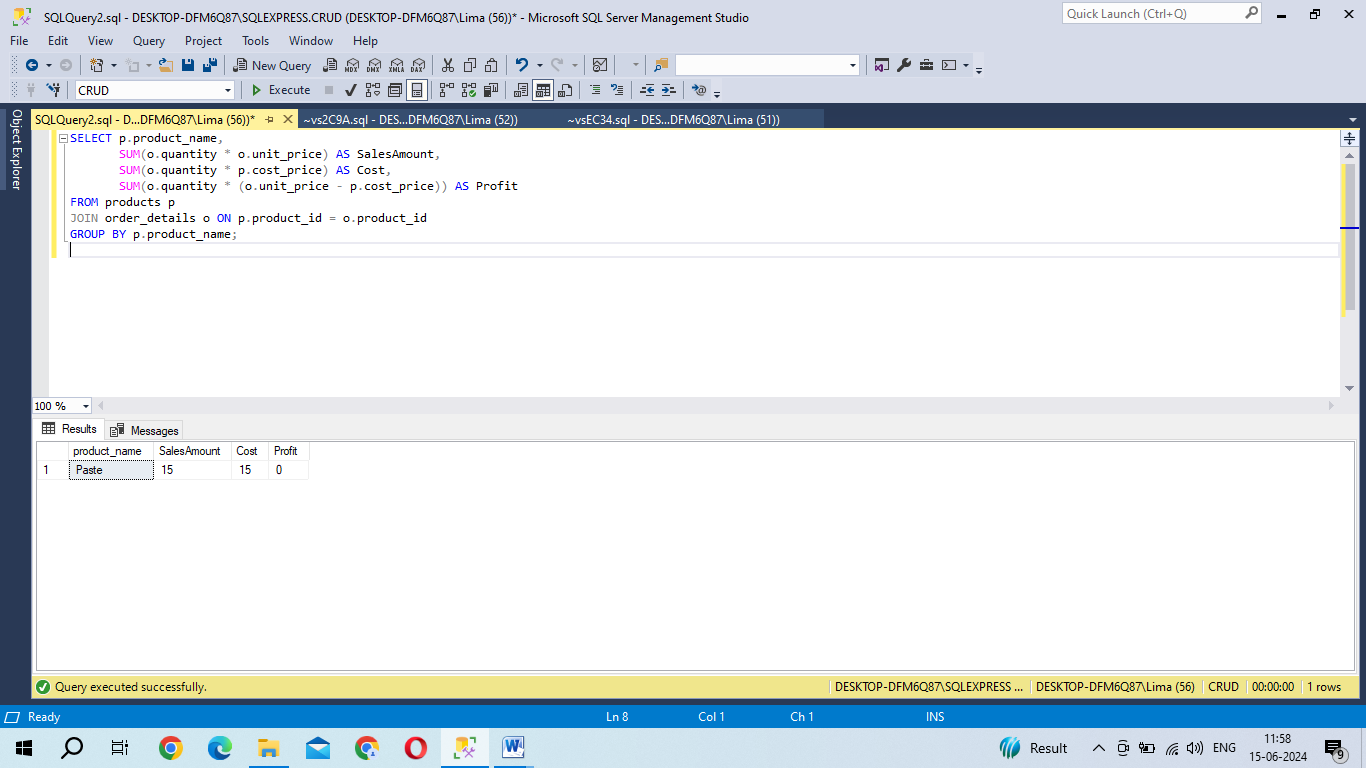
SUM(o.quantity \* p.cost\_price) AS Cost,

SUM(o.quantity \* (o.unit\_price - p.cost\_price)) AS Profit

FROM products p

JOIN order\_details o ON p.product\_id = o.product\_id

GROUP BY p.product\_name;



SELECT p.product\_name, c.category\_name,

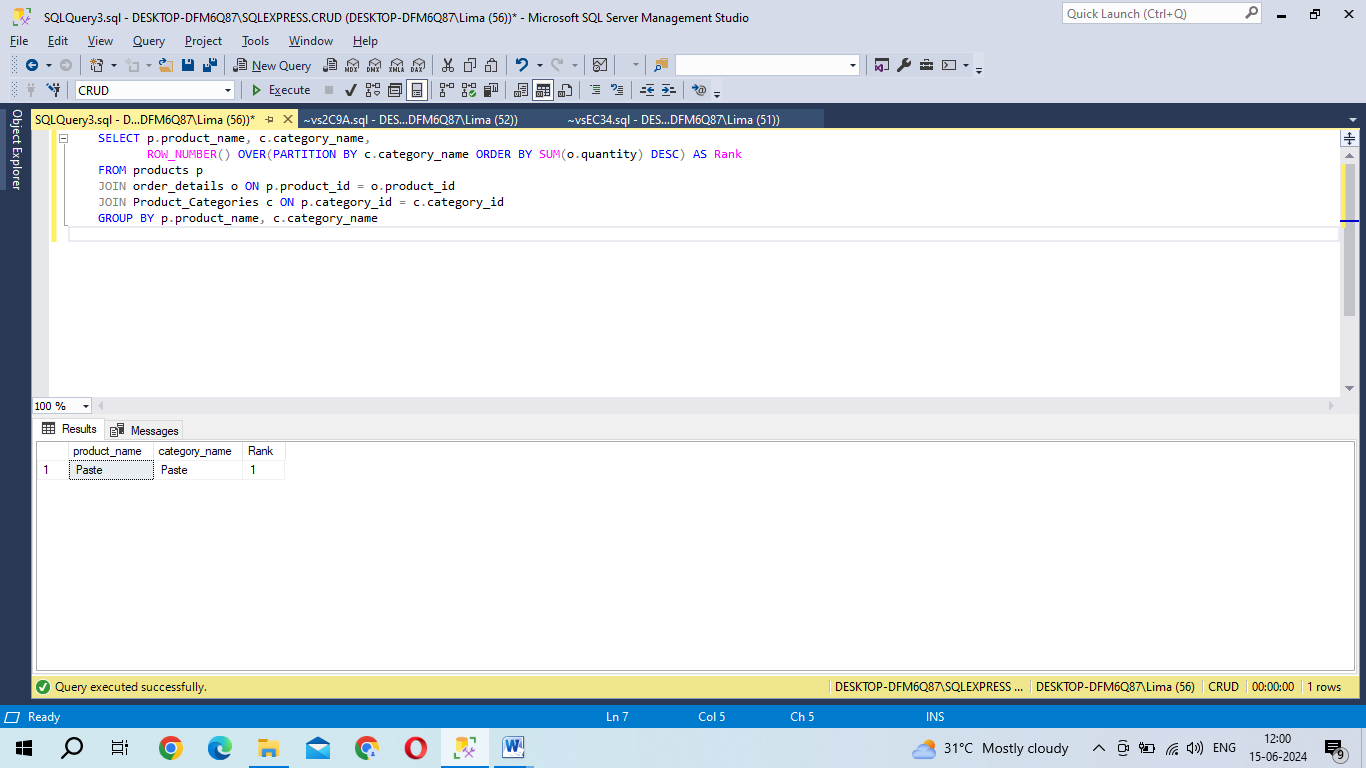
ROW\_NUMBER() OVER(PARTITION BY c.category\_name ORDER BY SUM(o.quantity) DESC) AS Rank

FROM products p

JOIN order\_details o ON p.product\_id = o.product\_id

JOIN Product\_Categories c ON p.category\_id = c.category\_id

GROUP BY p.product\_name, c.category\_name



List of Products with No Sales

SELECT p.product\_id, p.product\_name, p.category\_id,

p.unit\_price, p.cost\_price, p.description

FROM products p

LEFT JOIN order\_details o ON p.product\_id = o.product\_id

WHERE o.order\_id IS NULL;

