

**<Restaurant Management System>**

**<Musharraf Hussain Group (56)>**

**<Agha Ertaza (Khana Peena Restaurant)>**

**<Dr. Mansoor Ebrahim>**

**< https://github.com/Musarraf-Hussain/RestAuto.git >**

* **Requirement Analysis:**

1. **Functional Requirements:**

The functional requirements is describing the behavior of the system as it relates to the systems functionality.

**Food Order:**

Customer can order food what his want.

**Take Order:**

The chef will take order and if is available to make then he will confirm the order and start to prepare food.

**Serve Food:**

When the food is ready to be served then the chef will will alert the waiter, after serving the food the waiter will issue the order as served.

**Available Goods:**

The Chef will add what goods are available and the admin can see that data.

**Payment:**

The cashier will receive the payment if the customer is a member he or she will get discount.

**Database:**

All records to store, retrieve, define and maintain in a database of the system.

For example; Customers, Employees, Admin, Inventory Records, food items etc.

**Non Functional Requirements:**

The non-functional requirements elaborates a performance characteristics of the system.

**Privacy:**

Like a one data cannot access the data of other user and for a better database we can use SAP as a programming language because SAP uses database in the cloud saving and its means that the owner cannot access the database as well where the data of the user will much more private and save.

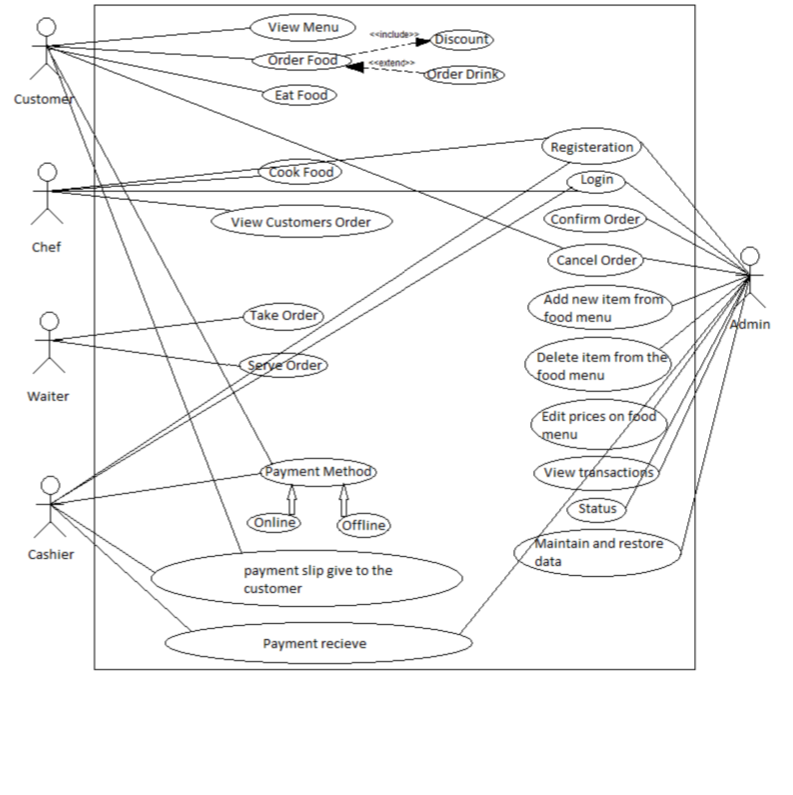
**Security:**

For the security we can use the SAP as well because it is the most safest and the powerful programming language for security because that stores the database in the cloud and another purpose of security is we distribute or divide the system into sub layers which means if the hacker attack a one layer it cannot access the second layer or we can easily identify that the system a got some bugs so we will resolve as soon as possible until the hacker get to the top layer of the system which have critical data.

**Performance:**

The performance is basically depend upon the reliability of the system which means if you have a less configuration in the computer or the operating system you can use a management system properly because it does not take much heavy computer because we have use less variable or less dependency one layer to another layer so it gives much good environment for the user for better experience in the performance of the system.

1. **USE CASE Diagram:**



**Use Case View:**

The use cases for each of the actors are described in this section.

**Customer Use Case:**

Use case: View menu, Order Food, Eat Food.

The customer can view menu and select food item for order than eat food.

**Chef Use Case:**

Use case: Register, login, Cook Food, View Customers order.

Chef need to be register and login. The chef can see the orders of customers and check whether order can be taken or not and then confirms the order and starts preparing the food.When the food is ready the chef alerts the waiter to serve the food. He can also edit what ingredients are available and what ingredients are demanded.

**Waiter Use Case:**

Use case: Register, login, Take Order, Serve Order.

Waiters need to be first register and login. The waiter can take and see the food orders and the read foods in kitchen to be served. After serving the food the waiter will mark the order as served.

**Cashier Use Case:**

Use case: Register, login, Take payment, Payment method, Payment slip give to the customers, Payment receive.

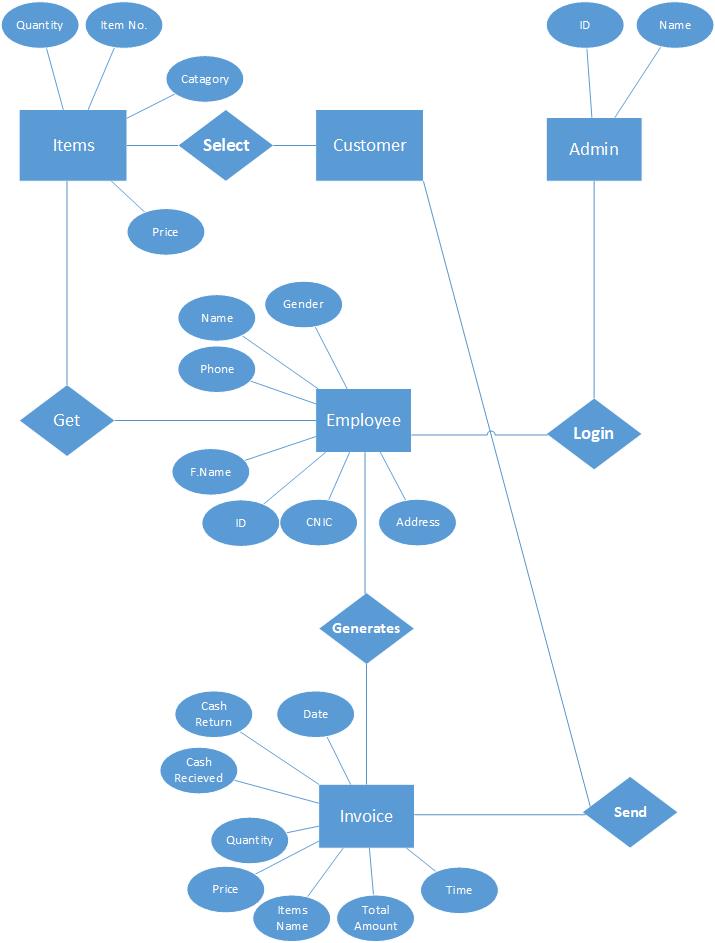
The Cashier need to be register and login. The Cashier can take payment from the customer and save it into the system database with respects to the food item and also check if the customer is eligible for discount. If yes then take the payment accordingly and then give the payment slip to the customer.

**Admin Use Case:**

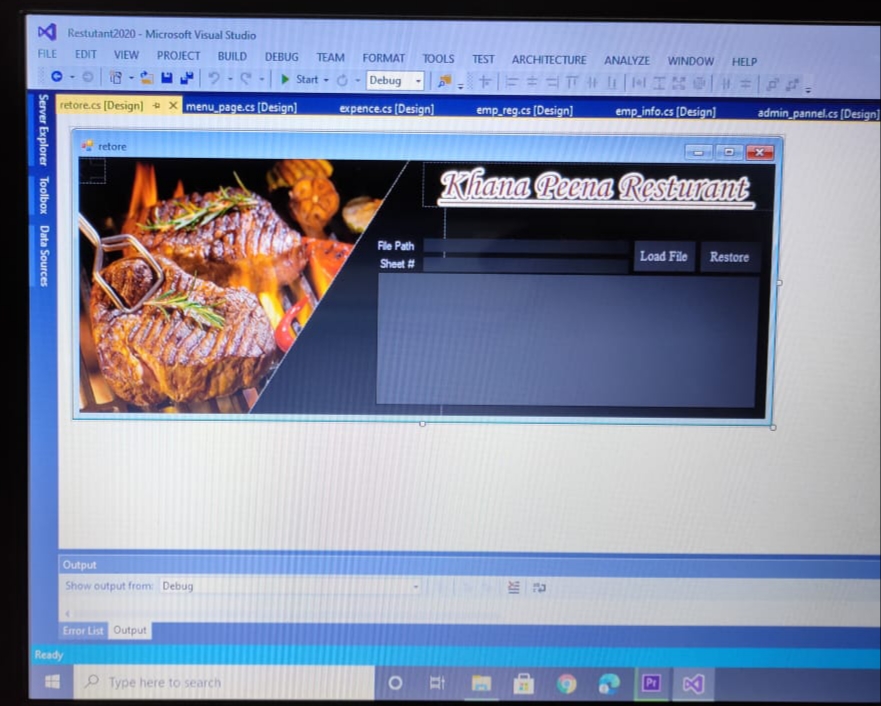
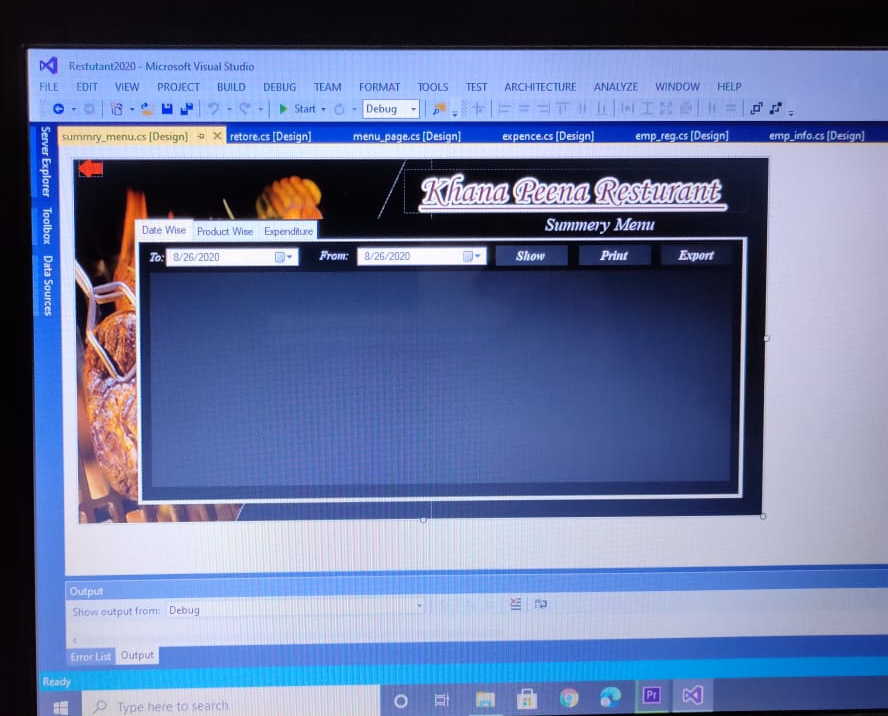
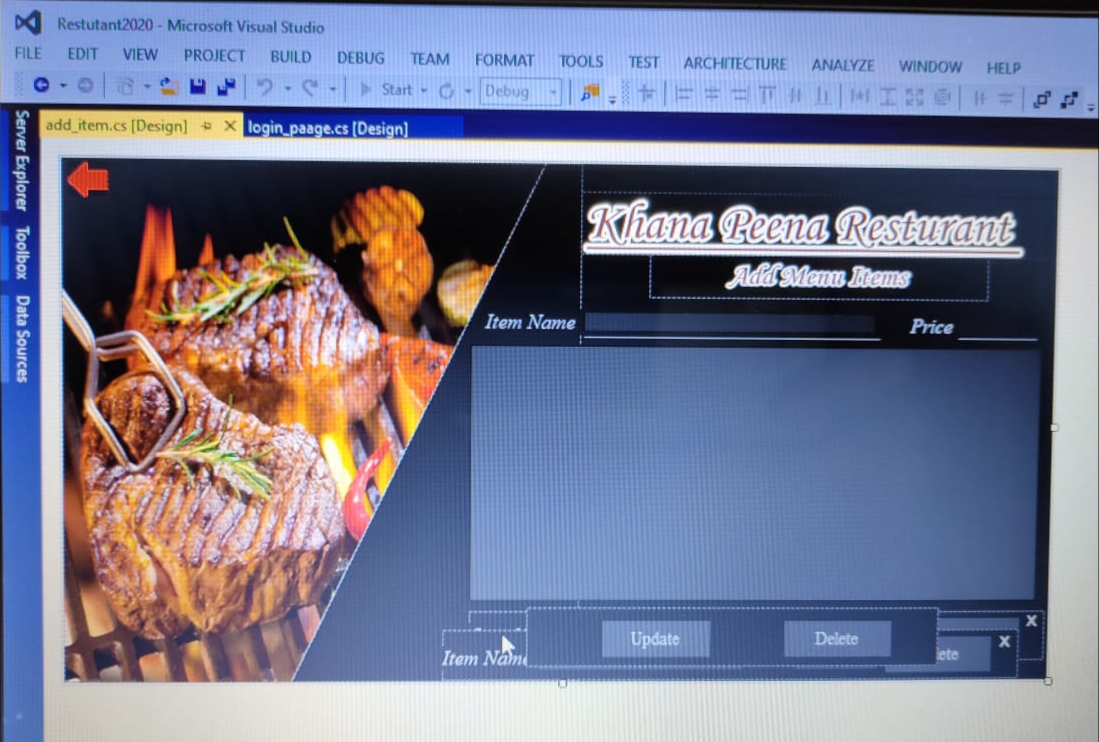
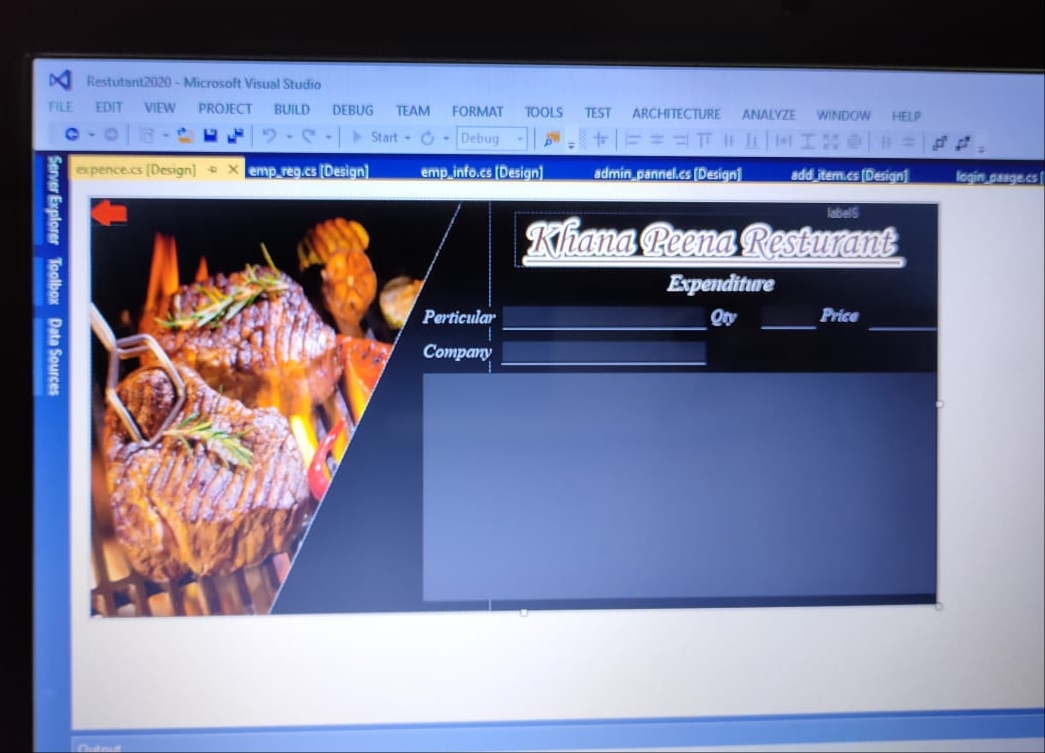
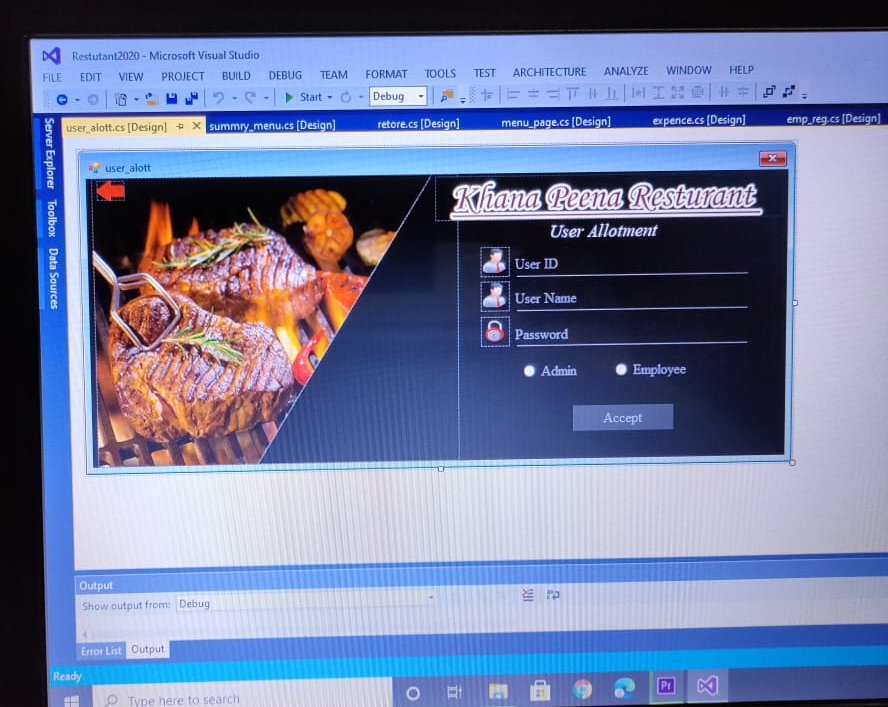
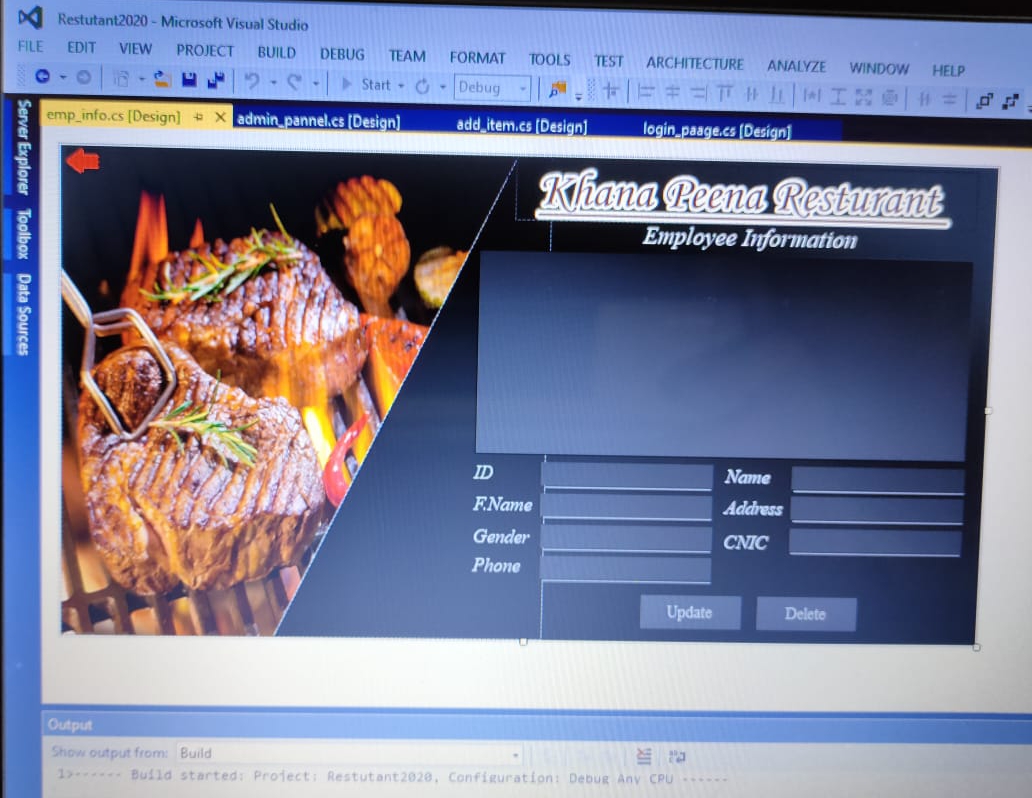
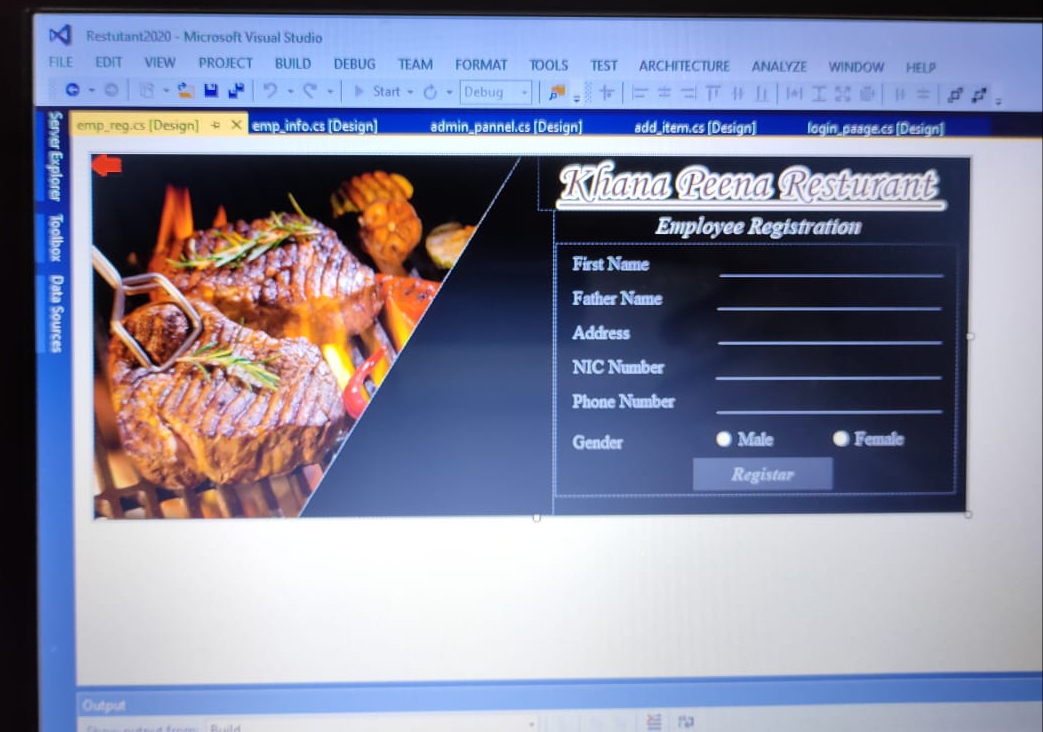
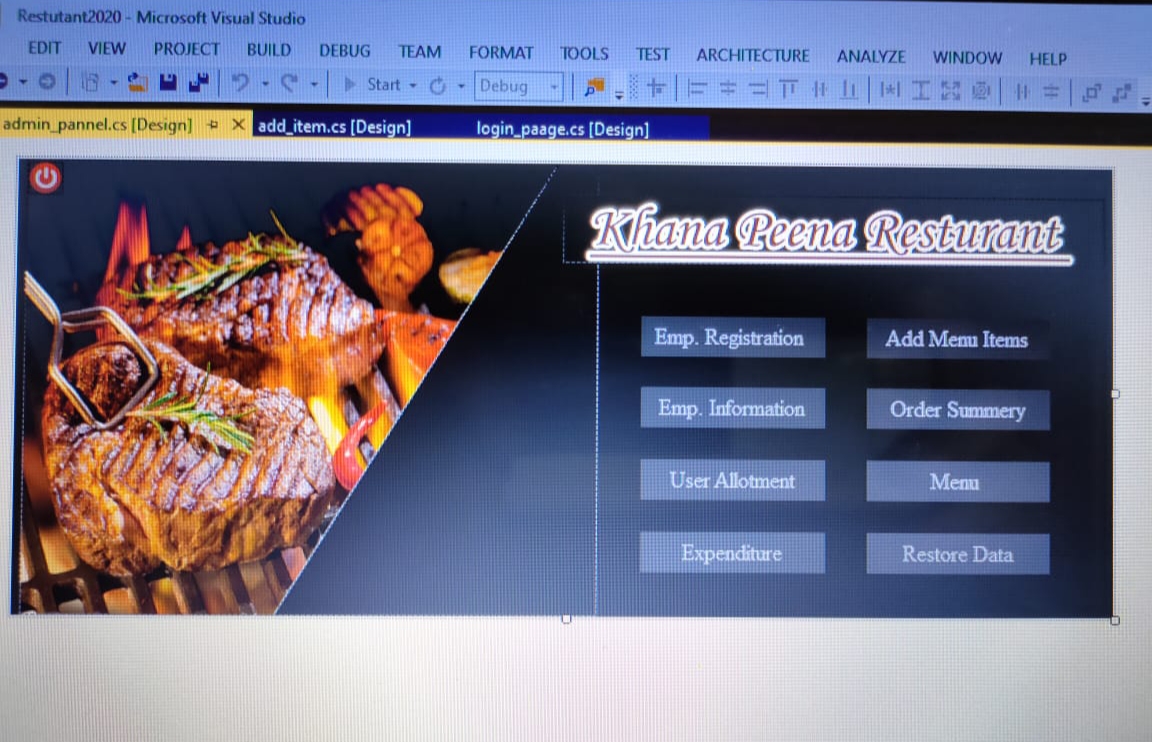
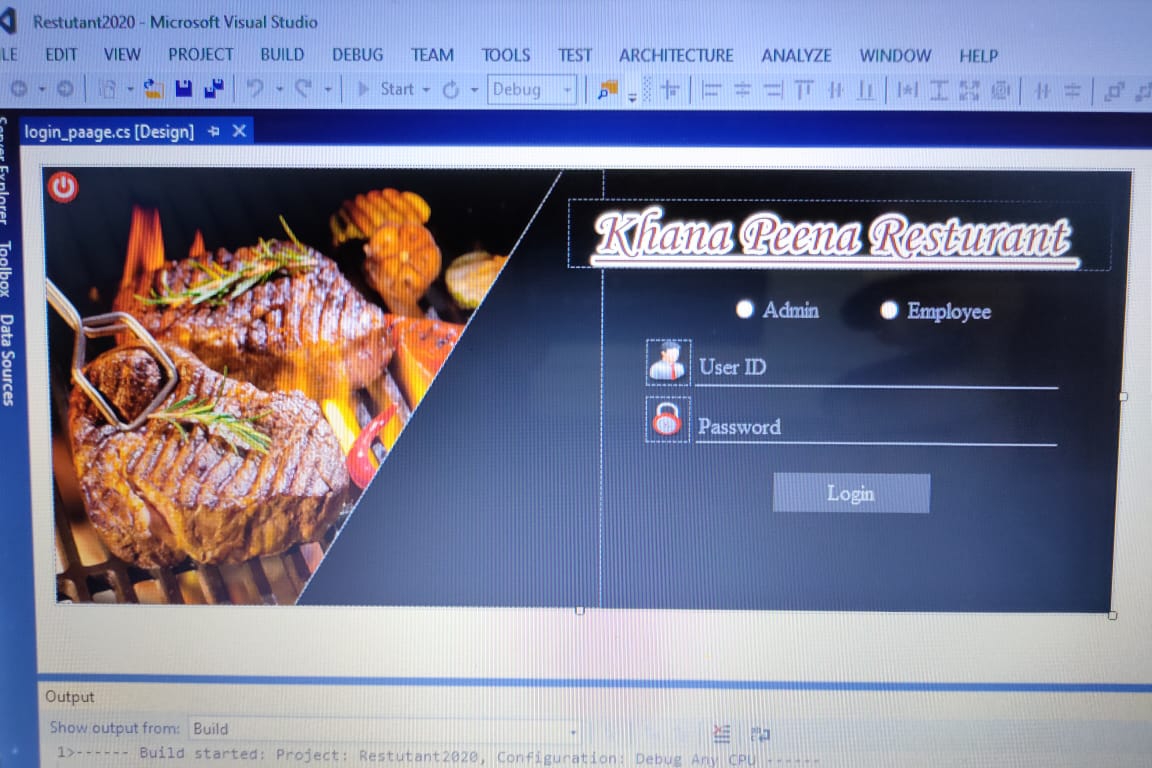
Use case: Register, login, Confirm Order, Cancel Order, Add new item from the food menu, Delete item from the food menu, Edit prices on food menu, View transactions, Status, Maintain and restore data.

First thing is Admin must be register and login. The Admin has full access to the system. He maintains the whole system to ensure better and secure services and solves any error appeared in the system.

1. **ERD**

****

* **Usability:**

****