# Pizza ordering System

A Pizza shore wants to launch a web application to take orders for its pizzas from customers.

Customer can order the type of pizza that he wants and also quantity. The system provided the below capabilities,

1. Setup Varity of pizza’s information
2. Setup type of flavors that user needs for that pizza
3. Conform the order

## Detailed Requirements

### User Setup

For every pizza type there is a particular id, and name of the pizza, there will be a verity of specials to that pizza type, and also the user can have the option to choose whether he can go for veg or non-veg or both.

1. Pizza id
2. Pizza name
3. Pizza type (veg, non-veg or both)
4. Pizza specialty
5. Pizza crust
6. Pizza’s price
7. Number of slices need to be made

**Screen 1: Home**

On the home screen the user can see the verity number of pizzas they might be of non-veg, veg or both mixes. In the home screen header bar, there will logo to left, name in the middle and right side we there will be order details.

|  |  |  |
| --- | --- | --- |
| S.no | Menu bar (navigation bar) | comments |
| 1 | Home | This will redirect to the (Main page). |
| 2 | Order page | This will redirect to the (Choosing type of pizza page) and here only they can place order. |
| 3 | Status of order page | This will redirect to the (Time required and status of the order). |

**Screen 2: Order page**

In this page the user can choose the type of the pizza, style of the pizza, flavors that need to added to the pizza and the quantity of pizzas (I.e., the use can order any number of pizzas)

1. Order id is auto generated. (Primary key)
2. The pizza id should be taken from the pizza.(Foreign key)
3. The order date (YYYY/MM/DD).
4. Order style and flavors need to be added.
5. How many pizzas required.
6. Address where we need to deliver the order.
7. Finally, the amount that they need to be paid. (The payment is only of credit card type)
8. Before conforming the order, there must be a dialog box the once the order is placed it cannot be cancelable
9. After conforming the order it will show as order is placed in the status page.

**Screen 3: Status page**

In this status page the user can view his order and how much time it will take to complete.

1. The order id.
2. The pizza id.
3. The status details.
4. And also, the quantity.

**General requirements:**

* All the pages will have the Logo, name in common.
* All the pages will have Home, Order and status page in common.
* In the order page he can see the placed order.
* On failures, the users will see the appropriate friendly error message (not exceptions).

**Technical Requirements:**

* Tables should be designed appropriately (proper data types, table/column names, constraints), use a SQL Server database
* The tables should be created only using SQL Files and these should be available in the repo
* Front end – ASP.NET Core MVC, HTML5, CSS, Bootstrap, JavaScript/jQuery
* Business Logic – Web API/Service in MVC
* Data Access – EF Core
* All names used in the code (classes, variables, properties etc.,) should be meaningful and follow consistent naming approach (Camel Case, Pascal Case etc.,)
* Classes/methods/properties should have proper accessibility modifiers
* Use code formatting to ensure the code is readable, write comments where required so that the reviewers can understand your code

**Important:**

* Create a new repo “Project A” in your account and use that for this exercise
* Push the code to the remote repository often or at least before end of each day.
* Repo should have a folder “DB Script” and it will have all the SQL Files (tables, stored procedure etc.,)
* Repo should have a folder “src” and it will have your ASP.NET Core Projects