Pizza Ordering Management System

A small pizza shop wants to allow their customers to manage their orders through web application.

Users in this document refer to Customers. The system provided the below capabilities,

* 1. Pizza Information
  2. View Pizza Details.
  3. Summary of Order.
  4. Confirmation of the Order.

The system will be secured and customers can perform any of the above functions.

## Detailed Requirements

### Pizza Setup

Each pizza will have unique id and pizza name. There is no screen to manage the pizza, the data is stored directly in the database and also the user can have the option to choose whether he can go for veg or non-veg or both.

The database table will have below data for all the pizza.

1. Pizza Id
2. Name
3. Crust
4. Speciality
5. IsVeg

### Screen 1: Home Page

This screen will prompt the user to view all pizza information, So the user can select their desired pizza. The page will also have a small pizza logo to left on the header.

Screen 2: View Pizza details

This screen Show the specifications of that pizza and user can chose the

1)quantity

2) crust

3) toppings, based on that it will display the price to the User.

And it shows two buttons Order now and cancel.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Menu Name** | **Comments** |
| 1 | Order now | This will redirect to the “Summary” screen |
| 2 | Cancel | This will redirect to the “Home” screen |

### Screen 3: Summary

This screen displays the

1)type of pizza User choose

2)Quantity

3) Crust

4) Toppings (based on the details we choose in the View details page)

5) delivery date

It has three buttons confirm order, alter order and cancel.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Menu Name** | **Comments** |
| 1 | Confirm order | This will redirect to the “ordered Confirmed” screen |
| 2 | Alter order | This will redirect to the “View pizza details” screen |
| 3 | Cancel | This will redirect to the “Home” screen |

### Screen 4: Order Confirmed Page

This screen displays the message “The pizza is ordered ” and it has a link “Home” which redirects to home Page.

### General Requirements

* All screens except “Login” and “Home” will have a button “Go to Home” to take the user back to the Home screen
* Whenever an operation is successful, the users should be shown a message as such. E.g., if the “Add Patient” is successful, it will display “Patient successfully added to the system”. It would be helpful to show the names/ids in these messages as well to make it usable.
* On failures, the users should be shown 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 (DB/Model first approach)
* 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 “ProjectA” 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