

Developing Web Application With Visual Studio C#, ASP.Net Core Razor Page and SQL Server

I. Case Study

You have been hired to develop a product stock management web application for a company.

Following are the excerpts from the *functional requirements* of the application and the assumption:

1. The application allows the *User* to create products.
2. The Application can show all the products.
3. The user can delete and edit the products.
4. The *User* can see details of the stock, for example total sell price in the stock , price average and profit.

Data:

Product table:

ProductId	ProductDescription	Quantity	BuyPrice	SellPrice
PROD101	T-Shirt	10	10.99	15.99
PROD102	Pants	4	29.99	49.99
PROD103	Jogging	9	19.99	29.99
PROD104	Shorts	12	29.83	50.29
PROD105	Flip-Flop	20	10.99	20.69
PROD106	Shoes	15	50.99	89.90

II. Technical Requirements

You should use the following technique to develop the application

- .Net Core Razor Page
 - Creating and Using Models
 - Creating and Using Views
- Using SQL Server Database in .Net Core
- Using **Entity Framework** in .Net Core
- Data Annotations
- Routing in MVC
- Design should be responsive and add some style to pages.

Part 1: Database Design and Implementation

For each table, specify Primary Key, data type.

- All the fields are ***required***
- Product Description must **not be longer than 50 chars.**

Create the models, the migration and create the database from your models.

Part 2: Display all the products

The index page should show in a table all the products and its information.

Each product must have two buttons “Edit” and “Delete”

At the Nav-bar, one new link must be added called “My Stock Info”, this link will redirect the user to see the total amount of the stock, the price average for the products and also the total profit for the stock

The products with less than quantity 10 must be marked in **RED**.

Part 3: Create Product Page

Create a product page must be called when a button named “Create Product” is clicked on the index page. The create product page must have a form with all validations for the product model.

If the form is valid, the user must return to index page, if not valid the user should stay at creation page and see the validation errors.

Part 4: Edit Product

Edit a product page must be called when the user presses the button “Edit” on the product. A page with a form (filled) should be called and the user can edit all the information in a product, **but its ID.**

If the form is valid, the user must return to index page, if not valid the user should stay at creation page and see the validation errors.

Part 5: Delete Product

Delete a product page must be called when the user presses the button “Delete” on the product. A page with all the information for that product must be called and a question “Are you sure that you want to delete this product?”. If the user presses “Yes” delete the product and go back to index page, if the user presses “No” go back do the index page too.