Creating an ASP.NET MVC Application for Online Laptop Sales

Building an ASP.NET MVC application for online laptop sales is a step-by-step process that involves designing the database schema, creating models, controllers, views, and implementing CRUD (Create, Read, Update, Delete) operations. Here is a writeup on how I went about it :

Create an ASP.NET MVC Project

We started by creating a new ASP.NET MVC project in Visual Studio.

We ensured that we chose the appropriate project template for MVC web applications.

Design the Schema

We designed the database schema that will store information about laptops available for sale.

Consider the entities and their relationships. In this case, we may have a "Laptop" entity.

Create Models

We defined model classes that correspond to the database tables. In this project, we created a "Laptop" model. We connected the model to the database directly using ADO.NET Entity Model

Ensure that we set up any necessary data annotations and validation rules in the SQL before creating and linking the model.

Create Controllers

- In the project, locate the "Controllers" folder.
- Right-click on it and choose "Add" > "Controller..."
- Select the "MVC 5 Controller with views, using Entity Framework" option and click "Add."

- Choose the model class (e.g., "Laptop") and configure the controller actions for CRUD operations (e.g., Create, Read, Update, Delete).
- These controller actions will handle HTTP requests and interact with the database using Entity Framework.

Create Views

In the "Controllers" folder, we will find auto-generated views for CRUD operations (e.g., Create.cshtml, Edit.cshtml, Details.cshtml, Delete.cshtml).

Customize these views as needed to provide a user-friendly interface for adding, editing, deleting, and viewing laptop records.

Ensure that we use HTML helpers and Bootstrap to create a responsive and visually appealing UI.

Implement CRUD Operations

Inside the controller actions (e.g., Create, Edit, Delete), write code to interact with the database using Entity Framework.

Implement the logic to handle Create, Read, Update, and Delete operations for laptops.

Ensure that data validation and error handling are properly implemented.

Testing

Run the application locally to test the CRUD operations.

Verify that we can create, read, update, and delete laptop records through the web interface.

Perform thorough testing to ensure the application functions as expected.

Push to GitHub

Once we are satisfied with the changes, we commit them using Git and push to our GitHub repository. This will track our SQL scripts and VS code.

git init

git add .

git commit -m "Creating a MVCSchoolDatabase"

git branch -M main

git remote add origin https://github.com/PrashastVats1/Phase2Projects.git

git push -u origin main

GitHub Link :

git remote add origin https://github.com/PrashastVats1/Phase2Projects.git