**Writeup for Phase-3 Project:**

Github link: https://github.com/niljames/simplilearn.git

Question:

Create an ASP.NET MVC web application to sell laptops online.

**Background of the problem statement:**

An existing retail company, Digital Retailers Inc, wants to take their business online and create an e-commerce portal that will enable people to purchase laptops online. They like to prototype new projects quickly using Agile methods and hence want to avoid a long requirement capture stage. They want to keep track of their codebase using Git so that multiple developers can work on it as required.

**You must use the following:**

Visual Studio ASP.NET MVC web Project

SQL Server 17 Express Edition or later

Check in the project source into Github using Git tools.

**Application Requirements:**

The pages must be done using Bootstrap and must be mobile-responsive.

All data must be stored and retrieved using Data Access Layer and/or EntityFramework.

**Following requirements should be met:**

Some of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.

The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.

The step-by-step process involved in completing this task should be documented.

**Explanation:**

* A database must be created first.
* I created a database called “**PhaseThree**” with four tables: **User, Product, Description** and **Address**.
* The following figure depicts the relationship between all the four tables and their columns as the primary keys and the foreign key.
* As seen, there’s a one-to-one relationship between **Product** and **Description** tables and a one-to-many relationship between **User** and **Address** tables.
* In the **Description** table, the “**productId**” is the foreign key and in the **Address** table, the “**userId**” is the foreign key.
* I used the code-first approach.

![Graphical user interface, application

Description automatically generated]()**Fig 1: Relationship between the tables in the database.**

**Explanation Via Images:**

![Graphical user interface, website

Description automatically generated]() **Fig 2: Home page without Login.**

* This is the initial landing page that the user lands on, when the application is launched/run.
* The homepage contains a navbar that contains the Brand name, the home tab, the product tab as well as the cart tab on the left side.
* On the right side of the navbar, the login and the register buttons are available.
* The body of the home page contains the heading as well as a button “About us” which when clicked pops up a modal that gives the description of the company.
* The footer exists at the end, which has the copyrights.

![Graphical user interface, text

Description automatically generated]() **Fig 3: About Us modal**

* This is the modal that pops up when the button “about us” is clicked. The modal contains a header which contains the title as well as the close button icon, a body which contains the description and the footer, which contains the close button to close the modal.

![A picture containing text, electronics, computer, screenshot

Description automatically generated]()**Fig 4: Register Page**

* A new user can click on the register button in the navbar to create an account.
* Fig 3 depicts the register page.
* It contains 4 fields:
* Username: The name that the user wants to be identified with.
* Password.
* Confirm Password: Must match the above mentioned password.
* UserType: A dropdown that contains “Admin” and “User”. Must choose either to create an account.
* If everything is done successfully, the account will be created and updated in the table “user” in the database.
* Contains a Register button that creates the account on successful inputting and a Login button, that takes you to the login page.

![Graphical user interface, website

Description automatically generated]()**Fig 5: On Successful Registration**

* On successful registration, the account is created and the user will be directed to the product page.

![A picture containing text, electronics, computer, screenshot

Description automatically generated]()**Fig 6: Passwords Mismatch**

* If the password entered and the confirm password entered are not the same, the “password do not match” error will be outputted.

![A picture containing text, electronics, computer

Description automatically generated]()**Fig 7: Login Page (Invalid Credentials)**

* If the user enters invalid credentials, an error will pop up.

![A picture containing text, electronics, computer, display

Description automatically generated]()**Fig 7: Login Page**

* This is the login page.
* Contains three fields, the username, password and the usertype.
* On successful login, the user will be taken to the homepage. In the navbar, the login and register button will be replaced with the logout button.

![Graphical user interface, application, website

Description automatically generated]() **Fig 8: Product Page**

* The products are represented in the form of a grid and are very responsive too.
* Each product is represented in a card and each card contains the product image, the product name, the price and two button.
* The View Details button will show more description about the product and the Add To Cart button will add the chosen product to the cart for purchase.
* All the products in the product page are retrieved from the **Product** table in the **PhaseThree** Database created.

**![Graphical user interface, application

Description automatically generated]()Fig 9: View Details (part-1)**

* On clicking the View Details button, the user will be directed to the details page, where he/she can view more details about the product, such as the OS, memory, etc.
* These details/descriptions are obtained/retrieved from the Description table of the PhaseThree Database.
* As seen in fig 10, it contains two buttons.
* Add to cart will add that specific product to the cart and Back to list will guide the user back to the product page.

**![Graphical user interface

Description automatically generated with medium confidence]() Fig 10: View Details (part-2)**

**![Graphical user interface, application

Description automatically generated]()Fig 11: Cart Page (part-1)**

**Graphical user interface, application, website

Description automatically generated Fig 12: Cart Page (part-2)**

* The cart page will contain a table that consists of the list of items/products added to the cart, their quantities, their sub-total and the total amount.
* It also contains the a form, where the user has to fill in the details of their address.
* Contains three buttons.
* Continue Shopping, will take the user back to the product page. Checkout will take the user to the payment page.
* On entering the details of the form, the values will then be updated in the Address table of the database, which will then be retrieved later for final order summary.
* The form contains the following fields:
* Email.
* First name.
* Last name.
* Address.
* Zipcode.
* City.
* State.
* Country.
* Phone number.

**![Graphical user interface, website

Description automatically generated]()Fig 13: Payment page**

* On clicking “Checkout”, the user is taken to a payment page, where the total amount to pay will be displayed on a jumbotron.
* I have used Stripe as the payment gateway.
* The jumbotron contains two buttons.
* Return to cart will navigate back to the cart page and Pay with card, will navigate to the payment modal that contains the input fields to enter the card details.

**Graphical user interface

Description automatically generated Fig 14: Payment modal**

* The payment modal contains 4 fields, where the email, card number, card expiry date and the cvv will have to be entered.

**Graphical user interface, application

Description automatically generated Fig 15: Final Summary Page.**

* On successful payment, this page will be generated where there exists a card, that contains the following.
* A token id generated via the payment gateway, to track the payment.
* Details of the user, who placed the order (Name and Phone number).
* A note of the payment being successful.
* A delivery date.
* The delivery address.
* Back to home page button.

**![A picture containing text, electronics, computer, screenshot

Description automatically generated]()Fig 16: Admin Login.**

* All the above functionalities were done using the user account. Furthermore functionalities can be performed like adding the product and description details. These functionalities can only be added by the admin.

**![Graphical user interface, website

Description automatically generated]()Fig 17: Admin HomePage.**

* This is the admin homepage. As we can notice, the navbar contains two more extra tabs called ProductAdmin and Description.

**![Graphical user interface

Description automatically generated]()Fig 18: ProductAdmin Page.**

* The ProductAdmin tab leads to this page, that contains the list of products as seen in the product page.
* It’s a CRUD table, that contains the option to create, edit, view details and delete the corresponding products.
* There already exists 4 products on the given table.
* On clicking create new, the following form opens.

**![Graphical user interface, text, website

Description automatically generated]()Fig 19: Creating a new product.**

* The create form contains three fields: Name, price and image.
* The create button, will create a new product.

**![Table

Description automatically generated]()Fig 20: After creation (CRUD table).**

**![Graphical user interface, application, website

Description automatically generated]() Fig 21: After creation (Product page).**

* The following changes can be seen in both CRUD table as well as the product page.

**![Graphical user interface, text, application

Description automatically generated]()Fig 22: View Details of Product. (From CRUD table)![Graphical user interface, text, website

Description automatically generated]()Fig 23: Edit Details (price) of Product. (From CRUD table)**

**![Table

Description automatically generated]()Fig 24: After editing price of Product. (From CRUD table)**

* Similarly, for description tab, there exists a CRUD table that performs the same functionalities as that of productAdmin.
* The figures are self-explanatory.

**![Table

Description automatically generated]()Fig 25: Description (From CRUD table)Graphical user interface

Description automatically generated Fig 26: Creating Description for a new product (From CRUD table)**

* Contains the productid that denoted the description is created for which product.

**![Table

Description automatically generated]()Fig 27: After creating the description. (From CRUD table)**

**![A screenshot of a computer

Description automatically generated with medium confidence]() Fig 28: After creating the description. (From product table)**

**![Graphical user interface, text, application

Description automatically generated]() Fig 29: The 4 tables with the values in database.**

* The following images depict the responsiveness of the site in mobile view. It depicts the responsiveness of all the pages in mobile-view.

**![Graphical user interface, website

Description automatically generated]()Fig 30.1: Mobile-View(Home-page)A screenshot of a computer

Description automatically generated with medium confidence Fig 30.2: Mobile-View(Navbar)A screenshot of a computer

Description automatically generated with medium confidence Fig 30.3: Mobile-View(Product-page)A screenshot of a computer

Description automatically generated with medium confidence Fig 30.4: Mobile-View(Description)![A screenshot of a computer

Description automatically generated]() Fig 30.5: Mobile-View(Cart-page part-1)A screenshot of a computer

Description automatically generated with medium confidence Fig 30.6: Mobile-View(Cart-page part-2)A screenshot of a computer

Description automatically generated Fig 30.7: Mobile-View(Payment-page)A screenshot of a computer

Description automatically generated Fig 30.8: Mobile-View(Final-Summary)A screenshot of a computer

Description automatically generated with medium confidence Fig 30.9: Mobile-View(Login-Form)![A screenshot of a computer

Description automatically generated with medium confidence]() Fig 30.10: Mobile-View(Register-Form)**