MS.NET

SPRINT-2 CASE STUDY

e-Commerce Shopping Store

(eCSS)

Document Revision History

Date	Revision No.	Author	Summary of Changes
20th Nov. 2008	1	Ganesh Desai	Initial Draft
15th Dec 2008	1	Anil Patil	Review
25th July 2011	1	Karthik	Review after integration Process
11th May 2015	1	Vaishali Kasture	Modifications as per new curriculum
13th July 2016	1	Vijay Vishwakarma	Incorporating changes as per new curriculum
30th July 2019	1	Amol Patil	Incorporating changes as per new curriculum
13th March 2020	2	Ramkumar Nandabaalan and Raghu KN	Incorporating changes as per the requirement of partial immersive batch

Table of Contents

INTRODUCTION

This document outlines SPRINT2 for the .NET Line of Technology (LOT). The project is to develop an Online Shopping Store. This document contains the work flow of the system and gives guidelines on how to build the functionality.

SETUP CHECKLIST

Minimum System Requirements

- Intel Pentium 4 and Windows 7 & above
- Memory 4 GB
- Internet Explorer 8.0 or higher
- SQL Server 2012 client and access to SQL Server 2012 server
- Visual Studio 2015 / 2017, Visual Studio Code

INSTRUCTIONS

- The code modules in the SPRINT2 should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory SPRINT2. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the MSDN
- Since this project work will span over week, you will need to take care of maintaining the code

PROBLEM STATEMENT

OBJECTIVE

Development of an eCommerce Shopping Store Application (eCSS)

ABSTRACT OF THE PROJECT

CyberShopee wants to set up Online shopping facility for their Customers. They sell various products belonging to various categories. E.g they have Electronics Items Category having products like TV, Camera etc.

This Online shopping Application should provide the following features to the customers:
1. Customer Registration
2. View Product Details
3. Purchase Products
4. Confirm the Order

Note: Other than first time registration, other operations are allowed only for registered customers.

Customers can add a given product to the shopping cart.
Customers can view products by category.
Customers can check out the shopping cart to place an order

Once the Order is confirmed, Cyber shopee will deliver the ordered products to the customer on his address specified during the registration. The Payment mode will be Cash On Delivery.

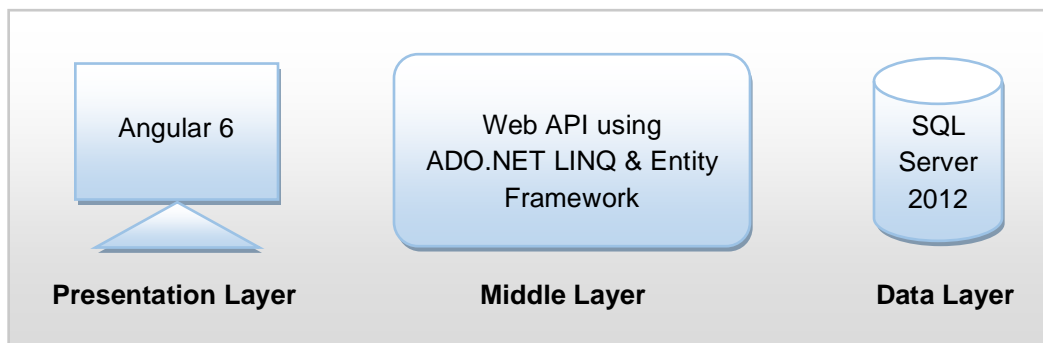Develop the Application for the above requirement using MS.NET Technology.

FUNCTIONAL COMPONENTS OF THE PROJECT

Application Architecture:
Distributed web applications traditionally to be designed and built across three logical tiers:

- Database Access Layer (DAL)
- Business Logic Layer (BLL)
- Presentation Layer (PL)

The DAL refers to the database itself, the stored procedures, and the component that provides an interface to the database. The BLL refers to the component that encapsulates all the business logic of the application. And, the Presentation layer refers to the web application pages.

Presentation Layer	Middle Layer	Data Layer
Angular 6	Web API using ADO.NET LINQ & Entity Framework	SQL Server 2012

Design Guidelines

- All the exceptions/errors to be captured and user friendly message to be displayed on the In Web API use ADO.NET Entity Framework. Make use of stored procedures. All the database interaction would be performed using Data Access Component.

TECHNOLOGY USED:

> *Presentation Layer*
> > *Angular 6*
> *Business Layer*
> > *Web API Using ADO.Net LINQ & EF*
> *Database Layer*

➢ *RDBMS:-* SQL Server 2012

IMPLEMENTATION

GUIDELINES ON THE FUNCTIONALITY TO BE BUILT:

The functionality and components to be built in each of the course modules of .NET LOT is as follows:

Course: SQL Server 2012

This section describes some of the basic steps involved in designing and creation of the database for the application.

Create Data Model - identify the different tables and fields that we will need, which would later be used for building the rest of the application.

Database Schema - Create the following database tables (Make your assumptions for creating all required fields for the tables) & if required some more tables apart from the mentioned below.

- Customers: This will contain the list of Customers.
- Products: This contains the list of products available in the shopping store for purchase.
- Categories: This will contain various Categories for the Products available.
- Orders: This will contain the Master details of Orders.
- OrderDetails: This will contain the details of the Products Ordered.
- Shopping Cart: This table will store the details about the Cart.

Course: ADO.NET 4.5, LINQ and Entity Framework

Write Data Access Components to perform any CRUD operations using ADO.NET & Entity Framework.

- Use LINQ to query record sets returned by Data Access Components
- Build the web application and deploy on the IIS server and test the application.

Course: ASP.NET Web API

- Create RESTful Web Services using ASP.NET Web API & C#-7.0 to expose CRUD operation services to be consumed by any HTTP client, for instance Angular6 client.

Course: Angular 6

Develop the prototypes for following functionalities:

i. **Login / Sign In**: Login screen would display asking user to enter 'User Id' & 'Passwd'. If the supplied user credentials are valid, then HomePage would be displayed, else appropriate Login error message would be displayed.
    Use Token based authentication

ii. **First time users Register**: First time new users would use this screen to create their profile (userId and password). On click of 'Submit' button new user entry (user id & password) would be saved into database table for further user authentication purpose.

iii. **Home Page**: On successful user authentication (validation of userid/password provided by the user in login screen) the homepage would be displayed. The Homepage would contain below sections/contents:

Header section: The header section would be common across all the pages and would mainly have –

Sign In Link – On click it would take to Login page

Account Link – On click it would take to Account History page which displays all the shopping carts details by the logged in user.

Cart Link – On click it would take to Shopping Cart screen which displays the cart details of the current active Shopping Cart. If no active cart then display message "No Active Cart found."

Main content section: It would display generic Welcome message giving overview of the site. Also it would display links/images of the most popular product items, on click of popular item link the Product Detail screen would be displayed showing the product description, amount and option of adding the item to the cart. Also a link to Category Page should be displayed, which will display all the categories and allow the user to select the product from that specific category.

iv. **Category View and select product**: This screen allows the customers to see all the categories and select the category to purchase the product from that

category. Depending upon the selection, it should go to Product Details screen displaying the details about the products in that category.

v.   **Product Detail:** This screen allows the Customers to see the products from the selected category. It should provide the option to add select product to Cart.

vi.   **Shopping Cart:** It would display all the products added to cart.

Remove option for removing any of the products from the shopping cart.

Update Shopping cart – This link to be provided for updating the shopping cart (sub-total amount, selected items to be removed). On click of this link the page would be refreshed with the updated subtotal amount (i.e. Quantity * Price)

Final Check Out – This link would be used by the user to do the final check out (for confirmation and submission of the order)

In this course you need to develop the user interface using Angular 2+. The screens should include the fields as per the functionality mentioned above. Include Validations where ever required.

Further enhancement:

Any additional enhancements made will be appreciated.