**Steps taken to do my Assignment E Shopping Cart**

1. Created a new Empty project with ASP.NET Core Web Application.
2. Added services needed to Configure services, middleware to configure section and the default mapping end point into the startup.cs file.
3. Created Models folder and in it created two model classes – Item.cs and Category.cs. In order to process the data in models, repositories are needed. Since DI is used, Interfaces are created for each repository first. Then register the interfaces with the corresponding repository as a service and inject them into the application.
4. Created Interfaces – created Item Repository Interface(IItemRepository) and Category Repository Interface(ICategoryRepository) in Models folder with the relevant methods in them.
5. Created Repositories that will implement above methods and properties – Created Category Repository and hard coded a list in it since the DB is not used yet. Created Item Repository and hard coded a list of new items since the DB is not used yet and added relevant methods in it. Bothe these repositories were created in Models folder. Registered the Interfaces with the respective repositories as a service in startup.cs so that it can be used and injected anywhere in the application.
6. Created Item Controller. Created Views for Item as List.cshtml.
7. Created ViewModels to display data instead of View Bag.
8. Created Partial Views by creating the \_Layout and \_ViewStart views and creating the \_ViewImports file.
9. Installed Bootstrap and JQuery to the project. Added bootstrap, CSS and jquery to the \_Layout view. Added Bootstrap and made the list page more appealing.
10. Installed relevant packages for EF core and Sql server. Created the AppDbContext.cs which is used to manage the database and added necessary dbsets into it. Added the connection string to the JSON file. Edited the Startup.cs class to configure the appdbcontext and database connection string.
11. Modified the Item Repository and Category Repository to use the database Db sets when rendering Items and Categories.
12. Added migrations to the Project and also downloaded the tools package and design package to the project.
13. Seeded the initial data to the project by changing the AppDbContext.cs file using another Migration and fixed the layout of the page.
14. Created a method to view details of an item in the Item Controller and created the Details view for the Item. Added Tag helpers in the view imports file. Created the Details view with correct tag helpers and updated the item list view with a link which opens the details page.
15. Added a navigation bar with Home and All Items navigation sections.
16. Created Partial view that is used by multiple views but not globally shared. A partial view(\_itemCard.cshtml) was created for the listing all items since the same view is used when rendering all items and a section of items. Then that partial view(\_itemCard.cshtml) was called in the List view template.
17. Created the home controller. Created HomeViewModel which will get the Items on sale and render to the home controller which is displayed on the view.
18. Created a view for the home controller to send the data to the index page in home views. \_Carousel.cshtml was created as a partial view so that it is called to the home view. Edited the home page to show a slider using partial views and to show the items which are on sale.
19. Creating a shopping cart.

* Added a model for the shopping cart item in Models folder named ShoppingCart Item.cs with relevant properties and added that as a dataset in the Appdbcontext file. Then migrated it to the db.
* Created a model for the Shopping Cart itself in the Models folder as ShoppingCart. Created all the methods that is needed in the shopping cart in the Shopping Cart class. Added scoped configuration to register the shopping cart in the startup.cs file. Added HTTP Context and Session services to the configure services section in Startup.cs file. Brought the middleware to support the session in order to use session. Added necessary use session code in the configure section in startup.cs file.
* Created the shopping cart controller and added all the required actions in it.
* Created index View for the Chopping Cart to display the cart in a table.
* Added the add to cart button to the partial view - \_itemCard.cshtml under the shared folder. Updated the details page to have the add to cart button too.

1. Created a new folder for components and added the shopping cart component with relevant method. And then created default.cshtml views, under the shopping cart summary folder which was create under shared > Components folder. Then added this shopping cart summary view component in the \_Layout view so that it is accessible throughout the application.
2. Created a Category Menu component and added the corresponding folder called CategoryMenu and created the category menu view component – default.cshtml in it. Then in order to display a dropdown menu on the navigation bar to filter items by category, the view component view was added to the \_Layout page. Modified the List method in the Items controller to retrieve correct list of items according to the category chosen from the dropdown list.
3. Added a custom tag helper to create a contact us page. – Created a new controller called Contact Controller. Created a new folder for tag helpers and created an Email tag helper inheriting from tag helper. The added required code for the tag helper class and then added the custom tag helper in the \_ViewImports file. Created the corresponding Contact folder under views and in a view in that folder, added required contact content and the email tag helper. Updated the layout page to have the contact us link on the navigation bar.
4. Create a model for Order and added the required properties. Created an OrderDetail model with all the properties relating to an order. Added a reference to the dB context to the order and order details. Create an interface for the Order called IOrder Repository. Created Order Repository class. Registered IOrderRepository with Order Repository in startup.cs file.
5. Create the Order Controller. Created a new folder named Order in the Views folder and created a new view called Checkout and created the form the user fills when checking out. Added checkout button in the shopping cart with the relevant controllers.
6. Added checkout action for the post method in the Order Controller with the required validations. Created the checkout complete action in order controller and created the CheckoutComplete view in the Order folder under views. Added Input validation on the form inputs in the Order model class. Added Input validation error message displaying to the checkout view template.

**How to use the E Shopping Cart**