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

1. Created a .NET CORE Web Application Project and selected the MVC Application Template to start building the project – EshoppingCart.
2. Created models Item.cs and Category.cs with the relevant properties.’
3. Created a Repository folder and created two repositories for the two models created in the previous step.
4. Created Dummy Mocks(MockCategoryRepository.cs and MockItemRepository.cs) in a newly created Mocks folder to provide data to the application using the created Interfaces/Repositories.
5. Added service configurations and implementations for the above created interfaces and dummy mocks in the Configure Services section of the statup.cs file using the AddTransient method.
6. Created the first controller – ItemController.cs and added a method to return all the Items.
7. Created the first view to show the list of Items on the application.
8. Created the ViewModels folder and added the first view model for Items and passed Item content to the List.cshtml template using the ItemController.
9. Added the relevant model link to the \_ViewImports.cs file to import the ViewModels folder when rendering List.cshtml view template and deleted unnecessary imports from the List view template.
10. Added the bower file. Edited the MockItemRepository.cs file and List.cshtml file with correct configurations.
11. Added Bootstrap to List.cshtml and edited the \_Layout.cshtml file accordingly.
12. Installed Microsoft.Extentions.Configuration.Json Package and Microsoft.EntityFrameworkCore.Design Package. Added DbSets to the ApplicationDbContext.cs File.
13. Added a connection string to the Database in the appsettings.json file.
14. Created real Repositories for Item and Category in a new folder called “Repositories” and replaced the Mock repositories in the startup.cs file with the real repositories in the repository folder. (Nothing was changed in the controllers because Interfaces are used to inject, and the dependency injection returns the second parameter from the real repository configuration( In the AddTransient method in startup.cs) )
15. Seeded data to the database using the DbInitializer.cs file created under models folder.
16. Added Partial views for the Items.
17. Created the Shopping Cart model and added Microsoft.AspNetCore.Session, Caching memory and Microsoft.AspNetCore.Http.Extentions packages to the project. added the relevant configurations int the startup.cs file to configure sessions.
18. Created a Shopping Cart Controller. Created a Shopping Cart View Model in the View Models folder. Added a view – by creating a new folder ShoppingCart(Shopping cart controller name) and added a view in there named Index.cshtml. Then added HTML in it accordingly to display the cart.
19. The Home page was created by editing the Home controller and created a new Home View Model in the View Model folder. Edited the existing home view – index.cshtml. Used Partial views to display the Items in the Indes.cshtmml file under Home view folder.
20. Added a Carousel.cshtml file to the shared folder under Views folder in order to use Partial views.
21. Used View Components to display details of cart. Created a new folder for Components and created the first View Component, Shopping Cart Summary. Created Components folder inside the Views folder and created another new folder named Shopping Cart Summary in the Components folder and created the view in it and added required code.
22. Added the navigation bar by editing the existing nav bar.
23. Created Tag helpers Folder to enable server-side code and created the first tag helper – EmailTagHelper in it. Added the tag helper in the ViewImports file. Added a new controller for the Contact Page called ContactController and added necessary commands in it. Created a new folder in the Views section in the same name as Contact and created the index.cshtml view for the contact page.
24. Configured the routing when the user wants to get Items in a single category. Edited the Item Controller according to this and added configuration method in the startup.cs file to make the function work.
25. Updated the Navigation bar to the new style, tag helpers and bootstrap.
26. Added a dynamic menu item to the navigation bar for the category filter of the items. Created a view component – Category Menu. Created the relevant view component in the components folder under views folder and created the default.cshtml file in it with correct tag helpers and navigation styles. Then edited the navigation bar to add the Category drop down list.

**How to use the E Shopping Cart**