System Requirements Specification Index

For

Grocery Delivery Application (InMemory)

Version 4.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
fullstack@iiht.com

TABLE OF CONTENTS

GROCERY DELIVERY APPLICATION SYSTEM

System Requirements Specification

1. Business-Requirement:

1.1 Problem Statement:

The purpose of this application is to allow the visitors to view the product, search product by name, add address before placing order and see the order details or many more functions below are mentioned.

1.2 Following is the requirement specifications:

	Grocery Delivery Application
Home Controller	
1	View All Available Products.
2	View list of categories. (as menu bar)
3	Allow you to place an order.
4	Add address for order delivery while placing order
5	Show a list of orders placed by you.

2. Resources available:

2.1 GroceryDelivery:

Names	Resource	Remarks	Status
Package Structure/Project			
wwwroot	CSS, JS, Lib	Inside all these directory contains all styling logic code and javascripts file.	Already Implemented
controller	HomeController	Homecontroller handle all action method for respective user interface view(cshtml file)	Partially Implemented
Models	Error view model cs file	Contain all basic error definition	Already Implemented
Views	Home, Shared	All View .cshtml file for user interface.	Already Implemented

2.2 GroceryDelivery.BusinessLayer:

Names	Resource	Remarks	Status
Package Structure			
Interfaces	Interface directory contain all interface for Services class	Inside this directory contains all business logic code and CURD Operation Logic method.	Already Implemented
Services, Repository	IGroceryServices, GroceryServices, IGroceryRepository cs file for Method and business logic	Using this all cs file performed all CURD operation.	Partially Implemented
ViewModels	Cs file for represent all view entities	All view entities setting class	Already Implemented

2.3 GroceryDelivery.DataLayer:

Names	Resource	Remarks	Status
Package Structure			
DataGenerator	This cs class contain all dummy data for inMemory Operation	Create and initialize a basic data for application alive for demo.	Already Implemented
GroceryDbContext	Contain all business logic for data set and Dbset setting	Using this cs file performed all Data related settings operations.	Already Implemented

2.4 GroceryDelivery.Entities:

Names	Resource	Remarks	Status
Package Structure			
Entities Class	ApplicationUser, MenuBar, Product, ProductOrder	Contain all entities property for application	Already Implemented

2.5 Package: GroceryDelivery.Tests

Resources

Note: - Under the GroceryDelivery.Tests contain All Test cases for code evaluation, please don't try to alter and edit it.

3. Rest Endpoints

Rest End-points to be exposed in the controller along with method details for the same to be created

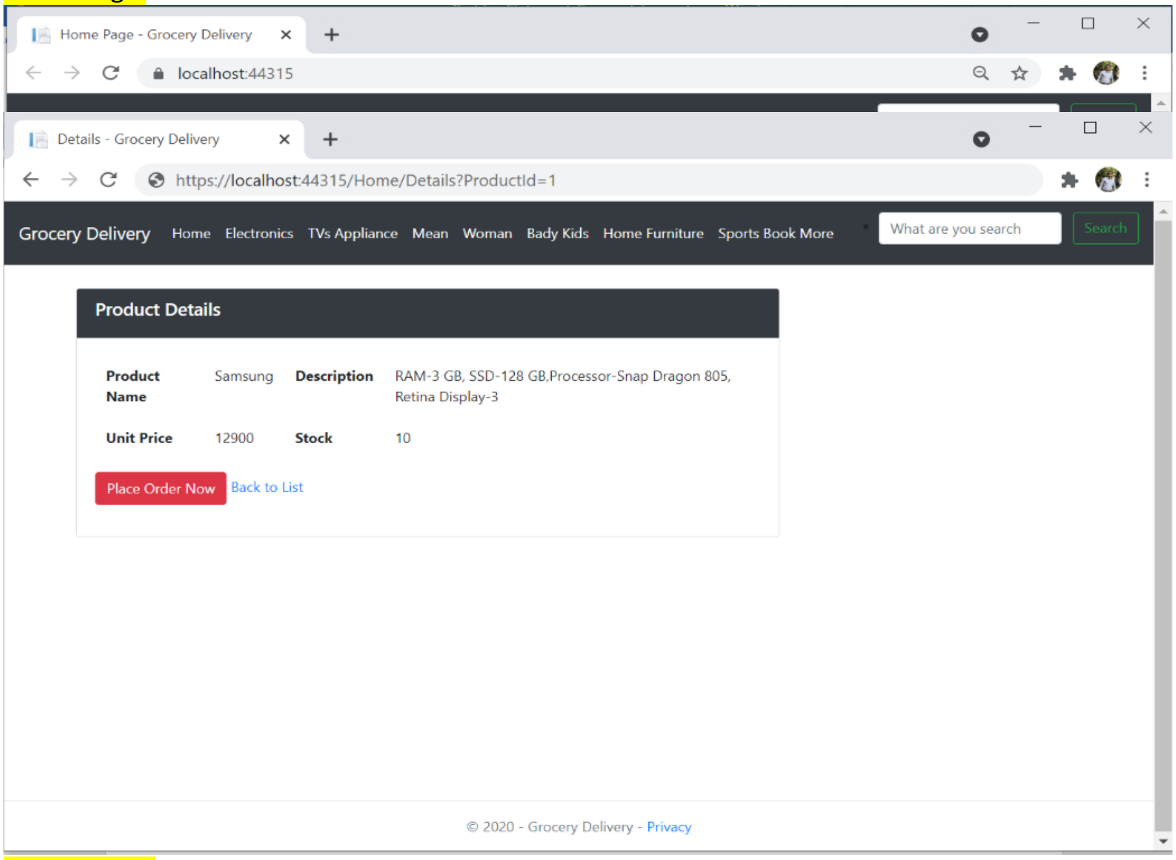
3.1 HomeController

URL Exposed		Purpose
/localhost:4431/		Show or Fetch all product
Http Method	GET	
Parameters	Int(Id)	
Return	<IEnumerable<Product>>	
/Home/Details?ProductId=1		Show all details of a product
Http Method	GET	
Parameter 1	ProductId	
Return	<Product>	
/Home/Placeorder?ProductId=1		Place an order for an item
Http Method	GET	
Parameter 1	ApplicationUser user	
Parameter 2	Int(ProductId)	
Return	<ApplicationUser>	
/Home/OrderInfo?userId=1		Order information page
Http Method	GET	
Parameter 1	Int (userId)	
Return	<IEnumerable<ProductOrder>>	
/?search=Samsung		Find a product or item
Http Method	GET	
Parameter 1	String(name)	
Return	<IEnumerable<Product>>	
/Home/Index/3		Go to respective category
Http Method	GET	
Parameter 1	Index	
Return	<Category>	

4.Expected wireframe:

Below are expected wireframes for creating the application GUI, you can also add new as expected.

Home Page.



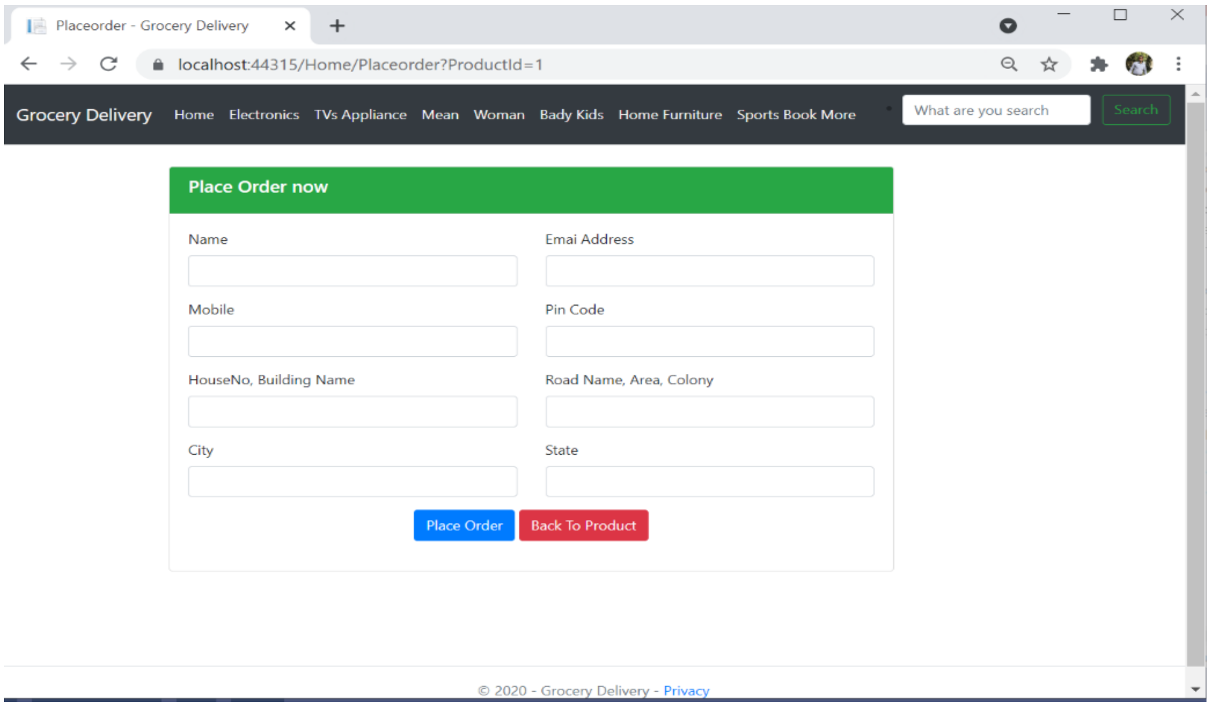
Product Details

Product Name	Samsung	**Description**	RAM-3 GB, SSD-128 GB,Processor-Snap Dragon 805, Retina Display-3
Unit Price	12900	**Stock**	10

© 2020 - Grocery Delivery - Privacy

Details Page.

Place Order Page

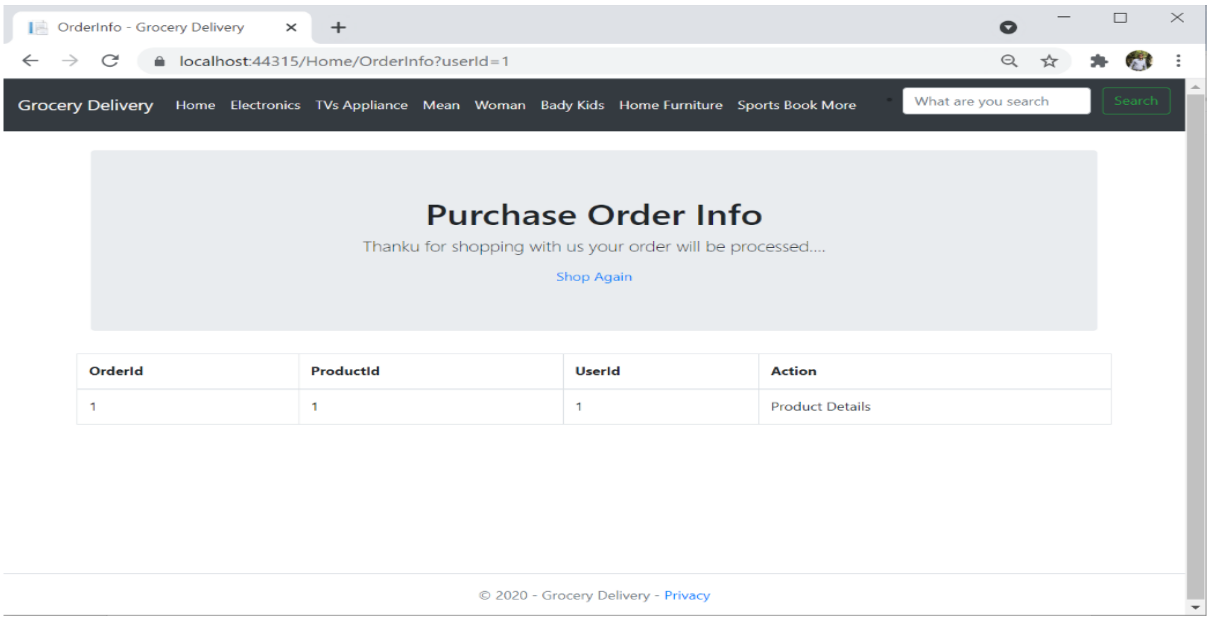


Place Order now

Name	Emai Address
Mobile	Pin Code
HouseNo, Building Name	Road Name, Area, Colony
City	State

© 2020 - Grocery Delivery - Privacy

OrderInfo Page


OrderInfo - Grocery Delivery

localhost:44315/Home/OrderInfo?userId=1

Grocery Delivery Home Electronics TVs Appliance Mean Woman Bady Kids Home Furniture Sports Book More

What are you search | Search

Purchase Order Info

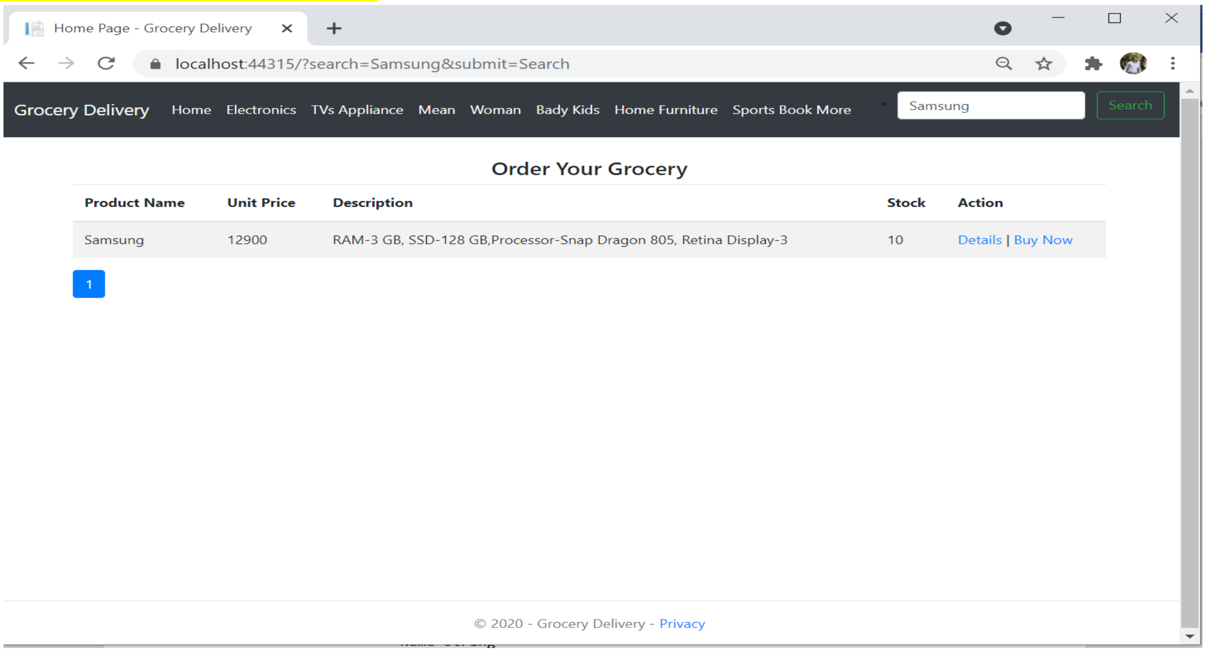
Thanku for shopping with us your order will be processed....

Shop Again

OrderId	ProductId	UserId	Action
1	1	1	Product Details

© 2020 - Grocery Delivery - Privacy

Search Page/Search Result page


Home Page - Grocery Delivery

localhost:44315/?search=Samsung&submit=Search

Grocery Delivery Home Electronics TVs Appliance Mean Woman Bady Kids Home Furniture Sports Book More

Samsung | Search

Order Your Grocery

Product Name	Unit Price	Description	Stock	Action
Samsung	12900	RAM-3 GB, SSD-128 GB,Processor-Snap Dragon 805, Retina Display-3	10	Details | Buy Now

1

© 2020 - Grocery Delivery - Privacy

5. Business Validations

5.1 Application User Entity:

- UserId Int
- Name string
- Email string
- MobileNumber long
- PinCode long
- HouseNo_Building_Name string
- Road_area string
- City string
- State string

5.2 Menubar Entity:

- Id int
- Title string
- Url string
- OpenInNewWindow bool

5.3 Product Entity:

- ProductId int
- ProductName string
- Description string
- Amount Double
- Stock int
- CatId int
- Photo string

5.4 Product Order Entity:

- OrderId int
- Product Id
- UsrId int

5.5 Common Constraints:

- Following validation constraints are to be added :
- All the value for Address Details Under the ApplicationUser: must not be null and min of 4 chars.
- All the category/Menu Bar value: must not be null and min of 3 chars
- All the product: must not be null and min of 3 chars.
- All the ProductOrder: must not be null.

- For all receiving Url parameter, validation check must be done and must throw custom exception if data is invalid
- Must not go and touch the test resources, as they will be used for Auto-Evaluation.
- **Database used is Embedded InMemory with pre-written entities data and value**.

6. Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**

2. **To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) Terminal → New Terminal.**

3. **On command prompt, cd into your project folder (cd <Your-Project-folder>).**

4. **To build your project use command:**
   (GroceryDelivery / dotnet build)

5. **To launch your application, Run the following command to run the application:**
   (GroceryDelivery / dotnet run)

6. **This editor Auto Saves the code.**

7. **To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.**

8. **To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

9. **To run the test cases in CMD, Run the following command to test the application:**
   (GroceryDelivery / dotnet test --logger "console;verbosity=detailed")
   (You can run this command multiple times to identify the test case status,

   and refactor code to make maximum test cases passed before final submission)

10. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B - command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.

11. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

12. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.
