**Project Title: Medicare**

DESCRIPTION

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

Medicare is a company that supplies medicines and a couple of other healthcare essentials at an affordable price. It was established in 2012 in Delhi, India. It had been serving fine all these years, however, the business analysts noticed a decline in sales since 2017. They found out that online ordering of medicines with companies, such as 100mg and mfine are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.  
You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

**Features of the application:**

1. Registration
2. Login
3. Payment gateway
4. Searching
5. Filtering
6. Sorting
7. Dynamic data
8. Responsive and compatible with different devices.

**Product Backlog:**

1. Create database and tables.
2. Initialize a Spring Boot project for the Back-End side.
3. Create REST APIs with spring Data JPA Repositories
4. Create a new Angular project for the Front-End side.
5. Create login and register pages.
6. Show all products to the home page.
7. Create a product details component.
8. Search a product by a category.
9. Search a product by a keyword.
10. Add products to the cart.
11. Show user’s cart.
12. Remove a product from the cart.
13. Update user account
14. Create the admin view
15. Delete a product for the admin
16. Add a new product for the admin
17. Add bootstrap and font awesome to the components.
18. Debug and test the project.

**Technologies and tools Used**

1. Angular: used in the front-end side to build modern single-page applications
2. Spring Boot: used in the back-end side to create the REST API and retrieve data from a database.
3. HTML/CSS: to create and format the content of the pages.
4. Bootstrap: to use some CSS and JavaScript designs.
5. Maven: to manage the project.
6. Visual Studio Code: to write and run the Angular code.
7. eclipse: to write and run the Spring Boot code.

**Core concepts used in the project.**

1. Object-Oriented: used to create and model objects for users and their credentials.
2. REST API: used to communicate between the back-end and the front-end sides.
3. Data Access Object: to abstract and encapsulate all access to the data source.
4. Object–Relational Mapping: to map the objects to the database.
5. Databases: used to store and retrieve data.
6. Data Sources: used to define a set of properties required to identify and access the database.
7. Collections: used some collections such Arraylist to store collection of data.
8. Exception Handling: used to catch problems that arises in the code especially in I/O blocks.

**How to run the program**

• clone project

o clone git : git clone <https://github.com/SwathiJI/Phase4.git>

• Import the “database\kitchen-story.sql” file to your database administration tool.

• Go to “Back-end\Kitchen-Story\src\main\resources\application.properties” file, open it.

• Edit some values of the database’ properties to be suit to your database administration tool.

• Run the back end project as a maven project:

o cd to your project “Back-end\Kitchen-Story”

o mvn compile

o mvn exec:java -Dexec.mainClass=com.simplilearn.KitchenStory

• Open another command line for the front-end part.

• cd to your project “Front-end-end\Kitchen-Story”

• install the following:

o npm install --save-dev

o npm install @angular/localize --save

o npm install bootstrap --save

o npm install font-awesome –save

• Run using ng serve –open

• It would be shown in <http://localhost:4200/>