Project Overview

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.

Technologies used

Technical Components for Front-end Development:

1. IDE: Visual Studio Code
2. Framework: AngularJS
3. Programming Language: TypeScript
4. HTML
5. CSS
6. Bootstrap

Technical Components for Back-end Development:

1. IDE: IntelliJ Idea
2. Framework: Spring or Spring Boot
3. Programming Language: Java
4. Web Server: Embedded Tomcat

Technical Components for Database Management:

1. Database: MySQL

DevOps and production technologies:

1. Git and GitHub
2.

Concepts Demonstration:

AngularJS Framework:

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly and succinctly. Its data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

Spring or Spring Boot Framework:

Spring Boot is an open-source Java-based Framework used to create a micro-Service. It is developed by the Pivotal team and is used to build stand-alone and production ready spring applications.

HTML

HTML stands for Hypertext Markup Language and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page are called Hypertext. As its name suggests, HTML is a Markup Language which means we use HTML to simply "markup" a text document with tags that tell a Web Browser how to structure it to display.

CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

Bootstrap

Bootstrap is the most popular front-end framework in the recent time. It is used to design templates for some interface components.

MySQL

MySQL is a fast, easy-to-use RDBMS. The MySQL database server, which manages database and tables, controls user access, and process SQL queries

MySQL-Bench- Benchmark and performance testing tools for the MySQL database server.

To connect Angular front-end application to Spring Boot back-end application REST API is used.

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

Admin Portal:

The admin portal deals with all the backend data generation and product information. The admin user should be able to:

1. Add or remove medicine details from the application to build a rich product line
2. Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices
3. Enable or disable a medicine product

User Portal:

It deals with the user activities. The end-user should be able to:

1. Sign-in to the application to maintain a record of activities
2. Search for products based on the search keyword
3. Apply filters and sort results based on different medicines to get the best deals
4. Add all the selected medicine items to the cart and customize the purchase at the end
5. Perform a seamless payment gateway
6. Get an order summary details page once the payment is complete

Project Management

Client	**SimpliLearn**
Consultant	Dakamanbha Ryngkhlem
Application Name	Medicare
Application Phase	Prototype
Project Objective	To develop a healthcare web application with a rich and user-friendly interface
Project Code Duration	15 working Days
Project Start Date	31st Jan, 2023
Project Finish Date	20th Feb, 2023

Requirements Development

Planned Start Date	**31st Jan, 2023**	**Actual Start Date**	**31st Jan, 2023**
Planned Finish Date	2nd Feb, 2023	Actual Finish Date	2nd Feb, 2023

Code Development

Planned Start Date	**3rd Feb. 2023**	**Actual Start Date**	**3rd Feb, 2023**
Planned Finish Date	14th Feb, 2023	**Actual Finish Date**	16th Feb, 2023
Planned Number of Sprints	3	**Actual Number of Sprints**	3
Planned Sprint Duration	8 working Days	**ctual Sprint Duration**	10 working Days

Project Documentation

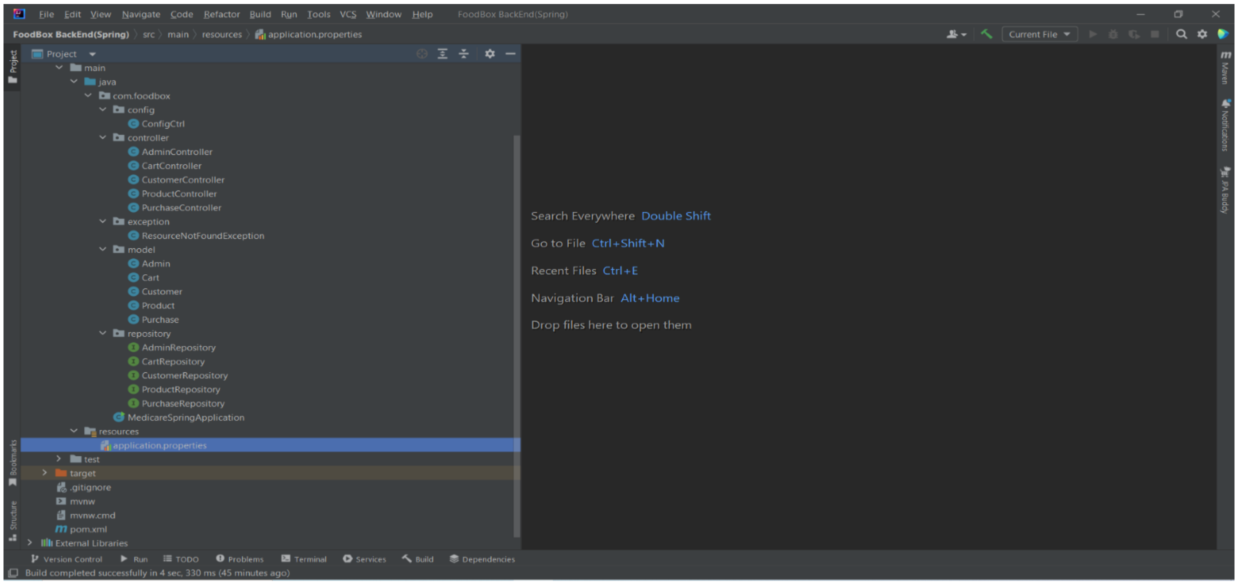
Planned Start Date	**14th Feb,2023**	**Actual Start Date**	**17th Feb,2023**
Planned Finish Date	16th Feb, 2023	**Actual Finish Date**	20th Feb, 2023

Project Summary

	Planned	**Actual**
Requirements Development	3 working days	3 working days
Code Development	8 working days	10 working days
Project Documentation	3 working days	2 working days
Total Project Days	**14 working days**	**15 working days**

Project Structure

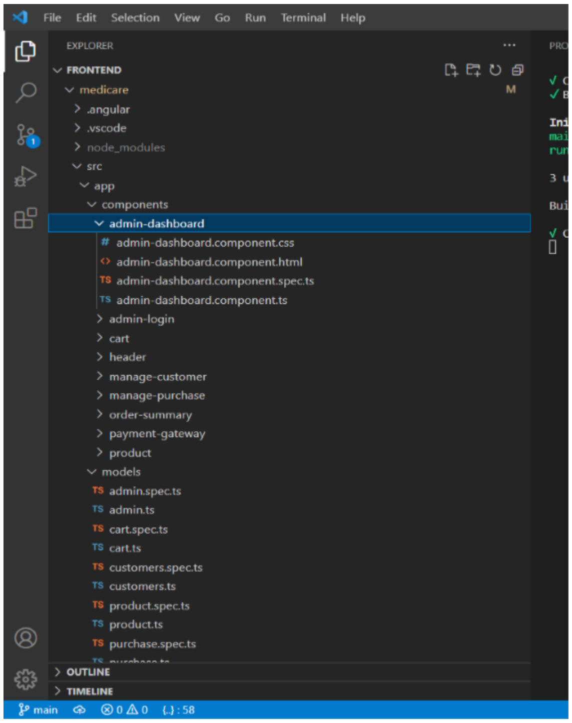
Back-end



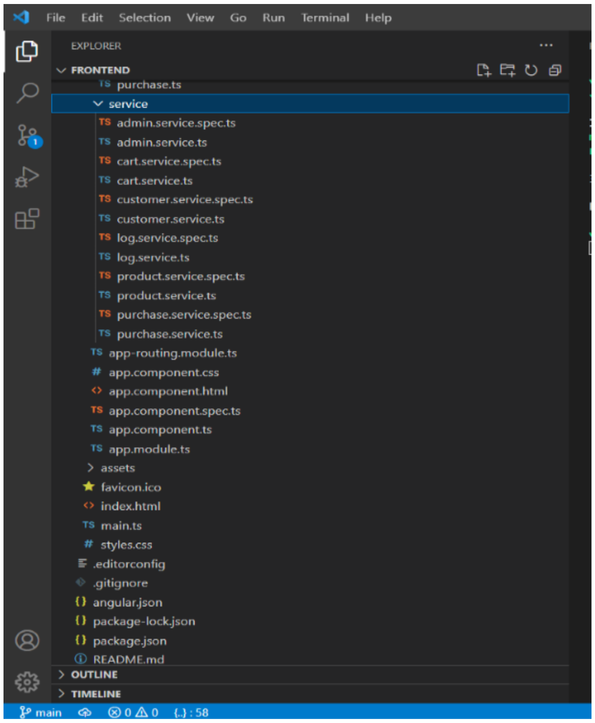
```
Project
  main
    java
      com.foodbox
        config
          ConfigCtrl
        controller
          AdminController
          CartController
          CustomerController
          ProductController
          PurchaseController
        exception
          ResourceNotFoundException
        model
          Admin
          Cart
          Customer
          Product
          Purchase
        repository
          AdminRepository
          CartRepository
          CustomerRepository
          ProductRepository
          PurchaseRepository
        MedicareSpringApplication
    resources
      application.properties
  test
target
.gitignore
mvnw
mvnw.cmd
pom.xml
External Libraries
```

Build completed successfully in 4 sec, 330 ms (45 minutes ago)

Front-End Structure



```
FRONTEND
  medicare                                M
    .angular
    .vscode
    node_modules
    src
      app
        components
          admin-dashboard
            admin-dashboard.component.css
            admin-dashboard.component.html
            admin-dashboard.component.spec.ts
            admin-dashboard.component.ts
          admin-login
          cart
          header
          manage-customer
          manage-purchase
          order-summary
          payment-gateway
          product
        models
          admin.spec.ts
          admin.ts
          cart.spec.ts
          cart.ts
          customers.spec.ts
          customers.ts
          product.spec.ts
          product.ts
          purchase.spec.ts
```



```
FRONTEND
          purchase.ts
        service
          admin.service.spec.ts
          admin.service.ts
          cart.service.spec.ts
          cart.service.ts
          customer.service.spec.ts
          customer.service.ts
          log.service.spec.ts
          log.service.ts
          product.service.spec.ts
          product.service.ts
          purchase.service.spec.ts
          purchase.service.ts
        app-routing.module.ts
        app.component.css
        app.component.html
        app.component.spec.ts
        app.component.ts
        app.module.ts
      assets
      favicon.ico
      index.html
      main.ts
      styles.css
    .editorconfig
    .gitignore
    angular.json
    package-lock.json
    package.json
    README.md
```

Back-end Rest API

Customers

1. To get customers by customer-email: http://localhost:8084/customers/{customerEmail}
2. To get all customers: http://localhost:8084/customers
3. To add User: http://localhost:8084/customers
4. To update User details: http://localhost:8080/update/{id}
5. To delete a User: http://localhost:8084/customers/{customerEmail}
6. To search for a customer by customer-email or customer-name or customer-contact or customer-address: http://localhost:8084/customers/search/{keyword}

Products

1. To get products: http://localhost:8084/products/cust
2. To add a product: http://localhost:8084/products
3. To update a product: http://localhost:8084/products/{id}
4. To delete a product: http://localhost:8084/products/{id}
5. To get a product by product by id: http://localhost:8084/products/{id}
6. To search for a product by a search term: http://localhost:8084/products/search/{keyword}
7. To get product by acidity: http://localhost:8080/products/acidity
8. To get product by painkillers: http://localhost:8080/products/painkillers
9. To get product by cold: http://localhost:8080/products/cold
10. To get product by headache: http://localhost:8080/products/headache

Cart

1. To add products to cart: http://localhost:8084/carts
2. To get products on cart: http://localhost:8084/carts
3. To update one item on cart: http://localhost:8084/carts/add/{id}
4. To update one item less on cart: http://localhost:8084/carts/minus/{id}
5. To delete an item in cart: http://localhost:8084/carts/{id}
6. To delete all items in cart: http://localhost:8084/carts

Admin

1. To get all products: http://localhost:8084/admin

Purchase

1. To get all purchased orders by transaction Id: http://localhost:8084/purchase
2. To add all purchase orders to the database: http://localhost:8084/purchase
3. To search for a purchase order: http://localhost:8084/purchase/search/{keyword}

Front-end Routes

1. Landing page: http://localhost:4200/products
2. Cart-page: http://localhost:4200/cart
3. Admin login page: http://localhost:4200/admin
4. Admin dash-board page: http://localhost:4200/admin-dashboard
5. Manage Customer page: http://localhost:4200/manageCustomer
6. Manage Purchase page: http://localhost:4200/managePurchase
7. Order Summary page: http://localhost:4200/orderSummary
8. Payment Gateway page: http://localhost:4200/paymentGateway

GitHub Repository

https://github.com/manbha03/CapStoneProject-Full-Stack-Java-Developer