Project Overview

Foodbox is a restaurant chain that delivers food items of different cuisines at affordable prices. It was established in 2014 in Bengaluru, India. It had been serving fine all these years; however, the business analysts noticed a decline in sales since 2016. They found out that the online ordering of food items with companies, such as Swiggy and Foodpanda were gaining more profit by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop an online food delivery web application with a rich and user-friendly interface.

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

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:

- Add or remove different cuisines to or from the application to build a rich product line
- Edit food item details like name, price, cuisine, description, and offers to keep it aligned to the current prices

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

- Sign-in to the application to maintain a record of activities
- Search for food items based on the search keyword
- Apply filters and sort results based on different cuisines to get the best deals
- Add all the selected food items to a cart and customize the purchase at the end
- Perform a seamless payment process
- 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 food ordering web application with a rich and user-friendly interface
Project Code Duration	15 working Days
Project Start Date	15th Feb, 2023
Project Finish Date	7th March, 2023

Requirements Development

Planned Start Date	**15th Feb, 2023**	**Actual Start Date**	**15th Feb, 2023**
Planned Finish Date	16th Feb, 2023	Actual Finish Date	16th Feb, 2023

Code Development

Planned Start Date	**17th Feb, 2023**	**Actual Start Date**	**17th Feb, 2023**
Planned Finish Date	28th Feb, 2023	**Actual Finish Date**	2nd March, 2023
Planned Number of Sprints	3	**Actual Number of Sprints**	3
Planned Sprint Duration	13 Working Days	**Actual Sprint Duration**	15 working Days

Project Documentation

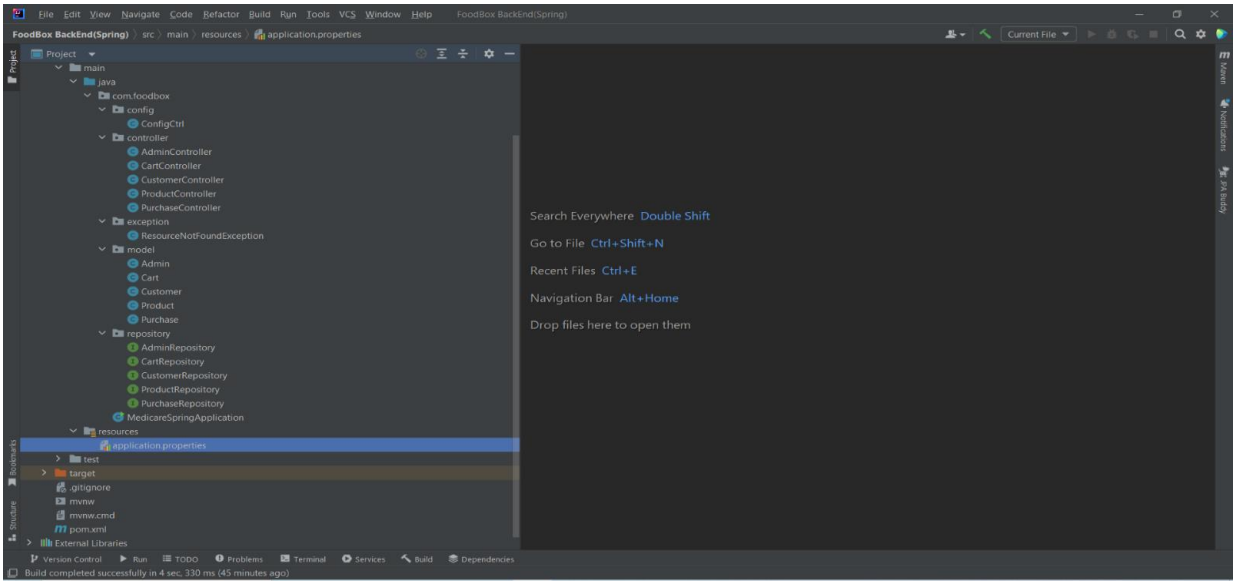
Planned Start Date	**1st March, 2023**	**Actual Start Date**	**3rd March, 2023**
Planned Finish Date	3rd March, 2023	**Actual Finish Date**	7th March, 2023

Project Summary

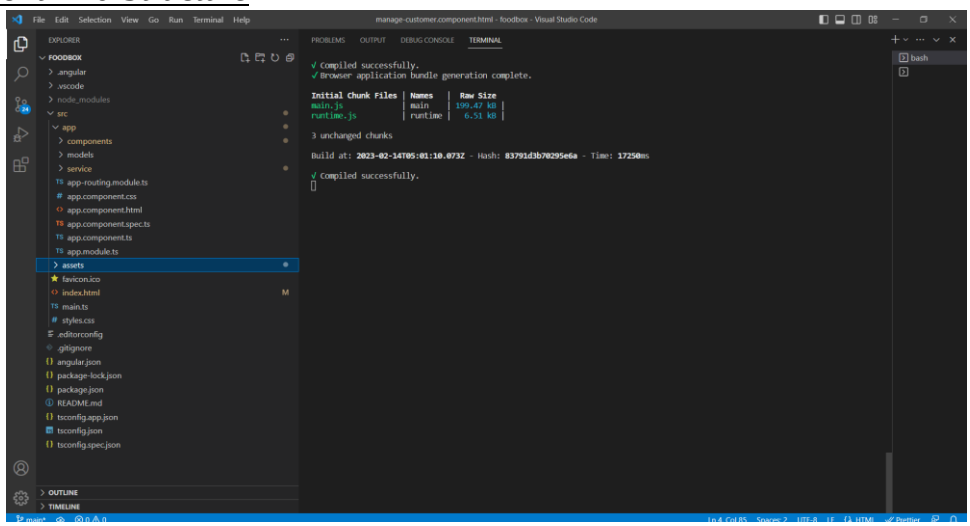
	Planned	Actual
Requirements Development	2 working days	2 working days
Code Development	8 working days	10 working days
Project Documentation	3 working days	3 working days
Total Project Days	**13 working days**	**15 working days**

Project Structure

Back-end



Front-End Structure



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 starter: http://localhost:8080/products/starter
8. To get product by main course: http://localhost:8080/products/maincourse
9. To get product by desert: http://localhost:8080/products/desert
10. To get product by beverages: http://localhost:8080/products/beverages

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.
6. Manage Customer page: http://localhost:4200/manageCustomer

7. Manage Purchase page: http://localhost:4200/managePurchase
8. Order Summary page: http://localhost:4200/orderSummary
9. Payment Gateway page: http://localhost:4200/paymentGateway

GitHub Repository

https://github.com/manbha03/JobReadiness-Project-1