31.05.2022

FOODBOX CAPSTONE PROJECT

Prepared by: PRIYANKA DAS

Project Objective :

Create a dynamic and responsive online food delivery web application for ordering food items of different cuisines from a restaurant.

Background of the problem statement :

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.
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.

The list of containers :

1. Create database and tables.
2. Add some rows and metadata to the tables
3. Initialize a Spring Boot project for the Back-End side.
4. Create REST APIs with spring Data JPA Repositories.
5. Create desired DAO methods for the Back-End side
6. Create a new Angular project for the Front-End side.
7. Create login and register pages and components.
8. Add cache to the login user
9. Logout user and remove cache
10. Show all products to the home page.
11. Show all products as cards.
12. Create a product details component.
13. Search a product by a category.
14. Search a product by a keyword.
15. Add products pages
16. Filter by page number
17. Sort product by different options
18. Add products to the cart.
19. Update total price in the cart status.
20. Show the payment gate and review the list
21. Add and remove products from the review list
22. Update the total price in the payment gate
23. Create the admin view
24. Update/Remove a product for the admin
25. Add a new product for the admin
26. Update the CSS design
27. Add bootstrap and font awesome to the components.
28. Debug and test the project.

Deployment :

1. Upload project to GitHub.
2. Create a t3.medium instance for master.
3. Create a t3.micro instance for slave.
4. Connect the two instances to the system
5. Create a Pipeline project on Jenkins.
6. Create a Jenkins file.
7. Generate a SSH key for GitHub.
8. Build the pipeline project.
9. Deploy the project.

Technologies and Tools Used :

- Angular: used in the front-end side to build modern single-page applications
- Spring Boot: used in the back-end side to create the REST API and retrieve data from a database.
- AWS EC2 instance: to use the instances as a VM and deploy the application
- AWS RDS: to upload the database online.
- Jenkins: to build the project from GitHub.
- GitHub: to upload the source code of the project.
- MobaXterm: to the instance from Windows OS.
- Selenium: for automation and testing.
- Apache: to use it as a web server.

Technologies and Tools Used :

- **HTML/CSS: to create and format the content of the pages.**
- **Bootstrap: to use some CSS and JavaScript designs.**
- **Maven: to manage the project.**
- **Visual Studio Code: to write and run the Angular code.**
- **IntelliJ: to write and run the Spring Boot code.**
- **MySQL: to use it as database management system.**
- **phpMyAdmin: to administrate and manage the database manually.**

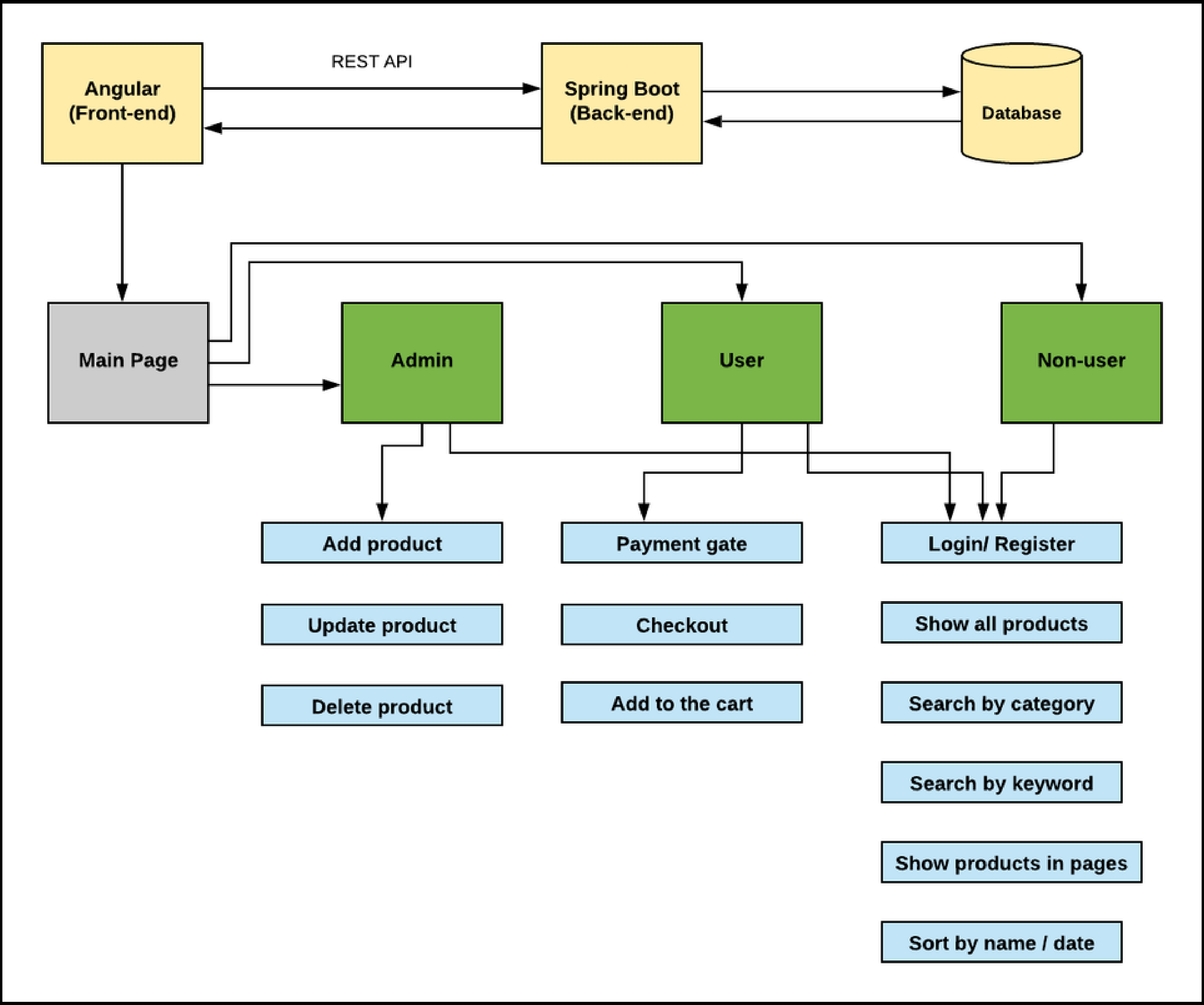
Core concepts used in the project :

- Object-Oriented: used to create and model objects for users and their credentials.
- REST API: used to communicate between the back-end and the front-end sides.
- Data Access Object: to abstract and encapsulate all access to the data source.
- Object–Relational Mapping: to map the objects to the database.
- Databases: used to store and retrieve data.
- Data Sources: used to define a set of properties required to identify and access the database.
- Collections: used some collections such Arraylist to store collection of data.
- Deployment: to deploy the local project to the end-users.
- Virtual Machine: use virtual instances to help to build, deploy and manage websites.
- Exception Handling: used to catch problems that arises in the code especially in I/O blocks.
- Single Web Page: apply the concept of a website that only contains one HTML page.

How to run the program locally :

- clone project clone git : git clone https://github.com/MujtabaMohsin/Foodbox
- Import the “\Back-End\foodbox\database\foodbox.sql” file to your database administration tool.
- Go to “\Back-End\foodbox\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: cd to your project “Back-end\foodbox” mvn compile mvn exec:java -Dexec.mainClass=com.simplilearn.foodbox
- Open another command line for the front-end part.
- cd to your project “Front-end-end\foodbox”.
- Run using ng serve –open.

FLOW CHART OF THE APPLICATION :



REST API
Angular (Front-end)
Spring Boot (Back-end)
Database
Main Page
Admin
User
Non-user
Add product
Update product
Delete product
Payment gate
Checkout
Add to the cart
Login/ Register
Show all products
Search by category
Search by keyword
Show products in pages
Sort by name / date