**CAPSTONE FOODBOX Project**

**DOCUMENTATION**

1. In this project, I have developed Restful Spring Boot E-Commerce portal FoodBox.com it uses internal memory database i.e., Spring H2 Data Base to store all the information about the food items, users who purchased it & this information can be fetched up by date & Customer name as well.
2. The Front end of this using Angular v8 as a JavaScript Framework.
3. Here, Admin can manage all the food items by categorizing them as food ID, food name, availability at the store & food price.
4. Admin can see the food ordered list filtered by date & category.

**Tools:**

1. Agile-Scrum-Jira
2. Spring Tool Suite 4
3. Spring boot initializer
4. Angular version 8.3.19
5. Postman
6. Git
7. GitHub

**Steps for Execution:**

First, Clone the project from the repository, FoodBox.com

GitHub Link:

As it i the RESTful API, it can also be verified on postman tool.

1. SWAGGER can also be used in spring boot RESTful API project so that the project documentation can be well understood and can easily be verified.
2. URL for H2 Database: **http://localhost:8080/h2**
3. URL for the SWAGGER documentation:

[**https://localhost:8080/swagger-ui.html**](https://localhost:8080/swagger-ui.html)

1. All the endpoints and example data will be present in that link and an option to try it out, Put the data into which will be in JSON format and simply try it out.

**Headers End-Point**

1. To add or update food items i.e.,

Post/Put food items----------------------------------- /food

1. Admin Registration------------------------------------- /registration
2. Admin Login---------------------------------------------- /login
3. Get/Delete food by Id---------------------------------- /food/{id}
4. To view all food items---------------------------------- /foods
5. Post/Put to add & update user ---------------------- /user
6. Get to get all users info. ----------------------------- /users
7. Get Users by:
8. Get users by userId ------------------------- /user/{userId}
9. Get users by username-------------------- /user/username/{userName}
10. Get users by userAge ---------------------- /user/userAge/{userAge}
11. Get/Delete users by id --------------------------------/user/{userId}

BY.

Kiran Kumar C