## horizontal line

Icon

Description automatically generated

FreshBox.com

A Java web app that is built using Spring Boot and deployed in Amazon EC2 using Jenkins CI/CD pipeline. It is a rich application that is built to enable customers to order food online and it can be managed by Restaurant admin.

**─**

Rajagopal Krishnasamy

rajagopal016@gmail.com

Project GitHub : <https://github.com/rajagopal016/FreshBoxFinal.git>

# Overview

This project is aimed to create a Spring Boot app that can act as ecommerce application for a restaurant. It provides a platform for users to browse through products of the restaurant and place order online. The website can be managed by an administrator of the restaurant. It is deployed in Tomcat using Jenkins CI/CD pipeline, hosted in AWS EC2 VM.

# Goals

1. Create a Spring Boot App, that can provide a platform for ecommerce operations of a restaurant
2. Create a AWS EC2 VM
3. Create a Jenkins pipeline that can perform CI and CD operations

# Specifications

1. A Java web app, that can act as platform for ecommerce operations of a restaurant

Tools Used:

* + 1. Java 17
    2. HTML and CSS
    3. Junit
    4. MySQL 8.0.26
    5. MySQL WorkBench 8.0 CE
    6. Apache Maven 3.8
    7. Spring Boot 2.6.1
    8. GitHub for version controlling the project <https://github.com/rajagopal016/FreshBoxFinal.git>

1. AWS EC2 VM was created to host the Jenkins pipeline run and to deploy the app in the Apache Tomcat Webserver. A Windows Server 2019 VM was created and the following SW were installed to realise the goals
   * 1. VM OS - Windows Server 2019
     2. Apache Tomcat 9.0
     3. Apache Maven 3.8.4
     4. Jenkins 2.325

# Milestones

## Sprint - 1

1. Understand the requirements of the project
2. Infrastructure requirements review and selection
3. Create AWS EC2 VM – Windows Server 2019
4. Install above mentioned software

## Sprint - 2

1. Database designing and ER Diagram
2. Class Designing
3. App Flow Designing
4. Start Jenkins(port 8080) and Tomcat(port 5053)

## Sprint - 3

1. HTML Designing of Admin Dashboard and login page, products, cuisine display management
2. Coding Server end for admin activities – controller designing, model making, and display
3. Perform final testing of the deployed app
4. Testing admin portal

## Sprint - 4

1. HTML Designing of User Portal – Home page, cart, filter and sort, user login and registration, purchase flow
2. Coding Server end for admin activities – controller designing, model making, and display
3. Perform final testing of the deployed app
4. Testing user portal
5. Documentation

Web Application Details :

Home Page : *http://localhost:5053 or http://localhost:5053/home or http://localhost:5053/index*

1. Display products
2. Option to filter and sort products
3. Facility to login and manage password for returning users
4. Registration facility for new users
5. Logged in users can add products to cart and purchase
6. Errors are properly handled

Graphical user interface, website

Description automatically generated

User Registration Page: *http://localhost:5053/userreg*

1. Provides form for registration
2. Mandatory fields will throw message, if user ignores
3. Errors are properly handled

Graphical user interface, text, application, email

Description automatically generated

User Login Page: *http://localhost:5053/userlogin*

1. Provides form for login
2. Mandatory fields will throw message, if user ignores
3. Login and user session is managed using cookies
4. Errors are properly handled

Graphical user interface, application

Description automatically generated

Home Page(after login):

1. Users can add products to cart
2. Users can filter and sort products

Graphical user interface, website

Description automatically generated

My Profile: *http://localhost:5053/myprofile*

Feature for the user to view his profile details

Graphical user interface, table

Description automatically generated

Cart page: [*http://localhost:5053/cart*](http://localhost:5053/cart)

1. Displays cart items
2. Facility to update quantity of items and delete items
3. Proceed to payment

Graphical user interface, application, table, Excel

Description automatically generated

Purchase Summary : [*http://localhost:5053/proceedToPurchase*](http://localhost:5053/proceedToPurchase)

1. Displays final summary to user before payment
2. Proceed to payment

Graphical user interface, text, application, Word

Description automatically generated

Payment Gateway : *http://localhost:5053/proceedToPayment*

1. Dummy payment gateway for representation
2. Provides facility for the user to choose desired payment method
3. Complete process

Graphical user interface, text, application

Description automatically generated

Order Summary:

1. Displays final order details to the user

Graphical user interface, text, application, Word, email

Description automatically generated

Admin Login Page : *http://localhost:5053/admin*

1. Provides form for admin login
2. Mandatory fields will throw message, if user ignores
3. Login and user session is managed using cookies
4. Errors are properly handled
5. For safety, admin side doesn’t have a link from the main web site and it can be accessed only by using <http://localhost:5053/admin>

Graphical user interface, application

Description automatically generated

Admin Dashboard : [*http://localhost:5053/adminlogin*](http://localhost:5053/adminlogin)

1. Provides centralized dashboard for the admin to manage products and cuisines

Graphical user interface, text, application

Description automatically generated

Product Management : [*http://localhost:5053/listProducts*](http://localhost:5053/listProducts)*?*

1. Lists all products, every column can be sorted
2. Products can be searched

Graphical user interface, text, application, email, Excel

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Add Food Item : *http://localhost:5053/addProducts*

1. Provides form for admin to add products
2. Mandatory fields will throw message, if user ignores
3. Errors are properly handled

Graphical user interface, text, application

Description automatically generated

1. Edit food items option provides facility to edit details of existing items

Graphical user interface, text, application, email

Description automatically generated

Cuisine Management : *http://localhost:5053/listCuisine?*

1. Lists all cuisines
2. Cuisines can be deleted

Graphical user interface, text, application, email

Description automatically generated

1. Facility to add new cuisines

Graphical user interface, text, application

Description automatically generated

Flow Chart :

Diagram

Description automatically generated

Diagram

Description automatically generated

Database Tables:

Text

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with low confidenceText

Description automatically generated

Classes :

A picture containing graphical user interface

Description automatically generated

Admin Controller : Performs admin login and admin dashboard control

Cuisine Controller : Cuisine display, add, delete

Eproduct Controller : Product display, edit, add, delete

Index Controller : Home page control – disply, sort, filter

Login Controller : User login, registration, signout, profile

Purchase Controller : Manages add to cart, cart operations, payment and order display

CuisineDAO, ProductDAO, UserDAO1 are DAO for respective DB tables

Admin, Cuisine, Product, User are entity models for respective tables

Graphical user interface, text

Description automatically generated

Above classes for junit test cases. Test cases are written for necessary function units

Text

Description automatically generated with low confidence

Above files are the html and image files for respective functions