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:

   i. Java 17
   ii. HTML and CSS
   iii. Junit
   iv. MySQL 8.0.26
   v. MySQL WorkBench 8.0 CE
   vi. Apache Maven 3.8
   vii. Spring Boot 2.6.1
   viii. GitHub for version controlling the project
   https://github.com/rajagopal016/FreshBoxFinal.git
2. 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
   i. VM OS - Windows Server 2019
   ii. Apache Tomcat 9.0
   iii. Apache Maven 3.8.4
   iv. Jenkins 2.325

Milestones

I. 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

II. Sprint - 2

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

III. 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

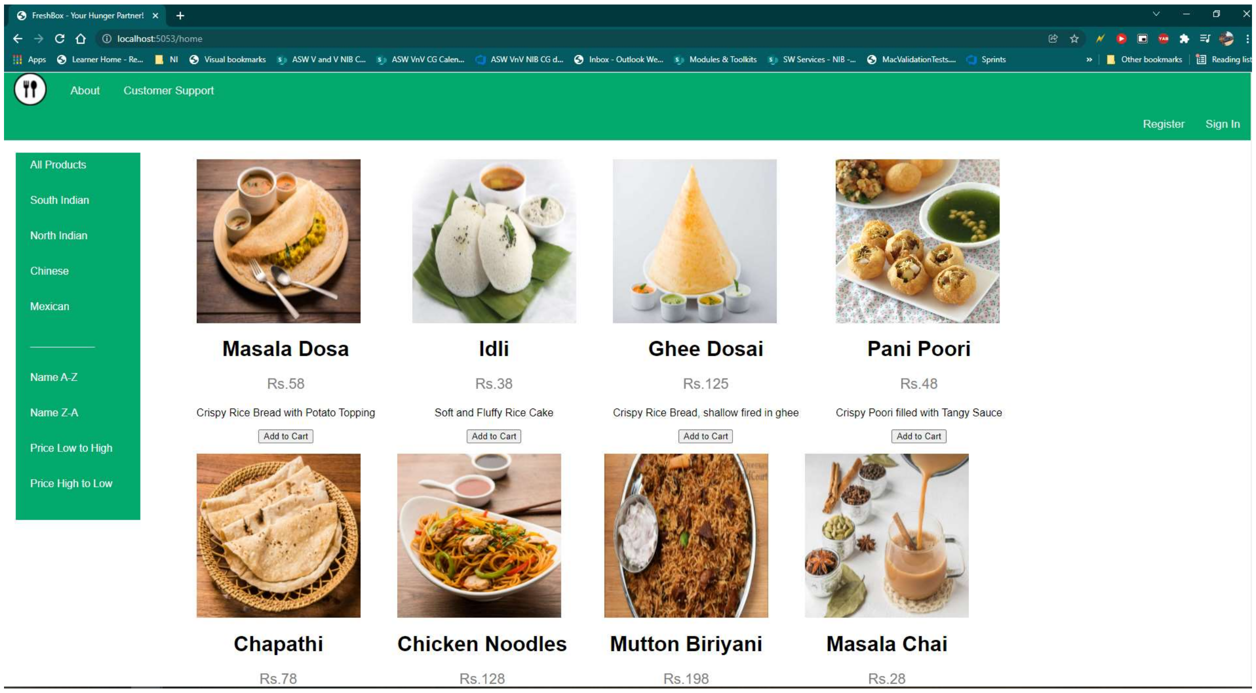
IV. Sprint - 4

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 :

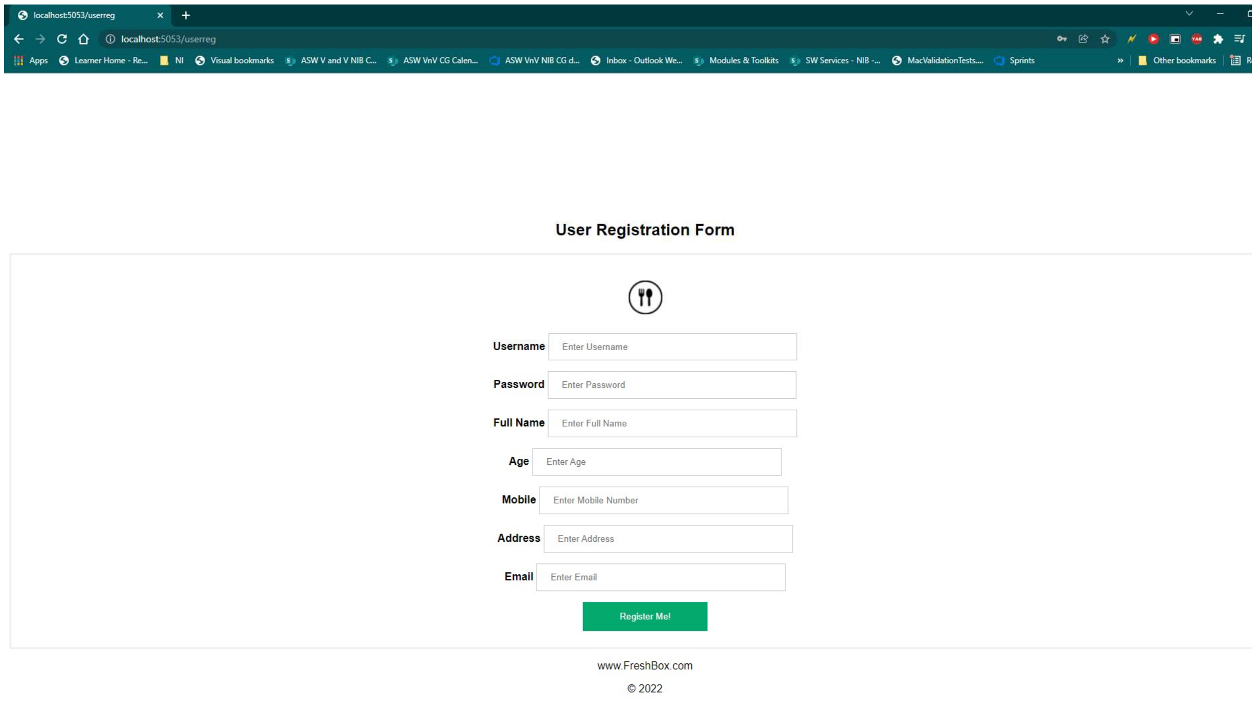
Home Page : *http://localhost:5053 or http://localhost:5053/home or http://localhost:5053/index*

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



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


localhost:5053/userreg

User Registration Form

Username: Enter Username
Password: Enter Password
Full Name: Enter Full Name
Age: Enter Age
Mobile: Enter Mobile Number
Address: Enter Address
Email: Enter Email

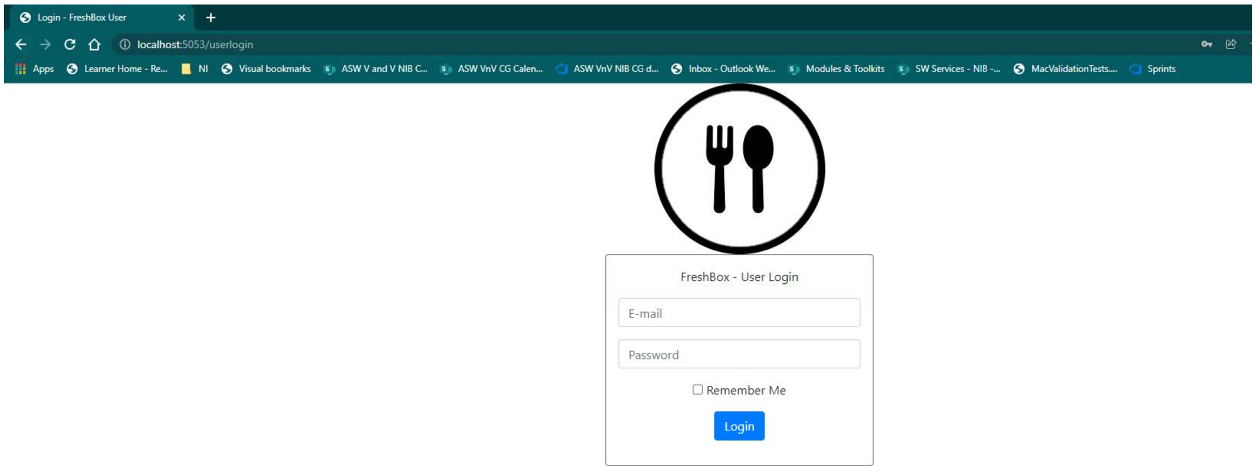
Register Me!

www.FreshBox.com

© 2022

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


Login - FreshBox User

localhost:5053/userlogin

FreshBox - User Login

E-mail

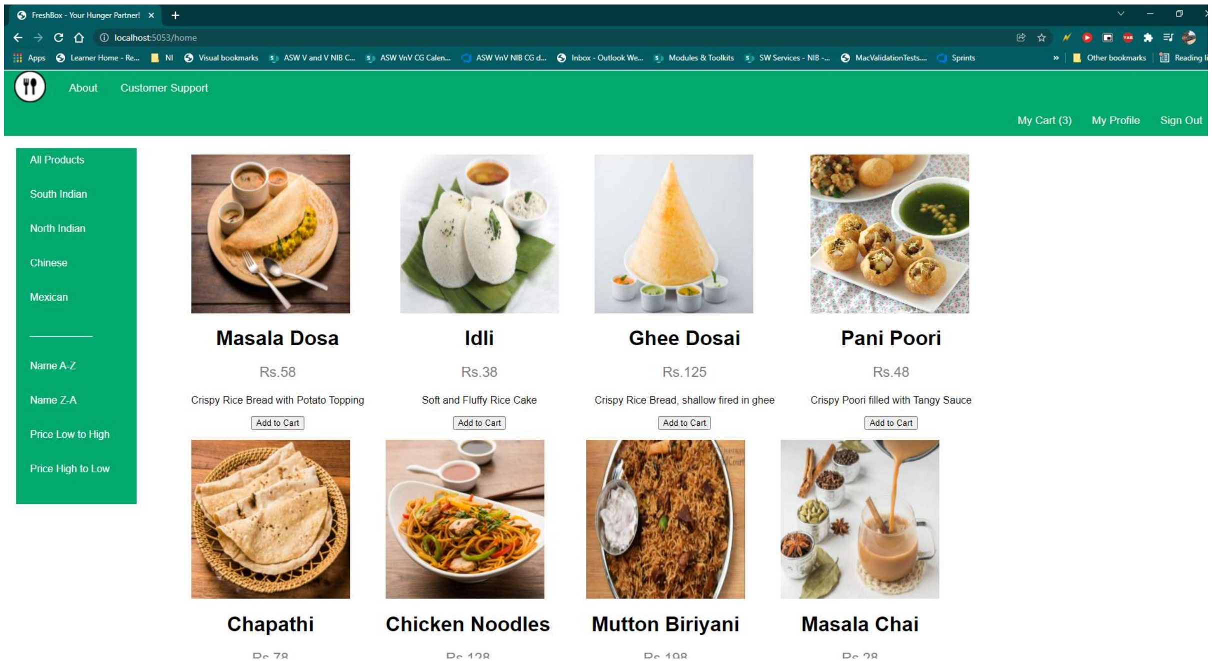
Password

Remember Me

Login

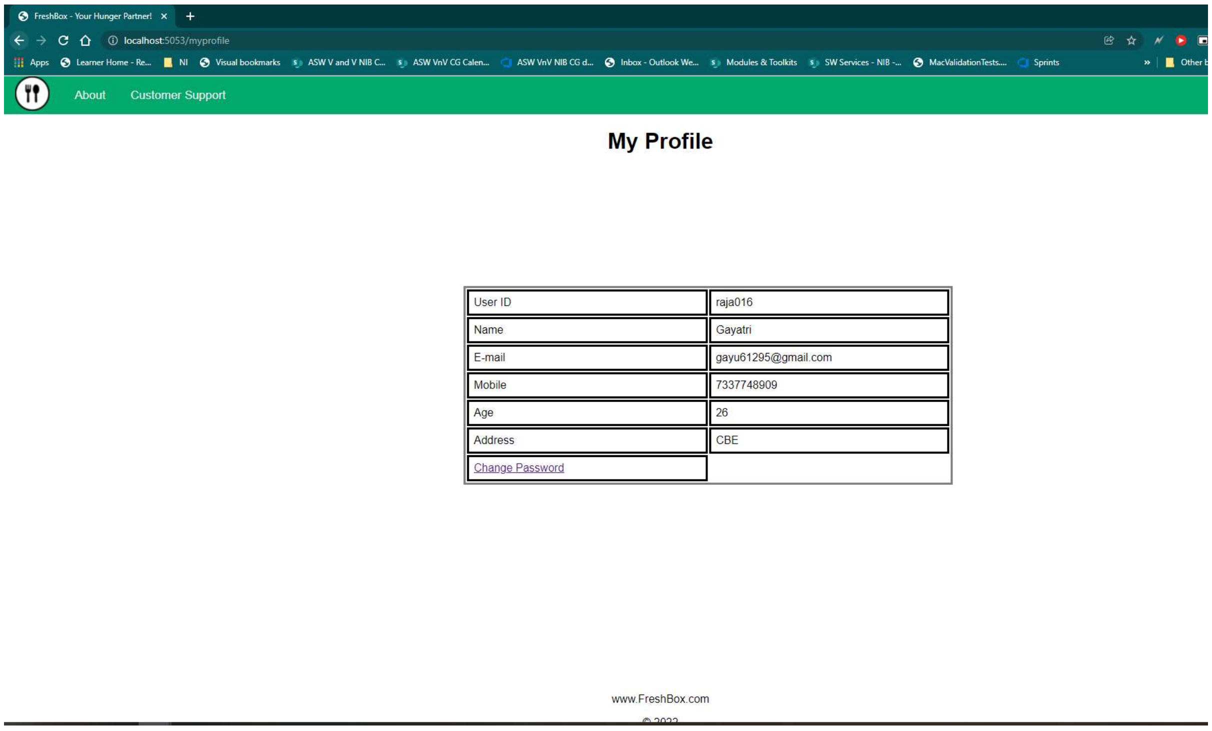
Home Page(after login):

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



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

Feature for the user to view his profile details



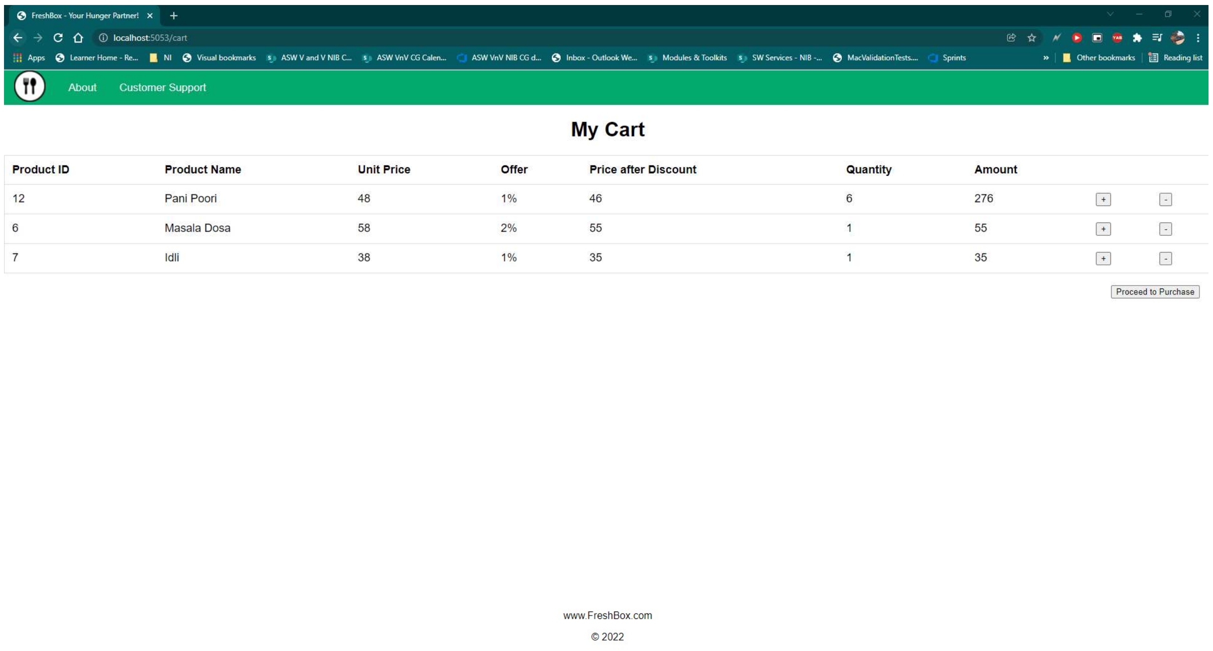
My Profile

User ID	raja016
Name	Gayatri
E-mail	gayu61295@gmail.com
Mobile	7337748909
Age	26
Address	CBE
Change Password	

www.FreshBox.com

Cart page: http://localhost:5053/cart

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



My Cart

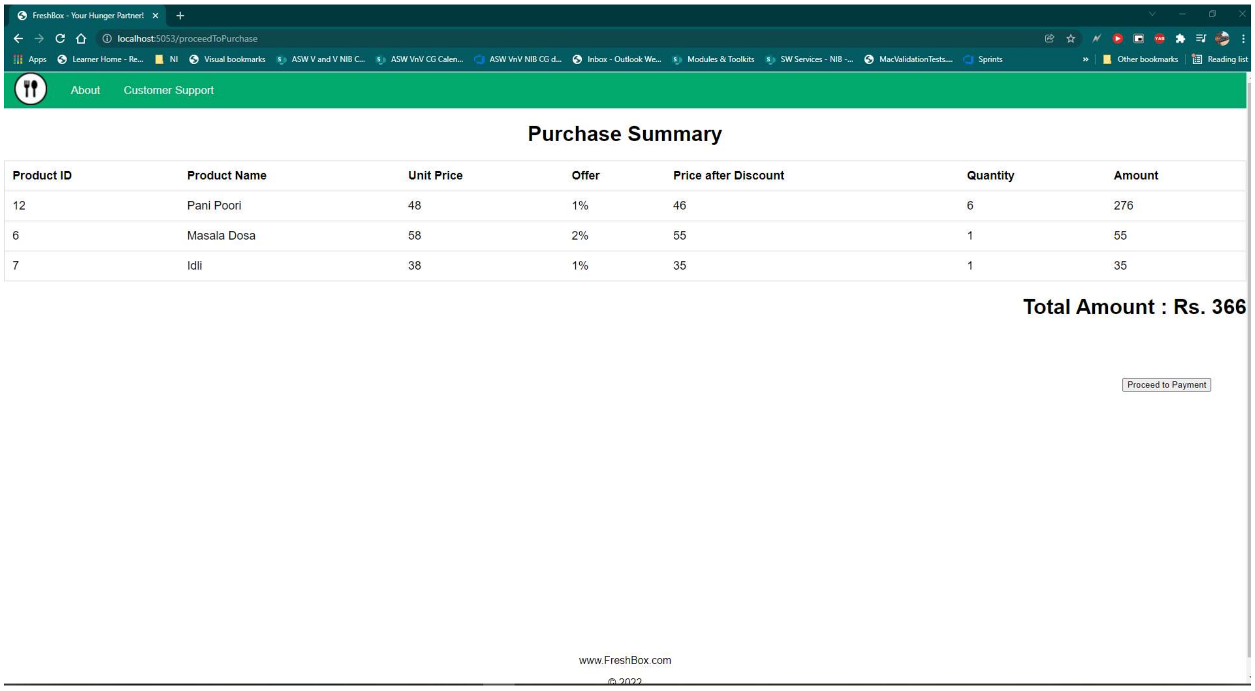
Product ID	Product Name	Unit Price	Offer	Price after Discount	Quantity	Amount		
12	Pani Poori	48	1%	46	6	276	+	-
6	Masala Dosa	58	2%	55	1	55	+	-
7	Idli	38	1%	35	1	35	+	-

www.FreshBox.com

© 2022

Purchase Summary : http://localhost:5053/proceedToPurchase

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


FreshBox - Your Hunger Partner!

localhost:5053/proceedToPurchase

About Customer Support

Purchase Summary

Product ID	Product Name	Unit Price	Offer	Price after Discount	Quantity	Amount
12	Pani Poori	48	1%	46	6	276
6	Masala Dosa	58	2%	55	1	55
7	Idli	38	1%	35	1	35

Total Amount : Rs. 366

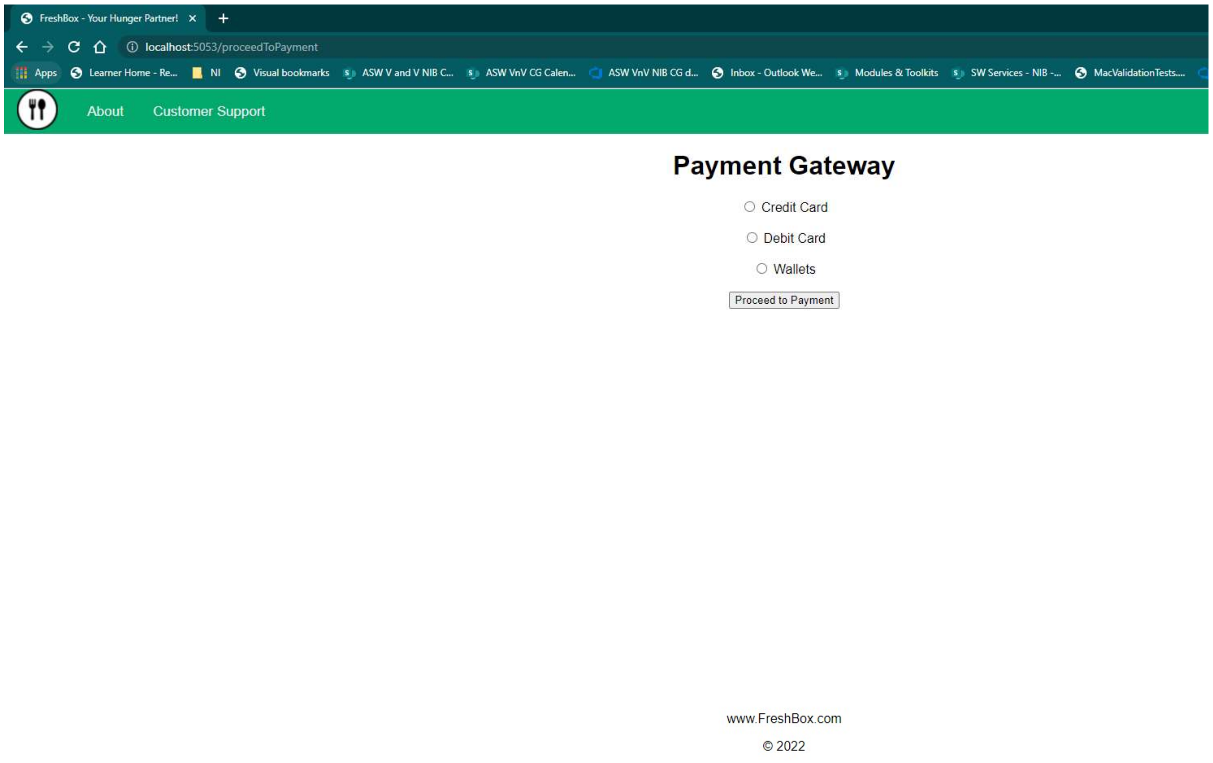
Proceed to Payment

www.FreshBox.com

© 2022

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


FreshBox - Your Hunger Partner!

localhost:5053/proceedToPayment

About Customer Support

Payment Gateway

- Credit Card
- Debit Card
- Wallets

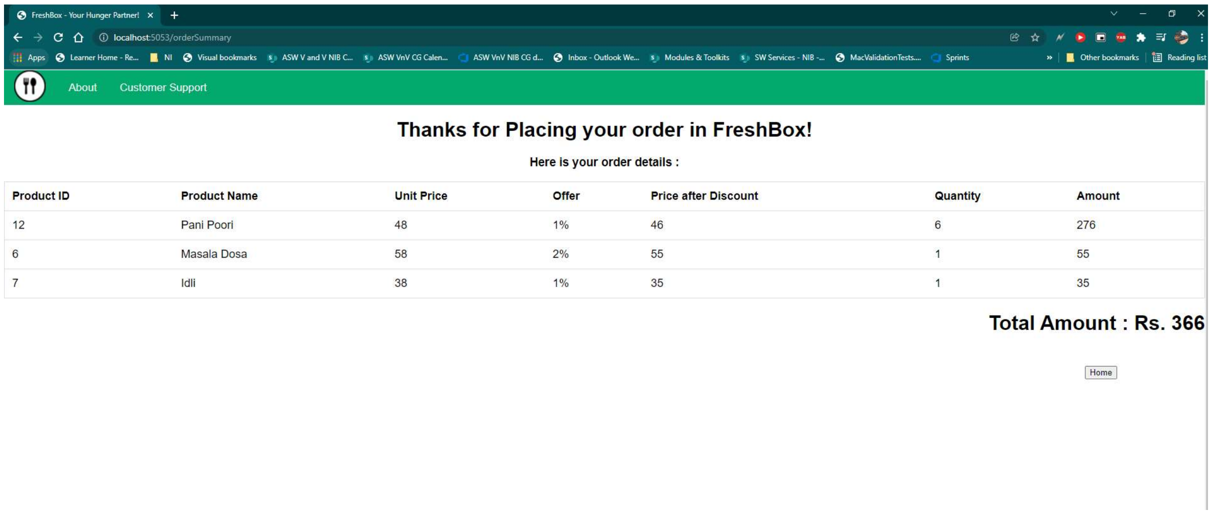
Proceed to Payment

www.FreshBox.com

© 2022

Order Summary:

1. Displays final order details to the user



Thanks for Placing your order in FreshBox!

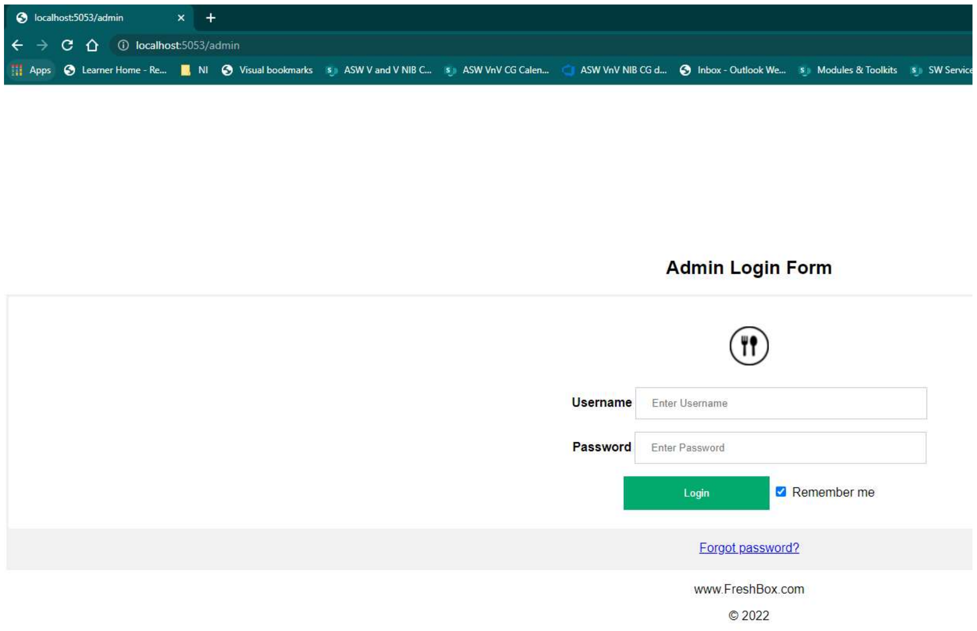
Here is your order details :

Product ID	Product Name	Unit Price	Offer	Price after Discount	Quantity	Amount
12	Pani Poori	48	1%	46	6	276
6	Masala Dosa	58	2%	55	1	55
7	Idli	38	1%	35	1	35

Total Amount : Rs. 366

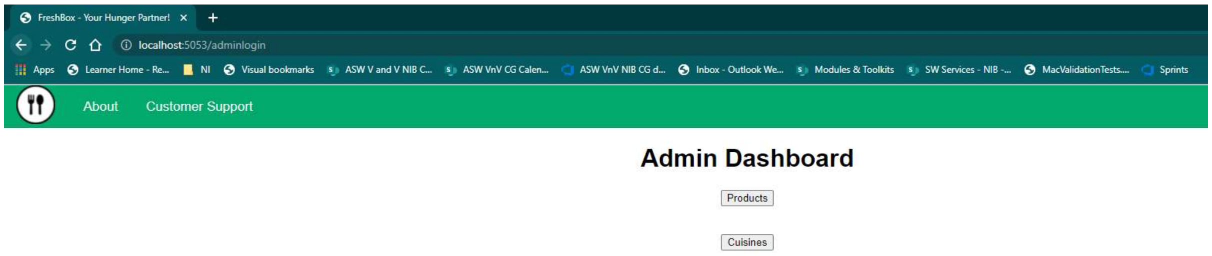
Admin Login Page : *http://localhost:5053/admin*

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



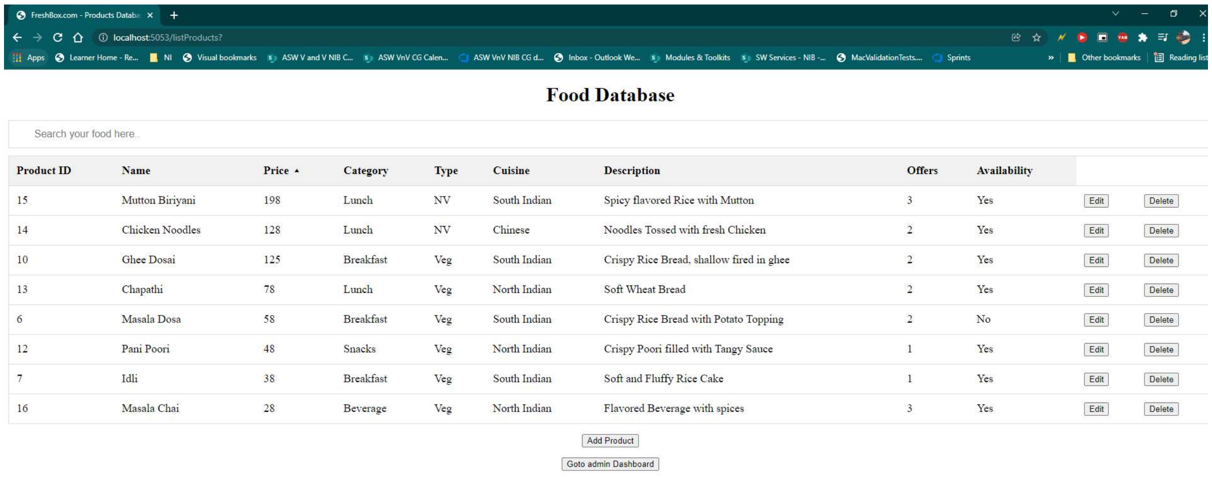
Admin Dashboard : *http://localhost:5053/adminlogin*

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



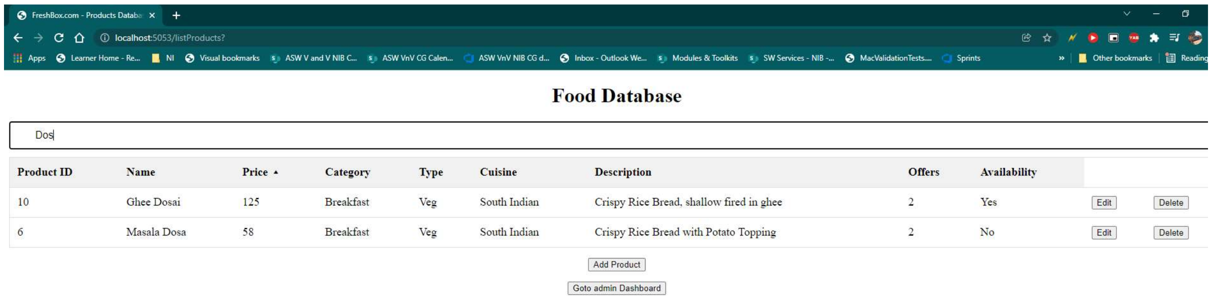
Product Management : *http://localhost:5053/listProducts*?

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


FreshBox.com - Products Database
localhost:5053/listProducts?
Search your food here..
Add Product
Goto admin Dashboard

Food Database

Product ID	Name	Price ▲	Category	Type	Cuisine	Description	Offers	Availability		
15	Mutton Biriyani	198	Lunch	NV	South Indian	Spicy flavored Rice with Mutton	3	Yes	Edit	Delete
14	Chicken Noodles	128	Lunch	NV	Chinese	Noodles Tossed with fresh Chicken	2	Yes	Edit	Delete
10	Ghee Dosai	125	Breakfast	Veg	South Indian	Crispy Rice Bread, shallow fired in ghee	2	Yes	Edit	Delete
13	Chapathi	78	Lunch	Veg	North Indian	Soft Wheat Bread	2	Yes	Edit	Delete
6	Masala Dosa	58	Breakfast	Veg	South Indian	Crispy Rice Bread with Potato Topping	2	No	Edit	Delete
12	Pani Poori	48	Snacks	Veg	North Indian	Crispy Poori filled with Tangy Sauce	1	Yes	Edit	Delete
7	Idli	38	Breakfast	Veg	South Indian	Soft and Fluffy Rice Cake	1	Yes	Edit	Delete
16	Masala Chai	28	Beverage	Veg	North Indian	Flavored Beverage with spices	3	Yes	Edit	Delete



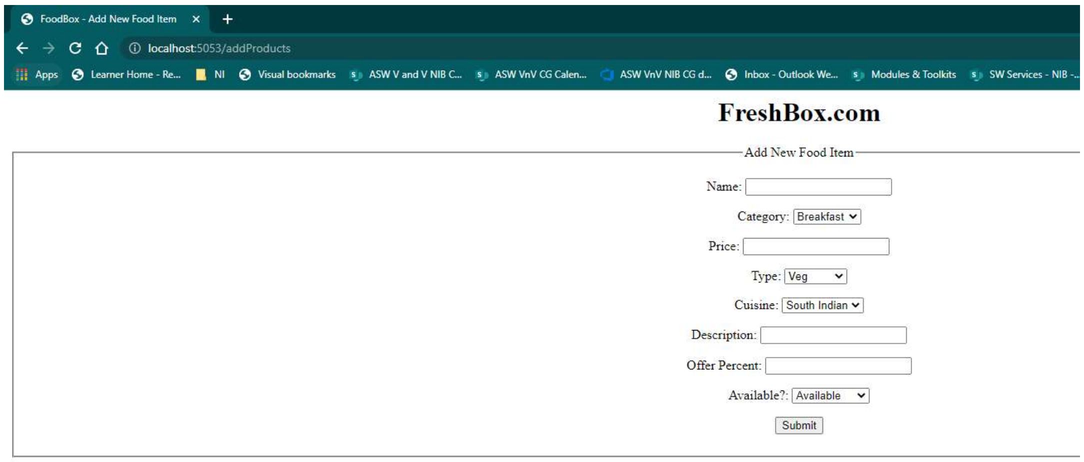
Food Database

Dos

Product ID	Name	Price ▲	Category	Type	Cuisine	Description	Offers	Availability		
10	Ghee Dosai	125	Breakfast	Veg	South Indian	Crispy Rice Bread, shallow fired in ghee	2	Yes	Edit	Delete
6	Masala Dosa	58	Breakfast	Veg	South Indian	Crispy Rice Bread with Potato Topping	2	No	Edit	Delete

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

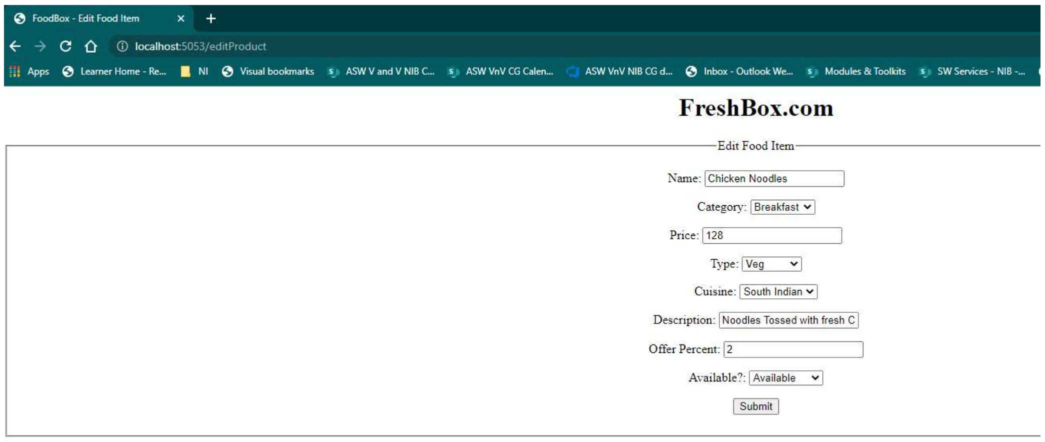


FreshBox.com

Add New Food Item

Name:	
Category:	Breakfast
Price:	
Type:	Veg
Cuisine:	South Indian
Description:	
Offer Percent:	
Available?:	Available

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



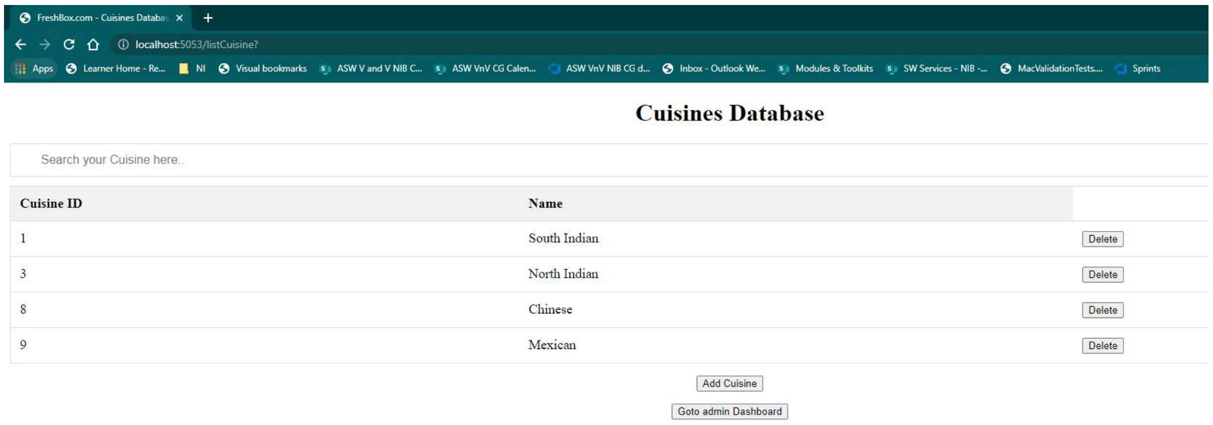
FreshBox.com

Edit Food Item

Name:	Chicken Noodles
Category:	Breakfast
Price:	128
Type:	Veg
Cuisine:	South Indian
Description:	Noodles Tossed with fresh C
Offer Percent:	2
Available?:	Available

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

1. Lists all cuisines
2. Cuisines can be deleted

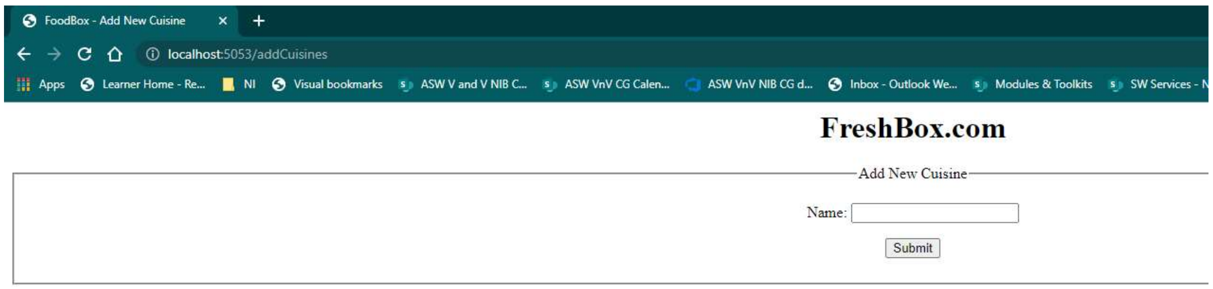


Cuisines Database

Search your Cuisine here..

Cuisine ID	Name	
1	South Indian	Delete
3	North Indian	Delete
8	Chinese	Delete
9	Mexican	Delete

3. Facility to add new cuisines

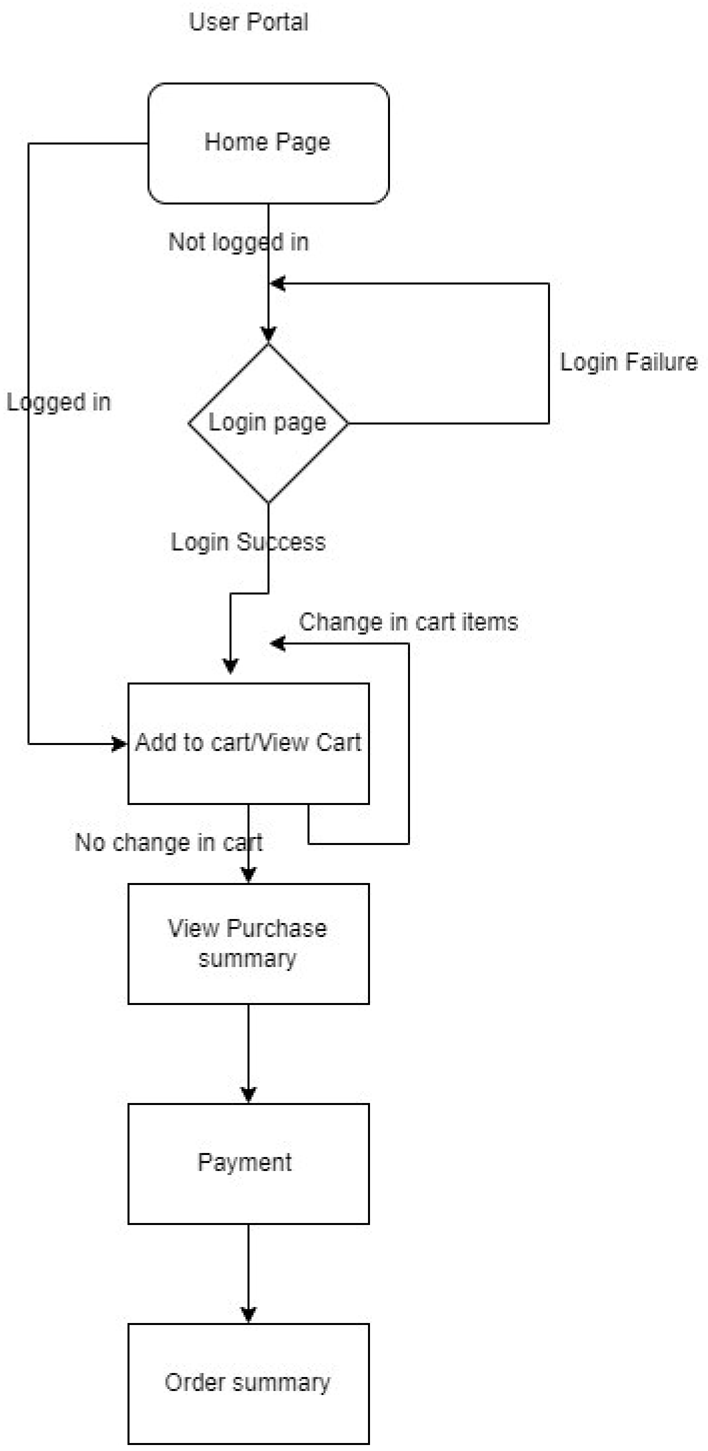


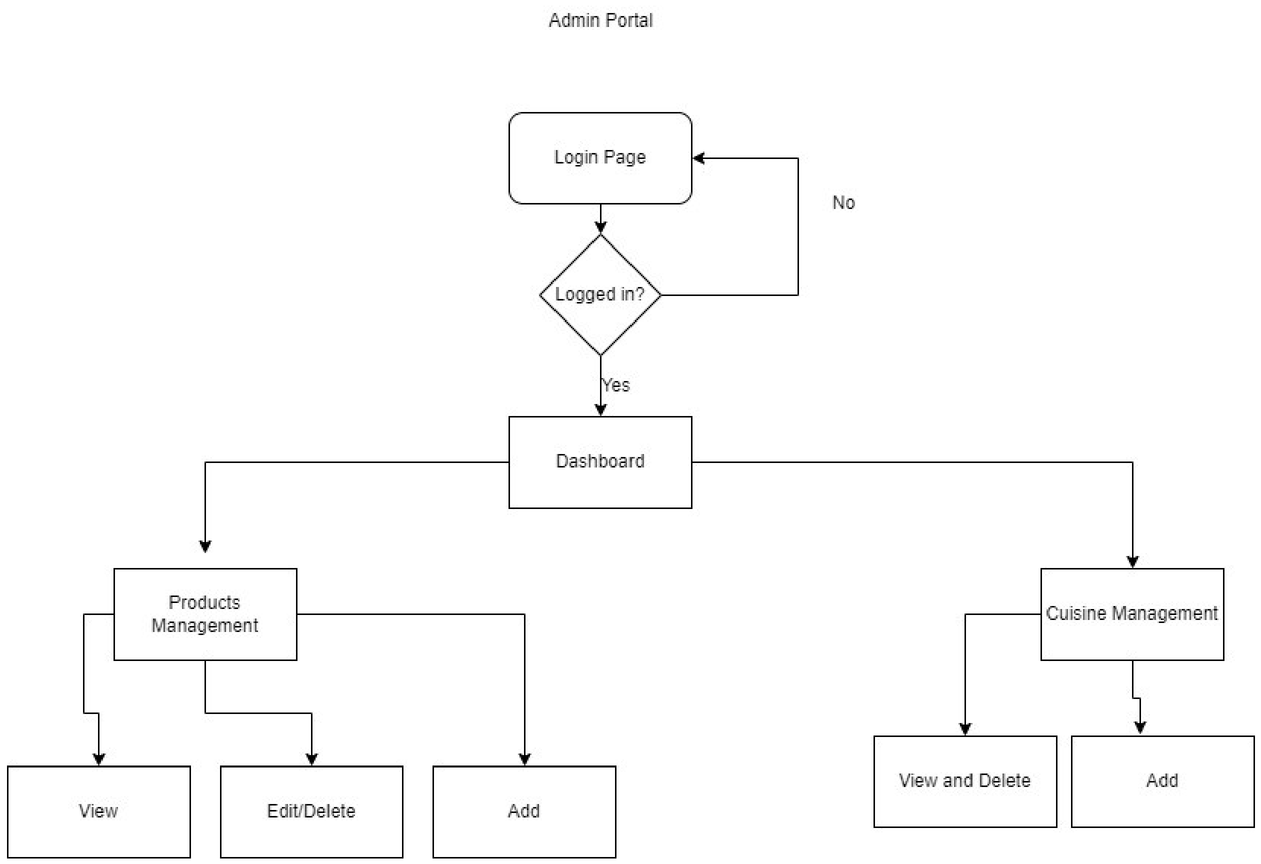
FreshBox.com

Add New Cuisine

Name:

Flow Chart :


User Portal
Home Page
Not logged in
Login Failure
Logged in
Login page
Login Success
Change in cart items
Add to cart/View Cart
No change in cart
View Purchase summary
Payment
Order summary



Database Tables:

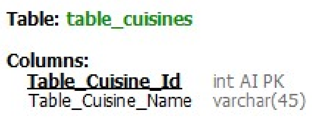


Table: table_cuisines

Columns:

Table_Cuisine_Id	int AI PK
Table_Cuisine_Name	varchar(45)

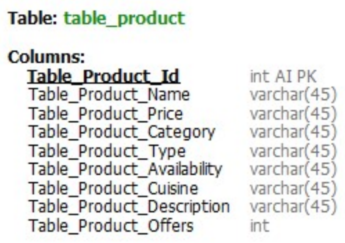


Table: table_product

Columns:

Table_Product_Id	int AI PK
Table_Product_Name	varchar(45)
Table_Product_Price	varchar(45)
Table_Product_Category	varchar(45)
Table_Product_Type	varchar(45)
Table_Product_Availability	varchar(45)
Table_Product_Cuisine	varchar(45)
Table_Product_Description	varchar(45)
Table_Product_Offers	int

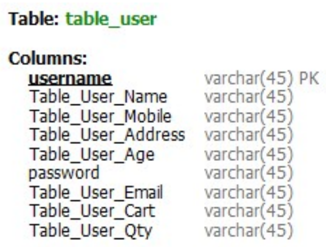
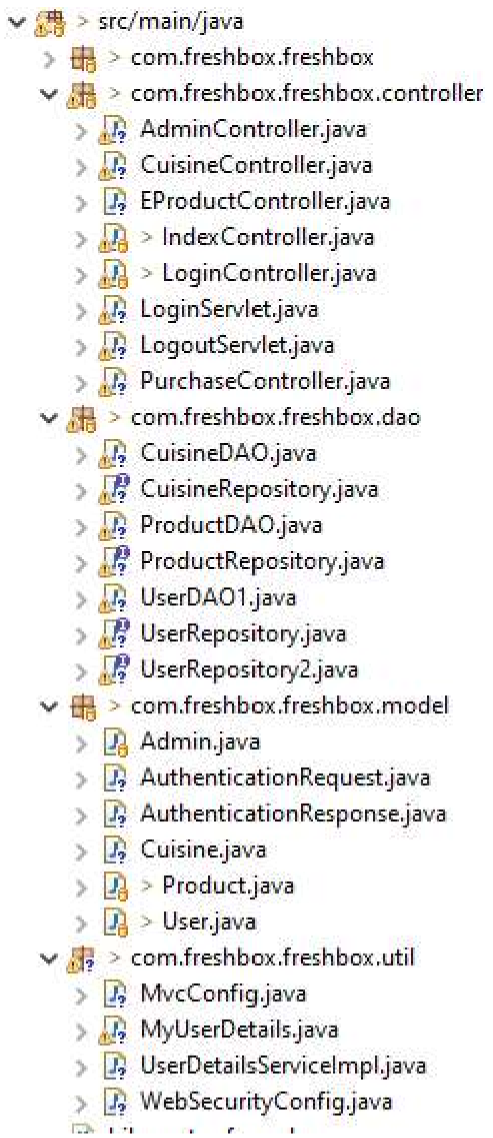


Table: table_user

Columns:

username	varchar(45) PK
Table_User_Name	varchar(45)
Table_User_Mobile	varchar(45)
Table_User_Address	varchar(45)
Table_User_Age	varchar(45)
password	varchar(45)
Table_User_Email	varchar(45)
Table_User_Cart	varchar(45)
Table_User_Qty	varchar(45)

Classes :



```
src/main/java
  com.freshbox.freshbox
  com.freshbox.freshbox.controller
    AdminController.java
    CuisineController.java
    EProductController.java
    IndexController.java
    LoginController.java
    LoginServlet.java
    LogoutServlet.java
    PurchaseController.java
  com.freshbox.freshbox.dao
    CuisineDAO.java
    CuisineRepository.java
    ProductDAO.java
    ProductRepository.java
    UserDAO1.java
    UserRepository.java
    UserRepository2.java
  com.freshbox.freshbox.model
    Admin.java
    AuthenticationRequest.java
    AuthenticationResponse.java
    Cuisine.java
    Product.java
    User.java
  com.freshbox.freshbox.util
    MvcConfig.java
    MyUserDetails.java
    UserDetailsServiceImpl.java
    WebSecurityConfig.java
```

Admin Controller : Performs admin login and admin dashboard control

Cuisine Controller : Cuisine display, add, delete

Eproduct Controller : Product display, edit, add, delete

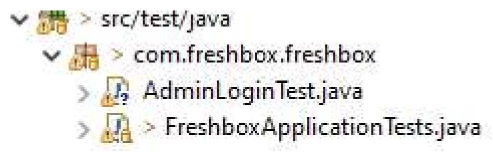
Index Controller : Home page control – disply, sort, filter

Login Controller : User login, registration, signout, profile

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

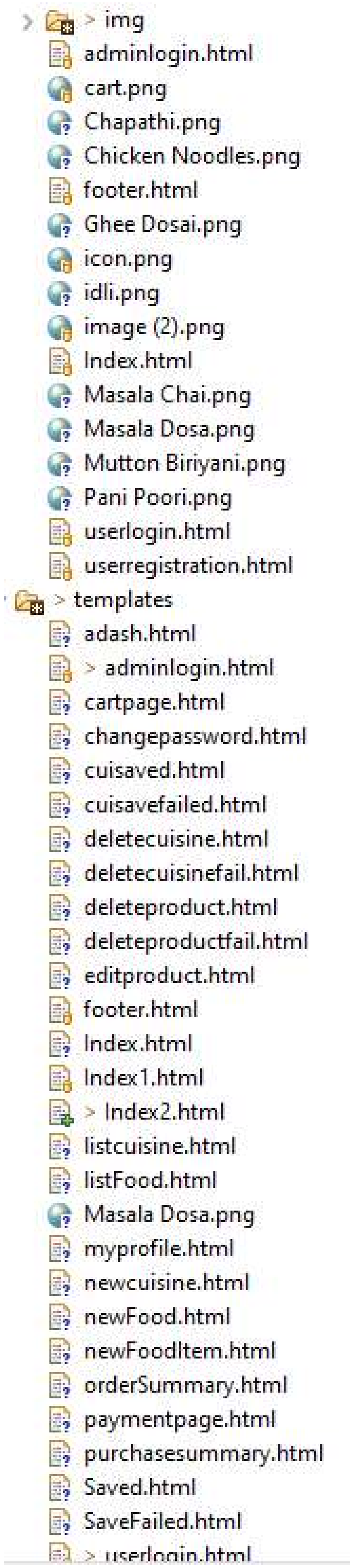
CuisineDAO, ProductDAO, UserDAO1 are DAO for respective DB tables

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



```
src/test/java
  com.freshbox.freshbox
    AdminLoginTest.java
    FreshboxApplicationTests.java
```

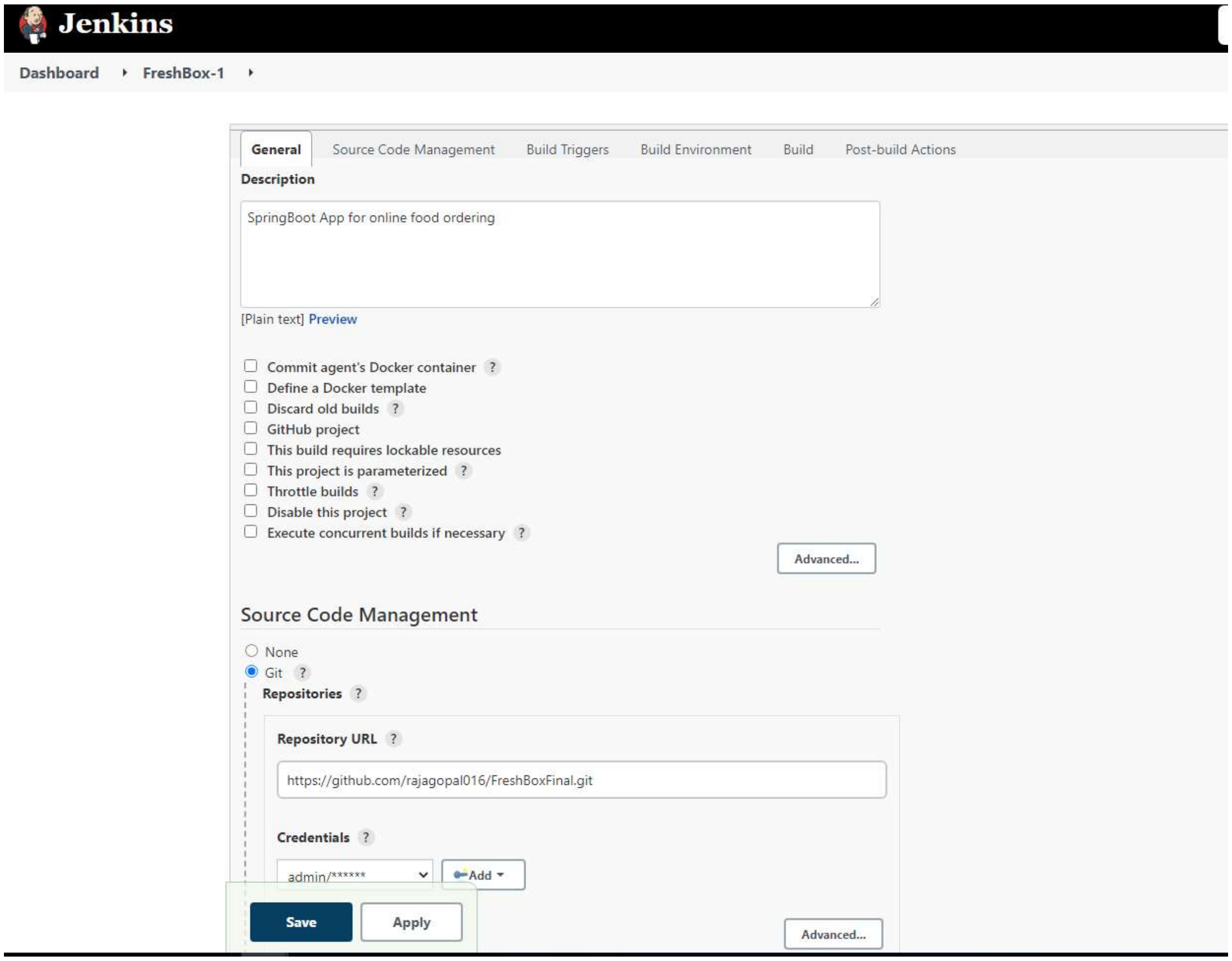
Above classes for junit test cases. Test cases are written for necessary function units



```
> img
  adminlogin.html
  cart.png
  Chapathi.png
  Chicken Noodles.png
  footer.html
  Ghee Dosai.png
  icon.png
  idli.png
  image (2).png
  Index.html
  Masala Chai.png
  Masala Dosa.png
  Mutton Biriyani.png
  Pani Poori.png
  userlogin.html
  userregistration.html
> templates
  adash.html
  > adminlogin.html
  cartpage.html
  changepassword.html
  cuisaved.html
  cuisavefailed.html
  deletecuisine.html
  deletecuisinefail.html
  deleteproduct.html
  deleteproductfail.html
  editproduct.html
  footer.html
  Index.html
  Index1.html
  > Index2.html
  listcuisine.html
  listFood.html
  Masala Dosa.png
  myprofile.html
  newcuisine.html
  newFood.html
  newFoodItem.html
  orderSummary.html
  paymentpage.html
  purchasesummary.html
  Saved.html
  SaveFailed.html
  > userlogin.html
```

Above files are the html and image files for respective functions

Jenkins Pipeline:



Jenkins

Dashboard ▸ FreshBox-1 ▸

Description

SpringBoot App for online food ordering

[Plain text] Preview

- [ ] Commit agent's Docker container
- [ ] Define a Docker template
- [ ] Discard old builds
- [ ] GitHub project
- [ ] This build requires lockable resources
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project
- [ ] Execute concurrent builds if necessary

Source Code Management

None [✓ Git]

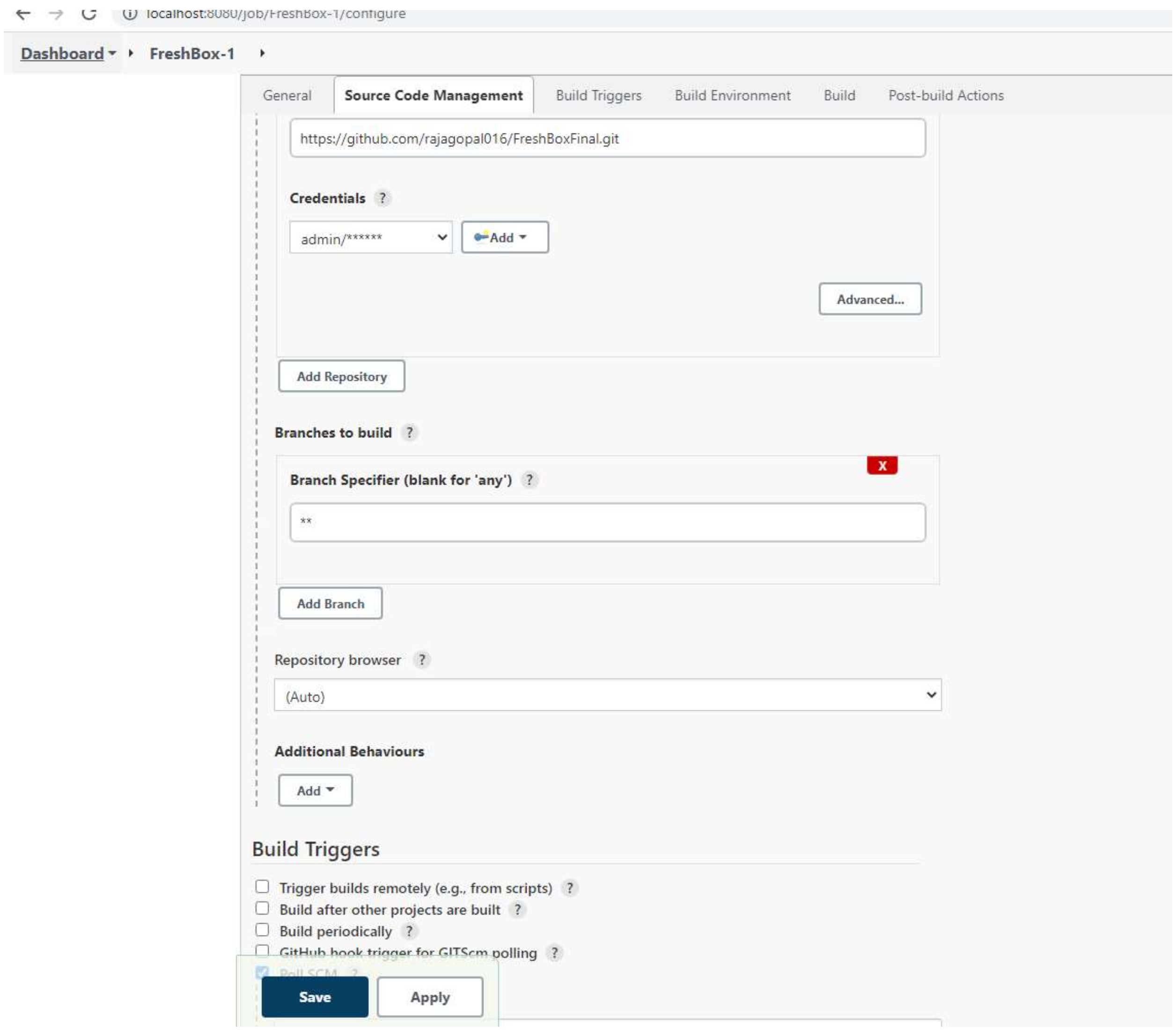
Repositories

Repository URL

https://github.com/rajagopal016/FreshBoxFinal.git

Credentials

admin/******


localhost:8080/job/FreshBox-1/configure

Dashboard ▸ FreshBox-1 ▸

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

https://github.com/rajagopal016/FreshBoxFinal.git

Credentials ?

admin/****** | Add

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

**

Add Branch

Repository browser ?

(Auto)

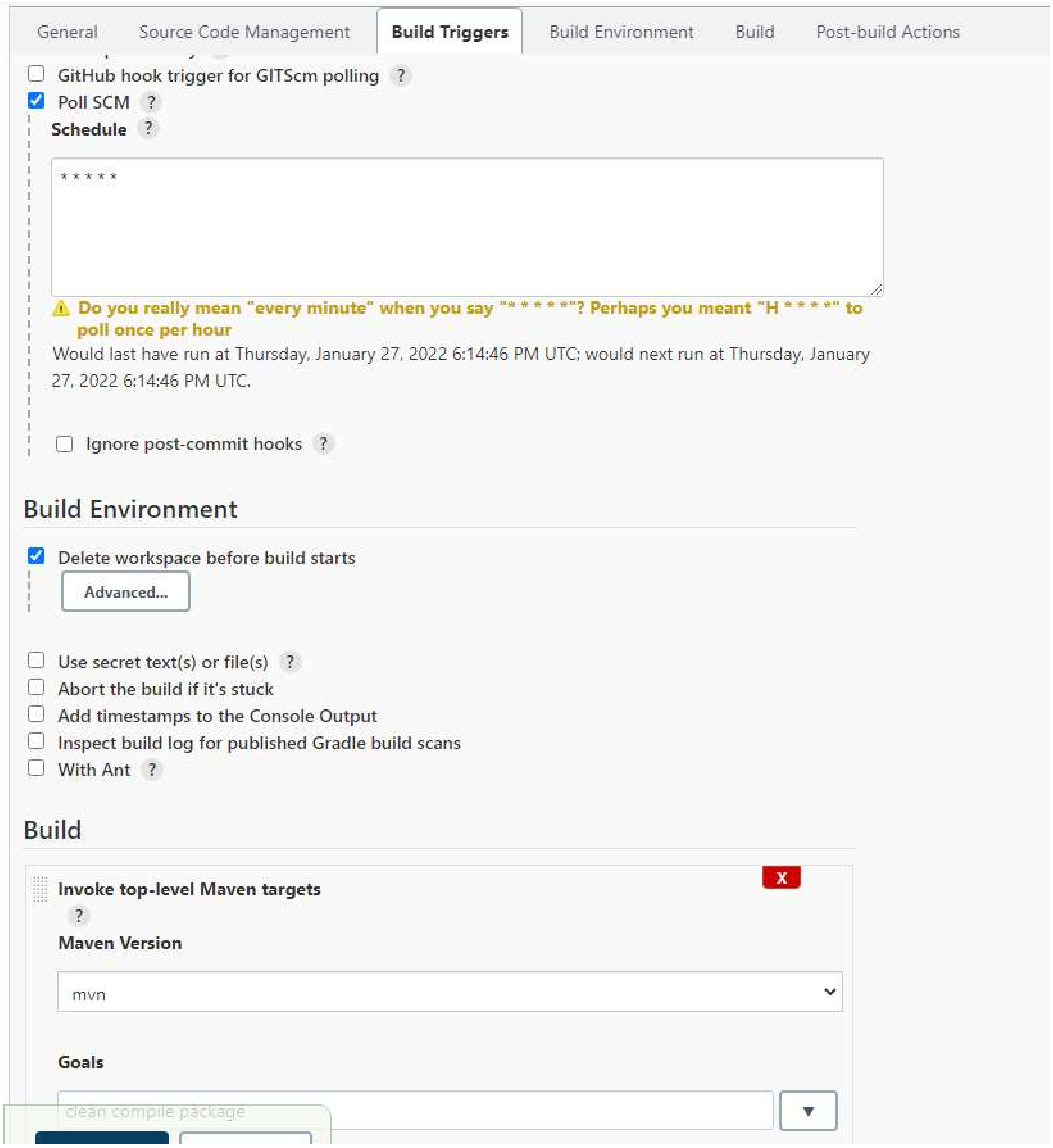
Additional Behaviours

Add

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts) ?
- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [ ] GitHub hook trigger for GITScm polling ?
- [✓] Poll SCM ?

Save | Apply



- [ ] GitHub hook trigger for GITScm polling
- [x] Poll SCM

Schedule

```
* * * * *
```

⚠ **Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour**

Would last have run at Thursday, January 27, 2022 6:14:46 PM UTC; would next run at Thursday, January 27, 2022 6:14:46 PM UTC.

- [ ] Ignore post-commit hooks

Build Environment

- [x] Delete workspace before build starts

- [ ] Use secret text(s) or file(s)
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant

Build

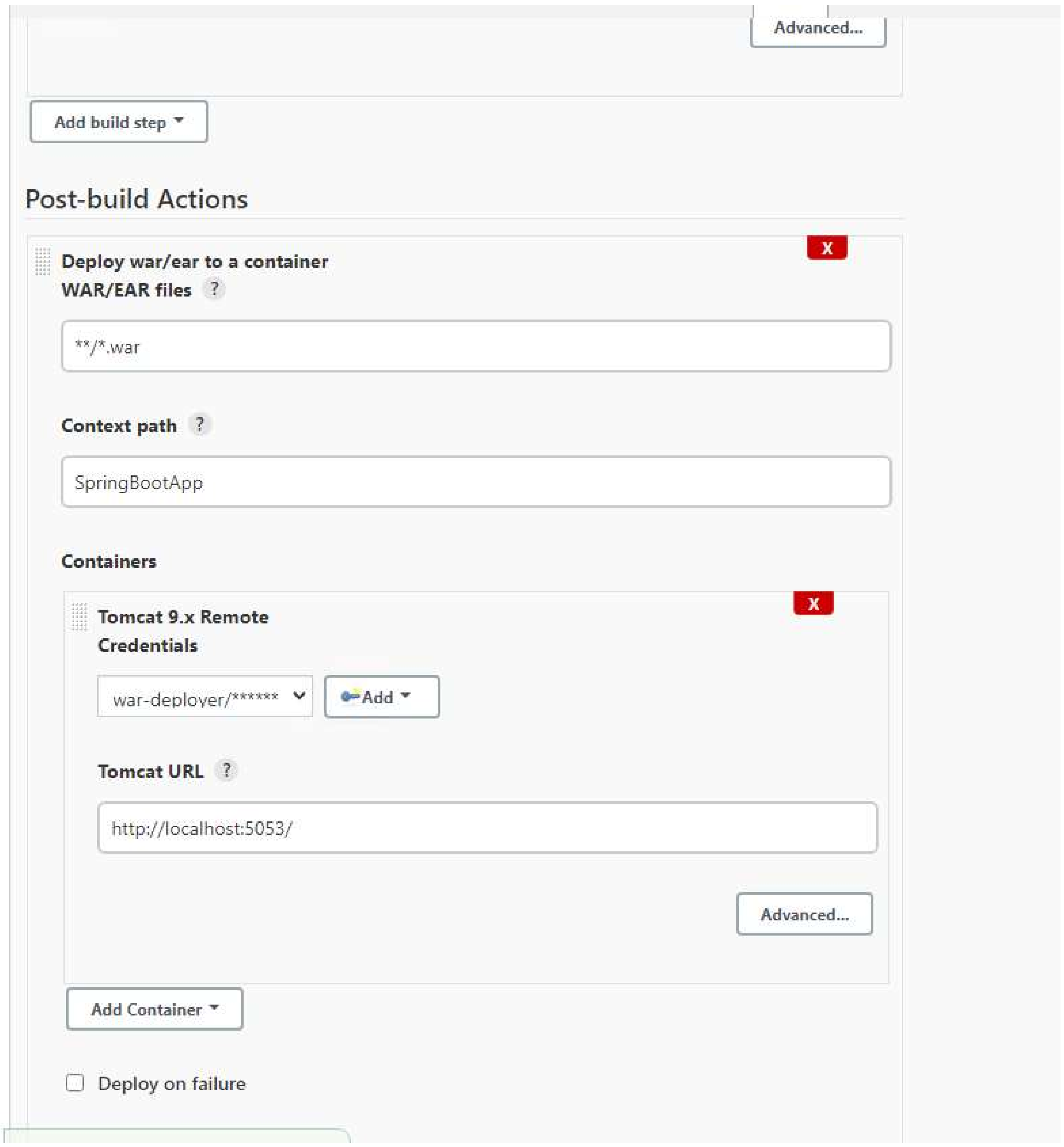
Invoke top-level Maven targets

Maven Version

mvn

Goals

clean compile package


Post-build Actions

Deploy war/ear to a container

WAR/EAR files: **/*.war

Context path: SpringBootApp

Containers

Tomcat 9.x Remote

Credentials: war-deployer/******

Tomcat URL: http://localhost:5053/

Deploy on failure [ ]