

# Foodbox

**Githublink:**

## **DESCRIPTION**

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.

### **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

### **Recommended technologies:**

1. Database management: MySQL and Oracle
2. Backend logic: Java programming, NodeJS
3. Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript
4. Automation and testing technologies: Selenium, Jasmine, and TestNG
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

### **Project development guidelines:**

- The project will be delivered within four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
- The learner can use any technology from the above-mentioned technologies for different layers of the project.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The learner must implement a CI/CD pipeline using Jenkins.

- The learner should also deploy and host the application on an AWS EC2 instance.
- The learner should also implement automation testing before the application enters the CI/CD pipeline.
- The learner should use Git branching to do basic automation testing of the application in it separately.
- The learner should make a rich frontend of the application, which is user-friendly and easy for the user to navigate through the application.
- There will be two portals in the application, namely admin and user portal.

#### **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
- Enable or disable the food items

#### **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