WIPRO NGA Program – C P Java

Capstone Project Presentation – 16th Oct 2024

Project Title Here – Organic Farms

Presented by – Parth Gurval and Vattikuti Sravani

Problem Statement:

Context: You are tasked with developing an online marketplace platform that connects customers with farm partners for agricultural products.

1.Customer: The customer can browse and view products available on the platform. They do not have access to modify or add products, but they can search for and view product details.

2.Farm Partner: The farm partner can add new products to the platform and also view the existing products, including those they have added. Their role is focused on supplying products and maintaining their listings.

3.Admin: The admin has a supervisory role and is responsible for monitoring the overall activity on the website. This includes viewing customer information, farm partner details, and ensuring that the marketplace functions smoothly. The admin does not directly interact with products but oversees both customer and farm partner activities.

Need for the Project / Pain Points:

Problem Statement:

Before this platform, customers had to physically visit farms to buy products, which was time-consuming, limited product options, and lacked transparency on prices and availability. Farm partners struggled to reach a wider audience and efficiently manage their sales. There was also no system for admins to monitor transactions and user activities.

Solution:

This platform allows:
- **Customers:** To browse and buy farm products from home, with full product transparency.
- **Farm partners:** To easily add and manage products online, reaching more customers.
- **Admins:** To monitor customer and farm partner activities, ensuring smooth platform operation.

Solution Design: Organic Farms Platform

Overview:
A web-based platform designed for customers to buy organic farm products online, farm partners to manage their products, and admins to monitor activities on the platform.

Features:
- Seamless integration of multiple farm partners.
- Real-time availability of products based on user preferences.
- User reviews and ratings to ensure product quality.
- Secure online payments with multiple payment options.

Architecture:
Frontend: Built with HTML5, CSS3, TypeScript, and Angular for a dynamic user experience.
Backend: Developed using Core Java, Spring Boot, RESTful APIs, and MySQL for efficient data storage.

Testing:
Unit and integration testing implemented using JUnit for the backend services to ensure robust functionality and error handling.
This architecture ensures a smooth and scalable platform for organic farms.

Microservices Architecture for Organic Farms Platform:

- **Admin Service:**
  Manages the overall platform, including monitoring customers and farm partners.
Cart Service:
  Handles adding, updating, and removing products in the customer's shopping cart.
- **Customer Service:**
   Manages customer registration, login, and profile information.
- **FarmPartner Service:**
  Allows farm partners to add, update, and view their organic products.
- **JWT Service:**
  Provides authentication and authorization using JSON Web Tokens (JWT).
- **OrderAPI Service:**
  Handles the creation and management of customer orders.
- **OrderDetails Service:**
  Manages order details like product information, quantities, and status.
- **Payment Service:**
  Facilitates payment processing and transaction handling.
- **Product Service:**
  Manages product inventory, availability, and CRUD operations for organic products.
- **Rating Service:**
  Collects and manages product reviews and ratings from customers.
- **PayPal Service:**
  Interfaces with PayPal for processing secure online payments.

JWT Authentication

- Manual JWT Implementation:
- JWT tokens are manually generated after successful login.
- Each request includes token validation for user authentication.
- Tokens are verified to grant or restrict access to various resources.

Dockerization

- Containerization Strategy:
- Each microservice is containerized using Docker for consistency across development, testing, and production environments.
- Docker Compose:
- Used for managing multi-container Docker applications and orchestrating services.
- Benefits:
- Simplified deployment.
- Enhanced scalability.
- Environment consistency.

Tech Stack for Organic Farms:

Frontend: HTML5, CSS3, TypeScript, Angular

Backend: Java Core, Spring Boot

Database: MySQL

APIs: RESTful API architecture

Microservices: Spring Boot-based microservices

Authentication: JWT-based authentication

Testing: JUnit for backend unit testing

Deployment & Orchestration: Docker, Kubernetes (optional)

Backlog
Sprint 1: Set up backend with Spring Boot, JPA, JWT, and payment gateway integration
Sprint 2: Develop Angular frontend for managing products and user interactions.
Sprint 3: Implement advanced features like real-time availability, reviews, and user
Sprint 4: Design microservices architecture with Docker, inter-service communication,
Sprint 5: Conduct testing, deploy on Kubernetes, and complete documentation.

Conclusion:

The Organic Farms project provides a seamless platform for customers to buy organic products online, offering real-time availability, secure payments, and user reviews. Using microservices with Spring Boot and Angular ensures scalability and reliability, while JWT-based authentication enhances security. This project promotes easy access to fresh produce, benefiting both customers and farm partners.