



YumStop – Online Food Delivery System

YumStop is a robust, full-stack web application inspired by industry leaders like **Swiggy** and **Zomato**. Built with a focus on scalability and clean architecture, it bridges the gap between hungry users and their favorite meals through a seamless digital experience.



Project Overview

This project serves as a comprehensive demonstration of the **MVC (Model-View-Controller) design pattern**, featuring a dual-interface system for both customers and administrators. It handles everything from secure authentication to complex order management and database persistence.



Key Highlights

- 🔥 **Modern UI:** Clean, intuitive interface for an enhanced user experience.
- 🧠 **Solid Architecture:** Decoupled Service and Repository layers for maintainability.
- 👑 **Role-Based Control:** Dedicated dashboards for Users and Administrators.
- 💻 **End-to-End Workflow:** From browsing menus to simulated checkout and order tracking.



Feature Set



Customer Experience

- Authentication:** Secure registration, login, and "Forgot Password" functionality.
- Menu Discovery:** Browse a rich catalog of food items with real-time price and image display.
- Cart Management:** Add, remove, and review items before ordering.
- Order Tracking:** View personal order history and current status.
- Support:** Integrated feedback system and contact support module.



Admin Management

- Analytics Dashboard:** Quick overview of system performance.
- User Management:** Monitor registered customers and their activity.
- Inventory Control:** Full CRUD operations (Create, Read, Update, Delete) for the food menu.
- Order Oversight:** Manage and track all incoming orders from across the platform.



Tech Stack

| Layer | Technology |
|------------|--|
| Backend | Java, Spring Boot |
| Frontend | JSP (JavaServer Pages), HTML5, CSS3, Bootstrap |
| Database | MySQL |
| ORM | Spring Data JPA / Hibernate |
| Security | Session-based Authentication |
| Build Tool | Maven |

System Architecture

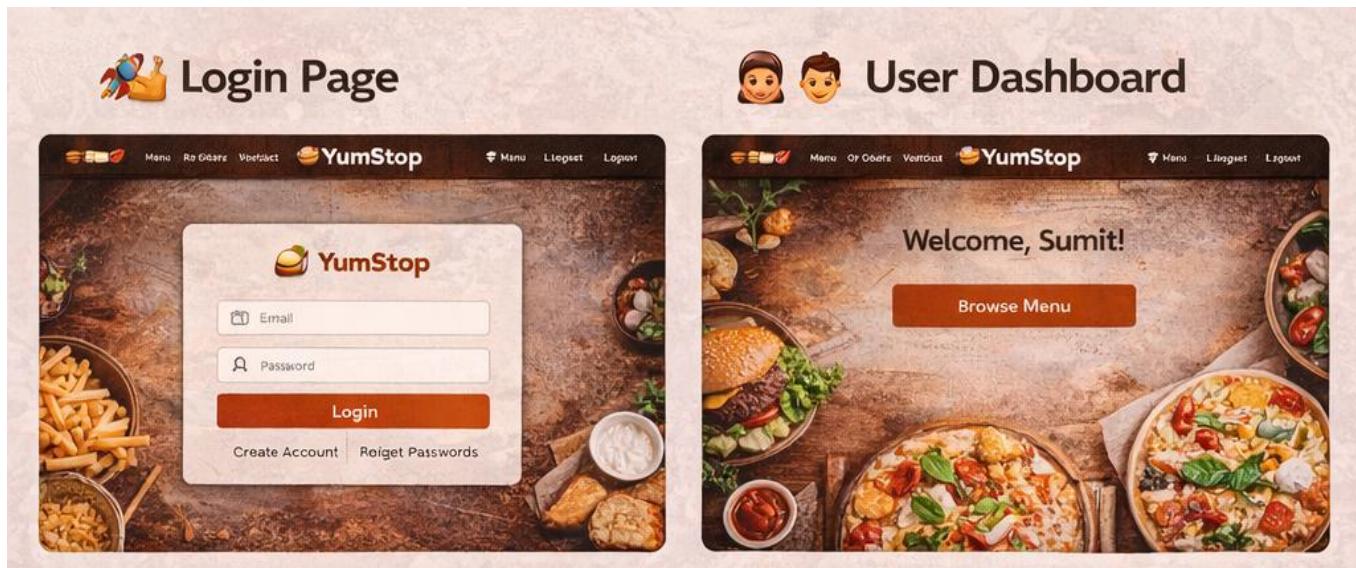
The application follows the **Industry Standard MVC Pattern**, ensuring a clear separation of concerns:

Plaintext

Client (JSP) <---> Controller <---> Service Layer <---> Repository <---> MySQL Database

- **Controller:** Handles incoming HTTP requests and routes them to the correct view.
- **Service:** Contains the business logic and data processing.
- **Repository:** Abstracts the database communication using JPA.

Visual Tour



Food Menu

Food Ordering

The Food Menu screen shows a grid of three items: Pizza, Burger, and Pasta, each with a price of ₹ 199 and an 'Add to Cart' button.

The Food Ordering screen shows a large image of a pizza on a wooden board. The cart summary indicates 1 item (Pizza) at ₹ 199, with a total of ₹ 199. A 'Proceed to Payment' button is visible.

Payment Page

Feedback Page

The Payment Page shows a Razorpay payment interface for ₹ 199 via Cards or UPI.

The Feedback Page displays the Admin Dashboard with order history and a list of users (Samit, Rahul, Aman) with their details and order history.

Admin Dashboard

Admin Dashboard

The Admin Dashboard shows a user list and a detailed view of a user's orders.

The Admin Dashboard also includes a 'Menu Management' section for adding new food items, with fields for Item Name, Price, and Image URL.

⚙️ Installation & Setup

1. Prerequisites

- JDK 17 or higher
- Maven 3.6+
- MySQL Server

2. Clone and Configure

Bash

```
git clone https://github.com/YOUR_USERNAME/YumStop-Food-Delivery-System.git
```

```
cd YumStop-Food-Delivery-System
```

3. Database Setup

Create the database in your MySQL instance:

SQL

```
CREATE DATABASE yumstop;
```

Update `src/main/resources/application.properties`:

Properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/yumstop
```

```
spring.datasource.username=YOUR_USERNAME
```

```
spring.datasource.password=YOUR_PASSWORD
```

```
spring.jpa.hibernate.ddl-auto=update
```

4. Run the App

Bash

```
mvn spring-boot:run
```

Visit <http://localhost:8080> in your browser.

Learning Outcomes

- Implementing **Session Management** for secure user state.
- Mapping complex **JPA Entity Relationships** (One-to-Many, Many-to-One).
- Building dynamic web pages using **JSP and Taglibs**.
- Designing an administrative backend for real-world data management.

Roadmap & Future Enhancements

- [] **Spring Security:** Implement JWT and OAuth2.
 - [] **Payment Integration:** Live Razorpay/Stripe gateway.
 - [] **Cloud Deployment:** Host on AWS or Render.
 - [] **Containerization:** Add Docker support for easy deployment.
-

Author

Mohit Kumar Gupta

 *B.Tech in Computer Science & Engineering*  Java | Spring Boot | MySQL | Full-Stack Developer

 If you find this project helpful, please give it a star on GitHub!