



*Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)*

---

## **FOOD DELIVERY APPLICATION**

---

### **The domain of the Project:**

Backend Application Development

### **COURSE NAME**

**Core Java**

### **Team Mentors (and their designation):**

**Yaseer Alur**

**Junior Software Engineer**

### **Team Members:**

**Ms. Siya Kumari**

### **Period of the project**

**May 2025 to November 2025**



*Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)*  
**Declaration**

The project titled "**Food Delivery Application**" has been mentored by **Mr. Yaseer Alur**, organised by SURE Trust, from **19th May 2025 to 19th November 2025**, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

**Team Members:**

Ms. Siya Kumari

**Mentor's Name:** Mr. Yaseer Alur

**Junior software developer**

**Designation — SURE Trust**

Prof. Radhakumari

Executive Director & Founder

SURE Trust



*Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)*

## Table of contents

1. Executive summary
2. Introduction
3. Project Objectives
4. Methodology & Results
5. Social / Industry relevance of the project
6. Learning & Reflection
7. Future Scope & Conclusion



## *Executive Summary*

---

- **Objectives:**

To develop a secure and scalable **backend-based Food Delivery Application** that manages users, restaurants, orders, payments, delivery tracking, and reviews using Java and Spring Boot.

- **Methods:**

The project is implemented as a **RESTful backend system** following a layered architecture. **Spring Security with JWT** is used for authentication and authorization, and **MySQL with Spring Data JPA** is used for data persistence. APIs are tested using Postman.

- **Key Findings:**

The system successfully supports secure user authentication, role-based access control, order lifecycle management, payment simulation, delivery partner assignment, notifications, and review handling through well-defined APIs.

- **Recommendations:**

The application can be further enhanced by integrating a frontend interface, real payment gateways, live delivery tracking, and performance optimizations for production use.

- **Executive Highlights:**

This project demonstrates practical backend development skills, secure API design, and real-world application architecture suitable for industry-level systems.



## *Introduction*

---

- **Background and Context of the Project:**

With the rapid growth of digital services, online food delivery platforms have become an essential part of modern lifestyles. Such systems require a reliable backend to manage users, restaurants, orders, payments, and delivery operations securely and efficiently. This project focuses on developing the backend of a food delivery system that supports these core functionalities using Core Java and Spring Boot.

- **Problem Statement / Goals of the Project:**

Traditional food ordering processes are manual, time-consuming, and prone to errors. The goal of this project is to design and implement a backend-based Food Delivery Application that automates order management, ensures secure user authentication, and provides a structured system for restaurants, customers, and delivery partners through RESTful APIs.

- **Scope and Limitations of the Project:**

The scope of this project is limited to backend development, including API design, authentication, database management, and business logic implementation. Frontend technologies such as HTML, CSS, or JavaScript are not included. The application uses mock payment processing instead of real payment gateway integration, and delivery tracking is simulated at the backend level.

- **Innovation Component in the Project:**

The project incorporates JWT-based authentication and role-based authorization to ensure secure and stateless communication. A layered architecture is used to improve scalability and maintainability. The design allows easy future integration with frontend applications, real payment gateways, and advanced delivery tracking features, making it adaptable for real-world use.



### **Project Objectives**

---

- **Objectives and Goals of the Project:**

- To design and develop a backend-based Food Delivery Application using Core Java and Spring Boot.
- To implement secure user authentication and authorization using Spring Security and JWT.
- To create **RESTful APIs** for managing users, restaurants, menu items, orders, payments, delivery tracking, notifications, and reviews.
- To apply a **layered architecture** (Controller, Service, Repository, Entity, DTO, Security) for better code organization, scalability, and maintainability.
- To use **MySQL with Spring Data JPA** for efficient database management and persistence.
- To ensure proper validation, error handling, and role-based access control across the application.

- **Expected outcomes and deliverables:**

- A fully functional **backend system** for a food delivery application with secure API endpoints.
- JWT-protected APIs supporting role-based access for customers, restaurant owners, delivery partners, and administrators.
- A structured and maintainable codebase following industry-standard backend architecture.
- Project documentation explaining system design, implementation details, and future enhancement possibilities.



### **Methodology and Results**

---

- **Methods/Technology used:**

The project is developed as a **backend-based RESTful application** using **Core Java and Spring Boot**. REST APIs are designed to handle user authentication, restaurant and menu management, order processing, payment handling, delivery tracking, notifications, and reviews.

**Spring Security with JWT (JSON Web Token)** is used to implement secure, stateless authentication and role-based authorization.

**Spring Data JPA** is used for database interaction, enabling efficient CRUD operations and relationship management between entities.

- **Tools / Software Used:**

**Java (JDK 21)** – Core programming language

**Spring Boot** – Backend framework

**Spring Security** – Authentication and authorization

**JWT** – Token-based security

**Spring Data JPA** – ORM and database access

**MySQL** – Relational database

**Postman** – API testing

**IntelliJ IDEA** – Development IDE

**GitHub** – Version control and source code management

- **Project Architecture:**

The application follows a layered architecture, which ensures separation of concerns and maintainability:

- **Controller Layer** – Handles HTTP requests and API endpoints
- **Service Layer** – Contains business logic and validations
- **Repository Layer** – Manages database operations using JPA
- **Entity Layer** – Represents database tables
- **DTO Layer** – Handles request and response data transfer
- **Security Layer** – Manages JWT authentication and authorization

This architecture improves scalability, readability, and ease of future enhancements.



*Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)*  
**Final project working screenshots along with supporting explanation:**

The screenshot shows the Postman interface with a successful GET request to `http://localhost:8080/api/auth/test`. The response status is `200 OK` with a duration of `1.63 s` and a size of `358 B`. The response body contains the message `JWT Security is working!`.

**Figure 1: JWT Security is working**

The screenshot shows the Postman interface with a successful POST request to `http://localhost:8080/api/auth/register`. The request body is a JSON object with fields `name`, `age`, and `address`. The response status is `201 Created` with a duration of `3.51 s` and a size of `552 B`. The response body contains the message `User registered successfully`.

**Figure 2: User registration API request and response**



Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for 'My Workspace' (Collections, Environments, History, Flows), 'API Network' (Cloud View, Find and replace, Console, Terminal), and a 'BETA' section for 'Files'. The main workspace is titled 'Overview' and shows a POST request to 'http://localhost:80'. The 'Body' tab is selected, containing raw JSON input:

```
1 {  
2   "emailId": "aman.user@gmail.com",  
3   "password": "Aman@123"  
4 }
```

The response status is '200 OK' with a duration of '1.33 s' and a size of '663 B'. The response body is displayed as JSON:

```
1 {  
2   ... "token": "eyJhbGciOiJIUzI1NiJ9.  
3   eyJzdWIiOiJhbWFlUlNvZzXJA21haWwUZ29tIiwiZWQiJoxNzY2MjIyMDcwLC1leHaiOjE3NjYzMdg0NzB9.  
4   JMKuoq7lvAPxUlZ7UHZ3Hgcl1T39Yqpro9HHVuaoHW7pmuc1ffjuYyfkxZ9ITAMXG4PGRY7nsXXPusel04A",  
5   ... "emailId": "aman.user@gmail.com",  
6   ... "name": "Aman Kumar",  
7   ... "role": "CUSTOMER",  
8   ... "userId": 9,  
9   ... "message": "Login successful",  
10  ... "success": true  
11 }
```

**Figure 3: User login API request and response**

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace is titled "My Workspace" and contains a "Collections" section with items like "Search collections", "API Documentation #reference", "REST API basics: CRUD, test & variable", and "RESTful API Basics #blueprint". The central area shows an "Overview" tab and a "POST http://localhost:80" request. The "Body" tab is selected, showing a raw JSON payload:

```
2   "restaurantId": 1,
3   "deliveryAddress": "Jhumri Telaiya, Jharkhand",
4   "customerNotes": "Please deliver fast",
5   "items": [
6     {

```

The "Test Results" tab shows a successful response: "201 Created" with a response time of "290 ms" and a size of "1022 B". The response body is displayed as:

```
1   {
2     "id": 2,
3     "customerId": 9,
4     "customerName": "Aman Kumar",
5     "customerPhone": "9123456788",
6     "restaurantId": 1,
7     "restaurantName": "Mamma Mia Ultimate Trattoria",
8     "items": [
9       {
10         "id": 2,
11         "menuItemID": 1,
```

**Figure 4: Order created API request and response**

Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

My Workspace

Overview

GET http://localhost:8080/api/restaurants

Send

Body (14) Headers (10) Cookies Scripts Settings

200 OK 1.43 s 1.38 KB

```
{
  "id": 1,
  "name": "Mamma Mia Ultimate Trattoria",
  "description": "Ultimate Italian fine dining experience",
  "address": "123 Little Italy, Ultimate District",
  "cuisineType": "Italian",
  "phone": "9876543211",
  "email": "ultimate@mammamia.com",
  "status": "ACTIVE",
  "rating": 0.0,
  "imageUrl": "https://example.com/ultimate.jpg",
  "openingHours": "12:00 PM - 12:00 AM",
  "deliveryFee": 5.0,
  "estimatedDeliveryTime": 20,
  "ownerId": 7
}
```

**Figure 5: Get all restaurants**

My Workspace

Overview

POST http://localhost:8080/api/menu-items/createMenuItem

Send

Body (14) Headers (10) Cookies Scripts Settings

201 Created 2.15 s 821 B

```
{
  "id": 10,
  "name": "Butter Chicken",
  "description": "Tender chicken in rich tomato gravy",
  "category": "MAIN_COURSE",
  "price": 350.0,
  "status": "AVAILABLE",
  "imageUrl": "https://example.com/butter-chicken.jpg",
  "restaurantId": 1,
  "restaurantName": "Mamma Mia Ultimate Trattoria",
  "createdAt": "2025-12-20T20:54:39.7105330",
  "message": "Menu item created successfully",
  "success": true,
  "menuItems": [],
  "categories": []
}
```

**Figure 6: Created menu-items API request and response**



*Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)*

**Project GitHub Link :-** <https://github.com/sure-trust/SIYA-KUMARI-g19-java/tree/main/Final%20capstone%20project>



***Learning and Reflection***

---

**New Learnings (Technology, Management, etc.):**

During the development of this Food Delivery Application, I gained strong practical exposure to **backend development using Java and Spring Boot**. I learned how to design and implement **RESTful APIs** following industry-standard practices and how to organize code using a **layered architecture** (Controller, Service, Repository, DTO, and Security layers).

One of the key technical learnings was implementing **Spring Security with JWT-based authentication and authorization**. This helped me understand stateless security, token generation, validation, and role-based access control for different users such as customers, restaurant owners, delivery partners, and administrators.

I also learned effective **database design and management using MySQL and Spring Data JPA**, including entity relationships, CRUD operations, and data validation. Testing APIs using **Postman** improved my understanding of request-response flow, error handling, and debugging backend issues.

From a management perspective, I learned how to **plan features, break down tasks, manage time**, and iteratively improve the application. Using **GitHub for version control** helped me understand code management and project organization.

**Overall Experience :**

The overall experience of working on this project was highly valuable and challenging. Developing a real-world backend application helped me convert theoretical knowledge into practical implementation. I faced challenges while implementing security, handling validations, and managing different modules, which improved my problem-solving and debugging skills.

This project increased my confidence in backend development and gave me a clear understanding of how industry-level applications are structured and maintained. It also improved my ability to work independently, take ownership of tasks, and build a complete backend system from scratch.

Overall, this project served as a strong learning foundation for real-world backend development and prepared me for future projects and professional roles in Java backend development.



### **Conclusion and Future Scope**

---

- **Conclusion:**

The objective of this project was to design and develop a **secure, scalable backend-based Food Delivery Application** using Java and Spring Boot. This objective was successfully achieved by implementing a structured and modular backend system that manages users, restaurants, menu items, orders, payments, delivery tracking, notifications, and reviews through RESTful APIs.

The project demonstrates effective use of **layered architecture, Spring Security with JWT-based authentication, and MySQL with Spring Data JPA** for data persistence. Secure role-based access control, proper validation, and error handling were implemented to ensure reliability and maintainability of the system.

Overall, the project meets its intended goals and provides a strong foundation for a real-world food delivery application backend, reflecting practical backend development and secure API design skills.

- **Future scope :**

The current implementation focuses on backend functionality; however, the project can be further enhanced in several ways:

- Integration of a frontend application using technologies such as React or Angular for user interaction.
- Implementation of real payment gateway integration (e.g., UPI, Razorpay, Stripe) instead of simulated payments.
- Addition of real-time delivery tracking using GPS and WebSocket-based communication.
- Performance optimization through caching mechanisms and database query optimization.
- Deployment of the application on cloud platforms such as AWS or Azure with CI/CD pipelines.
- Enhancement of notification services using email and SMS gateways.

These enhancements would make the application production-ready and suitable for large-scale commercial use.