

# FULL STACK DEVELOPER TEST – CYMELLE TECHNOLOGIES

## The Task

You are required to develop a Spring Boot backend service for Cymelle Technologies platform. The service should manage products, users, and rides/orders. This project will test your ability to design a service layer, create RESTful APIs, handle data relationships, implement authentication, and write integration tests.

## Requirements

### 1. Product Management (CRUD Operations)

- Implement functionality to **add, update, delete, and view products**.
- Each product should have the following attributes: id, name, description, price, stockQuantity, category.
- Ensure **data validation** for product input.

### 2. User Registration and Authentication

- Implement **user registration and authentication** using Spring Security.
- Include **roles**: ADMIN and CUSTOMER.
- Ensure that **only authenticated users** can place orders/rides and **only admins** can manage products.
- Use **JWT (JSON Web Token)** for securing API endpoints.

### 3. Order Management (Ecommerce) / Ride Management (Hailing)

#### Ecommerce Orders:

- Place an order, view order details, and update order status (PENDING, SHIPPED, DELIVERED).
- An order should include: list of products, quantities, total cost, and reference to the customer who placed it.

#### Hailing App Rides:

- Request a ride, view ride details, and update ride status (REQUESTED, ACCEPTED, COMPLETED).
- A ride should include: userId, driverId, pickupLocation, dropoffLocation, fare, and status.

## 4. Search Functionality

- Implement **search** for products (by name or category).
- Implement **search** for rides/orders (by user or status).
- The search should return **paginated results**.

## 5. Bonus Features (Optional)

- **Swagger Documentation:** Integrate Swagger UI for API documentation.
- **Payment Gateway Simulation:** Add a dummy payment status to ecommerce orders.
- **Docker Support:** Package the application in a Docker container.
- **Database:** Use **PostgreSQL** or **MySQL**.
- **CI/CD Pipeline:** Configure automated tests on commits and optional deployment.

## How to Submit

- Push your code to a **GitHub repository**.
- Provide clear **API documentation** (Postman collection or Swagger).
- Include a **README file** explaining the project structure, setup instructions, and any relevant notes.
- Ensure instructions for running the project locally (including database setup) are included.

## Deliverables

- **GitHub repository link**
- **Postman collection or Swagger API documentation**
- **README file**