Sales Management System - Project Documentation

# 📌 Project Overview

Sales Management System is a full-stack web application developed using Spring Boot (Java 19, Spring Boot 3.4.5) for backend microservices and Angular 19 for the frontend. It provides user authentication, order management, and user management functionalities.

# 📁 Repository Structure

sales-management-system/  
│  
├── auth-service/ # Microservice for authentication and JWT token generation  
├── user-service/ # Microservice for user CRUD operations  
├── order-service/ # Microservice for order CRUD operations  
├── Sales-Application-Frontend/ # Angular 19 frontend  
├── docker-compose.yml # Docker Compose configuration for local deployment  
└── swarm/ # Docker Swarm configuration files

# 🛠 Backend Technology Stack

- Java 19  
- Spring Boot 3.4.5  
- Spring Security + JWT  
- Spring Data JPA  
- Redis for session/token cache  
- OpenFeign for inter-service communication  
- Swagger/OpenAPI for API documentation  
- PostgreSQL (or MySQL, depending on setup)

# 🧑‍💻 Frontend Technology Stack

- Angular 19 with standalone components  
- Reactive Forms & Form Validations  
- JWT-based AuthGuard and Token Interceptor  
- Styled components using SCSS

# 📘 API Documentation (Swagger)

Each microservice exposes Swagger UI at:  
- Auth Service: http://localhost:8081//swagger-ui/swagger-ui/index.html  
- User Service: http://localhost:8083//swagger-ui/swagger-ui/index.html  
- Order Service: http://localhost:8082//swagger-ui/swagger-ui/index.html

# 🚀 Deployment Details

Local Deployment via Docker Compose:

docker-compose up --build

Docker Swarm Configuration:

# Initialize swarm  
docker swarm init  
  
# Deploy stack  
docker stack deploy -c docker-compose.yml sales-app

# 📦 Deliverables

- ✅ Source Code Repository (GitHub/GitLab)  
- ✅ Documentation (README.md + Word document)  
- ✅ Deployment Configuration (Docker Compose + Swarm)  
- ✅ API Documentation (Swagger UI per microservice)

Please find some sample screenshots from the system.





