# CAPSTONE PROJECT E-COMMERCE APPLICATION BATCH-8-JAVA J2EE

**Name:- Rakesh Kumar Dey**

**Batch:- 8 (JAVA J2EE)**

Group-4(ecommerce-application):

|  |  |  |
| --- | --- | --- |
| SL. | NAME | E-MAIL |
| 1 | Hussain Peera Pinjari | [h16656970@gmail.com](mailto:h16656970@gmail.com) |
| 2 | Chintalapudi Dhanusha | [dhanushachintalapudi13@gmail.com](mailto:dhanushachintalapudi13@gmail.com) |
| 3 | Abhishek Narayan | [abhisheknarayan9118@gmail.com](mailto:abhisheknarayan9118@gmail.com) |
| 4 | Gudisi Eekshitha | [gudisieekshitha25@gmail.com](mailto:gudisieekshitha25@gmail.com) |
| 5 | Payani Dhanasekhar | [sekhard433@gmail.com](mailto:sekhard433@gmail.com) |
| 6 | Bikkireddy Teja | [bikkireddyteja999@gmail.com](mailto:bikkireddyteja999@gmail.com) |
| 7 | Rizwana Tadipatri | [rizwanatadipatri@gmail.com](mailto:rizwanatadipatri@gmail.com) |
| 8 | Zeba Qumar | [zebaquamar19@gmail.com](mailto:zebaquamar19@gmail.com) |
| 9 | Ankit Chauhan | [ankit.chauhan0381@gmail.com](mailto:ankit.chauhan0381@gmail.com) |
| 10 | Rakesh Kumar Dey | [rakeshkumardey2@gmail.com](mailto:rakeshkumardey2@gmail.com) |
|  |  |  |

TEAM LEADER :- Ankit Chauhan

**Objective:**

Develop a comprehensive eCommerce application using Spring Microservices with features for authentication, product management, cart management (including item entry), order processing, and customer management.

**Phases & Tasks:**

### Planning Phase

**Requirements Gathering (1 day)**

* **Microservices Identification:** Define the core microservices: Auth, Customer, Product, Cart, Order.
* **API Gateway Interaction:** Outline the API Gateway's role in routing requests between microservices.

**Architecture Design (1 day)**

* **Microservices Architecture:** Design the overall architecture, emphasizing loose coupling and independent deployment.
* **Service Discovery:** Implement Eureka Server for service registration and discovery.
* **API Gateway Setup:** Configure the API Gateway (e.g., Zuul, Spring Cloud Gateway) for routing requests.

### Development Phase

**Day 1: Infrastructure Setup**

* **Eureka Server:** Set up the Eureka Server for service registration and discovery.
* **API Gateway:** Implement the API Gateway to route requests to the appropriate microservices.

**Day 2: Customer Service**

* **CRUD Operations:** Develop CRUD operations for customer profiles (create, read, update, delete).
* **Auth Integration:** Integrate with the Auth Service to handle customer data upon registration.

**Day 3: Product Service**

* **Product Management:** Build functionalities for managing products, including categories and search.
* **APIs:** Implement APIs for product CRUD operations.

**Day 4-5: Cart Service**

* **Cart Management:** Develop features for managing the shopping cart (add, update, delete items).
* **Product Integration:** Integrate with the Product Service to fetch product details.
* **Item Entry:** Implement the item\_entry table to store individual cart items.

### Day 6: Order Service

* **Order Processing:** Implement functionalities for placing orders, tracking order status, and managing order history.
* **Cart Integration:** Integrate with the Cart Service to process cart items during order placement.

### Day 7-8: Auth Service

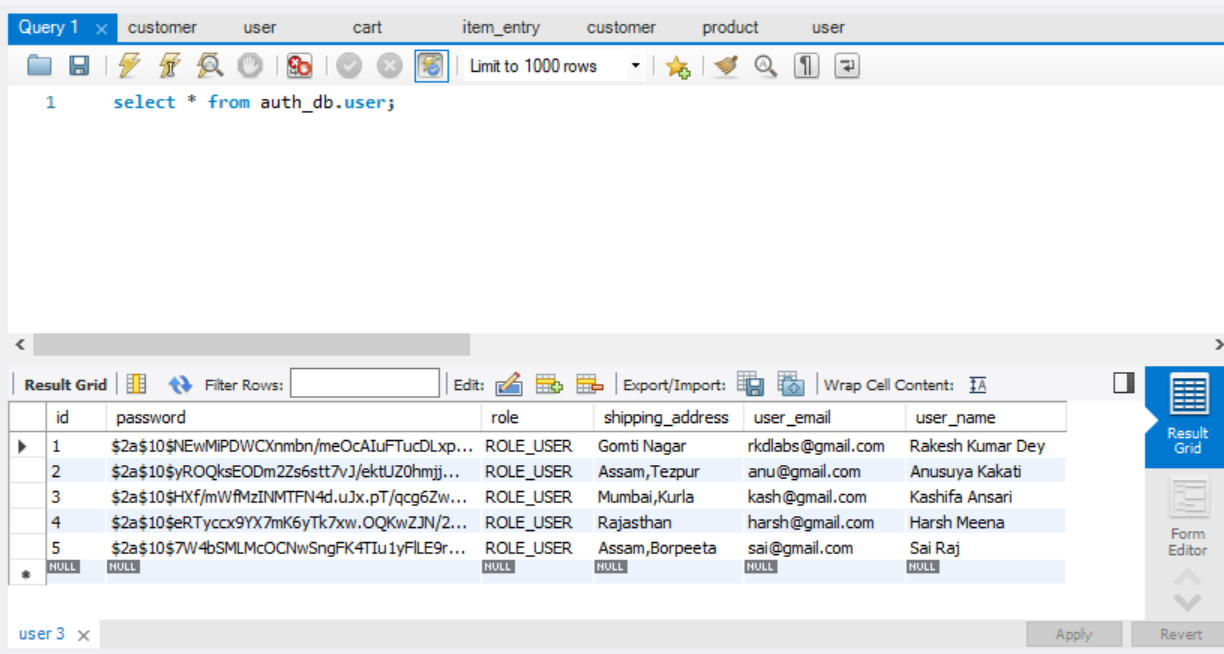
* **Authentication:** Create registration and login functionalities for user accounts.
* **JWT:** Implement JSON Web Token (JWT) generation and validation for secure access control.
* **Password Hashing:** Use strong password hashing algorithms (e.g., bcrypt) to protect user credentials.
* **Authorization:** Implement mechanisms to control access to different resources based on user roles and permissions.

# Microservices

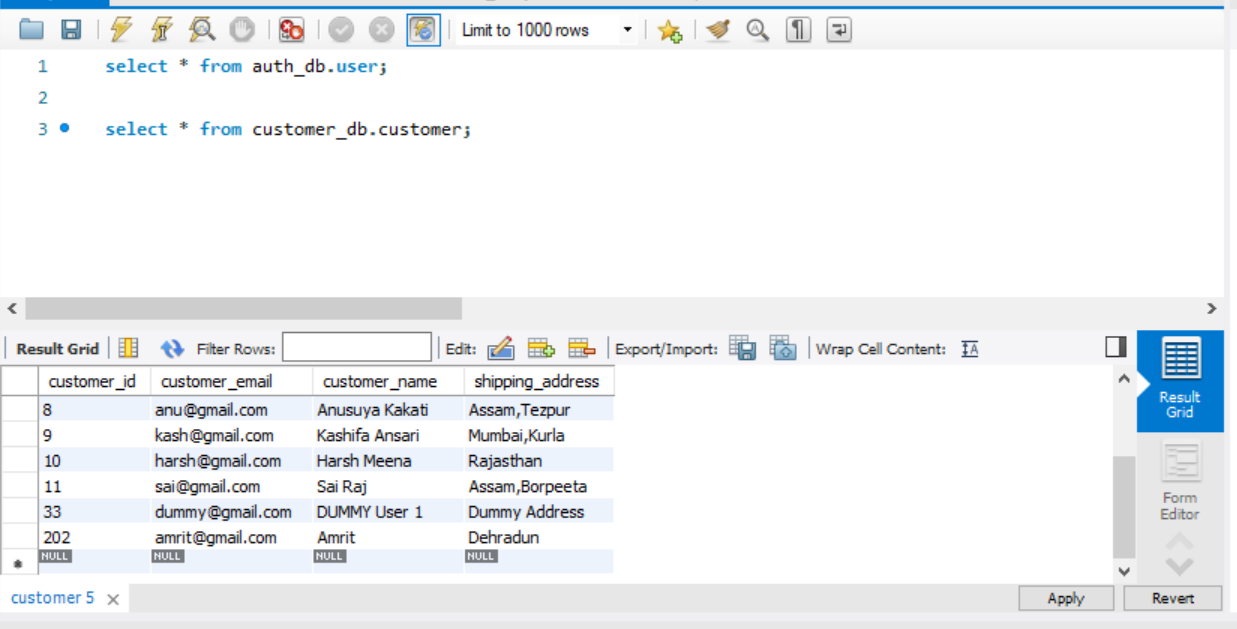
1. **E-commerce Service Registry (Eureka Server)**
   * Service Discovery: Enables dynamic discovery and registration of microservices.
   * Load Balancing: Distributes requests across available service instances.
   * Monitoring: Provides a UI for tracking service health and status.
2. **E-commerce API Gateway**
   * Centralized Routing: Directs client requests to the appropriate microservices.
   * Security: Implements authentication and authorization at the gateway level.
   * Traffic Management: Handles load balancing, rate limiting, and API monitoring.
3. **E-commerce Config Server**
   * Centralized Configuration: Manages and serves externalized configurations for microservices.
   * Environment Support: Provides environment-specific configurations (e.g., dev, prod).
   * Dynamic Updates: Allows microservices to refresh configurations without downtime.
4. **Customer Service**
   * Profile Management: Manages customer profiles, including personal details and addresses.
   * Order History: Provides access to customer order history and status updates.
   * Authentication Integration: Secures customer data through integration with the Authentication Service.
5. **Authentication Service**
   * User Management: Handles user registration, login, and role-based access.
   * JWT Security: Secures API access with JWT-based authentication.
   * Password and Session Management: Manages user sessions and password resets.
6. **Product Service**
   * Product Catalogue: Manages product listings, categories, and details.
   * Search & Filter: Provides search and filtering options for products.
   * Inventory Check: Ensures product availability before purchase.
7. **Cart Service**
   * Cart Management: Manages the user's shopping cart, including adding, updating, and removing items.
   * Persistent Cart: Maintains the cart state across user sessions.
   * Checkout Preparation: Prepares the cart for the order placement process.
8. **Order Service**
   * Order Lifecycle Management: Oversees order placement, payment, and status updates.
   * Payment Integration: Processes payments securely.
   * Order Tracking: Provides real-time order tracking and history.

**Database Design**

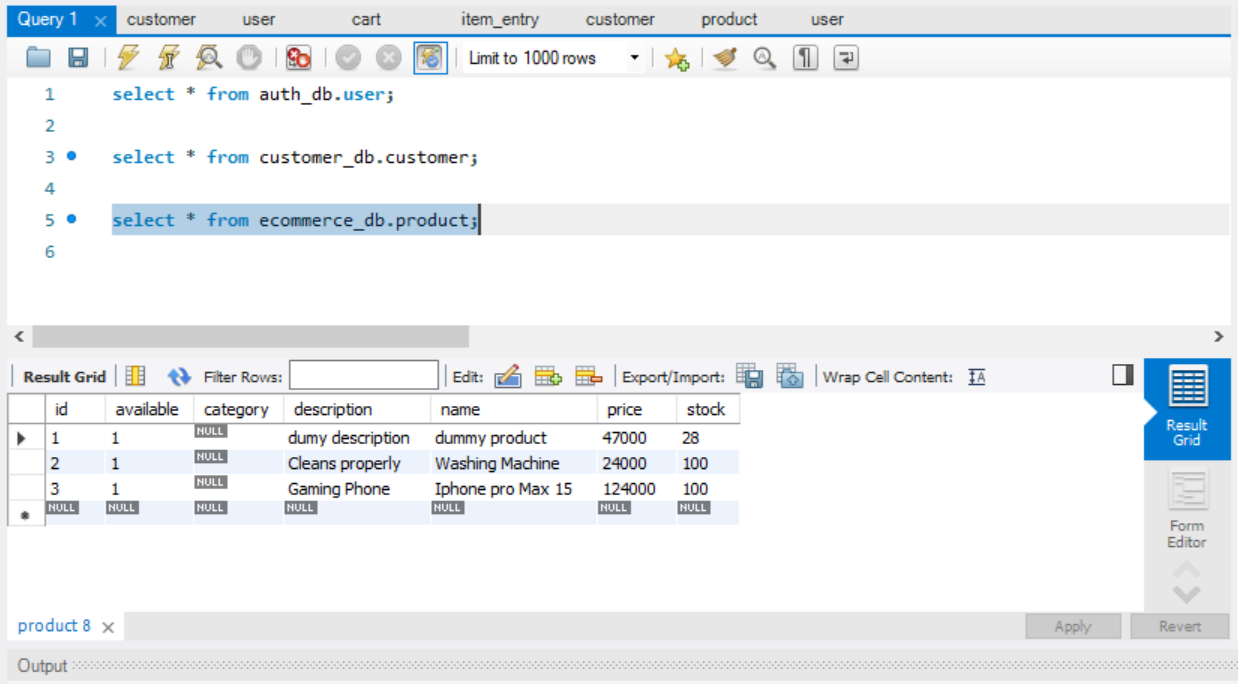
* + **Tables:**
    - users: id, shipping\_address ,user\_name, password, user\_email, role



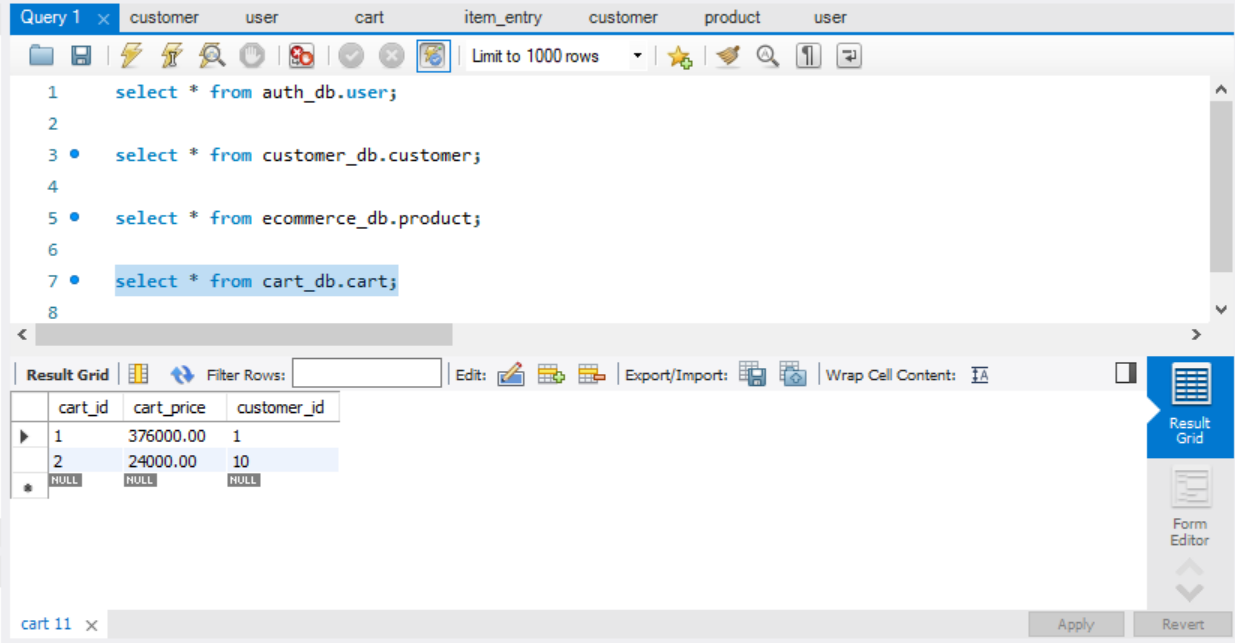
* + - customers: customer\_id, customer\_name, customer\_email, shipping\_address



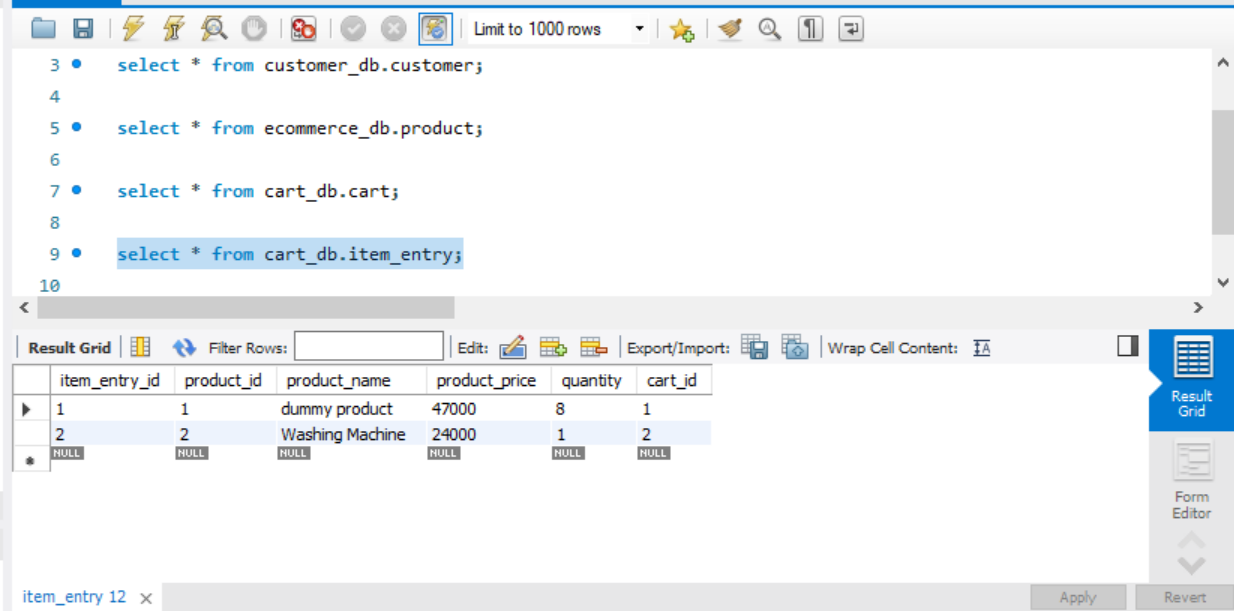
* + - products: id, name, description, price, category, stock, available



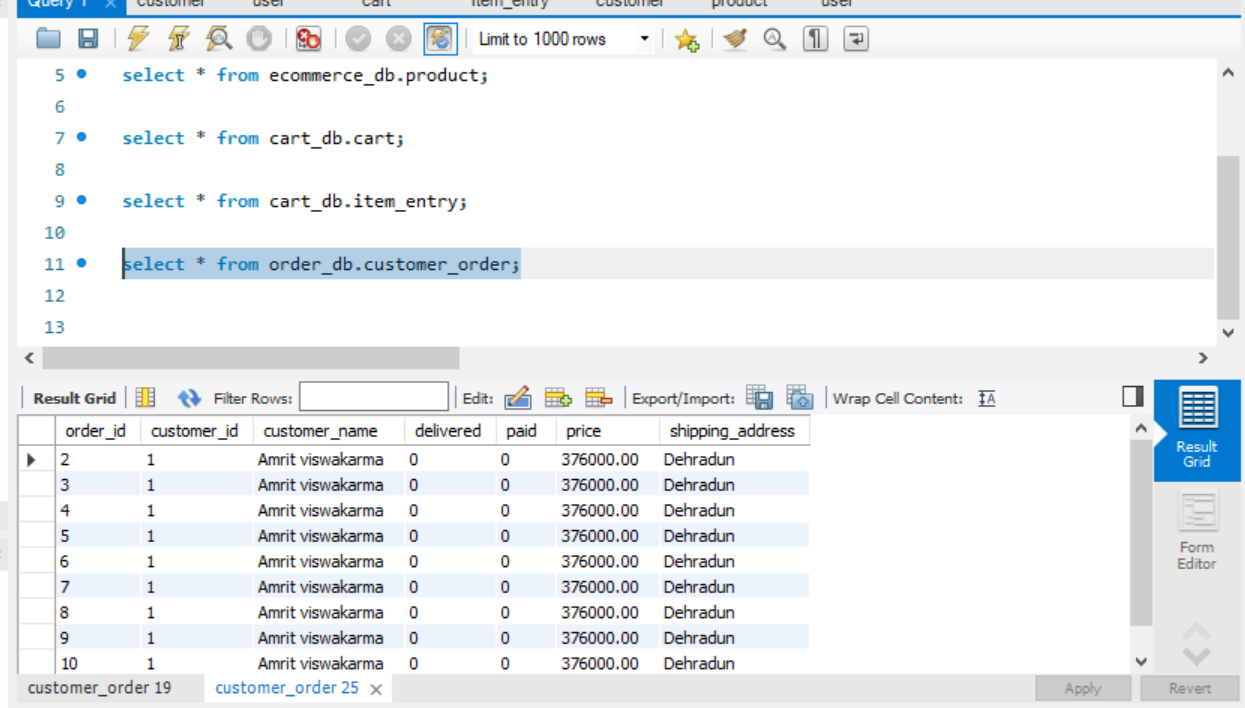
* + - carts: cart\_id, customer\_id, cart\_price



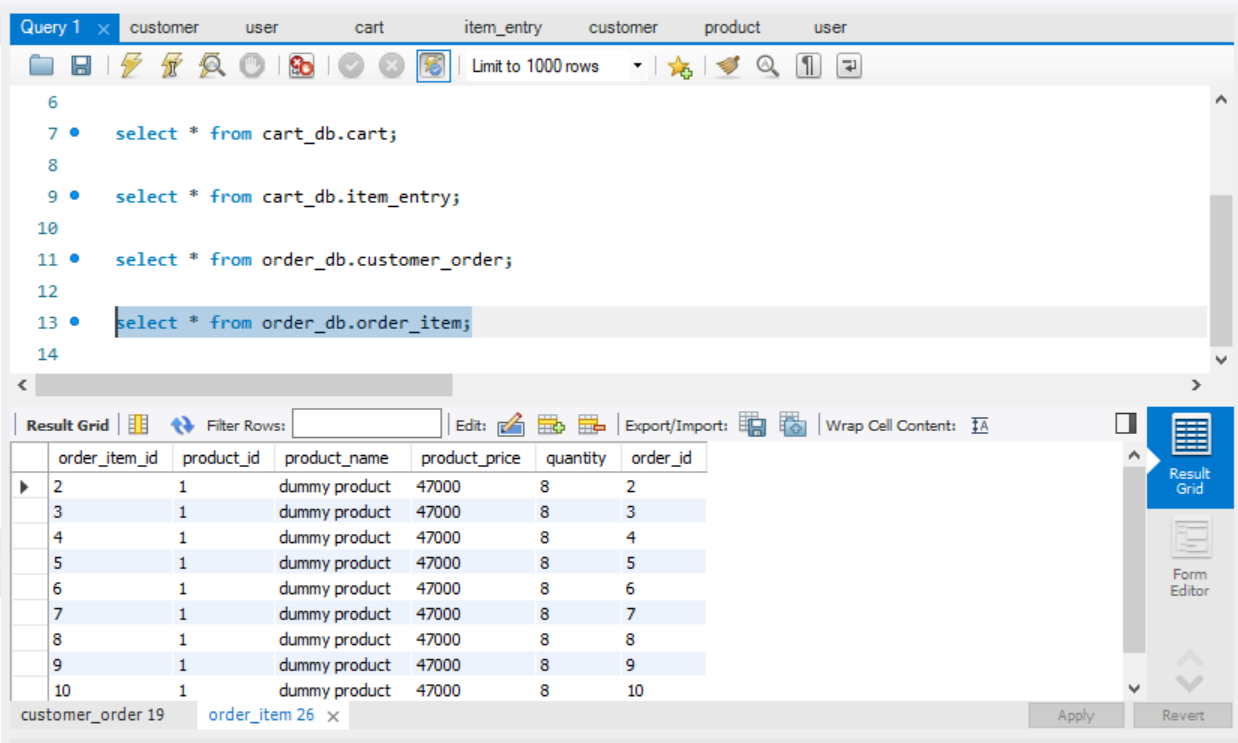
* + - item\_entry:item\_entry\_ id, cart\_id, product\_id, quantity, product\_price, product\_name



* + - customer\_orders: order\_id, customer\_id, customer\_name, delivered, paid, shipping\_address



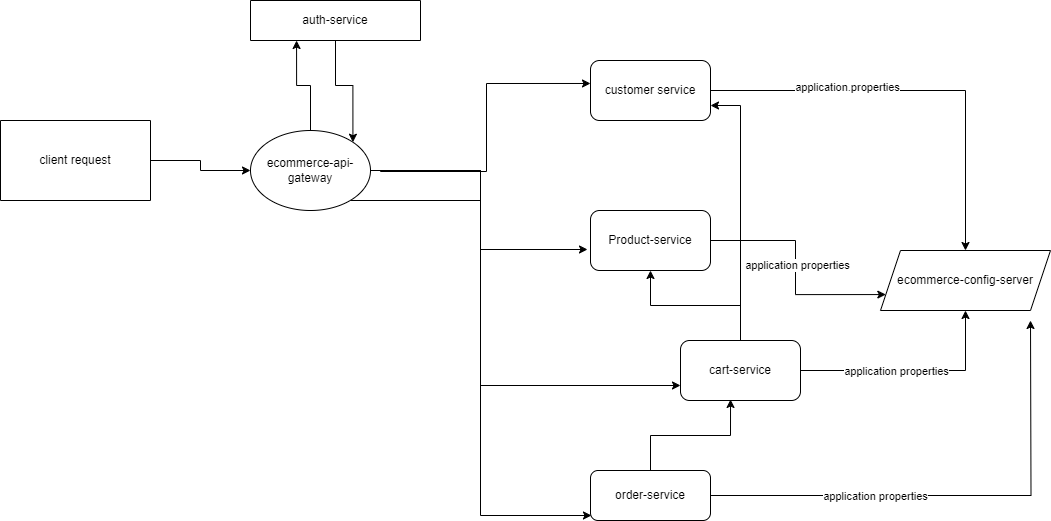
* + - order\_items: order\_item\_id, order\_id, product\_id, product\_price, quantity, product\_name



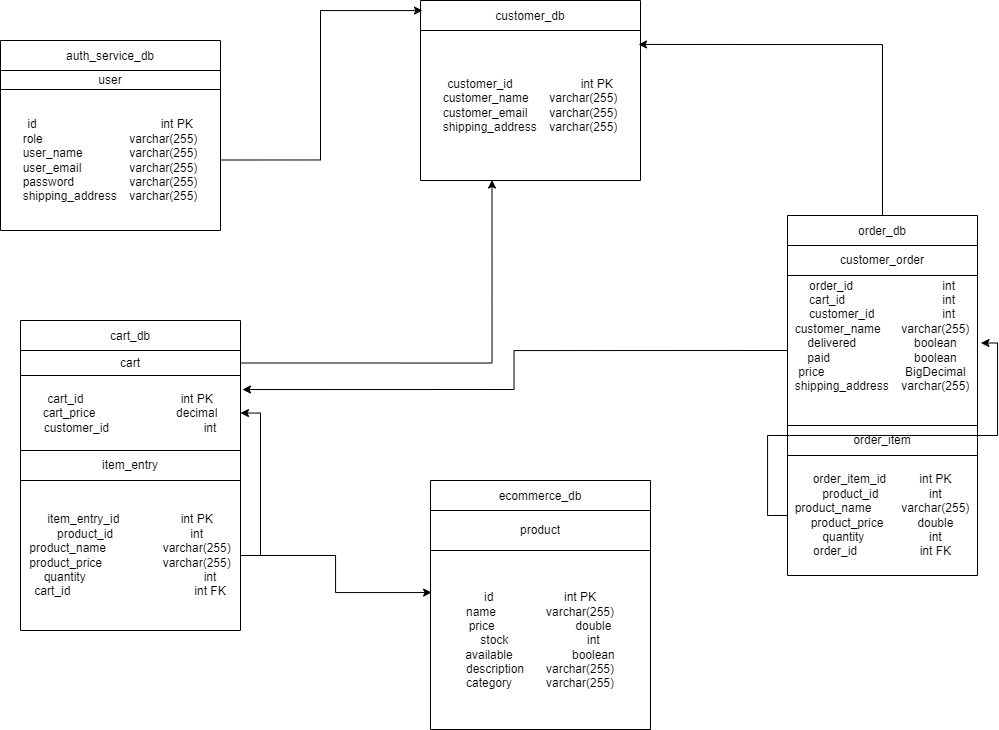
* + **Relationships:**
    - The customers table links to users.
    - The carts table links to users, and item\_entry links to carts and products.
    - The orders table links to customer and carts, and order\_items links to orders and products.

**use case diagrams:-**

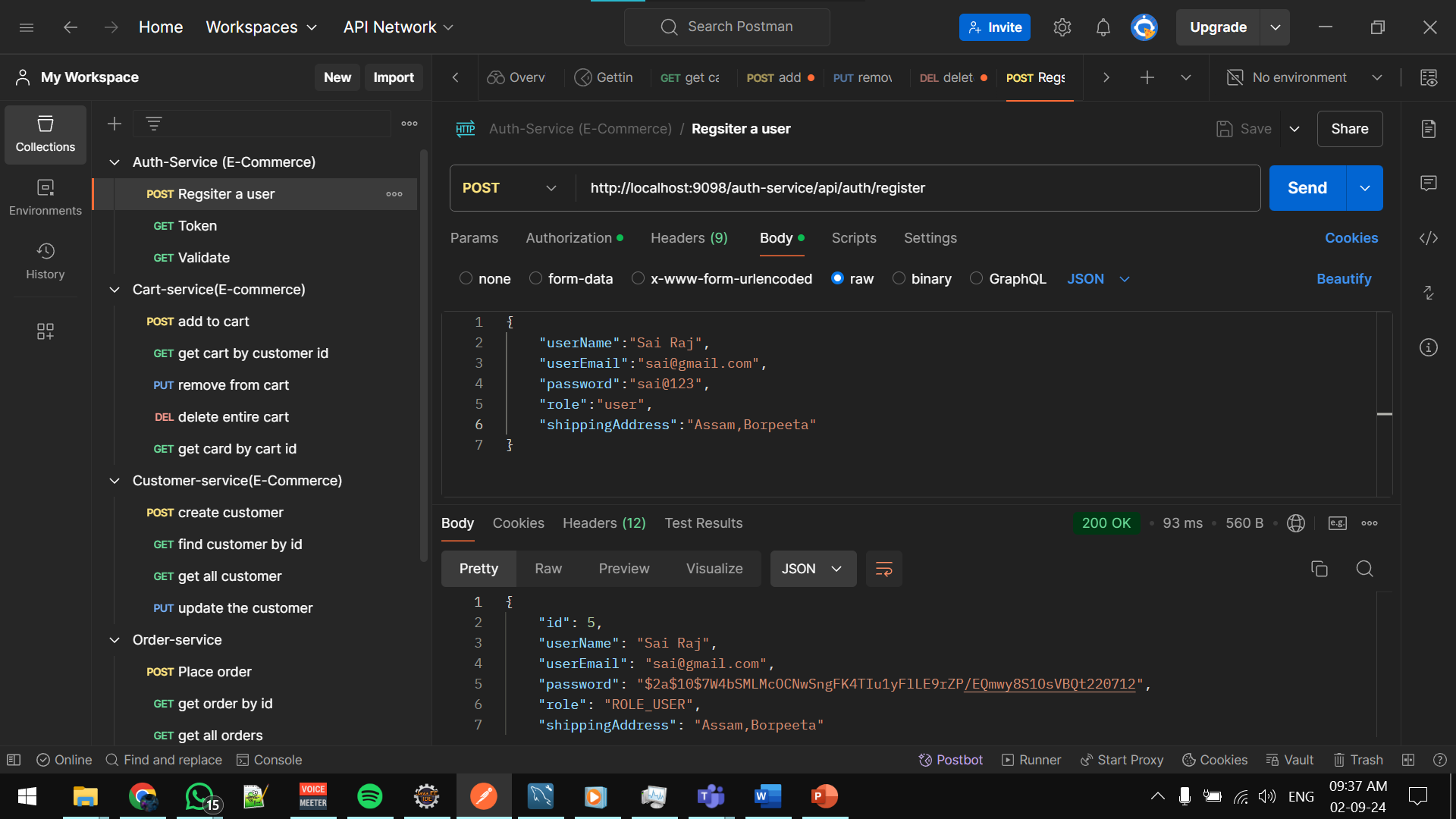
**Flow Diagram:-**



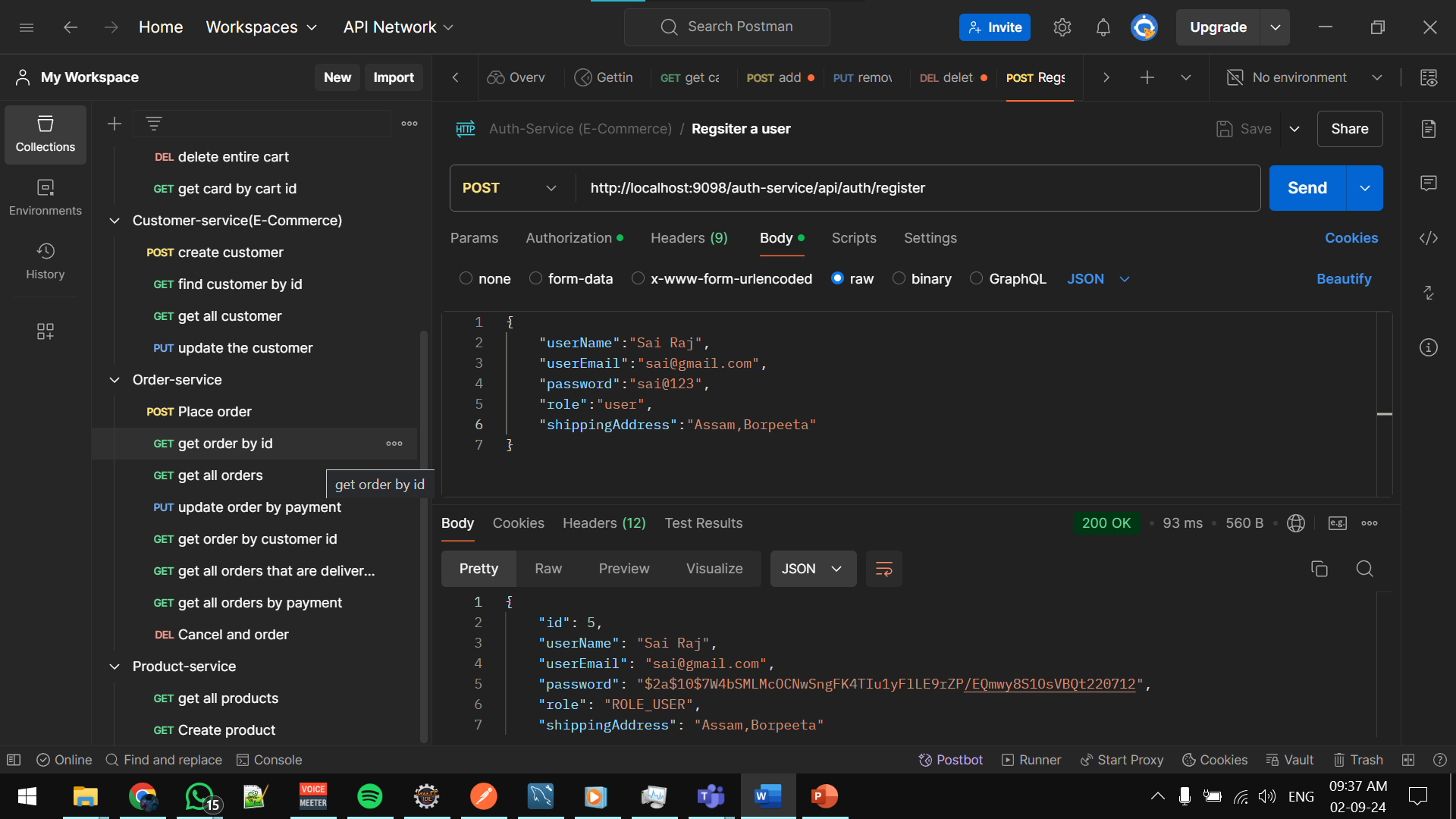
**Database Flow Diagram:-**



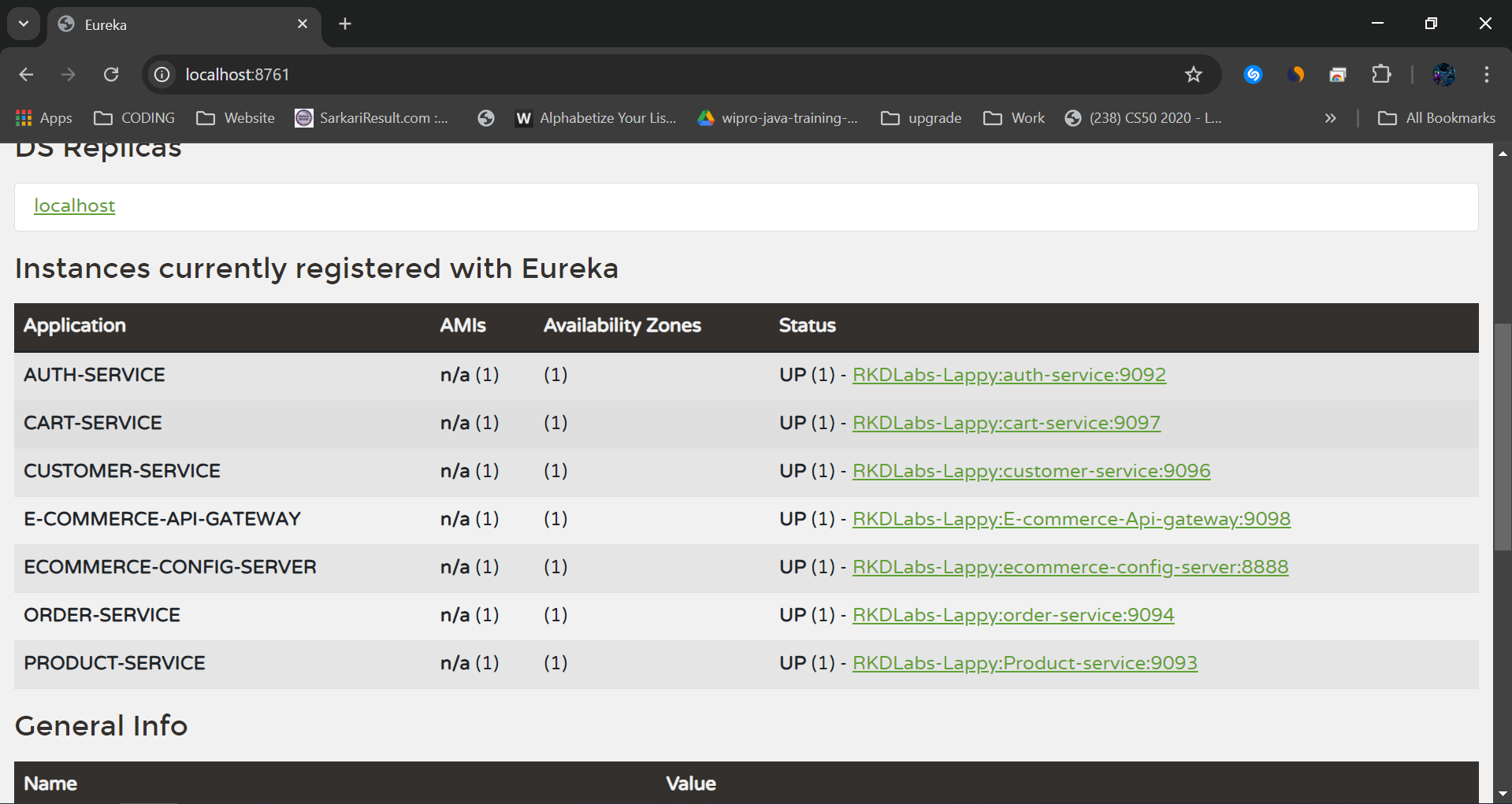
**Postman Folder Structure :- 1.**

****

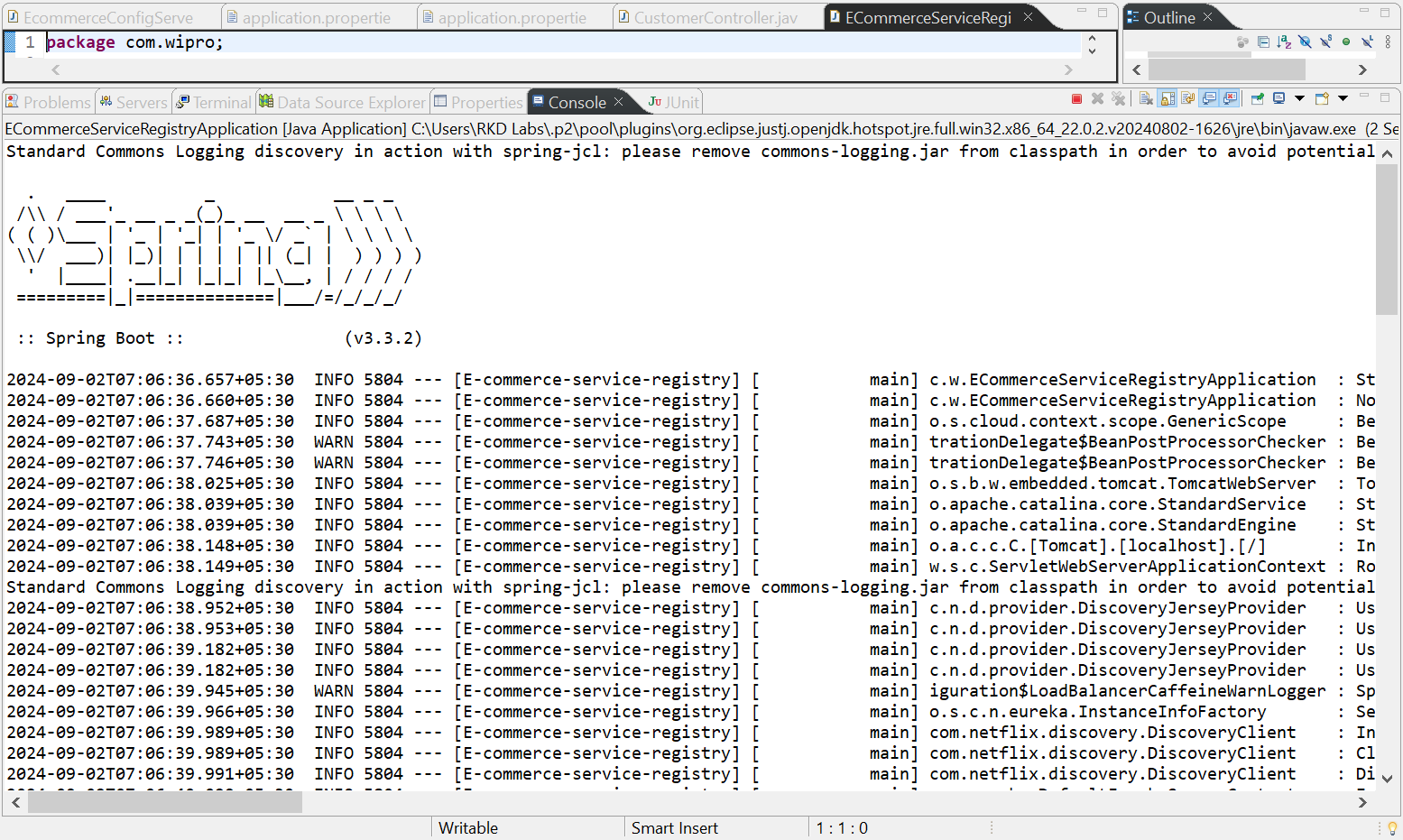
2.

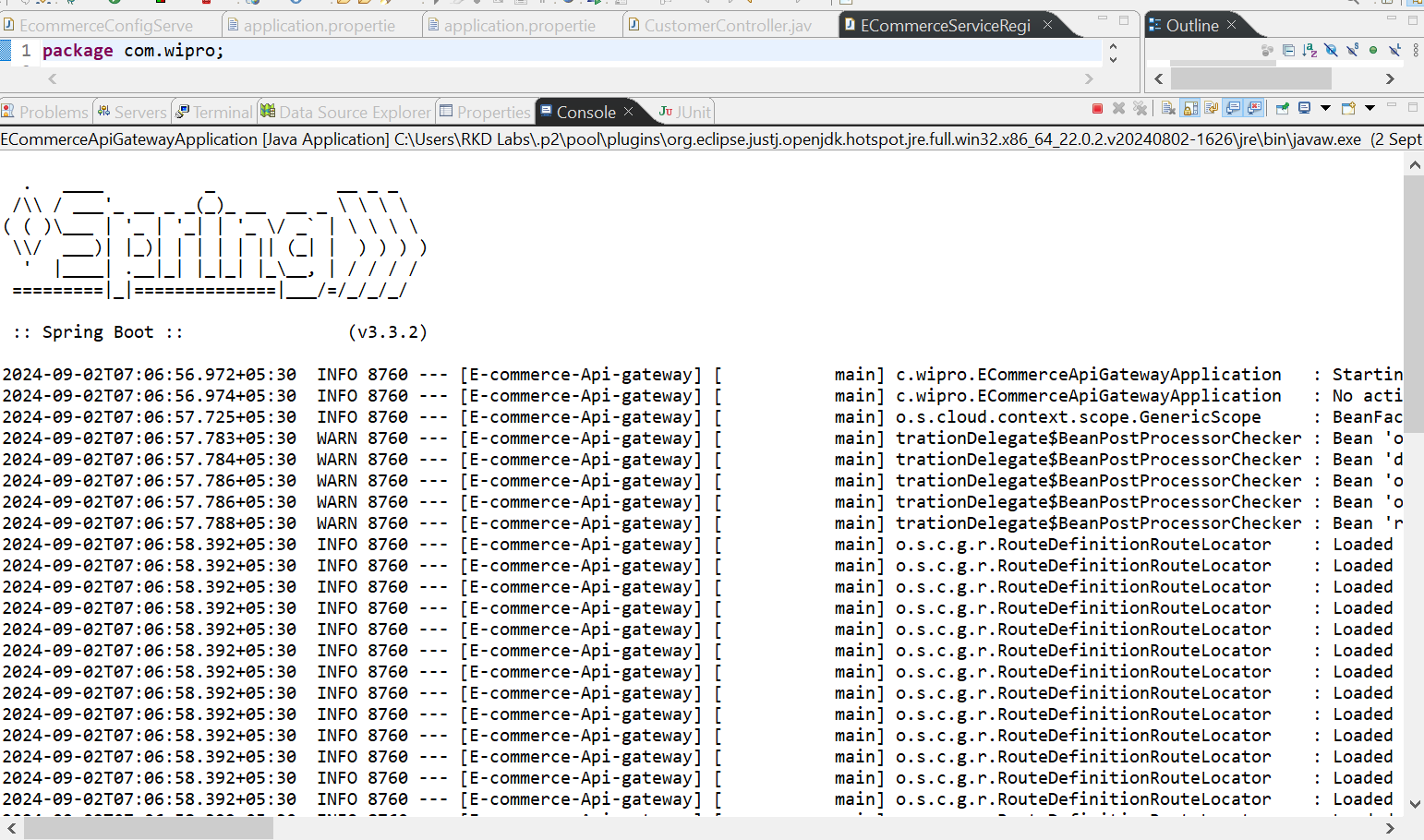


**This is the Eureka Service where we show all Microservices:**

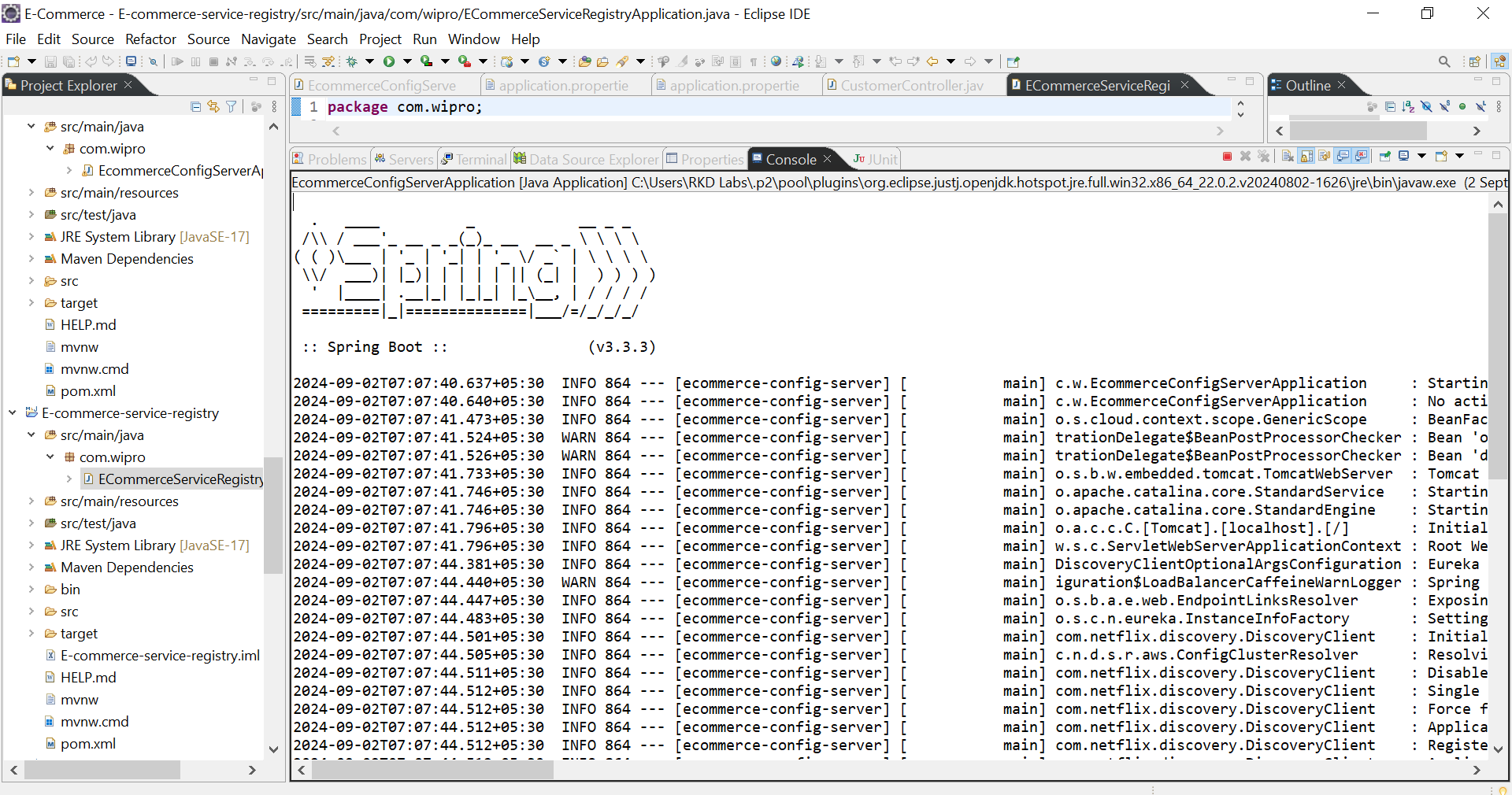


**Running E-commerce-service-registry:-**

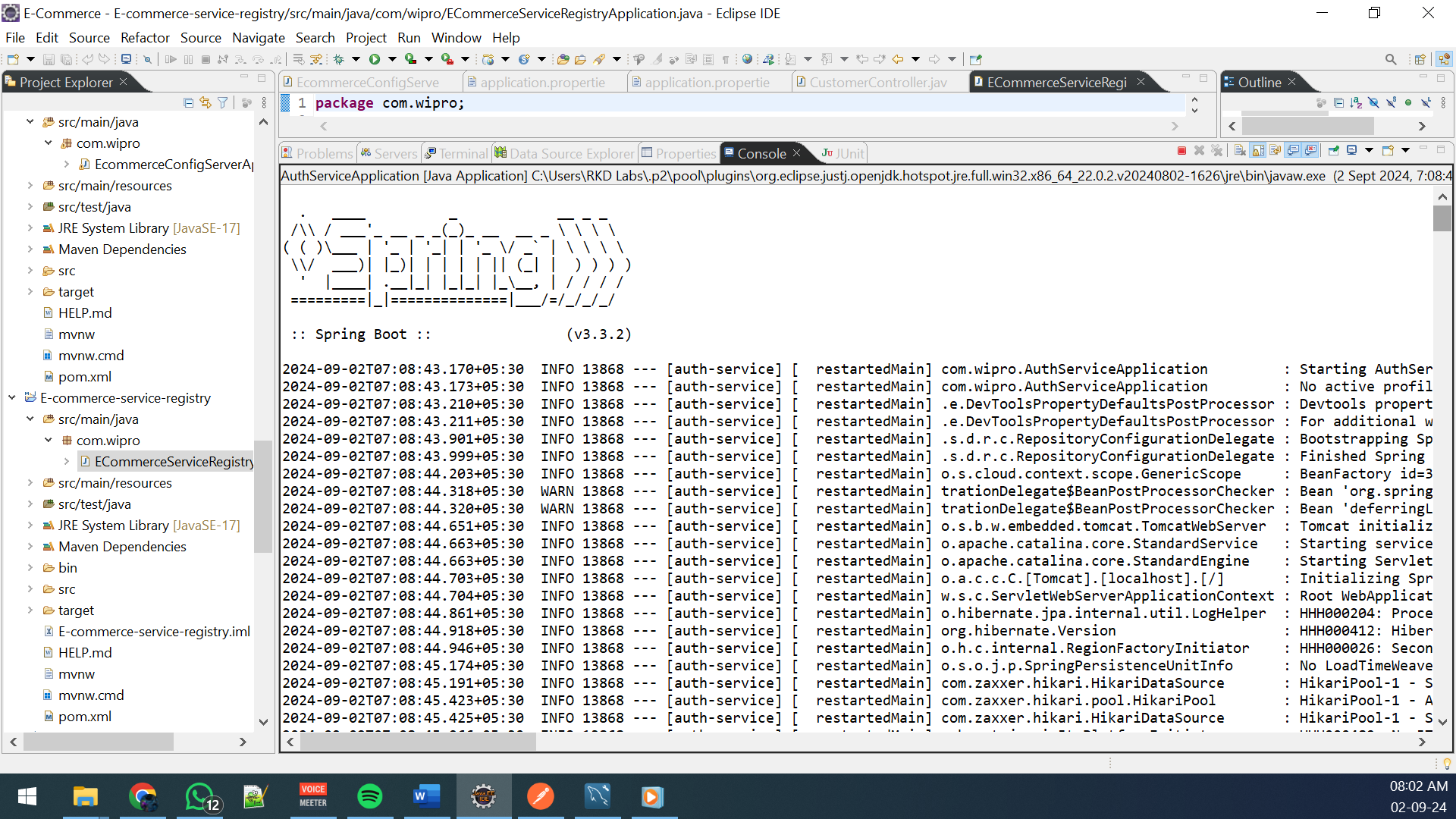
****

**Running E-commerce-ApiGateway:-  
**

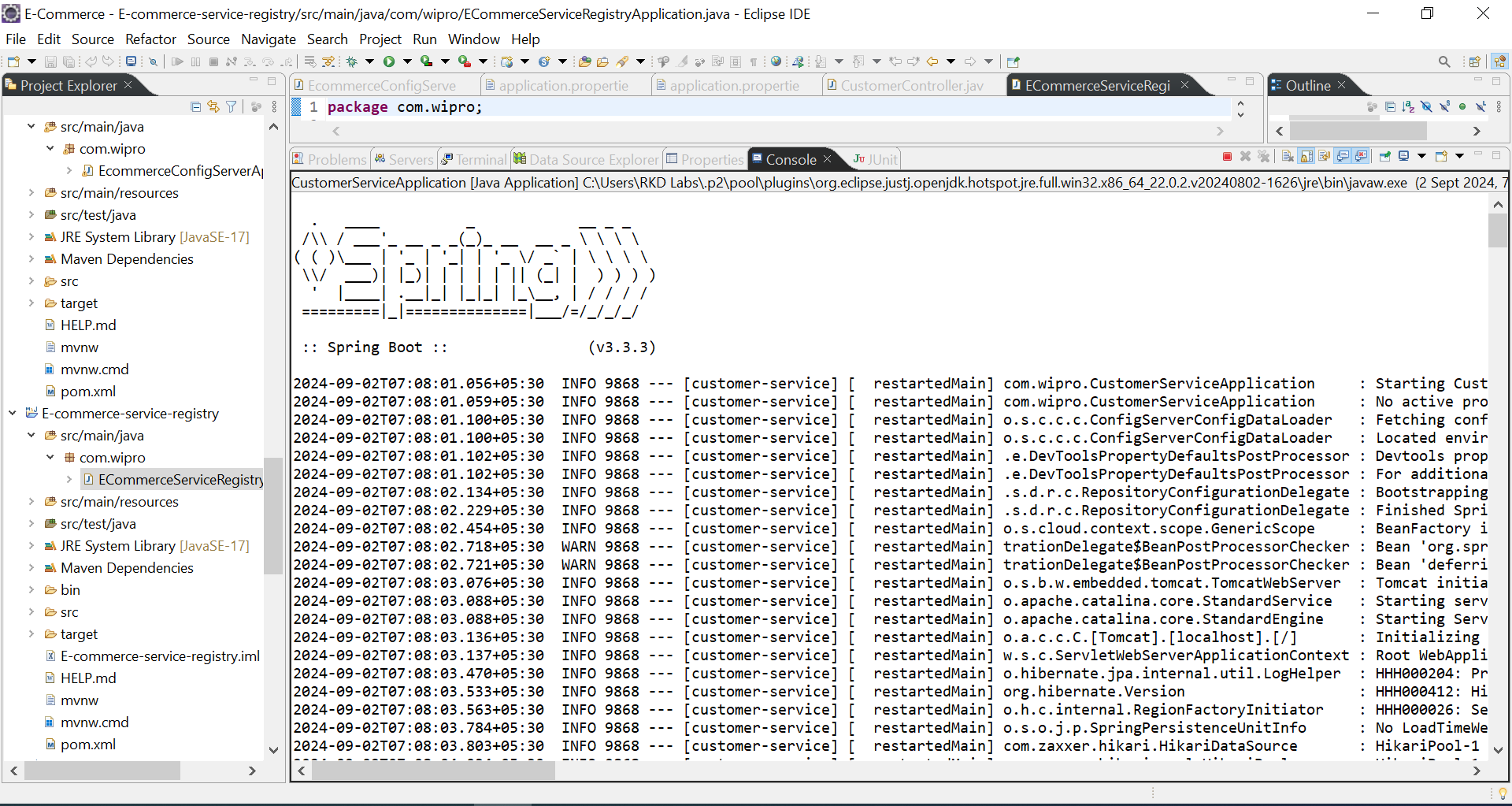
**Running Ecommerce-config-server:-**

****

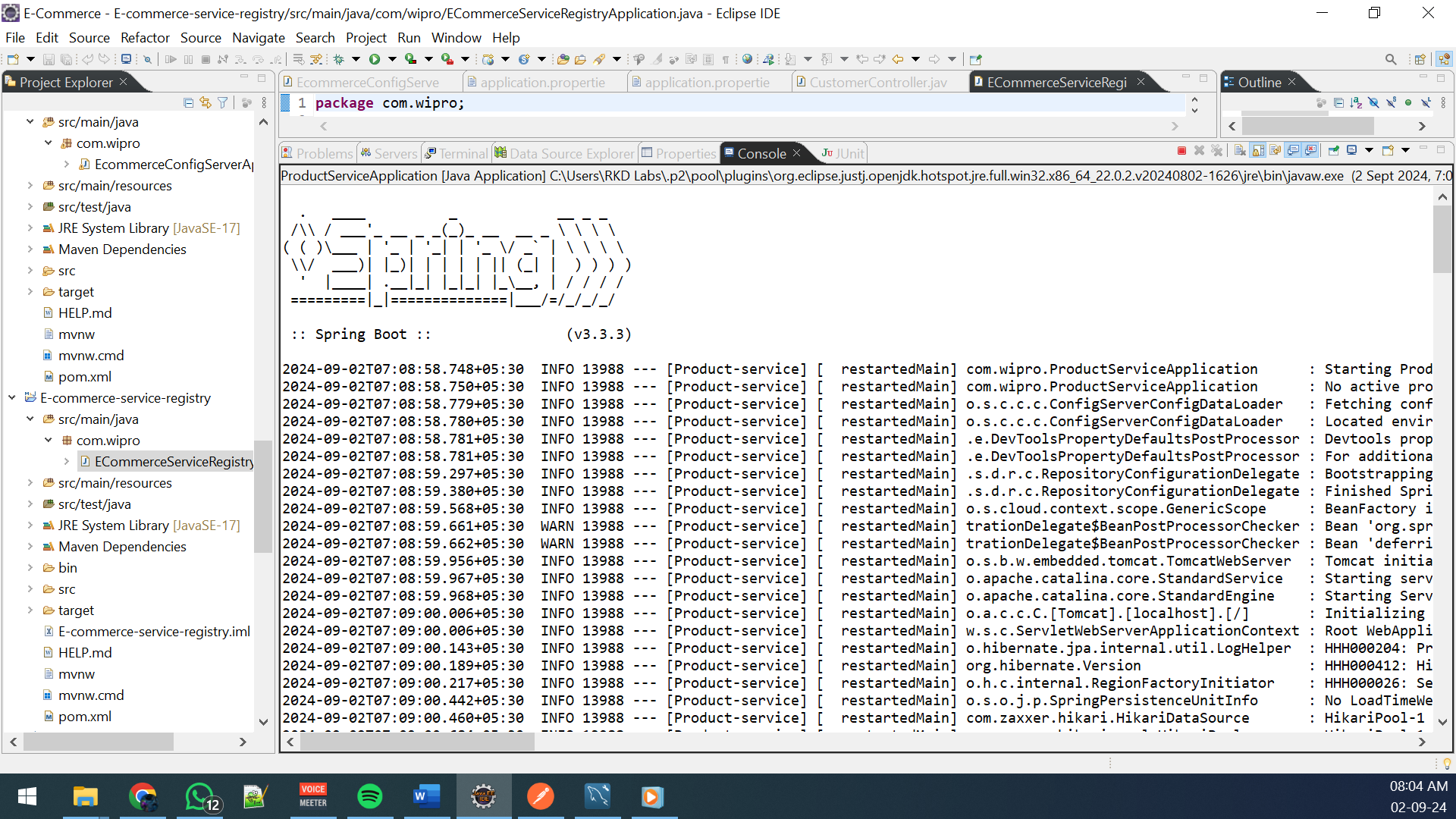
**Running Auth-service:-**



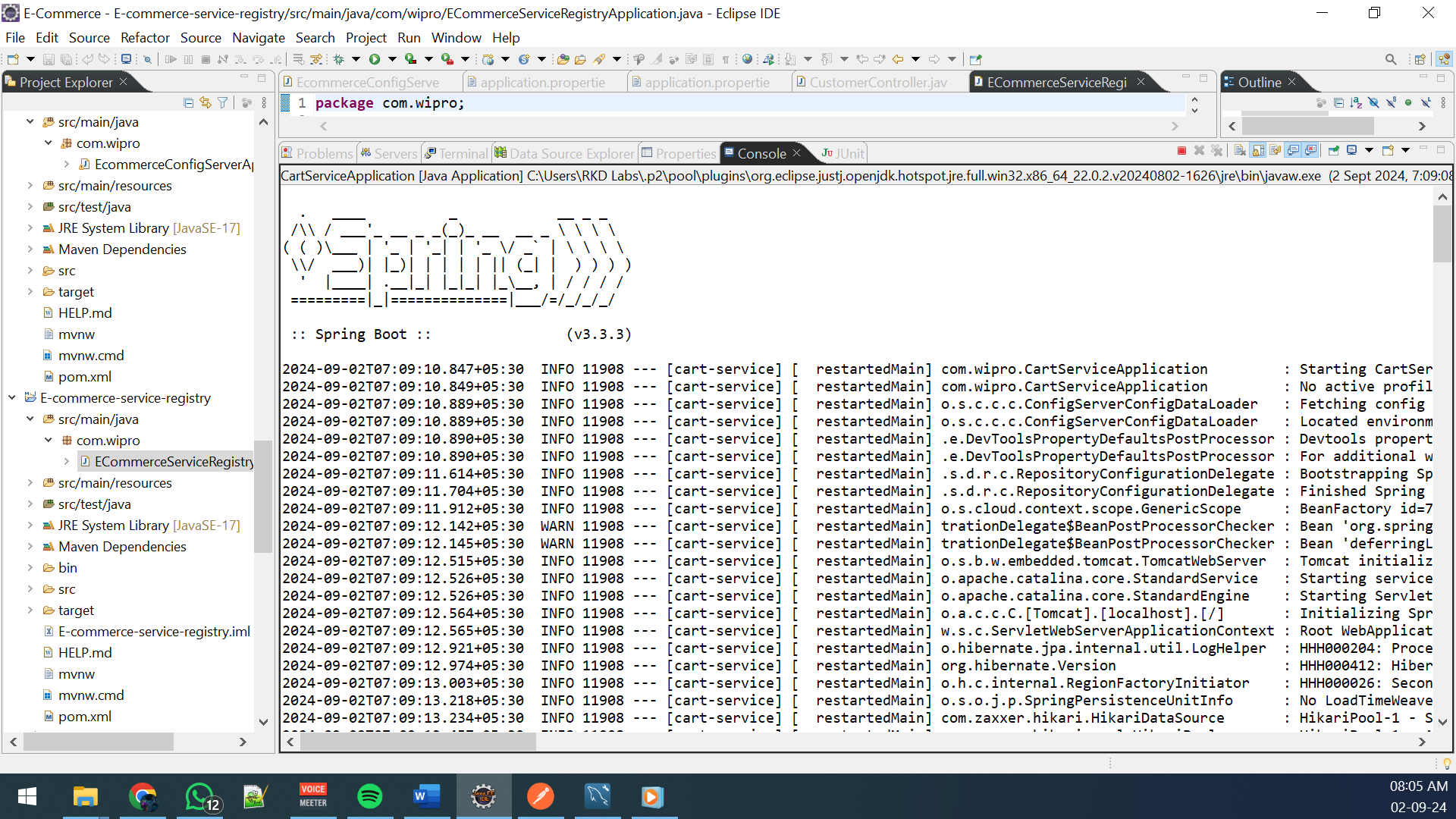
**Running customer-service:-**

****

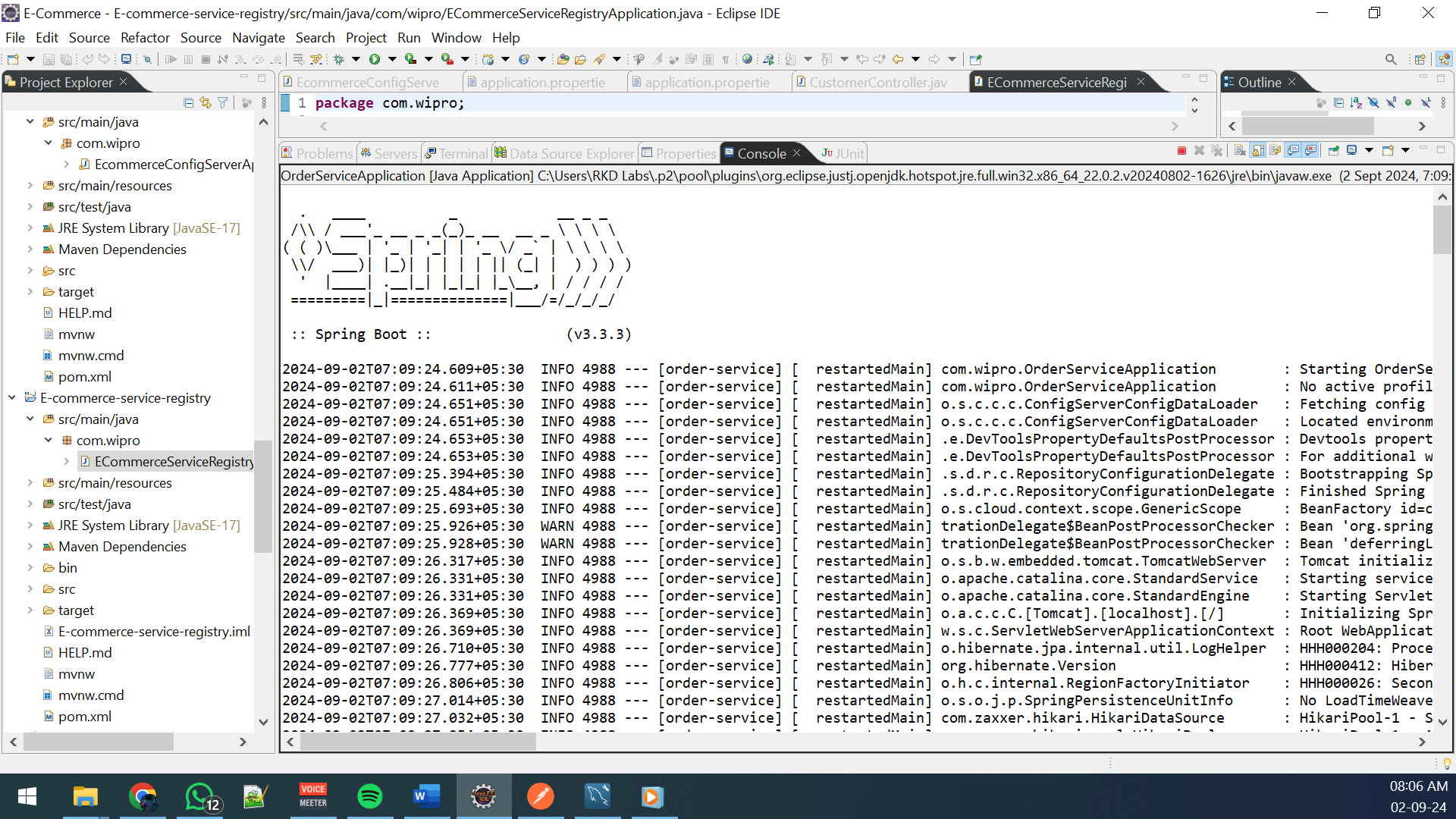
**Running Product-service:-**



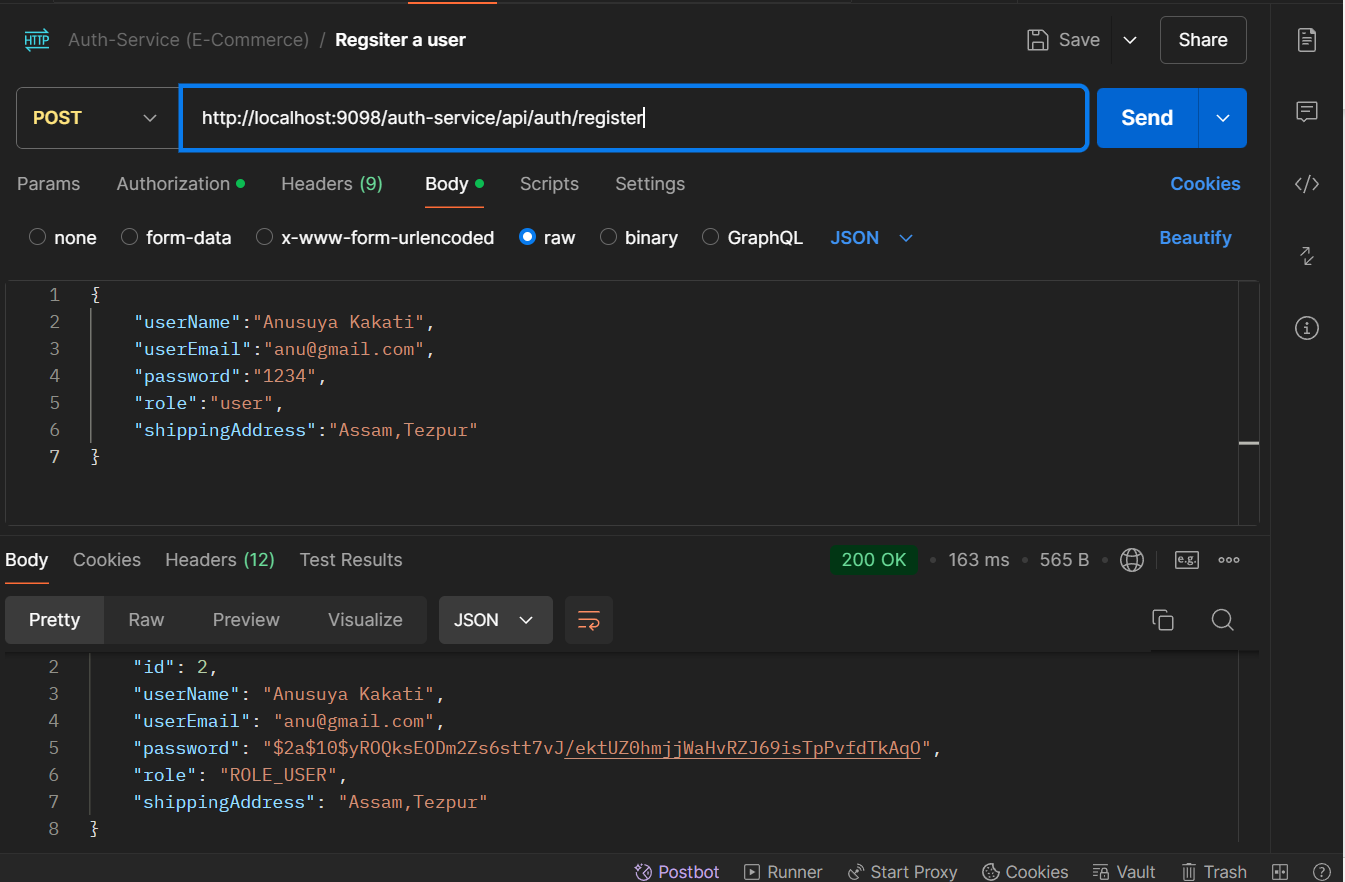
**Running Cart-service:-**

****

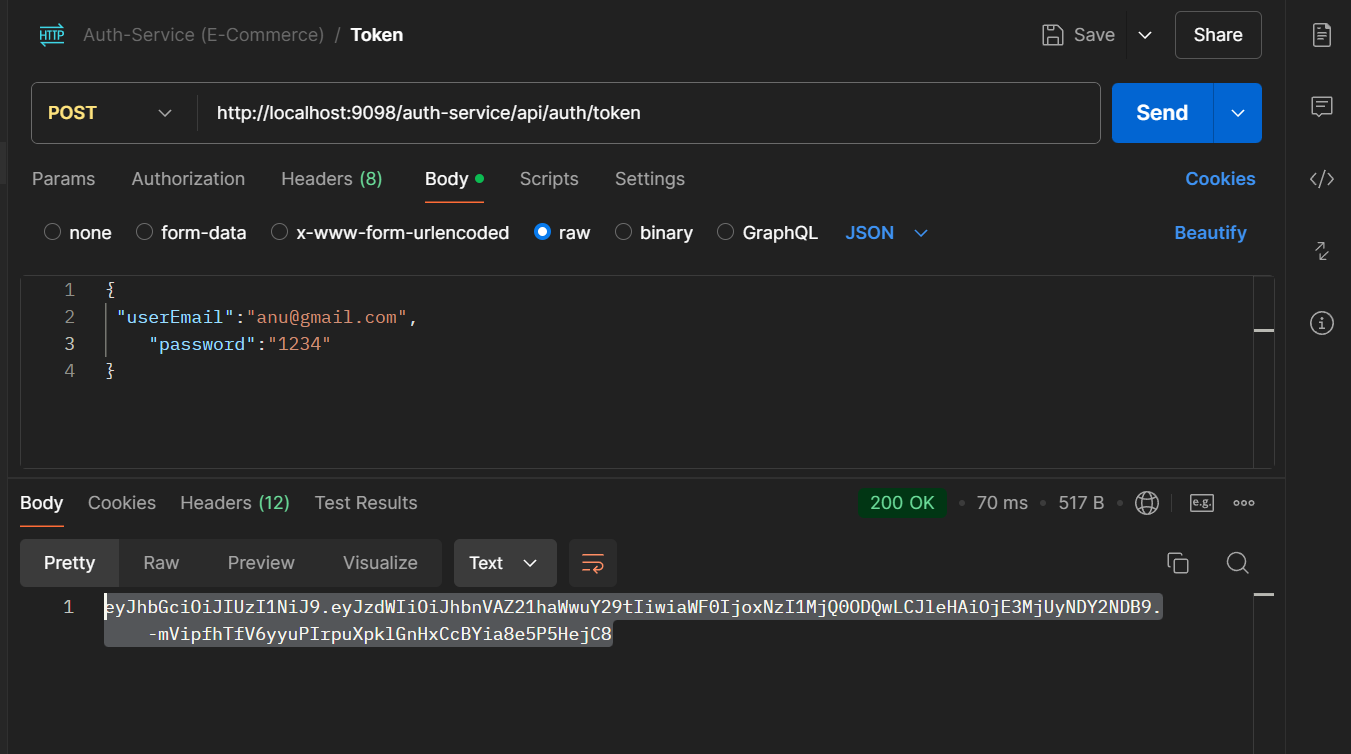
**Running Order-service:-**



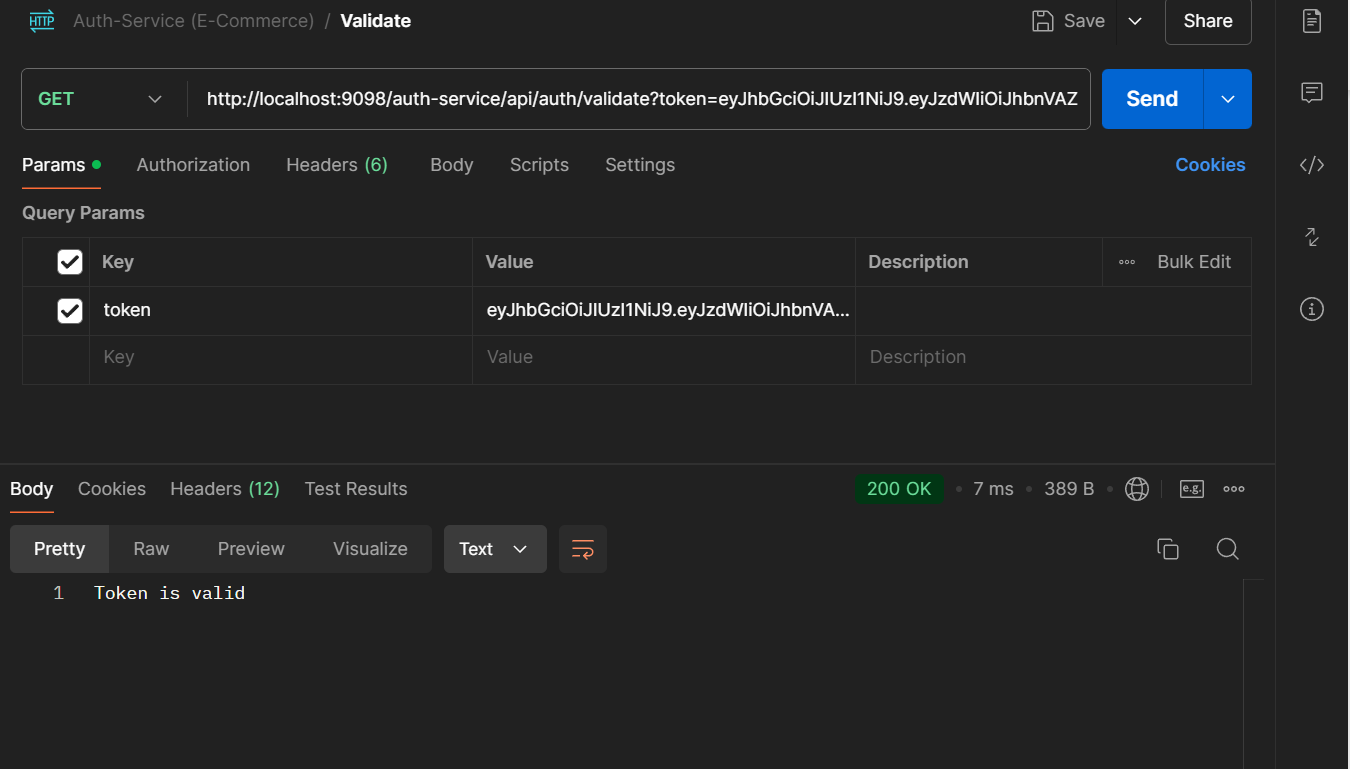
**Registrate User**

****

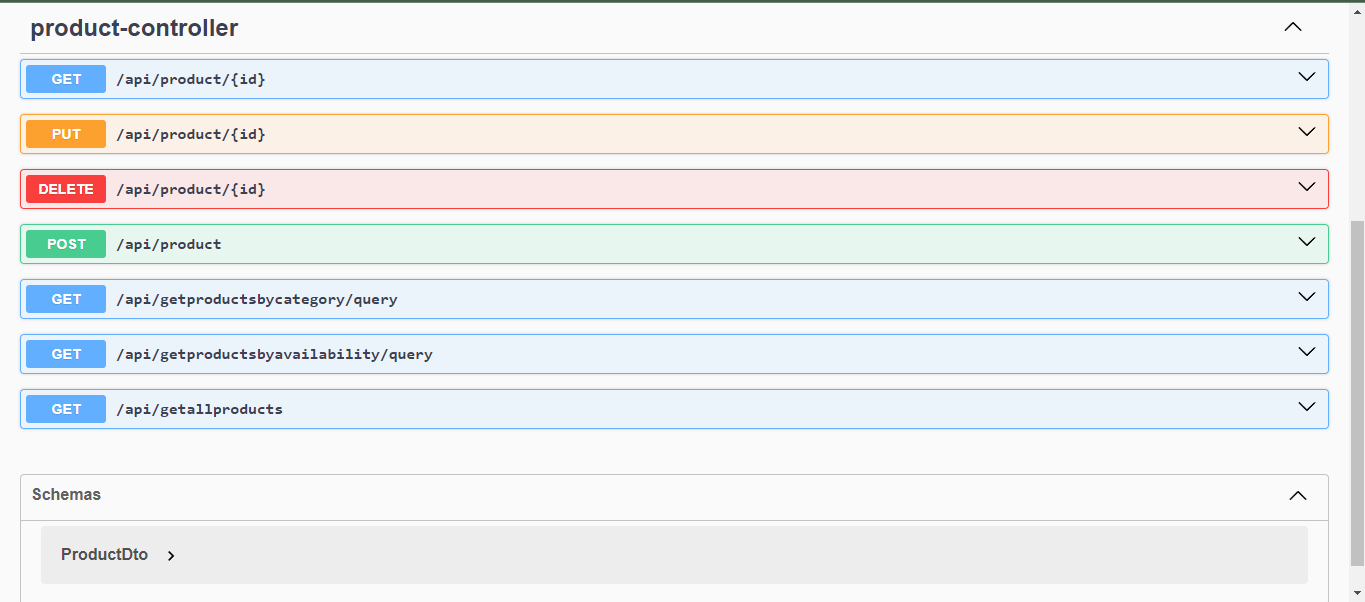
**Token generation**

****

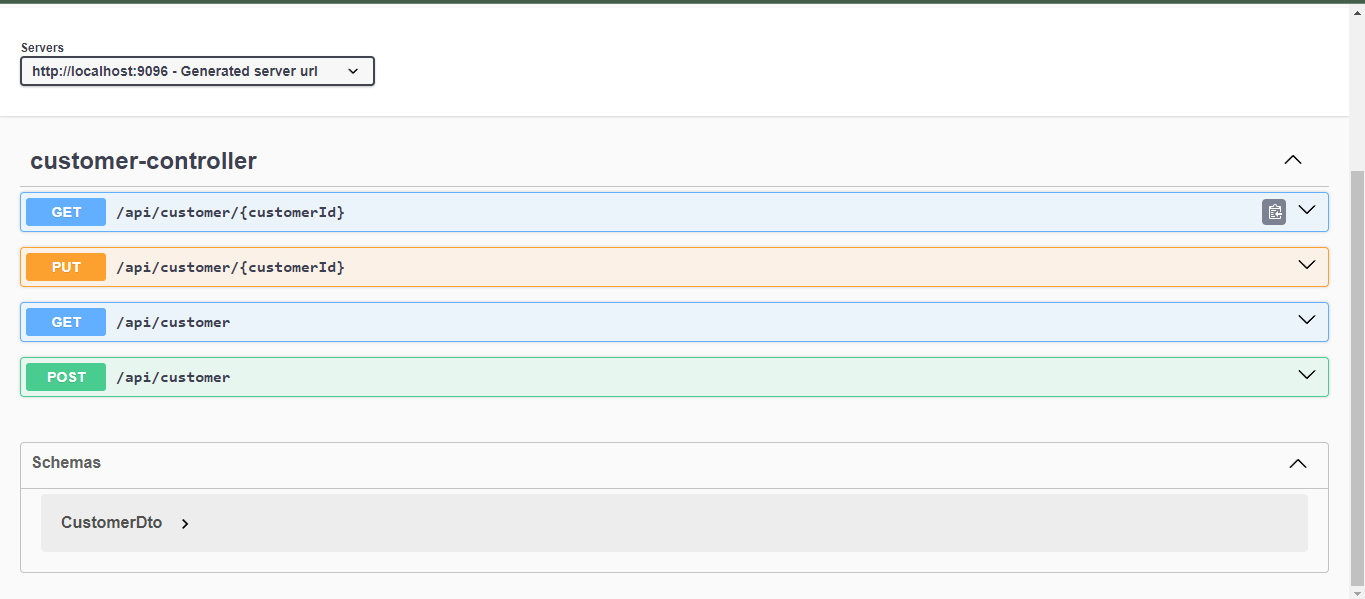
**Token validation**

****

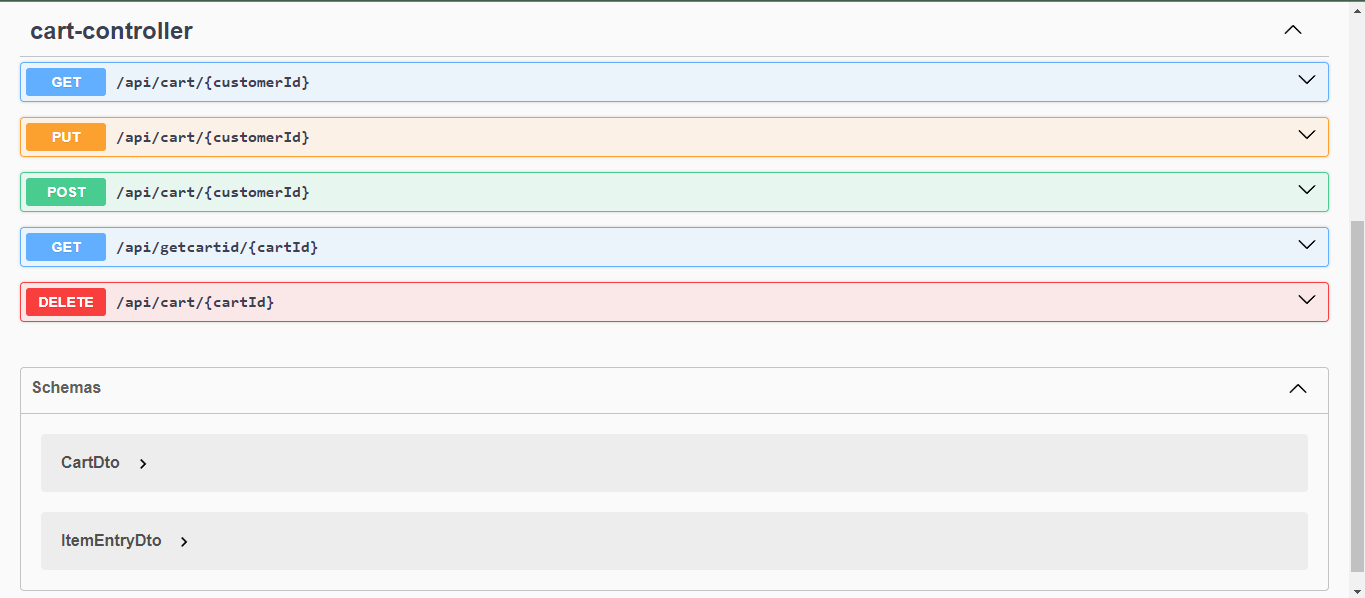
**Product microservice:-**

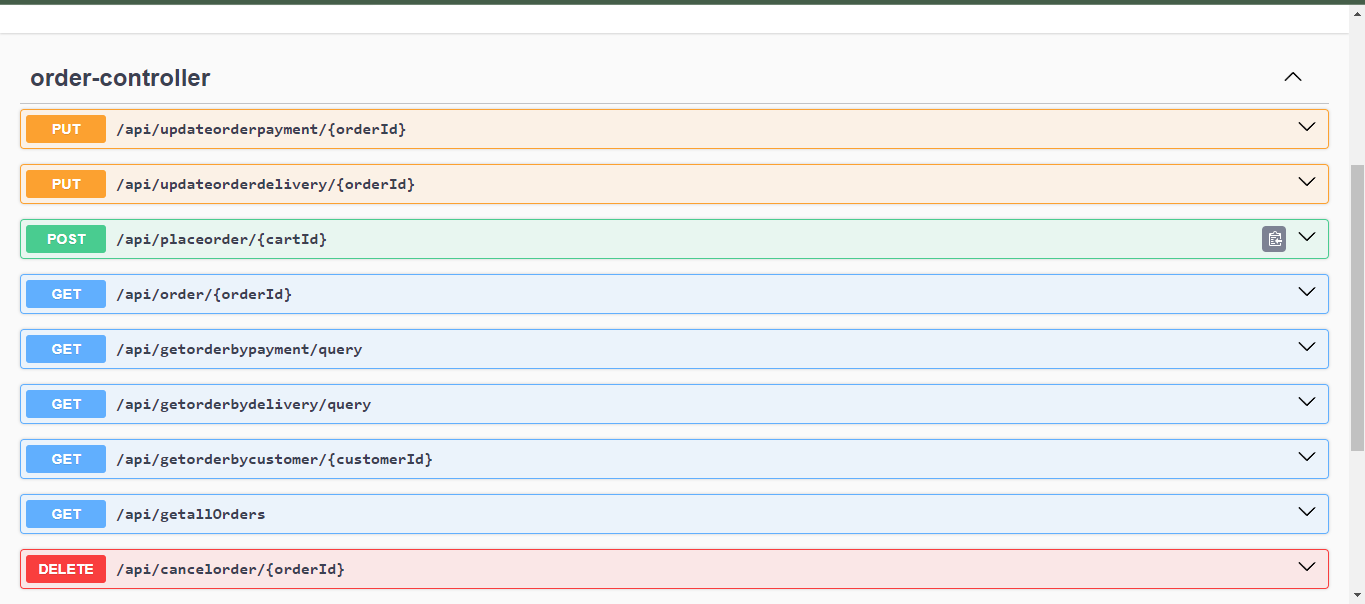


**Customer microservice:-**



**Cart microservice:-**



**Order microservice:-**