

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

Name: ZEBA QUMAR

Batch: 8 (JAVA J2EE)

Group-D (E-commerce-application):

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

- **Requirements Gathering (1 day):**
 - Identify required microservices: Auth-Service, Customer-Service, Product-Service, Cart-Service, Order-Service.
 - Define the interactions between services via API Gateway.
- **Architecture Design (1 day):**
 - Design the microservices architecture.
 - Set up the Eureka Server for Service Registry and Discovery.
 - Set up the API Gateway for routing requests.

Development:

- **Day 1:**
 - **Eureka Server & API Gateway Setup:**
 - Set up Eureka Server for service registration and discovery.
 - Implement API Gateway to route requests to microservices.
- **Day 2:**
 - **Customer-Service Development:**
 - Develop CRUD operations for customer profiles.
 - Integrate with Auth-Service to handle customer data upon registration.
- **Day 3:**
 - **Product-Service Development:**
 - Build functionalities for managing products, including categories and search.
 - Implement APIs for product CRUD operations.
- **Day 4-5:**
 - **Cart-Service Development:**
 - Develop features for managing the shopping cart, including adding, updating, and deleting items.

- Integrate with Product-Service to fetch product details.
 - Implement the item_entry table to store individual cart items.

- **Day 6:**
 - **Order-Service Development:**
 - Implement order processing functionalities, including placing orders and tracking status.
 - Integrate with Cart-Service to process the cart items during order placement.

- **Day 7-8:**
 - **Auth-Service Development:**
 - Create registration and login functionalities.
 - Implement JWT token generation and validation for secure access.

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 his

Database Design

- **Tables:**

- users: id, shipping_address ,user_name, password, user_email, role

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 •  SELECT * FROM auth_service_db.user;
```

The result grid displays the following data:

	id	password	role	shipping_address	user_email	user_name
▶	1	\$2a\$10\$Yc08D6SorRJQ4VJTN09mOLWO53hom...	ROLE_USER	Mohan nagar,UP	shark@126	sharik
2	2	\$2a\$10\$onRvVy8AYlGBBrKOrhZ1ne2rlqPjdxo...	ROLE_USER	bakan,rajasthan	toshi@1902	toshi
52	52	\$2a\$10\$ihMWGyqv9xkCFIk.JO61vefpCpBS6Ao...	ROLE_USER	jaipur,rajasthan	anaya@1902	anaya
102	102	\$2a\$10\$4loujark2Ox3d1Ed8nPol.OHAB2nveRFp...	ROLE_USER	krishna nagar,delhi	kavita@1500	kavita
152	152	\$2a\$10\$iTnB5kG29UL/rHOn/SIDHOd.uc4/cW3Y...	ROLE_USER	saket,delhi	sunita@1504.com	sunita
*	NULL	NULL	NULL	NULL	NULL	NULL

- customers: customer_id, customer_name, customer_email, shipping_address

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 •  SELECT * FROM customer_db.customer;
```

The result grid displays the following data:

	customer_id	customer_email	customer_name	shipping_address
▶	102	sahil@123	sahil	karol bagh,delhi
	152	shark@126	shark	Mohan nagar,UP
	202	toshi@1902	toshi	bakan,rajasthan
	252	anaya@1902	anaya	jaipur,rajasthan
	302	kavita@1500	kavita	krishna nagar,delhi
	352	sunita@1504.com	sunita	saket,delhi
*	NULL	NULL	NULL	NULL

- o products: id, name, description, price, category, stock, available

1 • `SELECT * FROM ecommerce_db.product;`

The screenshot shows the MySQL Workbench interface with a results grid. The grid has columns: id, available, category, description, name, price, and stock. The data includes:

	id	available	category	description	name	price	stock
▶	2	1	vehicle	Mahindra designed car	car	450000	10
	3	1	Books	true story based novel	A love story	400	0
	52	1	mobile	A Smartphone made by Apple that combines a ...	Iphone 15	69600	5
	53	1	mobile	Samsung mobile phones are known for their rob...	Samsung Galaxy S23	46999	25
	102	1	Makeup	24hr moisture and waterproof glossy lipstick	Lakme 9to5 Lipstick	259	23
	152	1	cosmetic	sulphate free shampoo that prevent to reduce ...	Tresemme Shampoo	687	58
	153	1	cosmetic	repair your rough skin and feel like smooth cream	Ponds winter creame	45	9
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

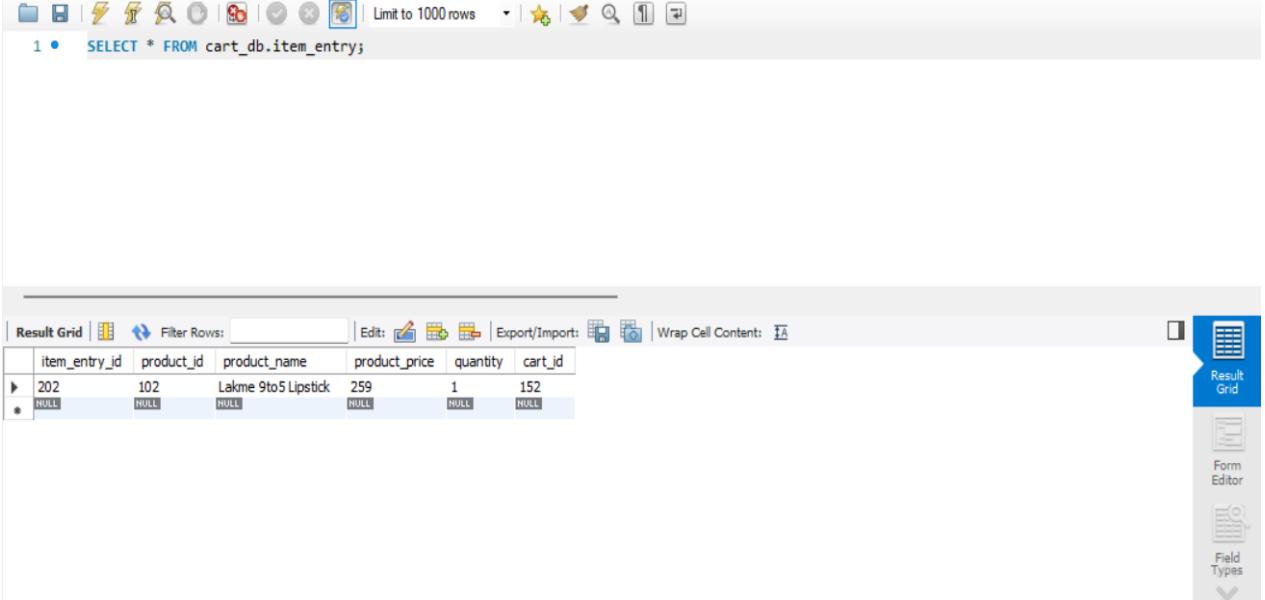
- o carts: cart_id, customer_id, cart_price

1 • `SELECT * FROM cart_db.cart;`

The screenshot shows the MySQL Workbench interface with a results grid. The grid has columns: cart_id, cart_price, and customer_id. The data is:

	cart_id	cart_price	customer_id
▶	152	259.00	152
*	NULL	NULL	NULL

- o item_entry:item_entry_id, cart_id, product_id, quantity, product_price, product_name

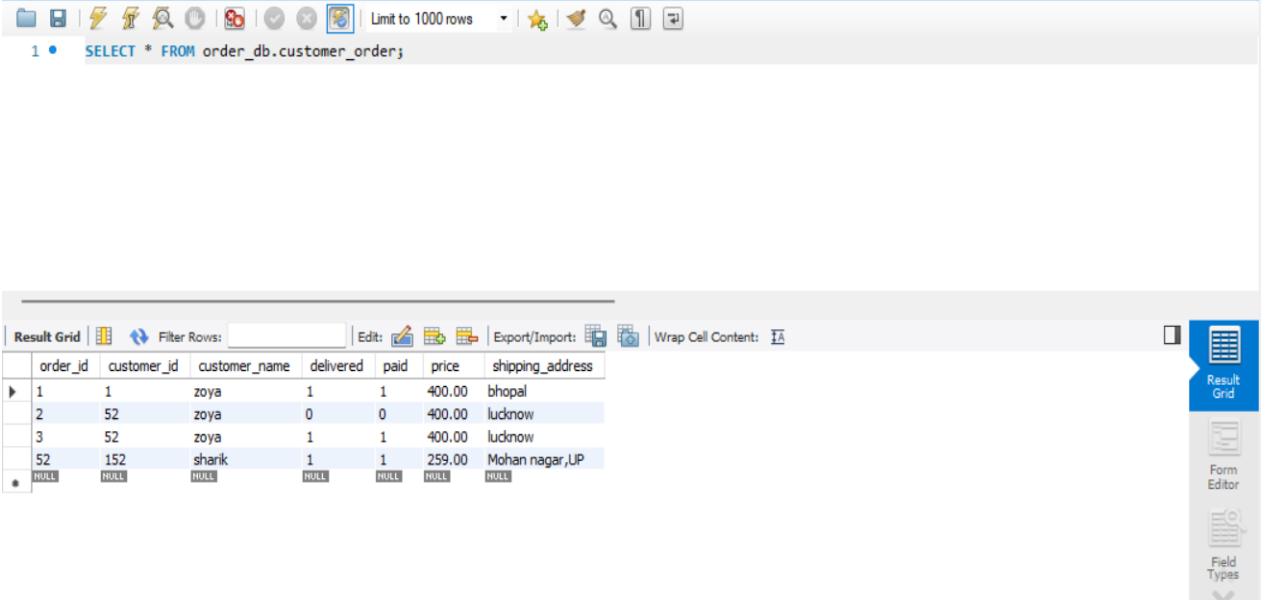


The screenshot shows the MySQL Workbench interface with a query editor at the top containing the SQL command: `SELECT * FROM cart_db.item_entry;`. Below the query is the result grid, which displays one row of data:

item_entry_id	product_id	product_name	product_price	quantity	cart_id
202	102	Lakme 9to5 Lipstick	259	1	152

On the right side of the interface, there is a vertical toolbar with three buttons: "Result Grid" (selected), "Form Editor", and "Field Types".

- o customer_orders: order_id, customer_id, customer_name, delivered, paid, shipping_address



The screenshot shows the MySQL Workbench interface with a query editor at the top containing the SQL command: `SELECT * FROM order_db.customer_order;`. Below the query is the result grid, which displays five rows of data:

order_id	customer_id	customer_name	delivered	paid	price	shipping_address
1	1	zoya	1	1	400.00	bhopal
2	52	zoya	0	0	400.00	lucknow
3	52	zoya	1	1	400.00	lucknow
52	152	sharik	1	1	259.00	Mohan nagar,UP

On the right side of the interface, there is a vertical toolbar with three buttons: "Result Grid" (selected), "Form Editor", and "Field Types".

- o order_items: order_item_id, order_id, product_id, product_price, quantity, product_name

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the SQL command: `SELECT * FROM order_db.order_item;`. The result grid displays the following data:

order_item_id	product_id	product_name	product_price	quantity	order_id
1	3	A love story	400	1	1
2	3	A love story	400	1	2
3	3	A love story	400	1	3
52	102	Lakme 9to5 Lipstick	259	1	52
NULL	NULL	NULL	NULL	NULL	NULL

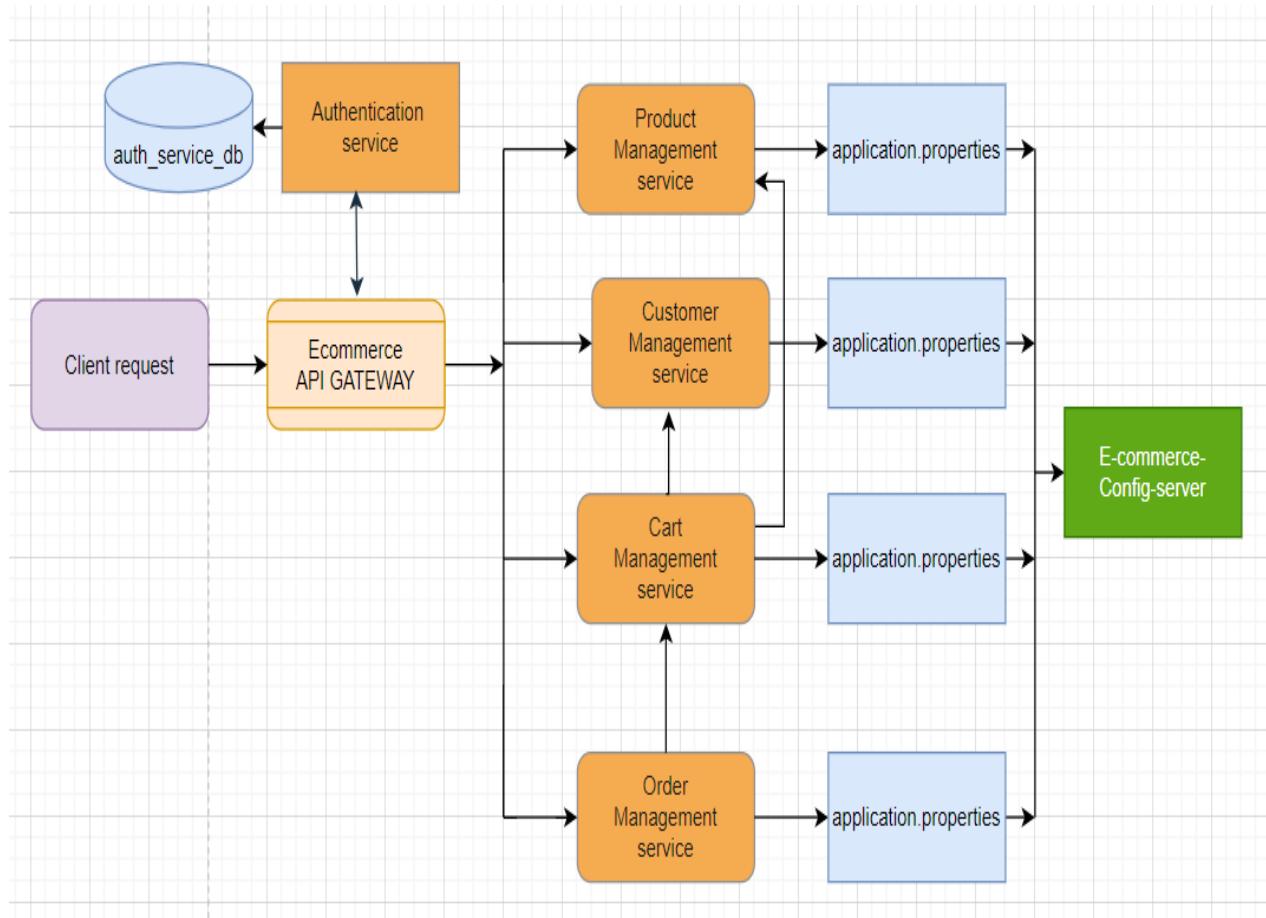
On the right side of the result grid, there is a vertical toolbar with three options: 'Result Grid' (selected), 'Form Editor', and 'Field Types'.

- **Relationships:**

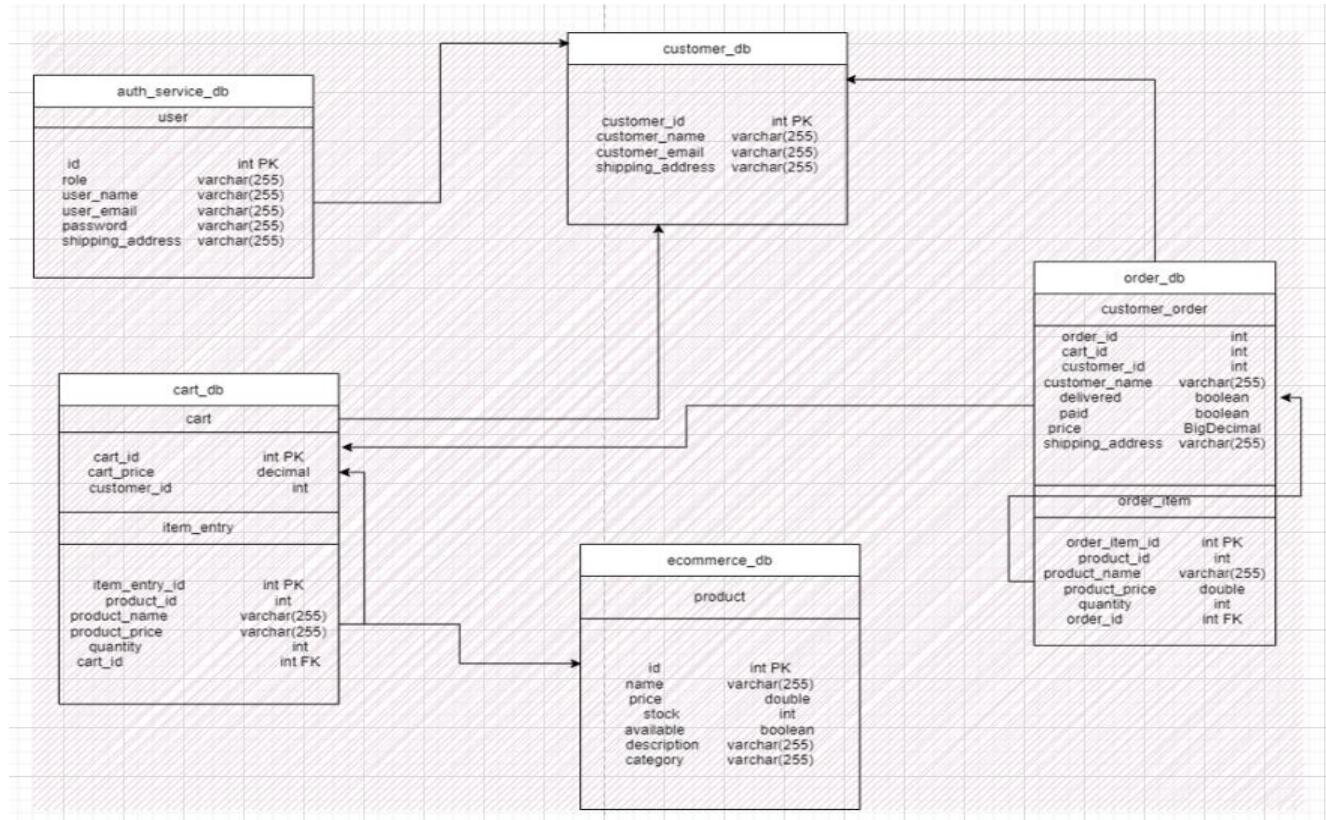
- o The customers table links to users.
- o The carts table links to users, and item_entry links to carts and products.
- o The orders table links to customer and carts, and order_items links to orders and products.

USE CASE DIAGRAMS:-

Flow Diagram:-



Database Flow Diagram:



Postman Folder Structure :-

1.

The screenshot shows the Postman interface with the following structure:

- My Workspace** (left sidebar):
 - Collections: **Auth-service** (selected), **cart-services**, **customer-service**, **order-service**, **product-service**, **REST API basics: CRUD, test & varia...**
 - Environments: None
 - History: None
- Auth-service** (main panel):
 - Overview**: Shows 11 requests: POST create user, POST token-generate, GET validate, POST create cart, PUT remove product from cart, DEL delete entire file, GET New Request, GET find by cart id, GET get customer by id, GET get all customers, PUT update customer, POST place order, GET get order by id, PUT order payment, PUT order delivery, GET view product with id, POST create product, GET getallproducts, PUT edit product.
 - Auth-service** title and description: "Auth-service" and "Make things easier for your teammates with a complete collection description."
 - Created by: You (30 Aug 2024, 1:18 AM)
 - Actions: Save, Run, Fork, Share, etc.
- Bottom navigation: Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash.

This is the Eureka Service where we show all Microservices:

Renews threshold	13		
Renews (last min)	12		
DS Replicas			
localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
AUTH-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.10:auth-service:9092
CART-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.10:cart-service:9097
CUSTOMER-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.10:customer-service:9096
E-COMMERCE-API-GATEWAY	n/a (1)	(1)	UP (1) - 192.168.1.10:E-commerce-Api-gateway:9098
ECOMMERCE-CONFIGSERVER	n/a (1)	(1)	UP (1) - 192.168.1.10:ecommerce-configserver:8888
ORDER-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.10:order-service:9094
PRODUCT-SERVICE	n/a (1)	(1)	UP (1) - 192.168.1.10:Product-service:9093
General Info			
Name	Value		

Running E-commerce-service-registry:

```

package com.wipro;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
@EnableEurekaServer
public class ECommerceServiceRegistryApplication {
    public static void main(String[] args) {
        SpringApplication.run(ECommerceServiceRegistryApplication.class, args);
    }
}

```

```

:: Spring Boot ::   (v3.3.2)

2024-09-02T09:17:51.593+05:30 INFO 16392 --- [E-commerce-service-registry] [main] c.w.ECommerceServiceRegistryApplication : Starting ECommerceServiceRegis...
2024-09-02T09:17:51.600+05:30 INFO 16392 --- [E-commerce-service-registry] [main] c.w.ECommerceServiceRegistryApplication : No active profile set, falling back...
2024-09-02T09:17:53.095+05:30 INFO 16392 --- [E-commerce-service-registry] [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=401d2eaf-7ad6-3...
2024-09-02T09:17:54.021+05:30 WARN 16392 --- [E-commerce-service-registry] [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.c...
2024-09-02T09:17:54.022+05:30 WARN 16392 --- [E-commerce-service-registry] [main] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalancerInte...
2024-09-02T09:17:54.025+05:30 INFO 16392 --- [E-commerce-service-registry] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8...
2024-09-02T09:17:54.609+05:30 INFO 16392 --- [E-commerce-service-registry] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:17:54.610+05:30 INFO 16392 --- [E-commerce-service-registry] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache...
2024-09-02T09:17:54.766+05:30 INFO 16392 --- [E-commerce-service-registry] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded We...
2024-09-02T09:17:54.768+05:30 INFO 16392 --- [E-commerce-service-registry] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: ini...
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order to avoid potential conflicts
Picked up JAVA_TOOL_OPTIONS: -agentlib:jvmhook
Picked up _JAVA_OPTIONS:
2024-09-02T09:17:56.831+05:30 INFO 16392 --- [E-commerce-service-registry] [main] c.n.d.provider.DiscoveryJerseyProvider : Using JSON encoding codec Legac...
2024-09-02T09:17:56.834+05:30 INFO 16392 --- [E-commerce-service-registry] [main] c.n.d.provider.DiscoveryJerseyProvider : Using JSON decoding codec Legac...
2024-09-02T09:17:57.263+05:30 INFO 16392 --- [E-commerce-service-registry] [main] c.n.d.provider.DiscoveryJerseyProvider : Using XML encoding codec XStre...

```

Running E-commerce-Api Gateway:

```

package com.wipro;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
public class ECommerceApiGatewayApplication {
    public static void main(String[] args) {
        SpringApplication.run(ECommerceApiGatewayApplication.class, args);
    }
}

```

```

:: Spring Boot ::   (v3.3.2)

2024-09-02T09:19:44.511+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] c.wipro.ECommerceApiGatewayApplication : Starting ECommerceApiGatewayAppli...
2024-09-02T09:19:44.514+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] c.wipro.ECommerceApiGatewayApplication : No active profile set, falling back...
2024-09-02T09:19:46.541+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=5a646885-3506-3073-8...
2024-09-02T09:19:46.667+05:30 WARN 16412 --- [E-commerce-Api-gateway] [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.c...
2024-09-02T09:19:46.673+05:30 WARN 16412 --- [E-commerce-Api-gateway] [main] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalancerIntercept...
2024-09-02T09:19:46.687+05:30 WARN 16412 --- [E-commerce-Api-gateway] [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.c...
2024-09-02T09:19:46.691+05:30 WARN 16412 --- [E-commerce-Api-gateway] [main] trationDelegate$BeanPostProcessorChecker : Bean 'reactiveDeferringLoadBalanc...
2024-09-02T09:19:48.091+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [After]
2024-09-02T09:19:48.091+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Before]
2024-09-02T09:19:48.091+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Between]
2024-09-02T09:19:48.091+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Cookie]
2024-09-02T09:19:48.091+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Header]
2024-09-02T09:19:48.093+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Host]
2024-09-02T09:19:48.093+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Method]
2024-09-02T09:19:48.093+05:30 INFO 16412 --- [E-commerce-Api-gateway] [main] o.s.c.g.r.RouteDefinitionRouteLocator : Loaded RoutePredicateFactory [Path]

```

Running Ecommerce-config-server:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "E-Commerce_Final - ecommerce-configserver". It includes modules like auth-service, cart-service, customer-service, E-commerce-Api-gateway, and ecommerce-configserver.
- Outline View:** Displays the package structure: com.wipro > EcommerceConfigserverApplication.
- Console View:** Shows the Spring Boot logs for the EcommerceConfigserverApplication. The logs indicate the application is starting up, using profile 'default', and connecting to cloud context. It also shows the discovery client registering with Eureka.

```

package com.wipro;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
@EnableConfigServer
public class EcommerceConfigserverApplication {
    public static void main(String[] args) {
        SpringApplication.run(EcommerceConfigserverApplication.class, args);
    }
}

:: Spring Boot ::      (v3.3.3)

2024-09-02T09:22:29.101+05:30 INFO 5492 --- [ecommerce-configserver] [main] c.w.EcommerceConfigserverApplication : Starting EcommerceConfigserverApplication using
2024-09-02T09:22:29.104+05:30 INFO 5492 --- [ecommerce-configserver] [main] c.w.EcommerceConfigserverApplication : No active profile set, falling back to 1 default:
2024-09-02T09:22:31.420+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=f7d8db1-282c-350e-b4cf-f7a7edfc
2024-09-02T09:22:31.537+05:30 WARN 5492 --- [ecommerce-configserver] [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.LoadBalancedClientConfiguration' of type
2024-09-02T09:22:31.544+05:30 WARN 5492 --- [ecommerce-configserver] [main] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalancerInterceptor' of type
2024-09-02T09:22:32.028+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8888 (http)
2024-09-02T09:22:32.047+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:22:32.048+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.21]
2024-09-02T09:22:32.147+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.a.c.c.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2024-09-02T09:22:32.149+05:30 INFO 5492 --- [ecommerce-configserver] [main] w.s.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed
2024-09-02T09:22:36.480+05:30 INFO 5492 --- [ecommerce-configserver] [main] DiscoveryClientOptionalArgsConfiguration : Eureka HTTP Client uses RestTemplate.
2024-09-02T09:22:36.652+05:30 WARN 5492 --- [ecommerce-configserver] [main] igration$LoadBalanceCaffeineLog : Spring Cloud LoadBalancer is currently working
2024-09-02T09:22:36.674+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 1 endpoint beneath base path '/actuator'
2024-09-02T09:22:36.772+05:30 INFO 5492 --- [ecommerce-configserver] [main] o.s.c.n.eureka.InstanceInfoFactory : Setting initial instance status as: STARTING
2024-09-02T09:22:36.823+05:30 INFO 5492 --- [ecommerce-configserver] [main] com.netflix.discovery.DiscoveryClient : Initializing Eureka in region us-east-1
2024-09-02T09:22:36.833+05:30 INFO 5492 --- [ecommerce-configserver] [main] n.d.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2024-09-02T09:22:36.852+05:30 INFO 5492 --- [ecommerce-configserver] [main] com.netflix.discovery.DiscoveryClient : Disable delta property : false
2024-09-02T09:22:36.853+05:30 INFO 5492 --- [ecommerce-configserver] [main] com.netflix.discovery.DiscoveryClient : Single vip registry refresh property : null

```

Running Auth-service:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "E-Commerce_Final - auth-service". It includes modules like auth-service, cart-service, customer-service, E-commerce-Api-gateway, ecommerce-configserver, and customerServiceApp.
- Outline View:** Displays the package structure: com.wipro > AuthServiceApplication.
- Console View:** Shows the Spring Boot logs for the AuthServiceApplication. The logs indicate the application is starting up, using Java 22.0.1, and connecting to cloud context. It also shows the discovery client registering with Eureka.

```

package com.wipro;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
@EnableDiscoveryClient
public class AuthServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(AuthServiceApplication.class, args);
    }
}

:: Spring Boot ::      (v3.3.2)

2024-09-02T09:41:33.498+05:30 INFO 20628 --- [auth-service] [restartedMain] com.wipro.AuthServiceApplication : Starting AuthServiceApplication using Java 22.0.1 with F
2024-09-02T09:41:33.505+05:30 INFO 20628 --- [auth-service] [restartedMain] com.wipro.AuthServiceApplication : No active profile set, falling back to 1 default profile:
2024-09-02T09:41:33.619+05:30 INFO 20628 --- [auth-service] [restartedMain] e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.restart.enabled=true' for additional web related logging consider setting the
2024-09-02T09:41:33.620+05:30 INFO 20628 --- [auth-service] [restartedMain] e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the
2024-09-02T09:41:35.404+05:30 INFO 20628 --- [auth-service] [restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode
2024-09-02T09:41:35.600+05:30 INFO 20628 --- [auth-service] [restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 177 ms. Four
2024-09-02T09:41:36.104+05:30 INFO 20628 --- [auth-service] [restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=320e2584-368d-32be-8a42-141ad6d0314a
2024-09-02T09:41:36.393+05:30 WARN 20628 --- [auth-service] [restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.LoadBalancedClientConfiguration' of type
2024-09-02T09:41:36.397+05:30 WARN 20628 --- [auth-service] [restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalancerInterceptor' of type [org.springfram
2024-09-02T09:41:37.093+05:30 INFO 20628 --- [auth-service] [restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 9092 (http)
2024-09-02T09:41:37.116+05:30 INFO 20628 --- [auth-service] [restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:41:37.210+05:30 INFO 20628 --- [auth-service] [restartedMain] o.a.c.c.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2024-09-02T09:41:37.212+05:30 INFO 20628 --- [auth-service] [restartedMain] w.s.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in
2024-09-02T09:41:37.537+05:30 INFO 20628 --- [auth-service] [restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-09-02T09:41:37.671+05:30 INFO 20628 --- [auth-service] [restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.5.2.Final
2024-09-02T09:41:37.750+05:30 INFO 20628 --- [auth-service] [restartedMain] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
2024-09-02T09:41:38.257+05:30 INFO 20628 --- [auth-service] [restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer

```

Running customer-service:

The screenshot shows the Eclipse IDE interface with the CustomerServiceApplication.java file open in the editor. The code defines a main method that runs the CustomerServiceApplication class. The execution log (Console tab) shows the application starting up, including the Spring Boot startup message and various configuration logs. The Project Explorer and Outline tabs are also visible.

```

package com.wipro;
import org.modelmapper.ModelMapper;
@SpringBootApplication
@EnableDiscoveryClient
public class CustomerServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(CustomerServiceApplication.class, args);
    }
}

:: Spring Boot ::      (v3.3.3)

2024-09-02T09:25:22.981+05:30 INFO 2460 --- [customer-service] [ restartedMain] com.wipro.CustomerServiceApplication : Starting CustomerServiceApplication using Java 22.0.1
2024-09-02T09:25:22.984+05:30 INFO 2460 --- [customer-service] [ restartedMain] com.wipro.CustomerServiceApplication : No active profile set, falling back to 1 default profile: 'default'
2024-09-02T09:25:23.050+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Fetching config from server at : http://localhost:8811
2024-09-02T09:25:23.050+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Located environment: name=customer-service, profiles=default
2024-09-02T09:25:23.053+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.restart.enabled' to false to disable
2024-09-02T09:25:23.053+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode
2024-09-02T09:25:23.053+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 163 ms. Found 0 JPA repository interfaces
2024-09-02T09:25:23.055+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=fdb6eb8-693d-3bb5-9f10-178ac2bea1eb
2024-09-02T09:25:23.055+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=fdb6eb8-693d-3bb5-9f10-178ac2bea1eb
2024-09-02T09:25:23.055+05:30 WARN 2460 --- [customer-service] [ restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.LoadBalancerInterceptor' of type [org.springframework.cloud.client.loadbalancer.LoadBalancerInterceptor] is not managed by the current BeanFactory. Please check if you have defined a BeanFactoryAware bean
2024-09-02T09:25:23.055+05:30 WARN 2460 --- [customer-service] [ restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalanceInterceptor' of type [org.springframework.cloud.client.loadbalancer.deferring.DeferringLoadBalanceInterceptor] is not managed by the current BeanFactory. Please check if you have defined a BeanFactoryAware bean
2024-09-02T09:25:23.055+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:25:23.055+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.28]
2024-09-02T09:25:23.055+05:30 INFO 2460 --- [customer-service] [ restartedMain] o.apache.catalina.core.StandardEngine : Initializing Spring embedded WebApplicationContext: Root WebApplicationContext: initialization completed
2024-09-02T09:25:27.355+05:30 INFO 2460 --- [customer-service] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed

```

Running Product-service:

The screenshot shows the Eclipse IDE interface with the ProductServiceApplication.java file open in the editor. The code defines a main method that runs the ProductServiceApplication class. The execution log (Console tab) shows the application starting up, including the Spring Boot startup message and various configuration logs. The Project Explorer and Outline tabs are also visible.

```

package com.wipro;
import org.modelmapper.ModelMapper;
@SpringBootApplication
@EnableDiscoveryClient
public class ProductServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(ProductServiceApplication.class, args);
    }
}

:: Spring Boot ::      (v3.3.3)

2024-09-02T09:43:33.734+05:30 INFO 10392 --- [product-service] [ restartedMain] com.wipro.ProductServiceApplication : Starting ProductServiceApplication using Java 22.0.1
2024-09-02T09:43:33.739+05:30 INFO 10392 --- [product-service] [ restartedMain] com.wipro.ProductServiceApplication : No active profile set, falling back to 1 default profile: 'default'
2024-09-02T09:43:33.831+05:30 INFO 10392 --- [product-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Fetching config from server at : http://localhost:8811
2024-09-02T09:43:33.831+05:30 INFO 10392 --- [product-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Located environment: name=Product-service, profiles=default
2024-09-02T09:43:33.835+05:30 INFO 10392 --- [product-service] [ restartedMain] o.e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.restart.enabled' to false to disable
2024-09-02T09:43:33.835+05:30 INFO 10392 --- [product-service] [ restartedMain] o.e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting 'spring.devtools.log.level' to 'DEBUG'
2024-09-02T09:43:35.479+05:30 INFO 10392 --- [product-service] [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode
2024-09-02T09:43:35.654+05:30 INFO 10392 --- [product-service] [ restartedMain] o.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 163 ms. Found 0 JPA repository interfaces
2024-09-02T09:43:36.172+05:30 INFO 10392 --- [product-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=40820f33-ce6a-3dc4-a809-982773a31afe
2024-09-02T09:43:36.444+05:30 WARN 10392 --- [product-service] [ restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.LoadBalancerInterceptor' of type [org.springframework.cloud.client.loadbalancer.LoadBalancerInterceptor] is not managed by the current BeanFactory. Please check if you have defined a BeanFactoryAware bean
2024-09-02T09:43:36.449+05:30 WARN 10392 --- [product-service] [ restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'deferringLoadBalanceInterceptor' of type [org.springframework.cloud.client.loadbalancer.deferring.DeferringLoadBalanceInterceptor] is not managed by the current BeanFactory. Please check if you have defined a BeanFactoryAware bean
2024-09-02T09:43:37.139+05:30 INFO 10392 --- [product-service] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:43:37.168+05:30 INFO 10392 --- [product-service] [ restartedMain] o.apache.catalina.core.StandardService : Starting Servlet engine: [Apache Tomcat/10.1.28]
2024-09-02T09:43:37.169+05:30 INFO 10392 --- [product-service] [ restartedMain] o.apache.catalina.core.StandardEngine : Initializing Spring embedded WebApplicationContext: Root WebApplicationContext: initialization completed
2024-09-02T09:43:37.278+05:30 INFO 10392 --- [product-service] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[] : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-09-02T09:43:37.623+05:30 INFO 10392 --- [product-service] [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000041: Hibernate ORM core version 6.5.2.Final
2024-09-02T09:43:37.732+05:30 INFO 10392 --- [product-service] [ restartedMain] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
2024-09-02T09:43:37.810+05:30 INFO 10392 --- [product-service] [ restartedMain] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled

```

Running Cart-service:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "E-Commerce_Final - cart-service". It includes modules like com.wipro.service, src/main/resources, src/test/java, JRE System Library [JavaSE-17], Maven Dependencies, src, target, and Product-service.
- Code Editor:** Displays the `CartServiceApplication.java` file. The code is as follows:

```

1 package com.wipro;
2
3 import org.modelmapper.ModelMapper;
4
5 @SpringBootApplication
6 @EnableDiscoveryClient
7 @EnableFeignClients(defaultConfiguration = FeignClientConfig.class)
8 public class CartServiceApplication {
9
10    public static void main(String[] args) {
11        SpringApplication.run(CartServiceApplication.class, args);
12    }
13}

```

- Terminal:** Shows the command-line output of the application's startup. Key logs include:

```

2024-09-02T09:44:25.782+05:30 INFO 21464 --- [cart-service] [ restartedMain] com.wipro.CartServiceApplication : Starting CartServiceApplication using Java 22.0.1 with F
2024-09-02T09:44:25.786+05:30 INFO 21464 --- [cart-service] [ restartedMain] com.wipro.CartServiceApplication : No active profile set, falling back to 1 default profile
2024-09-02T09:44:25.887+05:30 INFO 21464 --- [cart-service] [ restartedMain] ConfigServerConfigDataLoader : Fetching config from server at : http://localhost:8888
2024-09-02T09:44:25.894+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Located environment: name=cart-service, profiles=[defau
2024-09-02T09:44:25.894+05:30 INFO 21464 --- [cart-service] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools
2024-09-02T09:44:25.894+05:30 INFO 21464 --- [cart-service] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the
2024-09-02T09:44:25.894+05:30 INFO 21464 --- [cart-service] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT m
2024-09-02T09:44:25.894+05:30 INFO 21464 --- [cart-service] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 173 ms. Four
2024-09-02T09:44:26.143+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=8f9348d2-0022-3436-a6a7-7de9b1ee08c6
2024-09-02T09:44:26.659+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=8f9348d2-0022-3436-a6a7-7de9b1ee08c6
2024-09-02T09:44:29.235+05:30 WARN 21464 --- [cart-service] [ restartedMain] o.s.b.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8087 (http)
2024-09-02T09:44:29.234+05:30 WARN 21464 --- [cart-service] [ restartedMain] o.s.b.embedded.tomcat.TomcatWebServer : Starting service [Tomcat]
2024-09-02T09:44:29.973+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.apache.catalina.core.StandardService : Starting Service engine: [Apache Tomcat/10.1.28]
2024-09-02T09:44:30.059+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.a.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-09-02T09:44:30.063+05:30 INFO 21464 --- [cart-service] [ restartedMain] w.s.c.WebServletServerApplication : Root WebApplicationContext: initialization completed in
2024-09-02T09:44:30.988+05:30 INFO 21464 --- [cart-service] [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default
2024-09-02T09:44:31.158+05:30 INFO 21464 --- [cart-service] [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.5.2.Final

```

Running Order-service:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "E-Commerce_Final - order-service". It includes modules like com.wipro.service, src/main/resources, src/test/java, JRE System Library [JavaSE-17], Maven Dependencies, src, target, and Product-service.
- Code Editor:** Displays the `OrderServiceApplication.java` file. The code is as follows:

```

1 package com.wipro;
2
3 import org.modelmapper.ModelMapper;
4
5 @SpringBootApplication
6 @EnableDiscoveryClient
7 @EnableFeignClients(defaultConfiguration = FeignClientConfig.class)
8 public class OrderServiceApplication {
9
10    public static void main(String[] args) {
11        SpringApplication.run(OrderServiceApplication.class, args);
12    }
13}

```

- Terminal:** Shows the command-line output of the application's startup. Key logs include:

```

2024-09-02T09:46:34.078+05:30 INFO 17036 --- [order-service] [ restartedMain] com.wipro.OrderServiceApplication : Starting OrderServiceApplication using Java 22.0.1 with
2024-09-02T09:46:34.081+05:30 INFO 17036 --- [order-service] [ restartedMain] com.wipro.OrderServiceApplication : No active profile set, falling back to 1 default profile
2024-09-02T09:46:34.149+05:30 INFO 17036 --- [order-service] [ restartedMain] ConfigServerConfigDataLoader : Fetching config from server at : http://localhost:8888
2024-09-02T09:46:34.149+05:30 INFO 17036 --- [order-service] [ restartedMain] o.s.c.c.ConfigServerConfigDataLoader : Located environment: name=order-service, profiles=[defau
2024-09-02T09:46:34.152+05:30 INFO 17036 --- [order-service] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools
2024-09-02T09:46:34.152+05:30 INFO 17036 --- [order-service] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the
2024-09-02T09:46:36.011+05:30 INFO 17036 --- [order-service] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT r
2024-09-02T09:46:36.206+05:30 INFO 17036 --- [order-service] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 181 ms. For
2024-09-02T09:46:36.376+05:30 INFO 17036 --- [order-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=e9b693a9-e33-36b8-ba05-4cb199b7fe6
2024-09-02T09:46:37.225+05:30 WARN 17036 --- [order-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=e9b693a9-e33-36b8-ba05-4cb199b7fe6
2024-09-02T09:46:37.231+05:30 WARN 17036 --- [order-service] [ restartedMain] o.s.cloud.context.scope.GenericScope : BeanFactory id=e9b693a9-e33-36b8-ba05-4cb199b7fe6
2024-09-02T09:46:37.389+05:30 INFO 17036 --- [order-service] [ restartedMain] o.s.b.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 9094 (http)
2024-09-02T09:46:38.014+05:30 INFO 17036 --- [order-service] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-02T09:46:38.015+05:30 INFO 17036 --- [order-service] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.28]
2024-09-02T09:46:38.127+05:30 INFO 17036 --- [order-service] [ restartedMain] o.a.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-09-02T09:46:38.130+05:30 INFO 17036 --- [order-service] [ restartedMain] w.s.c.WebServletServerApplication : Root WebApplicationContext: initialization completed in
2024-09-02T09:46:38.929+05:30 INFO 17036 --- [order-service] [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: defau
2024-09-02T09:46:39.084+05:30 INFO 17036 --- [order-service] [ restartedMain] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.5.2.Final
2024-09-02T09:46:39.159+05:30 INFO 17036 --- [order-service] [ restartedMain] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled

```

Registration User

The screenshot shows a Postman request to `http://localhost:9098/auth-service/api/auth/register`. The request method is `POST`. The body contains the following JSON:

```
1 "userName": "zeba",
2 "userEmail": "zeba@1907.com",
3 "password": "zeba@1907",
4 "role": "user",
5 "shippingAddress": "rajiv chowk,delhi"
```

The response status is `200 OK` with a response time of 934 ms and a size of 562 B. The response body is:

```
1 {
2   "id": 202,
3   "userName": "zeba",
4   "userEmail": "zeba@1907.com",
5   "password": "$2a$10$zAtlCTZYnC0QXKmcfIE60uk.k0RZGWQD0lFktsrtp0t.b9/clw40m",
6   "role": "ROLE_USER",
7   "shippingAddress": "rajiv chowk,delhi"
8 }
```

Token generation

The screenshot shows a Postman request to `http://localhost:9098/auth-service/api/auth/token`. The request method is `POST`. The body contains the following JSON:

```
1 {
2   "userEmail": "zeba@1907.com",
3   "password": "zeba@1907"
4 }
```

The response status is `200 OK` with a response time of 363 ms and a size of 517 B. The response body is:

```
1 eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ6ZWJhQDE5MDcuY29tIiwiaWF0IjoxNzI1MjE4NzQ1LCJleHAiOjE3MjUyMjA1NDV9.JG00ZxvRIOrOf44gpxnQBM9JS5am2hfZBol38q07Rgo
```

Token validation

The screenshot shows the Postman interface for validating a token. The URL is `http://localhost:9098/auth-service/api/auth/validate?token=eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJ6ZWJhQDE5MDcuY29tliwiaWF0ijoxNzI1MjE4NzQ1L...`. The response status is 200 OK, with a duration of 21 ms and a size of 389 B. The response body contains the message "Token is valid".

Product microservice:-

The screenshot shows the product-controller API endpoints. The endpoints listed are:

- GET /api/product/{id}
- PUT /api/product/{id}
- DELETE /api/product/{id}
- POST /api/product
- GET /api/getproductsbycategory/query
- GET /api/getproductsbyavailability/query
- GET /api/getallproducts

Below the endpoints, there is a section for Schemas, which includes a ProductDto schema.

Customer microservice:-

Servers
http://localhost:9096 - Generated server url ▾

customer-controller

- GET /api/customer/{customerId} ▾
- PUT /api/customer/{customerId} ▾
- GET /api/customer ▾
- POST /api/customer ▾

Schemas

- CustomerDto >

Cart microservice:-

cart-controller

- GET /api/cart/{customerId} ▾
- PUT /api/cart/{customerId} ▾
- POST /api/cart/{customerId} ▾
- GET /api/getcartid/{cartId} ▾
- DELETE /api/cart/{cartId} ▾

Schemas

- CartDto >
- ItemEntryDto >

Order microservice:-

order-controller	
PUT	/api/updateorderpayment/{orderId}
PUT	/api/updateorderdelivery/{orderId}
POST	/api/placeorder/{cartId}
GET	/api/order/{orderId}
GET	/api/getorderbypayment/query
GET	/api/getorderbydelivery/query
GET	/api/getorderbycustomer/{customerId}
GET	/api/getallOrders
DELETE	/api/cancelorder/{orderId}

THANK
YOU...