```
======================
eCommerce Application
======================

Project Architecture : https://www.youtube.com/watch?v=IbOMX4OYjSM

Backend : SpringBoot + Data JPA + Data Rest + MySQL DB

Frontend : Angular 17v

====================
Application Modules
====================

1) Products & Categories

2) Cart

3) Checkout

4) Payment

5) User Management

6) Orders

========================================================================================
01-Assignment : DB Design
========================================================================================

=================
Database Design
=================

1) product_category

                category_id             PK
                category_name

2) product

                product_id              PK
                name
                desc
                title
                unit_price
                image_url
                active
                units_stock
                date_created
                last_updated
                category_id             FK


3)      Customer

                customer_id  PK
                name
                email
                pwd
                phno

4) shipping_address
```

```
                addr_id         PK
                house_num
                street
                city
                state
                zipcode
                country

4) Order

                order_id                        PK
                order_tracking_num
                total_quantity
                total_price
                order_status
                date_created
                last_updated
                razor_pay_payment_id

                customer_id             FK
                addr_id                 FK


5) Order_items

                order_item_id   PK
                image_url
                unit_price
                quantity
                product_id      FK
                order_id        FK

==================================================================================
02-Assignment : Create Project + Create Entities with Association Mapping + Repositories
==================================================================================


===============================================================
Step-1) Create Spring Boot Application with below starters
===============================================================

a) spring-boot-starter-data-jpa

b) spring-boot-starter-data-rest

c) spring-boot-devtools

d) mysql-connector-j

==================================
Step-2) Create Entity Classes
==================================

@Entity
@Table(name="product_category")
public class ProductCategory {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String categoryName;

    @OneToMany(mappedBy = "category", cascade = CascadeType.ALL)
    private Set<Product> products;

}

@Entity
@Table(name="product")
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    private String description;

    private String title;

    private BigDecimal unitPrice;

    private String imageUrl;

    private boolean active;

    private int unitsInStock;

    private Date dateCreated;

    private Date lastUpdate;

    @ManyToOne
    @JoinColumn(name = "category_id", nullable = false)
    private ProductCategory category;

}
```

```
=======================================
Step-3) Create Repository Interfaces
======================================
```

```
@RepositoryRestResource(path = "products")
public interface ProductRepository extends JpaRepository<Product, Long> {
}

@RepositoryRestResource(path = "product-category")
public interface ProductCategoryRepository extends JpaRepository<ProductCategory, Long> {
}
```

```
==========================================================
Step-4) Configure data source properties in props file
=========================================================
```

```
spring.application.name=ecommerce_backend


spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/ashokit_ecomm
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true
spring.data.rest.base-path=/api

==========================================================
Step-5) Run the application and test tables creation
=========================================================


==========================================================
Step-6) Insert data into db tables using sql queries
==========================================================

1) insert data into category table
2) insert data into product table (for each category at least 5 products)

=========================================================
Step-7) Test api functionality using postman
=========================================================

Note: If we observe the response, the primary columns data will not come in response.


=================================
Step-8 : Expose Ids for entities
==================================

@Configuration
public class MyDataRestConfig implements RepositoryRestConfigurer {

    @Autowired
    private EntityManager entityManager;

    @Override
    public void configureRepositoryRestConfiguration(RepositoryRestConfiguration config, CorsRegistry
cors) {
        exposeIds(config);
    }

    private void exposeIds(RepositoryRestConfiguration config) {

        // get all entity classes from the entity manager
        Set<EntityType<?>> entities = entityManager.getMetamodel().getEntities();

        // create an array for the entity types
        List<Class> entityClasses = new ArrayList<>();

        // get entity types for entities
        for (EntityType tempEntityType : entities) {
            entityClasses.add(tempEntityType.getJavaType());
        }

        // expose ids for domain types
        Class[] domainTypes = entityClasses.toArray(new Class[0]);
        config.exposeIdsFor(domainTypes);
    }
}

=======================================================
Step-9 : Write Custom Methods in ProductRepository
======================================================

@RepositoryRestResource(path = "products")
public interface ProductRepository extends JpaRepository<Product, Long> {

    Page<Product> findByCategoryId(@Param("id") Long id, Pageable pageable);
    Page<Product> findByNameContaining(@Param("name") String name, Pageable pageable);
}
```

Note: For the above 2 findBy methods implementation will be provided by data jpa in runtime.

```
==========================================================
Step-10 : Test backend api functionality with below URLS
==========================================================

URL-1 : http://localhost:8080/api/products

URL-2 : http://localhost:8080/api/products/1

URL-3 : http://localhost:8080/api/product-category

URL-4 : http://localhost:8080/api/product-category/1

URL-5 : http://localhost:8080/api/product-category/1/products

URL-6 : http://localhost:8080/api/products/search/findByCategoryId?id={1}

URL-7 : http://localhost:8080/api/products/search/findByNameContaining?name={apple}


==============
DB Queries
==============

INSERT INTO product_category(category_name) VALUES ('Laptops');
INSERT INTO product_category(category_name) VALUES ('Mobiles');
INSERT INTO product_category(category_name) VALUES ('Shirts');
INSERT INTO product_category(category_name) VALUES ('Beauty');


--------------------------------------------------------------------------------
INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('DELL-LAPTOP-1000', 'DELL - Laptop', 'Processor: Intel Core i5-1235U 12th Generation (up to
4.40 GHz, 12MB 10 Cores)',
'assets/images/products/laptops/dell-laptop-1000.png',1,100,19.99,1, NOW());

INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('HP-LAPTOP-1001', 'HP - Laptop', 'Processor: Intel Core i5-1235U 12th Generation (up to 4.40
GHz, 12MB 10 Cores)',
'assets/images/products/laptops/hp-laptop-1001.png',1,100,59.99,1, NOW());

INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('ACER-LAPTOP-1002', 'ACER - Laptop', 'Acer Aspire Lite 12th Gen Intel Core i5-1235U Thin and
Light Laptop ',
'assets/images/products/laptops/acer-laptop-1002.png',1,100,49.99,1, NOW());

INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('LENOVO-LAPTOP-1003', 'Lenovo ThinkPad E14', 'Lenovo ThinkPad E14 AMD Ryzen 5',
'assets/images/products/laptops/lenovo-laptop-1003.png',1,100,45.00,1, NOW());

---------------------------------------------------------------------------------------------------

INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('Apple-IPhone-1000', 'Apple-Iphone', 'Apple iPhone 13 (128GB) - Green',
'assets/images/products/mobiles/apple-mobile-1000.png',1,100,59,2, NOW());
```

```
INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('RED-MI-1001', 'Redmi 13C', 'Redmi 13C (Stardust Black, 6GB RAM, 128GB Storage)',
'assets/images/products/mobiles/redmi-mobile-1001.png',1,100,29,2, NOW());

INSERT INTO ecomm.product (title, name, description, image_url, active, units_in_stock,unit_price,
category_id, date_created)
VALUES ('SAMSUNG-Galaxy-1002', 'Samsung Mobile', 'SAMSUNG Galaxy F14 5G 6GB RAM 128GB STORAGE',
'assets/images/products/mobiles/samsung-mobile-1002.png',1,100,39,2, NOW());

INSERT INTO ecomm.product (title, name, description, image_url, active, units_in_stock,unit_price,
category_id, date_created)
VALUES ('POCO C65', 'POCO C65', 'POCO C65 (Pastel Green 4GB RAM 128GB Storage)',
'assets/images/products/mobiles/poco-mobile-1003.png',1,100,49,2, NOW());

----------------------------------------------------------------------------------------------------
-------

INSERT INTO product (title, name, description, image_url, active, units_in_stock,
unit_price, category_id, date_created)
VALUES ('Mens Cotton Shirt', 'Mens Cotton Shirt', 'Amazon Brand - Symbol Mens Cotton
Shirt','assets/images/products/clothes/shirt-1000.png',1,100,5,3, NOW());

----------------------------------------------------------------------------------------------------
----------
```