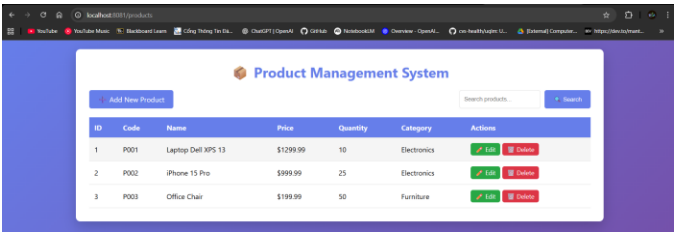
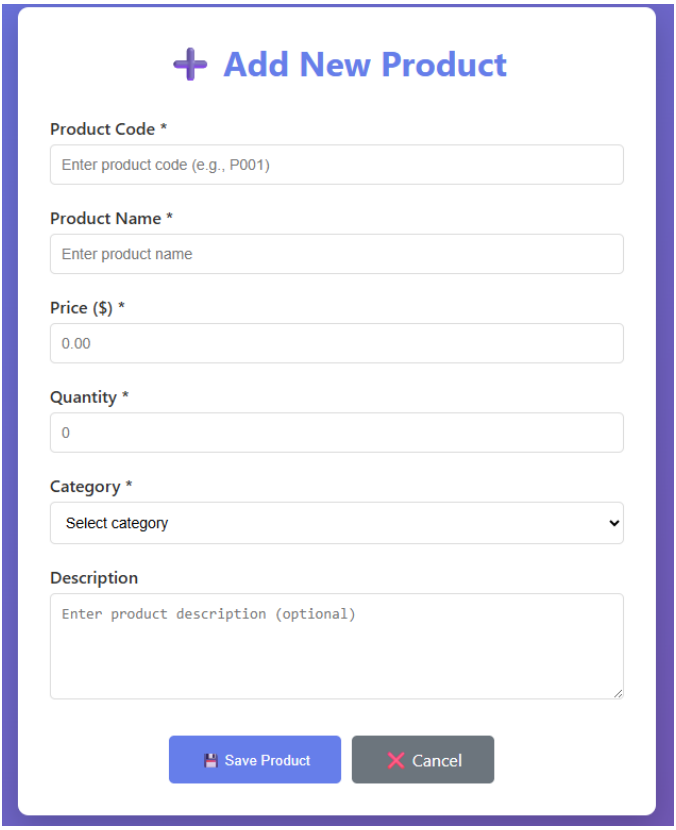
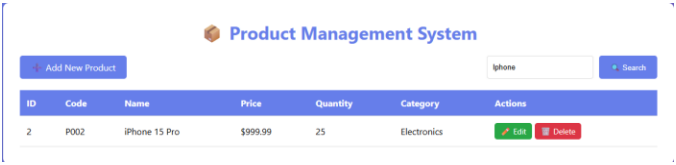
Exercise Lab8 ITCSIU22266







Entity:

Maps Java class to database table.

```
public Product(String productCode, String name, BigDecimal price, Integer quantity, String category, String description) {
    this.productCode = productCode;
    this.name = name;
    this.price = price;
    this.quantity = quantity;
    this.category = category;
    this.description = description;
}
```

Repository:

```
@Repository
public interface ProductRepository extends JpaRepository<Product, Long> {

    // Spring Data JPA generates implementation automatically!

    // Custom query methods (derived from method names)
    List<Product> findByCategory(String category);

    List<Product> findByNameContaining(String keyword);

    List<Product> findByPriceBetween(BigDecimal minPrice, BigDecimal maxPrice);

    List<Product> findByCategoryOrderByPriceAsc(String category);

    boolean existsByProductCode(String productCode);

    // All basic CRUD methods inherited from JpaRepository:
    // - findAll()
    // - findById(Long id)
    // - save(Product product)
    // - deleteById(Long id)
    // - count()
    // - existsById(Long id)
}
```

Extends JpaRepository<Product, Long> -
gets free CRUD methods:

- findAll(), findById(), save(), deleteById()

Custom queries derived from method names (Spring Data JPA magic):

- findByCategory() -> SELECT * FROM products WHERE category = ?
- findByNameContaining() -> SELECT * FROM products WHERE name LIKE %?%

Service:

```java
public interface ProductService {

    List<Product> getAllProducts();

    Optional<Product> getProductById(Long id);

    Product saveProduct(Product product);

    void deleteProduct(Long id);

    List<Product> searchProducts(String keyword);

    List<Product> getProductsByCategory(String category);
}
```

Business logic layer between controller and repository

DB connection:

```
# 1. Database Configuration (MSSQL Changes)
# Use the SQL Server JDBC URL format
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=product_management;encrypt=true;trustServerCertificate=true
# Replace 'your_username' and 'your_password' with your MSSQL credentials
spring.datasource.username=sa
spring.datasource.password=sa
# Use the MSSQL JDBC Driver
spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver
```