Sprint 1

Day 1:-

Created the maven webapp front end

Added spring mvc dependencies.

Create the MVC structure-->com.ecomm.controller, web-inf/views/index.jsp [bootstrap 4]

Created web.xml

Created dispatcher-servlet.xml

Day 2 :-

Created the maven quick start back end project

Added dependencies in back end pom.xml

{spring-core, spring-web, spring-context, hibernate-core, H2-jar etc}.Also added the backend POM jar to frontend POM.

Created com.ecomm.model, com.ecomm.config ,com.ecomm.dao, com.ecomm.dao.impl packages.

Day 3:-

Configured DBConfig.java

a) createdautowired beans for database connectivity

b) createdautowired beans for hibernate dialect and syntax

c) createdautowired beans for hibernate transaction management

Sprint 2

Day 1:-

Create the userdetail ,cart ,supplier,orderdetail,product,category model class,

Create the table in H2

Create CRUD method in DAO--> Defined it in DAOimpl

Day 2:-

Added the autowiredDaoImpl bean in hiberConfig.java

Created the register web-flow to direct user to fill up forms for address as per the attributed in User,Address class using Spring form

Day 3:-

Created the controller methods to save the registration data by taking it from spring form and sending it as user object parameter to CRUD method of DAOimpl.

Created the supplier and category tables in H2.

Created Supplier and Category Dao, DaoImpl.

Included Supplier and Category DaoImpl @Autowired bean in the DBConfig.java

Sprint 3

Day 1:-

Created Product.jsp with supplier and category tab and forms.

Created controller methods for Supplier and Category CRUD.

Performed CRUD operations for Supplier and Category.

insert()

update()

delete()

List<Supplier> listItems();

Day 2:-

Created Product.java (MultipartFile-->Transient), ProductDao.java, ProductDaoImpl.java

insert()

List<Supplier> listItems();

Day 3:-

Add product form along with multipart file upload option.

Create the controller methods for calling all the crud operations.

Sprint 4

Day 1:-

Create the Login.jsp

Create the login option in Header.jsp on top navbar

Write the controller method to go to the Login.jsp(String+@RequestMapping)

Day 2:-

Implement spring security in xml configuration file to intercept every user request and check the authentication and authorization of user.

Add spring security dependencies in pom.xml

Add the filterChainProxy class in web.xml

Add the filter-mapping of security class in web.xml

Day 3:-

Modify all CRUD operation URLs and prefix "/login" to each of the URLs

Add the loggedIn() and error() in Login()

Display the username/email/id on the top navbar once any user logs in

Sprint 5

Day 1:-

Created Cart.java and OrderDetail.java

Created CartDao.java and OrderDetailDao.java

Day 2:-

Created CartDaoImpl.java and CartDaoImpl.java

Added the annotated Cart and Order model class in DBConfig.java

Day 3:-

Added the Cart.jsp

Created checkout flow to maintain the logical flow of form where user could select at which address product should be delivered.

Sprint 6

Day 1:-

Added cart controller for /TotalProductDisplay,/AddToCart, /checkout,/reciept, /deleteProduct,/Cart/payment

Added cart in Header.jsp

Day 2:-

Add hibernate/java validation dependencies in your pom.xml and add the validation annotation in your model classes of Category/Product/Supplier/UserDetail against every relevant attribute.

Day 3:-

Create css, and js on JSP pages for all forms to show error messages if invalid data in entered.