DEVELOPER NAME: - DIPAK SINH
PROJECT NAME: - SPORTYSHOES.COM
E-MAIL ID: - SINGHDK.SINGH68@GMAIL.COM

Overview

The project name SportyShoes.com was divided into 2 Sprints of 1 week each. And for each Sprint what we have done is shown below.

Sprint

1st Sprint: -

1. **Initialized the project, made it a Spring boot starter project and added dependencies: MySQL Connector, and Hibernate libraries.**
2. **Setting MySQL database and entities relationship.**
3. **Configured hibernate and mapped classes as entities.**
4. **Planned the structure of the website and navigation across it.**
5. **Made the controller files, repositories, interfaces and other related java files.**
6. **Made a Header and footer HTML pages to include it across every page.**

2nd Sprint: -

1. **Made the Login and User Registration page.**
2. **Admin Login: This is authorised according to data in the database for admin only.**
3. **Manage Products: Add, Delete, Update Products.**
4. **Admin can view user list as to how many users had been registered on the website.**
5. **User can view the order details on the order page.**
6. **Modified the navigation with the help of a link and added the navigation to the home page in the Header Tag.**
7. **Tested the code and fixed identified bugs.**

Operations available in the project

1. **Login Validation**
2. **New User Registration**
3. **Adding new Product**
4. **Updating old Product**
5. **Deleting old Product**
6. **Listing products on the admin page as well as the main website.**
7. **Validation of roles**
8. **Logout validation**

➢ Core java Module

- **@Controller: to use the classes as a controller class.**
- **@Service: To indicate class as Service.**
- **@Repository: To indicate classes/interfaces as a Repository to contact the Database.**
- **@Entity: To indicate classes as the table in the Database.**
- **@Autowired: to auto-connect between Spring Beans, Services and Repositories.**
- **@PostMapping: to indicate URL links with the Servlet post method.**
- **@GetMapping: to indicate URL links with the Get method in servlet.**
- **@RequestParam and @RequestBody: Get values from the webpage.**
- **javax.servlet.http.HttpSession: To manage Sessions with HTTP protocol.**
- **org.springframework.ui.Model: to send data to view.**
- **java.sql.SQLException: To manage Database exceptions**
- **JpaRepository/crudRepository: to get methods for CRUD operations.**
- **jpa.repository.Query (@Query): To write native queries for custom methods for CRUD operations.**
- **ThymeLeaf Template tags in HTML.**
- **@SpringBootApplication: To initialize spring boot.**