**PE\_HSF302\_FA25\_TrialTest\_Note**

FALL 2025  
Subject: HSF302  
Duration: 90 minutes

1. **­­­­Working with DB connection string from application.properties**

spring.application.name=pe-hsf302-studentcode  
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName= LaptopShopDB2025;encrypt=true;trustServerCertificate=true;  
spring.datasource.username=sa  
spring.datasource.password=1234  
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver

spring.jpa.hibernate.ddl-auto=create

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

spring.thymeleaf.cache=false  
spring.thymeleaf.prefix=classpath:/templates/  
spring.thymeleaf.suffix=.html

1. **Working with the pom.xml**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>mssql-jdbc</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

1. **Config on SpringBootApplication project**

@SpringBootApplication  
@ComponentScan({"hsf302.se.fu.vn.controller","hsf302.se.fu.vn.service"})  
@EnableJpaRepositories(basePackages = "hsf302.se.fu.vn.repository")  
@EntityScan(basePackages = "hsf302.se.fu.vn")  
public class DemoLab01Application {  
 public static void main(String[] args) {  
 SpringApplication.run(DemoLab01Application.class, args);

}  
}

1. **Table**

* **users (User Authentication Table)**

|  |  |
| --- | --- |
| **user\_id** | INT **IDENTITY(1,1)** PRIMARY KEY |
| **email** | VARCHAR(100) UNIQUE NOT NULL |
| **password** | VARCHAR(50) NOT NULL |
| **role** | VARCHAR(20) NOT NULL (role IN (‘Admin’, 'Staff', 'Member')) |

* **manufacturers (Computer Manufacturers Table)**

|  |  |
| --- | --- |
| **manufacturer\_id** | INT **IDENTITY(1,1)** PRIMARY KEY |
| **manufacturer\_name** | VARCHAR(100) NOT NULL UNIQUE |
| **country** | VARCHAR(100) |

* **computers (Main Products Table)**

|  |  |
| --- | --- |
| **computer\_id** | INT **IDENTITY(1,1)** PRIMARY KEY |
| **computer\_model** | VARCHAR(100) NOT NULL |
| **type** | VARCHAR(50) NOT NULL |
| **production\_year** | INT (production\_year BETWEEN 1990 AND YEAR(GETDATE())) |
| **price** | DECIMAL(10,2) (price >= 0) |
| **manufacturer\_id** | INT NOT NULL FOREIGN KEY (manufacturer\_id) REFERENCES manufacturers(manufacturer\_id) |

1. **Data**

* **users**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Id** | **Email** | **Password** | **Role** |
| 1 | admin@laptopshop.com | @1 | Admin |
| 2 | staff@laptopshop.com | @2 | Staff |
| 3 | member@laptopshop.com | @3 | Member |

* **manufacturers**

|  |  |  |
| --- | --- | --- |
| **manufacturer\_id** | **manufacturer\_name** | **country** |
| 1 | Dell | USA |
| 2 | Lenovo | China |
| 3 | HP | USA |

* **computers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **computer\_id** | **computer\_model** | **type** | **production\_year** | **price** | **manufacturer\_id** |
| 1 | XPS 13 | Ultrabook | 2023 | 1299.99 | 1 |
| 2 | ThinkPad X1 Carbon | Business Laptop | 2023 | 1499.99 | 2 |
| 3 | Pavilion 15 | Consumer Laptop | 2022 | 699.99 | 3 |
| 4 | Inspiron 14 | Budget Laptop | 2023 | 549.99 | 1 |

**6. Thymeleaf**

<html xmlns:th="http://www.thymeleaf.org">

<link th:href="@{hsf302.css}" type="text/css" rel="stylesheet">

<tr th:each="user : ${users}">  
 <td th:text="${user.username}"></td>  
</tr>

<div class="error-message" th:if="${error}" th:text="${error}">

<div class="form-group">  
 <label for="username">Username:</label>  
 <input type="text" name="username" id="username" th:field="\*{username}">  
 <span class="error-message" th:if="${#fields.hasErrors('username')}"  
 th:errors="\*{username}"></span>  
</div>

<form method="post" th:action="@{/product/create}" th:object="${product}">  
 <table>

<tr>  
 <td>Product Name:</td>  
 <td>

<input type="text" id="name" name="name" th:field="\*{name}"/>

</td>

</tr>

</table>

</form>

<a th:href="@{list}">Back to List</a>

**NOTE : This session just for reference, student can use the other approach to do Practical Exam and student must change to run your program**