

2025년도 SW프레임워크 중간고사

※ 학번과 이름을 반드시 기입해주세요.

학번 :

이름 :

✓ 아래 사항을 동의해주세요.

시험 부정행위에 대한 동의서

본인은 이 시험을 치르는 동안 모든 규칙과 지침을 준수할 것을 동의합니다. 부정행위가 적발될 경우, 본인이 치른 해당 시험은 0점으로 처리됨을 이해하고 동의합니다. 부정행위에는 다음과 같은 행위가 포함됩니다:

- 시험지 또는 답안을 다른 수험생과 공유
- 금지된 자료(웹 서핑, AI 등등)나 기기의 사용
- 시험 중 통신 기능 사용
- 대리 시험 응시
- 단, 1장 분량의 학생 개인이 준비한 강의자료는 확인 가능

본인은 이러한 규정을 위반할 경우 시험 성적 무효와 함께 학적상의 불이익을 받을 수 있음을 인지하고 있습니다. 이에 본인은 위 내용에 대해 충분히 이해하고 동의하는 바,

아래에 서명합니다.

날짜 : _____

학번 : _____

이름 : _____ (인)

※ 시험은 다음과 같은 방식으로 진행됩니다.

시험 진행 방식

- 시험은 100% 실습으로 진행됩니다.
- 시험 시간 20분 전에 자신이 시험에 사용할 컴퓨터 점검을 마쳤다는 가정하에 진행됩니다. 따라서, 시험 과정에서 본인이 사용하는 컴퓨터의 이상으로 인해 발생할 수 있는 상황에 대해서는 스스로 해결해야 합니다.
- 시험 시간에 사용하는 이미 개발 환경이 설정된 개인 PC, 노트북 등을 이용하나 개발 중 문제가 발생한 오류 부분은 학생 본인에게 책임이 있습니다.

- 시험 시간은 60분입니다.
- 시험이 시작되면 2페이지의 개발 진행 방법을 확인하고 하나씩 진행하시면 됩니다.
- 모든 단계를 완료하게 되면 하나의 웹 SW가 완성됩니다.
- 단계별로 점수가 () 안에 기재되어 있습니다.
- 시험이 종료되면 개발 결과물을 두고, 모두 퇴장합니다.
- 호명하는 학생들은 한 명씩 자신의 컴퓨터로 와서 시험결과에 대해 함께 채점을 진행합니다. 이때 교수님이 결과에 관해 설명을 요구할 수도 있습니다. 부담 없이 본인이 아는 대로 설명해주시면 됩니다.
- 채점이 완료된 학생은 바로 퇴실하셔도 무방합니다.

○ 요구사항

- 메인(home) 페이지 개발
- 상품(product) 목록 페이지 개발
 - 1) 검색 조건에 따른 검색 기능 구현
 - 2) 페이징 구현(이전, 다음 버튼만 구현)
 - 3) 등록 버튼을 통한 페이지 이동
- 상품 상세 및 등록 페이지 개발
 - 1) html로 상세 정보를 조회하는 페이지 작성

(최종산출물)

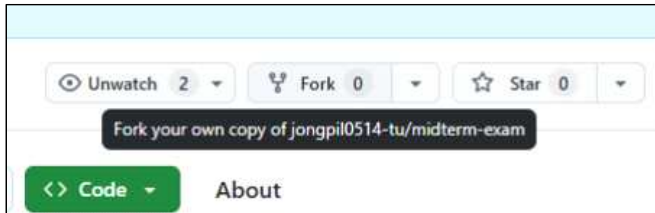
- 메인 페이지 개발
- 상품 리스트 출력 페이지 개발
- 상품 상세 페이지 개발

○ 중요 정보

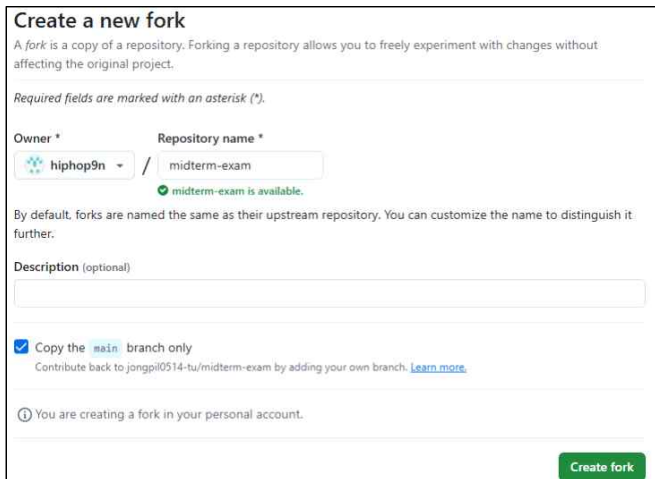
- <https://github.com/jongpil0514-tu/midterm-exam>
- git에서 본인 repository로 Fork 후에 git pull 받고, 프로젝트 import 하기
- STS4 상에 midterm-exam을 제외한 나머지 프로젝트 삭제.
- ※ 왼쪽 프로젝트에 midterm-exam이 아닌 프로젝트 발견되는 경우 퇴실 조치
- Dbeaver
 - 1) Dbeaver로 테이블 정보 띄워두기
 - 2) Select * from tb_product 실행해 두기
- Browser
 - 1) 브라우저에 http://localhost:8080/home
 - 2) 브라우저에 http://localhost:8080/product/list
 - 3) 브라우저에 http://localhost:8080/product/detail/19
- STS4(이클립스) 작업 결과물

Step 1. [Git 활용] Spring Boot 프로젝트 생성 및 셋팅 (총 5점)

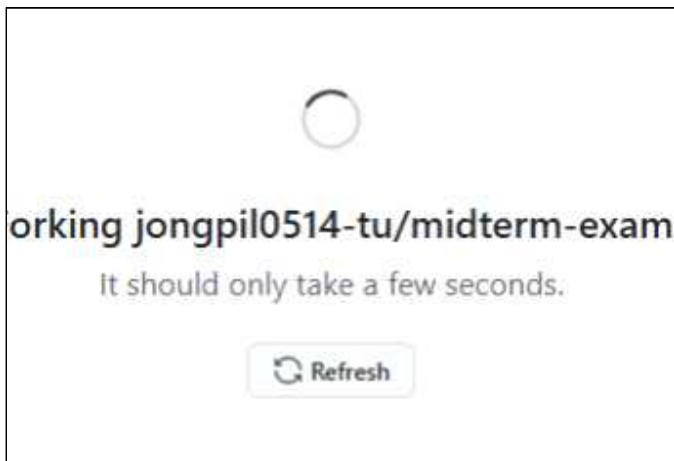
1) 위 URL에 Git Fork를 통해 본인의 github로 Repository를 복사



< git fork >

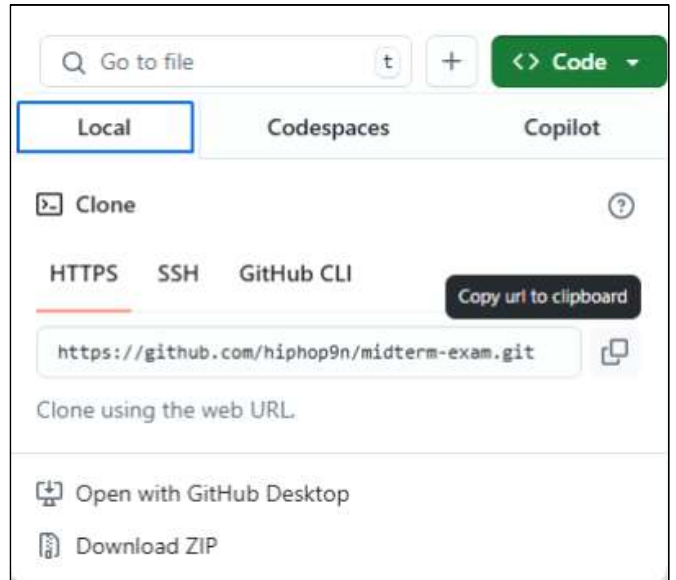


< create a new fork : repository 이름 변경하지 말 것 >

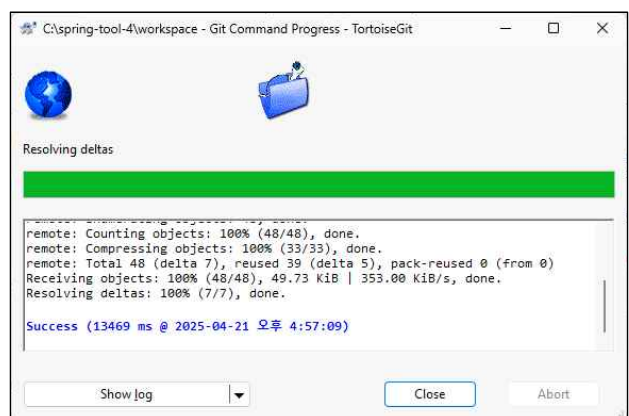
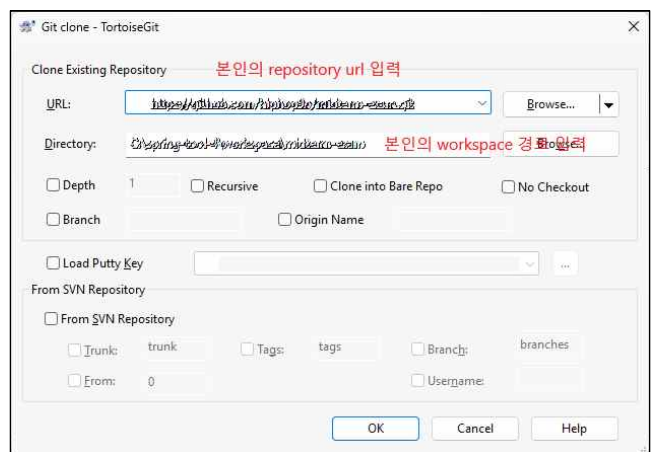


< fork 진행 중 >

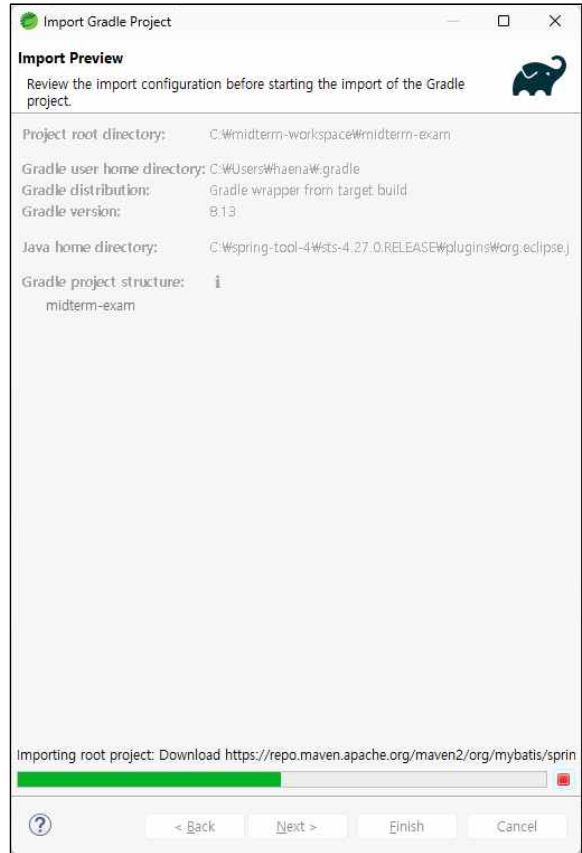
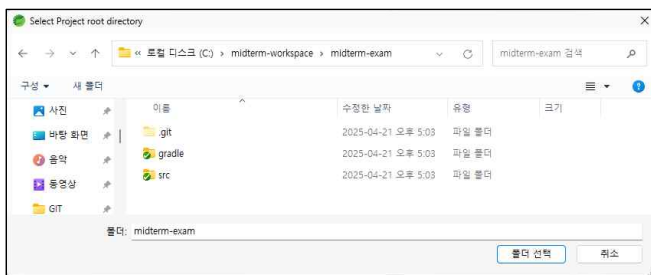
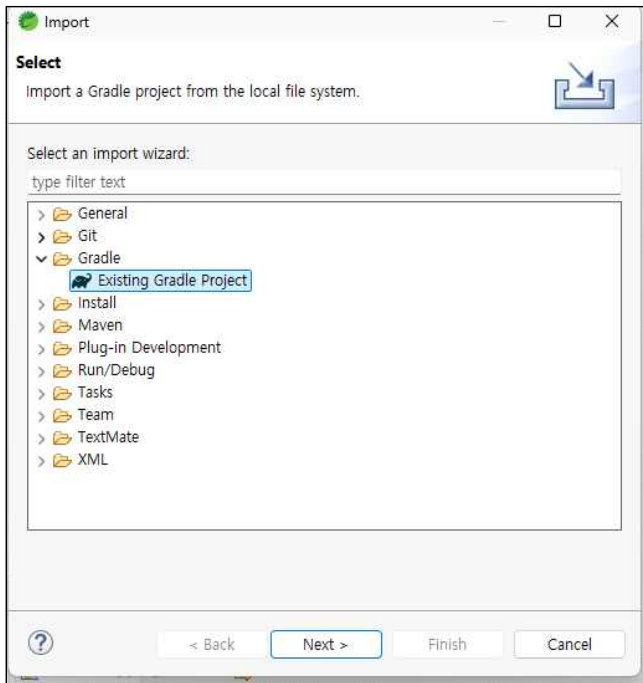
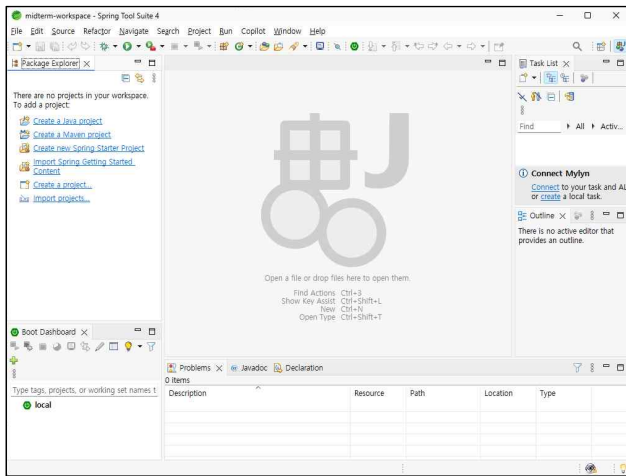
2) 자신의 Repository의 Code 버튼을 눌러서 Git URL을 복사



3) 자신의 개발 환경 내에 local 폴더에 git clone을 통해 소스 다운로드



4) STS4에 소스 Import



위와 같이 DataSource 관련 오류가 나면 정상 import 된 것.

Step 3 문제부터 해결하면 정상적으로 사용할 수 있음.

※ 힌트를 위해 기존 소스를 그대로 두었으나, 본인이 작성한 코드로 제출하고 싶다면 각 문제가 요구하는 형식만 맞추면 나머지는 어떻게 작성하든 감점 요인 아님

Step 2. DataBase 설정 (총 10점)

1.0 Database 생성 및 계정 할당

※ 아래 정보로 DB 스크립트를 생성하시오.

DB name : book
 USER name : {본인의 ID}
 USER pw : {본인의 PW}

2.1 tb_product 테이블을 아래 ER-D를 보고 Table을 만드시오.(Dbeaver로 테이블 더블클릭하여 열어두기) (5점)

물리 이름	논리 이름	도메인	데이터 타입	널 허용
product_seq	상품순번	N/A	INT(11) UNSIGNED	N-N
product_type	상품유형	N/A	VARCHAR(20)	N-N
product_name	상품명	N/A	VARCHAR(50)	N-N
product_price	상품가격	N/A	INT(10) UNSIGNED	N-N
reg_dt	등록일시	N/A	DATETIME	N-N
mod_dt	수정일시	N/A	DATETIME	N-N

테이블명:	tb_product
Engine:	InnoDB
Auto Increment:	1
Charset:	utf8mb4
Collation:	utf8mb4_general_ci
Description:	상품

컬럼명	#	Data Type	Not Null	Auto Increment	Key	디폴트	Extra	Comment
product_seq	1	int(11) unsigned	[v]	[v]	PK		auto_increment	상품순번
product_type	2	varchar(20)	[]	[]		NULL		상품유형
product_name	3	varchar(50)	[]	[]		NULL		상품명
product_price	4	int(10) unsigned	[]	[]		NULL		상품가격
reg_dt	5	datetime	[]	[]		NULL		등록일시
mod_dt	6	datetime	[]	[]		NULL		수정일시

- Table name : tb_product
- PK : product_seq (Auto Increment 속성)
- Table engine : InnoDB
- character set : utf8mb4
- collate : utf8mb4_general_ci

	123	A-Z pro	A-Z produ	123 prc	reg.	mod
11	11	식품	라면1PACK	8,000	-21 13:51:37	-21 13:51:37.0
12	12	전자기기	갤럭시S25	1,100,000	-21 13:51:37	-21 13:51:37.0
13	13	AI	챗GPT	30,000	-21 13:51:38	-21 13:51:38.0
14	14	자동차	요소수	10,000	-21 13:51:38	-21 13:51:38.0
15	15	식품	라면1PACK	8,000	-21 13:51:38	-21 13:51:38.0
16	16	전자기기	갤럭시S25	1,100,000	-21 13:51:38	-21 13:51:38.0
17	17	AI	챗GPT	30,000	-21 13:51:39	-21 13:51:39.0
18	18	자동차	요소수	10,000	-21 13:51:39	-21 13:51:39.0
19	19	식품	라면1PACK	8,000	-21 13:51:39	-21 13:51:39.0
20	20	전자기기	갤럭시S25	1,100,000	-21 13:51:39	-21 13:51:39.0

2.2. 그림과 같이 출력될 수 있도록 샘플 데이터를 20개를 삽입한다. (reg_dt, mod_dt는 NOW()를 사용하여 입력) (5점)

- 컬럼 : product_seq(auto increment), product_type, product_name, product_price, reg_dt, mod_dt
- [AI, 챗GPT, 30000], [자동차, 요소수, 10000], [식품, 라면 1PACK, 8000], [전자기기, 갤럭시S25, 110000] 데이터를 반복하여 총 20개 생성
- script 작성 후 총 5회 수행하여 목록 20개 생성되었는지 확인(마지막 seq가 20번인지 확인)
- SELECT * FROM tb_product 실행해두기

Step 3. application.properties 수정 & 프로그램 구동 (총 15점)

3.1 application.properties 설정을 완성하시오. (10점)

```
spring.application.name=midterm-exam

spring.thymeleaf.enabled=true
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html

spring.datasource.url=jdbc:mariadb://
spring.datasource.username=
spring.datasource.password=
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

mybatis.mapper-locations=classpath:/mybatis/mapper/**/*.xml
mybatis.configuration.map-underscore-to-camel-case=true
```

< application.properties >

3.2 boot application이 구동된 후 System.out.println을 통해 “중간고사 프로젝트 생성 성공” 메시지를 띄우시오. (5점)



:: Spring Boot ::

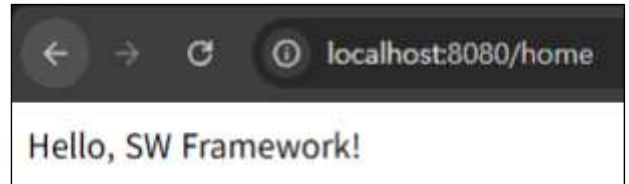
(v3.4.4)

```
2025-04-21T11:56:44.800+09:00 INFO 40176 --
\midterm-exam)
2025-04-21T11:56:44.802+09:00 INFO 40176 --
2025-04-21T11:56:44.832+09:00 INFO 40176 --
2025-04-21T11:56:44.832+09:00 INFO 40176 --
2025-04-21T11:56:45.263+09:00 WARN 40176 --
2025-04-21T11:56:45.514+09:00 INFO 40176 --
2025-04-21T11:56:45.525+09:00 INFO 40176 --
2025-04-21T11:56:45.525+09:00 INFO 40176 --
2025-04-21T11:56:45.565+09:00 INFO 40176 --
2025-04-21T11:56:45.566+09:00 INFO 40176 --
2025-04-21T11:56:45.905+09:00 INFO 40176 --
2025-04-21T11:56:45.936+09:00 INFO 40176 --
2025-04-21T11:56:45.942+09:00 INFO 40176 --
중간고사 프로젝트 생성 성공
```

Step 4. HomeController 생성 (총 10점)

4.1 src/main/java/com/midterm/home 패키지 하위에 HomeController.java를 생성하여 “http://localhost:8080/home”를 요청하면 “Hello, SW Framework!”를 출력하도록 코딩하고, Lombok(@Slf4j 사용)을 사용하여 “메인페이지 요청” 로그를 INFO 레벨로 작성하시오. (10점)

- 1) 페이지 출력만하면 5점
- 2) Lombok을 통해 로그까지 찍으면 10점



```
p.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServ
p.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherSe
p.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
com.midterm.home.HomeController : 메인페이지 요청
```


Step 5. 상품 목록 페이지 (총 45점)

프로젝트 최상위에 있는 sample 폴더에 관련 파일 활용

5.1 프로젝트 최상위에 있는 sample 폴더에 step5.html 파일을 이용하여 src/main/resources/templates/product 폴더(패키지)를 생성하고, list.html을 완성하시오.(5점)

- 목록 출력 되어야 함
- 목록 항목 중 상품 이름에 a 링크를 설정하여 상세 페이지 이동 기능 추가
- 검색 했을 때 브라우저의 URL에 검색 조건이 표기되고, 주소를 복사하여 새탭에 넣었을 때 페이지가 재현 되어야 함

5.2 src/main/java/midterm/product 폴더(패키지)를 생성하고, "http://localhost:8080/product/list"를 요청했을 때 상품 목록과 검색을 모두 할 수 있는 Controller 메서드를 ProductController.java에 완성하시오.(5점)

- sample 하위에 ProductController.java 파일 활용
- 관련 어노테이션 정확히 설정할 것
- 생성자 방식으로 ProductService 객체 주입
- @ModelAttribute 방식으로 ProductForm 객체를 받아서 처리

5.3 src/main/java/midterm/product 폴더(패키지)를 생성하고, 검색 및 페이지징 처리하는 Service 메서드를 포함하는 ProductService.java를 완성하시오.(20점)

- sample 하위에 ProductService.java 파일 활용
- 생성자 방식으로 ProductMapper 객체 주입
- 관련 어노테이션 정확히 설정할 것
- @Transactional 어노테이션 사용법에 맞게 정확히 명시할 것
- 비즈니스 로직 자유롭게 작성하되 TotalCount, List를 조회할 때 정확히 가져와야 함.
- 로그가 정상 출력되어야 함
- 페이지징 로직을 통해 번호가 정확히 출력되도록 작성
- 페이지징 노출 목록(PAGE_BLOCK_LIST)

5.4 src/main/java/midterm/product 폴더(패키지)를 생성하고, Lombok을 사용하여 DTO인 ProductForm.java를 완성하시오.(5점)

- pageNum, searchProductName, searchProductType
- 관련 어노테이션 정확히 설정할 것
- 샘플 소스 없이 직접 생성

5.5 src/main/java/midterm/product/mapper 폴더(패키지)를 생성하고, 상품 개수와 상품 목록을 조회하는 ProductMapper.java를 완성하시오.(5점)

- 상품 개수 메서드명 : selectProductTotalCount(리턴 타입 : int)
- 상품 목록 메서드명 : selectProductList(리턴타입 : List)
- 관련 어노테이션 정확히 설정할 것
- 샘플 소스 없이 직접 생성
- 검색을 처리하기 위한 Mybatis 문법 사용

5.6 src/main/resources/mybatis/mapper 폴더(패키지)를 생성하고, 아래 ProductMapper.xml을 완성하시오.(5점)

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="">

    <!-- 상품 전체 개수 조회 -->
    <select id="selectProductTotalCount"
resultType="int"
parameterType="java.util.HashMap">
        <![CDATA[

    ]]>
    </select>

    <!-- 상품 목록 조회 -->
    <select id="selectProductList"
resultType="java.util.HashMap"
parameterType="java.util.HashMap">
        <![CDATA[

    ]]>
    </select>

</mapper>
```

< ProductMapper.xml >

Step 5 테스트 시나리오 (검색 요청이 있을 때 마다 URL 변경되어야 함)

- 기본 목록 페이징 출력 : 5개씩 총 4페이지 / 1,2,3,4 페이지 모두 정상 출력

← → ↻

localhost:8080/product/list

☆ ⓘ ⋮

중간 고사 검색 페이지

상품 유형 검색: 상품 이름 검색:

번호	상품유형	상품명	상품가격	등록일시	수정일시
20	전자기기	갤럭시S25	1100000	2025-04-21 13:51:39	2025-04-21 13:51:39
19	식품	리면1PACK	8000	2025-04-21 13:51:39	2025-04-21 13:51:39
18	자동차	요소수	10000	2025-04-21 13:51:39	2025-04-21 13:51:39
17	AI	챗GPT	30000	2025-04-21 13:51:39	2025-04-21 13:51:39
16	전자기기	갤럭시S25	1100000	2025-04-21 13:51:38	2025-04-21 13:51:38

1

2

3

4

< 기본 목록 페이지 >

- 상품 유형 검색어로 AI 입력 후 5개 목록 정상 출력

← → ↻

localhost:8080/product/list?pageNum=1&searchProductType=AI&searchProductName=

☆ ⓘ ⋮

중간 고사 검색 페이지

상품 유형 검색: 상품 이름 검색:

번호	상품유형	상품명	상품가격	등록일시	수정일시
17	AI	챗GPT	30000	2025-04-21 13:51:39	2025-04-21 13:51:39
13	AI	챗GPT	30000	2025-04-21 13:51:38	2025-04-21 13:51:38
9	AI	챗GPT	30000	2025-04-21 13:51:37	2025-04-21 13:51:37
5	AI	챗GPT	30000	2025-04-21 13:51:36	2025-04-21 13:51:36
1	AI	챗GPT	30000	2025-04-21 10:58:30	2025-04-21 10:58:30

1

< 상품 유형 검색 - 검색어 : AI >

- 상품 이름 검색어로 S25 입력 후 5개 목록 정상 출력

← → ↻

localhost:8080/product/list?pageNum=1&searchProductType=&searchProductName=S25

☆ ⓘ ⋮

중간 고사 검색 페이지

상품 유형 검색: 상품 이름 검색:

번호	상품유형	상품명	상품가격	등록일시	수정일시
20	전자기기	갤럭시S25	1100000	2025-04-21 13:51:39	2025-04-21 13:51:39
16	전자기기	갤럭시S25	1100000	2025-04-21 13:51:38	2025-04-21 13:51:38
12	전자기기	갤럭시S25	1100000	2025-04-21 13:51:37	2025-04-21 13:51:37
8	전자기기	갤럭시S25	1100000	2025-04-21 13:51:36	2025-04-21 13:51:36
4	전자기기	갤럭시S25	1100000	2025-04-21 10:58:30	2025-04-21 10:58:30

1

< 상품 이름 검색 - 검색어 : S25>

Step 6. 상세 페이지 (총 15점)

6.1 sample폴더의 step6.html을 통해 상세 페이지를 출력하도록 src/main/resources/templates/product 패키지 하위에 detail.html을 생성하고, ProductController.java, ProductService.java, ProductMapper.java, ProductMapper.xml를 완성하여 아래 코드가 구동되도록 코딩하시오.

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

<head>
    <meta charset="UTF-8">
    <title>상품 상세</title>

    <style>
        table { width: 100%; border-collapse: collapse; margin-top: 20px; }
        th { background-color: #f2f2f2; text-align: center; }
        td { text-align: center; }
        th, td { border: 1px solid #ddd; padding: 8px; }

        .url-link { text-decoration: none; color: blue; display: inline-block; margin-top: 20px; }
    </style>
</head>

<body>
    <div>
        <h2>도서 상세</h2>
        <table>
            <tbody>
                <tr>
                    <th>번호</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.product_seq}"</td>
                </tr>
                <tr>
                    <th>상품유형</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.product_type}"</td>
                </tr>
                <tr>
                    <th>상품명</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.product_name}"</td>
                </tr>
                <tr>
                    <th>상품가격</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.product_price}"</td>
                </tr>
                <tr>
                    <th>등록일시</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.reg_dt}"</td>
                </tr>
                <tr>
                    <th>수정일시</th>
                    <td>
                        th:text="${result.PRODUCT_DETAIL.mod_dt}"</td>
                </tr>
            </tbody>
        </table>

        <a class="url-link" href="@{/product/list}" th:text="목록"></a>
    </div>
</body>
</html>
```

< step6.html >

localhost:8080/product/detail/19

도서 상세

번호	19
상품유형	식품
상품명	라면1PACK
상품가격	8000
등록일시	2025-04-21 13:51:39
수정일시	2025-04-21 13:51:39

목록

- http://localhost:8080/product/detail/19 요청하면 위와 같이 출력되어야 함
- HTML 상에 있는 변수명 변경 불가
- Controller, Service, Mapper, xml 등 비즈니스 로직을 자유롭게 작성하여 아래 페이지를 구현하여 출력하면 점수 획득
- RestFul URL 요청으로 처리해야 함
- 등록일시, 수정일시는 다를 수 있으나 나머지 정보는 똑같이 출력되어야 함.