

# Software Project1 Proposal

소프트웨어프로젝트1

# 프로젝트 개요

- 사진을 업로드 하고 해당 사진에 대한 간단한 소개를 작성하는 홈페이지 구현
  - ✓ 사진 업로드
  - ✓ 사진 관련 타이틀 작성
  - ✓ 사진 관련 글 작성


# 프로젝트 목표

- Spring Boot를 활용한 서버 구현
  - ✓ MVC 패턴을 이용한 서버 구현
  - ✓ H2 database 사용
  - ✓ 파일 업로드 리소스 컨트롤
- 웹 페이지 구현
  - ✓ 사진을 보여주는 홈페이지 구현
  - ✓ 사진 업로드 페이지 구현

# Version Control

- JAVA
  - ✓ **Openjdk17**
- Spring
  - ✓ **gradle 3.2.5**
  - ✓ **H2 database**
  - ✓ **Spring JPA**
  - ✓ **Thymeleaf**
  - ✓ **Spring Web**
  - ✓ **Lombok**

# 프로젝트 목표



**Project**  
☒ Gradle - Groovy ☐ Gradle - Kotlin  
☐ Maven

**Language**  
☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (RC1) ☐ 3.2.6 (SNAPSHOT) ☒ 3.2.5  
☐ 3.1.12 (SNAPSHOT) ☐ 3.1.11

**Project Metadata**  

Group

com.example

Artifact

Proj1\_2023160602

Name

Proj1\_2023160602

Description

Demo project for Spring Boot

Package name

com.example.Proj1\_2023160602

Packaging

☒ Jar ☐ War

Java

☐ 22 ☐ 21 ☒ 17

**Dependencies** [ADD DEPENDENCIES... CTRL + B](#)

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

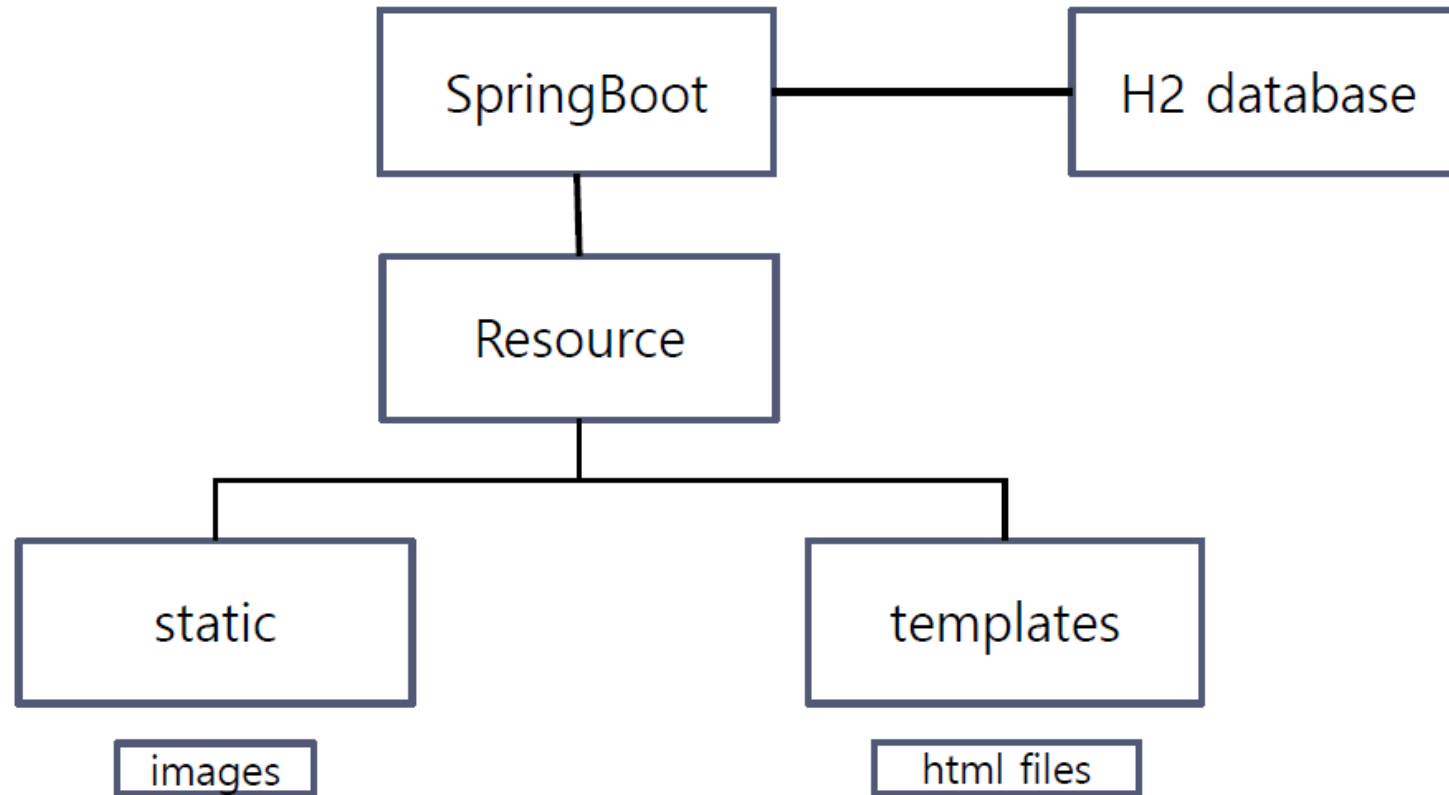
**Thymeleaf** TEMPLATE ENGINES  
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

**Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

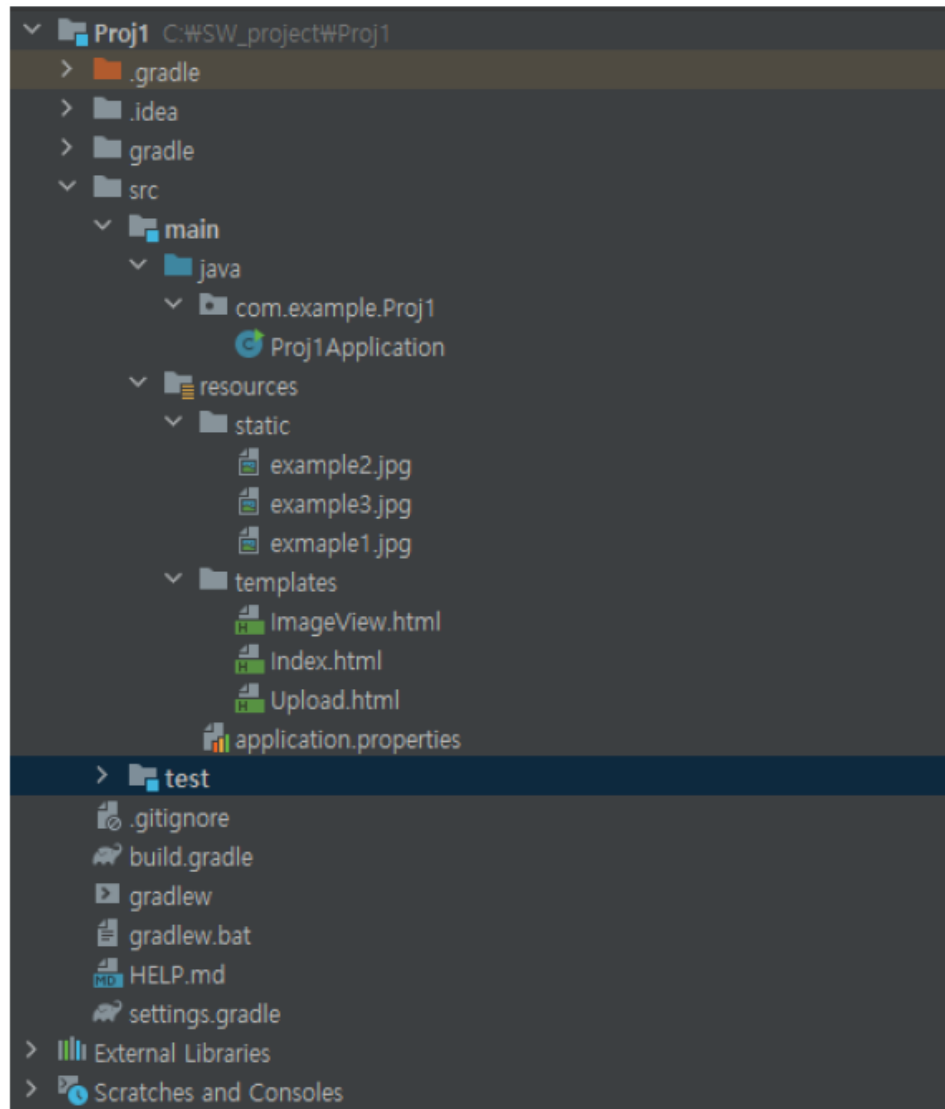
**H2 Database** SQL  
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

**Lombok** DEVELOPER TOOLS  
Java annotation library which helps to reduce boilerplate code.

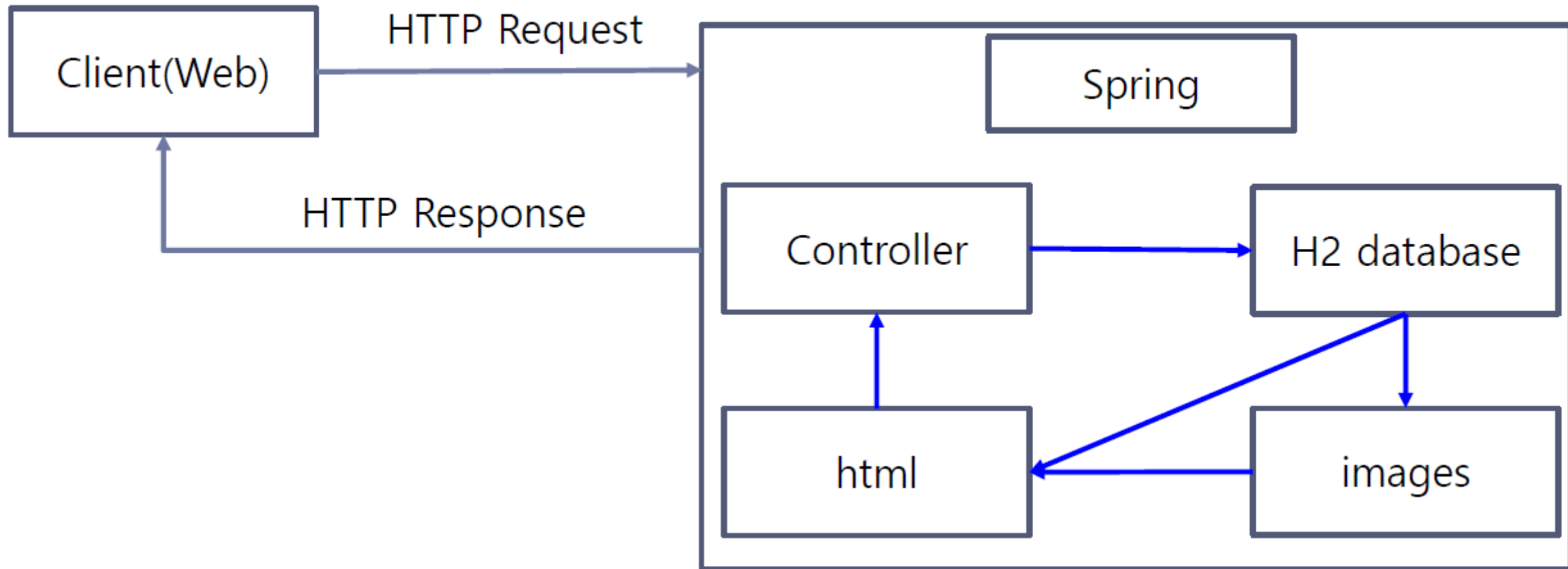
# 시스템 아키텍처



# 시스템 아키텍처



# 예상 흐름도





# Index.html

사진 올리기



Title



Title



Title

- Index.html
  - ✓ 사진과 간단한 타이틀 업로드 최신순으로 시각화
  - ✓ 타이틀은 최대 길이 20
  - ✓ 사진과 타이틀은 모두 클릭 가능하도록 하며, 클릭할 시 ImageView.html을 통해 관련 내용 시각화
  - ✓ 사진 올리기 버튼 누를 시, Upload.html로 이동
  - ✓ “localhost:8080/”과 “localhost:8080/index.html” 모두 접근 가능하도록 처리할 것

# Upload.html

Input : Title(최대 20자)

Input : Comment(최대 100자)

파일 선택 :

- Spring Boot를 활용한 서버 구현
  - ✓ 파일 이미지만 받을 수 있도록 처리할 것
  - ✓ 이미지는 .jpg, .jpeg, .png 파일만 업로드 가능
  - ✓ Title은 최대 20자, Comment는 최대 100자 제한
  - ✓ 업로드 완료 후. Index.html로 이동

# ImageView.html



Title

Comment(최대 100자)

수정

삭제

- ImageView.html
  - ✓ Index.html에서 누른 사진 또는 타이틀에 대한 정보 시각화
  - ✓ 삭제 시 DB에서 제거
  - ✓ 수정 시 Upload.html 양식으로 돌아가서 데이터 수정

# 제출 목록

- 과제 기한
  - ✓ **Due date : ~ 5/30(목)**
- 제출 파일 (Project + Database)
  - ✓ **Spring 프로젝트 명 : Proj1\_학번**
    - 코드 주석 : 코드에 대한 간단한 설명 적을 것
  - ✓ **Database 명 : Proj1\_DB\_학번**
  - ✓ **Database 경로 : ~/Proj1\_DB\_학번 ( ~는 home directory)**
    - **Username : sa**
    - **Password :**

# 제출 목록

- 보고서
  - ✓ Introduction ( 5줄 이상)
    - ✓ 과제에 대한 전반적인 요약
- Flow Chart
  - ✓ 과제 코드의 전반적인 흐름 설명
  - ✓ 각 view(html) 마다의 흐름도 설명
- Result (사진 첨부)
  - ✓ 프로젝트 내에 MVC Pattern 설명
  - ✓ 각 view(html) 별 구현 방식 설명 및 결과 화면 설명
- Consideration
  - ✓ 구현 시 발생한 문제점 및 개선방안 설명

# End