

소프트웨어프로젝트1

Project 1

담당교수: 이우신 교수님

학과: 컴퓨터정보공학부

학번: 2019202021

이름: 정성엽

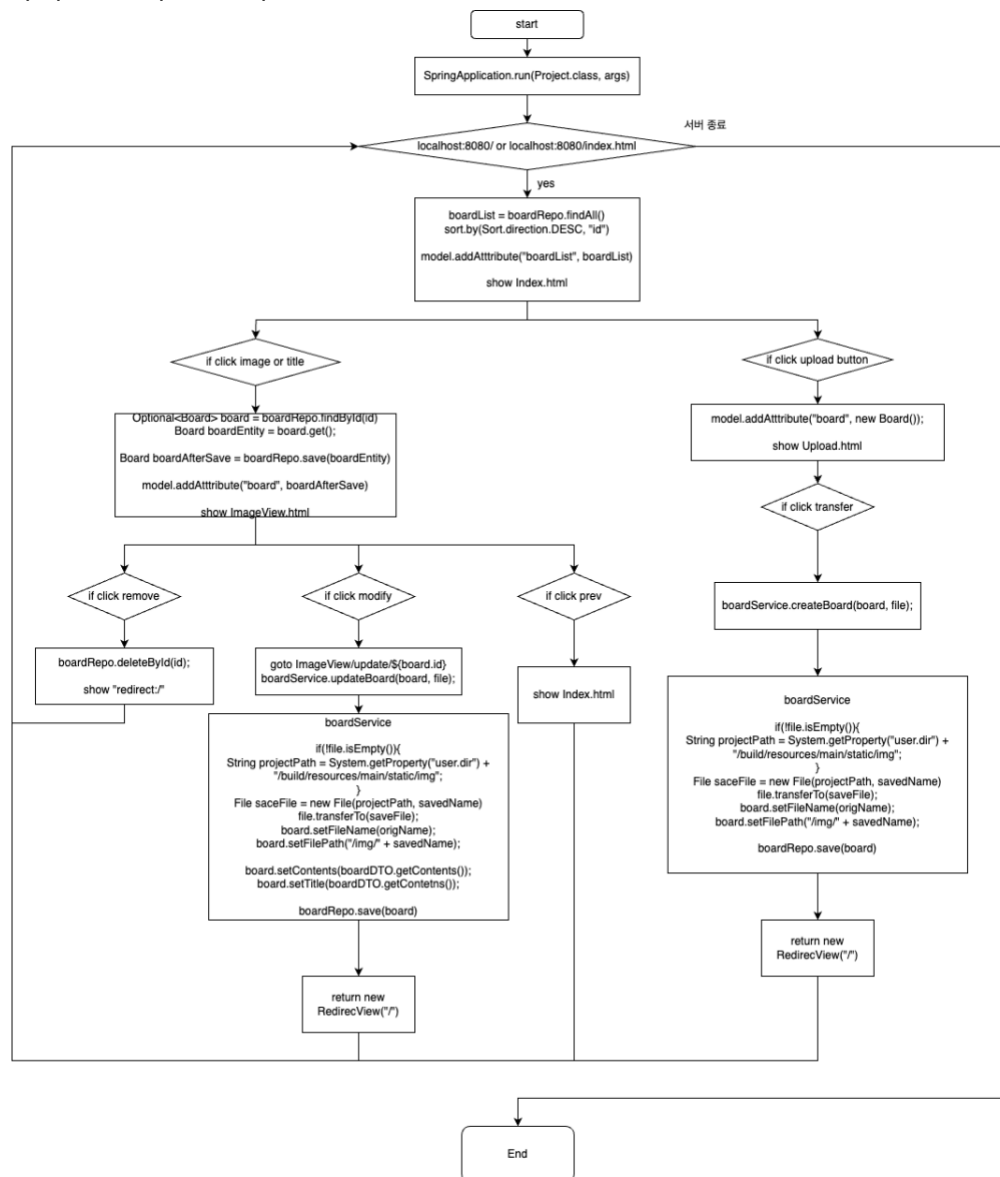
제출일: 2023/06/04

1. Introduction

사진을 업로드 하고 해당 사진에 대한 간단한 소개를 작성하는 홈페이지를 구현한다. 이 때 Spring Boot를 활용하여 서버를 구현하되, MVC 패턴을 이용하여 구현하고 H2 database를 사용한다. 그리고 파일 업로드 리소스를 컨트롤 한다. 웹페이지 구현할 때는 사진들을 보여주는 홈페이지 구현과 사진 업로드 또는 수정하는 페이지를 구현한다. 사진이나 제목 클릭 시 ImageView.html로 넘어가고 Upload나 수정할 시 Upload.html 양식으로 돌아가서 데이터를 수정한다.

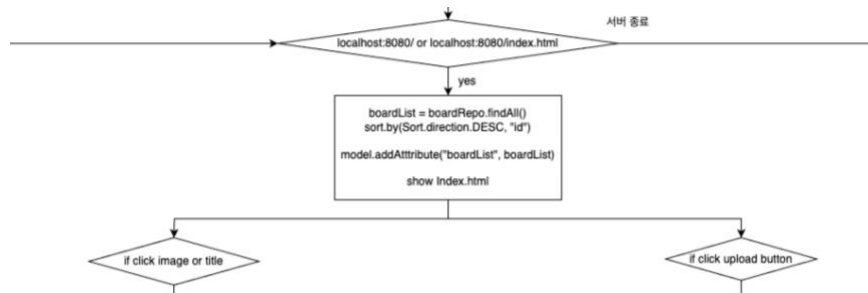
2. Flow Chart

A. 과제 코드의 전반적인 흐름



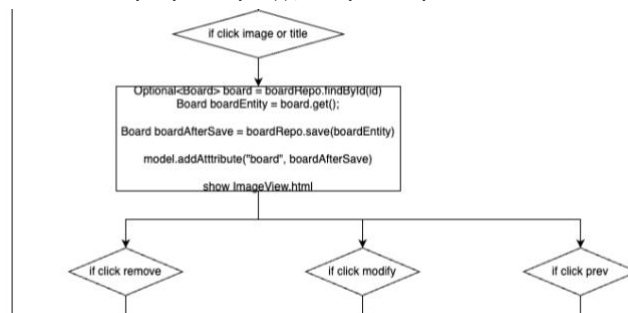
서버를 시작하면 SpringApplication.run 명령어에 의해 Spring boot의 어플리케이션을 초기화하고 설정한다. 내장된 Tomcat 서버를 시작하고, 웹 어플리케이션을 배포한다. 필요한 빈(Beans)들을 로드하고, Dependency Injection을 수행한다. 그리고 어플리케이션 Context를 생성하고 필요한 Beans들을 구성 후 어플리케이션을 실행한다. 위의 그림은 Index.html부터 접속했을 때를 기준으로

로 했을 때의 FlowChart이다.

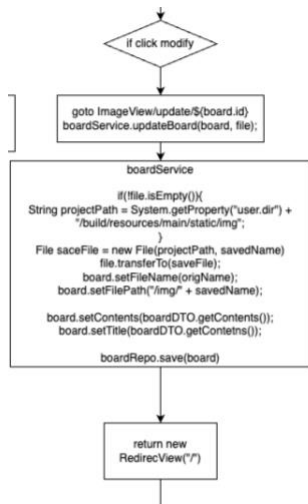


만약 브라우저에서 “localhost:8080/” 또는 “localhost:8080/index.html”으로 접속했다고 한다면 h2 database의 board 요소들을 Array에 저장하고 이를 addAttribute를 통해 html로 넘길 수 있도록 설정한다. 이후 Index.html을 나타낸다.

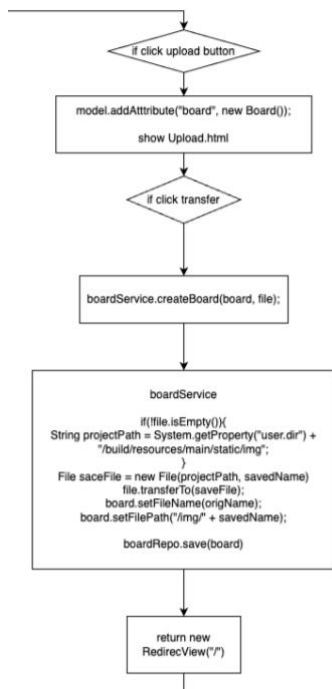
Index.html의 요소 중 image source 또는 title을 클릭하는 경우 ImageView/{board.id} 로 이동하도록 하여 해당 사진의 상세정보 페이지로 넘어가도록 하며, 만약 button 요소를 클릭하는 경우 Upload로 이동하여 새로운 글을 추가할 수 있도록 한다.



만약 브라우저에서 “localhost:8080/ImageView/{id}”로 접속하거나 Index.html에서 이미지나 제목을 클릭한다면 해당 id의 상세페이지를 나타내도록 한다. 이 때 id를 통해 findById 함수를 사용하여 찾으며 boardEntity에 해당 정보를 저장한다. 이후 이를 boardAfterSave에도 저장하여 addAttribute를 통해 html로 넘겨준다. 그 이후 상세페이지를 나타내는 ImageView.html을 나타낸다. 만약 삭제 버튼을 누른다면 post를 통해 해당 페이지에서의 삭제를 알리고, 수정이라면 수정하는 페이지로 넘어간다. 이전으로 버튼을 누른다면 Index.html로 돌아온다.

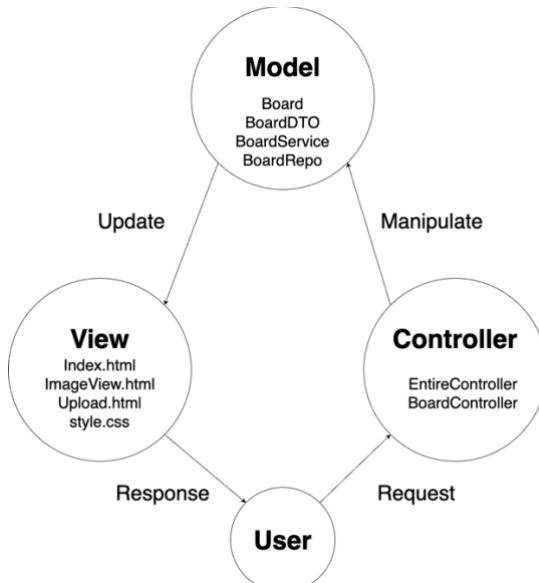


수정 버튼을 누르면 Upload.html 양식으로 ImageView/update/\${board.id}로 이동한다. BoardService의 uploadBoard를 호출하여 수정을 하며 만약 새로 들어간 파일이 없다면 제목과 내용만 수정하고, 있다면 파일을 다시 받아서 이를 서버에 넘기고 해당 경로를 저장하고 제목과 내용을 같이 수정한다. 이때 이미지는 build/resources/main/static/img 폴더 내에 저장하고 h2 database에는 id, title, content, 그리고 파일 이름, 파일 경로를 저장한다. 이후 홈 화면으로 redirect 하여 돌아간다.



index.html에서 사진 업로드 버튼을 클릭한 경우 새로운 board를 만들어 addAttribute 하고 Upload.html을 띄운다. 새로운 내용과 사진을 올리면 /board로 이동하고 새로운 board 내용을 h2 database에 저장한다. 이 때 사진은 수정했을 때와 마찬가지로 build/resources/main/static/img 폴더에 저장한다.

3. Result



이번 프로젝트의 MVC 패턴을 그림으로 설명하면 위와 같다. 먼저 Model은 Board, BoardDTO, BoardService, BoardRepo가 포함된다. 이는 @Entity, @Repository, @Service에 해당하며 데이터를 나타내고 상태를 관리하고 비즈니스 로직을 처리한다. 또한 파일 업로드 및 게시판의 경우, 데이터베이스와의 상호 작용을 담당하며 파일의 저장, 수정, 삭제 등을 처리한다.

View는 Index.html, ImageView.html, Upload.html, style.css가 포함되며, html파일 또는 css 파일이 해당한다. 이는 브라우저에서 사용자에게 데이터를 시각적으로 보여 주는 역할을 한다.

Controller는 EntireController, BoardController가 해당하고 @Controller가 해당한다. 이는 사용자의 요청을 받고 처리하는 역할을 한다.

```

Run -> [Proj1_2019020201] [Proj12019020201Application.main]
> Task :compileJava UP-TO-DATE
> Task :processResources
> Task :classes
> Task :[Proj12019020201Application.main]
21:19:49.487 [Thread-0] DEB org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.de

...

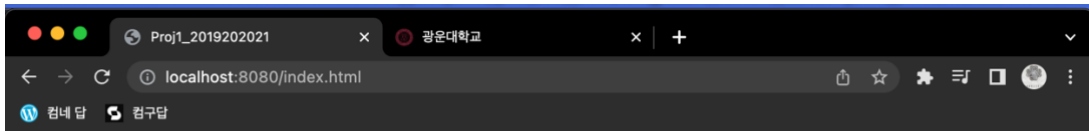
2023-06-04 21:19:49.773 INFO 5319 --- [ restartedMain] c.e.p.Proj12019020201Application : Starting Proj12019020201Application using Java 11.0.19 on j
2023-06-04 21:19:49.774 INFO 5319 --- [ restartedMain] c.e.p.Proj12019020201Application : No active profile set, falling back to 1 default profile: "
2023-06-04 21:19:49.814 INFO 5319 --- [ restartedMain] s.DevToolsPropertyDefaultPostProcessor : DevTools property defaults active! Set 'spring.devtools.adc
2023-06-04 21:19:49.824 INFO 5319 --- [ restartedMain] s.DevToolsPropertyDefaultPostProcessor : For additional web related logging consider setting the 'lo
2023-06-04 21:19:50.152 INFO 5319 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-06-04 21:19:50.181 INFO 5319 --- [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 23 ms. Found 1
2023-06-04 21:19:50.515 INFO 5319 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8088 (http)
2023-06-04 21:19:50.521 INFO 5319 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-04 21:19:50.522 INFO 5319 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.74]
2023-06-04 21:19:50.560 INFO 5319 --- [ restartedMain] o.s.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-06-04 21:19:50.568 INFO 5319 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 746
2023-06-04 21:19:50.576 INFO 5319 --- [ restartedMain] org.springframework.hateoas.config.HypermediaConfiguration : HypermediaConfiguration
2023-06-04 21:19:50.677 INFO 5319 --- [ restartedMain] org.springframework.hateoas.config.HypermediaConfiguration : HypermediaConfiguration
2023-06-04 21:19:50.685 INFO 5319 --- [ restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/Proj1_08_2019020201'. Database av
2023-06-04 21:19:50.778 INFO 5319 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HH000284: Processing PersistenceUnitInfo [name: default]
2023-06-04 21:19:50.805 INFO 5319 --- [ restartedMain] org.hibernate.Version : HH000012: Hibernate ORM core version 5.6.15.Final
2023-06-04 21:19:50.895 INFO 5319 --- [ restartedMain] o.hibernate.annotations.common.Version : HH00000001: Hibernate Commons Annotations 5.1.3.Final
2023-06-04 21:19:50.964 INFO 5319 --- [ restartedMain] org.hibernate.dialect.Dialect : HH000040: Using dialect: org.hibernate.dialect.H2Dialect
2023-06-04 21:19:51.242 INFO 5319 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HH000049: Using JtaPlatform implementation: [org.hibernate
2023-06-04 21:19:51.244 INFO 5319 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit '
2023-06-04 21:19:51.457 INFO 5319 --- [ restartedMain] org.springframework.hateoas.config.HypermediaConfiguration : HypermediaConfiguration
2023-06-04 21:19:51.545 INFO 5319 --- [ restartedMain] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index
2023-06-04 21:19:51.643 INFO 5319 --- [ restartedMain] o.s.b.a.o.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-06-04 21:19:51.659 INFO 5319 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8088 (http) with context path ''
2023-06-04 21:19:51.665 INFO 5319 --- [ restartedMain] c.e.p.Proj12019020201Application : Started Proj12019020201Application in 2.171 seconds (JVM ru
  
```

IntelliJ에서 해당 프로젝트를 run 하였을 때 화면으로 정상적으로 서버가 부팅되었다.



Proj1_2019202021

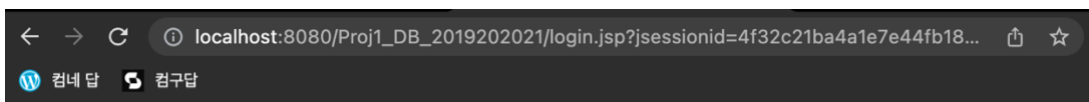
사진 올리기



Proj1_2019202021

사진 올리기

localhost:8080과 localhost:8080/index.html 모두 접속이 가능하다. 개인 프로젝트를 나타내기 위해 프로젝트명을 나타내었다. 또한 사진 올리기 버튼이 존재하며 현재 h2 데이터베이스에 아무것도 없기 때문에 빈 화면이 나타난다.



English Preferences Tools Help

Login

Saved Settings: Generic H2 (Server)

Setting Name: Generic H2 (Server) Save Remove

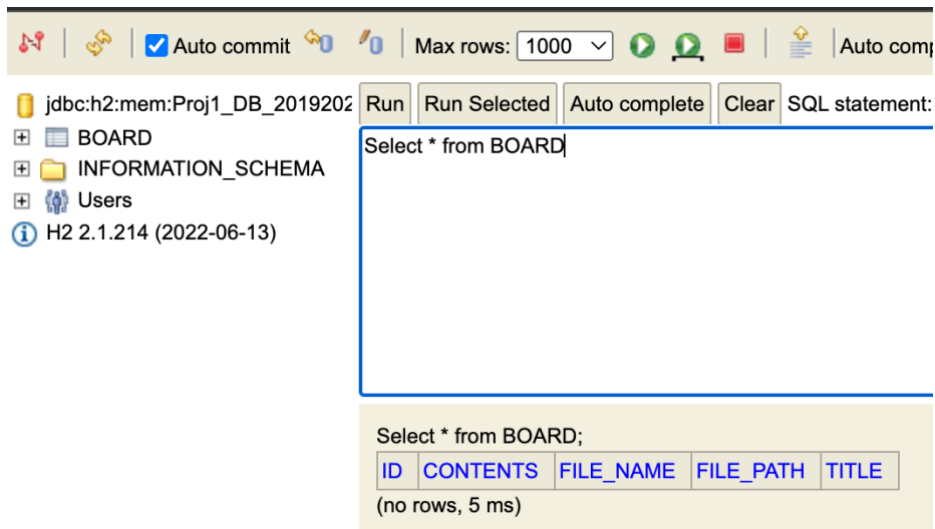
Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:Proj1_DB_2019202021

User Name: sa

Password:

Connect Test Connection



h2_database에 접근하여 BOARD의 정보를 읽으면 아무것도 저장되지 않음을 확인할 수 있다. 또한 BOARD는 ID, CONTENTS, FILE_NAME, FILE_PATH, TITLE의 Column을 가진다.



생성

파일 선택

선택된 파일 없음

전송

사진 올리기 버튼을 눌렀을 때의 화면이다. 생성에 대한 화면임을 나타내도록 text를 두었고, 맨 위 textarea는 title, 그 다음 textarea는 content이며 아래 파일 선택을 하면 컴퓨터의 파일에 접근할 수 있다. 그리고 전송 버튼을 누를 수 있도록 하였다.

생성

12345678901234567890
12345678901234567890123456789012345678901234567890123456789012 34567890123456789012345678901234567890

또한 title은 20자, content는 100자로 한정되있도록 html에서 maxlength를 설정하였다. 아래는 업로드 예시이다.

생성

Celebrity

아이유 2021년 01월 27일 발매

파일 선택

celebrity.jpg

전송

이를 전송하면 아래 사진과 같다.

Proj1_2019202021



Celebrity

사진 올리기

이와 같이 홈 화면으로 이동하고 해당 사진과 title에 대해 정보를 나타내는 것을 볼 수 있다. 이 때 해당 사진을 클릭하면 아래와 같다.



Celebrity

아이유 2021년 01월 27일 발매

수정 이전으로 삭제

해당 id별 이미지와 title, content 모두 나타나고 수정 버튼, 이전으로 버튼, 삭제 버튼이 존재한다. 이때 수정 버튼을 누르면 아래와 같은 화면으로 이동한다.

h2 database에는 아래 사진과 같이 저장되었다.

<

h2 database에서 FILE_PATH를 보면 /img/ directory 아래 의미를 알 수 없는 이름으로 되어있는 jpg를 확인할 수 있는데, 이는 UUID를 random으로 받아와서 저장을 했기 때문이다.

수정

Celebrity

아이유 2021년 01월 27일 발매

파일 선택

선택된 파일 없음

수정

기존 title과 content는 유지되어 있으며 파일 선택은 비워져있다. 이 때 파일을 선택하지 않고 수정 버튼을 누르면 사진은 그대로 유지되며 파일을 선택했다면 해당 사진으로 변경한다. 사진 업로드 버튼과 같은 html을 공유하지만 if문과 unless문을 통해 구별하였다.

아래와 같이 content만 수정해보았다.

수정

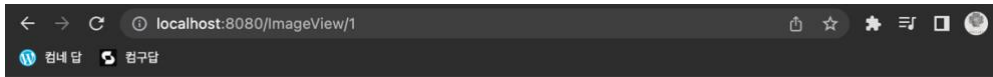
Celebrity

2021년 1월 27일에 발매된 아이유의 정규 5집 LILAC의 선공개 곡.

파일 선택

선택된 파일 없음

수정



Celebrity

2021년 1월 27일에 발매된 아이유의 정규 5집 LILAC의 선공개 곡.

[수정](#) [이전으로](#) [삭제](#)

Content가 정상적으로 바뀌었고 제목과 사진은 유지되었다.

수정

밤편지

2017년 3월 24일에 발매된 아이유의 정규 4집 palette의 첫 번째 선공개 곡.

[파일 선택](#) 밤편지.jpg

[수정](#)

Proj1_2019202021

사진 올리기



밤편지



밤편지

2017년 3월 24일에 발매된 아이유의 정규 4집 Palette의 첫 번째 선공개 곡.

수정 | 이전으로 | 삭제

title, content, 이미지를 모두 수정하였을 때도 모두 정상적으로 변경됨을 확인할 수 있다.

h2 database에는 아래와 같이 수정되었다.

jdbc:h2:mem:Proj1_DB_20192021 Run Run Selected Auto complete Clear SQL statement:

Select * from BOARD

ID	CONTENTS	FILE_NAME	FILE_PATH	TITLE
1	2017년 3월 24일에 발매된 아이유의 정규 4집 Palette의 첫 번째 선공개 곡.	밤편지.jpg	/img/07b40053-f90d-4cc2-812e-5433b86eac80.jpg	밤편지

(1 row, 1 ms)

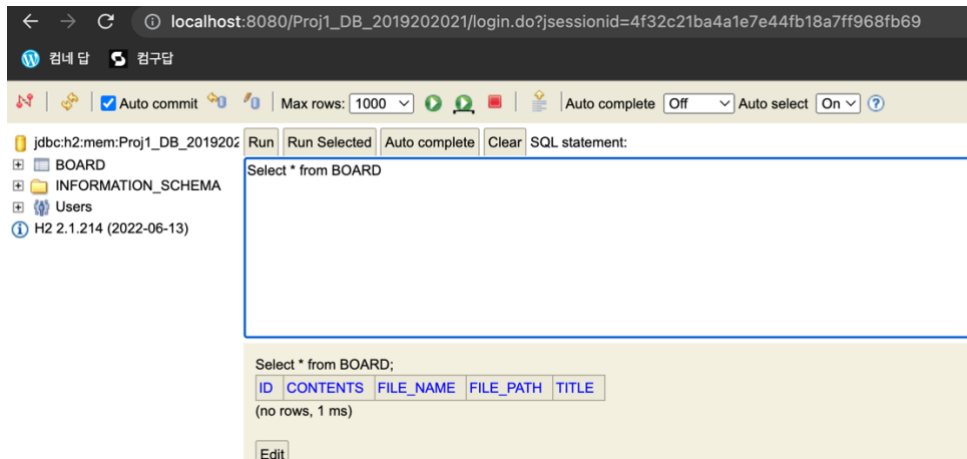
Edit

이제 해당 게시글을 삭제 버튼을 눌러서 삭제하면 아래와 같이 index화면에서도 삭제됨을 확인할 수 있다.



Proj1_2019202021

사진 올리기

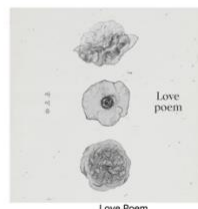


h2 database에서도 삭제되었다.

여러 사진을 넣어도 정상적으로 작동한다. 이는 아래 사진을 통해 확인 할 수 있다.



Proj1_2019202021



Love Poem



Celebrity

사진 올리기



Proj1_2019202021



사랑이 참



말만자



가을아침

사진 올리기

목록들은 좌우 스크롤을 통해 확인 할 수 있다. 또한 등록의 최신순으로 정렬하여 출력하도록 설계하였다.

h2 database를 통해 최신순으로 나열되어있음을 확인할 수 있다.(id의 역순)

Select * from BOARD;

ID	CONTENTS	FILE_NAME	FILE_PATH	TITLE
2	이른 아침 작은 새들 노랫소리 들려오면 언제나 그랬듯 아쉽게 잠을 깬다 창문 하나 햇살 가득 눈부시게 비쳐오고 서늘한 냉기에 재채기할까 말까 음	가을아침.jpg	/img/cf5e0da8-f24b-4cbe-ac0f-e17040defc40.jpg	가을아침
3	이 밤 그날의 반딧불을 당신의 창 가까이 보낼게요 음 사랑한다는 말이에요	밤편지.jpg	/img/f532621d-66b5-41e2-8cfd-adc5dd70bdd4.jpg	밤편지
4	미리 말할게 사과는 안 해 아무 말 없이 너 후회 안 해? 다 관심 없잖아 친구야 뭐야 Oh you know what to do	사랑이 잘.jpg	/img/4d91a165-4cf4-4b82-822b-231c05850367.jpg	사랑이 잘
5	세상의 모서리 구부정하게 커버린 골칫거리 outsider	celebrity.jpg	/img/3cb2364d-c7c1-4d52-9bbf-225324f26f62.jpg	Celebrity
6	누구를 위해 누군가 기도하고 있나 봐 숨죽여 쓴 사랑시가 낮게 들리는 듯해	love poem.jpg	/img/93507d0c-6953-43f7-a4c7-5e362b716712.jpg	Love Poem

(5 rows, 1 ms)

추가적으로 삭제된 id가 있다면 해당 id부터 다시 채우는 것이 아닌 다음 id부터 이어서 작성하도록 하였다. 1번 id는 사용과정에서 삭제하였기에 보이지 않는다.

4. Consideration

이번 소프트웨어프로젝트1의 Project 1은 Spring Boot를 통해서 이미지 업로드가 가능한 게시판 Web server를 구현을 하였다. 지난 소프트웨어프로젝트 실습 시간 때 설계된 게시판 작성 예시를 활용하여 기본적으로 게시글 upload, 수정, 삭제, 그리고 display에 대해 참고하여 제작하였으며, 해당 강의자료의 예시는 create.html 과 update.html을 따로 두어 등록과 수정이 따로 html이 존재하였지만 이를 if와 unless를 통해 해당 board.id가 비어있다면 등록 있다면 수정을 하도록 분리하였다.

이미지 업로드를 구현하기 전에 static 폴더에 미리 사진을 두어 해당 사진을 기준으로 먼저 html과 Controller, Service 등을 구현하였고 index, ImageView, Upload에 대해 업로드를 제외하고 정상 동작함을 확인하고 업로드를 구현하도록 했다. 이후 transferTo 함수를 통해 static 폴더에 저장됨을 확인하였다. 그래서 이 사진

을 바로 불러올 수 있도록 html에서 `${board.filePath}`를 통해 가져오려고 했지만 가져올 수 없었다. 처음에는 path가 잘못되어 사진을 못 가져오는 것이라 생각하여 경로를 계속 수정하였지만 계속 고쳐지지 않았다. 이에 대해 확인해보니 src 폴더의 하위에 있는 static에 추가할 경우 build 이후에 추가된 파일은 확인할 수 없음을 확인하였고, build 이후에 파일을 가져오는 build 폴더의 하위에 있는 static에 추가하여 build 된 이후에도 추가된 파일을 읽고 가져올 수 있도록 하였다.