

## 2025년도 SW프레임워크 기말고사

※ 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
- 메신저 등으로 응시기간내 어떤 소통이라도 하는 행위
- 금지된 자료(인터넷 웹, AI 등) 사용.
- 강의자료는 활용 가능
- 대리 시험 응시

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)

### 시험 진행 방식

- 시험은 100% 실습으로 진행됩니다.
- 시험 시간 30분 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 90분입니다.
- 시험이 시작되면 2페이지의 개발 진행 방법을 확인하고 하나씩 진행하시면 됩니다.
- 모든 단계를 완료하게 되면 하나의 웹 SW가 완성됩니다.
- 단계별로 점수가 () 안에 기재되어 있습니다.
- 시험이 종료된 학생들도 시험 제출 안내가 있으니 퇴실하시면 안됩니다.
- 총점 40점으로 기말고사 평가 기준 40%에 그대로 반영됩니다.

### ○ 중요 정보

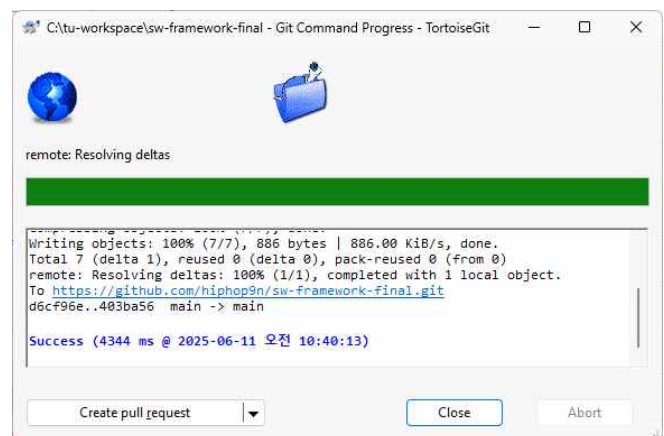
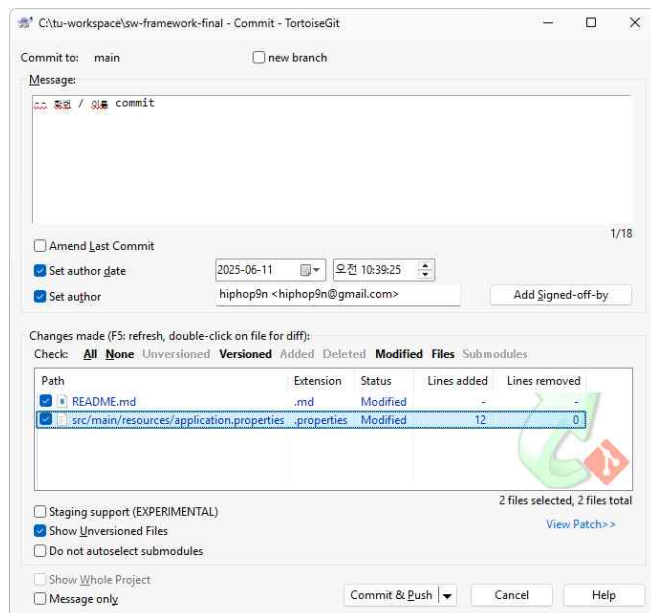
- <https://github.com/jongpil0514-tu/sw-framework-final>

- git에서 본인 repository로 Fork 후에 git clone 해서 자신의 PC에 소스를 다운로드 받은 후에 STS4에 gradle project로 Import 하세요
- “e-class > 과제” 메뉴로 이동하여 기말고사 제출 메뉴에 캡처 2개, 소스 2개 업로드 예정 (업로드 시간은 시험 종료 후 따로 부여할 것)

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

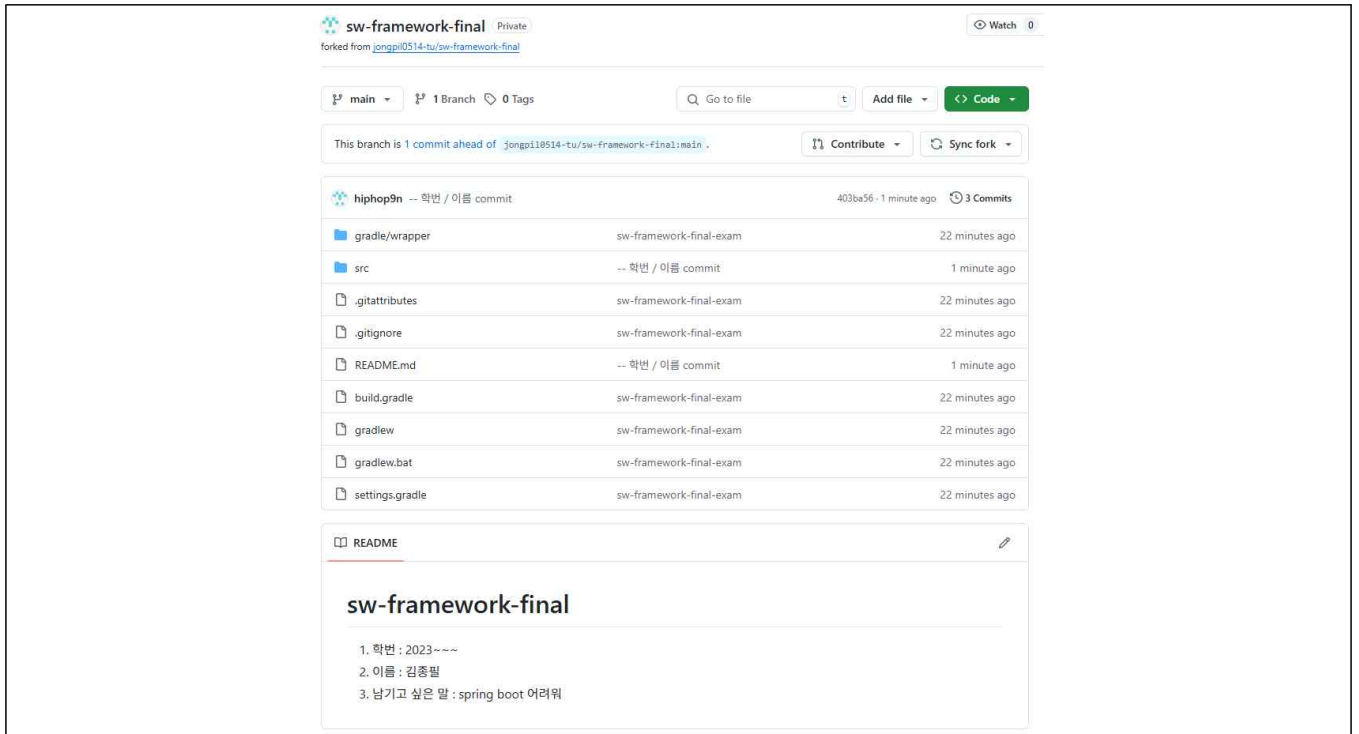
### - 부분점수 없음

- 1) 위 URL에 Git Fork를 통해 본인의 github로 Repository를 복사  
: <https://github.com/jongpil0514-tu/sw-framework-final>
- 2) fork한 레포지토리로 이동 후 자신의 Repository의 Code 버튼을 눌러서 Git URL을 복사  
: 본인 계정의 repository의 URL을 복사했는지 꼭 확인
- 3) 자신의 개발 환경 내에 local 폴더에 git clone을 통해 소스 다운로드
- 4) STS4에 소스 Import
- 5) application.properties를 수정하여 thymeleaf, database, mybatis 설정을 완성한다.
- 6) README.md 에 자신의 학번과 이름을 작성
- 7) tortoiseGit 혹은 git bash를 통해 수정한 README.md 파일을 git commit & push



- 8) 자신의 repository에 수정된 README.md가 정상으로 올라가 있으면 아래와 같이 github repository main이 자신이 작성한 학번과 이름으로 변경되어 있을 것!

- 정상적으로 commit & push가 안되는 학생은 일단 시험 진행할 것! 시험지 마지막에 대체 업로드 방법 서술되어 있음.



## Step 2. DataBase 설정 (2점)

### - 부분점수 없음

2.1 아래와 같이 테이블을 생성하시오.

사용 database : book(아이디, 비밀번호는 자신의 계정정보)

생성 테이블 이름 : tb\_users

Properties

Data

엔티티 관계도

테이블명:

tb\_users

Engine:

InnoDB

Auto Increment:

1

Charset:

utf8mb4

Collation:

utf8mb4\_general\_ci

Description:

Partitioned

Columns

Constraints

Foreign Keys

References

Triggers

Indexes

Partitions

Statistics

DDL

Virtual

컬럼명	#	Data Type	Not Null	Auto Increment	Key	디폴트	Extra	Comment
123 user_seq	1	int(10) unsigned	[v]	[v]	PRI		auto_increment	회원순번
A-Z user_id	2	varchar(100)	[ ]	[ ]		NULL		회원아이디
A-Z user_pw	3	varchar(500)	[v]	[ ]				비밀번호(sha256)
A-Z user_nm	4	varchar(100)	[ ]	[ ]		NULL		회원명
A-Z user_email	5	varchar(100)	[ ]	[ ]		NULL		회원이메일
reg_dt	6	datetime	[ ]	[ ]		NULL		등록일시
mod_dt	7	datetime	[ ]	[ ]		NULL		수정일시

2.2. 그림과 같이 출력될 수 있도록 샘플 데이터를 11개를 삽입한다.

- 비밀번호 원문은 "1234", 비밀번호는 sha2('1234', 256) 함수를 사용하여 sha-256으로 변환하여 저장
- reg\_dt, mod\_dt는 NOW()를 사용하여 입력
- 아이디, 이메일은 겹치지 않도록 입력
- 첫 번째는 본인 이름 입력, 나머지는 자신의 이름을 기준으로 "성"만 바꿔서 등록
  - ex) 김종필, 이종필, 박종필, 최종필, ...

- dbeaver에서 `Select * FROM tb_users` SQL을 실행하여 아래와 같이 결과를 출력하여 캡처
- “e-class > 과제” 메뉴로 이동하여 기말고사 제출에 캡처 이미지를 생성하여 시험 종료 후 업로드

	123 user_seq	A/Z user_id	A/Z user_pw	A/Z user_nm	A/Z user_email	reg_dt	mod_dt
1	1	final-exam-1	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	김학생	final-exam-1@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
2	2	final-exam-2	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	이학생	final-exam-2@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
3	3	final-exam-3	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	박학생	final-exam-3@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
4	4	final-exam-4	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	최학생	final-exam-4@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
5	5	final-exam-5	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	강학생	final-exam-5@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
6	6	final-exam-6	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	김학생	final-exam-6@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
7	7	final-exam-7	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	이학생	final-exam-7@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
8	8	final-exam-8	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	박학생	final-exam-8@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
9	9	final-exam-9	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	최학생	final-exam-9@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
10	10	final-exam-10	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	강학생	final-exam-10@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
11	11	final-exam-11	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4	고학생	final-exam-11@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000

### Step 3. 프로그램 구동 (3점)

#### - 부분점수 없음

- boot application이 구동된 후 `System.out.println`을 통해 “기말고사 프로젝트 생성 성공” 메시지를 띄우시오.

```

Problems @ Javadoc Declaration Search Console X Gradle Tasks Gradle Executions
final-exam - FinalExamApplication [Spring Boot App] C:\spring-tool-4\sts-4.27.0.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.5
2025-06-09T20:24:37.016+09:00 INFO 30548 --- [final-exam] [ restartedMain] com.framework.FinalExamApplication : Start
2025-06-09T20:24:37.018+09:00 INFO 30548 --- [final-exam] [ restartedMain] com.framework.FinalExamApplication : No ac
2025-06-09T20:24:37.048+09:00 INFO 30548 --- [final-exam] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devto
2025-06-09T20:24:37.048+09:00 INFO 30548 --- [final-exam] [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For a
2025-06-09T20:24:37.478+09:00 WARN 30548 --- [final-exam] [ restartedMain] o.m.s.mapper.ClassPathMapperScanner : No My
2025-06-09T20:24:37.701+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomca
2025-06-09T20:24:37.712+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.apache.catalina.core.StandardService : Start
2025-06-09T20:24:37.713+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.apache.catalina.core.StandardEngine : Start
2025-06-09T20:24:37.744+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initi
2025-06-09T20:24:37.744+09:00 INFO 30548 --- [final-exam] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root
2025-06-09T20:24:37.935+09:00 WARN 30548 --- [final-exam] [ restartedMain] ion$DefaultTemplateResolverConfiguration : Canno
2025-06-09T20:24:38.071+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveR
2025-06-09T20:24:38.099+09:00 INFO 30548 --- [final-exam] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomca
2025-06-09T20:24:38.104+09:00 INFO 30548 --- [final-exam] [ restartedMain] com.framework.FinalExamApplication : Start
기말고사 프로젝트 생성 성공

```

### Step 4. HomeController 생성 (4점)

#### - 부분점수 없음

- `src/main/java/com/framework/home` 패키지 하위에 `HomeController.java`를 생성
- “`http://localhost:8080/home`”를 요청하면 “Hello, SW Framework!”를 출력하도록 코딩
- Lombok(`@Slf4j` 사용)을 사용하여 “메인페이지 요청” 로그를 INFO 레벨로 작성하시오.

```

o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
com.framework.home.HomeController : 메인페이지 요청

```

### Step 5. 회원가입 폼 만들기 (5점)

#### - 요구사항 충족 못할 때 마다 -1 점

- 회원 아이디, 비밀번호, 회원 이름, 회원 이메일을 입력 받는 회원가입 폼을 만드시오.



- 회원아이디, 회원 이름, 회원 이메일은 input type="text" 형태로 입력할 수 있도록 생성(필수)
  - 입력 아이디 : aaa
  - 입력 이메일 : exam-aaa@tukorea.ac.kr
  - 입력 비밀번호 : 5678
  - 입력 이름 : 프레임워크
  - 파라미터 이름 : userId, userPw, userNm, userEmail
- 비밀번호 input type="hidden"으로 입력할 수 있도록 생성
  - 비밀번호 DB 저장 값은 sha256 알고리즘을 통해 저장
- 등록일시(reg\_dt), 수정일시(mod\_dt)는 DB에서 NOW()를 사용하여 입력
- rest 요청을 사용하여 회원가입 요청(/user/request-register)
  - xhr, fetch 어떤 것을 사용해도 무방
- DB상 동일 아이디 중복 입력 검증(아이디를 "final-exam-1"로 입력했을 때 페이지에 "중복 된 아이디입니다." 출력)
- 회원가입 성공할 시 "회원가입 성공" 메시지 출력
- 회원 가입 성공 후 dbeaver에서 회원 가입 정보를 포함한 정보 캡처
  - Step 2에서 aaa 아이디 입력값이 정상적으로 출력되어 아래와 같이 총 12개의 row가 보이도록 캡처

	123 user_seq	A-Z user_id	A-Z user_pw	A-Z user_nm	A-Z user_email	reg_dt	mod_dt
1	1	final-exam-1	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	김학생	final-exam-1@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
2	2	final-exam-2	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	이학생	final-exam-2@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
3	3	final-exam-3	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	박학생	final-exam-3@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
4	4	final-exam-4	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	최학생	final-exam-4@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
5	5	final-exam-5	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	강학생	final-exam-5@tukorea.ac.kr	2025-06-09 19:52:23.000	2025-06-09 19:52:23.000
6	6	final-exam-6	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	김학생	final-exam-6@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
7	7	final-exam-7	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	이학생	final-exam-7@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
8	8	final-exam-8	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	박학생	final-exam-8@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
9	9	final-exam-9	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	최학생	final-exam-9@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
10	10	final-exam-10	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	강학생	final-exam-10@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
11	11	final-exam-11	03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e131	고학생	final-exam-11@tukorea.ac.kr	2025-06-09 21:40:13.000	2025-06-09 21:40:13.000
12	12	aaa	f8638b979b2f4f793ddb6dbd197e0ee25a7a6ea32b0ae22f5e3c	SW 프레임워크 개발고사	exam-aaa@tukorea.ac.kr	2025-06-10 11:11:26.000	2025-06-10 11:11:26.000

## Step 6. 회원 목록 출력 (10점)

### - 요구사항 충족 못할 때 마다 -1 점

- 회원 순번, 아이디, 이름, 이메일, 등록일시, 수정일시 출력
- 회원 아이디와 회원 이메일을 검색하는 목록 페이지 만들기
- <http://localhost:8080/user/list> 요청
- 한 페이지 당 4개씩 목록 출력(LIMIT 4)
- 페이징은 총 3 페이지 출력되어야 함.
  - 회원가입을 통해 만든 회원 정보가 3페이지에 존재해야 함.(아이디 "aaa")
  - thymeleaf 변수 이름 : USERS\_LIST(회원 목록), user\_info(개별 회원 변수)
    - ex) \${user\_info.user\_seq}, \${result.USERS\_LIST} 등으로 사용
- 검색 INPUT은 input type="text"로 생성. pageNum은 input type="hidden"으로 생성
  - 파라미터 이름 : searchUserId, searchUserEmail, userPageNum
- userPageNum, searchUserId, searchUserEmail 파라미터를 통해 검색 시 GET 방식으로 요청했을 경우, 검색조건과 페이지 번호를 설정하여 목록이 재현되어야 함
  - <http://localhost:8080/user/list?userPageNum=2> -> 2 페이지 목록 재현
- 회원가입을 통해 만든 아이디(aaa), 이메일(exam-aaa@tukorea.ac.kr)이 검색되어야 함.
  - <http://localhost:8080/user/list?searchUserId=aaa>
  - <http://localhost:8080/user/list?searchUserEmail=exam-aaa>
- css를 통한 디자인은 없어도 됨
- 회원별 상세 페이지를 "회원아이디"에 링크 추가
  - RestFul URL로 상세페이지 호출 : [http://localhost:8080/user/detail/{user\\_seq}](http://localhost:8080/user/detail/{user_seq}) 형태로 url 생성할 것

## Step 7. 로그인 폼 만들기 (5점)

### - 요구사항 충족 못할 때 마다 -1 점

- 회원 아이디, 비밀번호를 입력 받아 로그인 버튼을 클릭하면 로그인할 수 있는 폼을 만드시오.
- 아이디는 input type="text", 비밀번호는 input type="hidden"으로 입력
- rest 요청을 사용하여 로그인 요청(/user/request-login)
  - xhr, fetch 어떤 것을 사용해도 무방
- DB상 아이디 유효성 검증 (아이디를 "bbb"입력했을 경우 "존재하지 않은 정보입니다" 출력)
- DB상 비밀번호 유효성 검증(아이디를 "aaa", 비밀번호를 "1234"로 입력하면 "회원 정보가 잘못되었습니다" 출력)
- 로그인 성공시 페이지에 "로그인 성공" 메시지 출력

## Step 8. 회원 상세 페이지 만들기 (5점)

### - 요구사항 충족 못할 때 마다 -1 점

- 회원 아이디, 회원 이름, 회원 이메일, 등록일시, 수정일시를 출력하는 상세페이지 만들기
- 목록에서 클릭했을 때 RestFul URL을 통해 상세페이지로 전환
- 회원 아이디는 th:text로 출력
- 회원 이름, 회원 이메일 수정 가능하도록 input type="text"로 생성하여 DB 정보가 input box에 설정되어 있어야 함.
- 목록 버튼 눌렀을 때 목록 페이지로 이동할 수 있도록 버튼 추가

## Step 9. 회원 상세에서 삭제 기능 만들기 (5점)

### - 부분점수 없음

- 회원 정보를 삭제할 수 있는 기능 만들기(삭제 버튼 추가)
- 삭제 기능 정상 동작하여 DB 상 데이터가 삭제 기능 정상 구현이 확인되면 점수 획득
- user\_seq로 삭제 SQL 동작 되도록 작성할 것

## ※ 기말고사 과제 제출 및 평가

- 작업물을 "e-class > 과제 > 기말고사 제출" 업로드 해주세요.
  - 제출 시간 이후, 수정 업로드시에는 0점 처리 될 수 있으니 주의
- 제출할 작업물은 총 3개입니다.
  - 초기 회원 정보 생성 DB 정보 캡처 이미지(11개 row)
  - 회원가입 후 DB 정보 캡처 이미지(12개 row)
  - 전체 작업물 압축 파일(zip)
  - 압축은 폴더 전체를 "압축하기"로 zip 파일로 생성 후 업로드
  - ex) tu-workspace 하위에 final-exam 폴더에 작업했다면 final-exam 폴더 선택 후 압축하기
  - 파일명은 자신의 학번 이름.zip으로 해주세요.
- 채점환경에 DB는 위 요구사항과 동일하게 구성하여, 제출한 소스를 구동하여 채점 후 공지

### ○ github.com Commit & Push 가산점 (5점)

- 자신의 github repository 경로 꼭 기재하세요.
- git 소스 push 시간이 응시시간 이후인 소스가 1개라도 있을 경우 가산점 제외
- 압축 파일과 결과물이 다르면 가산점 제외
- <https://github.com/>\_\_\_\_\_