

대학교LMS 서비스

3조 홍한별 | 하지원 | 노영수



01 기획

- 프로젝트 소개
- 기획의도
- 일정

02 개발환경

- 개발환경 소개

03 DB 구성도

- ERD

04 기본환경설정

- config

05 시연 및 코드 설명

- 동영상 시연
- 중요 코드 설명

06 마무리

- 인원별 구현한 기능
- 아쉬운 점과 개선할 점

A photograph of a group of people in a meeting, with a blue overlay. The text '이' is prominently displayed in the center.

이

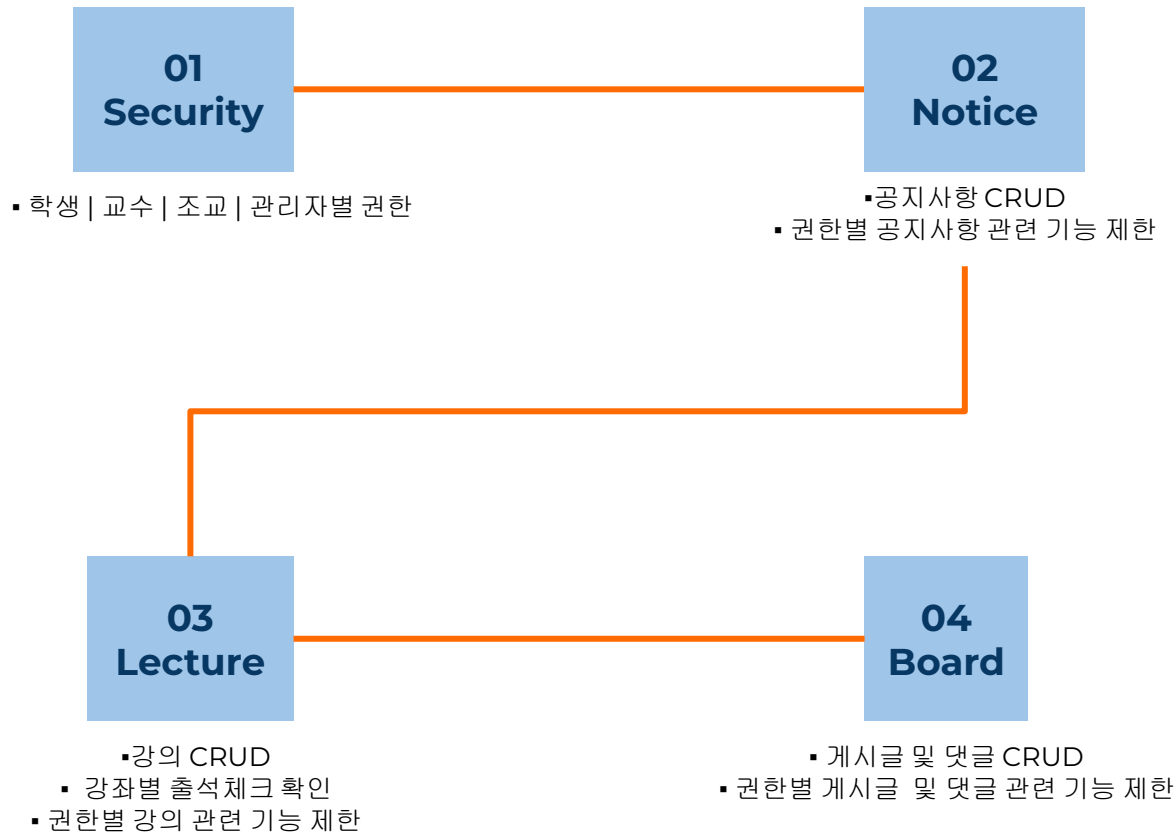
기획

프로젝트 소개

- 주제 : 대학교 LMS (Learning Management Service)
- 주제선정 이유
 1. 조원들이 학교 온라인동영상 Learning 사이트를 이용해 보았다는 점
 2. 2주간 Spring Framework수업을 통해 배운 hibernate와 security를 적용해보기 적합해보이는 주제를 찾았습니다.



기획의도

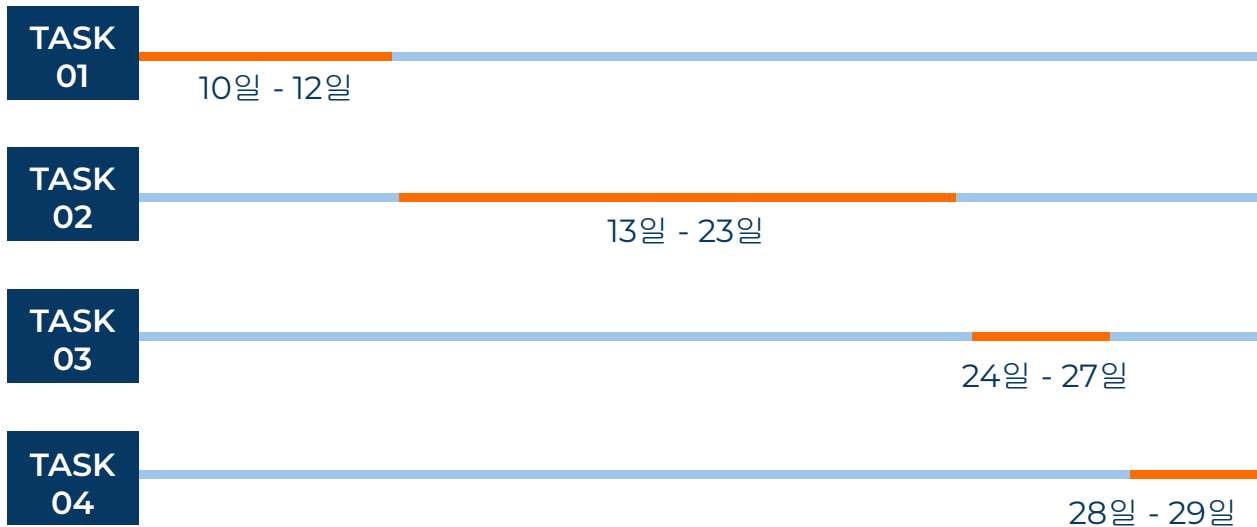


TASK01 : 프로젝트 구상 종합 | 프론트 기준 흐름 작성 | 함수 흐름도 작성 | 데이터베이스 구상

TASK02 : 코드 작성 및 수정 | 데이터베이스 수정

TASK03 : **CSS** 적용 | 추가 기능 구현 | 코드 오류 점검

TASK04 : 기능별 파일 정리 | **ppt** 작성 | 최종 점검



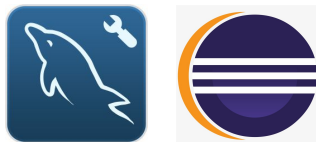


02

개발환경

개발환경

도구



Eclipse IDE 4.25.0 | MySQL Workbench 8.0.13

언어



Spring Framework | SQL | Java | Javascript | JQuery | Bootstrap

서버



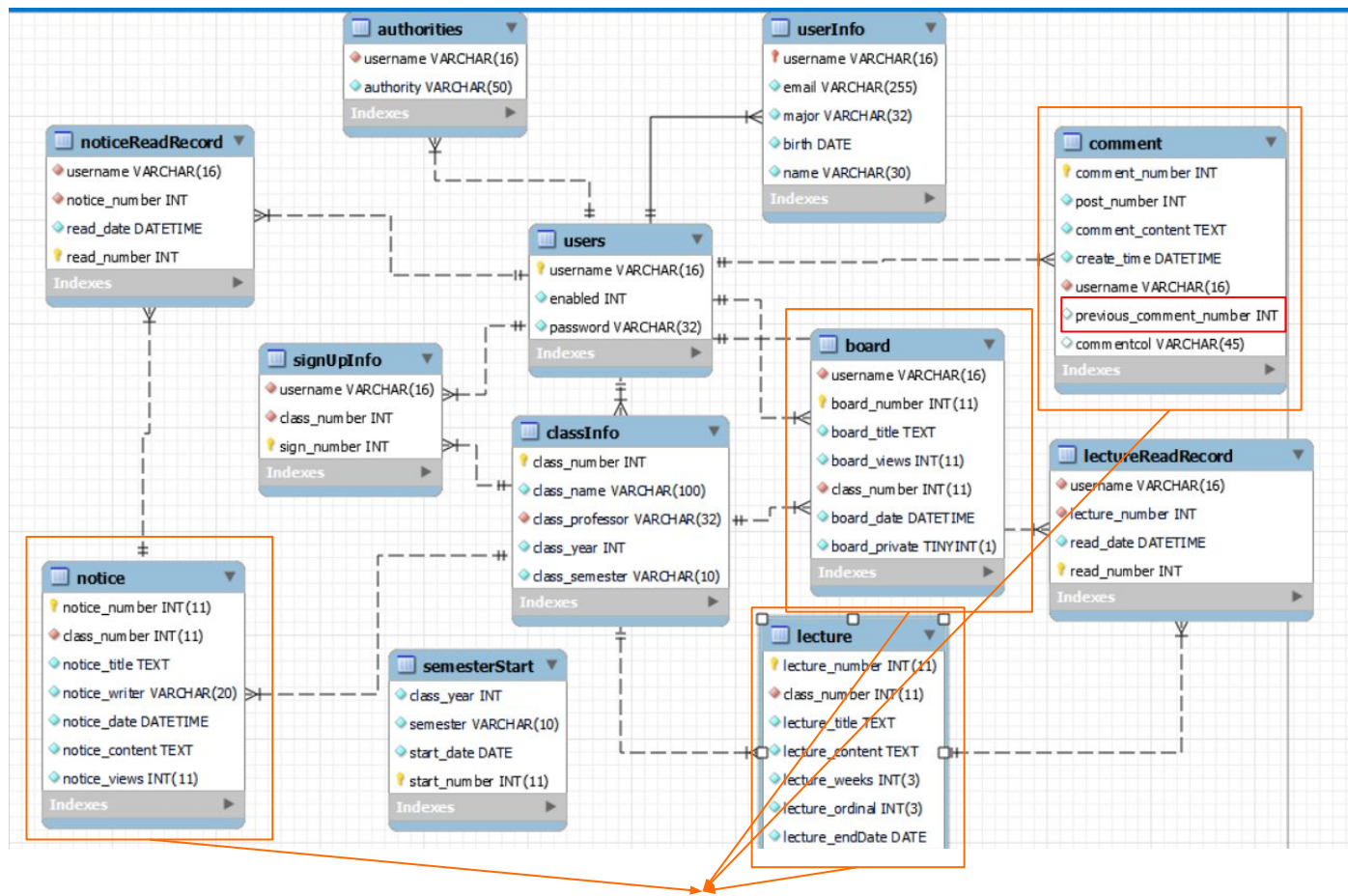
Apache Tomcat 9.0



03

DB 구성도

DB 구성도



Notice, Lecture, Board, Comment 테이블의 경우 삭제할 때 자신과 Foreign Key로 연동된 데이터들도 함께 삭제가 되도록 ON DELETE CASCADE 옵션으로 설정했고 나머지 table과 같은 경우에는 삭제가 없다고 가정했기에 NO ACTION으로 설정했습니다.



04

기본 환경 설정

POM.XML

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  |
  <groupId>com.code</groupId>
  <artifactId>cinema-pr-demo</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>

  <name>eClass</name>

  <properties>
    <springframework.version>5.0.2.RELEASE</springframework.version>
    <springsecurity.version>5.0.0.RELEASE</springsecurity.version>

    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- Spring MVC support -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${springframework.version}</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-web</artifactId>
      <version>${springframework.version}</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-config</artifactId>
      <version>${springframework.version}</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-taglibs</artifactId>
      <version>${springframework.version}</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.12</version>
    </dependency>

    <dependency>
      <groupId>com.mchange</groupId>
      <artifactId>c3p0</artifactId>
      <version>0.9.5.2</version>
    </dependency>

    <!-- Servlet, JSP and JSTL support -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
    </dependency>

    <dependency>
      <groupId>javax.servlet.jsp</groupId>
      <artifactId>javax.servlet.jsp-api</artifactId>
      <version>2.3.1</version>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>

    <!-- to compensate for java 9+ not including jaxb -->
    <dependency>
      <groupId>javax.xml.bind</groupId>
      <artifactId>jaxb-api</artifactId>
      <version>2.3.0</version>
    </dependency>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- springorm -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>${springframework.version}</version>
    </dependency>

    <!-- hibernate -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.4.28.Final</version>
    </dependency>

    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.10.0</version>
    </dependency>
  </dependencies>

  <!-- TO DO: Add support for Maven WAR Plugin -->

  <build>
    <finalName>spring-security-demo</finalName>
    <pluginManagement>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.2.0</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```


Security Config

```
@Configuration // xml 파일이다.  
@EnableWebSecurity // 웹 보안을 활성화 하겠다.  
public class DemoSecurityConfig extends WebSecurityConfigurerAdapter{  
  
    @Autowired // 보안에서 쓰일 데이터 소스 자동 변 주입  
    private DataSource securityDataSource;  
  
    @Autowired  
    private EclassService eclassService;  
  
    @Override // 사용자 세부 서비스를 설정하기 위한 오버라이딩  
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
  
        auth.jdbcAuthentication().dataSource(securityDataSource);  
  
    }  
  
    @Override // 인터셉터로 요청을 안전하게 보호하는 방법을 설정하기 위한 오버라이딩  
    protected void configure(HttpSecurity http) throws Exception {  
  
        http  
            .authorizeRequests()  
                .antMatchers("/css/**").permitAll()  
                .antMatchers("/img/**").permitAll()  
                .antMatchers("/admin/**").hasRole("ADMIN")  
                .antMatchers("/mypage").authenticated()  
                .antMatchers("/courses/**").authenticated()  
                .anyRequest().permitAll()  
            .and()  
            .formLogin()  
            .loginPage("/login")  
  
            .loginProcessingUrl("/authenticateTheUser")  
            .successHandler( // 로그인 성공 후 핸들러  
                new AuthenticationSuccessHandler() { // 익명 객체 사용  
                    @Override  
                    public void onAuthenticationSuccess(HttpServletRequest request,  
                        if(!eclassService.confirmPassword(authentication.getName()))  
  
                        response.sendRedirect("/eclass/");  
                    }else {  
                        response.sendRedirect("/eclass/passwordChange");  
                    }  
                }  
            )  
            .permitAll()  
            .and()  
            .logout().logoutSuccessUrl("/")  
            .permitAll()  
            .and()  
            .exceptionHandling().accessDeniedPage("/access-denied");  
  
        CharacterEncodingFilter filter = new CharacterEncodingFilter();  
        filter.setEncoding("UTF-8");  
        filter.setForceEncoding(true);  
        http.addFilterBefore(filter, CsrfFilter.class);  
  
    }  
}
```

로그인 성공 후 비밀번호가 초기 설정
비밀번호인지 확인하여 비밀번호 변경을
권유하기 위한 설정

URL에 따른 접근가능 여부 설정

POST 통신에서의 인코딩형식 설정

}

Error Controller

```
1 import org.springframework.http.HttpStatus;
2
3 @ControllerAdvice
4 public class ErrorController {
5
6     @ExceptionHandler(Exception.class)
7     public String handleException(Exception e, Model model) {
8
9         model.addAttribute("message", "에러가 발생하였습니다 잠시후에 다시 시도해주시기 바랍니다.");
10        System.out.println("에러메세지 : " + e.getMessage());
11        return "error";
12    }
13
14    @ExceptionHandler(NoHandlerFoundException.class)
15    @ResponseStatus(value=HttpStatus.NOT_FOUND)
16    public String handle404Exception(Model model) {
17
18        model.addAttribute("message", "해당 페이지를 찾을 수 없습니다 잠시후에 다시 시도해주시기 바랍니다.");
19        return "error";
20    }
21 }
```

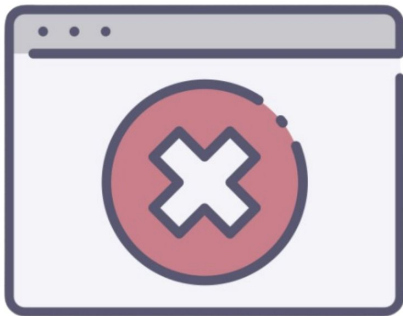
MySpringMvcDispatcherServletInitializer Class

```
@Override
protected void customizeRegistration(Dynamic registration) {
    // TODO Auto-generated method stub
    registration.setInitParameter("throwExceptionIfNoHandlerFound", "true");
}
```

Error.jsp 페이지

```
<body>

<div>
<h1>${message}</h1>
<a href="${pageContext.request.contextPath}/">첫화면으로 돌아가기</a>
</div>
```



해당 페이지를 찾을 수 없습니다 잠시후에 다시 시도해주시기 바랍니다.

[첫화면으로 돌아가기](#)

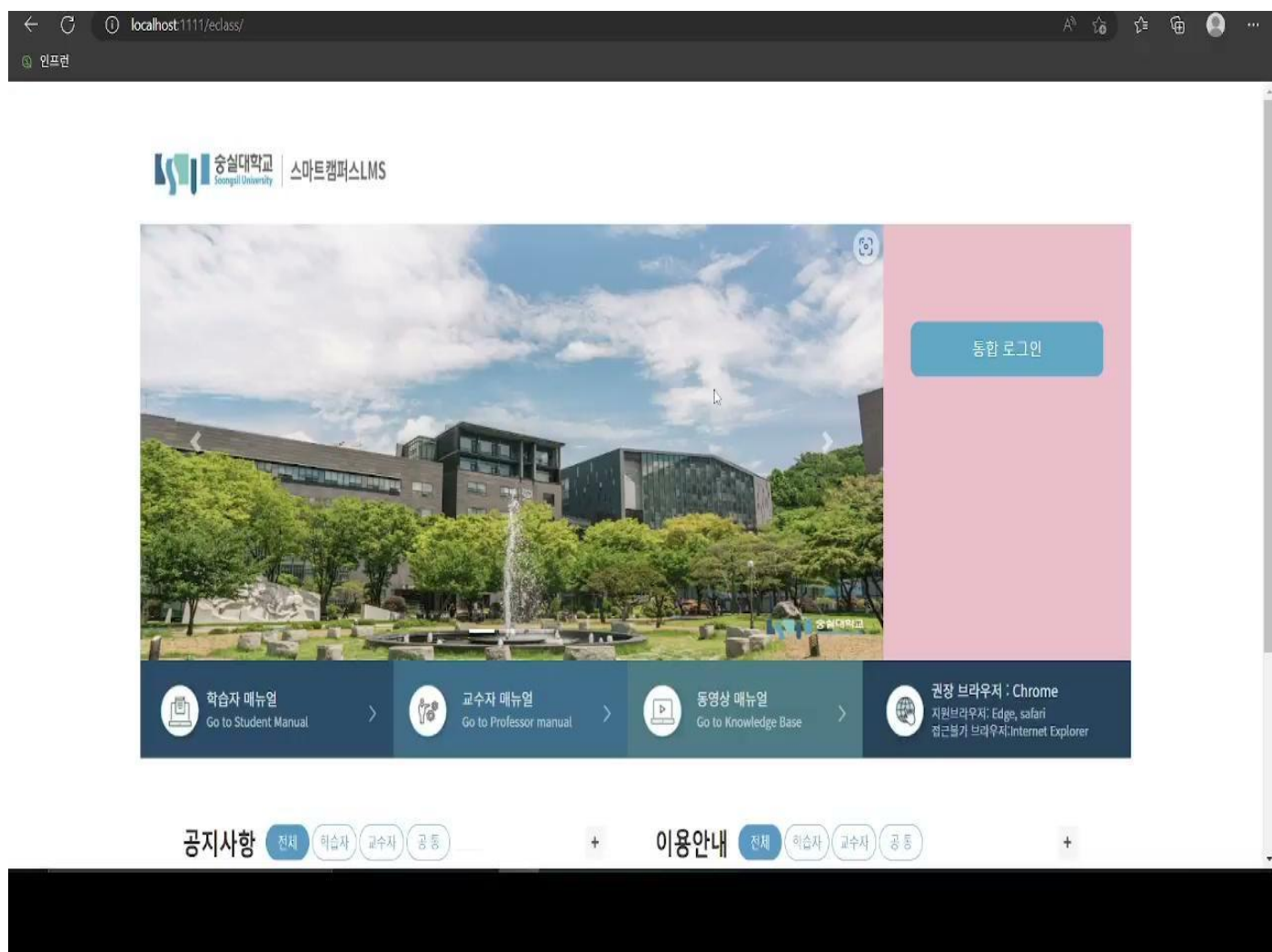


05

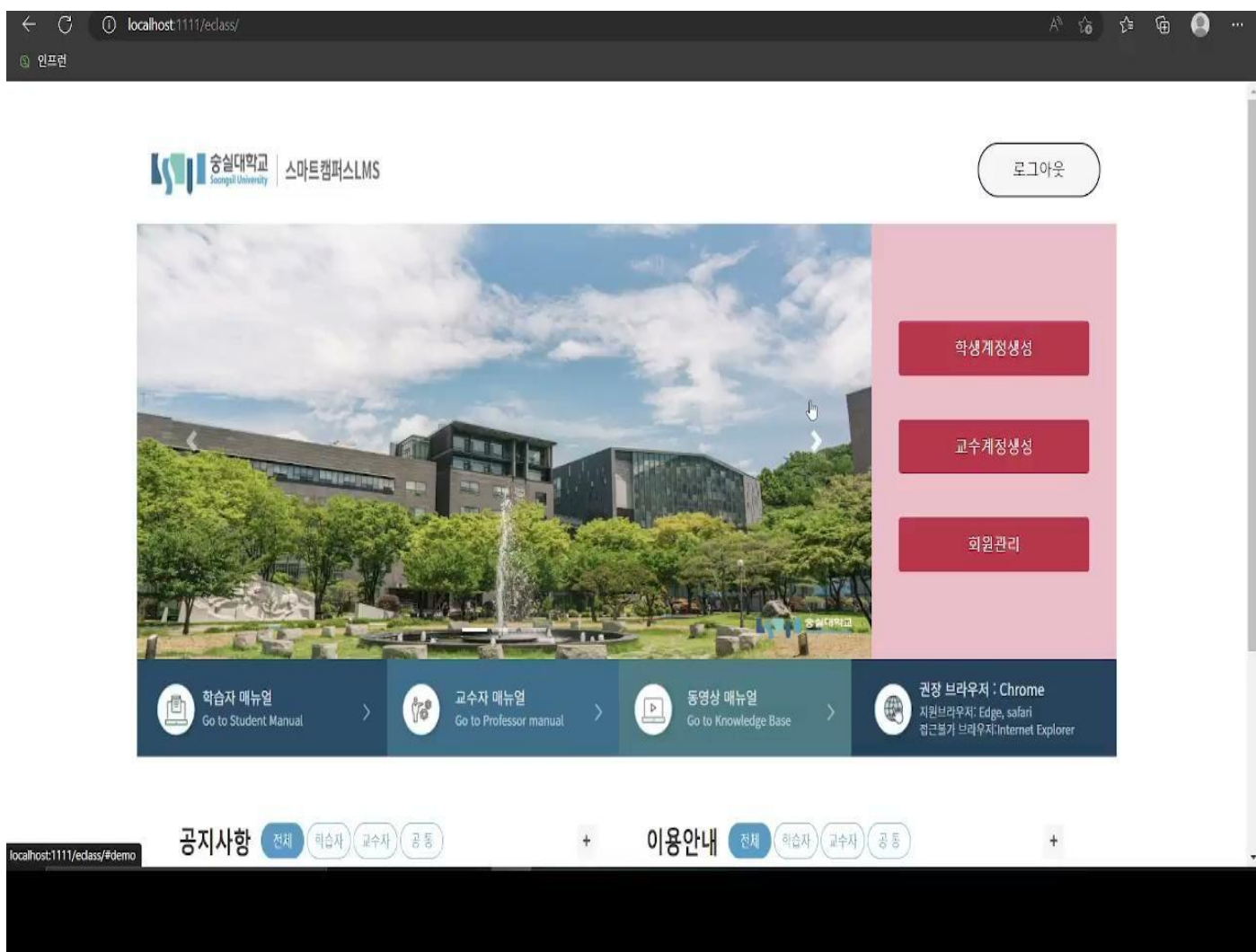
시연 및
코드 설명

홈 페이지 기능 시연

(관리자가 아닌 경우의 기능)



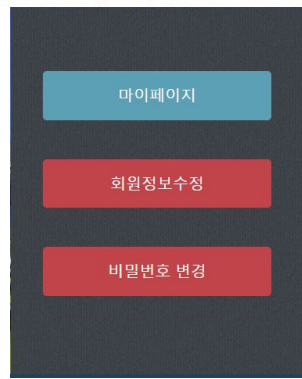
홈 페이지 기능 시연 (관리자 기능)





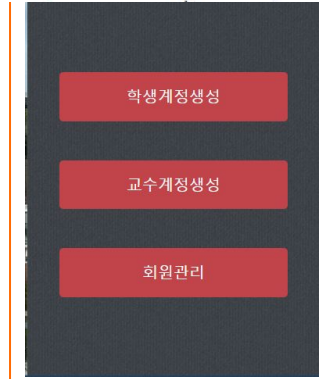
로그인한 아이디의 권한이 **Admin**이 아닌 경우

```
<security:authorize access="hasAnyRole('PROFESSOR', 'STUDENT', 'ASSISTANT')">
```

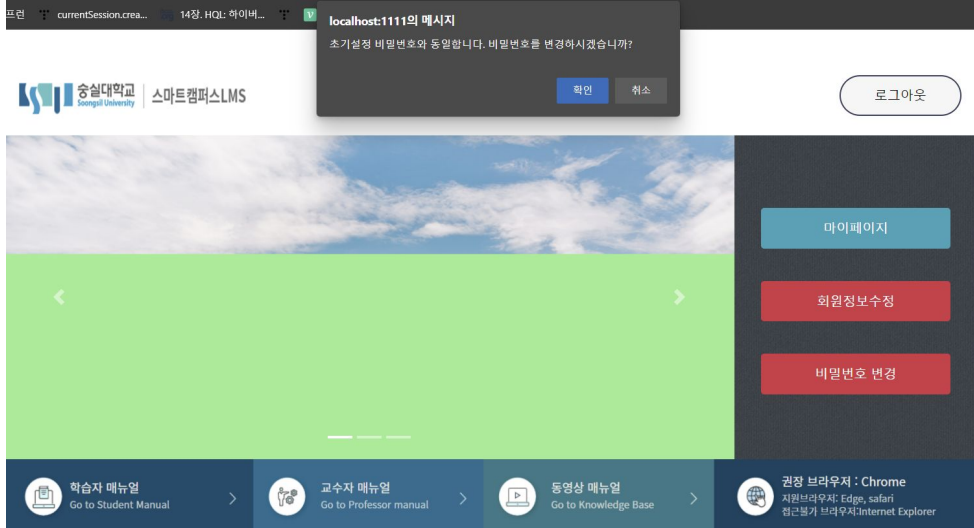


로그인한 아이디의 권한이 **Admin**인 경우

```
<security:authorize access="hasRole('ADMIN')">
```



홈 페이지 Login SuccessHandler



비밀번호 변경

새로운 비밀번호

새 비밀번호 확인

변경 취소

새로운 비밀번호

새 비밀번호 확인

입력한 두 비밀번호가 동일합니다.

입력한 두 비밀번호가 동일하지 않습니다. 다른 비밀번호를 입력해 주세요.

이전 비밀번호와 동일합니다. 다른 비밀번호를 입력해 주세요.

홈 페이지 Login SuccessHandler

```
let confirmSubmitCheck = '';  
function confirmSubmit(){
```

비밀번호 변경 버튼의 submit() 을 조절하기 위한 변수

```
    if(confirmSubmitCheck){  
        const form = document.querySelector('#passwordChangeForm');  
        form.submit();  
    }  
}
```

확인하기 위한 input을 입력시 발생하는 이벤트

```
function confirmInputPassword(){  
  
    const passwordInput = document.querySelector('#changePassword');  
    const passwordInputConfirm = document.querySelector('#changePasswordConfrim');  
    const changeBtn = document.querySelector('.changePW');  
  
    const confirmMessage = document.querySelector('#confirmMessage');  
    if(passwordInput.value == passwordInputConfirm.value){  
  
        if('${userList.password}'.replace('{noop}', '' ) != passwordInput.value){  
            confirmMessage.innerText = '입력한 두 비밀번호가 동일합니다.'  
            confirmMessage.style.color="blue";  
            confirmSubmitCheck = 'OK';  
        }else{  
            confirmMessage.innerText = '이전 비밀번호와 동일합니다. 다른 비밀번호를 입력해 주세요.'  
            confirmMessage.style.color="red";  
            confirmSubmitCheck = '';  
        }  
    }else{  
  
        confirmMessage.innerText = '입력한 두 비밀번호가 동일하지 않습니다. 다른 비밀번호를 입력해 주세요.'  
        confirmMessage.style.color="red";  
        confirmSubmitCheck = '';  
    }  
}
```


마이페이지 기능 시연

localhost:1111/edclass/

인프런

숭실대학교 SmartCampus LMS

로그아웃



마이페이지

회원정보수정

비밀번호 변경

학습자 매뉴얼
Go to Student Manual

교수자 매뉴얼
Go to Professor manual

동영상 매뉴얼
Go to Knowledge Base

권장 브라우저 : Chrome
지원브라우저: Edge, safari
접근불가 브라우저: Internet Explorer

공지사항 전체 학습자 교수자 공통

이용안내 전체 학습자 교수자 공통

마이페이지

2023년 1학기 ▼
2023년 1학기
2022년 겨울학기
2022년 2학기
2022년 여름학기
2022년 1학기

학기 선택

수강신청한 과목의 학기 정보

자바의정석

2023-1학기 | 세종오교수님

자바의정석2

2023-1학기 | 세종오교수님

과목홈 바로가기

과목홈 바로가기

해당 학기의 수강신청된 과목 정보

해당 과목페이지로 이동하는 버튼

←

↻

localhost/1111/edass/courses/6

A⁺

...

인프런

숭실대학교

Sungshil University

스마트캠퍼스LMS

로그아웃

마이페이지

홈

공지사항

강의콘텐츠

게시판

출결 현황

최근 공지사항

<div> 중간고사 시험장 변경 안내 404 호에서 301로 변경되었습니다. 확인하면 주변... </div>	<div>게시일시</div> <div>2023년 01월 29일 22시 10분</div>
<div> 중간 고사 날짜 중간 고사는 돌아오는 주 수요일에 보겠습니다. </div>	<div>게시일시</div> <div>2023년 01월 29일 22시 10분</div>
<div> 팀 프로젝트 팀 발표 1조 : 홍한별 노영수 하지원 ... </div>	<div>게시일시</div> <div>2023년 01월 29일 22시 10분</div>
<div> 과제 #1 제출 기한 변경 제출 마감 기한이 3/10 에서 3/17로 일주일 연장... </div>	<div>게시일시</div> <div>2023년 01월 29일 22시 10분</div>
<div> 자바의 정석 첫 공지사항 안녕하세요 여러분 처음으로 공지사항을 작성하네요 감의... </div>	<div>게시일시</div> <div>2023년 01월 29일 22시 10분</div>

마·오·표·오·지



0-0



공지사항



강의 콘텐츠



게시판



총금 100원

공지사항

- 마이페이지
- 홈
- 공지사항
- 강의콘텐츠
- 게시판
- 출결 현황

글쓰기

공지사항 검색

안 읽은 공지만 보기

- | | | |
|---|---|-------------------------------|
|  | 중간고사 시험장 변경 안내
404 호에서 301로 변경되었습니다. 확인하면 주변 ... | 게시일시
2023년 01월 29일 22시 10분 |
|  | 중간 고사 날짜
중간 고사는 돌아오는 주 수요일에 보겠습니다. | 게시일시
2023년 01월 29일 22시 10분 |
|  | 팀 프로젝트 팀 발표
1조 : 홍한별 노영수 하지원 | 게시일시
2023년 01월 29일 22시 10분 |
|  | 과제 #1 제출 기한 변경
제출 마감 기한이 3/10 에서 3/17로 일주일 연장... | 게시일시
2023년 01월 29일 22시 10분 |
|  | 자바의 정석 첫 공지사항
안녕하세요 여러분 처음으로 공지사항을 작성하네요 강의... | 게시일시
2023년 01월 29일 22시 10분 |

권한이 조교이거나 교수이거나 조교인 경우에만 보이도록 설정

```
<security:authorize access="hasAnyRole('PROFESSOR','ASSISTANT')">
<form:form action="{pageContext.request.contextPath}/courses/notice/writeForm/{classNumber}" method="POST">
    <button id="writeNotice" type="submit" class="btn btn-secondary">글쓰기</button>
</form:form>
</security:authorize>
```


공지사항

- 안 읽은 공지 -

마이페이지

홈

공지사항

강의콘텐츠

게시판

출결 현황

글쓰기

공지사항 검색

안 읽은 공지만 보기



중간고사 시험장 변경 안내
404 호에서 301로 변경되었습니다. 확인하면 주변 ...

게시일시
2023년 01월 29일 22시 10분



중간 고사 날짜
중간 고사는 돌아오는 주 수요일에 보겠습니다.

게시일시
2023년 01월 29일 22시 10분



팀 프로젝트 팀 발표
1조 : 홍한별 노영수 하지원

게시일시
2023년 01월 29일 22시 10분



과제 #1 제출 기한 변경
제출 마감 기한이 3/10 에서 3/17로 일주일 연장...

게시일시
2023년 01월 29일 22시 10분



자바의 정석 첫 공지사항
안녕하세요 여러분 처음으로 공지사항을 작성하네요 강의...

게시일시
2023년 01월 29일 22시 10분

```
session.currentSession = session.data().getCurrentSession();
Query<Notice> theQuery =
    currentSession.createQuery(
        "from Notice as n where n.classNumber=:classNumber "
        + "AND NOT n.noticeNumber IN "
        + "(SELECT nrr.noticeNumber from NoticeReadRecord as nrr where nrr.userName=:userName)", Notice.class);
theQuery.setParameter("classNumber", classNumber);
theQuery.setParameter("userName", userName);
```

숭실대학교

Sungshil University

스마트캠퍼스LMS

로그아웃

Q&A 게시판

글쓰기

검색결과 191 개

학번 ▾
제출

No	제목	작성자	등록일시	조회수
171	2	20162326	01-20 03:39	0
172	3	20162326	01-20 03:39	0
173	4	20162326	01-20 03:39	0
174	5	20162326	01-20 03:39	0
175	6	20162326	01-20 03:39	0
176	7	20162326	01-20 03:39	0
177	8	20162326	01-20 03:39	0
178	12	20162326	01-20 03:39	0
179	asd	20162326	01-20 03:39	0
180	asd	20162326	01-19 18:24	3

이전
11 12 13 14 15 16 17 18 19 20

```
public class BoardPage {
```

```
    private boolean previous;  
    private boolean next;  
    private int start;  
    private int end;  
    private List<Board> boardList;  
    private String searchTaget;  
    private String searchContent;  
    private int searchCount;  
    private int pageNumber;  
    private int classNumber;
```

```
    public BoardPage() {  
        searchTaget="";  
        searchContent="";  
        pageNumber = 1;  
    }  
    public BoardPage(int classNumber) {  
        this();  
        this.classNumber=classNumber;  
    }  
}
```

```
    public void restSetting() {  
        if(pageNumber/10.0>1) previous = true;  
        if(Math.round((searchCount-1)/100) > Math.round((pageNumber-1)/10)) next = true;  
        start = ((pageNumber-1)/10)*10 +1;  
        end = next? start+9: (searchCount-1)/10+1;  
    }  
}
```

Q&A 게시판



강의콘텐츠



게시판



출결현황

파트캠퍼스LMS

로그아웃

Q&A 게시판

글쓰기

검색결과 208 개

검색어 입력

제출

No	제목	작성자	제목 내용 등록 이름 학번	조회수
1	sad	오세종	01-28 23:07	4
2	답 댓글 확인용22322	20162326	01-26 11:33	56
3	답 두번째 게시물	20162327	01-20 03:39	1
4	답 12312323213213	20162326	01-20 03:39	12
5	답 1	20162326	01-20 03:39	22
6	3	20162326	01-20 03:39	0
7	4	20162326	01-20 03:39	1
8	5	20162326	01-20 03:39	0
9	6	20162326	01-20 03:39	1
10	7	20162326	01-20 03:39	0

1 2 3 4 5 6 7 8 9 10 다음

2	🔒 댓글 확인용22322	20162326	01-26 11:33	56
3	🔒 두번째 게시글	20162327	01-20 03:39	1
4	🔒 12312323213213	20162326	01-20 03:39	12

비밀글에 대한 접근통제를 위해 학생인 권한인 사람인 경우에만 글쓴이와 비교하여 글쓴이일 경우 <td> 의 onclick 속성에 readPost() 함수를 적용해주었습니다.

```

<c:when test="${board.boardPrivate}">
  <security:authorize
    access="hasAnyRole('PROFESSOR','ASSISTANT')">
    <td onClick="javascript:readPost('${board.boardNumber}')"> 
    <c:choose>
      <c:when test="${p.username.equals(board.userName)}">
        <td onClick="javascript:readPost('${board.boardNumber}')"> 
      </c:otherwise>
    </c:choose>
  </security:authorize>
</c:when>

```

이번 주에 올라온 과제#1의 문제 1번에 대한 질문입니다.

작성자:20162327 등록일시:2023-01-29T17:56:40 조회수:8

이번 주에 올라온 과제#1의 문제 1번에 대한 질문입니다.

1 + 1 = 2 이라 나와 있는데 찰흙 하나 + 찰흙 하나 = 찰흙 하나 인데 왜 2인가요?

게시글을 볼때 띄어쓰기가 적용되도록 textarea 에 작성한 enter를
 로 변환하여 데이터에 저장하여 페이지에서 볼때 적용되도록 했습니다.

```
function formSubmit(){
    const form = document.querySelector('#writeForm');
    const textarea = document.querySelector('textarea');

    let input2 = document.createElement('input');
    input2.setAttribute('type', 'hidden');
    input2.setAttribute('name', 'boardContent');
    input2.setAttribute('value', textarea.value.replaceAll("\n", "<br>"));

    form.appendChild(input2);
    form.submit();
}
```

Q&A 게시판

게시글 수정 Q&A 게시판

글 제목

댓글 확인용22322

내용

가
나
다
라
마
바
사
아
자
차
카

☒비밀글 설정 (게시판 목록에 자물쇠로 표시되고 교수/조교와 작성자만 열람할 수 있습니다.)

앞의 내용과 다르게 수정할 경우 textarea 안에 데이터가 들어가게 되므로
 을 \r\n으로 변경하여 데이터를 넣어주었습니다.

```
window.onload = function(){  
  
    if(${post.boardPrivate == true}){  
        document.getElementById('checkbox').checked=true;  
    }  
    const textarea = document.querySelector('textarea');  
    textarea.innerHTML = textarea.innerHTML.replaceAll('&lt;br&gt;', '\r\n');  
}
```

Q&A 게시판 -댓글 Data 구조-

작성자 : 20162327	등록
1	답글
작성자 : 20162326 (작성자)	수정 삭제
12	답글
작성자 : 20162327	
3	답글
작성자 : 20162327	
5	답글
작성자 : 20162326 (작성자)	수정 삭제
1	답글

```
private List<List<Comment>> lectureDataComment (List<Comment> commentList){
    HashMap<Integer,Integer> locationMap = new HashMap<>();
    List<List<Comment>> lectureDataComment = new ArrayList<>();
    for(Comment comment: commentList) {
        Integer previousCommentNumber = comment.getPreviousCommentNumber();
        if(previousCommentNumber == null) {
            lectureDataComment.add(new ArrayList<>());
            lectureDataComment.get(lectureDataComment.size()-1).add(comment);
            locationMap.put(comment.getCommentNumber(),lectureDataComment.size()-1);
        }else {
            int location = locationMap.get(previousCommentNumber);
            lectureDataComment.get(location).add(comment);
            locationMap.put(comment.getCommentNumber(),location);
            locationMap.remove(location);
        }
    }
    return lectureDataComment;
}
```

DB에서 Comment Table 에 존재하는
previousCommentNumber와 해당
CommentNumber가 위치한 List의 위치를
저장하는 Map을 이용해서 데이터를 한차례
가공

Q&A 게시판 -댓글 삭제 수정-

```
@PostMapping("/commentDelete/{classNumber}")
public String commentDelete(@RequestParam("postNumber")int boardNumber,int commentNumber,@PathVariable int classNumber,Model theModel) {

    eclassService.deleteComment(commentNumber);

    theModel.addAttribute("url",new StringBuilder("/courses/board/read/").append(classNumber).toString());
    ArrayList<InputData> inputDataList = new ArrayList<>();
    inputDataList.add(new InputData("boardNumber",boardNumber+""));
    theModel.addAttribute("inputs",inputDataList);
    return "redirect";
}

@PostMapping("/commentUpdate/{classNumber}")
public String commentUpdate(Comment comment,@PathVariable int classNumber,RedirectAttributes redirectAttributes,Model theModel) {

    System.out.println(comment);
    int boardNumber = comment.getPostNumber();
    System.out.println("boardNumber : "+boardNumber);
    eclassService.updateComment(comment);

    theModel.addAttribute("url",new StringBuilder("/courses/board/read/").append(classNumber).toString());
    ArrayList<InputData> inputDataList = new ArrayList<>();
    inputDataList.add(new InputData("boardNumber",boardNumber+""));
    theModel.addAttribute("inputs",inputDataList);
    return "redirect";
}
```

```
<script>

    window.onload = function(){

        const form = document.querySelector('form');

        form.submit();

    }

</script>
```

Redirect.jsp Page

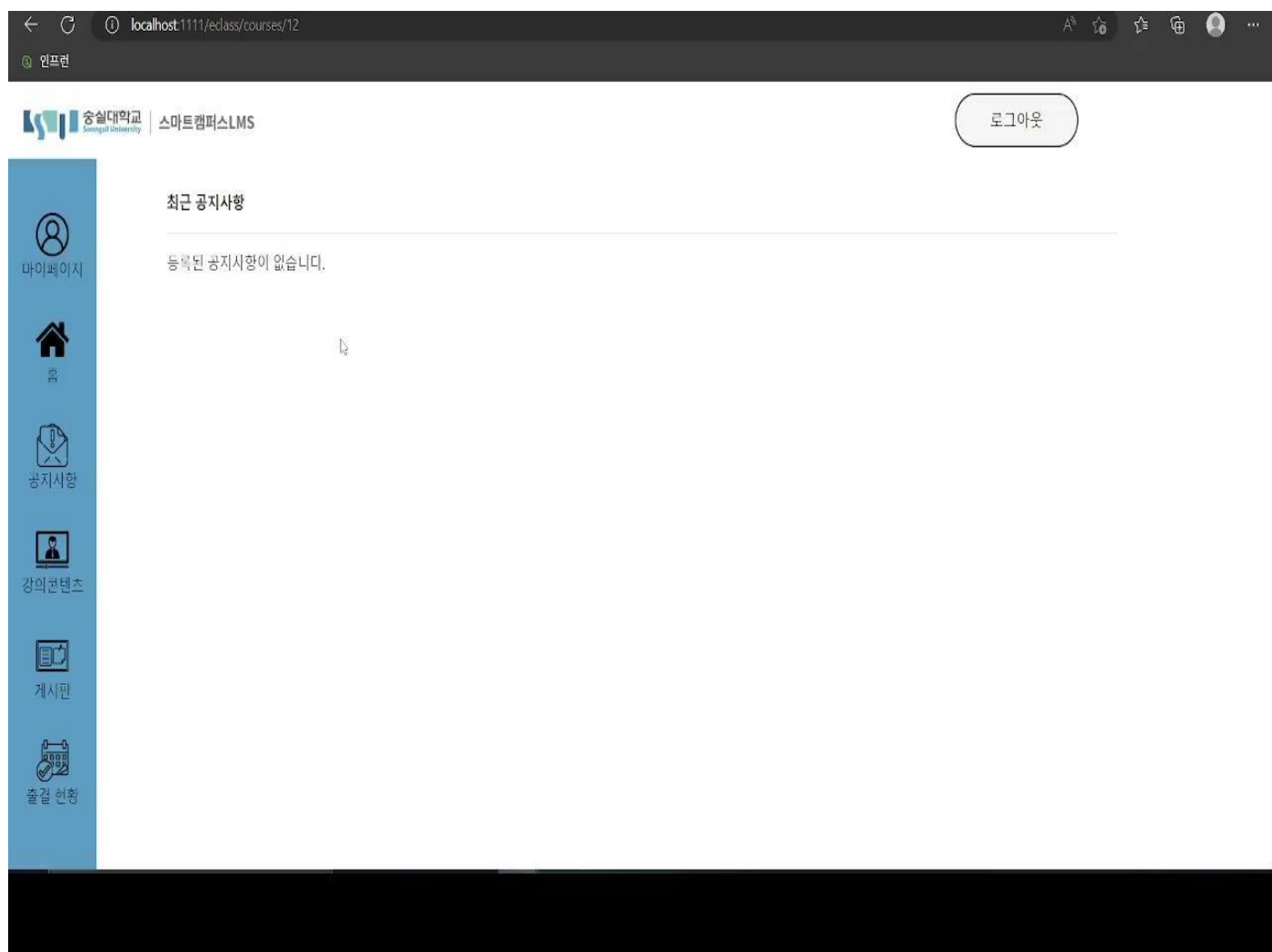
```
</script>
</head>
<body>

    <form:form action="${pageContext.request.contextPath}${url}" method="POST">
        <c:forEach var="input" items="${inputs}">
            <input type="hidden" name="${input.inputName}" value="${input.inputValue}"/>
        </c:forEach>
    </form:form>

</body>
```

댓글 삭제 수정시 Post 통신으로 이동할 수 있는 해당 게시글로 이동할 수 있도록 위와 같이 설정

Lecture 기능 시연



Lecture

충실대학교
Chungshil University

스마트캠퍼스LMS

로그아웃

마이페이지

홈

공지사항

강의콘텐츠

게시판

출결 현황

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

▼ 모든 메뉴 열기

▼ 7주차

▼ 8주차

▲ 9주차

1차시

9주차 1회차

2차시

9주차 2회차

▼ 10주차

▼ 11주차

▼ 12주차

<c:if test="\${username eq (checkProf)}">

<input type="hidden" name="LectureWeeks" value="\${i}" />

<input type="hidden" name="LectureOrdinal" value="\${j}" />

\${j}차시

<button class="btn btn-secondary addLectureContent" type="submit">추가하기</button><hr>

</c:if>

추가하기

2023-02-02

추가하기

2023-02-02

추가하기 버튼의 경우 해당 과목의 교수인 경우에만 보이도록 설정



마감일 : 2023-03-16 00:00:00 | 출석 인정 기간 : 2023-03-02 00:00:00 ~ 2023-03-16 00:00:00

출석 인정 기간이 아닙니다.

sadasdasdsadasdsadasdsadasdsadasdsadasdsadsdddddddddd asd asd sad sad asd as das dasdasdasdsad sad asd asd asd

마

수정하기

삭제하기

글을 쓸 수 있는 사람도 해당 과목의 교수인 경우 나타나도록 설정했습니다.

Lecture

Lecture

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

▼ 모든 메뉴 열기

▲ 1 1주차

asd	2023-03-16
asdasd	2023-01-25
asd	2023-01-24
asd	2023-01-24
sad	2023-01-25

1 주차 2 차시 등록된 강의가 없습니다.

▲ 2 2주차

2 주차 1 차시 등록된 강의가 없습니다.

2sad	2023-01-31
------	------------

```
private List<List<List<Lecture>>> lectureDataConstruct (List<Lecture> lectureList){
    List<List<List<Lecture>>> structuredList = new ArrayList<List<Lecture>>>();
    for(int i = 0; i < lectureList.get(0).getLectureWeeks(); i++) {
        List<List<Lecture>> innerList = new ArrayList<List<Lecture>>();
        for(int j =0; j <2; j++) {
            if(i == lectureList.get(0).getLectureWeeks()-1 && j == lectureList.get(0).getLectureOrdinal() ) break;
            innerList.add(new ArrayList<Lecture>());
        }
        structuredList.add(innerList);
    }
    for(Lecture lecture : lectureList) {
        structuredList.get(lecture.getLectureWeeks()-1).get(lecture.getLectureOrdinal()-1).add(lecture);
    }
    return structuredList;
}
```

Front에서 데이터 출력을
편하게 하기위해 강의List
데이터를 한 차례 가공을
하여 Front로 전달

Lecture



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

▼ 모든 메뉴 열기

▲ 9 주차

1차시

추가하기

9주차 1회차

2023-02-02

2차시

추가하기

9주차 2회차

2023-02-02

```
public int todayWeeks(LocalDate startDate) {  
    LocalDate today = LocalDate.now();  
    int betweenWeeks = (int)ChronoUnit.WEEKS.between(startDate, today)+1;  
    return betweenWeeks > 15 ? 15 : betweenWeeks;  
}
```

강의 콘텐츠로 올 때 현재의 주차를 계산하여 넘겨주고 자바스크립트를 이용해서 해당 주차가 열리고 위치가 이동하도록 설정했습니다.

Lecture -attendance-



주차	차시	강의 제목	출석
1주차	1차시	1주차 1회차	결석
	2차시	1주차 2회차	결석
2주차	1차시	2주차 1회차	결석
	2차시	2주차 2회차	결석
3주차	1차시	3주차 1회차	결석
	2차시	3주차 2회차	결석
4주차	1차시	4주차 1회차	결석
	2차시	4주차 2회차	결석
5주차	1차시	5주차 1회차	결석
	2차시	5주차 2회차	결석
6주차	1차시	6주차 1회차	결석
	2차시	6주차 2회차	결석

@Override

public Map<Integer,String>

```
String sql = "select lecture_number, max("
    + "if( date_add(?,INTERVAL (lecture_weeks-1) WEEK) <= read_date AND read_date < lecture_endDate , 2 , "
    + " if( date_add(?,INTERVAL (lecture_weeks-1) week) > read_date OR "
    + " date_add(lecture_endDate,INTERVAL 1 WEEK) < read_date, 0 ,1))) as attendance from lecturereadrecord "
    + "natural join lecture where username = ? AND class_number = ? group by lecture_number";
```

```
int lectureNumber = rs.getInt("lecture_number");
int attendanceNumber = rs.getInt("attendance");
String attendance = attendanceNumber==0?"결석": attendanceNumber==1? "지각" : "출석";
```

```
AttendanceInfos.put(lectureNumber,attendance);
```

```
<td>${attendanceInfos.getDefault(lecture.lectureNumber, "결석")}</td>
```



06

마무리

아쉬운 점

- 공지사항 Q&A게시판, 강의 각각의 차이를 제대로 살리지 못했다.
(첨부 파일관련 및 동영상 재생에 따른 출석 여부)
- DB 구조를 처음에 제대로 잡지 않아 중간에 바꿔야 했던 부분
- 프로젝트 준비하는 공간이 학원이 아닌 각 자 집이 었던 부분
- 많은 학생 데이터를 엑셀 파일을 읽어 아이디 생성하는 부분을 포기하여
현재 학생 User 아이디를 만드는 경우 하나씩 만들어야 하는 부분
- 하이버네이트에 대한 이해의 부족으로 인한 구현제한

Thanks
