



# 대학교 포털사이트 구현하기 ver\_02

오은혜

01

기획의도

02

개발환경

03

**DB**설계

04

기능구현



01

기획의도

대학교 포털 사이트 구현하기  
ver\_01에서 개발환경을 바꾸어  
프로젝트를 제작하여  
각 기술의 차이점을 익히기 위해





02

개발환경





Spring  
Boot



1

공통

- 운영체제 : window 10
- 소스코드 작성 도구 : **IntelliJ**
- 웹 서버 : Apache Tomcat 9.0
- Build Tool : **Gradle**

2

Front-End

HTML5, CSS3, **Thymleaf**, JavaScript, J-Query

3

Back-End

JAVA, **Spring Boot**, JPA

4

DataBase

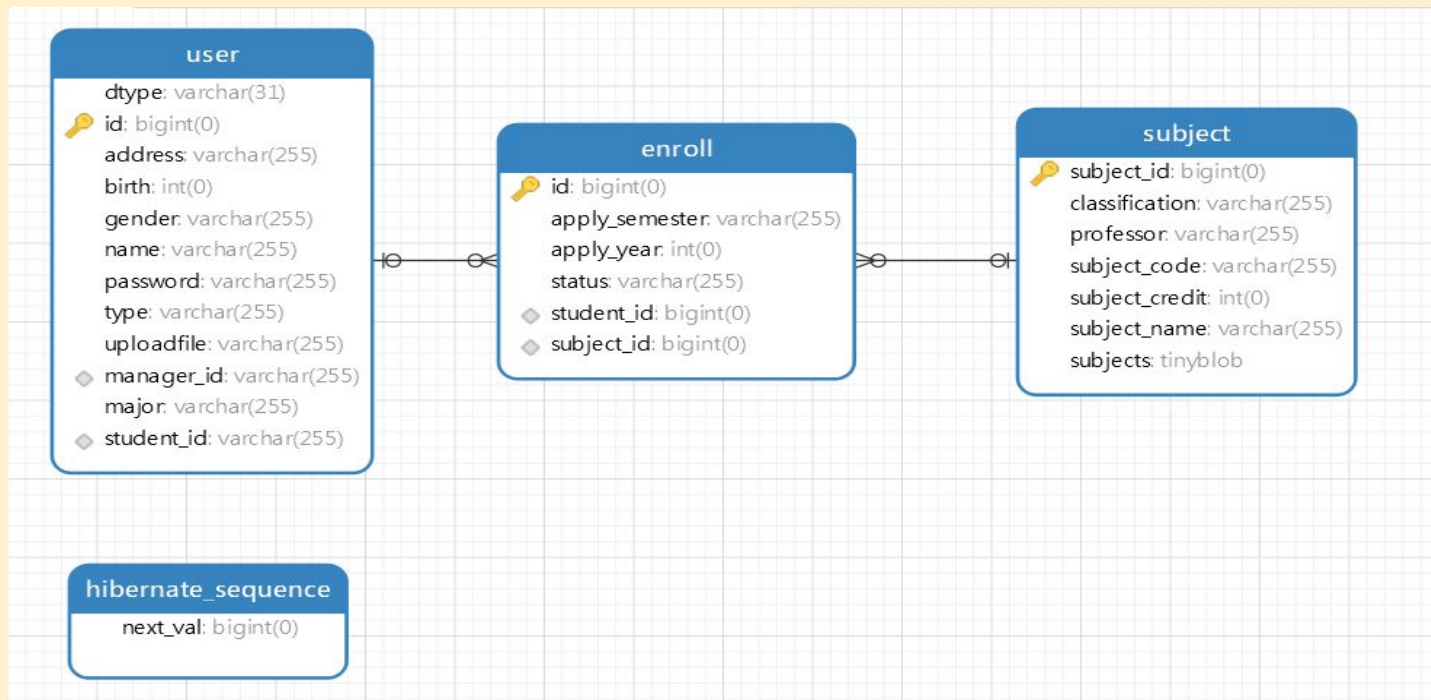
MySQL



03

DB설계

# DB 설계







## 04

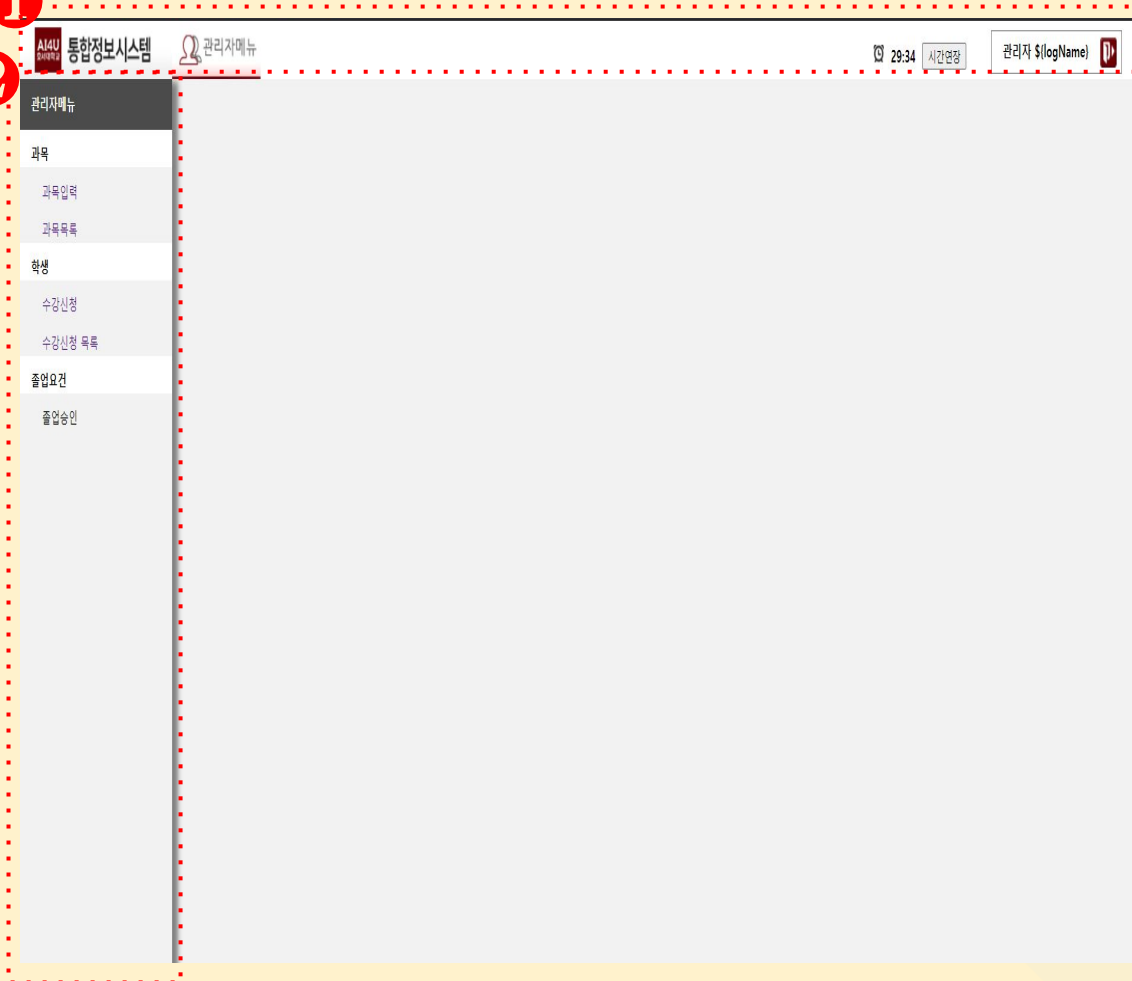
기능구현

# 요구 사항 정리\_관리자

구분	1depth	2depth	3depth	4depth	기능내용	비고
관리자_메인	과목입력				과목 입력 기능	
	과목목록	과목수정			입력한 과목 목록 확인 및 수정 기능	
	수강신청				수강 신청	
	수강신청목록	수강 취소			학번, 수강신청상태별 검색 및 수강 취소 기능	

1

2



## Description

1. Header fragment
2. Nav fragment

## 기술 상세\_managerLayout.html

```
<html lang="kr" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" href="/css/fragments/bodyHeader.css">
  <link rel="stylesheet" href="/css/fragments/nav.css">
  <link rel="stylesheet" href="/css/reset.css">
  <script src="/jqLib/jquery-3.2.1.min.js"></script>
  <script type="text/javascript"...>
</head>
<body>
  <div th:replace="fragments/manager/bodyHeader.html :: bodyHeader"></div>
  <section>
    <div th:replace="fragments/manager/nav.html :: nav"></div>
    <article>
    </article>
  </section>
</body>
</html>
```

managerLayout.html

## fragments

### fragments/manager/bodyHeader.html

```
<html xmlns:th="http://www.thymeleaf.org">
<header id="header" th:fragment="bodyHeader">
  <div id="logo_wrap">
    <div id="logo">
      
    </div>
    
  </div>
  <div id="out_wrap">
    <div id="timer">
      
      <span id="counter"></span><input type="button" value="시간연장" onclick="counter_reset()">
    </div>
    <div id="logout">
      <span>관리자 ${logName}</span>
      
    </div>
  </div>
</header>
```

### fragments/manager/nav.html

```
<html xmlns:th="http://www.thymeleaf.org">
<script th:src="@{/jqLib/jquery-3.2.1.min.js}"></script>
<nav id="nav" th:fragment="nav">
  <ul>관리자메뉴
    <li><span>과목</span>
      <ul>
        <li class="hover_b"><a th:href="@{/subjects/new}">과목입력</a></li>
        <li class="hover_b"><a th:href="@{/subjects}">과목목록</a></li>
      </ul>
    </li>
    <li>학생
      <ul>
        <li class="hover_b"><a th:href="@{/manager/enroll}">수강신청</a></li>
        <li class="hover_b"><a th:href="@{/manager/enrolls}">수강신청 목록</a></li>
      </ul>
    </li>
```

AI4U  
고려대학교

통합정보시스템

관리자메뉴

시간연장관리자 \$(logName)

관리자메뉴

과목

과목입력

과목목록

학생

수강신청

수강신청 목록

졸업요건

졸업승인

과목 입력

과목코드

과목명

학점

이수구분

담당교수

전송

2

[JPA] localhost:8080 내용: ring boot - Th 이미 존재하는 과목코드입니다. 확인

## Description

1. 과목 입력
2. alert()  
- 중복되는 과목코드 입력시



## 기술 상세\_과목입력

```
@GetMapping("/subjects/new")
public String createForm(){
    return "subjects/createSubjectForm";
}
```

```
@PostMapping("/subjects/new")
public String create(SubjectForm form, String subjectCode, Model model){
    if(subjectService.findBySubjectCode(subjectCode).isEmpty()) {
        Subject subject = new Subject();
        subject.setId(form.getId());
        subject.setSubjectCode(form.getSubjectCode());
        subject.setSubjectName(form.getSubjectName());
        subject.setSubjectCredit(form.getSubjectCredit());
        subject.setClassification(form.getClassification());
        subject.setProfessor(form.getProfessor());
        subjectService.saveSubject(subject);
        return "redirect:/subjects";
    }else{
        model.addAttribute( attributeName: "msg", attributeValue: "이미 존재하는 과목코드입니다.");
        return "subjects/createSubjectForm";
    }
}
```

SubjectController

## Repository & Service & view

### SubjectRepository

```
public void save(Subject subject){
    if(subject.getId() == null){
        em.persist(subject);
    }else{
        em.merge(subject);
    }
} //save
```

### SubjectService

```
@Transactional
public void saveSubject(Subject subject){
    subjectRepository.save(subject);
}
```

### createSubjectForm.html

```
<script th:inline="javascript">
    /*<![CDATA[*]
    var msg = [[${msg}]];
    if(msg !=null)
        alert(msg);
    /*]]>*/
</script>
```

관리자메뉴

과목

과목입력

과목목록

학생

수강신청

수강신청 목록

졸업요건

졸업승인

## 과목 목록

1

#	과목ID	과목명	학점	이수구분	담당교수	수정
137	A002	JPA	3	전공필수	제이피	<a href="#">수정</a>
140	C001	심리학	3	전공선택	심리2	<a href="#">수정</a>
141	A001	JAVA	3	전공필수	김자바	<a href="#">수정</a>
142	B001	엑셀	2	교양필수	엑셀	<a href="#">수정</a>
149	B002	파워포인트	2	교양필수	파워	<a href="#">수정</a>
157	C002	경영학개론	3	자유선택	경영쓰 수정2	<a href="#">수정</a>

2

## Description

1. 과목 목록
2. 과목 수정

## 기술 상세\_과목목록

```
<tbody>
<tr th:each="subject : ${subjects}">
    <td th:text="${subject.id}"></td>
    <td th:text="${subject.subjectCode}"></td>
    <td th:text="${subject.subjectName}"></td>
    <td th:text="${subject.subjectCredit}"></td>
    <td th:text="${subject.classification}"></td>
    <td th:text="${subject.professor}"></td>
    <td>
        <a href="#" th:href="@{/subjects/{id}/edit (id=${subject.id})}"
            role="button">수정</a>
    </td>
</tr>
</tbody>
```

subjectList.html

## Controller & Repository & Service

### SubjectController

```
@GetMapping("/subjects")
public String list(Model model, Subject subject){
    List<Subject> subjects = subjectService.findSubjects();
    model.addAttribute( attribute: "subjects", subjects);
    return "subjects/subjectList";
}
```

### SubjectRepository

```
public List<Subject> findAll(){
    return em.createQuery( qString: "select sb from Subject sb", Subject.class)
        .getResultList();
}
```

### SubjectService

```
public List<Subject> findSubjects(){
    return subjectRepository.findAll();
}
```

과목 수정 </subjects/157/edit>

과목ID C002

과목명	경영학개론
-----	-------

학점 3

이수구분 자유선택

담당교수    경영쓰\_수정3

전송

과목 목록 /subjects

#	과목ID	과목명	학점	이수구분	담당교수	수정
137	A002	JPA	3	전공필수	제이피	<a href="#">수정</a>
140	C001	심리학	3	전공선택	심리2	<a href="#">수정</a>
141	A001	JAVA	3	전공필수	김자바	<a href="#">수정</a>
142	B001	엑셀	2	교양필수	엑셀	<a href="#">수정</a>
149	B002	파이썬입문	2	교양필수	파워	<a href="#">수정</a>
157	C002	경영학개론	3	자유선택	경영쓰 수정4	<a href="#">수정</a>

## Description

1. 과목 수정 > 수정 완료 후  
과목 목록 출력

## 기술 상세\_과목수정

```
@GetMapping("/subjects/{subjectId}/edit")
public String updateSubjectForm(@PathVariable("subjectId") Long subjectId, Model model){
    Subject subject = subjectService.findOne(subjectId);

    SubjectForm form = new SubjectForm();
    form.setId(subject.getId());
    form.setSubjectCode(subject.getSubjectCode());
    form.setSubjectName(subject.getSubjectName());
    form.setSubjectCredit(subject.getSubjectCredit());
    form.setClassification(subject.getClassification());
    form.setProfessor(subject.getProfessor());

    model.addAttribute("form", form);
    return "subjects/updateSubjectForm";
}

@PostMapping("/subjects/{subjectId}/edit")
public String updateSubject(@ModelAttribute("form") SubjectForm form){
    Subject subject = new Subject();
    subject.setId(form.getId());
    subject.setSubjectCode(form.getSubjectCode());
    subject.setSubjectName(form.getSubjectName());
    subject.setSubjectCredit(form.getSubjectCredit());
    subject.setClassification(form.getClassification());
    subject.setProfessor(form.getProfessor());

    subjectService.saveSubject(subject);
    return "redirect:/subjects";
}
```

**SubjectController**

## Repository & Service & view

### updateSubjectForm.html

```
<article>
    <h1>과목 수정</h1>
    <form th:object="${form}" method="post">
        <input type="hidden" th:field="*{id}"/>
    <label>
        과목ID <input type="text" th:field="*{subjectCode}"><br>
    </label>
    <label>
        과목명 <input type="text" th:field="*{subjectName}"><br>
    </label>
</form>
```

### SubjectRepository

```
public Subject findOne(Long id){
    return em.find(Subject.class, id);
}

//findOne
```

### SubjectService

```
public Subject findOne(Long subjectId){
    return subjectRepository.findOne(subjectId);
}
```



관리자메뉴

과목

과목입력

과목목록

학생

수강신청

수강신청 목록

졸업요건

졸업승인

## 수강신청

1

학번 v 신청

2

#	과목코드	과목명	학점	이수구분	담당교수
<input type="checkbox"/>	A002	JPA	3	전공필수	제이피
<input type="checkbox"/>	C001	심리학	3	전공선택	심리2
<input type="checkbox"/>	A001	JAVA	3	전공필수	김자바
<input type="checkbox"/>	B001	엑셀	2	교양필수	엑셀
<input type="checkbox"/>	B002	파워포인트	2	교양필수	파워
<input type="checkbox"/>	C002	경영학개론	3	자유선택	경영쓰_수정4

## Description

1. 학번 선택 및 신청버튼
2. 과목 목록

## 기술 상세\_수강신청

```
@GetMapping("/manager/enroll")
public String createForm(Model model, Subject subject){
    List<Student> students = studentService.findStudents();
    List<Subject> subjects = subjectService.findSubjects();

    model.addAttribute(attributeName: "students", students);
    model.addAttribute(attributeName: "subjects", subjects);
    return "enroll/manager/enrollForm";
}

@PostMapping("/manager/enroll")
public String enroll(@RequestParam("studentId") Long studentId,
                    @RequestParam("subjects") Long subjects){
    enrollService.enroll(studentId, subjects);
    return "redirect:/manager/enrolls";
}
```

EnrollController

## Domain & Service

### Enroll

```
public static Enroll createEnroll(Student student, Subject subject){
    Enroll enroll = new Enroll();
    enroll.setStudent(student);
    enroll.setSubject(subject);
    enroll.setStatus(EnrollStatus.Enrolment);
    return enroll;
}
```

### EnrollService

```
@Transactional
public Long enroll(Long studentId, Long subjectId){

    //엔티티 조회
    Student student = studentService.findOne(studentId);
    Subject subject = subjectService.findOne(subjectId);

    //수강 생성
    Enroll enroll = Enroll.createEnroll(student, subject);

    //수강 저장
    enrollRepository.save(enroll);
    return enroll.getId();
}
```

관리자메뉴

과목

과목입력

과목목록

학생

수강신청

수강신청 목록

졸업요건

졸업승인

## 수강신청 목록

1

학번  수강신청상태  검색

2

#	학번	과목코드	과목명	학점	이수구분	담당교수	취소
150	12345	A002	JPA	3	전공필수	제이피	
151	12345	C001	심리학	3	전공선택	심리2	
152	12345	A001	JAVA	3	전공필수	김자바	<a href="#">CANCEL</a>
153	12345	B001	엑셀	2	교양필수	엑셀	<a href="#">CANCEL</a>
154	12345	B001	엑셀	2	교양필수	엑셀	<a href="#">CANCEL</a>
155	12345	C001	심리학	3	전공선택	심리2	<a href="#">CANCEL</a>
156	20120438	A002	JPA	3	전공필수	제이피	<a href="#">CANCEL</a>
158	20120438	B002	파워포인트	2	교양필수	파워	
159	20120438	B001	엑셀	2	교양필수	엑셀	
160	20120438	A002	JPA	3	전공필수	제이피	
161	20120438	A002	JPA	3	전공필수	제이피	<a href="#">CANCEL</a>

3

## Description

1. 학번, 수강신청상태 및 검색버튼
2. 수강신청 목록
3. 수강 취소 기능

## 기술 상세\_수강목록

```
<form th:object="{enrollSearch}" >
    <input type="text" th:field="*{studentId}" placeholder="학번"/>
    <select th:field="*{enrollStatus}">
        <option value="">수강신청상태</option>
        <option th:each="status : ${T(springboot.jpatest.domain.EnrollStatus).values()}"
            th:value="{status}"
            th:text="{status}">option
        </option>
    </select>
    <button type="submit">검색</button>
</form>

<tr th:each="enroll : ${enrolls}" >
    <td th:text="{enroll.id}"></td>
    <td th:text="{enroll.student.studentId}"></td>
    <td th:text="{enroll.subject.subjectCode}"></td>
    <td th:text="{enroll.subject.subjectName}"></td>
    <td th:text="{enroll.subject.subjectCredit}"></td>
    <td th:text="{enroll.subject.classification}"></td>
    <td th:text="{enroll.subject.professor}"></td>
    <td>
        <a th:if="{enroll.status.name() == 'Enrolment'}" href="#"
            th:href="'javascript:cancel(' + {enroll.id} + ')">CANCEL</a>
    </td>
</tr>
</tr>
```

## Controller & Repository

### EnrollController

```
@GetMapping("/manager/enrolls")
public String managerEnrolls(@ModelAttribute("enrollSearch") EnrollSearch enrollSearch, Model model) {
    List<Enroll> enrolls = enrollService.findEnrolls(enrollSearch);
    model.addAttribute("attributeName: enrolls", enrolls);
    return "enroll/manager/enrollList";
}
```

### EnrollRepository

```
public List<Enroll> findAllByCriteria(EnrollSearch enrollSearch) {
    CriteriaBuilder cb = em.getCriteriaBuilder();
    CriteriaQuery<Enroll> cq = cb.createQuery(Enroll.class);
    Root<Enroll> e = cq.from(Enroll.class);
    Join<Enroll, Student> s = e.join("student", JoinType.INNER); //학생과 조인
    List<Predicate> criteria = new ArrayList<>();
    //수강 신청 상태 검색
    if (enrollSearch.getEnrollStatus() != null) {
        Predicate status = cb.equal(e.get("status"),
            enrollSearch.getEnrollStatus());
        criteria.add(status);
    }
    //학번 검색
    if (StringUtils.hasText(enrollSearch.getStudentId())) {
        Predicate studentId =
            cb.like(s.<get>("studentId"), pattern: "%" +
                enrollSearch.getStudentId() + "%");
        criteria.add(studentId);
    }
    cq.where(cb.and(criteria.toArray(new Predicate[criteria.size()]));
    TypedQuery<Enroll> query = em.createQuery(cq).setMaxResults(1000); //최대1000건
    return query.getResultList();
}
```

## 기술 상세\_수강취소

```
<tr th:each="enroll : ${enrolls}" ">
  <td th:text="${enroll.id}"></td>
  <td th:text="${enroll.student.studentId}"></td>
  <td th:text="${enroll.subject.subjectCode}"></td>
  <td th:text="${enroll.subject.subjectName}"></td>
  <td th:text="${enroll.subject.subjectCredit}"></td>
  <td th:text="${enroll.subject.classification}"></td>
  <td th:text="${enroll.subject.professor}"></td>
  <td>
    <a th:if="${enroll.status.name() == 'Enrolment'}" href="#"
      th:href="'javascript:cancel('${enroll.id}+)'">CANCEL</a>
  </td>
</tr>
</script>
<script>
  function cancel(id) {
    var form = document.createElement("form");
    form.setAttribute("method", "post");
    form.setAttribute("action", "/enrolls/" + id + "/cancel");
    document.body.appendChild(form);
    form.submit();
  }
</script>
```

enrollList.html

## Controller & Service & Domain

### EnrollController

```
@PostMapping("/enrolls/{enrollId}/cancel")
public String cancelEnroll(@PathVariable("enrollId") Long enrollId){
    enrollService.cancelEnroll(enrollId);
    return "redirect:/manager/enrolls";
}
```

### EnrollService

```
@Transactional
public void cancelEnroll(Long enrollId){
    //수강 엔티티 조회
    Enroll enroll = enrollRepository.findOne(enrollId);

    //철회하는 로직
    enroll.cancel();
}
```

### Enroll

```
public void cancel(){
    if(EnrollStatus.Completed == getStatus()){
        throw new IllegalStateException("이미 수강 완료된 강의입니다.");
    }
    this.setStatus(EnrollStatus.CANCEL);
}
```



**Thank you !**