



대학교 포털사이트 구현하기

오은혜

01

기획의도

02

개발환경

03

DB설계

04

기능구현



01

기획의도

공공기관, 대학교 등에서
사용하는 프로그램을
구현해보고자
대학교 포털 사이트를 구현





02

개발환경



1

공통

- 운영체제 : window 10
- 소스코드 작성 도구 : Eclipse EE
- 웹 서버 : Apache Tomcat 9.0

2

Front-End

HTML5, CSS3, JavaScript, J-Query, Ajax

3

Back-End

Spring, JAVA, JSP, MVC Model2, MyBatis

4

DataBase

Oracle



03

DB설계

SYSTEM.BASICINFO		
P *	STUDENTID	VARCHAR2 (8 BYTE)
*	PASSWORD	VARCHAR2 (100 BYTE)
*	NAME	VARCHAR2 (10 BYTE)
	GENDER	VARCHAR2 (5 BYTE)
	BIRTH	NUMBER (6)
	ACAYEAR	NUMBER (1)
	EMAIL	VARCHAR2 (100 BYTE)
	ADDRESS	VARCHAR2 (300 BYTE)
	PHONE	VARCHAR2 (13 BYTE)
	UPLOADFILE	VARCHAR2 (300 BYTE)
	MMAJOR	VARCHAR2 (100 BYTE)
	DOUMAJOR	VARCHAR2 (100 BYTE)
BASICINFO_PK (STUDENTID)		
BASICINFO_PK (STUDENTID)		

SYSTEM.CLASS		
*	STUDENTID	VARCHAR2 (8 BYTE)
*	APPYEAR	NUMBER (4)
*	APPSEMESTER	VARCHAR2 (10 BYTE)
UF	SUBJECTID	VARCHAR2 (100 BYTE)
CLASS_SUBJECTID_UN (SUBJECTID)		
SUBJECTID_FK (SUBJECTID)		



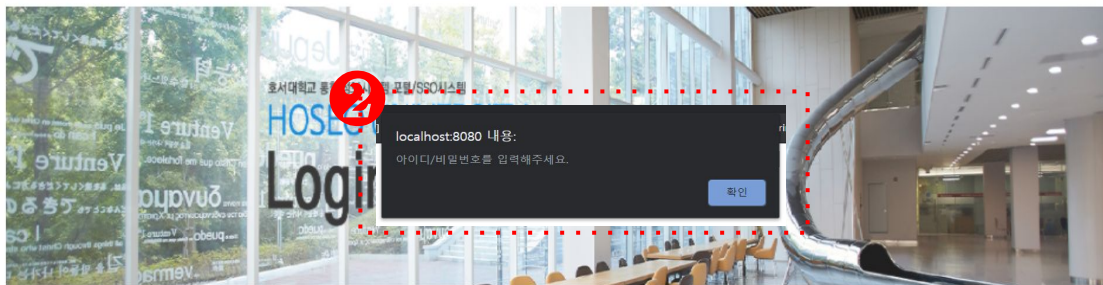
SYSTEM.SUBJECT		
*	SEQ	NUMBER (5)
P *	SUBJECTID	VARCHAR2 (100 BYTE)
*	SUBCREDIT	NUMBER (1)
*	CLASSIFI	VARCHAR2 (8 BYTE)
*	PROFESSOR	VARCHAR2 (100 BYTE)
SUBJECTID_PK (SUBJECTID)		
SUBJECTID_PK (SUBJECTID)		

DB 수정중입니다!



04

기능구현



1

로그인

☐ 학번/교번 저장

학번/교번 찾기 | 회원가입

공지사항	문의사항
[교수학습센터] 2020-2학기 학습법워크숍 학습성찰일지워크숍 추가.	2020-10-27
[국제협력원] 코로나-19 확산에 따른 해외연수 미 실시 안내	2020-10-26
[LINC+사업단] 2020학년도 LINC+사업단 캠퍼스디자인실 또는 Adven..	2020-10-26
[공학교육혁신센터] 파이벤(스텍코딩)과정, PLC프로그래밍 과정 참여.	2020-10-26
2020학년도 2학기 중소기업 취업연계 장학금 신청 (희망사다리 장학금.	2020-10-26
[대학일자리센터] 취업준비생이 알아야 하는 노동법 특강 안내	2020-10-23
[LINC+사업단] 2020년 산학협력 페스티벌 Venture 1st UNISTRY DAY..	2020-10-23

Description

- 로그인 기능
 - Spring Security 이용
- 알림창
 - 아이디/비밀번호 입력에 따라 JavaScript로 alert()

기술 상세_로그인

```
@RequestMapping(value = "/login")
public ModelAndView login(ModelAndView mv, MemberVO vo, HttpServletRequest request) {
    String id = vo.getStudentid();
    String password = vo.getPassword();

    if (id.length() > 0 && password.length() > 0) {
        vo = service.selectOne(vo);
        if (vo == null) {
            mv.setViewName("home");
            mv.addObject("message", "등록되지 않은 아이디입니다.");
        } else {
            if(passwordEncoder.matches(password, vo.getPassword())) {
                HttpSession session = request.getSession();
                session.setAttribute("logID", id);
                session.setAttribute("logName", vo.getName());
                session.setAttribute("logPW", password);
                if ("admin".equals(id)) mv.setViewName("manager/mag_layout");
                else mv.setViewName("layout/layout");
            } else {
                mv.setViewName("home");
                mv.addObject("message", "등록되지 않은 비밀번호입니다.");
            } // Password 비교
        } // selectOne 성공 여부
    } else { // 입력값 check
        mv.setViewName("home");
        mv.addObject("message", "아이디/비밀번호를 입력해주세요.");
    }
    return mv;
} // login
```

MemberController

Spring Security

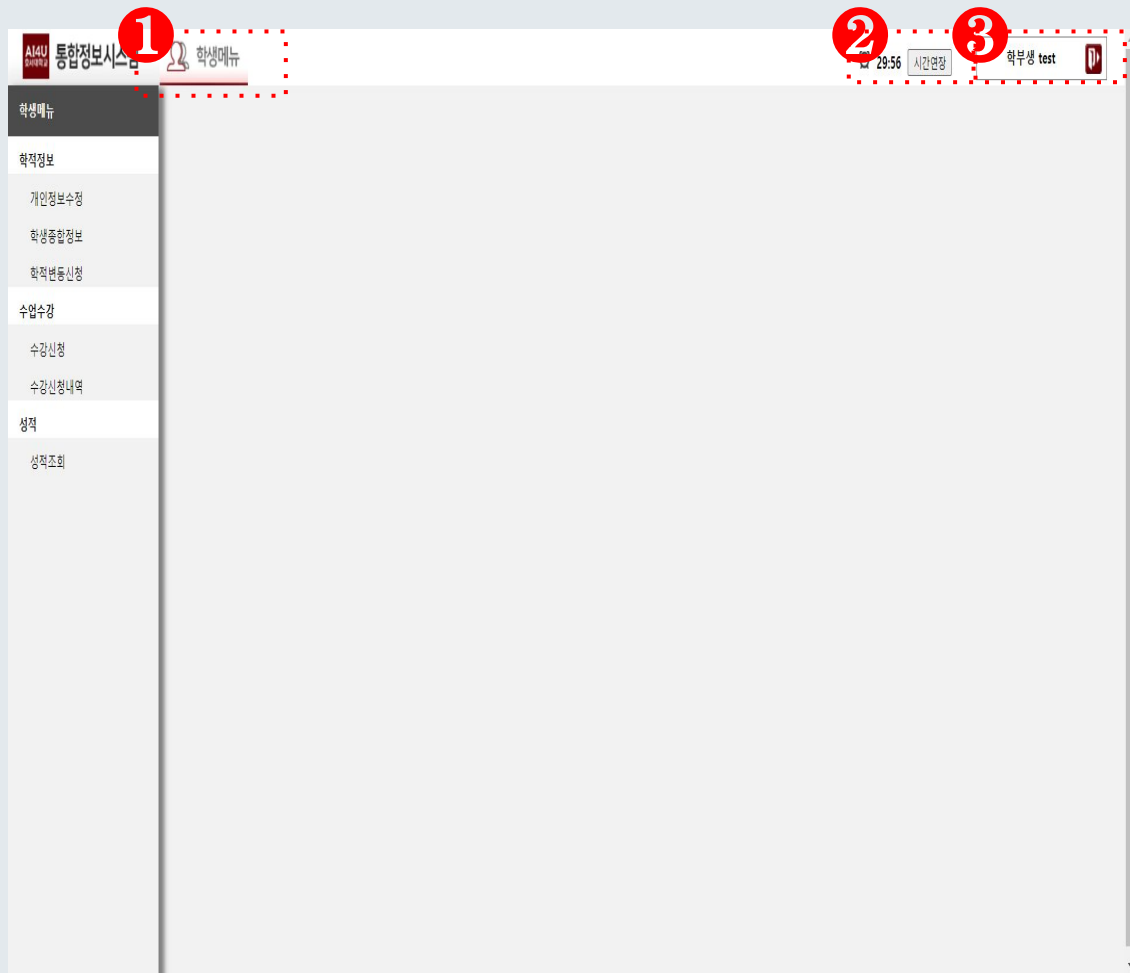
servlet-context.xml

```
<!--스프링시큐리티 라이브러리 BcryptPasswordEncoder -->
<beans:bean id="bcryptPasswordEncoder"
class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder" />
</beans:beans>
```

pom.xml

```
<!--스프링시큐리티 web 라이브러리 (설치된 스프링 버전과 일치해야 함)-->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>4.2.4.RELEASE</version>
</dependency>

<!--스프링시큐리티 core 라이브러리-->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>4.2.4.RELEASE</version>
</dependency>
```



Description

1. 학생 메인 페이지
 - ID로 학생/관리자 화면 구분
2. Time Out
 - 로그인 30분 후 자동 로그아웃
 - 시간연장 버튼 클릭 > 30분 연장
 - JavaScript
3. 사용자 구분 및 로그아웃
 - 로그인 ID로 학생/이름 출력
 - 나가기 버튼 클릭 > 로그아웃

기술 상세_학생 메인 페이지

```
@RequestMapping(value = "/login")
public ModelAndView login(ModelAndView mv, MemberVO vo, HttpServletRequest
    String id = vo.getStudentid();
    String password = vo.getPassword();

    if (id.length() > 0 && pas
        vo = service.selectOne
        if (vo == null) {
            mv.setViewName("home");
            mv.addObject("message", "아이디/비밀번호를 입력해주세요.");
        } else {
            if (passwordEncoder.matches(password, vo.getPassword())) {
                HttpSession session = request.getSession();
                session.setAttribute("logID", id);
                session.setAttribute("logName", vo.getName());
                session.setAttribute("logPW", password);
                if ("admin".equals(id)) mv.setViewName("manager/mag_layout");
                else mv.setViewName("layout/layout");
            } else {
                mv.setViewName("home");
                mv.addObject("message", "등록되지 않은 비밀번호입니다.");
            } // Password 비교
        } // selectOne 성공 여부
    } else { // 입력값 check
        mv.setViewName("home");
        mv.addObject("message", "아이디/비밀번호를 입력해주세요.");
    }
    return mv;
} // login
```

- 아이디값에 따라 사용자 구분
- Session에 아이디/이름 저장

MemberController

Time Out / 시간 연장 button

```
var tid;
var cnt = parseInt(60*30); //초기 값 (초 단위)
function counter_init() {
    tid = setInterval("counter_run()", 1000);
}

function counter_reset() {
    clearInterval(tid);
    cnt = parseInt(60*30);
    counter_init();
}

function counter_run() {
    document.all.counter.innerText = time_format(cnt);
    cnt--;
    if (cnt < 0) {
        clearInterval(tid);
        self.location = "logout";
    }
}

function time_format(s) {
    var nHour=0;
    var nMin=0;
    var nSec=0;
    if (s>0) {
        nMin = parseInt(s/60);
        nSec = s%60;

        if (nMin>60) {
            nMin = nMin%60;
        }
    }
    if (nSec<10) nSec = "0"+nSec;
    if (nMin<10) nMin = "0"+nMin;

    return ""+nMin+":"+nSec;
}
```


학생메뉴

학자정보

개인정보수정

학생종합정보

학적변동신청

수업수강

수강신청

수강신청내역

성적

성적조회

학생메뉴 > 학생메뉴 > 수업수강 > 수강신청

2

학년도 * 2020 학기 * 1학기 학번/성명 12345 test

수강신청

저장

3

NO	과목명	학점	이수구분	담당교수	선택
1	Java	3	전공필수	김교수	<input type="checkbox"/>
2	Jsp	3	전공과목	이교수	<input type="checkbox"/>
3	Servlet	3	전공과목	박교수	<input type="checkbox"/>
4	Spring	3	전공필수	오교수	<input type="checkbox"/>
5	MyBatis	3	전공과목	한교수	<input type="checkbox"/>
6	재미있는 엑셀	2	균형교양	한교수	<input type="checkbox"/>
7	전공탐색	1	기본인성	조교	<input type="checkbox"/>
8	심리학개론	3	자유선택	김심리	<input type="checkbox"/>

Description

1. 메뉴바

- 메뉴 클릭 > ajax를 이용하여 빈 화면에 출력

2. 수강신청 기본 정보

- 연도는 컨트롤러에서 올해 연도 구하는 로직이용
- 학번 / 성명은 session 정보

3. 과목 내역

- 과목 DB에 있는 리스트 출력

기술 상세_학생_수강 신청

```
@RequestMapping(value = "/st_classf")
public ModelAndView st_classf(ModelAndView mv, SubjectVO vo) {
    int y = Calendar.getInstance().get(Calendar.YEAR);
    mv.addObject("y", y);
    List<SubjectVO> sbjList = sbjService.sbjList(vo);
    mv.addObject("sbjList", sbjList);
    mv.setViewName("student/st_classf");
    return mv;
} // st_classf
```

- 올해 연도 구하기
- sbjList 를 mv에 담기

MemberController

과목 List 출력

subjectMapper.xml

```
<mapper namespace="com.portal.subjectMapper">

    <select id="sbjList" resultType="vo.SubjectVO">
        select * from subject
    </select>

</mapper>
```

layout/layout.jsp

```
$('#class').click(function() {
    $.ajax({
        type : 'post',
        url : 'st_classf',
        success : function(result) {
            $('#article').html(result);
        },
        error : function() {
            alert('서버에러(st_classf)');
        }
    }); //ajax
}); //st_classList
```

2 수강신청 저장

NO	과목명	학점	이수구분	담당교수	선택
1	Java	3	전공필수	김교수	<input type="checkbox"/>
2	Jsp	3	전공과목	이교수	<input type="checkbox"/>
3	Servlet	3	전공과목	박교수	<input type="checkbox"/>
4	Spring	3	전공필수	오교수	<input type="checkbox"/>
5	MyBatis	3	전공과목	한교수	<input type="checkbox"/>
6	재미있는 엑셀	2	균형교양	한교수	<input type="checkbox"/>
7	전공탐색	1	기본인성	조교	<input type="checkbox"/>
8	심리학개론	3	자유선택	김심리	<input type="checkbox"/>

1



Document - Chrome

localhost:8080/proj/st_class?appYear=2020&appSemester=1semester&studentid=12345&name=st&subjectids=재미있는%20엑셀

학생메뉴 > 학생메뉴 > 수업수강 > 수강신청내역확인

학년도 * 2020 학기 * 1학기 학번/성명 12345 test

3 수강신청내역 확인 삭제

NO	교과목명	학점	이수구분	담당교수	선택
1	MyBatis	3	전공과목	한교수	<input type="checkbox"/>
2	심리학개론	3	자유선택	김심리	<input type="checkbox"/>
3	Spring	3	전공필수	오교수	<input type="checkbox"/>
4	Java	3	전공필수	김교수	<input type="checkbox"/>
5	Jsp	3	전공과목	이교수	<input type="checkbox"/>
6	전공탐색	1	기본인성	조교	<input type="checkbox"/>
7	재미있는 엑셀	2	균형교양	한교수	<input type="checkbox"/>
합계		18			

Description

1. 과목 선택
- 수강하고 싶은 과목 선택
2. 저장버튼 클릭
- 버튼 클릭 > CList에 선택된 과목 Insert
- 수강 내역 확인 팝업창
3. 수강 신청 내역 확인
- 올해 수강 신청 내역 확인

기술 상세_학생_수강신청 페이지

```
@RequestMapping(value = "/st_class")
public ModelAndView st_class(ModelAndView mv, ClassVO cvo) throws SQLIntegrityConstraintViolationException{
    int y = Calendar.getInstance().get(Calendar.YEAR);
    mv.addObject("y", y);

    try {
        if(cvo.getSubjectids() != null && cvo.getSubjectids().length > 0) {
            if (cService.clInsert(cvo)
                List<ClassVO> cllist =
                mv.addObject("cllist",
                mv.setViewName("student/doFinish");
            }
        } else {
            mv.addObject("fCode", "F");
            mv.addObject("message", "선택한 과목이 없습니다.");
            mv.setViewName("student/doFinish");
        }
    } catch (Exception e) {
        System.out.println("에러" + e.toString());
        mv.addObject("fCode", "F");
        mv.addObject("message", "이미 선택한 과목이 있습니다.");
        mv.setViewName("student/doFinish");
    }

    return mv;
} // st_class
```

- 상황별(선택한 과목이 없을 경우 / 이미 선택한 과목을 선택한 경우) 경고창

MemberController

수강신청 팝업창 & insert

classMapper.xml

```
<insert id="clInsert">
    insert all
    <foreach collection="subjectids" item="it">
        into class values(#{studentid},#{appYear},#{appSemester},#{it})
    </foreach>
    select * from dual
</insert>
```

st_class.jsp

```
<script>
function classReq(){

    var formData= $('#classForm').serialize();
    console.log('form' +formData);
    var url = "st_class?" +formData;
    window.open(url,"_blank",
        "toolbar=no,menubar=yes,scrollbars=yes,resizable=yes,width=1200,height=500");

} //classReq()
</script>
```

학생메뉴>학생메뉴>수업수강>수강신청내역확인

학년도 * 2020 학기 * 1학기 학번/성명 12345 test

수강 내역 확인 삭제

NO	교과목명	학점	이수구분	담당교수	선택
1	MyBatis	3	전공과목	한교수	<input checked="" type="checkbox"/>
2	심리학개론	3	자유선택	김심리	<input checked="" type="checkbox"/>
3	Spring	3	전공필수	오교수	<input checked="" type="checkbox"/>
4	Java	3	전공필수	김교수	<input type="checkbox"/>
5	Jsp	3	전공과목	이교수	<input type="checkbox"/>
6				조교	<input type="checkbox"/>
7				한교수	<input type="checkbox"/>
합계					

Semester=1semester&studentid=12345&name=test&subjectids=MyBatis

localhost:8080 내용:

해당 강의가 삭제되었습니다.

확인

학생메뉴>학생메뉴>수업수강>수강신청

학년도 * 2020 학기 * 1학기

수강신청 저장

NO	과목명	학점	이수구분	담당교수	선택
1	Java	3	전공필수	김교수	<input type="checkbox"/>
2	Jsp	3	전공과목	이교수	<input type="checkbox"/>
3	Servlet	3	전공과목	박교수	<input type="checkbox"/>
4	Spring	3	전공필수	오교수	<input type="checkbox"/>
5	MyBatis	3	전공과목	한교수	<input type="checkbox"/>
6	재미있는 엑셀	2	교형교양	한교수	<input type="checkbox"/>
7	전공탐색	1	기본인성	조교	<input type="checkbox"/>
8	심리학개론	3	자유선택	김심리	<input type="checkbox"/>

Description

1. 과목 선택
- 삭제하고 싶은 과목 선택
2. 삭제버튼 클릭
- 버튼 클릭 > CList에 선택된 과목 delete
3. 알림창
4. 알림창 확인 클릭 후
- 팝업창 self.close()
- 수강신청 선택 클릭되었던 내역 체크 해제

기술 상세_학생_수강 신청 삭제 페이지

```
@RequestMapping(value = "/st_classDel")
public ModelAndView st_classDel(ModelAndView mv, ClassVO cvo) {
    int cnt = cService.deleteClass(cvo);
    if(cnt>0) {
        mv.addObject("message", "해당 강의가 삭제되었습니다.");
        mv.addObject("fCode", "S");
    }else {
        System.out.println("nono" + cnt);
        mv.addObject("fCode", "F");
    }
    mv.setViewName("student/doFinish");
    return mv;
}
// st_classDel
```

MemberController

```
<delete id="deleteClass">
    delete from class where subjectid in
    <foreach collection="subjectids" item="it" open="(" close=")" separator=",">
        #{it}
    </foreach>
</delete>
```

classMapper.xml

팝업창 닫기 & 체크 해제

student/doFinish.jsp

```
<!-- 삭제 성공 후 창닫기 -->
<c:if test="${fCode=='S'}">
<script>
    alert('${message}');
    self.close();

    opener.$('input:checkbox[name="subjectids"]').each(function(){
        this.checked = false;
    }); //checkbox 모두 해제
</script>
</c:if>
```

1

학생메뉴>학생메뉴>수업수강>수강내역 (UCR0123Q)

2

학년도 * 2020 학기 * 1학기 학번/성명 12345 test

조회

수강신청내역 [총 건]

NO	교과목명	학점	이수구분	담당교수
합계		0		



학생메뉴>학생메뉴>수업수강>수강내역 (UCR0123Q)

학년도 * 2020 학기 * 1학기 학번/성명 12345 test

조회

3

수강신청내역 [총 3 건]

4

NO	교과목명	학점	이수구분	담당교수
1	Servlet	3	전공과목	박교수
2	재미있는 엑셀	2	교형교양	한교수
3	전공탐색	1	기본인성	조교
합계		6		

Description

1. 정보
 - 학년도, 학번 및 성명은 컨트롤러, session 정보
2. 조회 버튼
 - 클릭 시 요청 정보에 따라 수강신청내역 출력
3. 요청정보의 총 수강신청내역
4. 수강신청내역
 - selectList

기술 상세_학생_수강 신청 내역 페이지

```
@RequestMapping(value = "/st_classListf")
public ModelAndView st_classListf(ModelAndView mv, ClassVO cvo) {
    int y = Calendar.getInstance().get(Calendar.YEAR);
    mv.addObject("y", y);
    mv.setViewName("student/st_classList");
    return mv;
}

@RequestMapping(value = "/st_classList")
public ModelAndView st_classList(ModelAndView mv, ClassVO cvo) {
    int y = Calendar.getInstance().get(Calendar.YEAR);
    mv.addObject("y", y);
    List<ClassVO> cllist = cService.classList(cvo);
    System.out.println(cllist);
    if(cllist.size() > 0) {
        mv.addObject("cllist", cllist);
        mv.setViewName("student/st_classList");
    } else {
        mv.addObject("fCode", "F");
        mv.addObject("message", "수강 신청 내역이 없습니다.");
        mv.setViewName("student/finish");
    }
    return mv;
}
```

MemberController

```
<!-- st_classList 에서 조건(학년도, 학기, 학번, 이름으로 조회하면 조건에 맞는 수강list -->
<select id="classList" resultType="vo.ClassVO">
    select * from class c Join subject s ON c.subjectid = s.subjectid
    where c.appYear = #{appYear} and c.appSemester = #{appSemester} and c.studentid = #{studentid}
</select>
```

classMapper.xml

합계

```
<c:set var = "sum" value="${cllist.size()}" />
수강신청내역 [총 <c:out value="${sum}" /> 건]
</div>
<div id="list">
    <table>
        <tr>
            <td>NO</td>
            <td>교과목명</td>
            <td>학점</td>
            <td>이수구분</td>
            <td>담당교수</td>
        </tr>
        <c:set var = "sum" value="0"/>
        <c:forEach var = "cllist" items="${cllist}" varStatus="status">
            <tr>
                <td>${status.count}</td>
                <td>${cllist.subjectid}</td>
                <td>${cllist.subCredit}</td>
                <td>${cllist.classifi}</td>
                <td>${cllist.professor}</td>
            </tr>
            <c:set var="sum" value="${sum} + cllist.subCredit" />
        </c:forEach>
        <tr>
            <td>합계</td>
            <td></td>
            <td><c:out value="${sum}" /></td>
```

student/st_classList.jsp

로그인

학번/교번 저장

로그인

학번/교번 찾기

회원가입

공지사항	문의사항
[교수학습센터] 2020-2학기 학습법워크숍-학습성찰일지워크숍 추가..	2020-10-27
[국제협력팀] 코로나-19학산에 따른 해외연수 미실시 안내	2020-10-26
[LINC+사업단]2020학년도 LINC+사업단 캠퍼스디자인실 또는 Adven..	2020-10-26
[공학교육혁신센터] 파이선(스택코딩)과정, PLC프로그래밍 과정 참여..	2020-10-26
2020학년도 2학기 중소기업 취업연계 장학금 신청 (희망사다리 장학금..	2020-10-26
[대학일자리센터] 취업준비생이 알아야 하는 노동법 특강 안내	2020-10-23
[LINC+사업단] 2020년 산학협력 페스티벌 Venture 1st UNISTRY DAY..	2020-10-23

1

개인정보처리방침

이메일무단수집거부

오시는길

교내전화번호

학내백신다운로드

호서대학교

HOSEO UNIVERSITY

아산캠퍼스 충청남도 아산시 배방읍 호서로79번길20 (우)31499 TEL:041-540-5114

천안캠퍼스 충청남도 천안시 동남구 호서대길 T2 (우)31066 TEL:041-560-8114

COPYRIGHT(C) 2011 HOSEO UNIVERSITY. ALL RIGHT RESERVED

호서대학교 위치

[HOME]

Description

- 오시는 길 > 클릭
- 위치 표시

student/st_clsList.jsp 기능 구현

기술 상세_오시는길

```
<script type="text/javascript"
src="//dapi.kakao.com/v2/maps/sdk.js?appkey=5aa24558714e59eec84f58205d2a4fbf&libraries=services">
</script>
<body>
<h2>호서대학교 위치</h2>
<br><hr>
<div id="map" style="width:90%;height:400px;"></div>
</script>
//BODY 영역에 작성
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(36.799167, 127.133306), // 지도의 중심좌
        level: 3 // 지도의 확대 레벨
    };

// 지도 생성
var map = new kakao.maps.Map(mapContainer, mapOption);

// 복수개의 마커 와 설명(info)을 위한 Data
var positions = [
    { content:'<div>천안 캠퍼스</div>', // 마커 롤오버시 표시할 내용
      latlng:new kakao.maps.LatLng(36.827655, 127.183389) },
    { content:'<div>아산 캠퍼스</div>',
      latlng:new kakao.maps.LatLng(36.73614259017708, 127.07436939936595) },
];

//마커 이미지의 이미지 주소
var imageSrc = "http://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerStar.png";

// infoTest 2) 마커에 표시할 인포윈도우 생성
var infowindow = new kakao.maps.InfoWindow({ zIndex:1 });

// 지도의 중심좌표를 표시 위치에 따라 재설정 하기위해 추가
var bounds = new kakao.maps.LatLngBounds();

for (var i=0;i<positions.length;i++) {

    // 마커 이미지의 이미지 크기 입니다
    var imageSize = new kakao.maps.Size(24, 35);
    // 마커 이미지를 생성합니다
    var markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize);
```

- Kakao Map API 를 이용하여 구현

address.jsp

오시는길

```
// 마커 생성
var marker = new kakao.maps.Marker({
    map: map, //마커를 표시할 지도
    position: positions[i].latlng, //마커위치
    title: positions[i].content,
    // 마커에 마우스를 올리면 타이틀이 표시됨 -> 그러나 html 적용안됨
    image: markerImage // 마커 이미지
});

positions[i].content);

//마커에 표시할 인포윈도우 생성
aps.InfoWindow({
    tions[i].content // 인포윈도우에 표시할 내용 : html 적용됨
});

// 인수로 주어진 좌표를 포함하도록 영역 정보를 확장한다.
bounds.extend(positions[i].latlng);
map.setBounds(bounds);

// 인포윈도우를 표시하는 함수
function makeOverListener(map,marker,infowindow,infoContent) {
    return function() {
        // 이때 infowindow 는 상단에 전역으로 정의 해 놓음
        infowindow.setContent(infoContent);
        infowindow.open(map,marker);
    };
} // makeOverListener

//인포윈도우를 닫는 함수
function makeOutListener(infowindow) {
    return function() {
        infowindow.close();
    };
} // makeOverListener
```


Thank you !