네트워크 프로그래밍 구현

서정현

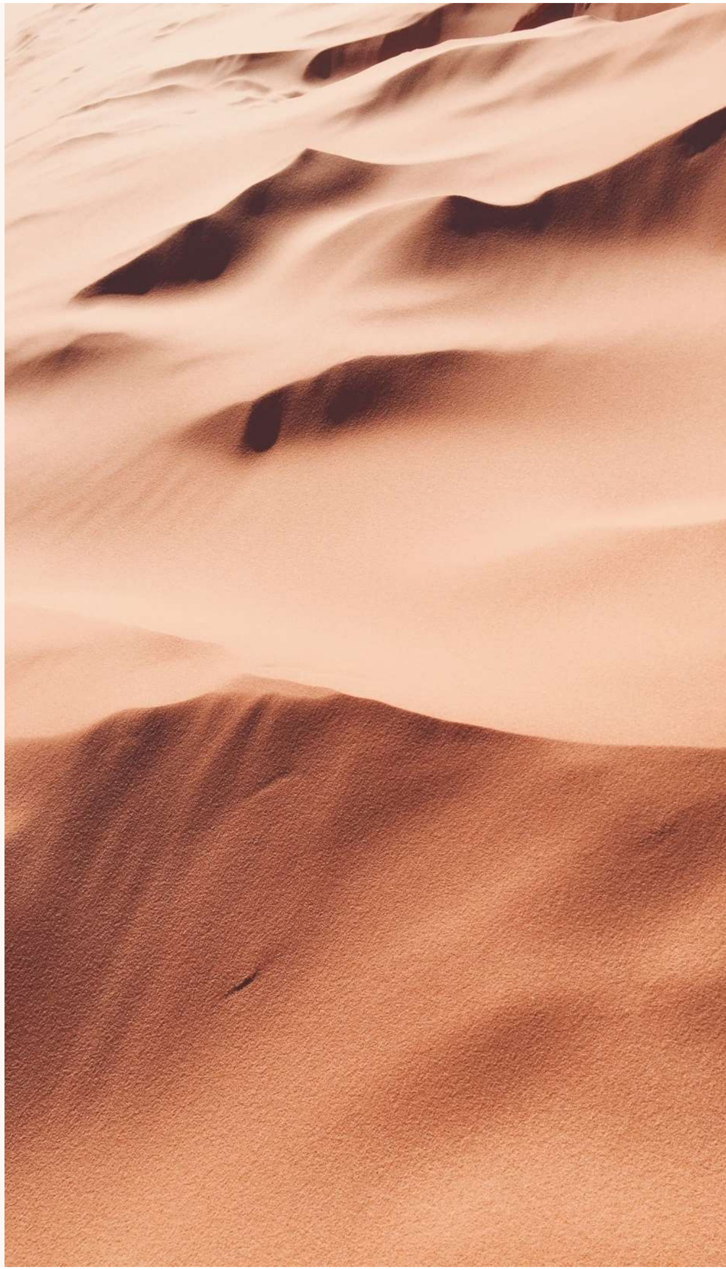
목차 A table of contents.

1. 문제 1 – 프로젝트 생성 및 구성

2. 문제 2 – 화면 구현

3. 문제 3 – 기능 구현

4. 문제 4 – 실행



1. 문제 1 – 프로젝트 생성 및 구성

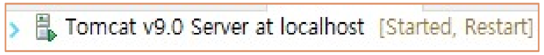
1. 프로젝트명 : College

2. WAS : Tomcat 9

3. DB : java2_College

4. 개발 도구 : Eclipse, workbench

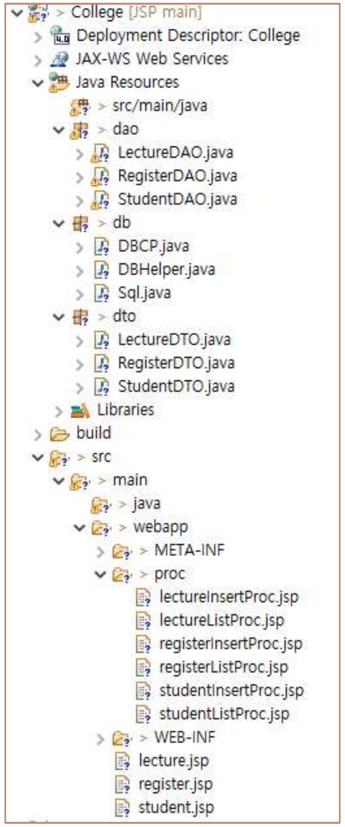
❷



```
> Tomcat v9.0 Server at localhost  [Started, Restart]
```

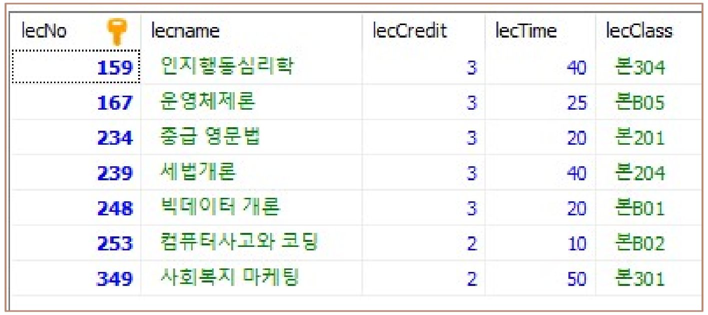
❹

❶

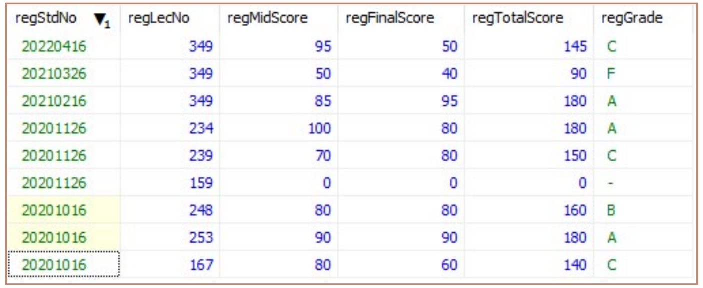


```
v > College [JSP main]
  > Deployment Descriptor: College
  > JAX-WS Web Services
  v Java Resources
      > src/main/java
    v > dao
      > LectureDAO.java
      > RegisterDAO.java
      > StudentDAO.java
    v > db
      > DBCP.java
      > DBHelper.java
      > Sql.java
    v > dto
      > LectureDTO.java
      > RegisterDTO.java
      > StudentDTO.java
    > Libraries
  > build
  v > src
    v > main
        > java
      v > webapp
        > > META-INF
        v > proc
            lectureInsertProc.jsp
            lectureListProc.jsp
            registerInsertProc.jsp
            registerListProc.jsp
            studentInsertProc.jsp
            studentListProc.jsp
        > > WEB-INF
          lecture.jsp
          register.jsp
          student.jsp
```

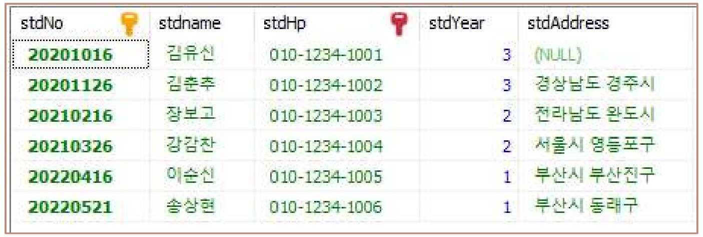
❸



lecNo	lecname	lecCredit	lecTime	lecClass
159	인지행동심리학	3	40	본304
167	운영체제론	3	25	본B05
234	중급 영문법	3	20	본201
239	세법개론	3	40	본204
248	빅데이터 개론	3	20	본B01
253	컴퓨터사고와 코딩	2	10	본B02
349	사회복지 마케팅	2	50	본301



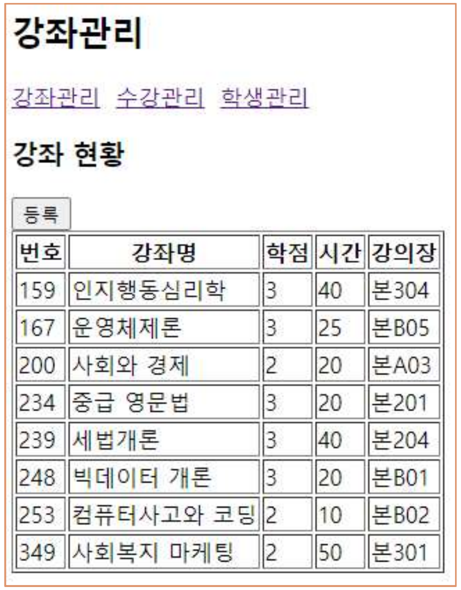
regStdNo	regLecNo	regMidScore	regFinalScore	regTotalScore	regGrade
20220416	349	95	50	145	C
20210326	349	50	40	90	F
20210216	349	85	95	180	A
20201126	234	100	80	180	A
20201126	239	70	80	150	C
20201126	159	0	0	0	-
20201016	248	80	80	160	B
20201016	253	90	90	180	A
20201016	167	80	60	140	C



stdNo	stdname	stdHp	stdYear	stdAddress
20201016	김유신	010-1234-1001	3	(NULL)
20201126	김춘추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

2. 문제 2 – 화면 구현 – College(도서)

lecture(강좌)



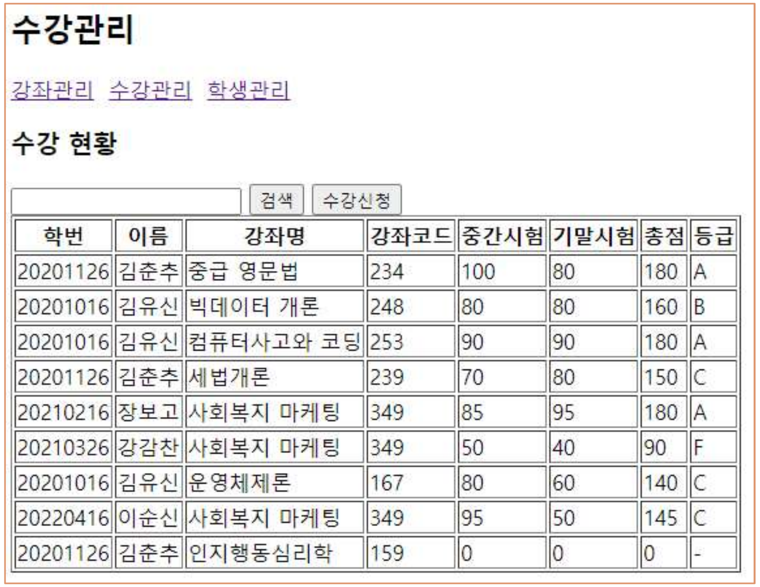
강좌관리

강좌관리 수강관리 학생관리

강좌 현황

번호	강좌명	학점	시간	강의장
159	인지행동심리학	3	40	본304
167	운영체제론	3	25	본B05
200	사회와 경제	2	20	본A03
234	중급 영문법	3	20	본201
239	세법개론	3	40	본204
248	빅데이터 개론	3	20	본B01
253	컴퓨터사고와 코딩	2	10	본B02
349	사회복지 마케팅	2	50	본301

register(수강)



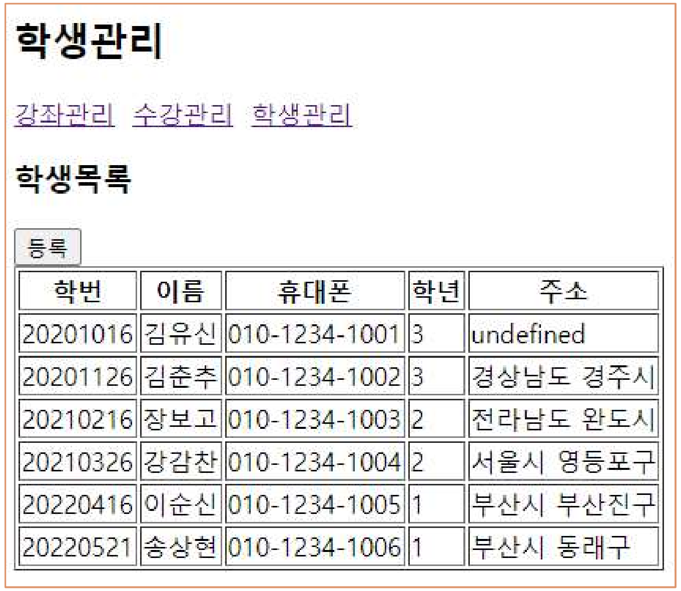
수강관리

강좌관리 수강관리 학생관리

수강 현황

학번	이름	강좌명	강좌코드	중간시험	기말시험	총점	등급
20201126	김춘추	중급 영문법	234	100	80	180	A
20201016	김유신	빅데이터 개론	248	80	80	160	B
20201016	김유신	컴퓨터사고와 코딩	253	90	90	180	A
20201126	김춘추	세법개론	239	70	80	150	C
20210216	장보고	사회복지 마케팅	349	85	95	180	A
20210326	강감찬	사회복지 마케팅	349	50	40	90	F
20201016	김유신	운영체제론	167	80	60	140	C
20220416	이순신	사회복지 마케팅	349	95	50	145	C
20201126	김춘추	인지행동심리학	159	0	0	0	-

student(학생)



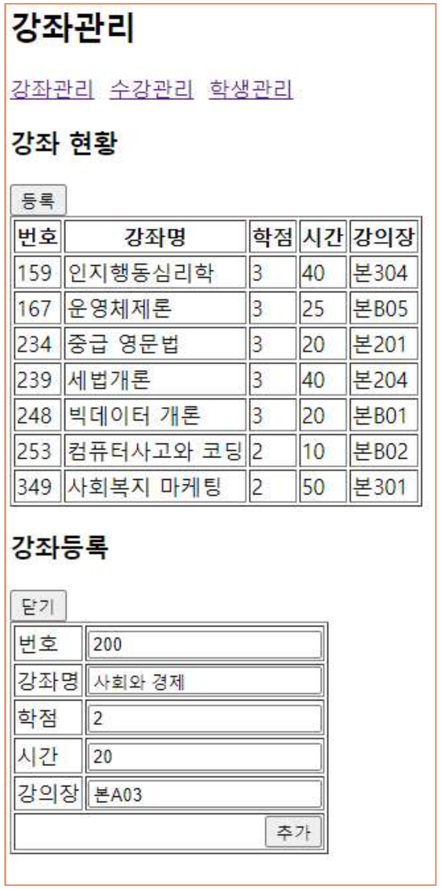
학생관리

강좌관리 수강관리 학생관리

학생목록

학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	undefined
20201126	김춘추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

3. 문제 3 – 기능 구현 - lecture(강좌) 기본 CRUD_목록, 등록



강좌관리

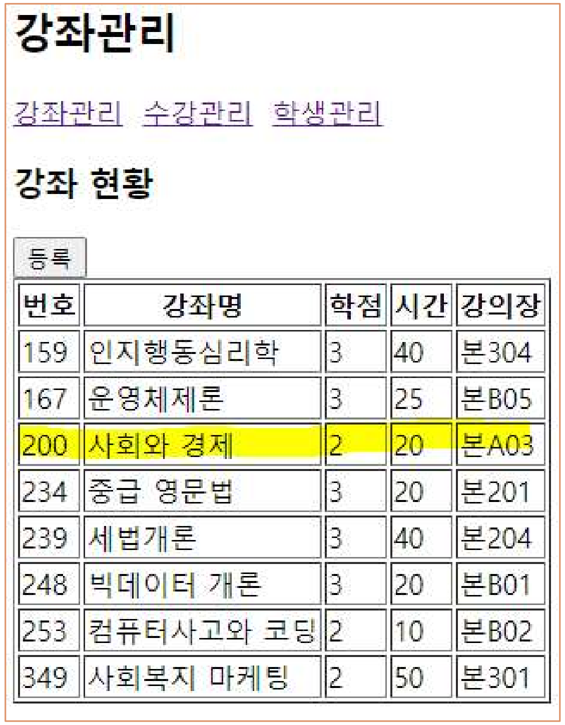
강좌관리 수강관리 학생관리

강좌 현황

번호	강좌명	학점	시간	강의장
159	인지행동심리학	3	40	본304
167	운영체제론	3	25	본B05
234	중급 영문법	3	20	본201
239	세법개론	3	40	본204
248	빅데이터 개론	3	20	본B01
253	컴퓨터사고와 코딩	2	10	본B02
349	사회복지 마케팅	2	50	본301

강좌등록

번호	200
강좌명	사회와 경제
학점	2
시간	20
강의장	본A03
	추가



강좌관리

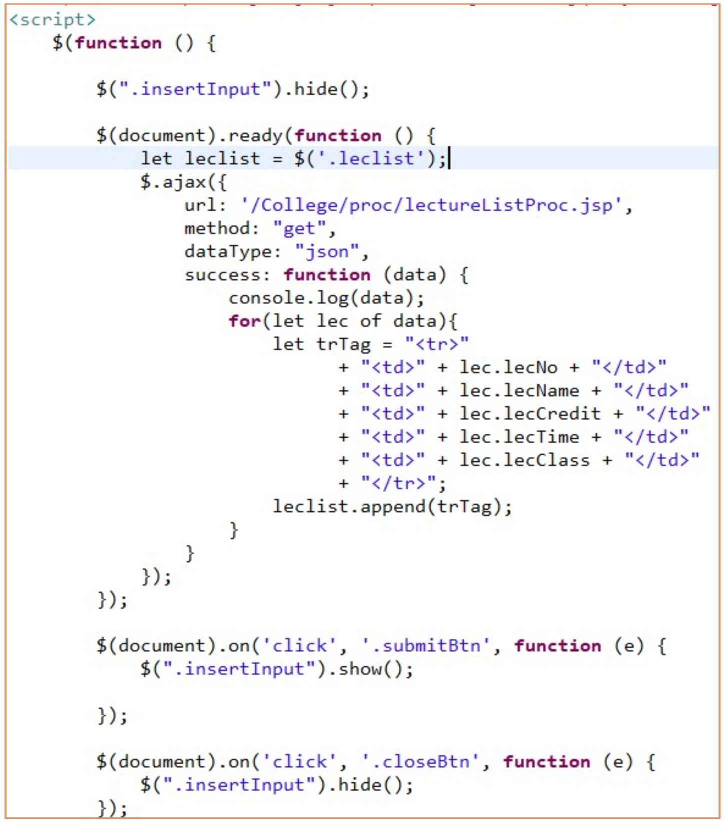
강좌관리 수강관리 학생관리

강좌 현황

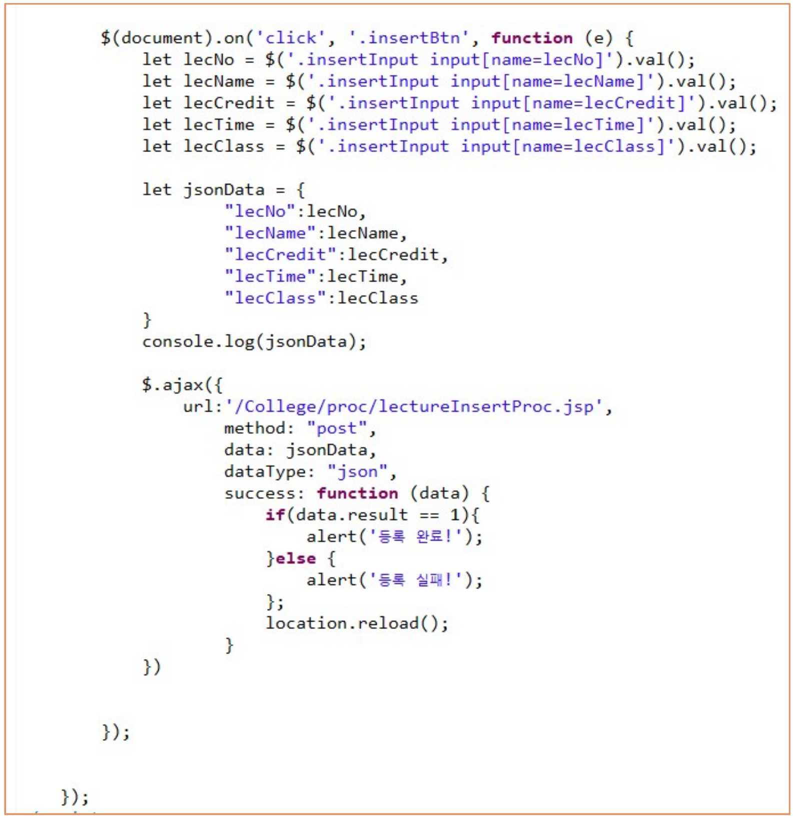
번호	강좌명	학점	시간	강의장
159	인지행동심리학	3	40	본304
167	운영체제론	3	25	본B05
200	사회와 경제	2	20	본A03
234	중급 영문법	3	20	본201
239	세법개론	3	40	본204
248	빅데이터 개론	3	20	본B01
253	컴퓨터사고와 코딩	2	10	본B02
349	사회복지 마케팅	2	50	본301

3. 문제 3 – 기능 구현 - lecture(강좌) 기본 CRUD_목록, 등록

lecture.jsp 스크립트 부분



```
<script>
    $(function () {

        $(".insertInput").hide();

        $(document).ready(function () {
            let leclist = $('.leclist');
            $.ajax({
                url: '/College/proc/lectureListProc.jsp',
                method: "get",
                dataType: "json",
                success: function (data) {
                    console.log(data);
                    for(let lec of data){
                        let trTag = "<tr>"
                              + "<td>" + lec.lecNo + "</td>"
                              + "<td>" + lec.lecName + "</td>"
                              + "<td>" + lec.lecCredit + "</td>"
                              + "<td>" + lec.lecTime + "</td>"
                              + "<td>" + lec.lecClass + "</td>"
                              + "</tr>";
                        leclist.append(trTag);
                    }
                }
            });
        });

        $(document).on('click', '.submitBtn', function (e) {
            $(".insertInput").show();

        });

        $(document).on('click', '.closeBtn', function (e) {
            $(".insertInput").hide();
        });
```

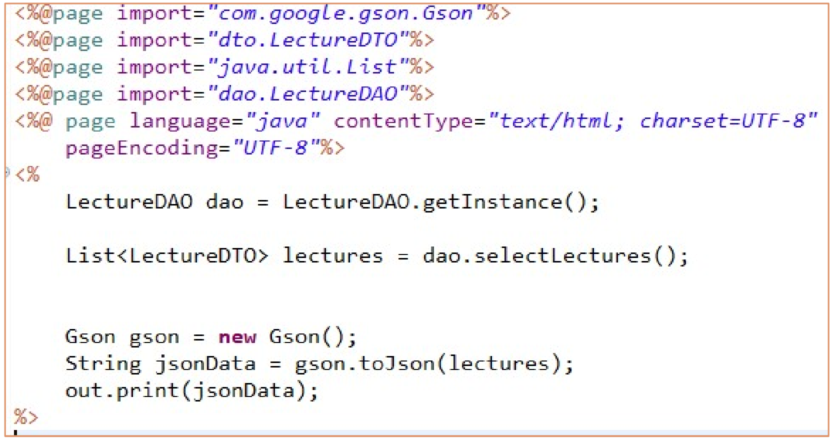


```
        $(document).on('click', '.insertBtn', function (e) {
            let lecNo = $('.insertInput input[name=lecNo]').val();
            let lecName = $('.insertInput input[name=lecName]').val();
            let lecCredit = $('.insertInput input[name=lecCredit]').val();
            let lecTime = $('.insertInput input[name=lecTime]').val();
            let lecClass = $('.insertInput input[name=lecClass]').val();

            let jsonData = {
                    "lecNo":lecNo,
                    "lecName":lecName,
                    "lecCredit":lecCredit,
                    "lecTime":lecTime,
                    "lecClass":lecClass
            }
            console.log(jsonData);

            $.ajax({
                url:'/College/proc/lectureInsertProc.jsp',
                    method: "post",
                    data: jsonData,
                    dataType: "json",
                    success: function (data) {
                        if(data.result == 1){
                            alert('등록 완료!');
                        }else {
                            alert('등록 실패!');
                        };
                        location.reload();
                    }
            })


        });


    });
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

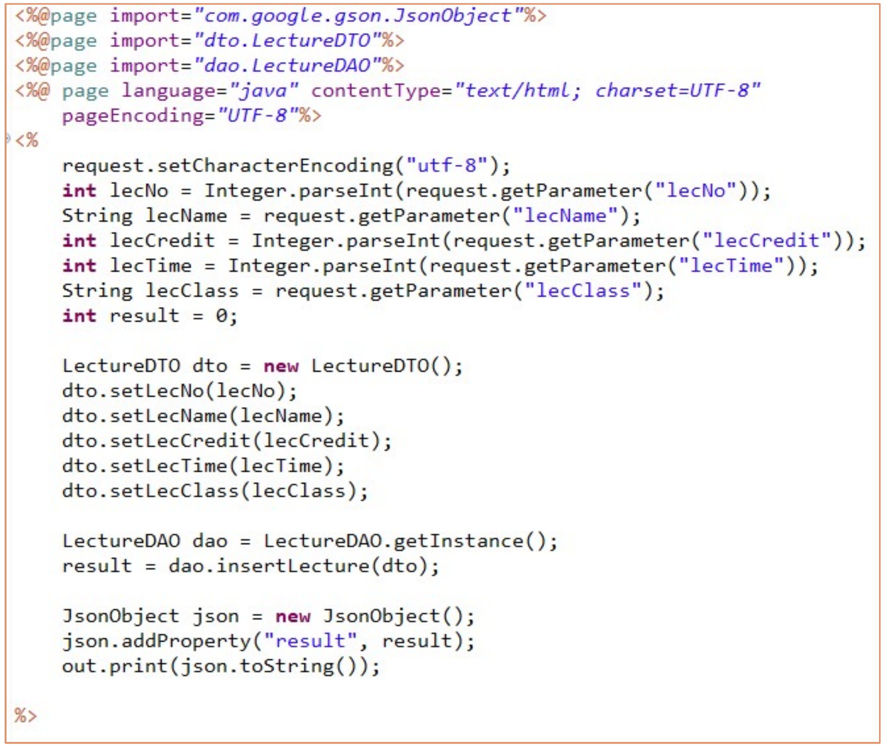
3. 문제 3 – 기능 구현 - lecture(강좌) 기본 CRUD_목록, 등록

lectureListProc.jsp



```
<%@page import="com.google.gson.Gson"%>
<%@page import="dto.LectureDTO"%>
<%@page import="java.util.List"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    LectureDAO dao = LectureDAO.getInstance();

    List<LectureDTO> lectures = dao.selectLectures();


    Gson gson = new Gson();
    String jsonData = gson.toJson(lectures);
    out.print(jsonData);
%>
```

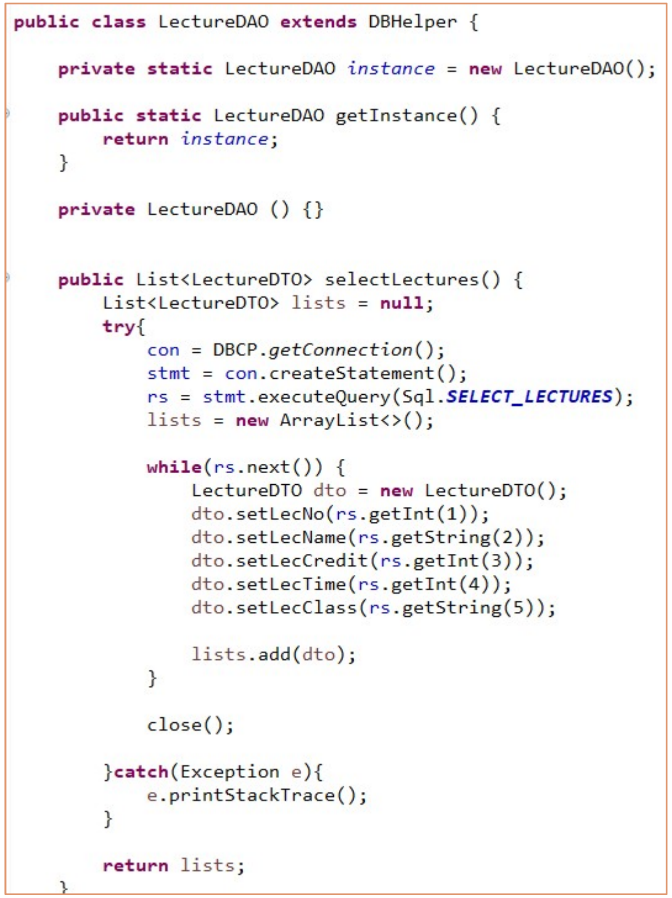
lecture ListProc.jsp



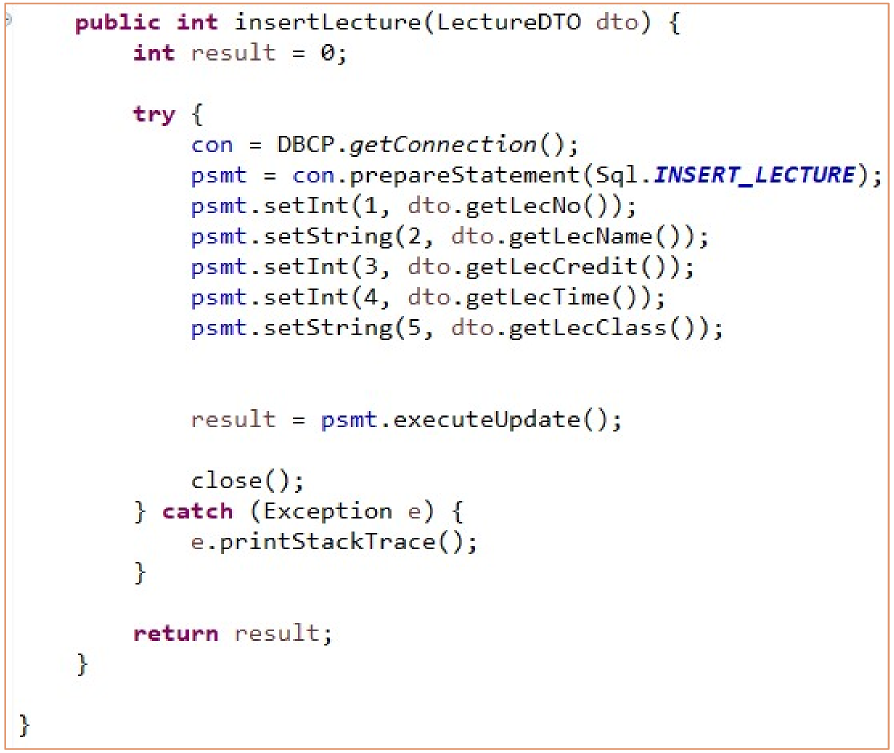
```
<%@page import="com.google.gson.JsonObject"%>
<%@page import="dto.LectureDTO"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
    int lecNo = Integer.parseInt(request.getParameter("lecNo"));
    String lecName = request.getParameter("lecName");
    int lecCredit = Integer.parseInt(request.getParameter("lecCredit"));
    int lecTime = Integer.parseInt(request.getParameter("lecTime"));
    String lecClass = request.getParameter("lecClass");
    int result = 0;

    LectureDTO dto = new LectureDTO();
    dto.setLecNo(lecNo);
    dto.setLecName(lecName);
    dto.setLecCredit(lecCredit);
    dto.setLecTime(lecTime);
    dto.setLecClass(lecClass);

    LectureDAO dao = LectureDAO.getInstance();
    result = dao.insertLecture(dto);

    JsonObject json = new JsonObject();
    json.addProperty("result", result);
    out.print(json.toString());

%>
```

3. 문제 3 – 기능 구현 - lecture(강좌) 기본 CRUD_목록, 등록

LectureDAO.java

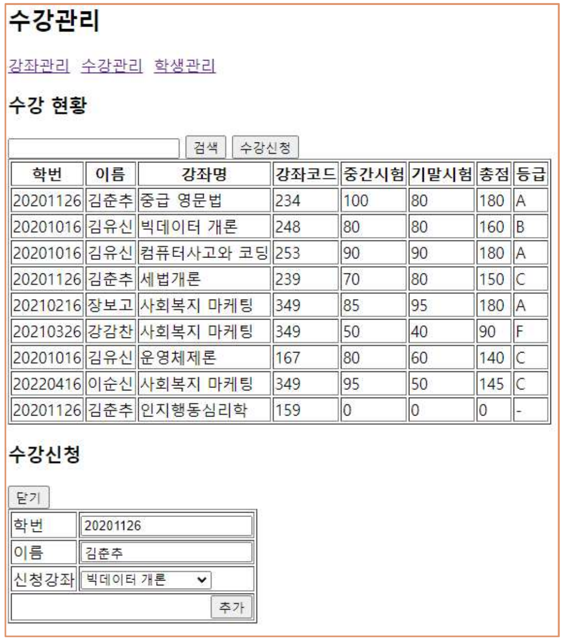


```
public class LectureDAO extends DBHelper {

    private static LectureDAO instance = new LectureDAO();

    public static LectureDAO getInstance() {
        return instance;
    }

    private LectureDAO () {}


    public List<LectureDTO> selectLectures() {
        List<LectureDTO> lists = null;
        try{
            con = DBCP.getConnection();
            stmt = con.createStatement();
            rs = stmt.executeQuery(Sql.SELECT_LECTURES);
            lists = new ArrayList<>();

            while(rs.next()) {
                LectureDTO dto = new LectureDTO();
                dto.setLecNo(rs.getInt(1));
                dto.setLecName(rs.getString(2));
                dto.setLecCredit(rs.getInt(3));
                dto.setLecTime(rs.getInt(4));
                dto.setLecClass(rs.getString(5));

                lists.add(dto);
            }

            close();

        }catch(Exception e){
            e.printStackTrace();
        }

        return lists;
    }
```



```
    public int insertLecture(LectureDTO dto) {
        int result = 0;

        try {
            con = DBCP.getConnection();
            psmt = con.prepareStatement(Sql.INSERT_LECTURE);
            psmt.setInt(1, dto.getLecNo());
            psmt.setString(2, dto.getLecName());
            psmt.setInt(3, dto.getLecCredit());
            psmt.setInt(4, dto.getLecTime());
            psmt.setString(5, dto.getLecClass());


            result = psmt.executeUpdate();

            close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return result;
    }

}
```

3. 문제 3 – 기능 구현 - register(수강) 기본 CRUD_목록, 등록



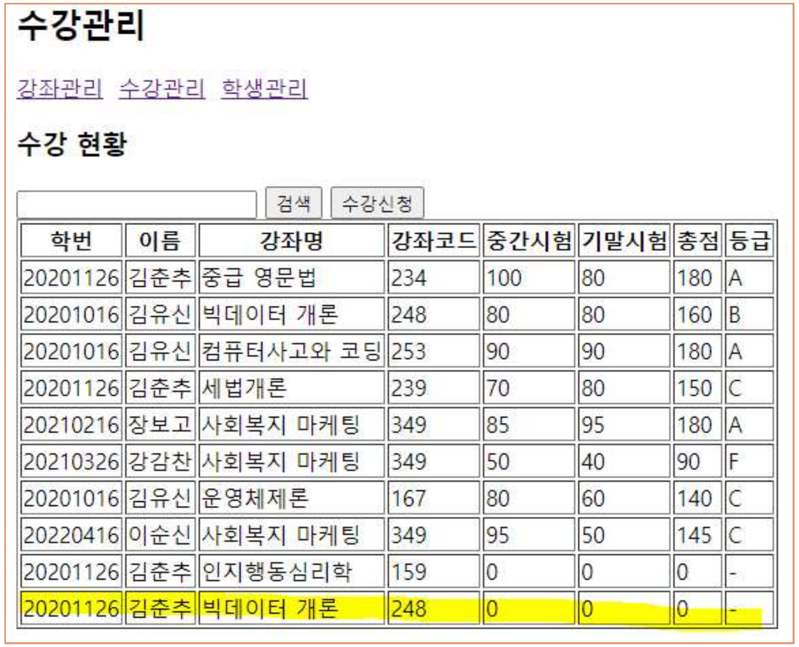
수강관리

수강 현황

학번	이름	강좌명	강좌코드	중간시험	기말시험	총점	등급
20201126	김춘추	중급 영문법	234	100	80	180	A
20201016	김유신	빅데이터 개론	248	80	80	160	B
20201016	김유신	컴퓨터사고와 코딩	253	90	90	180	A
20201126	김춘추	세법개론	239	70	80	150	C
20210216	장보고	사회복지 마케팅	349	85	95	180	A
20210326	강감찬	사회복지 마케팅	349	50	40	90	F
20201016	김유신	운영체제론	167	80	60	140	C
20220416	이순신	사회복지 마케팅	349	95	50	145	C
20201126	김춘추	인지행동심리학	159	0	0	0	-

수강신청

학번	20201126
이름	김춘추
신청강좌	빅데이터 개론

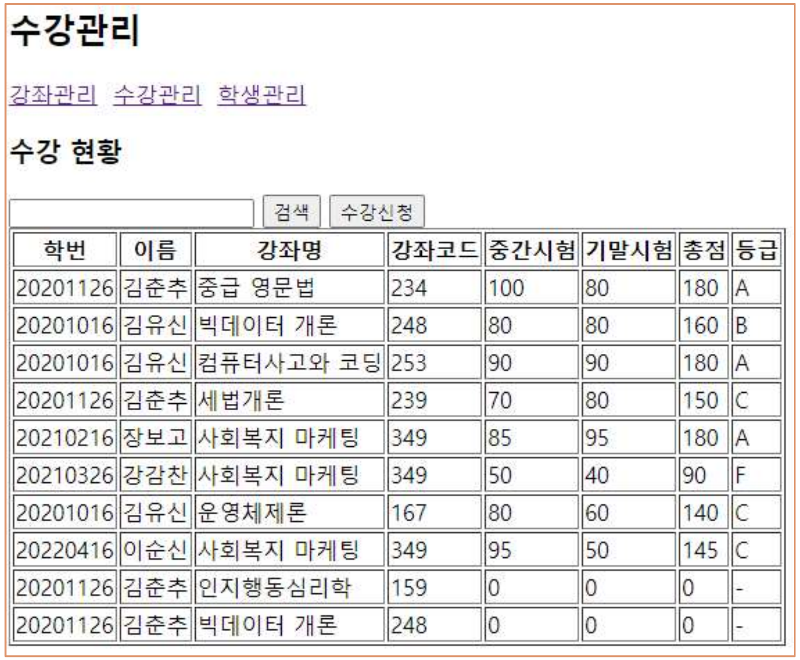


수강관리

수강 현황

학번	이름	강좌명	강좌코드	중간시험	기말시험	총점	등급
20201126	김춘추	중급 영문법	234	100	80	180	A
20201016	김유신	빅데이터 개론	248	80	80	160	B
20201016	김유신	컴퓨터사고와 코딩	253	90	90	180	A
20201126	김춘추	세법개론	239	70	80	150	C
20210216	장보고	사회복지 마케팅	349	85	95	180	A
20210326	강감찬	사회복지 마케팅	349	50	40	90	F
20201016	김유신	운영체제론	167	80	60	140	C
20220416	이순신	사회복지 마케팅	349	95	50	145	C
20201126	김춘추	인지행동심리학	159	0	0	0	-
20201126	김춘추	빅데이터 개론	248	0	0	0	-

3. 문제 3 – 기능 구현 - register(수강) 기본 CRUD_검색

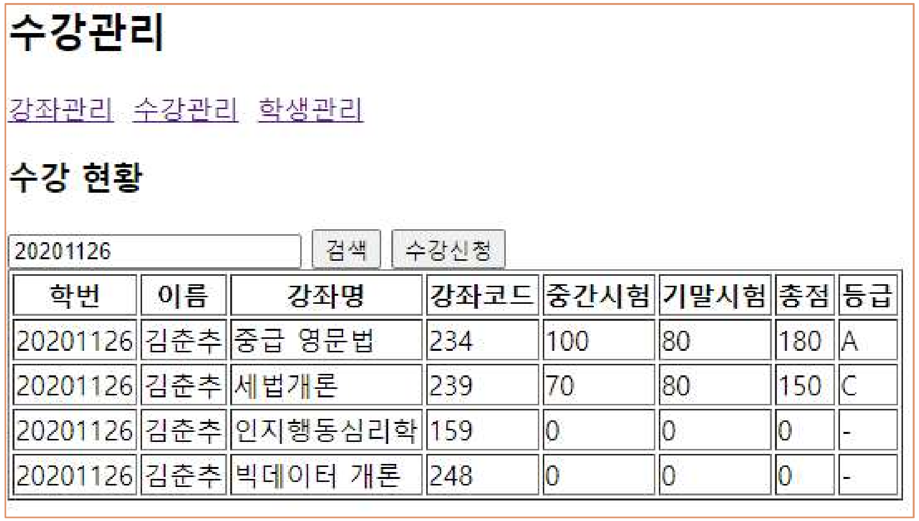


수강관리

강좌관리 수강관리 학생관리

수강 현황

학번	이름	강좌명	강좌코드	중간시험	기말시험	총점	등급
20201126	김춘추	중급 영문법	234	100	80	180	A
20201016	김유신	빅데이터 개론	248	80	80	160	B
20201016	김유신	컴퓨터사고와 코딩	253	90	90	180	A
20201126	김춘추	세법개론	239	70	80	150	C
20210216	장보고	사회복지 마케팅	349	85	95	180	A
20210326	강감찬	사회복지 마케팅	349	50	40	90	F
20201016	김유신	운영체제론	167	80	60	140	C
20220416	이순신	사회복지 마케팅	349	95	50	145	C
20201126	김춘추	인지행동심리학	159	0	0	0	-
20201126	김춘추	빅데이터 개론	248	0	0	0	-



수강관리

강좌관리 수강관리 학생관리

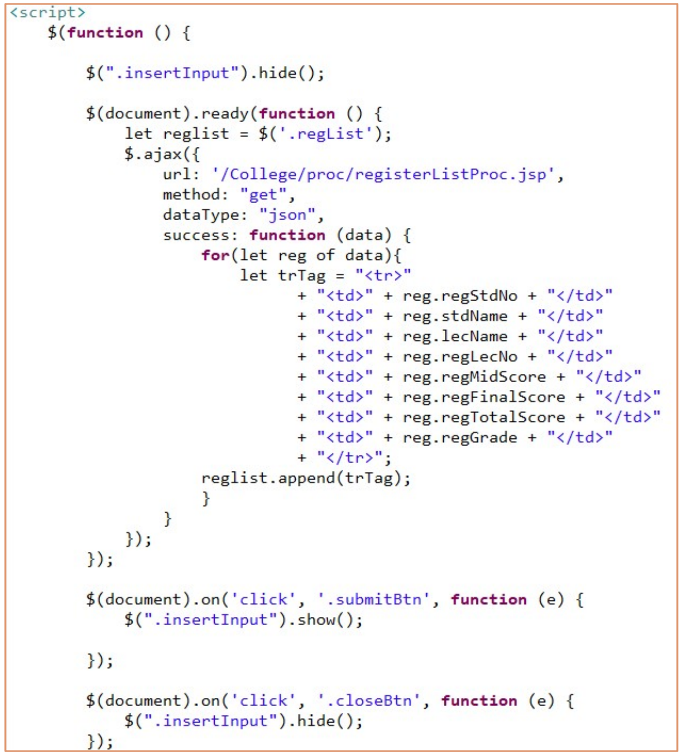
수강 현황

20201126

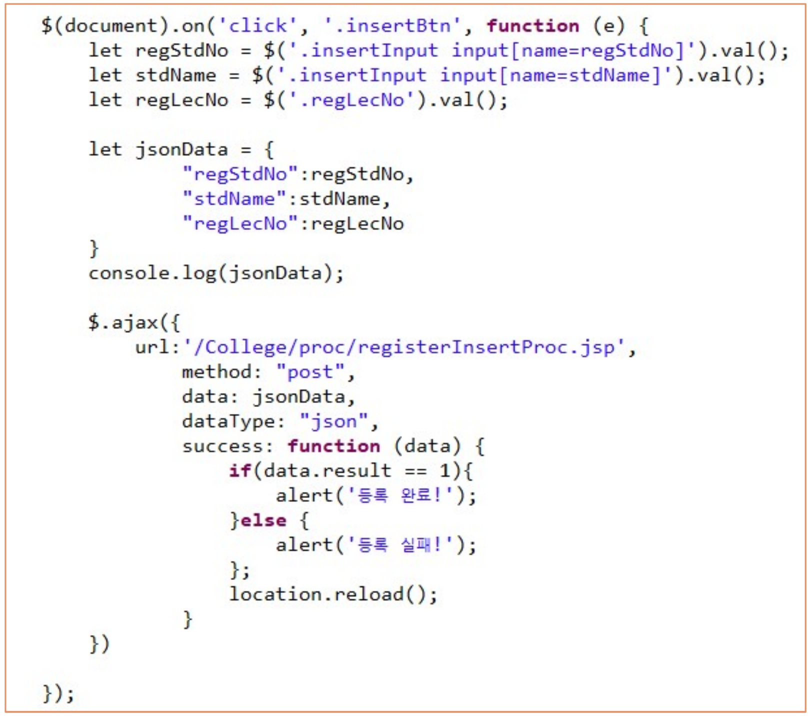
학번	이름	강좌명	강좌코드	중간시험	기말시험	총점	등급
20201126	김춘추	중급 영문법	234	100	80	180	A
20201126	김춘추	세법개론	239	70	80	150	C
20201126	김춘추	인지행동심리학	159	0	0	0	-
20201126	김춘추	빅데이터 개론	248	0	0	0	-

3. 문제 3 – 기능 구현 – register(수강) 기본 CRUD_목록, 등록

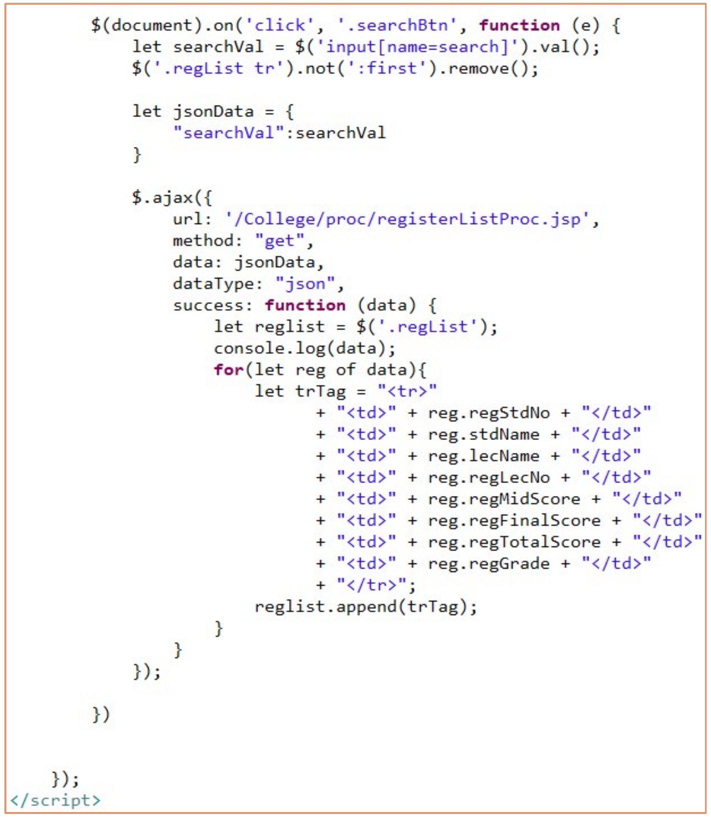
register.jsp 스크립트 부분



```
<script>
    $(function () {

        $(".insertInput").hide();

        $(document).ready(function () {
            let reglist = $('.regList');
            $.ajax({
                url: '/College/proc/registerListProc.jsp',
                method: "get",
                dataType: "json",
                success: function (data) {
                    for(let reg of data){
                        let trTag = "<tr>"
                              + "<td>" + reg.regStdNo + "</td>"
                              + "<td>" + reg.stdName + "</td>"
                              + "<td>" + reg.lecName + "</td>"
                              + "<td>" + reg.regLecNo + "</td>"
                              + "<td>" + reg.regMidScore + "</td>"
                              + "<td>" + reg.regFinalScore + "</td>"
                              + "<td>" + reg.regTotalScore + "</td>"
                              + "<td>" + reg.regGrade + "</td>"
                              + "</tr>";
                    reglist.append(trTag);
                    }
                }
            });
        });

        $(document).on('click', '.submitBtn', function (e) {
            $(".insertInput").show();

        });

        $(document).on('click', '.closeBtn', function (e) {
            $(".insertInput").hide();
        });
```



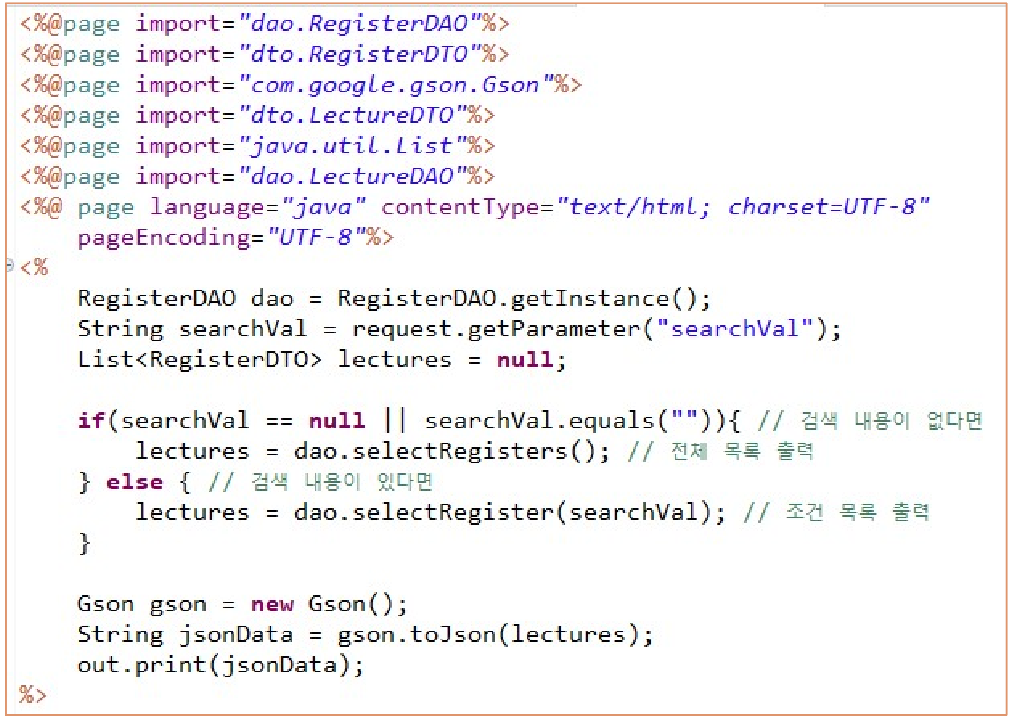
```
    $(document).on('click', '.insertBtn', function (e) {
        let regStdNo = $('.insertInput input[name=regStdNo]').val();
        let stdName = $('.insertInput input[name=stdName]').val();
        let regLecNo = $('.regLecNo').val();

        let jsonData = {
                "regStdNo":regStdNo,
                "stdName":stdName,
                "regLecNo":regLecNo
        }
        console.log(jsonData);

        $.ajax({
            url:'/College/proc/registerInsertProc.jsp',
                method: "post",
                data: jsonData,
                dataType: "json",
                success: function (data) {
                    if(data.result == 1){
                        alert('등록 완료!');
                    }else {
                        alert('등록 실패!');
                    };
                    location.reload();
                }
        })

    });
```



```
        $(document).on('click', '.searchBtn', function (e) {
            let searchVal = $('input[name=search]').val();
            $('.regList tr').not(':first').remove();

            let jsonData = {
                "searchVal":searchVal
            }

            $.ajax({
                url: '/College/proc/registerListProc.jsp',
                method: "get",
                data: jsonData,
                dataType: "json",
                success: function (data) {
                    let reglist = $('.regList');
                    console.log(data);
                    for(let reg of data){
                        let trTag = "<tr>"
                              + "<td>" + reg.regStdNo + "</td>"
                              + "<td>" + reg.stdName + "</td>"
                              + "<td>" + reg.lecName + "</td>"
                              + "<td>" + reg.regLecNo + "</td>"
                              + "<td>" + reg.regMidScore + "</td>"
                              + "<td>" + reg.regFinalScore + "</td>"
                              + "<td>" + reg.regTotalScore + "</td>"
                              + "<td>" + reg.regGrade + "</td>"
                              + "</tr>";
                        reglist.append(trTag);
                    }
                }
            });

        })


    });
</script>
```

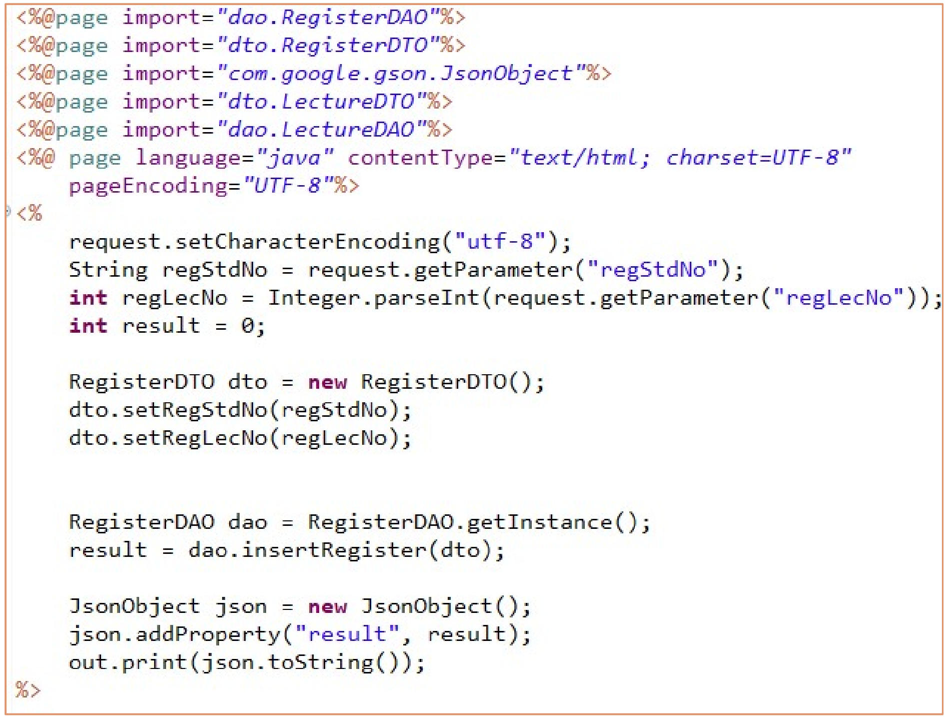
3. 문제 3 – 기능 구현 - register(수강) 기본 CRUD_목록, 등록

registerListProc.jsp



```
<%@page import="dao.RegisterDAO"%>
<%@page import="dto.RegisterDTO"%>
<%@page import="com.google.gson.Gson"%>
<%@page import="dto.LectureDTO"%>
<%@page import="java.util.List"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    RegisterDAO dao = RegisterDAO.getInstance();
    String searchVal = request.getParameter("searchVal");
    List<RegisterDTO> lectures = null;

    if(searchVal == null || searchVal.equals("")){ // 검색 내용이 없다면
        lectures = dao.selectRegisters(); // 전체 목록 출력
    } else { // 검색 내용이 있다면
        lectures = dao.selectRegister(searchVal); // 조건 목록 출력
    }

    Gson gson = new Gson();
    String jsonData = gson.toJson(lectures);
    out.print(jsonData);
%>
```

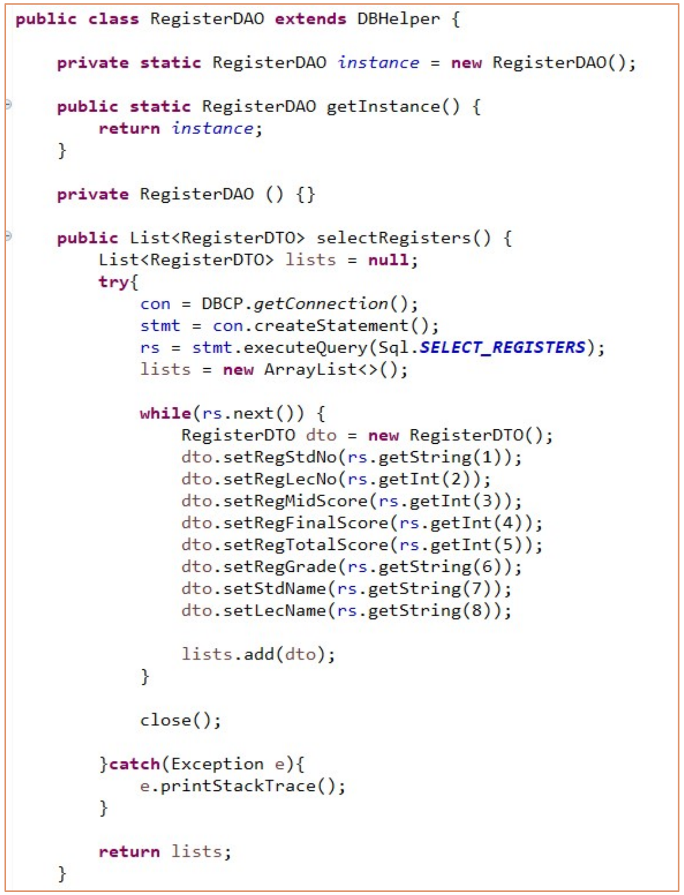
registerListProc.jsp



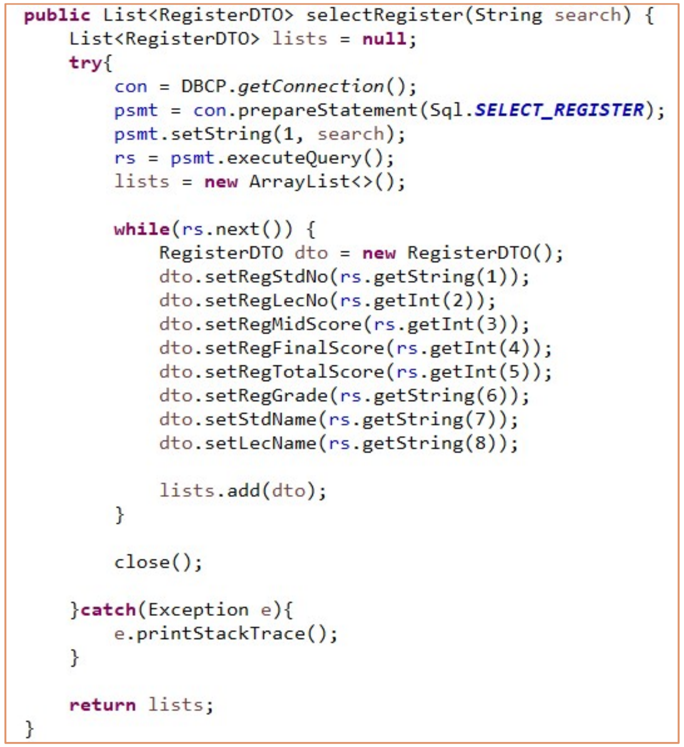
```
<%@page import="dao.RegisterDAO"%>
<%@page import="dto.RegisterDTO"%>
<%@page import="com.google.gson.JsonObject"%>
<%@page import="dto.LectureDTO"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
    String regStdNo = request.getParameter("regStdNo");
    int regLecNo = Integer.parseInt(request.getParameter("regLecNo"));
    int result = 0;

    RegisterDTO dto = new RegisterDTO();
    dto.setRegStdNo(regStdNo);
    dto.setRegLecNo(regLecNo);


    RegisterDAO dao = RegisterDAO.getInstance();
    result = dao.insertRegister(dto);

    JsonObject json = new JsonObject();
    json.addProperty("result", result);
    out.print(json.toString());
%>
```

3. 문제 3 – 기능 구현 – register(수강) 기본 CRUD_목록, 등록

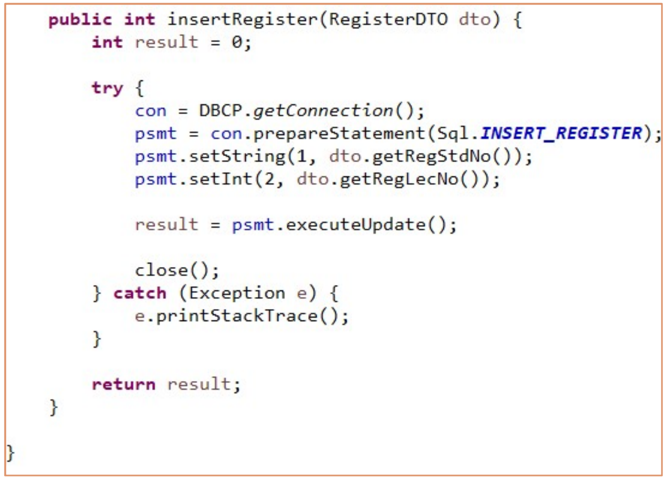
registerDAO.java



```java
public class RegisterDAO extends DBHelper {

    private static RegisterDAO instance = new RegisterDAO();

    public static RegisterDAO getInstance() {
        return instance;
    }

    private RegisterDAO () {}

    public List<RegisterDTO> selectRegisters() {
        List<RegisterDTO> lists = null;
        try{
            con = DBCP.getConnection();
            stmt = con.createStatement();
            rs = stmt.executeQuery(Sql.SELECT_REGISTERS);
            lists = new ArrayList<>();

            while(rs.next()) {
                RegisterDTO dto = new RegisterDTO();
                dto.setRegStdNo(rs.getString(1));
                dto.setRegLecNo(rs.getInt(2));
                dto.setRegMidScore(rs.getInt(3));
                dto.setRegFinalScore(rs.getInt(4));
                dto.setRegTotalScore(rs.getInt(5));
                dto.setRegGrade(rs.getString(6));
                dto.setStdName(rs.getString(7));
                dto.setLecName(rs.getString(8));

                lists.add(dto);
            }

            close();

        }catch(Exception e){
            e.printStackTrace();
        }

        return lists;
    }
```

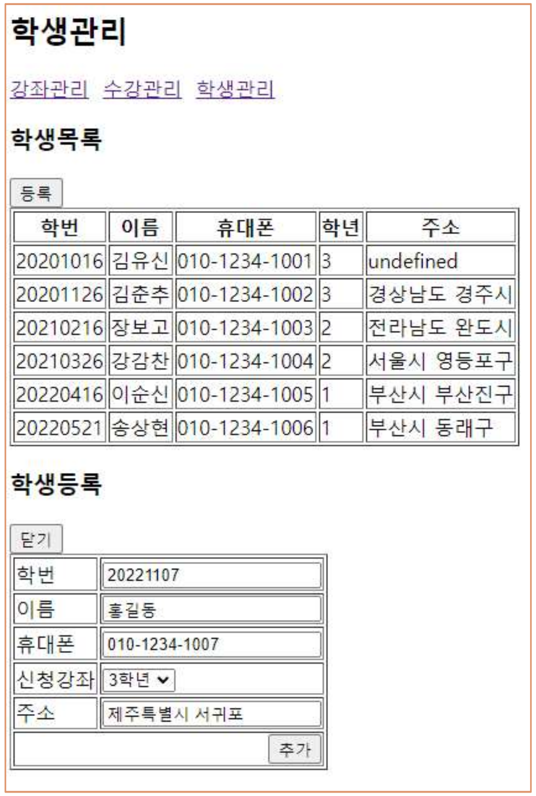


```java
public List<RegisterDTO> selectRegister(String search) {
    List<RegisterDTO> lists = null;
    try{
        con = DBCP.getConnection();
        psmt = con.prepareStatement(Sql.SELECT_REGISTER);
        psmt.setString(1, search);
        rs = psmt.executeQuery();
        lists = new ArrayList<>();

        while(rs.next()) {
            RegisterDTO dto = new RegisterDTO();
            dto.setRegStdNo(rs.getString(1));
            dto.setRegLecNo(rs.getInt(2));
            dto.setRegMidScore(rs.getInt(3));
            dto.setRegFinalScore(rs.getInt(4));
            dto.setRegTotalScore(rs.getInt(5));
            dto.setRegGrade(rs.getString(6));
            dto.setStdName(rs.getString(7));
            dto.setLecName(rs.getString(8));

            lists.add(dto);
        }

        close();

    }catch(Exception e){
        e.printStackTrace();
    }

    return lists;
}
```



```java
    public int insertRegister(RegisterDTO dto) {
        int result = 0;

        try {
            con = DBCP.getConnection();
            psmt = con.prepareStatement(Sql.INSERT_REGISTER);
            psmt.setString(1, dto.getRegStdNo());
            psmt.setInt(2, dto.getRegLecNo());

            result = psmt.executeUpdate();

            close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return result;
    }

}
```

3. 문제 3 – 기능 구현 - student(학생) 기본 CRUD_목록, 등록



학생관리

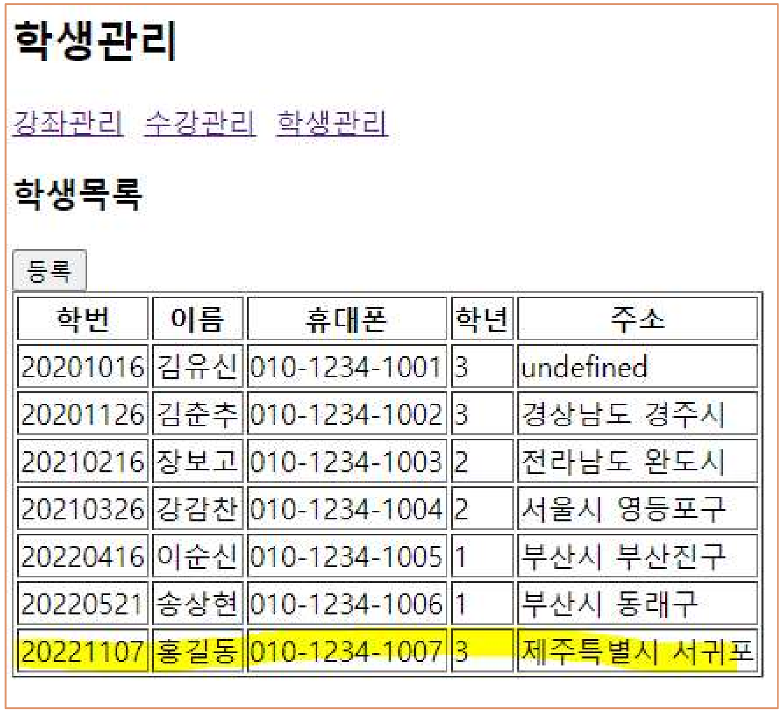
강좌관리 수강관리 학생관리

학생목록

학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	undefined
20201126	김춘추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구

학생등록

학번	20221107
이름	홍길동
휴대폰	010-1234-1007
신청강좌	3학년
주소	제주특별시 서귀포



학생관리

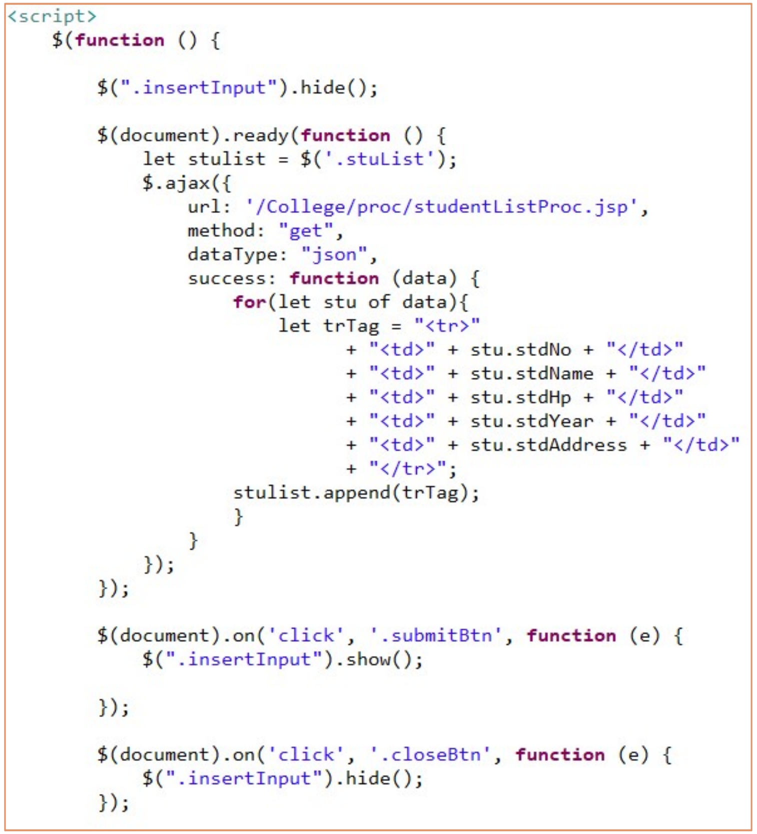
강좌관리 수강관리 학생관리

학생목록

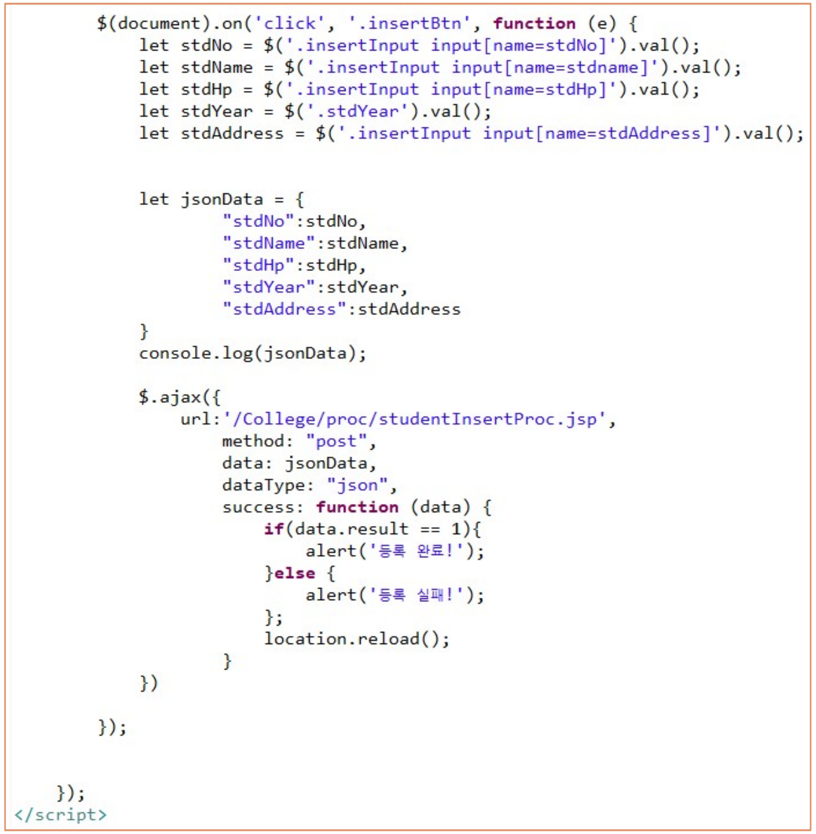
학번	이름	휴대폰	학년	주소
20201016	김유신	010-1234-1001	3	undefined
20201126	김춘추	010-1234-1002	3	경상남도 경주시
20210216	장보고	010-1234-1003	2	전라남도 완도시
20210326	강감찬	010-1234-1004	2	서울시 영등포구
20220416	이순신	010-1234-1005	1	부산시 부산진구
20220521	송상현	010-1234-1006	1	부산시 동래구
20221107	홍길동	010-1234-1007	3	제주특별시 서귀포

3. 문제 3 – 기능 구현 – student(학생) 기본 CRUD_목록, 등록

student.jsp 스크립트 부분



```
<script>
    $(function () {

        $(".insertInput").hide();

        $(document).ready(function () {
            let stulist = $('.stuList');
            $.ajax({
                url: '/College/proc/studentListProc.jsp',
                method: "get",
                dataType: "json",
                success: function (data) {
                    for(let stu of data){
                        let trTag = "<tr>"
                              + "<td>" + stu.stdNo + "</td>"
                              + "<td>" + stu.stdName + "</td>"
                              + "<td>" + stu.stdHp + "</td>"
                              + "<td>" + stu.stdYear + "</td>"
                              + "<td>" + stu.stdAddress + "</td>"
                              + "</tr>";
                    stulist.append(trTag);
                    }
                }
            });
        });

        $(document).on('click', '.submitBtn', function (e) {
            $(".insertInput").show();

        });

        $(document).on('click', '.closeBtn', function (e) {
            $(".insertInput").hide();
        });
```

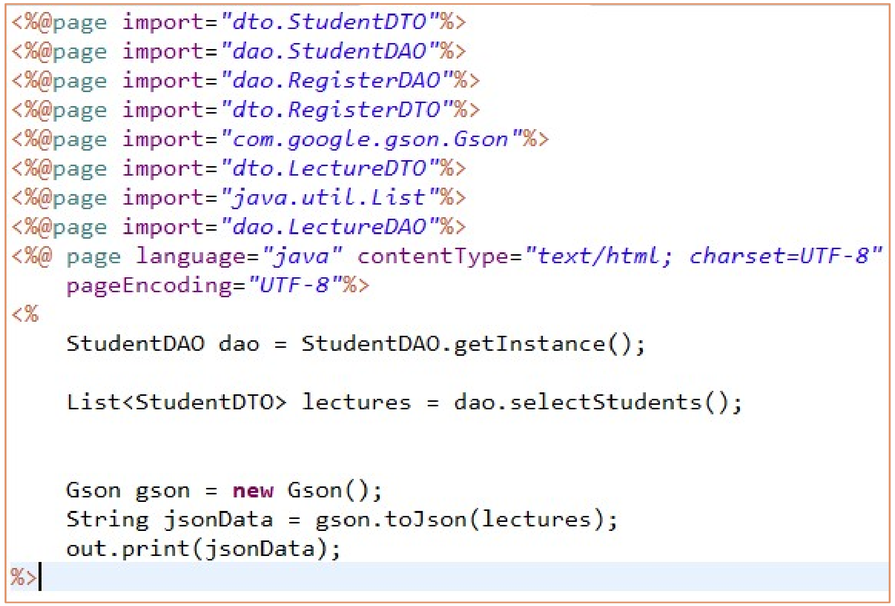


```
        $(document).on('click', '.insertBtn', function (e) {
            let stdNo = $('.insertInput input[name=stdNo]').val();
            let stdName = $('.insertInput input[name=stdname]').val();
            let stdHp = $('.insertInput input[name=stdHp]').val();
            let stdYear = $('.stdYear').val();
            let stdAddress = $('.insertInput input[name=stdAddress]').val();


            let jsonData = {
                    "stdNo":stdNo,
                    "stdName":stdName,
                    "stdHp":stdHp,
                    "stdYear":stdYear,
                    "stdAddress":stdAddress
            }
            console.log(jsonData);

            $.ajax({
                url:'/College/proc/studentInsertProc.jsp',
                    method: "post",
                    data: jsonData,
                    dataType: "json",
                    success: function (data) {
                        if(data.result == 1){
                            alert('등록 완료!');
                        }else {
                            alert('등록 실패!');
                        };
                        location.reload();
                    }
            })

        });


    });
</script>
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

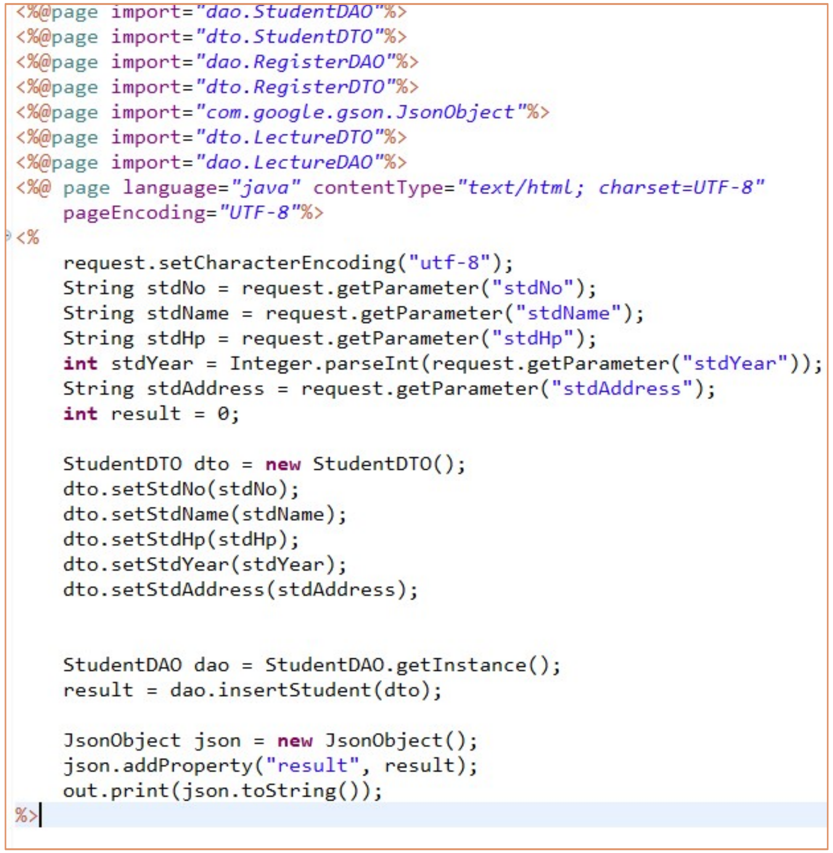
3. 문제 3 – 기능 구현 - student(학생) 기본 CRUD_목록, 등록

studentListProc.jsp



```
<%@page import="dto.StudentDTO"%>
<%@page import="dao.StudentDAO"%>
<%@page import="dao.RegisterDAO"%>
<%@page import="dto.RegisterDTO"%>
<%@page import="com.google.gson.Gson"%>
<%@page import="dto.LectureDTO"%>
<%@page import="java.util.List"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    StudentDAO dao = StudentDAO.getInstance();

    List<StudentDTO> lectures = dao.selectStudents();


    Gson gson = new Gson();
    String jsonData = gson.toJson(lectures);
    out.print(jsonData);
%>
```

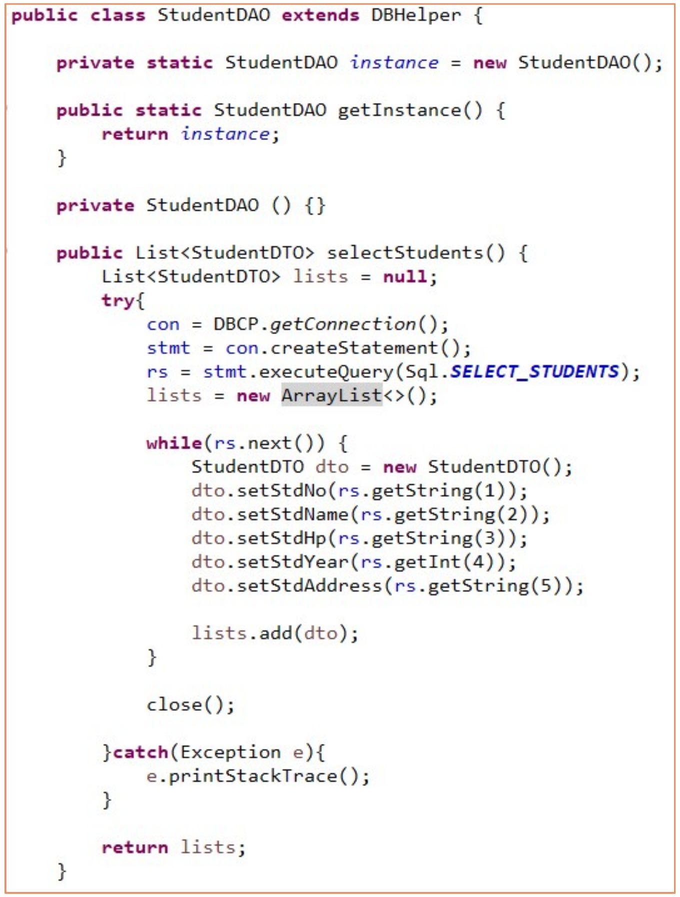
studentListProc.jsp



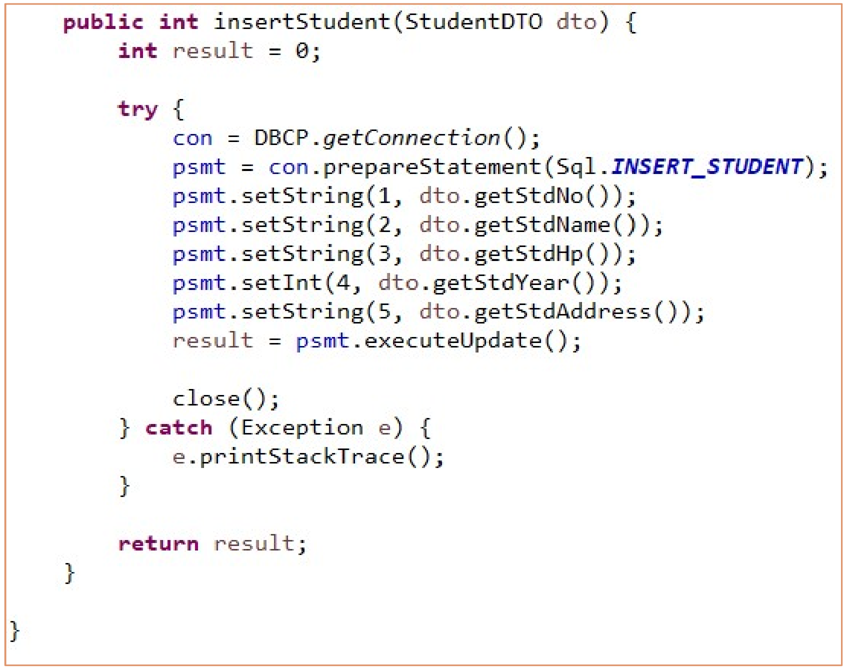
```
<%@page import="dao.StudentDAO"%>
<%@page import="dto.StudentDTO"%>
<%@page import="dao.RegisterDAO"%>
<%@page import="dto.RegisterDTO"%>
<%@page import="com.google.gson.JsonObject"%>
<%@page import="dto.LectureDTO"%>
<%@page import="dao.LectureDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
    String stdNo = request.getParameter("stdNo");
    String stdName = request.getParameter("stdName");
    String stdHp = request.getParameter("stdHp");
    int stdYear = Integer.parseInt(request.getParameter("stdYear"));
    String stdAddress = request.getParameter("stdAddress");
    int result = 0;

    StudentDTO dto = new StudentDTO();
    dto.setStdNo(stdNo);
    dto.setStdName(stdName);
    dto.setStdHp(stdHp);
    dto.setStdYear(stdYear);
    dto.setStdAddress(stdAddress);


    StudentDAO dao = StudentDAO.getInstance();
    result = dao.insertStudent(dto);

    JsonObject json = new JsonObject();
    json.addProperty("result", result);
    out.print(json.toString());
%>
```

3. 문제 3 – 기능 구현 – student(학생) 기본 CRUD_목록, 등록

studentDAO.java



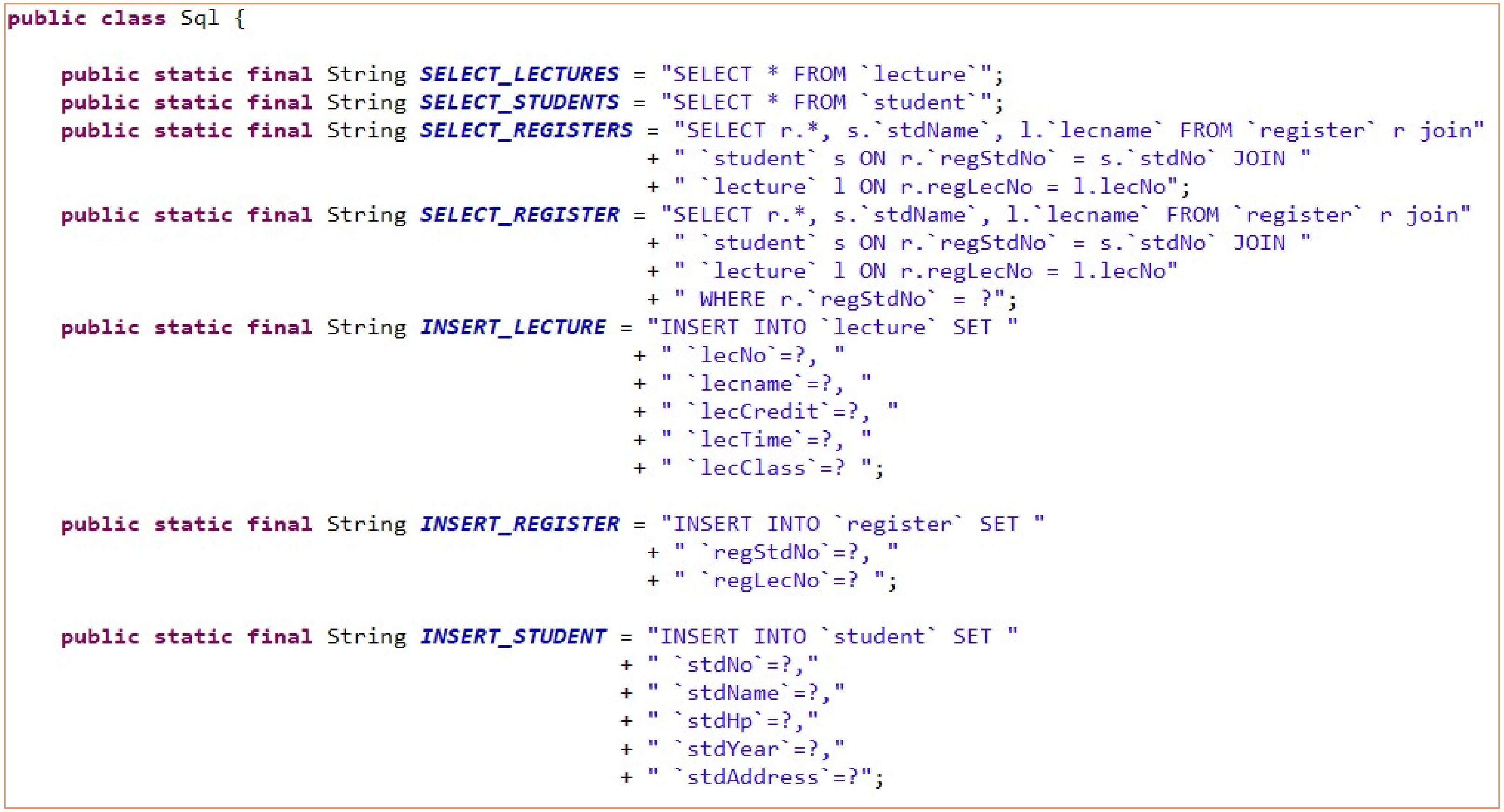
```java
public class StudentDAO extends DBHelper {

    private static StudentDAO instance = new StudentDAO();

    public static StudentDAO getInstance() {
        return instance;
    }

    private StudentDAO () {}

    public List<StudentDTO> selectStudents() {
        List<StudentDTO> lists = null;
        try{
            con = DBCP.getConnection();
            stmt = con.createStatement();
            rs = stmt.executeQuery(Sql.SELECT_STUDENTS);
            lists = new ArrayList<>();

            while(rs.next()) {
                StudentDTO dto = new StudentDTO();
                dto.setStdNo(rs.getString(1));
                dto.setStdName(rs.getString(2));
                dto.setStdHp(rs.getString(3));
                dto.setStdYear(rs.getInt(4));
                dto.setStdAddress(rs.getString(5));

                lists.add(dto);
            }

            close();

        }catch(Exception e){
            e.printStackTrace();
        }

        return lists;
    }
```



```java
    public int insertStudent(StudentDTO dto) {
        int result = 0;

        try {
            con = DBCP.getConnection();
            psmt = con.prepareStatement(Sql.INSERT_STUDENT);
            psmt.setString(1, dto.getStdNo());
            psmt.setString(2, dto.getStdName());
            psmt.setString(3, dto.getStdHp());
            psmt.setInt(4, dto.getStdYear());
            psmt.setString(5, dto.getStdAddress());
            result = psmt.executeUpdate();

            close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return result;
    }

}
```

3. 문제 3 – 기능 구현 – SQL

Sql.java



```
public class Sql {

    public static final String SELECT_LECTURES = "SELECT * FROM `lecture`";
    public static final String SELECT_STUDENTS = "SELECT * FROM `student`";
    public static final String SELECT_REGISTERS = "SELECT r.*, s.`stdName`, l.`lecname` FROM `register` r join"
                                                + " `student` s ON r.`regStdNo` = s.`stdNo` JOIN "
                                                + " `lecture` l ON r.regLecNo = l.lecNo";
    public static final String SELECT_REGISTER = "SELECT r.*, s.`stdName`, l.`lecname` FROM `register` r join"
                                                + " `student` s ON r.`regStdNo` = s.`stdNo` JOIN "
                                                + " `lecture` l ON r.regLecNo = l.lecNo"
                                                + " WHERE r.`regStdNo` = ?";
    public static final String INSERT_LECTURE = "INSERT INTO `lecture` SET "
                                               + " `lecNo`=?, "
                                               + " `lecname`=?, "
                                               + " `lecCredit`=?, "
                                               + " `lecTime`=?, "
                                               + " `lecClass`=? ";

    public static final String INSERT_REGISTER = "INSERT INTO `register` SET "
                                                + " `regStdNo`=?, "
                                                + " `regLecNo`=? ";

    public static final String INSERT_STUDENT = "INSERT INTO `student` SET "
                                              + " `stdNo`=?,"
                                              + " `stdName`=?,"
                                              + " `stdHp`=?,"
                                              + " `stdYear`=?,"
                                              + " `stdAddress`=?";
```

4. 문제 4 – 실행

실행부분은 문제 3에 작성 했습니다.

감사합니다