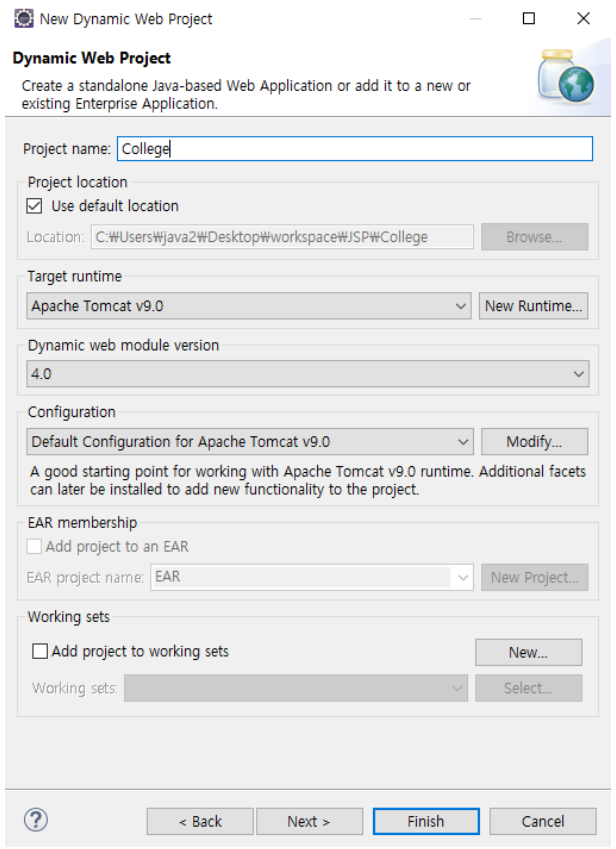
네트워크 프로그래밍 구현

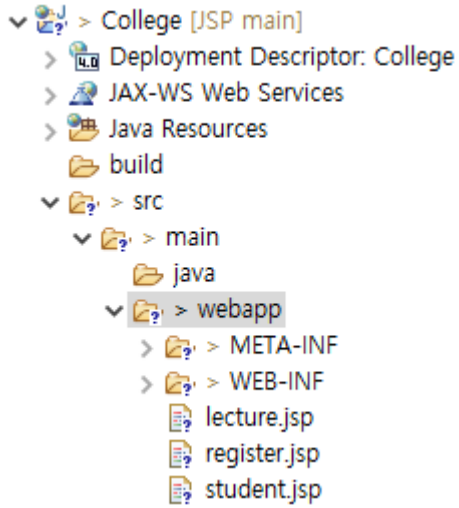
데이터베이스 & SQL

심규영

목차

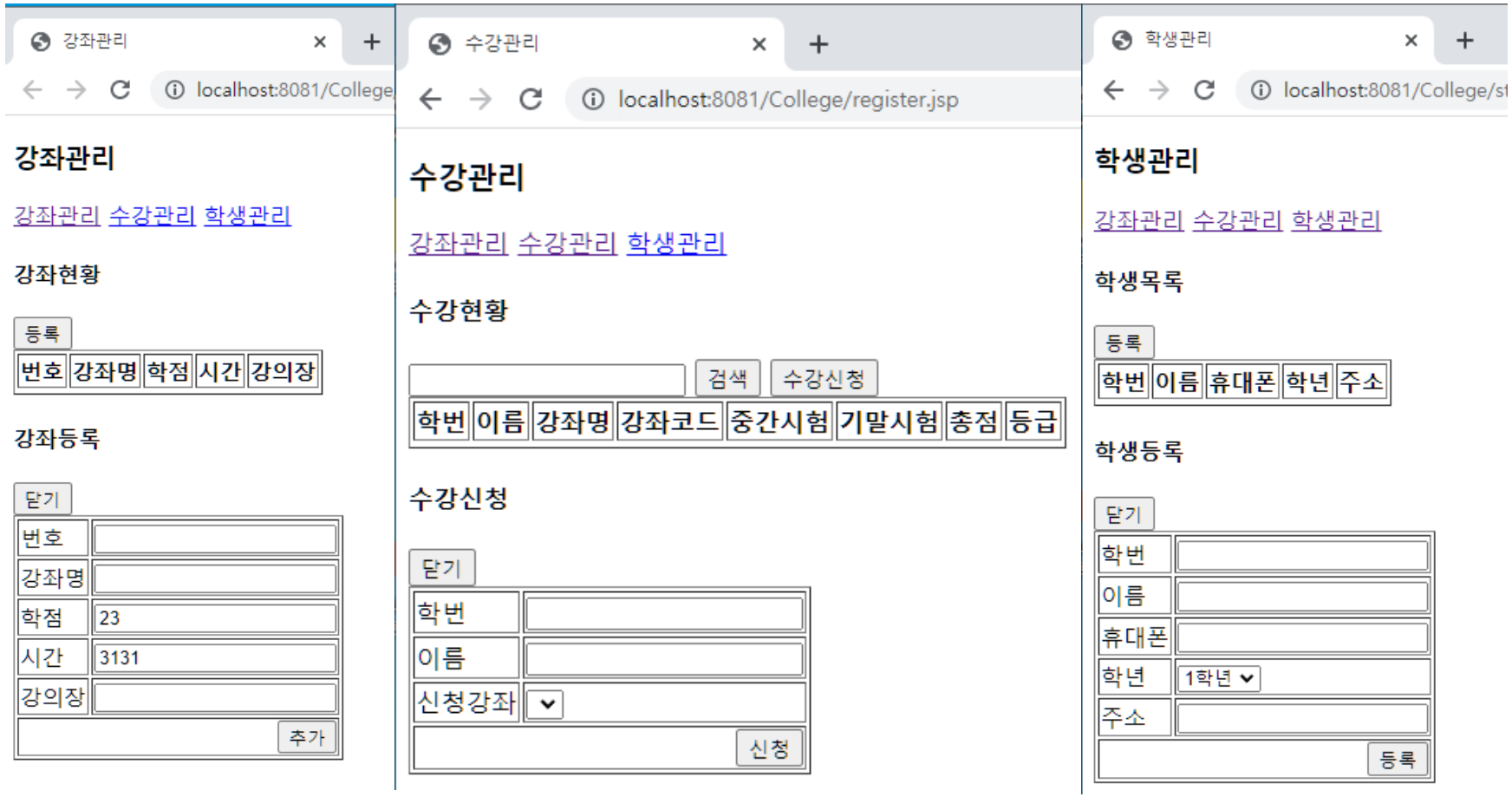
문제1) 프로젝트 생성 및 구현





- 프로젝트명 : College
- WAS : Tomcat 9
- DB : Java2_college
- 개발도구 : Eclipse, Workbench

문제2) 화면 구현



- 강좌관리 화면 구현
- 수강관리 화면 구현
- 학생관리 화면 구현

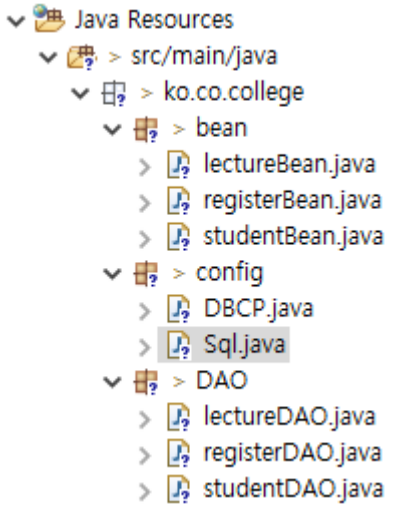
문제3) 기능 구현

```java
public class DBCP {
    protected Connection conn;
    protected PreparedStatement psmt;
    protected Statement stmt;
    protected ResultSet rs;

    protected Logger logger = LoggerFactory.getLogger(this.getClass());

    private static DataSource ds = null;
    public static Connection getConnection() throws NamingException, SQLException {
        if (ds == null) {
            ds = (DataSource) new InitialContext()
                    .lookup("java:comp/env/dbcp_java2_college");
        }
        return ds.getConnection();
    }

    /**
     * 클래스 종료
     */
    public void close() {
        try {
            if(conn!=null) conn.close();
            if(stmt!=null) stmt.close();
            if(psmt!=null) psmt.close();
            if(rs!=null) rs.close();
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("클래스 닫기 오류");
            logger.error(e.getMessage());
        }
    }
}
```

```xml
<!-- java2_college 커넥션 풀 전역 설정 -->
<Resource auth="Container"
driverClassName="com.mysql.cj.jdbc.Driver"
maxIdle="16" maxTotal="16" maxWaitMillis="3000"
name="dbcp_java2_college" password="1234"
type="javax.sql.DataSource"
url="jdbc:mysql://127.0.0.1:3306/java2_college"
username="root"/>
<!-- java2_college 커넥션 풀 자원 설정 -->
<ResourceLink
    global="dbcp_java2_college"
    name="dbcp_java2_college"
    type="javax.sql.DataSource"
/>
```

- 커넥션 풀 설정
- DBCP.class 생성

문제3) 기능 구현




```java
package ko.co.college.bean;

public class lectureBean {
    private int lecNo;
    private String lecName;
    private int lecCredit;
    private int lecTime;
    private String lecClass;
    public int getLecNo() {
        return lecNo;
    }
    public void setLecNo(int lecNo) {
        this.lecNo = lecNo;
    }
    public String getLecName() {
        return lecName;
    }
    public void setLecName(String lecName) {
```

```java
package ko.co.college.DAO;

import ko.co.college.config.DBCP;

public class lectureDAO extends DBCP {
    public lectureDAO () {
        try {
            conn = getConnection();
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("lecture 연결 오류");
            logger.error(e.getMessage());
        }
    }

    // Create

    // read

    // upload

    // delete
}
```

- 각각의 Bean 과 DAO 생성
- Sql.java 생성

```java
public class Sql {
    // lecture
    public static final String SELECT_LECTURE_LIST =
            "";

    // register

    // student
}
```

문제3) 기능 구현

- 강좌현황 보기 기능 구현
- 강좌 등록 숨기기
- 강좌 등록 DAO 기능 추가

```
/**
 * 강좌 리스트 불러오기
 * @return List<lectureBean>
 */
public List<lectureBean> readLectureList() {
    List<lectureBean> lecturelist = new ArrayList<lectureBean>();
    try {
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_LECTURE_LIST);
        while(rs.next()) {
            lectureBean lb = new lectureBean();
            lb.setLecNo(rs.getInt("lecNo"));
            lb.setLecName(rs.getString("lecName"));
            lb.setLecCredit(rs.getInt("lecCredit"));
            lb.setLecTime(rs.getInt("lecTime"));
            lb.setLecClass(rs.getString("lecClass"));
            lecturelist.add(lb);
        }
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("lecture list 불러오기 오류");
        logger.error(e.getMessage());
    }
    return lecturelist;
}
```

```
// lecture
public static final String SELECT_LECTURE_LIST =
        "select "
        + "`lecNo`, "
        + "`lecName`, "
        + "`lecCredit`, "
        + "`lecTime`, "
        + "`lecClass`"
        + "from `lecture`";
```

```
<section style="display:none;">
    <h4>강좌등록</h4>
    <input type="button" value="닫기">
    <table border="1">
        <tr>
```

```
// Create

/**
 * 강좌 등록
 * @param lb
 * @return either (1) the row count for SQL Data Manipula
 */
public int insertLecture(lectureBean lb) {
    int result = 0;
    try {
        psmt = conn.prepareStatement(Sql.INSERT_LECTURE);
        psmt.setInt(1, lb.getLecNo());
        psmt.setString(2, lb.getLecName());
        psmt.setInt(1, lb.getLecCredit());
        psmt.setInt(1, lb.getLecTime());
        psmt.setString(1, lb.getLecClass());
        result = psmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("lecture 등록 오류");
        logger.error(e.getMessage());
    }
    return result;
}
```

문제3) 기능 구현

```
    request.setCharacterEncoding("UTF-8");
    String lecNo = request.getParameter("lecNo");
    String lecName = request.getParameter("lecName");
    String lecCredit = request.getParameter("lecCredit");
    String lecTime = request.getParameter("lecTime");
    String lecClass = request.getParameter("lecClass");

    lectureBean lb = new lectureBean();
    lb.setLecNo(Integer.parseInt(lecNo));
    lb.setLecName(lecName);
    lb.setLecCredit(Integer.parseInt(lecCredit));
    lb.setLecTime(Integer.parseInt(lecTime));
    lb.setLecClass(lecClass);

    lectureDAO ldao = new lectureDAO();
    int result = ldao.insertLecture(lb);
    ldao.close();

    JsonObject json = new JsonObject();
    json.addProperty("result", result);
    json.addProperty("lecNo", lecNo);
    json.addProperty("lecName", lecName);
    json.addProperty("lecCredit", lecCredit);
    json.addProperty("lecTime", lecTime);
    json.addProperty("lecClass", lecClass);

    String jsonData = json.toString();

    out.print(jsonData);
```

```
$('.btnLectureOpen').click(function(){
    $('section').show();
});
$('.btnLectureClose').click(function(){
    $('section').hide();
});
```

```
$('input[type=submit]').click(function(){
    const lecNo     = $('input[name=lecNo]').val();
    const lecName   = $('input[name=lecName]').val();
    const lecCredit = $('input[name=lecCredit]').val();
    const lecTime   = $('input[name=lecTime]').val();
    const lecClass  = $('input[name=lecClass]').val();

    const jsonData = {
            "lecNo"     :lecNo,
            "lecName"   :lecName,
            "lecCredit" :lecCredit,
            "lecTime"   :lecTime,
            "lecClass"  :lecClass
    }

    $.ajax({
        url: './proc/lectureInsert.jsp',
        type: 'POST',
        data: jsonData,
        dataType: 'json',
        success: function(data){
            if(data.result > 0) {
                alert('강좌등록성공!');
                let tr = "<tr>"
                       + "<td>"+data.lecNo+"</td>"
                       + "<td>"+data.lecName+"</td>"
                       + "<td>"+data.lecCredit+"</td>"
                       + "<td>"+data.lecTime+"</td>"
                       + "<td>"+data.lecClass+"</td>"
                       + "</tr>";

                $('section').hide();
                $('.lectureList').append(tr);
            } else {
                alert("강좌등록실패!");
            }
        }
    });
});
```

- 강좌등록 DAO 기능 사용
- 강좌등록 화면 끄기 켜기 기능
- 강좌등록 기능 구현 및 동적 목록 출력

문제3) 기능 구현

```java
// read
/**
 * 학생 테이블 리스트 읽기
 * @return List<studentBean>
 */
public List<studentBean> studentList() {
    List<studentBean> sbs = new ArrayList<>();
    try {
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_STUDEN
        while(rs.next()) {
            studentBean sb = new studentBean();
            sb.setStdno(rs.getString(1));
            sb.setStdName(rs.getString(2));
            sb.setStdHp(rs.getString(3));
            sb.setStdYear(rs.getInt(4));
            sb.setStdAddress(rs.getString(5));
            sbs.add(sb);
        }
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("student 리스트 읽기 오류");
        logger.error(e.getMessage());
    }
    return sbs;
}
```

```java
// student
public static final String SELECT_STUDENT_LIST =
        "select * from `student`";
```

```jsp
<%
    studentDAO sdao = new studentDAO();
    List<studentBean> sbs = sdao.studentList();
    sdao.close();
%>
  <% for (studentBean sb : sbs) { %>
  <tr>
      <td><%= sb.getStdno() %><td>
      <td><%= sb.getStdName()%><td>
      <td><%= sb.getStdHp()%><td>
      <td><%= sb.getStdYear()%><td>
      <td><%= sb.getStdAddress()%><td>
  </tr>
  <% } %>
```

```jsp
<script src="https://ajax.googleapis.com/
<script>
    $(function(){
        $('.btnOpen').click(function(){
            $('section').show();
        });
        $('.btnClose').click(function(){
            $('section').hide();
        });
    });
</script>
```

-학생목록 리스트 출력

- 등록 버튼 클릭시

학생 등록 출력 및 닫기 시 제거 기능

문제3) 기능 구현

-학생 등록 기능 구현

- ajax 이용

- 동적 처리

```java
// Create
/**
 * 학생 등록
 * @param sb
 * @return
 */
public int studentInsert(studentBean sb) {
    int result = 0;
    try {
        psmt = conn.prepareStatement(Sql.INSERT_STUDENT);
        psmt.setString(1, sb.getStdno());
        psmt.setString(2, sb.getStdName());
        psmt.setString(3, sb.getStdHp());
        psmt.setInt(4, sb.getStdYear());
        psmt.setString(5, sb.getStdAddress());
        result = psmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("student 등록 오류!");
        logger.error(e.getMessage());
    }
    return result;
}
```

```java
public static final String INSERT_STUDENT =
        "insert into `student`"
        + " (`stdNo`,`stdName`,`stdHp`,`stdYear`,`stdAddress`)"
        + " values (?,?,?,?,?)";
```

```java
request.setCharacterEncoding("UTF-8");
String studentId = request.getParameter("studentId");
String studentName = request.getParameter("studentName");
String studentHp = request.getParameter("studentHp");
String studentYear = request.getParameter("studentYear");
String studentAddr = request.getParameter("studentAddr");

studentBean sb = new studentBean();
sb.setStdno(studentId);
sb.setStdName(studentName);
sb.setStdHp(studentHp);
sb.setStdYear(Integer.parseInt(studentYear));
sb.setStdAddress(studentAddr);

studentDAO sdao = new studentDAO();
int result = sdao.studentInsert(sb);
sdao.close();

JsonObject json = new JsonObject();
json.addProperty("result", result);
json.addProperty("studentId", studentId);
json.addProperty("studentName", studentName);
json.addProperty("studentHp", studentHp);
json.addProperty("studentYear", studentYear);
json.addProperty("studentAddr", studentAddr);

String jsonData = json.toString();

out.print(jsonData);
%>
```

```javascript
$('input[type=submit]').click(function(){
    const studentId     = $('input[name=studentId]').val();
    const studentName   = $('input[name=studentName]').val();
    const studentHp     = $('input[name=studentHp]').val();
    const studentYear   = $('#studentYear option:selected').val();
    const studentAddr   = $('input[name=studentAddr]').val();

    const jsonData = {
            "studentId"     :studentId,
            "studentName"   :studentName,
            "studentHp"     :studentHp,
            "studentYear"   :studentYear,
            "studentAddr"   :studentAddr
    }

    $.ajax({
        url: './proc/studentInsert.jsp',
        type: 'POST',
        data: jsonData,
        dataType: 'json',
        success: function(data){
            if(data.result > 0) {
                alert('학생등록성공!');
                let tr = "<tr>"
                        + "<td>"+data.studentId+"</td>"
                        + "<td>"+data.studentName+"</td>"
                        + "<td>"+data.studentHp+"</td>"
                        + "<td>"+data.studentYear+"</td>"
                        + "<td>"+data.studentAddr+"</td>"
                        + "</tr>";

                $('section').hide();
                $('.studentList').append(tr);
            } else {
                alert("학생등록실패!");
            }
        }
    });
});
```

문제3) 기능 구현

```java
// read
/**
 * 수강테이블 리스트 읽기
 * @return
 */
public List<registerBean> readRegisterList() {
    List<registerBean> rbs = new ArrayList<registerBean>(
    try {
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_REGISTER_LIST);
        while(rs.next()) {
            registerBean rb = new registerBean();
            rb.setRegStdNo(rs.getString(1));
            rb.setStdName(rs.getString(2));
            rb.setLecName(rs.getString(3));
            rb.setRegLecNo(rs.getInt(4));
            rb.setRegMidScore(rs.getInt(5));
            rb.setRegFinalScore(rs.getInt(6));
            rb.setRegTotalScore(rs.getInt(7));
            rb.setRegGrade(rs.getString(8));
            rbs.add(rb);
        }
    } catch(Exception e) {
        e.printStackTrace();
        logger.error("register 리스트 읽기 오류");
        logger.error(e.getMessage());
    }
    return rbs;
}
```

```java
// register
public static final String SELECT_REGISTER_LIST =
        "select "
        + " r.`regStdNo`, "
        + " s.`stdName`, "
        + " l.`lecName`, "
        + " r.`regLecNo`,"
        + " r.`regMidScore`, "
        + " r.`regFinalScore`, "
        + " r.`regTotalScore`, "
        + " r.`regGrade`"
        + " from `register` as r"
        + " join `student` as s ON r.regStdNo = s.stdNo"
        + " join `lecture` as l ON r.regLecNo = l.lecNo";
```

```jsp
<%
    registerDAO rdao = new registerDAO();
    List<registerBean> rbs = rdao.readRegisterList();
    rdao.close();
%>
<% for (registerBean rb : rbs) { %>
<tr>
    <td><%= rb.getRegStdNo() %></td>
    <td><%= rb.getStdName()%></td>
    <td><%= rb.getLecName()%></td>
    <td><%= rb.getRegLecNo()%></td>
    <td><%= rb.getRegMidScore()%></td>
    <td><%= rb.getRegFinalScore()%></td>
    <td><%= rb.getRegTotalScore()%></td>
    <td><%= rb.getRegGrade()%></td>
</tr>
<% } %>
```

-수강현황 register 테이블 수강 데이터 출력

문제3) 기능 구현

-수강현황 register 테이블 수강 데이터 출력

- 학번 검색시 테이블 변경 (동적)

```java
/**
 * 수강현황 검색 결과 읽기
 * @param searchRegister
 * @return
 */
public List<registerBean> searchRegisterList(String searchRegister) {
    List<registerBean> rbs = new ArrayList<registerBean>();
    try {
        psmt = conn.prepareStatement(Sql.SEARCH_REGISTER_LIST);
        psmt.setString(1, searchRegister);
        rs = psmt.executeQuery();
        while(rs.next()) {
            registerBean rb = new registerBean();
            rb.setRegStdNo(rs.getString(1));
            rb.setStdName(rs.getString(2));
            rb.setLecName(rs.getString(3));
            rb.setRegLecNo(rs.getInt(4));
            rb.setRegMidScore(rs.getInt(5));
            rb.setRegFinalScore(rs.getInt(6));
            rb.setRegTotalScore(rs.getInt(7));
            rb.setRegGrade(rs.getString(8));
            rbs.add(rb);
        }
    } catch(Exception e) {
        e.printStackTrace();
        logger.error("register 검색 리스트 읽기 오류");
        logger.error(e.getMessage());
    }
    return rbs;
}
```

```java
public static final String SEARCH_REGISTER_LIST =
        "select "
        + " r.`regStdNo`, "
        + " s.`stdName`, "
        + " l.`lecName`, "
        + " r.`regLecNo`,"
        + " r.`regMidScore`, "
        + " r.`regFinalScore`, "
        + " r.`regTotalScore`, "
        + " r.`regGrade`"
        + " from `register` as r"
        + " join `student` as s ON r.regStdNo = s.stdNo"
        + " join `lecture` as l ON r.regLecNo = l.lecNo"
        + " WHERE r.`regStdNo`=?";
```

```jsp
<%
    request.setCharacterEncoding("UTF-8");
    String searchRegister = request.getParameter("searchRegister");

    registerDAO rdao = new registerDAO();
    List<registerBean> rbs = rdao.searchRegisterList(searchRegister);
    rdao.close();

    Gson gson = new Gson();
    String jsondata = gson.toJson(rbs);

    out.print(jsondata);
%>
```

문제3) 기능 구현

-학번 검색시 테이블 변경 (동적)

- 수강신청 버튼 클릭시 동적으로 수강목록 불러오는 기능

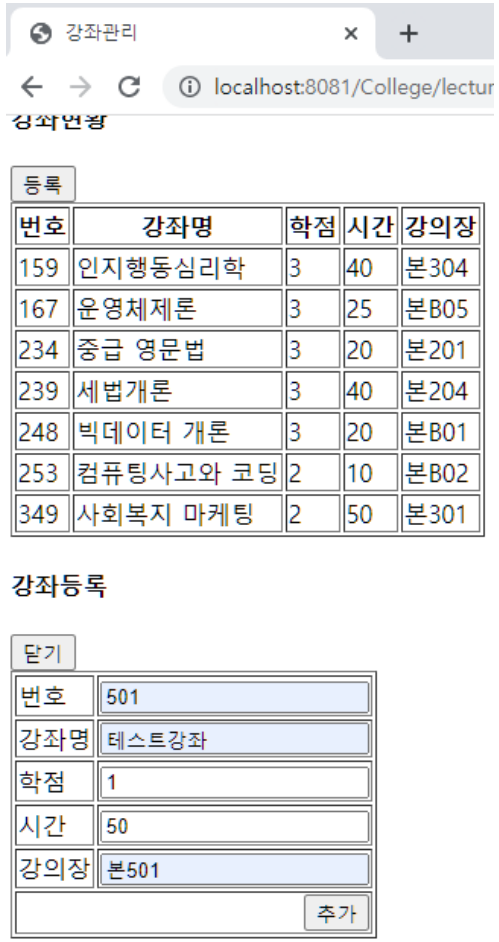
```
$('.btnSearchRegister').click(function(){
    const searchRegister = $('.searchRegister').val();

    const jsonData = {
            "searchRegister"    :searchRegister
    }

    $.ajax({
        url: './proc/searchRegister.jsp',
        type: 'POST',
        data: jsonData,
        dataType: 'json',
        success: function(data){
            $('section').hide();
            if(true) {
                $('.registerList > tbody').empty();
                for(const rbs of data){
                    const tags = "<tr>"
                    + "<td>"+rbs.regStdNo+"</td>"
                    + "<td>"+rbs.stdName+"</td>"
                    + "<td>"+rbs.lecName+"</td>"
                    + "<td>"+rbs.regLecNo+"</td>"
                    + "<td>"+rbs.regMidScore+"</td>"
                    + "<td>"+rbs.regFinalScore+"</td>"
                    + "<td>"+rbs.regTotalScore+"</td>"
                    + "<td>"+rbs.regGrade+"</td>"
                    + "</tr>";
                    $('.registerList').append(tags);
                }
            } else {
                alert("일치하는 학번이 없습니다.");
            }
        }
    });
})
```

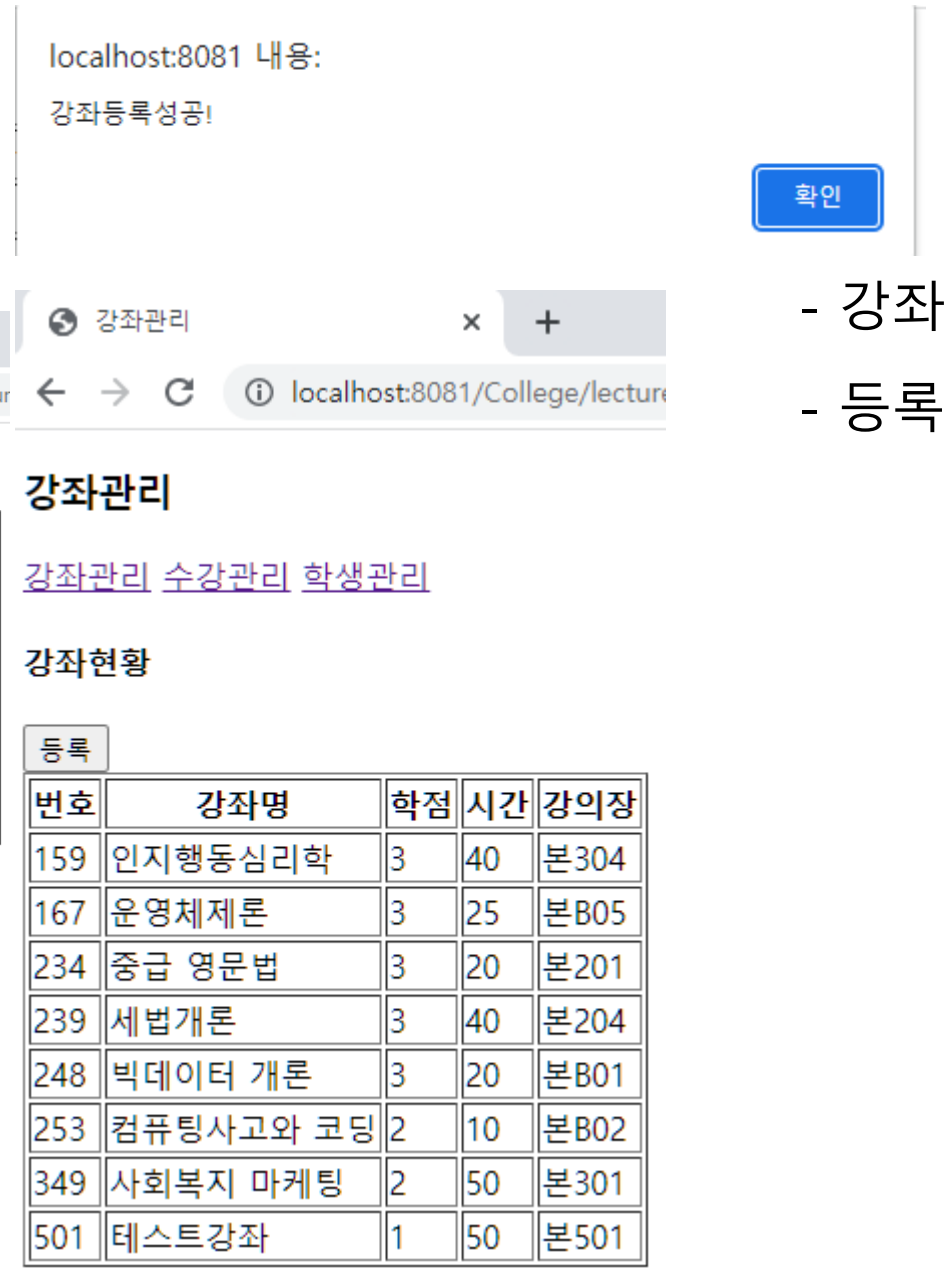
```
<%
    lectureDAO ldao = new lectureDAO();
    List<lectureBean> lbs = ldao.readLecNameList();
    ldao.close();

    Gson gson = new Gson();
    String jsondata = gson.toJson(lbs);

    out.print(jsondata);
%>
public List<lectureBean> readLecNameList() {
    List<lectureBean> lbs = new ArrayList<>();
    try {
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_LECTURE_NAME_LIST);
        while(rs.next()) {
            lectureBean lb = new lectureBean();
            lb.setLecName(rs.getString(1));
            lbs.add(lb);
        }
    } catch (Exception e) {
        e.printStackTrace();
        logger.error("lecture name list 불러오기 오류");
        logger.error(e.getMessage());
    }
    return lbs;
}
```

```
public static final String SELECT_LECTURE_NAME_LIST =
        "select `lecName` from `lecture`";
```

```
$(function(){
    $('.btnOpen').click(function(){
        $('section').show();

        $.ajax({
            url: './proc/lectureList.jsp',
            type: 'POST',
            dataType: 'json',
            success: function(data){
                for(const lbs of data) {
                    const option =
                    "<option value="+lbs.lecName+">"
                    + lbs.lecName + "</option>";
                    $('select[name=lecName]').append(option);
                }
            }
        });
    });
```

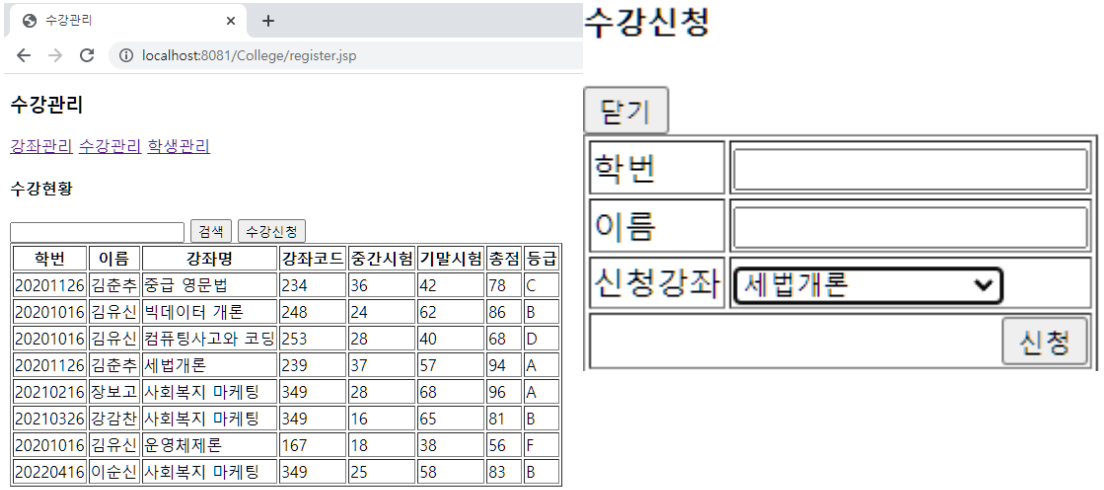
문제4) 실행

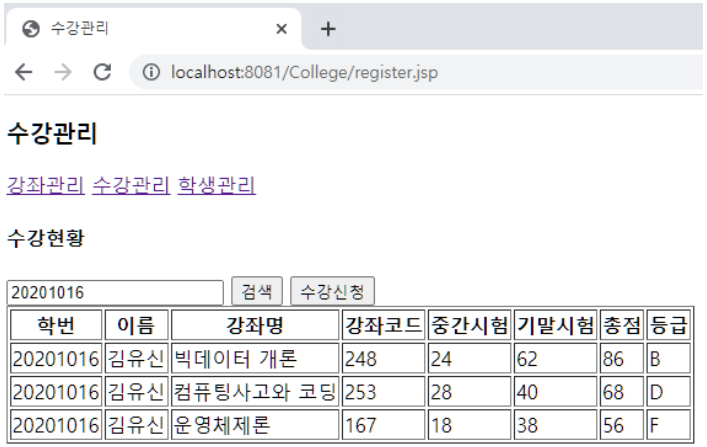




- 강좌 목록
- 등록 기능 동작 화면

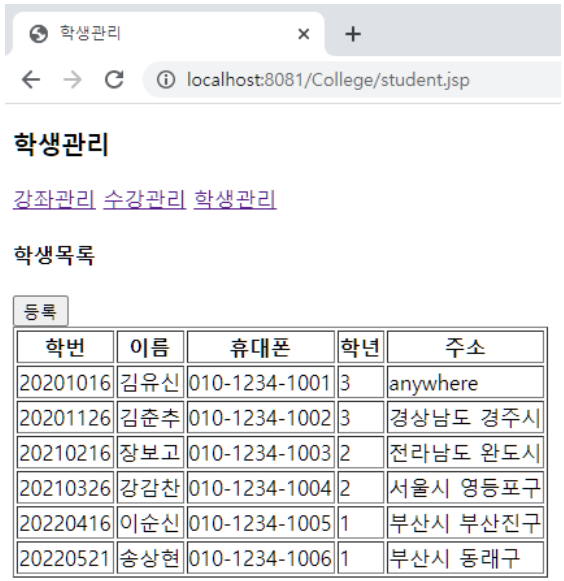
문제4) 실행

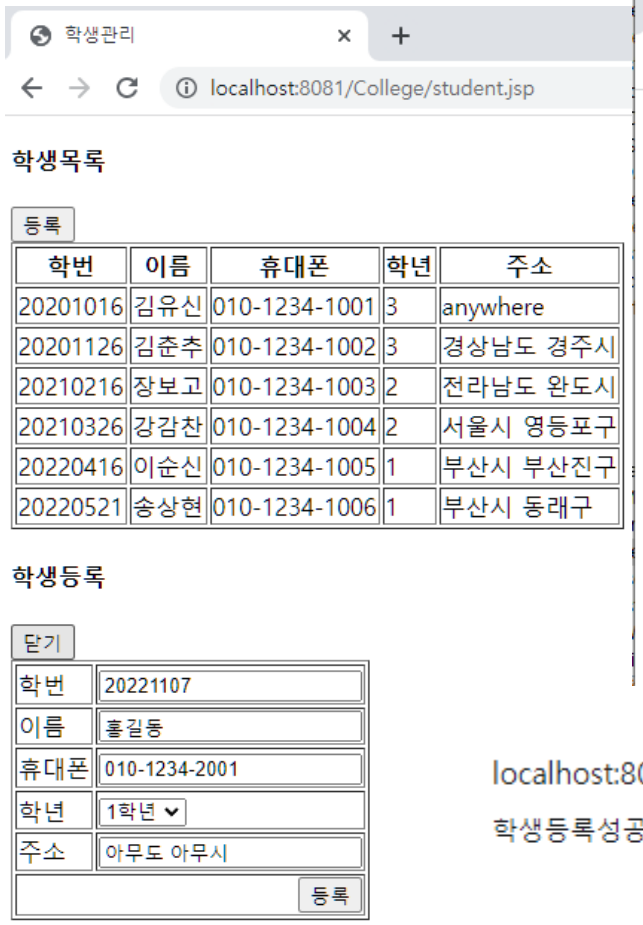




- 수강 목록 기능 동작 화면
- 학번 검색 기능
- 수강신청 버튼 클릭시 강좌목록 동적으로 불러오는 기능

문제4) 실행





- 학생 목록 기능 동작 화면