



ICE4016 데이터베이스 설계

<제목>

수강신청 페이지 만들기

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.
2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.
3. 나는 보고서의 내용을 조작하지 않겠습니다.
4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.
5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2022년 11 월 20일

학부 정보통신공학

학년 3

성명 김민겸

학번 12201863

1. 개요

○ STEP1: 4페이지에서 학생 아이디로 로그인 했을 경우에 Class 테이블에 입력된 값 조회 및 수강신청 가능한 table이 만들어져야 함

- Table의 크기(행x열)는 Class 테이블의 모든 필드를 포함하도록 만들 것
- Table의 마지막 열에는 “수강신청” 버튼을 추가
- 버튼을 눌렀을 경우 DB에 학생이 수강신청한 내용이 반영되도록 업데이트 할 것

2. 상세 설계 내용

구현한 기능

- 1) /sugang 페이지로 이동했을 때 현재 로그인한 사용자의 현재 수강 목록과 신청 가능한 강의 목록을 띄워준다.

```
router.get('/', async function (req, res) {
  // TODO
  // class 정보 불러오기
  const class_res = await selectSql.getClass(req.cookies.userId);
  const current_class_res = await selectSql.getCurrentClass(req.cookies.userId);
  const user_info_res = await selectSql.getUserInfo(req.cookies.userId);
  if (req.cookies.userName) { // user가 cookie에 존재하면
    // TODO
    // 불러온 class 정보 같이 넘겨주기
    res.render('select', {
      title1 : '현재 수강 목록',
      title2 : '수강 신청 목록',
      user : req.cookies.userName,
      current_class_res,
      class_res,
      user_info_res,
    });
  } else { // user가 존재하지 않으면 (login되지 않은 상태)
    res.redirect('/')
  }
});
```

그림 1 select.js

먼저 현재 수강 목록을 가져오기 위해 getCurrentClass함수를 정의하였고, 신청 가능한 강의 목록을 가져오기 위해 getClass 함수를 정의하였다. 그리고 가져온 변수들을 render 함수에서 select.hbs에 넘겨주어 변수를 사용하도록 하였다.

추가로, 수강신청 화면에서 사용자의 학번, 이름, 학과를 조회하기 위해 getUserInfo 함수를 정

의하였다.

```
// 이미 수강 중인 과목 제외하고 출력
getClass: async(studentId) => {
  const [rows] = await promisePool.query(`select *, (number_of_participants - current) as extra_seat \
  from class, (select count(*) as current from class_has_student) as cnt_class \
  where class.id not in (select class_id from class as c \
  inner join class_has_student as chs \
  on chs.class_Id=c.Id \
  where chs.student_Id=${studentId});`);

  return rows;
}
```

그림 2 sql.js getClass함수

not in 구문에 서브쿼리를 사용해서 이미 수강 중인 과목을 제외하고 출력하였다. 그리고 여석을 조회하기 위해 산술 연산자 '-'를 사용해 extra_seat이라는 별칭을 추가하였다.

```
getCurrentClass : async(studentId) => {
  const [rows] = await promisePool.query(`select * from class as c \
  inner join class_has_student as chs \
  on chs.class_Id=c.Id where student_Id=${studentId}`);

  return rows;
},
```

그림 3 sql.js getCurrentClass 함수

현재 로그인한 사용자가 수강 중인 강의 목록을 출력한다.

```
getUserInfo : async(studentId) => {
  const [rows] = await promisePool.query(`select student_id, s.name, d.name as major \
  from student as s, department as d \
  where s.student_id=${studentId} and s.major=d.id`);

  return rows;
},
// 이미 수강 중인 과목 제외하고 출력
```

그림 4 sql.js getUserInfo 함수

현재 로그인한 사용자의 학번, 이름, 전공을 출력한다. 이때 전공 이름을 출력하기 위해 department 테이블과 조인하였다.

```
<span class= "main-top_title">2022학년 2학기 수강신청</span> <br>
{{#each user_info_res}}
<span class="user-header">{{student_id}} {{name}} {{major}} </span>
{{/each}}
```

```

<tr class="table-header">
  <td>학수번호</td>
  <td>과목명</td>
  <td>참여자 수</td>
  <td>담당교수</td>
  <td>삭제</td>
</tr>
{{#each current_class_res}}
  <form method = "post">
    <tr>
      <td>{{Class_Id}}</td>
      <td>{{Name}}</td>
      <td>{{Number_Of_Participants}}</td>
      <td>{{Professor}}</td>

```

```

<h1>{{title2}} </h1>
<table>
  <tr class="table-header">
    <td>학수번호</td>
    <td>과목명</td>
    <td>담당교수</td>
    <td>현재 여석</td>
    <td>신청</td>
  </tr>
  {{#each class_res }}
    <form method = "post">
      <tr>
        <td>{{Id}}</td>
        <td>{{Name}}</td>
        <td>{{Professor}}</td>
        <td>{{extra_seat}} </td>
        <td style="border: none; margin-left: 10px;">

```

select.hbs에서 each 구문으로 루프를 돌면서 원하는 값들을 적절하게 출력해준다.

2) 현재 수강 목록에 대한 수강 삭제 기능

구현할 과제 목록에 없었지만, 편리하게 수강신청을 테스트하기 위해 추가로 구현하였다.

```

app.use("/", loginRouter);
app.use("/logout", logoutRouter);
app.use("/sugang", selectRouter);
app.use("/delete", deleteRouter);

import selectRouter from "../routes/select";
import deleteRouter from "../routes/delete";

```

delete.js 를 생성하고 index.js에서 위와 같이 router로 연결해주었다.

```
import express from "express";
import { deleteSql } from "../database/sql";

const router = express.Router();

router.post('/', async function(req, res) {
  const data = {
    classId : req.body.delBtn,
    studentId : req.cookies.userId,
  };
  await deleteSql.deleteClassHasStudent(data);
  res.redirect('/sugang');
});

module.exports = router;
```

그림 5 delete.js

```
// delete query
export const deleteSql = {
  deleteClassHasStudent : async(data) => {
    const sql = `delete from class_has_student where class_id='${data.classId}' and student_id=${data.studentId}`;
    await promisePool.query(sql);
  }
}
```

그림 6 sql.js

학생과 강의 조인 테이블인 class_has_student에서 해당하는 필드를 제외해주는 기능을 넣었다. 이때 select.hbs에서 delBtn이라는 이름의 필드를 가져와 그 강의의 ID를 classId 변수에 넣었고, 현재 사용자의 userId를 쿠키에서 가져와 studentId 필드에 넣어주었다.

```
<td style="border: none; margin-left: 10px;">
  <button
    style="margin-left: 10px;"
    name="delBtn"
    type="submit"
    value="{{Class_Id}}"
    formaction="/delete">수강삭제 </button></td>
</tr>
</form>
```

그림 7 select.hbs

위와 같이 각 row마다 수강삭제 기능을 넣어주면 된다.

3) 수강 신청 목록에 대한 수강 신청 기능

```

✓ router.post('/', async function(req, res) {
✓   const data = {
      classId : req.body.insertBtn,
      studentId : req.cookies.userId,
    };

    await insertSql.insertClassHasStudent(data);
    res.redirect('/sugang');
  })

  module.exports = router;

```

그림 8 select.js

```

// insert query
export const insertSql = {
  insertClassHasStudent : async(data) => {
    const sql = `insert into class_has_student (class_id, student_id) values ("${data.classId}", ${data.studentId});`;
    await promisePool.query(sql);
  }
}

```

그림 9 sql.js

sql.js에 class_has_student에 입력한 classId와 studentId를 insert하는 query를 작성하여 수강신청을 구현하였다.

```

<td>{{extra_seat}} </td>
<td style="border: none; margin-left: 10px;">
  <button
    style="margin-left: 10px;"
    name="insertBtn"
    type="submit"
    value="{{Id}}"
    formaction="/sugang">수강신청</button></td>
</tr>
</form>

```

그림 10 select.hbs

3. 실행 화면

The screenshot shows a web browser window with the URL localhost:3000/sugang. The page displays the user's profile (김민겸) and a list of courses. The '현재 수강 목록' (Current Course List) table shows three courses: ICE4020 (정보 보호론), GEE4011 (세상을 바꾸는 스타트업 이야기), and ICE4008 (컴퓨터 네트워크). The '수강 신청 목록' (Course Application List) table shows three courses: ICE4010 (이동통신), ICE4016 (Operating System), and ICE4008 (컴퓨터 네트워크). The application status for ICE4008 is '수강신청' (Course Application).

학수번호	과목명	참여자 수	담당교수	식재
ICE4020	정보 보호론	30	0	수강식재
GEE4011	세상을 바꾸는 스타트업 이야기	320	0	수강식재
ICE4008	컴퓨터 네트워크	50	0	수강식재

학수번호	과목명	담당교수	현재 여석	신청
ICE4010	이동통신	0	34	수강신청
ICE4016	Operating System	0	31	수강신청

그림 11 실행화면

현재 수강 목록에서 컴퓨터 네트워크에 대한 수강삭제를 진행하면,

The screenshot shows the same web browser window, but the '현재 수강 목록' (Current Course List) table now only contains two courses: ICE4020 (정보 보호론) and GEE4011 (세상을 바꾸는 스타트업 이야기). The '수강 신청 목록' (Course Application List) table remains the same, showing three courses: ICE4010 (이동통신), ICE4016 (Operating System), and ICE4008 (컴퓨터 네트워크). The application status for ICE4008 is '수강신청' (Course Application).

학수번호	과목명	참여자 수	담당교수	식재
ICE4020	정보 보호론	30	0	수강식재
GEE4011	세상을 바꾸는 스타트업 이야기	320	0	수강식재

학수번호	과목명	담당교수	현재 여석	신청
ICE4008	컴퓨터 네트워크	0	40	수강신청
ICE4010	이동통신	0	35	수강신청
ICE4016	Operating System	0	32	수강신청

위와 같이 변하고 실제 DB에서도 아래와 같이 반영되었다.

```
mysql> select * from class_has_student;
+----+-----+-----+
| Id | Class_Id | Student_Id |
+----+-----+-----+
| 1  | GEE4011  | 12200000  |
| 3  | GEE4011  | 12201236  |
| 4  | ICE4008  | 12201235  |
| 5  | ICE4008  | 12201234  |
| 9  | ICE4016  | 12201236  |
| 10 | ICE4016  | 12201235  |
| 11 | ICE4020  | 12201235  |
| 12 | ICE4020  | 12201863  |
| 14 | ICE4020  | 12200000  |
| 15 | GEE4011  | 12201863  |
+----+-----+-----+
10 rows in set (0.00 sec)
```

그림 12 12201863 Student_id가 ICE4008을 수강하고 있지 않음.

다시 수강신청 목록에서 컴퓨터 네트워크를 신청하면 아래와 같이 다시 현재 수강목록으로 돌아온다.

The screenshot shows a web browser window with the URL `localhost:3000/sugang`. The page title is "인하대학교 수강신청" (Inha University Course Selection). A navigation menu on the left includes "수강안내", "강의진단", "수강신청", "성적확인", "계절학기", "교과조회", "참고 Link", and "개인정보처리방침". The main content area shows the "2022학년 2학기 수강신청" (2022 Academic Year 2nd Semester Course Selection) for student "12201863 김민경 정보통신공학과" (12201863 Kim Min-gyeong, Information Communication Engineering). A link "2021 교육과정 개편 관련 경과조치 안내 (2020 이전 학번 필수 참조) -> 자세히보기" is visible. Below, there are two tables: "현재 수강 목록" (Current Course List) and "수강 신청 목록" (Course Selection List). The "현재 수강 목록" table shows three courses: ICE4020 (정보 보호론), GEE4011 (세상을 바꾸는 스타트업 이야기), and ICE4008 (컴퓨터 네트워크). The "수강 신청 목록" table shows two courses: ICE4010 (이동통신) and ICE4016 (Operating System). The bottom of the screenshot shows a Windows taskbar with the date and time "오후 2:24 2022-11-20".

학수번호	과목명	참여자 수	담당교수	식제
ICE4020	정보 보호론	30	0	수강식제
GEE4011	세상을 바꾸는 스타트업 이야기	320	0	수강식제
ICE4008	컴퓨터 네트워크	50	0	수강식제

학수번호	과목명	담당교수	현재 어석	신청
ICE4010	이동통신	0	34	수강신청
ICE4016	Operating System	0	31	수강신청

실제 DB에도 아래와 같이 반영되었음을 확인할 수 있다.


```
mysql> select * from class_has_student;
+----+-----+-----+
| Id | Class_Id | Student_Id |
+----+-----+-----+
| 1  | GEE4011  | 12200000  |
| 3  | GEE4011  | 12201236  |
| 4  | ICE4008  | 12201235  |
| 5  | ICE4008  | 12201234  |
| 9  | ICE4016  | 12201236  |
| 10 | ICE4016  | 12201235  |
| 11 | ICE4020  | 12201235  |
| 12 | ICE4020  | 12201863  |
| 14 | ICE4020  | 12200000  |
| 15 | GEE4011  | 12201863  |
| 25 | ICE4008  | 12201863  |
+----+-----+-----+
11 rows in set (0.00 sec)
```

그림 13 12201863 사용자가 ICE4008을 수강중임.