

# Project #3



과목명	데이터베이스 및 응용
학과	컴퓨터정보공학부
학번	2016722086, 2016722090
이름	함 지 훈, 김 기 우 (DB손해보험)
제출일자	2020. 12. 02

## 1. 주제 및 설명

이번 프로젝트의 주제는 학생 정보 관리 시스템으로 기존의 KLAS 시스템과 유사한 애플리케이션을 개발하는 것이다. 우리의 목표는 단순히 KLAS를 모방하는데 그치지 않고, KLAS 보다 더 다양한 기능을 제공하여 기존의 KLAS 보다 더 쓰기 편한 시스템을 만드는 것이다. 또한 우리 팀만의 개성을 살려서 유저 인터페이스도 참신하게 바꿔볼 것이다.

## 2. 구현 환경

- DBMS: Mysql



- 런타임 환경: Node.js



- 웹 프레임워크: express.js



- UI 라이브러리: React.js



- 개발 언어: javascript

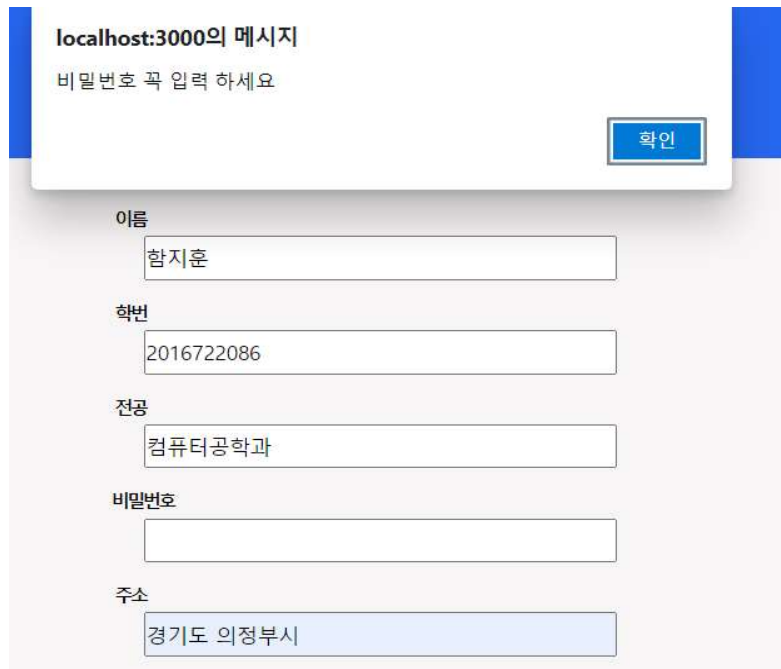
- 개발 언어: javascript



### 3. 주요기능 및 특별 기능 설명

#### ■ 회원가입

- 어떤 정보가 누락되었을 경우 그에 맞는 경고 메시지 출력



localhost:3000의 메시지  
비밀번호 꼭 입력 하세요

확인

이름  
함지훈

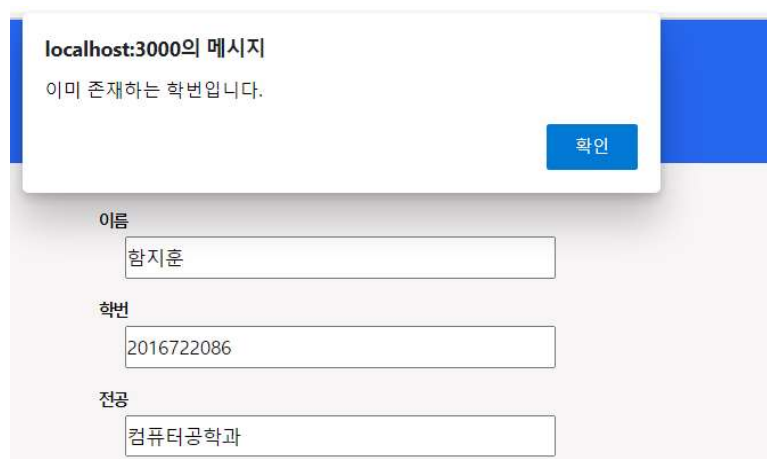
학번  
2016722086

전공  
컴퓨터공학과

비밀번호

주소  
경기도 의정부시

- 회원가입 완료시 자동으로 로그인 화면으로 이동
- 이미 존재하는 학번을 입력하면 경고 메시지 출력



localhost:3000의 메시지  
이미 존재하는 학번입니다.

확인

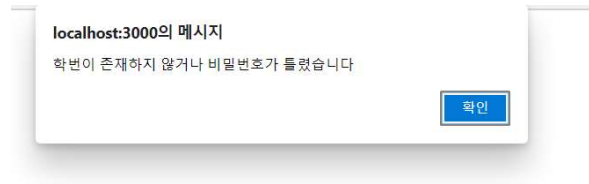
이름  
함지훈

학번  
2016722086

전공  
컴퓨터공학과

#### ■ 로그인

- 아이디 혹은 비밀번호가 일치하지 않을 시, 로그인 실패화면 출력



## 로그인

2016722086

\*\*\*\*\*

로그인

아직 회원이 아니신가요?  
회원가입

- 로그인이 성공하면 시간표 조회 화면으로 이동
- 로그인 시 해당 아이디가 회원가입 후 특정 시간(90일)이 경과되면 비밀번호 변경 제안 기능을 구현하려 하였으나 일정에 문제가 생겨 구현하지 못함

## ■ 비밀번호 변경

localhost:3000의 메시지

비밀번호 변경이 완료 되었습니다!

확인

비밀번호

\*\*\*\*\*

변경

- 비밀번호를 변경하면 완료 메시지가 뜨면서 바로 로그인 화면으로 redirection 된다.

## ■ 개인 정보 조회

함지훈님의 개인정보조회

제목	내용
이름	함지훈
학번	2016722086
전공	컴퓨터공학과
연락처	01086019275
이메일	jemma0422@naver.com
주소	의정부시
계좌번호	97300228301011

뒤로가기

- 시간표 화면에서 개인정보조회를 클릭하면 비밀번호를 제외한 개인정보를 볼 수 있음

## ■ 시간표 조회

## 합지훈 님의 시간표

합지훈의 2020년도 2학기의 시간표입니다.

	월	화	수	목	금
1교시					운영체제
2교시					
3교시	데이터베이스				
4교시			데이터베이스		
5교시	데이터통신	전자회로			
6교시			데이터통신	전자회로	

### \* 과목별 공지사항 \*

데이터통신	<a href="#">바로가기</a>
데이터베이스	<a href="#">바로가기</a>
전자회로	<a href="#">바로가기</a>
운영체제	<a href="#">바로가기</a>

[수강 신청](#)

[성적조회](#)

[친구 시간표 구경하기](#)

[비밀번호 변경하기](#)

[로그아웃](#)

[개인정보조회](#)

- 데이터베이스에서 시간표 정보를 가져온 후 시간표를 출력
- 과목별 공지사항 게시판으로 이동 가능
- 수강신청, 성적조회, 친구 시간표 구경, 비밀번호 변경, 개인정보조회를 버튼으로 선택하여 홈페이지 이동 가능

## ■ 수강신청

localhost:3000

localhost:3000의 메시지  
중복된 시간표가 있어 일부 과목이 신청되지 않았습니다

확인

**\* 검색결과 \***

강의명	교수명	연도	학과	강의실	학정번호	이수구분	강의시간	학점	신청
자료구조	이기훈	2020	소프트웨어학부	참313	99999999	교양	화목12	3	신청

**\* 신청목록 \***

강의명	교수명	연도	학과	강의실	학정번호	이수구분	강의시간	학점	신청
알고리즘	황호영	2020	컴퓨터정보공학부	옥101	10060023	전공	월수34	3	삭제
자료구조	이기훈	2020	소프트웨어학부	참313	99999999	교양	화목12	3	삭제

저장    나가기

localhost:3000의 메시지  
모든과목의 수강신청이 완료 되었습니다.

확인

**\* 신청목록 \***

강의명	교수명	연도	학과	강의실	학정번호	이수구분	강의시간	학점	신청
데이터통신	황호영	2020	컴퓨터정보공학부	새103	00000001	전공	월수56	3	삭제
자료구조	이기훈	2020	소프트웨어학부	참313	99999999	교양	화목12	3	삭제
데이터베이스	이기훈	2020	컴퓨터정보공학부	새202	12345678	전공	월수34	3	삭제
운영체제	김태석	2020	소프트웨어학부	새404	12341000	교양	금12	3	삭제

저장    나가기

- 시간표가 중복되는 경우 중복되지 않은 시간표들만 신청됨
- 과목 검색 시 해당되는 과목의 정보가 검색결과에 나옴
- 신청 버튼을 누르면 신청목록에 과목이 담기며 삭제도 가능
- 수강신청 완료는 저장 버튼을 통해 가능

## ■ 과목별 공지사항

## 공지사항

제목	글쓴이
이곳은 자료구조 공지사항 게시판 입니다.	운영자

뒤로가기

- 각 과목별 공지사항 바로가기 버튼을 누르면 과목에 해당하는 공지사항 홈페이지로 이동.

## ■ 학습결과

### 수강/성적 조회

학기	취득 학점	전공 평점	교양 평점	전체 평점
2020-2	12	0.00	0.00	0.00
2020-1	20	4.5	4.0	4.25
2019-2	21	3.0	4.0	3.5
2019-1	20	3.0	3.5	3.25
2018-2	19	4.1	4.3	4.2
종합	92	2.92	3.16	3.04



### 2020-2학기 성적

과목명	학점
데이터베이스	미입력
데이터통신	미입력
운영체제	미입력
자료구조	미입력

뒤로가기

- 이번학기에 신청한 과목까지 고려하여 전체 평점, 교양 평점, 전공 평점, 이수 학점을 볼

수 있음

- 학기별 성적 조회 가능
- 전공, 교양, 전체 평점에 대한 데이터 시각화 구현
- 이번 학기 성적 조회 가능

## ■ 친구 시간표 공유

친구는 어떤 수업을 듣고 있을까?

김기우의 2020년도 2학기의 시간표입니다.

	월	화	수	목	금
1교시					
2교시					
3교시	데이터베이스				
4교시			데이터베이스		
5교시	데이터통신	전자회로			
6교시			데이터통신	전자회로	

뒤로가기

- 친구의 학번을 입력하면 친구의 이름과 함께 친구의 시간표를 볼 수 있음



#### 4. 테이블 설명(구체적 상관관계는 E-R모델에서 명시)

##### 1. STUDENT(학생 개인 정보)

Sname	ID	Passwd	Phone	Email	Major	<u>SID</u>	Saccount	Bank	Address
- S_name: 학생 이름			VARCHAR(10)		NOT NULL				
- Address: 학생 거주지			VARCHAR(30)		NOT NULL				
- Phone: 학생 비상연락망			PHONENUM		NOT NULL				
- Email: 학생 이메일			VARCHAR(30)						
- Major: 학생 전공			VARCHAR(15)		NOT NULL				
- <u>SID</u> : 학번(Primary Key)			CHAR(10)		NOT NULL		PRIMARY KEY		
- S_account: 학생의 계좌번호			VARCHAR(20)		NOT NULL		UNIQUE		
- Passwd: 계정 로그인 비밀번호			VARCHAR(20)		NOT NULL				

##### 2. COURSE(과목 정보)

Cname	Prof	Year	Semester	Major	Room	<u>CCode</u>	Subject	Time	Credit
- Cname: 과목 이름			VARCHAR(10)		NOT NULL				
- Prof: 담당 교수 이름			VARCHAR(10)		NOT NULL				
- C_year: 연도			SMALLINT		NOT NULL				
- Major: 학과			VARCHAR(15)		NOT NULL				
- Room: 강의실			VARCHAR(10)		DEFAULT '미배정'				
- CCode: 학정번호			CHAR(8)		NOT NULL		PRIMARY KEY		
- C_subject: 교과(교양/전공)			COURSETYPE		NOT NULL		CHECK(C_subject IN('전공', '교양'))		
- C_time: 수업 시간대			VARCHAR(10)		NOT NULL				
- Credit: 이수학점			TINYINT		NOT NULL				

##### 3. PRIVATECOURSE(개인 별 수강현황)

(index)	PSID	PCCode	PTime	Grade
- idx: 튜플id(Primary Key)			int auto_increment	PRIMARY KEY
- PSID: 학번(Foreign Key)			CHAR(10)	NOT NULL
- PCCode: 학정번호(Foreign Key)			CHAR(8)	NOT NULL
- PTime: 수업 시간대			VARCHAR(10)	NOT NULL
- Grade: 수강 과목 성적			FLOAT	DEFAULT 0

##### 4. PROFESSOR(교수 정보)

<u>P_name</u>	Lab	PMajor
- P_name: 교수 이름		VARCHAR(10) NOT NULL PRIMARY KEY
- Lab: 연구실		VARCHAR(10) NOT NULL
- PMajor: 연구 분야		VARCHAR(15) NOT NULL

## CREATE TABLE문

### STUDENT

```
CREATE TABLE STUDENT
(
  S_name    VARCHAR(10)    NOT NULL,
  Address   VARCHAR(30)    NOT NULL,
  Phone     CHAR(11)       NOT NULL,
  Email     VARCHAR(30),
  Major     VARCHAR(15)    NOT NULL,
  SID       VARCHAR(10)    NOT NULL PRIMARY KEY,
  S_account VARCHAR(20)    NOT NULL UNIQUE,
  Passwd    VARCHAR(20)    NOT NULL
);
```

### COURSE

```
CREATE TABLE COURSE
(
  C_name    VARCHAR(10)    NOT NULL,
  Prof      VARCHAR(10),
  C_year    SMALLINT       NOT NULL,
  Major     VARCHAR(15)    NOT NULL,
  Room      VARCHAR(10)    DEFAULT '미배정',
  Ccode     CHAR(8)        NOT NULL PRIMARY KEY,
  C_subject CHAR(2)        NOT NULL CHECK(C_subject IN ('전공', '교양')),
  C_time    VARCHAR(10)    NOT NULL,
  Credit    TINYINT        NOT NULL,
  FOREIGN KEY (Prof) REFERENCES PROFESSOR(P_name)
  ON DELETE SET NULL ON UPDATE CASCADE
);
```

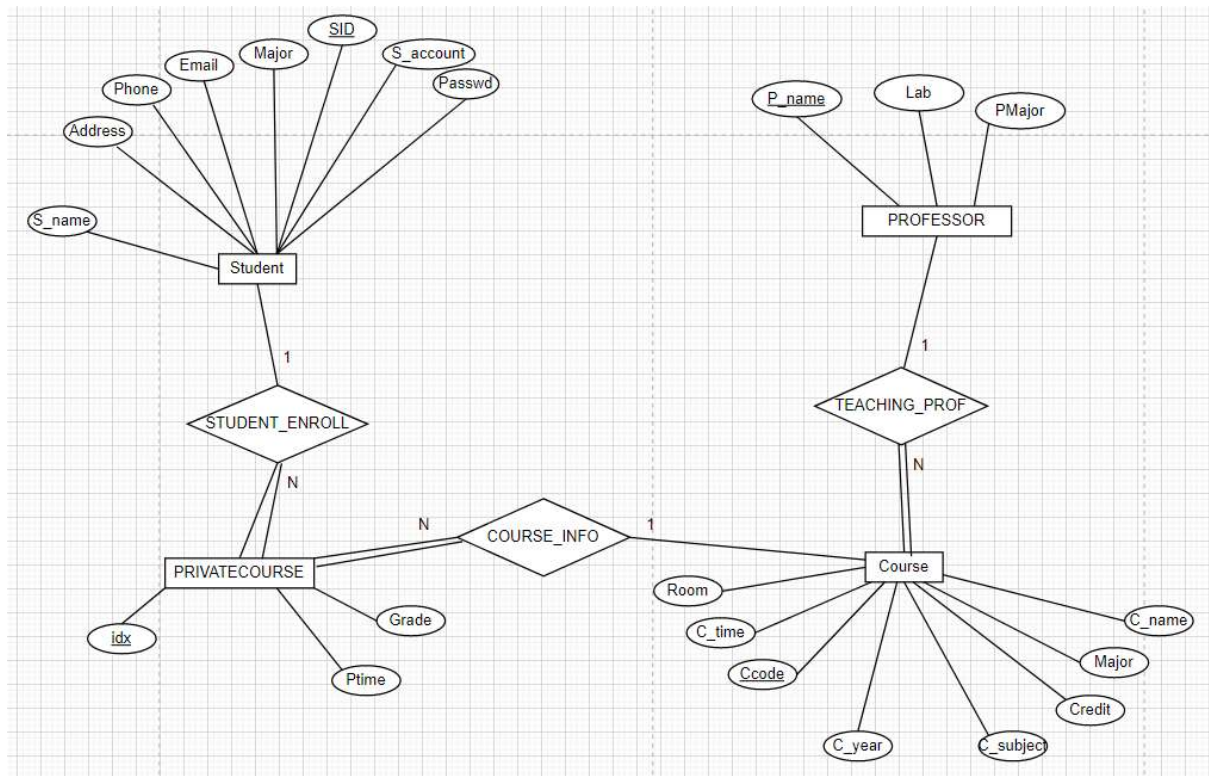
### PRIVATECOURSE

```
CREATE TABLE PRIVATECOURSE
(
  idx      int auto_increment PRIMARY KEY,
  PSID     VARCHAR(10)    NOT NULL,
  PCCode    CHAR(8)       NOT NULL,
  Ptime     VARCHAR(10)    NOT NULL,
  Grade     FLOAT         DEFAULT 0,
  FOREIGN KEY (PSID) REFERENCES STUDENT(SID)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (PCCode) REFERENCES COURSE(Ccode)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

### PROFESSOR

```
CREATE TABLE PROFESSOR
(
  P_name    VARCHAR(10)    NOT NULL PRIMARY KEY,
  Lab       VARCHAR(10)    NOT NULL,
  PMajor    VARCHAR(15)    NOT NULL
);
```

## 5. E-R 다이어그램



STUDENT 테이블의 (S\_name, ID, Address, Phone, Email, Major, S\_account, passwd, Totalcredit)은 각 학생 개인의 (이름, id, 주소, 전화번호, 이메일, 전공, 학번, 계좌번호, 비밀번호, 이수 학점, 전공 평점, 전체 평점)을 나타내고 STUDENT의 수강 과목 현황을 나타내는 STUDENT\_ENROLL 관계에 참여하고 있다.

PRIVATECOURSE 테이블의 (idx, PSID, PCCode, Ptime, Grade)는 각각 (index, 학번, 과정번호, 과목 시간대, 성적)을 나타내고 STUDENT\_ENROLL 관계에 foreign key인 PSID가 들어가서 참여한다. 과목의 정보를 나타내는 COURSE\_INFO 관계에는 또다른 foreign key인 PCCode가 들어가서 참여한다. 한 학생 당 여러 과목의 리스트로 수강 현황이 만들어지므로 STUDENT\_ENROLL 관계는 1:N의 비율로 참여한다.

COURSE 테이블의 (C\_name, Prof, C\_year, Major, Room, Ccode, C\_subject, C\_time, Credit)은 각각 과목의 (이름, 담당교수, 연도, 전공, 장소, 학정번호, 학과, 시간대, 학점)을 나타낸다. 과목정보를 나타내는 COURSE INFO 관계에 참여하고 있으며 한 과목이 여러 학생들의 수강현황에 들어갈 수 있으므로 테이블 COURSE와 PRIVATECOURSE가 1:N의 비율로 참여한다. 과목을 맡은 교수의 정보를 나타내는 TEACHING\_PROF 관계에서는 foreign key인 Prof가 들어가서 참여한다.

PROFESSOR 테이블의 (P\_name, Lab, PMajor)은 (교수 이름, 연구실 장소, 개인 전공)을 나타내고 한 명의 교수가 여러 과목을 맡을 수 있으므로 테이블 PROFESSOR와 COURSE가 TEACHING\_PROF관계에 1:N의 비율로 참여한다.

## 6. SQL문 작성 및 설명

No	SQL 문 종류	SQL 문
1	INSERT	<pre> 20 INSERT INTO COURSE(C_name, Prof, C_year, Major, Room, Ccode, C_subject, 21 C_time, Credit) 22 VALUES('데이터베이스', '이기훈', 2020, '컴퓨터공학과', '새202', '00000001', '전공', 23 '월수34', 3), 24 ('자료구조', '이기훈', 2020, '컴퓨터공학과', '삼313', '00000002', '전공', 25 '화목12', 3), 26 ('데이터통신', '황보영', 2020, '컴퓨터공학과', '새103', '00000003', '전공', 27 '월수56', 3), 28 ('운영체제', '김태석', 2020, '컴퓨터공학과', '새404', '00000004', '전공', 29 '금12', 3), 30 ('알고리즘', '황보영', 2020, '컴퓨터공학과', '목101', '00000005', '전공', 31 '월수34', 3), 32 ('전자회로', '공진홍', 2020, '전자공학과', '새105', '00000006', '전공', 33 '화목56', 3), 34 ('디지털신호처리', '심동규', 2020, '컴퓨터공학과', '새206', '00000007', '전공', 35 '금34', 3), 36 ('공학수학1', '민경진', 2020, '전자공학과', '비409', '00000009', '전공', 37 '화목56', 3), 38 ('회로이론', '박철수', 2020, '컴퓨터공학과', '비619', '00000010', '전공', 39 '화목65', 3), 40 ('선형대수학', '채형진', 2020, '소프트웨어학부', '비403', '00000011', '전공', 41 '월수65', 3), 42 ('과학기술윤리', '박지희', 2020, '전체공통', '비402', '00000013', '교양', 43 '월수21', 3), 44 ('정보사회와수학', '김영희', 2020, '전체공통', '비403', '00000014', '교양', 45 '화목56', 3), 46 ('공학설계입문', '김순택', 2020, '컴퓨터공학과', '비427', '00000015', '전공', 47 '금34', 3), 48 ('대학영어', '양정현', 2020, '전체공통', '한303', '00000016', '교양', 49 '화목12', 3); </pre> <p>과목별 데이터 삽입 사전작업, STUDENT, PRIVATECOURSE도 동일</p> <p>수강신청 할 때 필요한 과목정보, 학생 개인의 성적, 교수 정보 등 사전에 필요한 데이터들을 미리 SQL에서 삽입함, 회원가입 시 신규 회원 데이터를 삽입함.</p>
2	COUNT	<pre> select COUNT (*) AS count from COURSE where C_name=?; </pre> <p>COURSE 테이블에서 해당 과목의 개수를 반환, 이외에도 많은 쿼리가 COUNT를 사용</p> <p>과목 이름이 같은데 교수가 다른 경우가 있기 때문에 이에 대한 처리를 해주기 위해 사용.</p>
3	IN	<pre> select COUNT (*) as cnt from PRIVATECOURSE where (Ptime, PSID) IN (select p.Ptime, p.PSID FROM privatecourse as p where p.Ptime=? and p.PSID=?); </pre> <p>PRIVATECOURSE 테이블에서 PSID와 Ptime이 일치하는 튜플의 수를 반환</p> <p>수강하려는 과목의 시간대가 기존에 존재할 경우 중복되는 과목의 수를 반환, 중복되는 과목이 0개이면 과목이 신청되고 0이 아니면 과목이 신청되지 않는다.</p>

4	UPDATE	<pre>update STUDENT SET passwd=? where SID=?;</pre>
		STUDENT 테이블의 해당하는 SID를 갖는 튜플의 passwd 업데이트
		입력한 값으로 기존의 비밀번호를 update
5	INNER JOIN	<pre>select S_name, SID, Ptime, C_name from (student as s INNER JOIN (privatecourse as p) on p.PSID = s.SID) INNER JOIN course as c on p.PCCode=c.Ccode where s.SID = ?;</pre>
		STUDENT 테이블, PRIVATECOURSE 테이블, COURSE 테이블을 join한 후 해당하는 학번의 튜플에서 S_name, SID, Ptime, C_name 반환
		학생 개인의 시간표 조회
6	ORDER BY	<pre>select C_name, Grade from privatecourse, course where PCCode = CCode and PSID=? ORDER BY C_name ASC;</pre>
		PRIVATECOURSE, COURSE를 join한 후 해당하는 학번의 튜플들을 과목의 이름을 기준으로 오름차순 정렬
		학기 별 개인 성적 조회
7	AVG	<pre>SELECT AVG(Grade) as BB FROM privatecourse where PSID=?;</pre>
		PRIVATECOURSE 테이블에서 해당하는 학번의 학생의 Grade의 AVG를 반환
		해당하는 학번의 평점 구하기
8	GROUP BY	<pre>select AVG(Grade) as AA from privatecourse join course on PCCode = CCode where PSID=? GROUP BY C_subject;</pre>
		PRIVATECOURSE와 COURSE를 join한 후 해당하는 학번의 튜플을 전공/교양 그룹별로 묶어준 후 그룹별 AVG를 반환
		해당하는 학번의 전공 및 교양 평점 구하기

9	SUM	<pre>select SUM(CREDIT) as SUM FROM privatecourse join course on PCCode = Ccode WHERE PSID=?</pre>
		PRIVATECOURSE와 COURSE를 join한 후 해당하는 학번의 튜플의 credit의 SUM을 반환
		학생의 전체 이수 학점 구하기

## 7. SQL문 실행 화면 캡처

### 1. INSERT ... INTO ...

	C_name	Prof	C_year	Major	Room	Ccode	C_subject	C_time	Credit
▶	데이터베이스	이기훈	2020	컴퓨터공학과	새202	00000001	전공	월수34	3
	자료구조	이기훈	2020	컴퓨터공학과	잡313	00000002	전공	화목12	3
	데이터통신	황호영	2020	컴퓨터공학과	새103	00000003	전공	월수56	3
	운영체제	김태석	2020	컴퓨터공학과	새404	00000004	전공	금12	3
	알고리즘	황호영	2020	컴퓨터공학과	옥101	00000005	전공	월수34	3
	전자회로	공진홍	2020	전자공학과	새105	00000006	전공	화목56	3
	디지털신호처리	심동규	2020	컴퓨터공학과	새206	00000007	전공	금34	3
	공학수학1	민경진	2020	전자공학과	비409	00000009	전공	화목56	3
	회로이론	박철수	2020	컴퓨터공학과	비619	00000010	전공	화목65	3
	선형대수학	채형직	2020	소프트웨어학부	비403	00000011	전공	월수65	3
	과학기술윤리	박지희	2020	전체공통	비402	00000013	교양	월수21	3
	정보사회와수학	김영희	2020	전체공통	비403	00000014	교양	화목56	3
	공학설계입문	김순희	2020	컴퓨터공학과	비427	00000015	전공	금34	3
	대학영어	양창현	2020	전체공통	한303	00000016	교양	화목12	3
	전자회로	이준환	2020	컴퓨터공학과	새101	00000017	전공	화목65	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

미리 수강신청 가능한 과목들에 대한 정보를 넣어두기 위해 INSERT 문을 사용해서 data들을 삽입한다.

### 2. COUNT

	count
▶	2

수강신청										
전자회로										
검색										
* 검색결과 *										
강의명	교수명	연도	학과	강의실	학점번호	이수구분	강의시간	학점	신청	
전자회로	공진홍	2020	전자공학과	새105	00000006	전공	화목56	3	신청	
전자회로	이준환	2020	컴퓨터공학과	새101	00000017	전공	화목65	3	신청	

수강신청 할 때 이름이 중복되는 여러 과목들에 대한 로직 처리를 위해 해당하는 과목의 갯수를 계산한다.



### 3. IN

cnt
1

\* 신청목록 \*

강의명	교수명	연도	학과	강의실	학점번호	이수구분	강의시간	학점	신청
전자회로	공진용	2020	전자공학과	세105	00000006	전공	의목56	3	삭제
공학수학1	방경진	2020	전자공학과	비409	00000009	전공	의목56	3	삭제

[저장](#)
[나가기](#)



localhost:3000의 메시지

중복된 시간표가 있어 일부 과목이 신청되지 않았습니다

[확인](#)
[백](#)

수강 신청 시 시간대가 중복되는 과목이 있으면 해당 과목에 대한 신청을 reject하기 위해 사용한다.

### 4. UPDATE ... SET ...

passwd
1234



passwd
4321

비밀번호 변경 시 변경된 비밀번호로 정보를 업데이트하기 위해 사용한다.

### 5. INNER JOIN

	S_name	SID	Ptime	C_name
▶	김기우	2016722090	월수34	데이터베이스
	김기우	2016722090	월수56	데이터통신
	김기우	2016722090	화목56	전자회로
	김기우	2016722090	금34	디지털신호처리
	김기우	2016722090	월수21	과학기술윤리
	김기우	2016722090	화목65	회로이론
	김기우	2016722090	화목12	대학영어

김기우의 2020년도 2학기의 시간표입니다.

	월	화	수	목	금
1교시		대학영어	과학기술윤리		
2교시	과학기술윤리			대학영어	
3교시	데이터베이스				디지털신호처리
4교시			데이터베이스		
5교시	데이터통신	전자회로		회로이론	
6교시		회로이론	데이터통신	전자회로	

학생 개인의 수강 과목을 조회하여 시간표를 작성하기 위해 사용한다.

### 6. ORDER BY

	C_name	GRade
▶	과학기술윤리	2.5
	대학영어	3.5
	데이터베이스	4.5
	데이터통신	4
	디지털신호처리	3
	전자회로	3.5
	회로이론	3.5

2020-2학기 성적

과목명	학점
과학기술윤리	C+
대학영어	B+
데이터베이스	A+
데이터통신	A0
디지털신호처리	B0
전자회로	B+
회로이론	B+

과목 이름을 사전순으로 나열하여 성적 조회할 때 사용한다.

## 7. AVG

	AVG_GRADE
▶	3.5

학기	취득 학점	전공 평점	교양 평점	전체 평점
2020-2	21	3.70	3.00	3.50
2020-1	20	4.5	4.0	4.25
2019-2	21	3.0	4.0	3.5
2019-1	20	3.0	3.5	3.25
2018-2	19	4.1	4.3	4.2
총합	101	3.66	3.76	3.74

개인 성적 조회 시 전체 학점의 평균치를 구해서 출력하기 위해 사용한다.

## 8. GROUP BY

	C_subject	average
▶	전공	3.7
	교양	3



개인 성적 조회 시 평점을 전공/교양 별로 나눠서 출력하기 위해 사용한다.

## 9. SUM

	수강학점
▶	21

학기	취득 학점	전공 평점	교양 평점	전체 평점
2020-2	21	3.70	3.00	3.50
2020-1	20	4.5	4.0	4.25
2019-2	21	3.0	4.0	3.5
2019-1	20	3.0	3.5	3.25
2018-2	19	4.1	4.3	4.2
총합	101	3.66	3.76	3.74

개인 성적 조회 시 이번 학기에 신청한 과목 학점 수의 총합을 구해서 출력하기 위해 사용한다.



## 8. 고찰

### ■ 2016722086 함지훈

우리 팀은 프로젝트를 진행하면서 분업할 때 백엔드와 프론트엔드로 따로 나누지 않고 각 기능(사이트)단위로 담당하였다. 내가 맡은 기능은 회원가입, 수강신청, 친구 시간표 검색, 비밀번호 변경이었다. 나와 김기우 팀원 모두 React를 공부하고 적용하는 부분에서 막대한 시간이 소요되어 목표로 했던 추가 기능들의 구현이 많이 이루어지지 않았다.

사실 비교적 간단한 쿼리문들만 사용하여 쿼리에 대한 고찰보다는 백엔드/프론트엔드 구현에 대해 써볼 내용이 많다. 우선 백엔드 부분에서 가장 당황했던 부분은 서버에서 get, post를 구현해 놓는게 그냥 api를 구현하는 것과 다를 바가 없었다는 것이다. 왜냐하면 튜토리얼에서는 html에서 href를 사용하는 것이 url을 바꿔주는 역할인지 몰랐기 때문에 서버가 url을 변경해주는줄 알았는데 react를 사용하여 UI를 구현하다보니 url이 전혀 서버에 의해 바뀌지 않았기 때문이다.

다음으로 우리 프로그램은 홈페이지가 이동될 때마다 url이 변하지 않는다는 문제가 있다. 무조건 localhost:3000 고정이다. 이를 해결하기 위해 react router를 사용해보았지만, url이 변하는 대신 홈페이지가 이동되지 않는 다른 문제가 생겼다. 이 부분에서 왜 문제가 생겼냐면 우리는 홈페이지 이동에 if문을 사용하여 구현하였기 때문이다. 버튼을 눌렀을 때 페이지를 이동하고 싶으면 각 버튼을 눌렀을 때 갱신되는 state의 page변수를 달라지게 하여 다시 rendering이 되었을 때 다른 컴포넌트를 return하도록 구현하였다. 따라서 react router를 사용하면 다른 component를 return하지 않기 때문에 기존의 페이지가 유지되면서 url만 바뀌어 버렸다. 이 부분을 어떻게든 해결해보려 했지만 시간이 너무 촉박하여 더 손을 댈 수 없었다.

마지막으로 javascript의 non-blocking 성질 때문에 시간이 많이 소비되었다. 평소에는 잘 모르다가 서버에서 반복문을 중첩하여 여러 번 돌려야 하는 상황이 나오자, 내 예상대로 동작하지 않는 것을 보고 javascript의 코드는 sequential 하게 동작하지 않는다는 것을 깨달았다. 그래서 나는 반복문을 non-blocking 방식에 맞게 내가 설계를 해야 하는 줄 알고 다른 방식으로 시도해 보았지만 머리만 아플 뿐 순차적으로 진행되게 할 수 없었다. 콜백함수를 계속 중첩시켜서 구현하는 것도 이론상으로는 가능했지만 가독성이 너무 떨어지고 사실 그렇게 구현하기도 쉽지 않았다. 이를 해결하기 위해 async, await 패턴을 한번 사용해봤더니 반복문을 순차적으로 진행시킬 수 있었다. 구체적으로 말하자면 강제로 루프를 한 번씩만 실행시키고 시간을 delay한 후 다음 루프를 실행시키는 방식으로 이 문제를 극복할 수 있었다.

과제 기여에 대한 나의 점수는 노력을 정말 많이 했지만 url을 갱신하는 부분에서 아쉬움이 많이 남고, 서버 사이드에서의 렌더링도 시도해보지 못했기 때문에 9점을 주겠다.

## ■ 2016722090 김기우

프로젝트를 진행하면서 맨 처음에 웹을 따로 구현해본 경험이 없어서 많은 초기 공부가 필요했다. 그래서 기본 지식은 생겨도 익숙하지가 않아서 처음에 react에 많은 노력을 하였다. 함지훈 팀원과 분업을 하였고 나는 크게 로그인 페이지, 마이페이지, 성적 조회 페이지, 개인 정보 조회 페이지를 구현하였다. 가장 시간을 많이 사용하고 잘 안됐었던 부분은 페이지끼리 이동과 렌더링이었다. 앞서 함지훈 팀원의 고찰에서도 볼 수 있지만 url이 서버에 의해 바뀌지 않고 있다는 것을 알고 여러 시도 끝에 react router부분, if문을 이용해서 조건부로 각 페이지 정보들이 담겨있는 컴포넌트들을 리턴해주는 방식으로 구현하였다. 이러한 페이지 이동 이외에도 proxy 설정, html 문법, 그리고 의외로 css에서까지 많이 막혀 시간을 썼다. 서버부분에서의 SQL작성 등은 많은 시간이 소요되지는 않았다.

프로젝트를 완성하고 지금 기억해보면 금방 해결할 수 있는 문제들이었지만 당시에는 완전 익숙하지가 않아서 사소한 이슈에도 많은 해결시간을 썼었던 것 같다. 물론 지금도 프로젝트 1개를 완성할 정도만 다루었기 때문에 잘 다룰 정도로 익숙하지는 않지만 꽤 많이 해보고 능력이 쌓이면 굉장히 재밌을 것 같은 프로그래밍 공부 분야라고 느꼈다. 졸업 하기 전에 웹 구현 프로젝트의 경험을 한번이라도 쌓을 수 있는 기회라서 개인적으로 프로젝트에 대해 만족했고 완성하고 나서는 나름의 성취감도 느낄 수 있었다.

과제 기여에 대한 나의 점수는 웹에 대한 첫번째 접근이었지만 어느정도 만족한 반면에 함지훈 팀원 처럼 url 렌더링 구현 부분에서 아직도 아쉬움을 가지고 있기 때문에 9점이다.

## \* 시연 영상 YOUTUBE 링크

<https://www.youtube.com/watch?v=gaAXfE6gwpc&t=42s>