

커뮤니케이션 기록

2020.11.24. 회의내용 (전화통화로 1시간가량 진행)

진행 할 프로젝트 선정 완료

- 후보 : 영화예매시스템, 호텔예약시스템, 코로나확진자 알림 시스템, 경기지역화폐 사용처 알림 어플 등
- REACT 및 MySql을 이용한 호텔예약시스템 구축으로 선정

역할 분담

- 이영민 : 백엔드 개발, 프론트엔드 참여
- 정재준 : 프론트엔드 개발, github 레포지토리 생성 및 코드 정리

추후 진행할 사항

- REACT 및 MySql, 아마존 AWS에 대한 공부 및 기본적인 설계

2020.11.28. 회의내용 (만나서 1시간가량 진행)

README.md 작성 (간단하게)

- 역할 분담 및 웹앱에 대한 간략한 설명
- 의사소통 방식 선정 (전화통화 및 디스코드 활용, 실시간으로 피드백을 하고 그때그때 반영하기 위함)

2020.12.04. 회의내용 (전화통화로 20분가량 진행)

진행 할 프로젝트 변경

- REACT 및 MySql을 이용한 호텔예약시스템 구축에서 고객정보입력 웹앱 개발로 변경
- 사유 : 부족한 시간 및 능력 부족

추후 진행할 사항

- 필요한 부분에 대해서 공부한 후 진행

2020.12.05. 회의내용 (만나서 1시간가량 진행)

금주 및 차주는 시험기간이라 본격적인 개발을 할 여유가 없음

원활한 협업을 위해 각자 필요한 만큼 코딩 후 시험이 끝나는 12월 17일부터 깃허브에 게시, 병합

부족한 시간으로 인해 개발 시 상시 전화 혹은 디스코드 연결 후 라이브화면 송출을 통하여 의사소통, 깃에 커밋

추후 진행할 사항

- github에 개발 로그 남기기
- 부족한 공부 실시

2020.12.18. 회의내용 (만나서 2시간가량 진행)

본격적인 개발 시작

- 프론트엔드 개발 (정재준)
- 백엔드 서버 구축 및 백엔드 개발 (이영민)

2020.12.19. 회의내용 (디스코드로 8시간가량 진행)

코딩 내역 github에 게시

https://github.com/Jaejuna/JohnCena_Final_Project/commits/main

풀 이후 실행 여부 확인

디스코드 회의 및 개발 진행 내용 캡처

DISCORD

대화 찾기 또는 시작하기

친구

Nitro

개인 메시지 +

Jun

박근호

안혁진

이동준

엄승훈
ARK: Survival Evolved 하는 중

이재빈

이종원

교수님

chan0966

richtuna

React App - Chrome

음성 연결됨
@Jun

동영상

화면

이영민
#5275

행님
저 피자 좀 먹고 올게요

이영민
dd
오후 6:54
오ㅋ

Jun
오늘 오후 7:34
계심까 행님

이영민
오늘 오후 7:34
d o

Jun
오늘 오후 7:35
지금 메이 올라가데
@Jun에 메시지 보내기


번호	이름	이메일	성별	직업	상태
1	KIM	980103	M	student	이름
2	RAISONND	991102	M	professor	이름
3	NASAKURA	980708	F	teacher	이름
4	JANE	000415	F	teacher	이름

오전 9:15

카톡 기록 캡처


정재준

<https://github.com/Jaejuna>



Jaejuna - Overview
정재준 99. Jaejuna has 15 repositories availabl...
github.com

https://github.com/Jaejuna/JohnCena_Final_Project



Jaejuna/JohnCena_Final_Project
2020 오픈소스개발기초 수업 John Cena팀 기말 프...
github.com

오전 12:08

소프트웨어 개발

프로젝트 소개 - 개요, 목표 및 배경, 진행방법, 참여방법안내

리포지터리 - 구성도 및 설계 문서, 참여자의 커밋 로그, 참여자의 기여도

커뮤니케이션 기록 - 카톡 대화

오전 12:17

정재준

저 씻고 있을께 필요하시면 부르세영

오전 12:30

오전 12:30

ㅇㅇㅇㅇ

정재준

지금 부터 개발 중

행님

오늘 몇시에 주무실 예정이심까

오후 8:23

와 방금일어남...

오후 9:50

오늘안자 ㅎㅎ

나

한시간만이따가 깨워주라....

오후 9:59

정재준

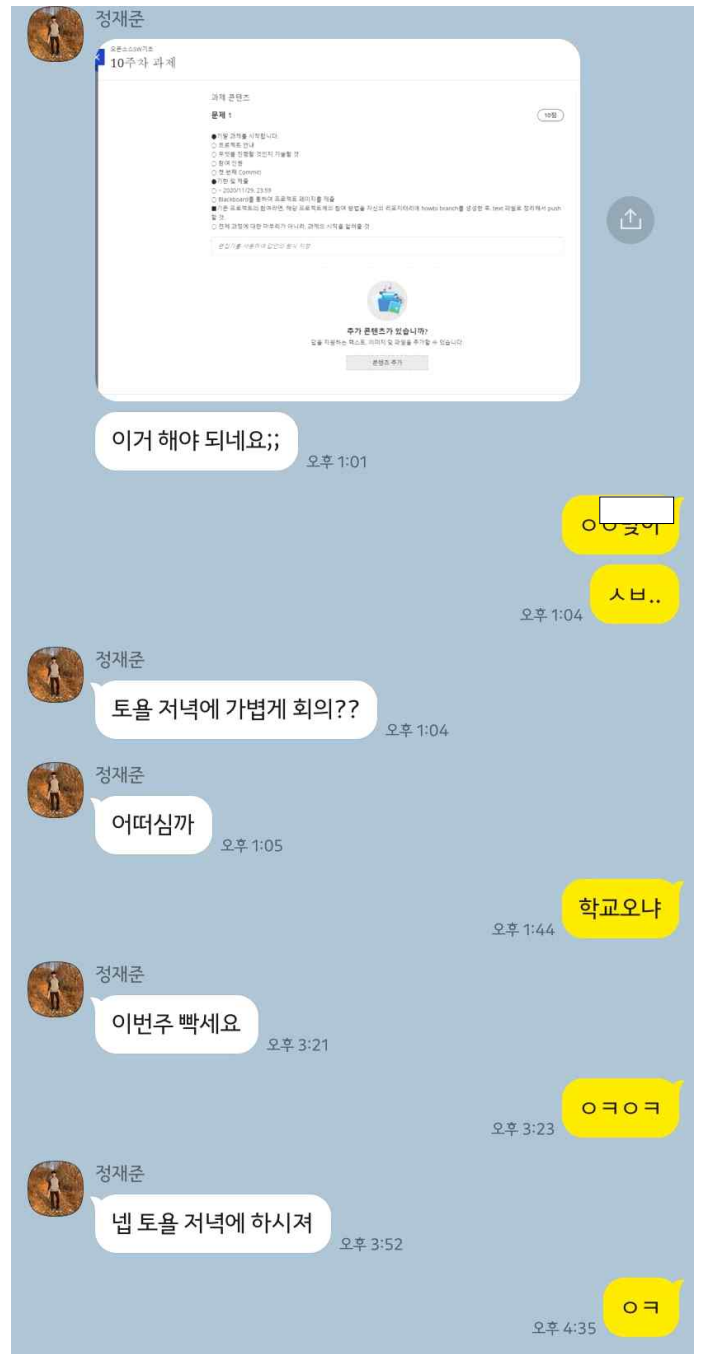
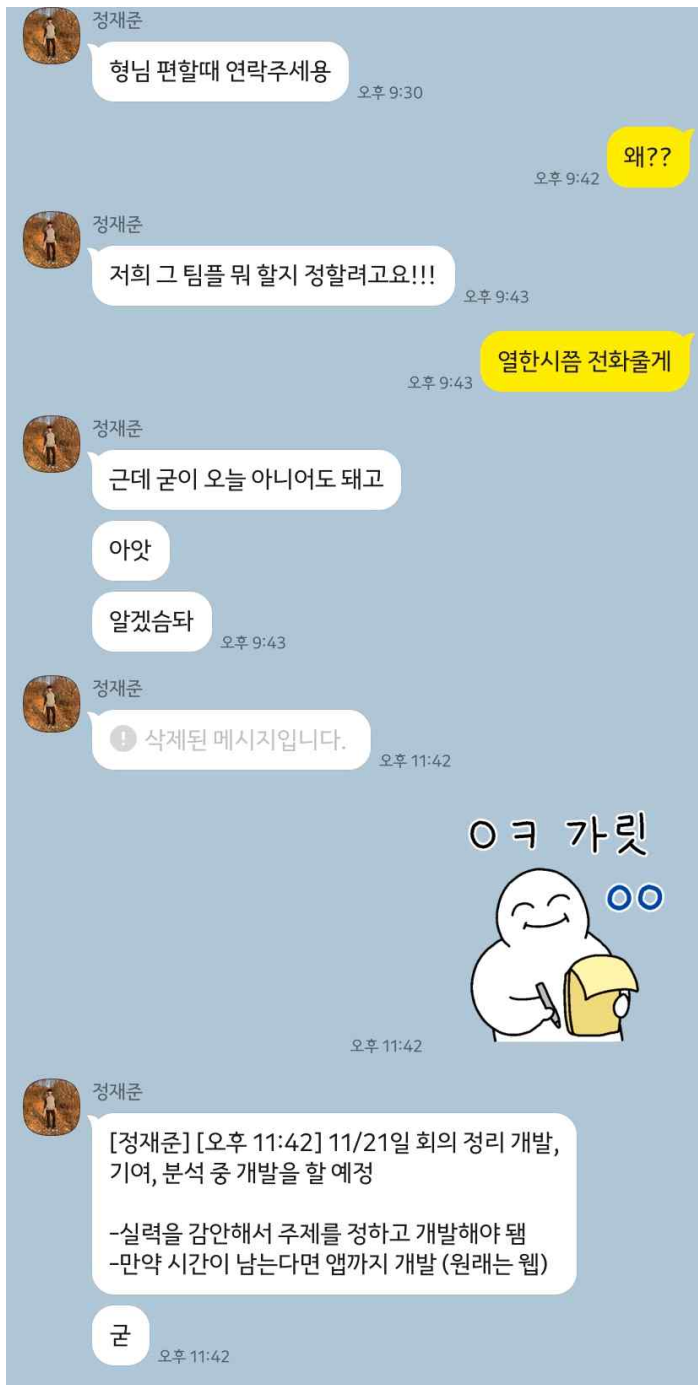
알겠습니다

오후 10:02

정재준

한시간 됐습다 행님

오후 11:10



코드 리뷰

DB서버는 이영민의 로컬호스트에서 불러오기로 진행
MySQL을 아마존 AWS에 연동하여 백엔드 서버 구축
React 및 MySQL에 대한 기본적인 사용법을 공부하기 위해 프로젝트 진행
사용 프로그램 : VSCode, HeidiSql

1. 아마존 AWS에 DB 생성

The screenshot displays the Amazon RDS console for a MySQL database instance named 'database-1'. The instance is in the '사용 가능' (Available) state. Key configuration details include:

- 요약 (Summary):**
 - DB 식별자: database-1
 - CPU: 2.30%
 - 상태: 사용 가능
 - 클래스: db.t2.micro
 - 역할: 인스턴스
 - 현재 활동: 1 연결
 - 엔진: MySQL Community
 - 리전 및 AZ: ap-northeast-2a
- 연결 & 보안 (Connect & Security):**
 - 엔드포인트 및 포트:** 엔드포인트: database-1.cgaelp5fw3n.ap-northeast-2.rds.amazonaws.com, 포트: 3306
 - 네트워킹:** 가용 영역: ap-northeast-2a, VPC: vpc-72fc4219, 서버넷 그룹: default-vpc-72fc4219, 서버넷: subnet-04a0086f, subnet-ab1509e7, subnet-22630659, subnet-cc4da393
 - 보안:** VPC 보안 그룹: default (sg-6ace0f12) (활성), 퍼블릭 액세스 가능성: 예, 인종 기관: rds-ca-2019, 인증 기관 날짜: Aug 23rd, 2024

2. HeidiSql을 통해 아마존 AWS에 있는 DB 접근

The screenshot shows the HeidiSQL connection configuration window. The configuration is as follows:

- 네트워크 유형:** MySQL (TCP/IP)
- Library:** libmysql.dll
- 호스트명 / IP:** database-1.cgaelp5fw3n.ap-northeast-2.rds.amazonaws.com
- 사용자:** admin
- 암호:** (masked with dots)
- 포트:** 3306
- 데이터베이스:** 세미콜론으로 구분
- 코멘트:** (empty)

Additional options shown include checkboxes for '자격 증명 프로토폴' and 'Windows 인증 사용', and a checkbox for '압축된 클라이언트/서버 프로토콜'.

3. 저장된 데이터 및 테이블 확인

management# - HeidiSQL 11.1.0.6116

파일 편집 검색 쿼리 도구 이동 도움말

호스트: database-1.cgaelpt5fw3n.ap-northeast-2.rds.amazonaws.com 쿼리*

management

- information_schema
- management
- mysql
- performance_schema
- sys

```
1 USE management;
2
3 SELECT * FROM CUSTOMER;
```

Filter ...

- > 열
- > SQL 함수
- > SQL 키워드
- > 스키마

CUSTOMER (4r x 8c)

id	image	NAME	birthday	gender	job	createdDate	isDeleted
1	https://placeimg.com/64/64/1	KIM	960703	M	student	2020-12-19 07:59:30	0
2	https://placeimg.com/64/64/2	RAYMOND	991102	M	professor	2020-12-19 07:59:30	0
3	https://placeimg.com/64/64/3	NAKAMURA	980708	F	teacher	2020-12-19 07:59:30	0
4	https://placeimg.com/64/64/4	JANE	000415	F	casher	2020-12-19 07:59:30	0

X 필터: 정규 표현식

```
3 SELECT CONNECTION_ID();
4 /* 문자 집합: utf8mb4 */
5 SHOW /*!50002 GLOBAL */ STATUS;
6 SELECT NOW();
7 SHOW VARIABLES;
8 /* 연결된 스레드-ID: 208 */
9 SHOW DATABASES;
10 /* "management" 섹션 시작 */
11 /* Loading file "C:\Users\young\AppData\Roaming\HeidiSQL\Backups\query-tab-2020-12-19_12-04-26-236.sql" (42 B) into query tab #1 ... */
12 USE management;
13 SELECT * FROM CUSTOMER;
14 /* 연결된 스레드-ID: 208 */
```

r3 : c24 (45 B) 연결됨: 00:00 h MySQL 8.0.20 가동 시간: 14:35 h 서버 시간: 오후 1 유류

4. 웹앱 실행 시 기본화면





search module added - Jaajun x 한양대학교 포털 x 과제 제출 x RDS - AWS Console x React App x +

localhost:3000

고객 관리 시스템

검색하기

고객 추가하기

번호	프로필 이미지	이름	생년월일	성별	직업	설정
1		KIM	960703	M	student	<button>삭제</button>
2		RAYMOND	991102	M	professor	<button>삭제</button>
3		NAKAMURA	980708	F	teacher	<button>삭제</button>
4		JANE	000415	F	casher	<button>삭제</button>

5. 고객 추가하기 실행

search module added - Jaejun...한양대학교 포털과제 제출RDS - AWS ConsoleReact Applocalhost:3000

고객 관리 시스템

고객 추가하기

번호	프로필 이미지	이름	생년월일	성별	직업	설정
1		KIM	960703	M	student	<div>삭제</div>
2		RAYMOND			professor	<div>삭제</div>
3		NAKAMURA			teacher	<div>삭제</div>
4		JANE			casher	<div>삭제</div>

고객 추가

파일 선택

회장.jpg

C:\FAKEPATH\회장.JPG

이름

JEONG

생년월일

961111

성별

M

직업

progamer

추가

닫기

search module added - Jaejun...한양대학교 포털과제 제출RDS - AWS ConsoleReact Applocalhost:3000

고객 관리 시스템

고객 추가하기

번호	프로필 이미지	이름	생년월일	성별	직업	설정
1		KIM	960703	M	student	<div>삭제</div>
2		RAYMOND	991102	M	professor	<div>삭제</div>
3		NAKAMURA	980708	F	teacher	<div>삭제</div>
4		JANE	000415	F	casher	<div>삭제</div>
8		JEONG	961111	M	progamer	<div>삭제</div>

6. DB 확인

management# - HeidiSQL 11.1.0.6116

파일 편집 검색 쿼리 도구 이동 도움말

데이터베이스 필터 테이블 필터

management

information_schema
management
mysql
performance_schema
sys

호스트: database-1.cgaelp5fw3n.ap-northeast-2.rds.amazonaws.com 쿼리*

1
2
3

USE management;

SELECT * FROM CUSTOMER;

Filter ...
열
SQL 함수
SQL 키워드
스니펫

CUSTOMER (5r x 8c)

id	image	NAME	birthday	gender	job	createdDate	isC
1	https://placeimg.com/64/64/1	KIM	960703	M	student	2020-12-19 07:59:30	
2	https://placeimg.com/64/64/2	RAYMOND	991102	M	professor	2020-12-19 07:59:30	
3	https://placeimg.com/64/64/3	NAKAMURA	980708	F	teacher	2020-12-19 07:59:30	
4	https://placeimg.com/64/64/4	JANE	000415	F	casher	2020-12-19 07:59:30	
8	/image/29ed286b0e44c43975eddd256baddb8f	JEONG	961111	M	progamer	2020-12-19 13:16:03	

필터: 정규 표현식

6
7
8
9
10
11
12
13
14
15
16
17

SELECT NOW();
SHOW VARIABLES;
/* 연결됨. 스키마 -ID: 208 */
SHOW DATABASES;
/* "management" 세션 시작 */
/* Loading file "C:\Users\young\AppData\Roaming\HeidiSQL\Backups\query-tab-2020-12-19_12-04-26-236.sql" (42 B) into query tab #1 ... */
USE management;
SELECT * FROM CUSTOMER;
/* 연결 받은 열: 8 rows 열: 4 행: 0 평균: 0 지속 시간: 2 쿼리: 0.015 초 */
USE management;
SELECT * FROM CUSTOMER;
/* 연결 받은 열: 8 rows 열: 5 행: 0 평균: 0 지속 시간: 2 쿼리: 0.031 초 */

r3 : c24 (45 B) 연결됨: 00:23 h MySQL 8.0.20 가동 시간: 14:58 h 서버 시간: 오후 1 유류

7. 고객 삭제하기 실행

search module added - Jaehun x 한양대학교 포털 x 과제 제출 x RDS - AWS Console x React App x +

localhost:3000

고객 관리 시스템

검색하기

고객 추가하기

번호	프로필 이미지	이름	생년월일	성별	직업	설정
1		KIM	960703	M	student	삭제
2		RAYMOND	991102	M	professor	삭제
3		NAKAMURA		F	teacher	삭제
4		JANE		F	casher	삭제
8		JEONG	961111	M	progamer	삭제

삭제 경고

선택한 고객 정보가 삭제됩니다.

삭제 닫기

검색하려면 여기에 입력하십시오.

오후 10:17 2020-12-19

8. DB 확인

management - HeidiSQL 11.1.0.6116

파일 편집 검색 쿼리 도구 이동 도움말

데이터베이스 필터 테이블 필터

호스트: database-1.cgaelpt5fw3n.ap-northeast-2.rds.amazonaws.com 쿼리+

management

- information_schema
- management
- mysql
- performance_schema
- sys

1 USE management;

2

3 SELECT * FROM CUSTOMER;

CUSTOMER (5r x 8c)

id	image	NAME	birthday	gender	job	createdDate	isDeleted
1	https://placeimg.com/64/64/1	KIM	960703	M	student	2020-12-19 07:59:30	0
2	https://placeimg.com/64/64/2	RAYMOND	991102	M	professor	2020-12-19 07:59:30	0
3	https://placeimg.com/64/64/3	NAKAMURA	980708	F	teacher	2020-12-19 07:59:30	0
4	https://placeimg.com/64/64/4	JANE	000415	F	casher	2020-12-19 07:59:30	0
8	/image/29ed286b0e44c43975eddd256baddb8f	JEONG	961111	M	programer	2020-12-19 13:16:03	1

필터: 정규 표현식

9 SHOW DATABASES;

10 /* "management" 세션 시작 */

11 /* Loading file "C:\Users\young\AppData\Roaming\HeidiSQL\Backups\query-tab-2020-12-19_12-04-26-236.sql" (42 B) into query tab #1 ... */

12 USE management;

13 SELECT * FROM CUSTOMER;

14 /* 연결: 0.015 초, 쿼리: 0.015 초, 결과: 4 행, 0 열, 0.015 초 */

15 USE management;

16 SELECT * FROM CUSTOMER;

17 /* 연결: 0.015 초, 쿼리: 0.015 초, 결과: 5 행, 0 열, 0.015 초 */

18 USE management;

19 SELECT * FROM CUSTOMER;

20 /* 연결: 0.015 초, 쿼리: 0.015 초, 결과: 5 행, 0 열, 0.015 초 */

연결됨: 00:24 h MySQL 8.0.20 가동 시간: 14:59 h 서버 시간: 오후 1: 유류

CUSTOMER 테이블에서 isDeleted의 값이 1로 변하면서 웹앱상에서 출력되지 않음

9. server.js 코드 확인

server.js - JohnCena_Final_Project - Visual Studio Code

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)

server.js

```
final > JS server.js > app.delete('/api/customers/:id') callback
23 const multer = require('multer');
24 const upload = multer({dest: './upload'})
25
26 app.get('/api/customers', (req, res) => {
27   connection.query(
28     "SELECT * FROM CUSTOMER WHERE isDeleted = 0",
29     (err, rows, fields) => {
30       res.send(rows);
31     }
32   );
33 });
34
35 app.use('/image', express.static('./upload'));
36
37 app.post('/api/customers', upload.single('image'), (req, res) => {
38   let sql = 'INSERT INTO CUSTOMER VALUES (null, ?, ?, ?, ?, ?, ?, now(), 0)';
39   let image = '/image/' + req.file.filename;
40   let NAME = req.body.NAME;
41   let birthday = req.body.birthday;
42   let gender = req.body.gender;
43   let job = req.body.job;
44   let params = [image, NAME, birthday, gender, job];
45   connection.query(sql, params,
46     (err, rows, fields) => {
47       res.send(rows);
48     }
49   );
50 });
51
52 app.delete('/api/customers/:id', (req, res) => {
53   let sql = 'UPDATE CUSTOMER SET isDeleted = 1 WHERE id = ?';
54   let params = [req.params.id];
55   connection.query(sql, params,
56     (err, rows, fields) => {
57       res.send(rows);
58     }
59   );
60 });
61
62 app.listen(port, () => console.log('Listening on port $(port)'));
```

문제: 클라이언트 콘솔 터미널

1: node

[1] Search for the keywords to learn more about each warning.

[1] To ignore, add // eslint-disable-next-line to the line before.

[1] Compiling...

[1] Compiled successfully!

CUSTOMER의 정보를 가져올 때, WHERE isDeleted = 0 인 조건을 주어서 삭제버튼이 눌리지 않은 데이터를 가져옴.
삭제버튼을 누르면 isDeleted 값이 1로 변함.