

# 구현노트

A1

김원배 박서연 오익준

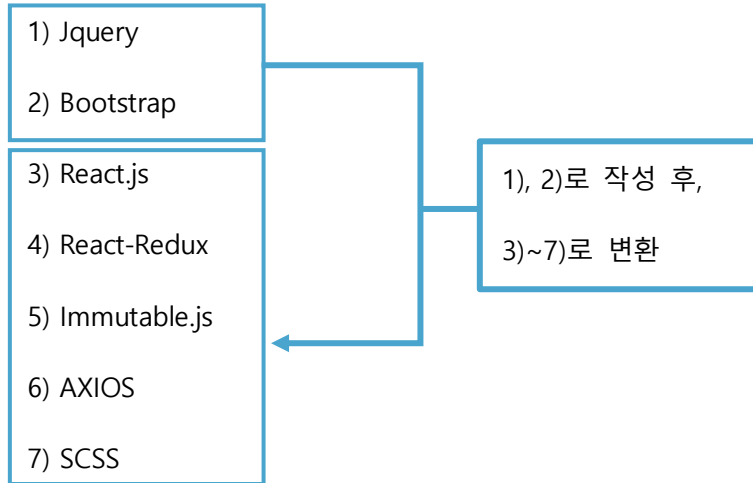
이세명 이효원 한예진

## 목차

1. 구현 프로그램 선정
  - A. Front
  - B. Back
2. 기능 구현
  - A. 고객
  - B. 관리자
3. 백엔드 구현

# 1. 구현 프로그램 선정

## A. Front



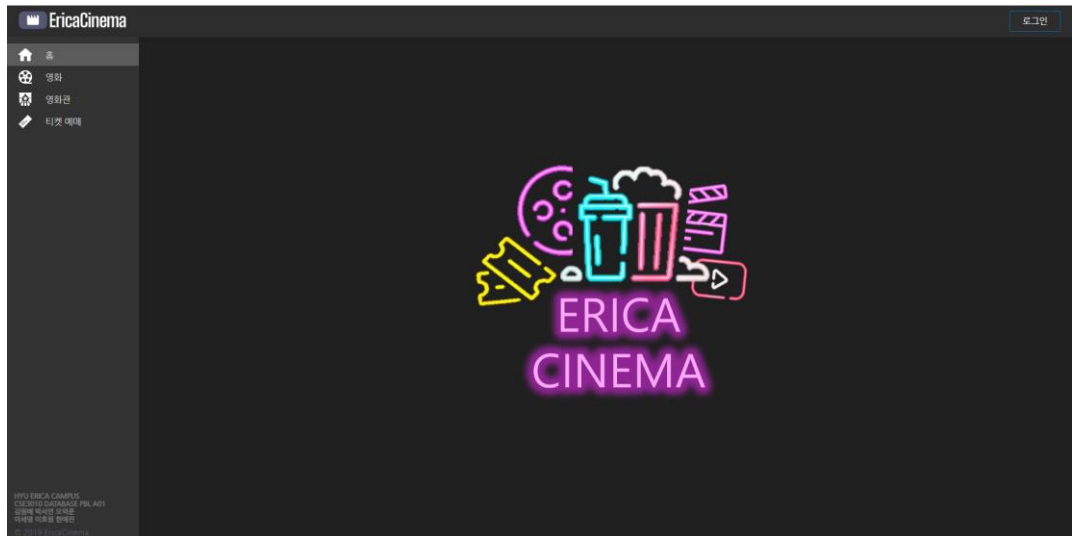
## B. Back

- 1) Node.js
- 2) Express.js
- 3) JWT
- 4) Maria DB

## 2. 기능 구현

### A. 고객

#### 1) 메인 페이지



## 2) 현재 상영 영화 목록



### 3) 영화관 목록

서울

경기

대전/충청

인천

강원

대구

부산/울산

강남

대학로

영화관 서울 - 대학로

주소 - 서울 종로구 태평로 (영륜동)

번호 - 1111-1112

영화관 층별 안내도

보기

상영시간표

12월 10일

18

익준이 군대간다

18분 | 2020.03.02 개봉

제 1관 (3D) 총 44석

14:00 39석

19

나를 찾아줘

107분 | 2019.11.27 개봉

제 2관 (2D) 총 108석

19:00 108석

15

블랙머니

113분 | 2019.11.13 개봉

제 3관 (3D) 총 108석

12:00 46석

12

라스트 크리스마스

103분 | 2019.12.05 개봉

제 1관 (3D) 총 44석

15:00 44석

1F 2F

원하는 층 선택

영화관 층별 안내도

CGV

'보기' 클릭 시

층별안내도 사진 로드

영화

지역

영화관

날짜

18

익준이 군대간다

서울 (1)

대학로

12월 10일

18

익준이 군대간다

서울 (1)

대학로

제 3관 (3D)

19

나를 찾아줘

서울 (1)

대학로

12:00 46석

12

라스트 크리스마스

서울 (1)

대학로

15:00 44석

12

나이트비스트 아웃

서울 (1)

대학로

12

포드 V 페라리

서울 (1)

대학로

12

파비안느에 관한...

서울 (1)

대학로

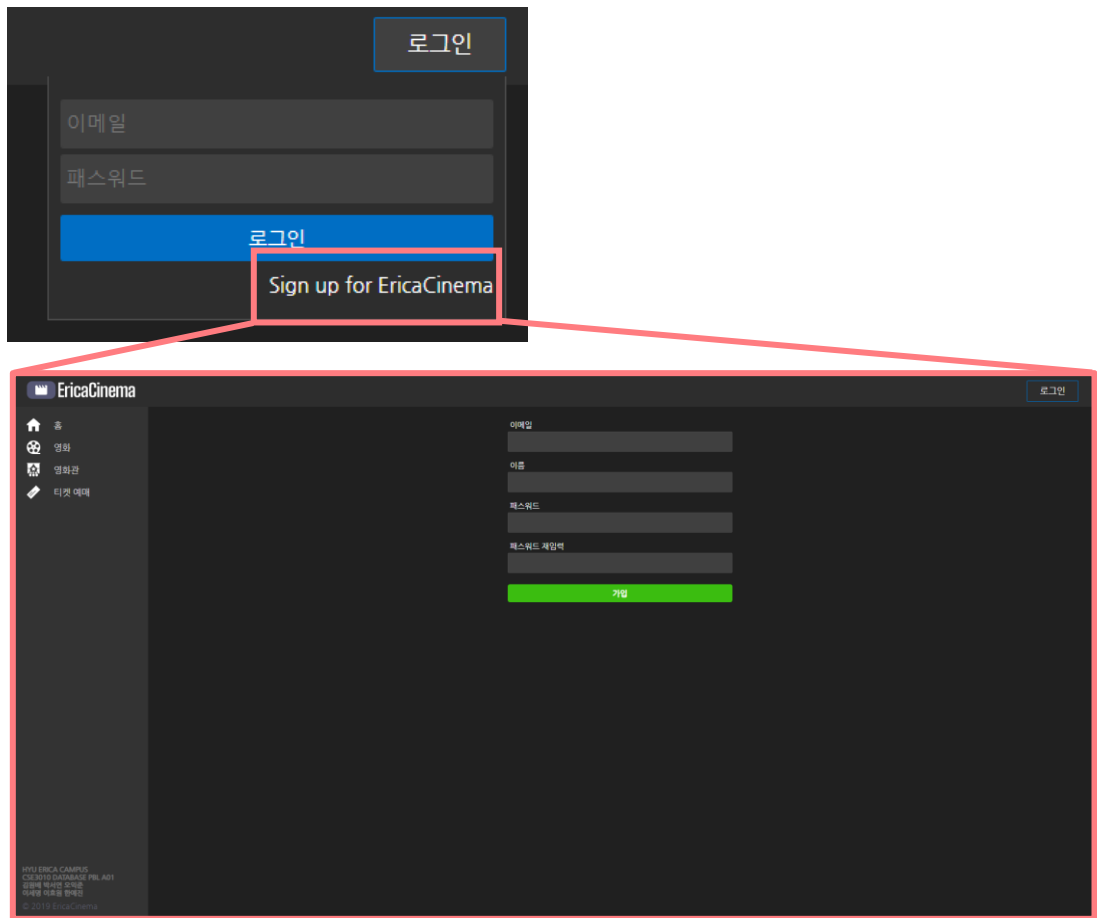
초기화

좌석선택

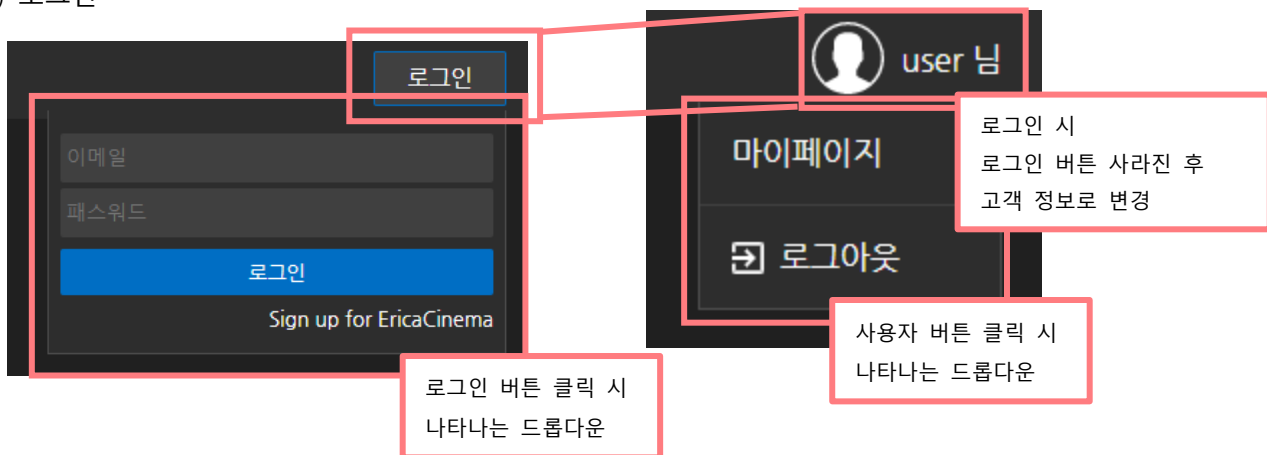
클릭 시 해당 영화

예매 창으로 넘어감

#### 4) 회원가입



#### 5) 로그인



## 6) 예매 및 결제

The interface is divided into several sections:

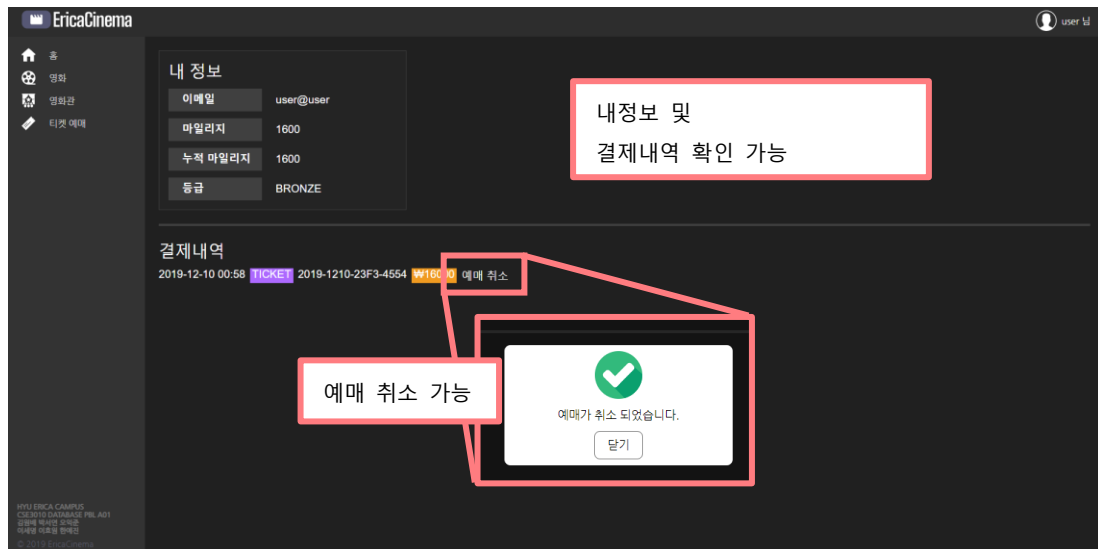
- Movie Selection:** A list of movies is shown on the left, including "익준이 군대간다" (Ikjun's Army Goes), "겨울왕국 2" (Frozen 2), "나를 찾아줘" (Find Me), "블랙머니" (Black Money), "라스트 크리스마스" (Last Christmas), "나이스 아웃" (Nice Out), "모드 V 페라리" (Mode V Ferrari), and "파비안스에 관한..." (About Fabian...). A red arrow points to the "익준이 군대간다" movie.
- Theater Selection:** A table shows theaters and their locations. "대학로" (Daehak-ro) is selected. A red arrow points to the "대학로" theater.
- Screen Selection:** A table shows the movie "익준이 군대간다" at "대학로" on "2019-12-10 14:00". A red arrow points to the "대학로" theater.
- Seat Selection:** A grid of seats is shown. Seats B4 and B5 are selected. A red arrow points to the "좌석선택" (Seat Selection) button.
- Payment:** A "결제하기" (Pay) button is shown. A red arrow points to the "결제하기" button.
- Confirmation:** A "예매 완료" (Booking Complete) screen is shown. A red arrow points to the "예매 완료" screen.

Additional details from the interface:

- Movie Details:** "익준이 군대간다" (Ikjun's Army Goes), 서울 - 대학로 (Seoul - Daehak-ro), 제 1관 (3D) (Screen 1 (3D)), 2019-12-10 14:00.
- Seat Details:** B4 normal ₩8000, B5 normal ₩8000.
- Total Price:** 총 16,000원 (Total 16,000 won).
- Payment Method:** 휴대폰 결제 (Mobile Payment).
- Payment Fields:** 이름 (Name), 주민등록번호 (Residence Registration Number), 휴대폰 전화번호 (\* 제외) (Mobile Phone Number (\* Excluded)).
- Payment Amount:** ₩16,000.
- Buttons:** 취소 (Cancel), 결제 (Pay).

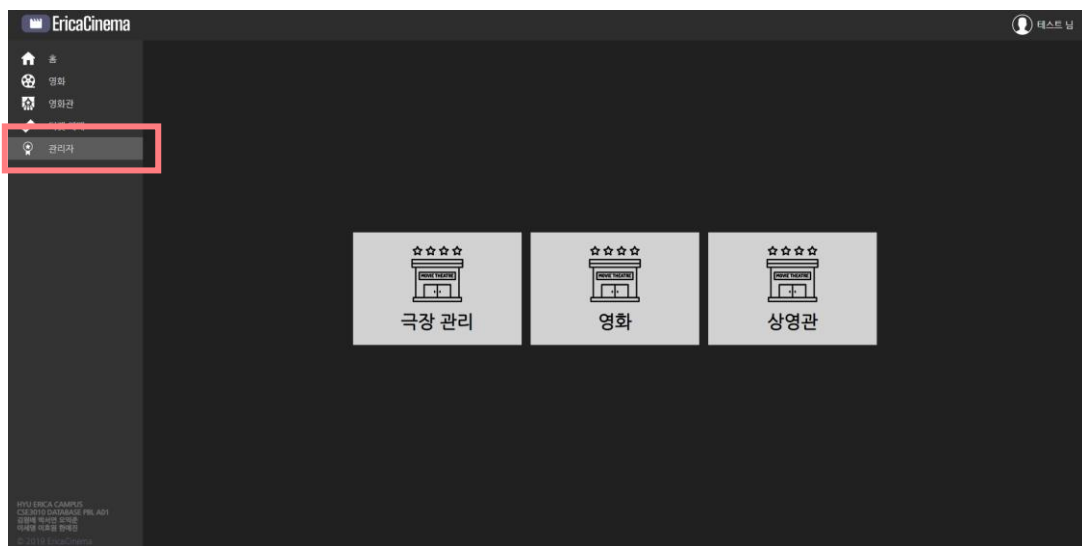


## 7) 마이페이지



## B. 관리자

### 1) 관리자 로그인 시 관리자 탭 생성



## 2) 극장 관리

극장 목록

극장 추가

상영관 관리

극장 고유번호

지역 선택

극장 이름

주소

전화번호

층

세부사항

층 추가

제출

이미지 추가 삭제

층별 안내도 추가

### 3) 영화 관리

영화 추가

영화 통계

영화 고유번호

영화 제목

영화 영어 제목

영화 런타임

영화 제작년도

영화 제작국가

영화 개봉일

영화 줄거리

영화 관람 등급

썸네일

업로드 / 변경

■ 현재 상영중

스틸컷 추가

제출

### 4) 상영관 관리

영화관 선택

-> 상영관 선택

-> 상영영화 선택

-> 상영시간 선택

홈

상영관 영화 관리

서울

강남

대학로

선택 극장 : 서울 - 대학로

제 1관 (3D) - 총 좌석 44

14:0 - 14:43

15:0 - 17:8

겨울왕국 2

영화 시작시간 설정 (날짜 시간 분)

2019-12-10

17

30

제출

×

이 시간대에 영화를 추가 할 수 없습니다.

닫기

겹치는 시간

영화 상영추가 불가

선택한 상영관의

이전에 선택한 시간 표시

### 3. 백엔드 구현

```
JS index.js
src > api > JS index.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const user = require('./user');
4  const movie = require('./movie');
5  const theater = require('./theater');
6  const ticket = require('./ticket');
7  const admin = require('./admin');
8
9  const logMiddleware = async (req, res, next) => {
10     const ip = req.headers['x-forwarded-for'] || req.connection.remoteAddress;
11     console.log(`IP: ${ip} \n referer: ${req.headers['referer']} \n originalUrl: ${req.originalUrl}`);
12     next();
13 }
14
15 router.use(logMiddleware);
16
17 router.use('/user', user);
18 router.use('/movie', movie);
19 router.use('/theater', theater);
20 router.use('/ticket', ticket);
21 router.use('/admin', admin);
22
23
24 module.exports = router;
```

메인 라우터

각각 서버라우터들이 있음.

데이터베이스에 튜플을 삽입, 삭제할때 모두 트랜잭션 처리를 함.

로그인 구현

```

  user
  JS controller.js
  JS index.js
  JS sql.js
```

유저 기능만을 담당하는 라우터 및 컨트롤러 (서비스 단)

## JWT (Json Web Token) 사용



JWT 의 설정값들을 관리하는 스크립트 생성

```
rc > config > JS jwt.config.js > [🔗] expiresIn
1  const algorithm = "HS256";
2  const secretKey = 'erica_cinema';
3  const expiresIn = '12h'; // 유효 시간
4
5  exports.algorithm = algorithm;
6  exports.secretKey = secretKey;
7  exports.expiresIn = expiresIn;
```

클라이언트에서 로그인 하는 경우가 두가지가 나뉨

로컬에 저장된 토큰을 가지고 로그인, 토큰이 없는 상태에서의 로그인

이미 토큰을 가지고 있을 경우 토큰을 검증하고 페이로드에 저장된 벨류들을 되돌려줌

없을 경우, 사용자가 입력한 값을 데이터베이스에 넘겨 검증시킴

검증이 완료되면 토큰을 생성하여 토큰을 반환함

## 회원가입 구현

사용자가 입력한 데이터가 올바른 값인지 검증을 한 후, 데이터베이스에 넣는 작업을 함.

## 관리자 페이지 구현

관리자페이지에서 요청하는 모든 요청들의 헤더에 Authorization에 토큰을 넣어 서버측에 요청을 보내면 서버에서 토큰을 검증하고 데이터베이스에서 사용자가 관리자인지 확인하는 작업을 함.

서버에서는 유저가 관리자인 경우에만 모든 유저에게 열람 되어지는 정보를 제외한 정보를 제공함. (열람 및 추가 할 때)

이미지 업로드의 경우에는 npm에 있는 multer 를 사용하여 업로드

영화관 구현

Field	Type	Null	Key	Default	Extra
code	int(10) unsigned	NO	PRI	NULL	
location	varchar(20)	YES		NULL	
name	varchar(20)	NO		NULL	
address	varchar(200)	YES		NULL	
phone	varchar(20)	YES		NULL	

Location은 영화관이 위치해 있는 지역 (서울, 경기 등) 을 의미

이부분 의 경우에는 추후에 정규화 작업이 필요

```
for the right syntax to use near 'theater_guide_map' at line 1
MariaDB [cinema]> desc theater_guide_map;
```

Field	Type	Null	Key	Default	Extra
theater_code	int(10) unsigned	NO	PRI	NULL	
floor	smallint(6)	NO	PRI	NULL	
comment	varchar(50)	YES		NULL	
img	varchar(200)	YES		NULL	

영화관 층별안내도. img는 서버내부 이미지가 저장된 경로를 저장함.

Comment는 클라이언트 상에서 보여질 층별문구 예) 1F 2F B1, 8층 서관 등

## 영화관 상영관 구현

Field	Type	Null	Key	Default	Extra
theater_code	int(10) unsigned	NO	PRI	NULL	
code	tinyint(3) unsigned	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
total_seats	tinyint(3) unsigned	NO		NULL	
row	tinyint(3) unsigned	NO		NULL	
col	tinyint(3) unsigned	NO		NULL	

(영화관, 상영관 코드) 가 PK

Row, col 은 상영관마다 크기 및 배치가 다르기 때문에 이를 구현하기 위해 넣음

## 영화관 상영관 좌석 구현

Field	Type	Null	Key	Default	Extra
theater_code	int(10) unsigned	NO	PRI	NULL	
screen_code	tinyint(3) unsigned	NO	PRI	NULL	
x	tinyint(3) unsigned	NO	PRI	NULL	
y	tinyint(3) unsigned	NO	PRI	NULL	
code	varchar(4)	NO		NULL	
grade	varchar(10)	NO		NULL	

관리자페이지에서 상영관 좌석을 디자인하여 서버에 제출하면

먼저 상영관에 기본적인 정보를 넣고 좌석 정보를 따로 DB에 넣어주기 위해 만든 테이블

X, y 는 각각 행, 열의 위치 예) (3, 4) 등 에 좌석이 존재한다는 의미. code의 경우 A01 A02 같은 사용자에게 실질적으로 보여지는 코드를 의미. Grade는 좌석이 기본석인지 커피석 등등 타입을 의미

## 영화 구현

먼저 테이블을 기본 영화정보 와 영화 스틸컷을 넣을 수 있는 테이블 두개를 만들.

```
MariaDB [cinema]> desc movie;
```

Field	Type	Null	Key	Default	Extra
code	varchar(20)	NO	PRI	NULL	
title	varchar(100)	YES		NULL	
title_en	varchar(100)	YES		NULL	
run_time	smallint(6)	YES		NULL	
product_year	varchar(10)	YES		NULL	
nation	varchar(50)	YES		NULL	
open	varchar(10)	YES		NULL	
synopsis	text	YES		NULL	
watch_grade	varchar(30)	YES		NULL	
thumbnail	varchar(200)	YES		NULL	
show_status	tinyint(1)	NO		0	
score_rate	decimal(3,1)	YES		NULL	
book_rate	decimal(4,1)	YES		NULL	
cumulative	int(11)	YES		NULL	

```
MariaDB [cinema]> desc movie_stillcut;
```

Field	Type	Null	Key	Default	Extra
movie_code	varchar(20)	NO	PRI	NULL	
id	tinyint(3) unsigned	NO	PRI	NULL	
is_external	tinyint(1)	NO		0	
img	varchar(1000)	YES		NULL	

스틸컷에서 movie\_code는 무비테이블의 code 외래키 참조

영화의 세부적인 내용들 예) 장르, 상영타입(4d, 3d, 2d, 2d 더빙 등), 감독, 배우 등은 무비의 code 를 외래키로 참조 하는 테이블을 따로 생성하여 추가할 수 있음. (확장성)



## 영화관별 상영시간 구현

```
MariaDB [cinema]> desc theater_movie_time;
```

Field	Type	Null	Key	Default	Extra
theater_code	int(10) unsigned	NO	PRI	NULL	
screen_code	tinyint(3) unsigned	NO	PRI	NULL	
movie_code	varchar(20)	NO	PRI	NULL	
start	datetime	NO	PRI	NULL	
end	datetime	NO		NULL	

5 rows in set (0.015 sec)

상영관에 영화시간표를 추가할 때 겹치는 시간에 추가 할 수 없도록 다음과 같은 쿼리문을 작성

```
const checkTimeValid = `
SELECT screen_code, start, end
FROM theater_movie_time
WHERE theater_code = ? AND screen_code = ? AND ((start <= ? AND ? <= end) || (start <= ? AND ? <= end) || (? <= start AND start <= ?) || (? <= end AND end <= ?))
`;
```

## 영화 예매 구현

### 회원 결제 정보 테이블

```
MariaDB [cinema]> desc payment;
```

Field	Type	Null	Key	Default	Extra
email	varchar(100)	NO	PRI	NULL	
time	datetime	NO	PRI	CURRENT_TIMESTAMP	
price	int(11)	NO		NULL	
type	varchar(20)	NO		NULL	
serial_number	varchar(100)	NO		NULL	

회원이 결제를 같은 시간 (밀리세컨드 까지 포함하여) 못한다고 판단하여

(email, time)을 PK로 잡음

Price 는 결제 총가격, type 은 어떤 종류의 상품을 구입했는지에 대한 타입 (예) 티켓, 굿즈, 기프티콘 등)

시리얼 넘버는 결제한 품목의 시리얼 넘버 (예를들면 티켓 타입같은 경우엔 티켓의 일련번호)

## 티켓 테이블

```
MariaDB [cinema]> desc ticket;
```

Field	Type	Null	Key	Default	Extra
serial_number	varchar(50)	NO	PRI	NULL	
user_email	varchar(100)	YES		NULL	

티켓의 일련번호를 저장하는 테이블. 유저이메일의 Null허용이 yes인 이유는 비회원도 있기때문

## 좌석 예매 정보 저장 테이블

```
MariaDB [cinema]> desc ticket_book;
```

Field	Type	Null	Key	Default	Extra
ticket_serial_number	varchar(50)	NO		NULL	
theater_code	int(10) unsigned	NO	PRI	NULL	
screen_code	tinyint(3) unsigned	NO	PRI	NULL	
movie_code	varchar(20)	NO	PRI	NULL	
start	datetime	NO	PRI	NULL	
seat_code	varchar(4)	NO	PRI	NULL	

티켓 시리얼 넘버에는 일련번호가 참조됨.

Theater\_code, screen\_code, movie\_code, start 는 상영시간에 있는 정보와 동일

- ➔ 정규화를 안한 이유는, 따로 또 저장된 다른 테이블을 참조해야 하는데 시간이 촉박하여 일단 그대로 냅둠.