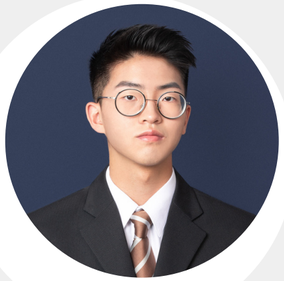
CINEMA

Team 1

Table of Content

Team Members



陳昌右 110590454

蔡佳彣 110590451

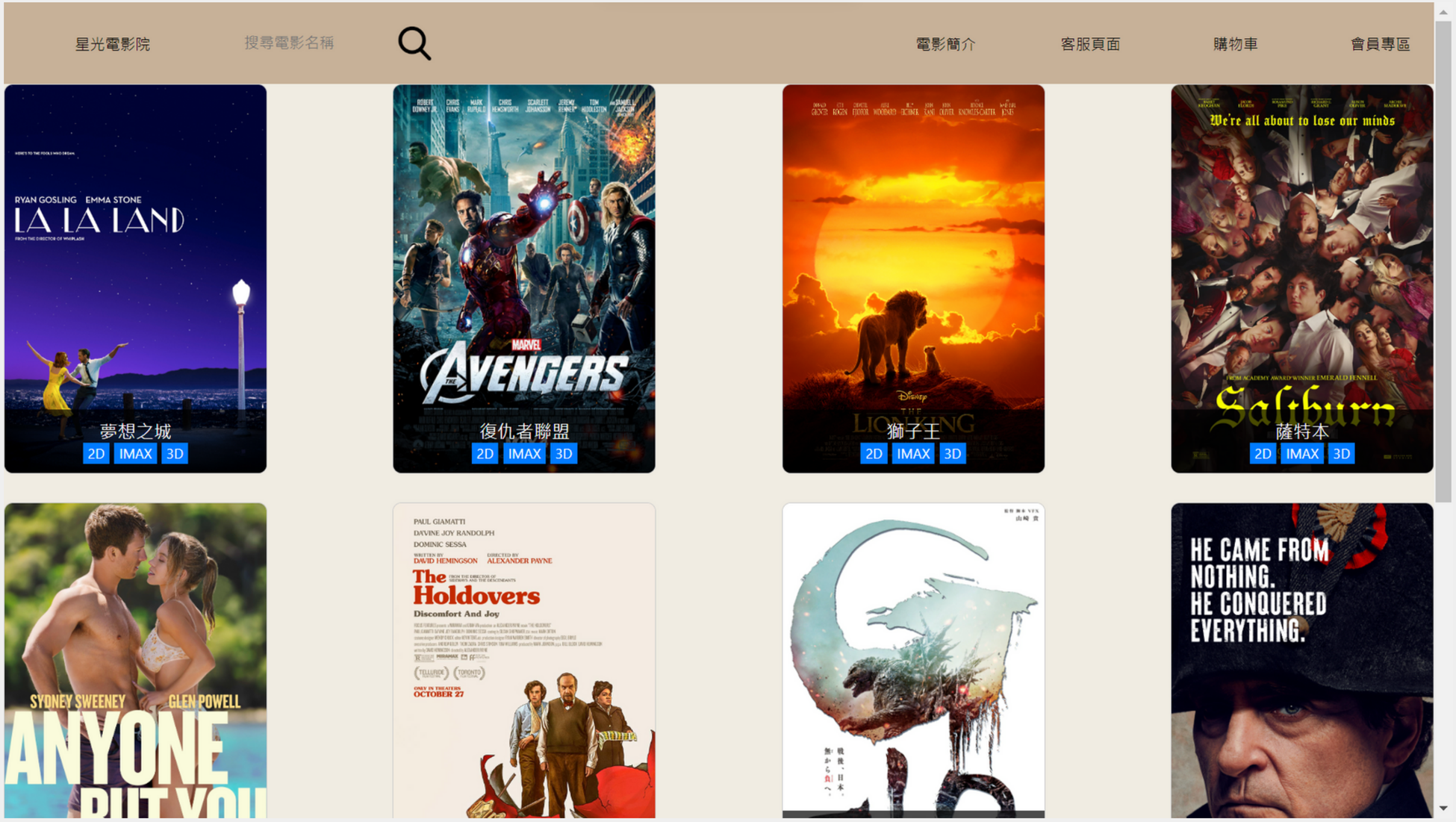
莊于潔 110590452

張哲瑋 107590004

李昶泰 110590455

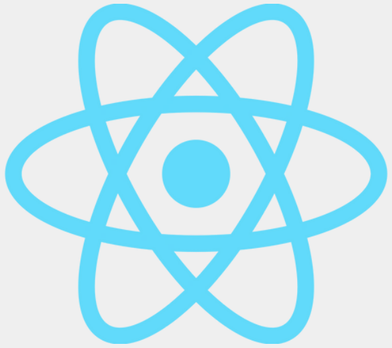
陳毅 110590456

電影院購票系統


星光電影院
搜尋電影名稱
電影簡介
客服頁面
購物車
會員專區
RYAN GOSLING EMMA STONE
LA LA LAND
夢想之城
2D IMAX 3D
復仇者聯盟
2D IMAX 3D
獅子王
2D IMAX 3D
We're all about to lose our minds
薩特本
2D IMAX 3D
SYDNEY SWEENEY GLEN POWELL
ANYONE
PAUL GIAMATTI
DAVINE JOY RANDOLPH
DOMINIC SESSA
DAVID HEMINGSON ALEXANDER PAYNE
The Holdovers
Discomfort And Joy
HE CAME FROM NOTHING.
HE CONQUERED EVERYTHING.

使用技術工具

Front-End



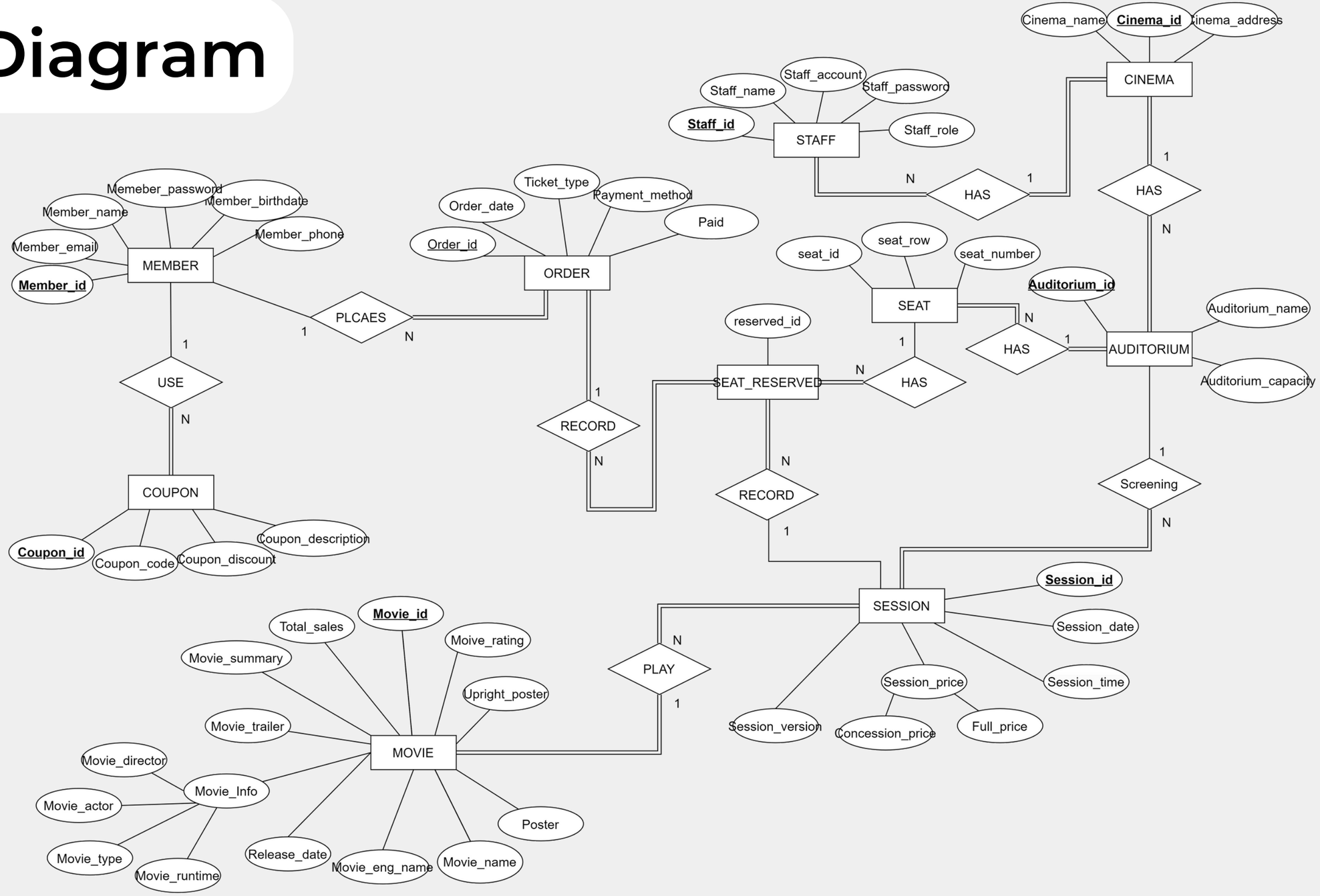
BackEnd

Version Control

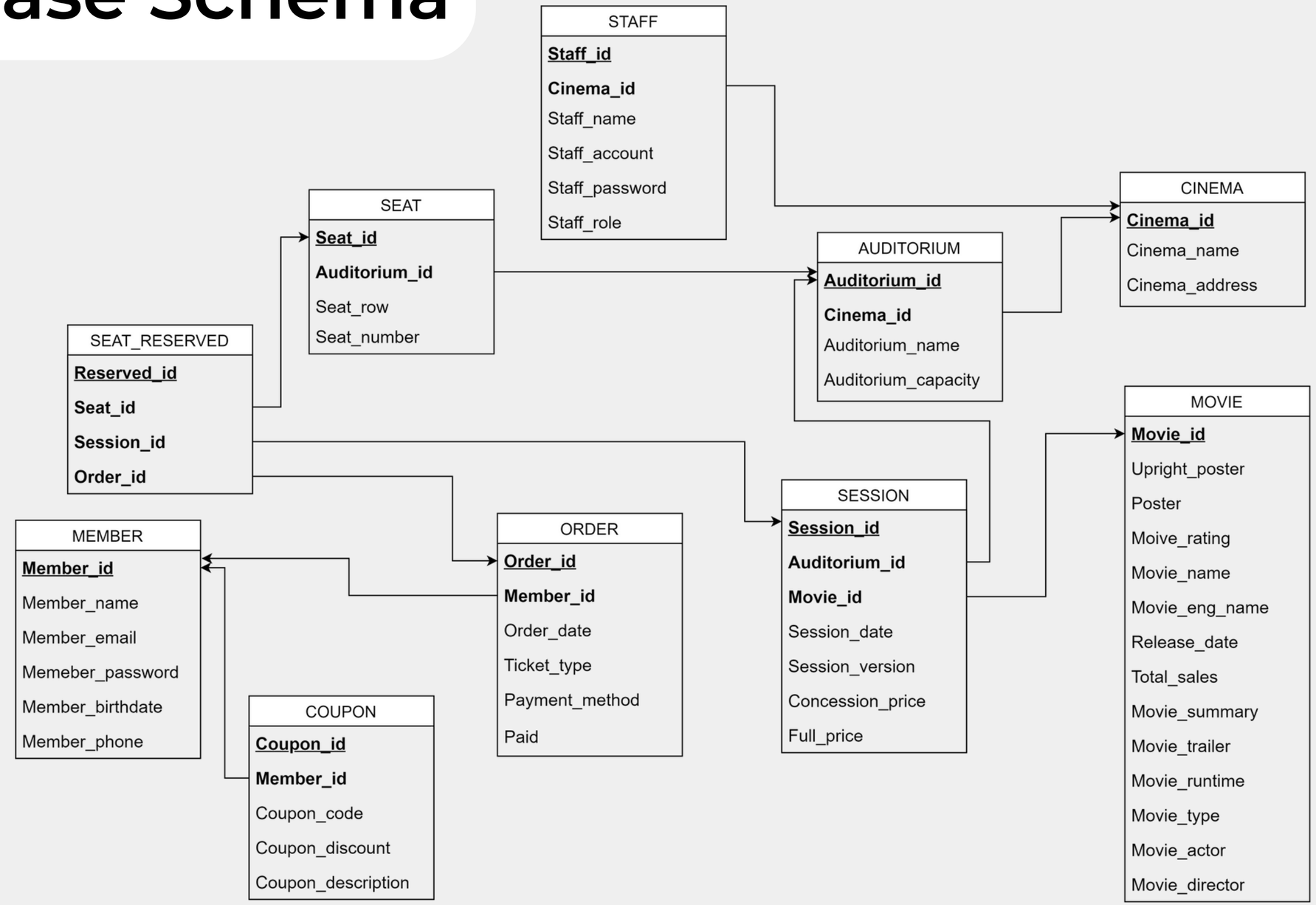
API

Database

ER Diagram

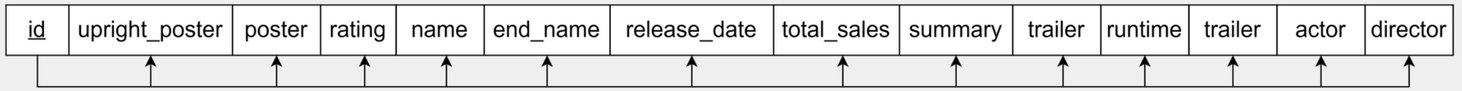

Cinema_name
Cinema_id
Cinema_address
CINEMA
Staff_account
Staff_name
Staff_password
Staff_id
STAFF
Staff_role
HAS
HAS
1
N
1
N
Memeber_password
Member_birthdate
Member_name
Member_phone
Member_email
MEMBER
Member_id
Ticket_type
Payment_method
Order_date
Paid
Order_id
ORDER
seat_id
seat_row
seat_number
SEAT
Auditorium_id
Auditorium_name
AUDITORIUM
Auditorium_capacity
PLCAES
1
N
reserved_id
SEAT_RESERVED
HAS
HAS
N
1
N
1
USE
1
N
RECORD
1
N
RECORD
N
1
Screening
1
N
COUPON
Coupon_id
Coupon_code
Coupon_discount
Coupon_description
SESSION
Session_id
Session_date
Session_time
Session_price
Full_price
Concession_price
Session_version
PLAY
N
1
Movie_id
Total_sales
Moive_rating
Movie_summary
Upright_poster
Movie_trailer
MOVIE
Movie_director
Movie_Info
Movie_actor
Movie_type
Movie_runtime
Release_date
Movie_eng_name
Movie_name
Poster

Database Schema

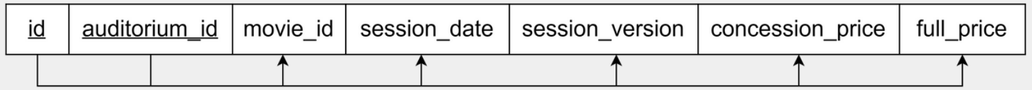


Function Dependency

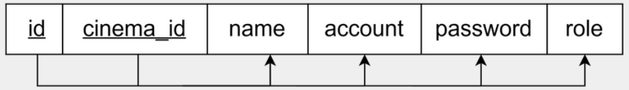
movie



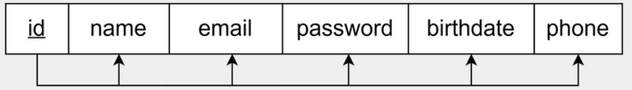
session



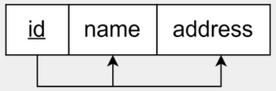
staff



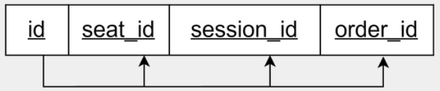
member



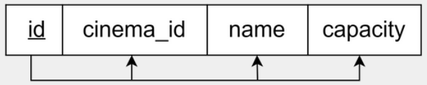
cinema



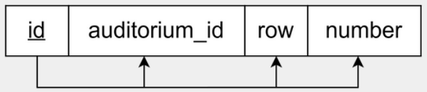
seat_reserved



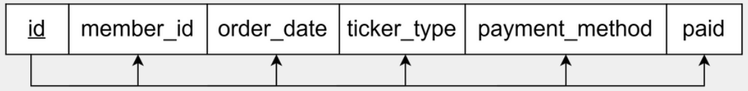
auditorium



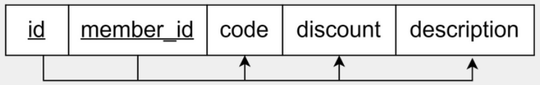
seat



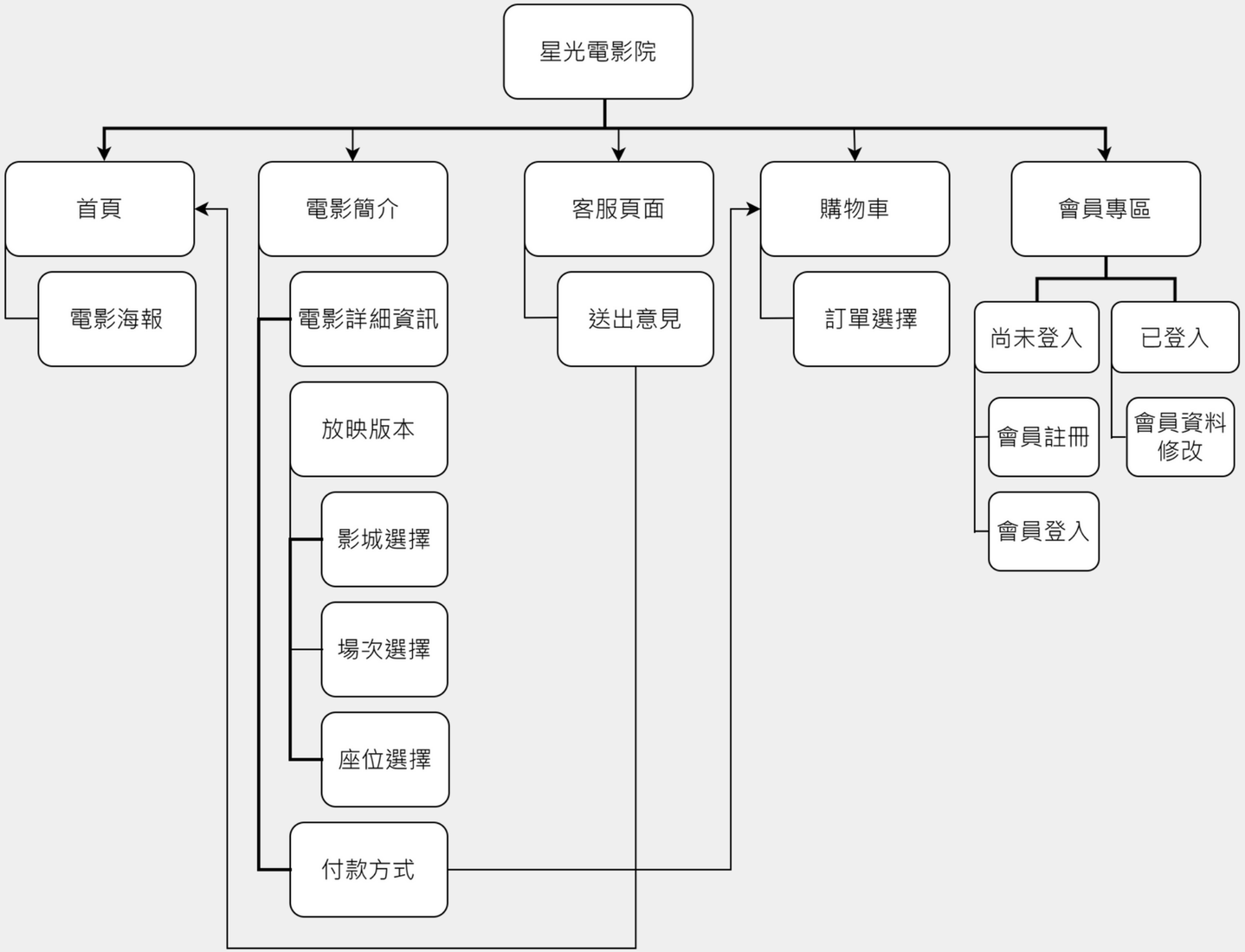
order



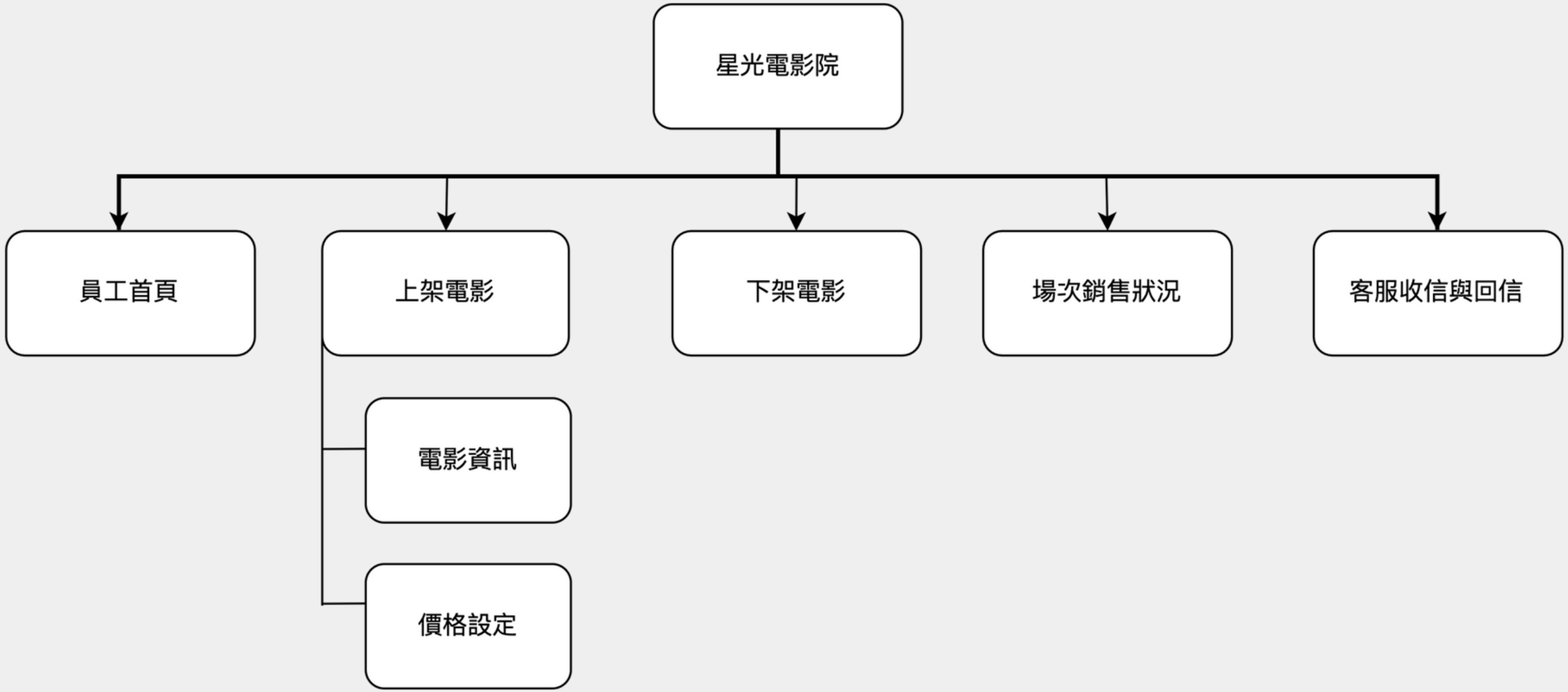
coupon_A



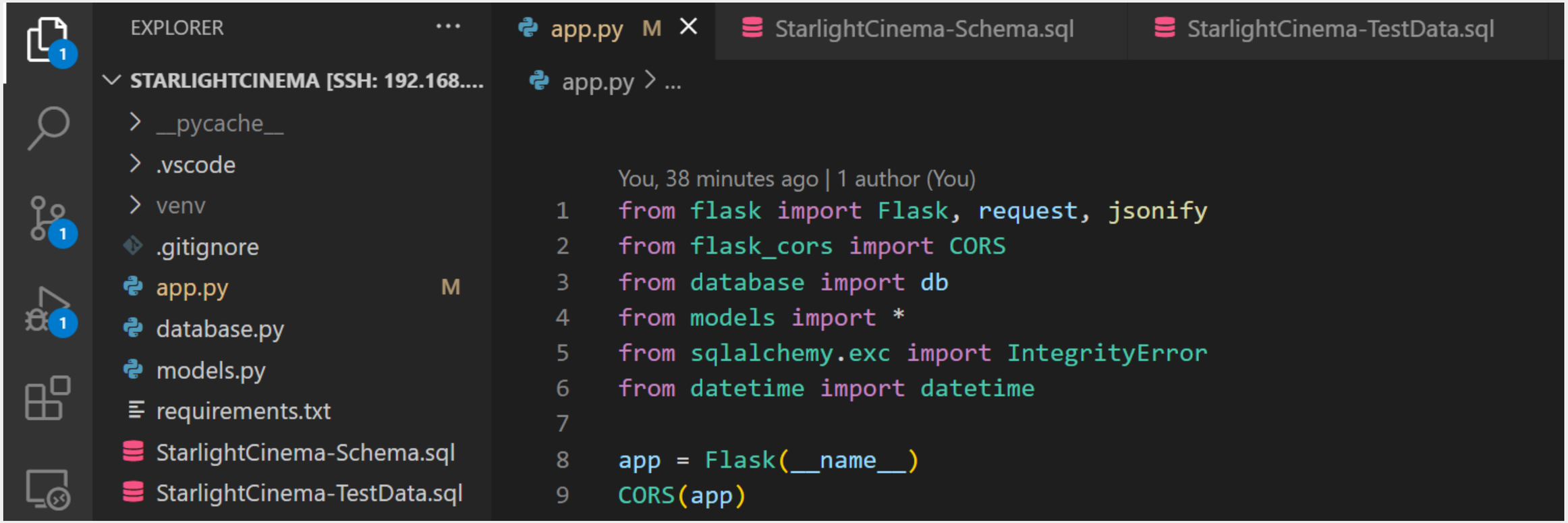
Front-End Architecture - member



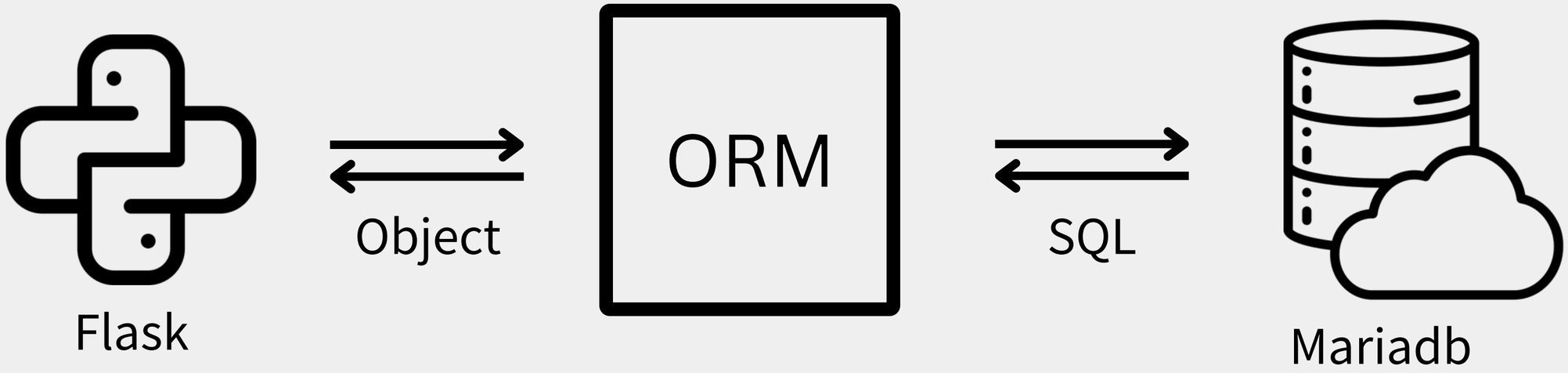
Front-End Architecture - staff



Backend Architecture



SQLAlchemy



- 抽象化和數據庫獨立性： 提供了一個高層次的抽象化,而不需要編寫特定數據庫的 SQL 語句
- 簡化複雜查詢：其強大的查詢 API 讓複雜的數據庫操作更加直觀和容易實現。強大的查詢
- 數據完整性：自動管理數據庫的完整性約束，提高數據質量和安全性。
- 靈活性和可擴展性：支持 ORM 模式到傳統 SQL 表達式的靈活操作，適應不同的開發需求。

models.py

```python
from database import db
You, 2 weeks ago | 1 author (You)
class Cinema(db.Model):
    __tablename__ = 'cinema'
    cinema_id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(255), unique=True, nullable=False)
    address = db.Column(db.String(255), nullable=False)
        You, 2 weeks ago • feature: add bind 0.0.0.0
You, 11 hours ago | 1 author (You)
class Auditorium(db.Model):
    __tablename__ = 'auditorium'
    auditorium_id = db.Column(db.Integer, primary_key=True)
    cinema_id = db.Column(db.Integer, db.ForeignKey('cinema.cinema_id'), nullable=False)
    name = db.Column(db.String(255), nullable=False)
    capacity = db.Column(db.Integer, nullable=False)
    version = db.Column(db.String(255), nullable=False)

You, 2 weeks ago | 1 author (You)
class Movie(db.Model):
    __tablename__ = 'movie'
    movie_id = db.Column(db.Integer, primary_key=True)
    upright_poster = db.Column(db.String(2083), nullable=False)
    poster = db.Column(db.String(2083), nullable=False)
    rating = db.Column(db.SmallInteger, nullable=False)
    name = db.Column(db.String(255), unique=True, nullable=False)
    eng_name = db.Column(db.String(255), unique=True, nullable=False)
    release_date = db.Column(db.Date, nullable=False)
    total_sales = db.Column(db.Integer, nullable=False)
    summary = db.Column(db.Text, nullable=False)
    trailer = db.Column(db.String(2083), nullable=False)
    runtime = db.Column(db.Time, nullable=False)
    type = db.Column(db.String(255), nullable=False)
    actor = db.Column(db.String(255), nullable=False)
    director = db.Column(db.String(255), nullable=False)
```

StarlightCinema-Schema.sql

You, 2 weeks ago | 1 author (You)

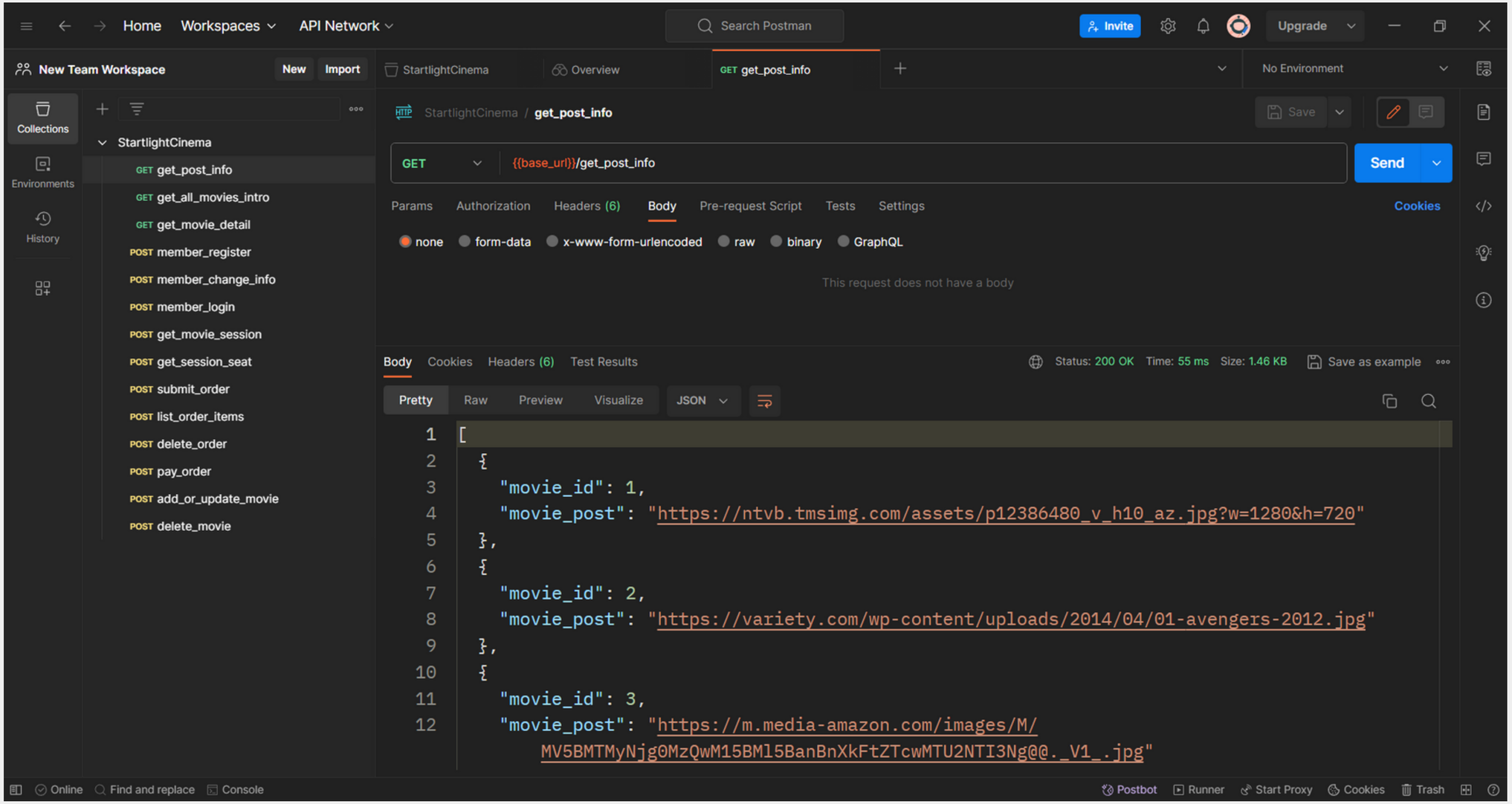
```sql
DROP DATABASE IF EXISTS StarlightCinema;
CREATE DATABASE IF NOT EXISTS StarlightCinema;
USE StarlightCinema;

CREATE TABLE `cinema` (
  `cinema_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `address` VARCHAR(255) NOT NULL,
  CONSTRAINT `pk_cinema` PRIMARY KEY (`cinema_id`),
  CONSTRAINT `uk_cinema_name` UNIQUE (`name`)
);

CREATE TABLE `auditorium` (
  `auditorium_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `cinema_id` INT UNSIGNED NOT NULL,
  `name` VARCHAR(255) NOT NULL,
  `capacity` INT UNSIGNED NOT NULL,
  `version` VARCHAR(255) NOT NULL,
  CONSTRAINT `pk_auditorium` PRIMARY KEY (`auditorium_id`),
  CONSTRAINT `fk_auditorium_cinema` FOREIGN KEY (`cinema_id`) REFERENCES `cinema` (`cinema_id`) ON UPD
);

CREATE TABLE `movie` (
  `movie_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `upright_poster` VARCHAR(2083) NOT NULL,
  `poster` VARCHAR(2083) NOT NULL,
  `rating` TINYINT UNSIGNED NOT NULL,
  `name` VARCHAR(255) NOT NULL,
  `eng_name` VARCHAR(255) NOT NULL,
  `release_date` DATE NOT NULL,
  `total_sales` INT UNSIGNED NOT NULL,
  `summary` TEXT NOT NULL,
  `trailer` VARCHAR(2083) NOT NULL,
  `runtime` TIME NOT NULL,
  `type` VARCHAR(255) NOT NULL,
  `actor` VARCHAR(255) NOT NULL,
  `director` VARCHAR(255) NOT NULL,
  CONSTRAINT `pk_movie` PRIMARY KEY (`movie_id`),
  CONSTRAINT `uk_movie_name` UNIQUE (`name`),
  CONSTRAINT `uk_movie_eng_name` UNIQUE (`eng_name`)
);
```

POSTMAN


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
New Team Workspace
New
Import
StartlightCinema
Overview
GET get_post_info
No Environment
Collections
Environments
History
StartlightCinema
GET get_post_info
GET get_all_movies_intro
GET get_movie_detail
POST member_register
POST member_change_info
POST member_login
POST get_movie_session
POST get_session_seat
POST submit_order
POST list_order_items
POST delete_order
POST pay_order
POST add_or_update_movie
POST delete_movie
StartlightCinema / get_post_info
Save
GET
{{base_url}}/get_post_info
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
This request does not have a body
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 55 ms
Size: 1.46 KB
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"movie_id": 1,
"movie_post": "https://ntvb.tmsimg.com/assets/p12386480_v_h10_az.jpg?w=1280&h=720"
},
{
"movie_id": 2,
"movie_post": "https://variety.com/wp-content/uploads/2014/04/01-avengers-2012.jpg"
},
{
"movie_id": 3,
"movie_post": "https://m.media-amazon.com/images/M/
MV5BMTMyNjg0MzQwM15BMl5BanBnXkFtZTcwMTU2NTI3Ng@@._V1_.jpg"
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash

Demo

https://youtu.be/hvbidkTmBi4

Challenges

- 前後端協調 API 格式
- 整合團隊並分工
- 購物車邏輯耦合
- 座位選擇邏輯
- 多人開發環境整合
- 時間估算過於理想
- 訂單及金流問題
- 多種付款方式整合

THANK YOU!

Q&A