

Movie Booking website

2017313008 김태은

➤ Overview of project

- **Topic**

- Movie Booking & Evaluation Website

- **Objective**

convenient and simple movie booking & evaluation web site

- users can book movie & cancel reservation
- users can evaluate movie (score, famous quote, review...)
- users can search movie
 - > watch movie review & schedule & information
- users can watch user information
 - (movie reservation details & membership information)

- **Functionality**

- Function 1: Join, Sign in, Sign out (회원가입, 로그인, 로그아웃)

사용자는 사이트의 회원일 경우에만 영화 예매, 검색, 평가 등을 할 수 있으므로, 먼저 회원가입을 해야 한다. 회원가입시에는 아이디, 이름, 비밀번호, 이메일 정보를 입력해야 하고, 이것은 USER_INFO DB에 저장된다. 로그인은 회원가입시 DB에 저장된 정보에서 해당하는 정보(ID, 비밀번호)를 찾을 수 있다면 성공적으로 로그인되는 것이다. 마지막으로 로그아웃은 현재 사용중인 회원에서 벗어나 새로운 회원이 접속할 수 있도록 한다.

- Function 2: Show User information (사용자 정보 보기)

- ✓ movie reservation details(영화 예매 정보 확인)
- ✓ membership information (멤버십 정보)

회원은 웹사이트에 저장된 자신의 정보를 확인할 수 있다. 이 정보에는 영화 예매 정보, 멤버십 정보가 있다. 회원은 자신의 정보 보기 창에서 둘 중 하나를 선택할 수 있다. 영화 예매 정보에서는 예매번호, 영화제목, 장르, 감독, 극장, 좌석, 상영시간을 확인할 수 있고, 예매 취소도 가능하다. 멤버십 정보에서는 회원번호, 멤버십 포인트, 가입날짜 등을 확인할 수 있다.

- Function 3: Search Movie (영화 검색)

회원은 현재 상영중인 영화를 검색하여 영화 정보(영화 포스터, 감독, 장르, 상영 시간 정보, 영화 리뷰)를 볼 수 있다. 영화 리뷰의 경우 이 회원 뿐만 아니라 다른 회원들이 등록한 것들도 모두 볼 수 있다.

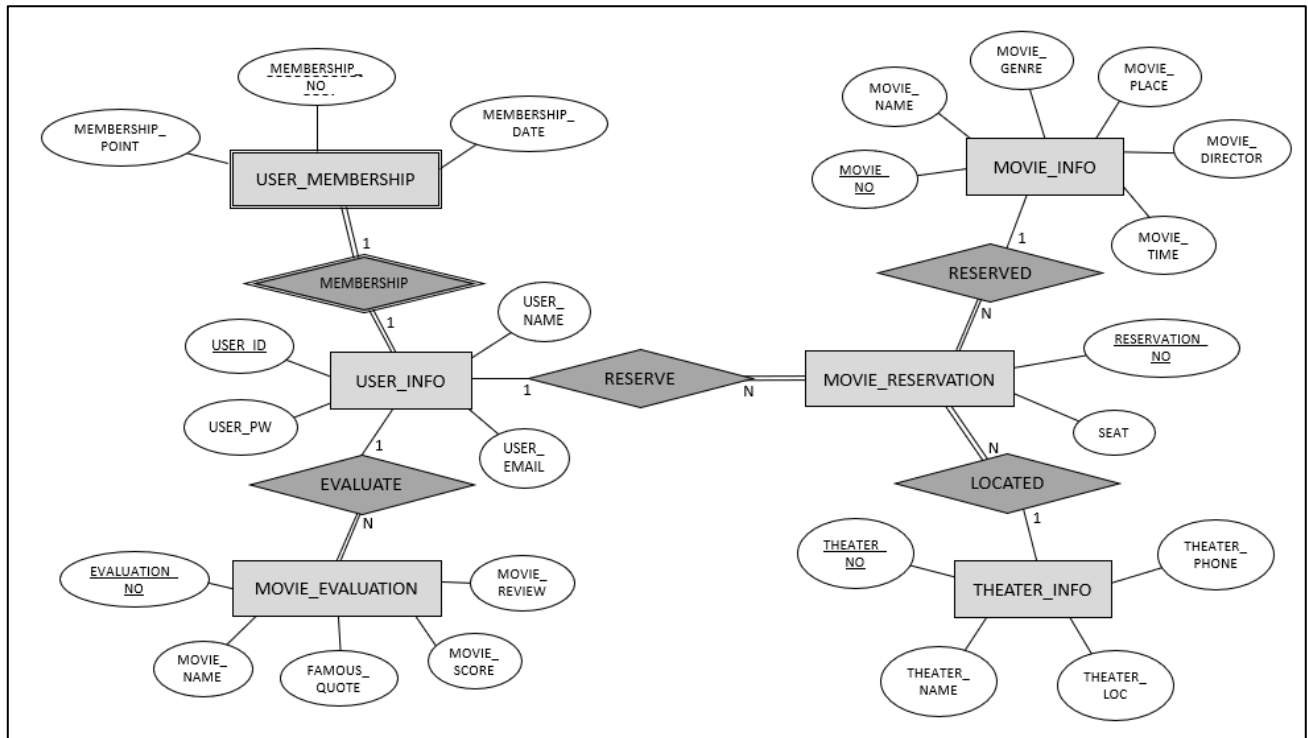
- Function 4: Book Movie & Cancel Reservation (영화 예매, 예매 취소)

회원은 극장, 영화, 시간, 좌석 등을 선택하여 영화를 예매할 수 있다. 영화 예매 취소의 경우, 사용자 영화 예매 정보 보기 창에서 예매 취소 버튼을 눌러서 할 수 있다.

- Function 5: Evaluate Movie (영화 평가)

회원은 현재 상영중인 영화들 중 선택하여 명대사, 평점, 리뷰를 입력하여 영화를 평가할 수 있다. 이러한 평가 정보는 MOVIE_EVALUATION DB에 저장되어 회원들이 영화 검색 시 다른 사람의 리뷰도 볼 수 있도록 했다.

➤ ER diagram



- 6개의 Entity type (USER_INFO, MOVIE_RESERVATION, USER_MEMBERSHIP, MOVIE_EVALUATION, MOVIE_INFO, THEATER_INFO)
- 5개의 Relationship type (MEMBERSHIP, EVALUATE, RESERVE, RESERVED, LOCATED)
- USER_MEMBERSHIP entity type은 weak entity type로서 USER_INFO entity type (owner entity type)에 종속적입니다. USER와 USER_MEMBERSHIP은 1:1 대응 관계이고, 또 서로가 반드시 존재해야 하므로 두 entity type 모두 MEMBERSHIP relationship type에 total participation을 가지고 있습니다.
- 하나의 USER는 여러 개의 MOVIE를 EVALUATE할 수 있기 때문에 USER_INFO entity type과 MOVIE_EVALUATION entity type사이에는 one to many(1:N)의 relationship type이 존재합니다. 또한 USER는 MOVIE를 EVALUATE 하지 않을 수 있지만, MOVIE_EVALUATION이 존재한다는 것은 그 EVALUATION을 한 USER가 있다는 뜻이므로 USER_INFO entity type과 MOVIE_EVALUATION entity type은 EVALUATE relationship type에 각각 partial, total participation을 가지고 있습니다.

- 하나의 USER는 MOVIE를 여러 번 RESERVE할 수 있기 때문에 USER_INFO entity type과 MOVIE_RESERVATION entity type 사이에는 one to many(1:N)의 relationship type이 존재합니다. 또한 USER는 MOVIE를 RESERVE 하지 않을 수 있지만, MOVIE_RESERVATION이 존재한다는 것은 그 RESERVATION을 한 USER가 있다는 뜻이므로 USER entity type과 MOVIE_RESERVATION entity type은 RESERVE relationship type에 각각 partial, total participation을 가지고 있습니다.
- 하나의 MOVIE(같은 영화제목이라도 다른 시간이면 다르게 취급)는 여러 번 RESERVED될 수 있기 때문에 MOVIE_INFO entity type과 MOVIE_RESERVATION entity type 사이에는 one to many(1:N)의 relationship type이 존재합니다. 또한 MOVIE는 RESERVED되지 않을 수 있지만, MOVIE_RESERVATION가 존재한다는 것은 RESERVED된 MOVIE가 있다는 뜻이므로 MOVIE_INFO entity type과 MOVIE_RESERVATION entity type은 RESERVED relationship에 각각 partial, total participation을 가지고 있습니다.
- 하나의 THEATER는 여러 번 LOCATED될 수 있기 때문에 THEATER_INFO entity type과 MOVIE_RESERVATION entity type 사이에는 one to many(1:N)의 relationship type이 존재합니다. 또한 THEATER는 LOCATED되지 않을 수 있지만, MOVIE_RESERVATION이 존재한다는 것은 LOCATED된 THEATER가 있다는 뜻이므로 THEATER_INFO entity type과 MOVIE_RESERVATION entity type은 LOCATED relationship에 각각 partial, total participation을 가지고 있습니다.
- MOVIE_INFO의 primary key는 MOVIE_NAME이 아닌 MOVIE_NO입니다. 같은 영화라도 시간에 따라 각각 따로 INSERT해주었기 때문에 이들은 모두 다른 MOVIE_NO를 가집니다. 따라서, MOVIE_INFO DB에는 같은 영화 제목을 가진 영화가 여러 번 등장할 수 있습니다. 이러한 이유로 MOVIE_EVALUATION에서 USER_ID는(누가 평가했는지에 대한 정보)는 USER_INFO의 primary key인 USER_ID를 foreign key로 참조하지만, MOVIE_NAME은 MOVIE_INFO를 참조하지 않습니다. (MOVIE_INFO에서 같은 제목의 영화라도 시간에 따라 다른 영화로 취급하기 때문에 MOVIE_NAME이 primary key가 아님)

- **USER_INFO entity type: 사이트 사용자의 정보를 저장 (회원가입 시 입력되는 정보들)**

- **USER_ID:** 사용자의 고유한 ID를 나타내며, primary key attribute입니다. VARCHAR(20) 데이터타입을 사용합니다. 사용자의 ID는 사용자를 식별하는 역할을 하므로 중복을 허용하지 않습니다.
- **USER_PW:** 사용자가 회원 가입 시 입력하는 비밀번호가 저장되며, VARCHAR(20) 데이터타입을 사용합니다. 사용자가 로그인 시에 입력한 ID와 PW를 바탕으로 올바른 입력인지 아닌지 체크합니다.
- **USER_NAME:** 사용자의 이름을 나타내며, VARCHAR(10) 데이터타입을 사용합니다. 회원가입 시 사용자가 입력하는 이름이 저장됩니다.
- **USER_EMAIL:** 사용자의 이메일을 나타내며, VARCHAR(30) 데이터타입을 사용합니다. 회원가입 시 사용자가 입력하는 이메일이 저장됩니다.

- **USER_MEMBERSHIP entity type: 사용자의 멤버십과 관련된 정보를 저장**

- **MEMBERSHIP_NO:** 각 멤버십 사용자의 고유한 NUMBER를 나타내며, partial key attribute입니다. INT(5) 데이터타입을 사용합니다.
- **MEMBERSHIP_POINT:** 각 멤버십 사용자가 현재 보유하고 있는 멤버십 포인트를 나타내며, INT(5) 데이터타입을 사용합니다. 영화를 예매할 때마다 포인트가 축적 (+100)되고, 예매를 취소할 때마다 포인트가 (-30)차감됩니다.
- **MEMBERSHIP_DATE:** 각 멤버십 사용자가 가입한 날짜이며, DATE 데이터타입을 사용합니다. 회원가입 시 멤버십이 저절로 생성되며 가입한 날짜도 저장됩니다.

- **MOVIE_EVALUATION entity type: 영화에 대한 평가 정보를 저장**

- **EVALUATION_NO:** 각 평가정보의 고유한 NUMBER를 나타내며, primary key attribute입니다. INT(5) 데이터타입을 사용합니다.
- **USER_ID:** 영화를 평가한 회원이 누구인지를 나타내며, USER_INFO의 primary key인 USER_INFO를 foreign key로 참조합니다. VARCHAR(20) 데이터타입을 사용합니다.
- **MOVIE_NAME:** 평가하는 영화의 제목을 나타내며, VARCHAR(30) 데이터타입을 사용합니다. 영화 평가 시 상영하는 영화 목록들 중 사용자가 선택하는 영화 제목이 저장됩니다.
- **FAMOUS_QUOTE:** 영화에서 인상깊은 구절 혹은 명대사를 나타내며, VARCHAR(100) 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 내용이 저장됩니다.

- **MOVIE_SCORE**: 영화에 대한 전체적인 평점을 나타내며, **DOUBLE** 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 평점이 저장됩니다.
- **MOVIE_REVIEW**: 영화에 대한 리뷰 혹은 비평을 나타내며, **VARCHAR(1000)** 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 내용이 저장됩니다.

● **MOVIE_RESERVATION entity type: 영화 예매 정보를 저장**

- **RESERVATION_NO**: 각 예매정보의 고유한 **NUMBER**를 나타내며, **primary key attribute**입니다. **INT(5)** 데이터타입을 사용합니다.
- **USER_ID**: 영화를 예매한 회원이 누구인지를 나타내며, **USER_INFO**의 **primary key**인 **USER_INFO**를 **foreign key**로 참조합니다. **VARCHAR(20)** 데이터타입을 사용합니다.
- **MOVIE_NO**: 어떤 영화가 예매되었는지를 나타내며, **MOVIE_INFO**의 **primary key**인 **MOVIE_NO**를 **foreign key**로 참조합니다. **INT(5)** 데이터타입을 사용합니다.
- **THEATER_NO**: 예매 시에 어떤 극장을 선택했는지를 나타내며, **THEATER_INFO**의 **primary key**인 **THEATER_NO**를 **foreign key**로 참조합니다. **INT(5)** 데이터타입을 사용합니다.
- **SEAT**: 영화관 좌석의 위치를 나타내며, 구역과 자리번호가 합쳐져서 저장됩니다. **VARCHAR(10)**을 사용합니다.

● **MOVIE_INFO entity type: 영화에 대한 정보를 저장(같은 영화 제목이라도 상영 시간이 다르면 각각을 따로 저장)**

초기 **DB**에 **insert**해주는 데이터입니다.

- **MOVIE_NO**: 각 영화정보의 고유한 **NUMBER**를 나타내며, **primary key attribute**입니다. **INT(5)** 데이터타입을 사용합니다.
- **MOVIE_NAME**: 영화 제목을 나타내며, **VARCHAR(50)** 데이터타입을 사용합니다.
- **MOVIE_GENRE**: 영화 장르를 나타내며, **VARCHAR(50)** 데이터타입을 사용합니다.
- **MOVIE_DIRECTOR**: 영화의 감독을 나타내며, **VARCHAR(20)** 데이터타입을 사용합니다. 하나의 영화에는 한 명의 감독이 있다고 가정합니다.
- **MOVIE_PLACE**: 각 영화가 어떤 관에서 상영되는지를 나타내며(ex: '1관', '2관',....), 같은 영화 제목이면 같은 관에서 상영되는 것으로 가정합니다. **VARCHAR(5)** 데이터타입을 사용합니다.

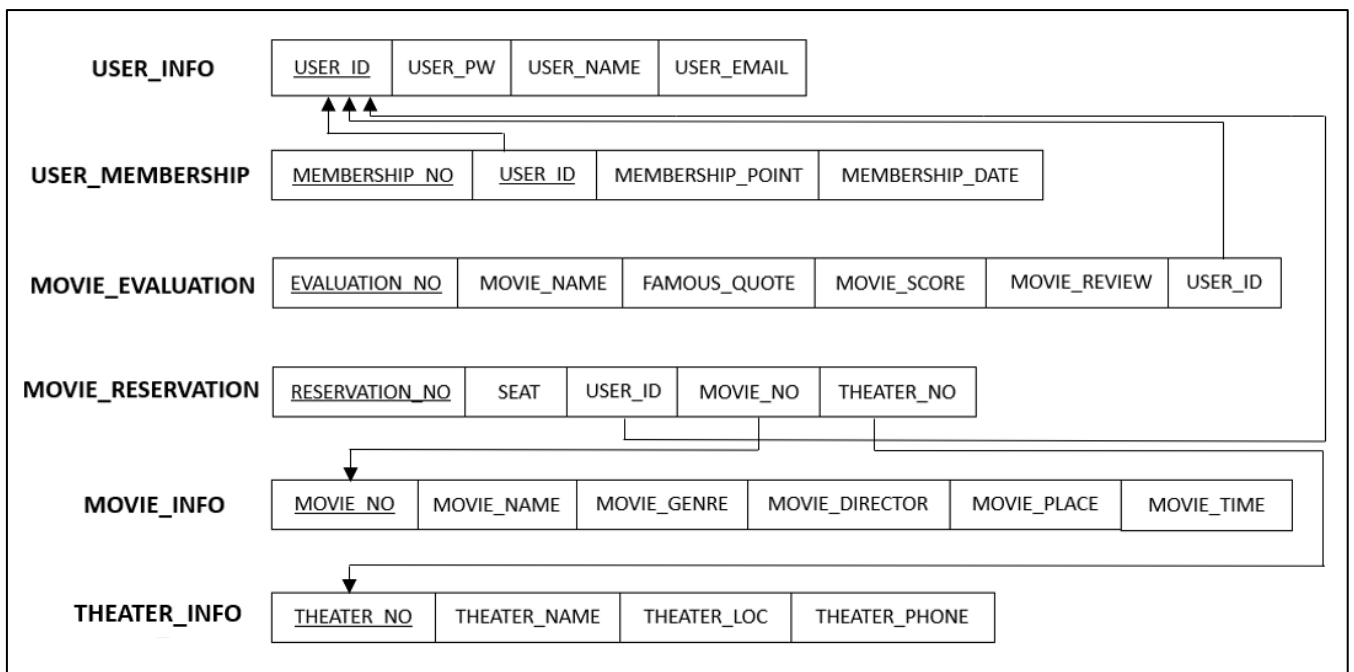
- MOVIE_TIME: 영화의 상영 시간을 나타내며, VARCHAR(20) 데이터타입을 사용합니다. 총 4가지 상영 시간이 있다고 가정합니다. 1) 08:00~11:00 / 2) 12:00~15:00 / 3) 16:00~19:00 / 4) 20:00 ~ 23:00

● THEATER_INFO entity type: 극장에 대한 정보를 저장

초기 DB에 insert해주는 데이터입니다.

- THEATER_NO: 각 극장정보의 고유한 NUMBER를 나타내며, primary key attribute입니다. INT(5) 데이터타입을 사용합니다.
- THEATER_NAME: 극장 이름을 나타내며, VARCHAR(20) 데이터타입을 사용합니다.
- THEATER_LOC: 극장 위치를 나타내며, VARCHAR(50) 데이터타입을 사용합니다.
- THEATER_PHONE: 극장 전화번호를 나타내며, VARCHAR(20) 데이터타입을 사용합니다.

➤ Relational Model



6개의 relation (USER_INFO, USER_MEMBERSHIP, MOVIE_EVALUATION, MOVIE_RESERVATION, MOVIE_INFO, THEATER_INFO)

- **USER table: 사용자 정보가 저장되는 table입니다. 회원가입 시 입력되는 정보가 저장되며 로그인 기능, 사용자 멤버십 정보 보기 기능에 이용됩니다. (PK: USER_ID)**

- **USER_ID:** 사용자의 고유한 ID를 나타내며, USER table의 primary key입니다. VARCHAR(20) 데이터타입을 사용합니다. 사용자의 ID는 사용자를 식별하는 역할을 하므로 중복을 허용하지 않습니다.
- **USER_PW:** 사용자가 회원 가입 시 입력하는 비밀번호가 저장되며, VARCHAR(20) 데이터타입을 사용합니다. 사용자가 로그인 시에 입력한 ID와 PW를 바탕으로 올바른 입력인지 아닌지 체크합니다.
- **USER_NAME:** 사용자의 이름을 나타내며, VARCHAR(10) 데이터타입을 사용합니다. 회원가입 시 사용자가 입력하는 이름이 저장됩니다.
- **USER_EMAIL:** 사용자의 이메일을 나타내며, VARCHAR(30) 데이터타입을 사용합니다. 회원가입 시 사용자가 입력하는 이메일이 저장됩니다.

- **USER_MEMBERSHIP table: 사용자 멤버십 정보가 저장되는 table입니다. 멤버십은 회원가입 시 자동 생성되며, 사용자 멤버십 정보 보기 기능에 이용됩니다. (PK: MEMBERSHIP_NO, USER_ID, FK: USER_ID)**

Weak entity type로서 USER (Owner entity type)에 종속적입니다. USER_INFO table의 USER_ID를 foreign key로 참조합니다. USER_ID는 foreign key이자 primary key입니다. MEMBERSHIP_NO는 USER_MEMBERSHIP entity type의 partial key로, USER_ID와 함께 USER_MEMBERSHIP table의 primary key 역할을 합니다.

- **MEMBERSHIP_NO:** 각 멤버십 사용자의 고유한 NUMBER를 나타내며, USER_INFO table의 USER_ID와 함께 USER_MEMBERSHIP table의 primary key 역할을 합니다. INT(5) 데이터타입을 사용합니다.
- **USER_ID:** 각 멤버십 정보가 참조하는 사용자의 고유한 ID를 나타내며, USER_MEMBERSHIP table의 MEMBERSHIP_NO와 함께 USER_MEMBERSHIP table의 primary key 역할을 합니다. VARCHAR(20) 데이터타입을 사용합니다.
- **MEMBERSHIP_POINT:** 각 멤버십 사용자가 현재 보유하고 있는 멤버십 포인트를 나타내며, INT(5) 데이터타입을 사용합니다. 영화를 예매할 때마다 포인트가 축적(+100)되고, 예매를 취소할 때마다 포인트가 (-30)감소됩니다.
- **MEMBERSHIP_DATE:** 각 멤버십 사용자가 가입한 날짜이며, DATE 데이터타입을 사용합니다. 회원가입 시 멤버십이 저절로 생성되며 가입한 날짜도 저장됩니다.

- **MOVIE_EVALUATION table:** 영화에 대한 평가 정보가 저장되는 table입니다. 영화 평가, 검색 기능에 이용됩니다.

(PK: EVALUATION_NO , FK: USER_ID)

- EVALUATION_NO: 각 평가정보의 고유한 NUMBER를 나타내며, primary key로 영화 평가 정보를 식별하는 역할을 한다. INT(5) 데이터타입을 사용합니다.
- USER_ID: 영화를 평가한 회원이 누구인지를 나타내며, USER_INFO의 primary key인 USER_INFO를 foreign key로 참조합니다. VARCHAR(20) 데이터타입을 사용합니다.
- MOVIE_NAME: 평가하는 영화의 제목을 나타내며, VARCHAR(30) 데이터타입을 사용합니다. 영화 평가 시 상영하는 영화 목록들 중 사용자가 선택하는 영화 제목이 저장됩니다.
- FAMOUS_QUOTE: 영화에서 인상깊은 구절 혹은 명대사를 나타내며, VARCHAR(100) 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 내용이 저장됩니다.
- MOVIE_SCORE: 영화에 대한 전체적인 평점을 나타내며, DOUBLE 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 평점이 저장됩니다.
- MOVIE_REVIEW: 영화에 대한 리뷰 혹은 비평을 나타내며, VARCHAR(1000) 데이터타입을 사용합니다. 영화 평가 시 사용자가 입력하는 내용이 저장됩니다.

- **MOVIE_RESERVATION table:** 영화 예매 정보가 저장되는 table입니다. 영화 예매, 예매 정보 보기 기능에 이용됩니다.

(PK: RESERVATION_NO , FK: USER_ID, MOVIE_NO, THEATER_NO)

- RESERVATION_NO: 각 예매정보의 고유한 NUMBER를 나타내며, primary key로 영화 예매 정보를 식별하는 역할입니다. INT(5) 데이터타입을 사용합니다.
- USER_ID: 영화를 예매한 회원이 누구인지를 나타내며, USER_INFO의 primary key인 USER_INFO를 foreign key로 참조합니다. VARCHAR(20) 데이터타입을 사용합니다.
- MOVIE_NO: 어떤 영화가 예매되었는지를 나타내며, MOVIE_INFO의 primary key인 MOVIE_NO를 foreign key로 참조합니다. INT(5) 데이터타입을 사용합니다.
- THEATER_NO: 예매 시에 어떤 극장을 선택했는지를 나타내며, THEATER_INFO의 primary key인 THEATER_NO를 foreign key로 참조합니다. INT(5) 데이터타입을 사용합니다.
- SEAT: 영화관 좌석의 위치를 나타내며, 구역과 자리번호가 합쳐져서 저장됩니다. VARCHAR(10)을 사용합니다.

- **MOVIE_INFO table:** 영화에 대한 정보를 저장(같은 영화 제목이라도 상영 시간이 다르면 각각을 따로 저장)하는 table입니다. 영화 검색, 예매, 평가 기능에 이용됩니다.

초기 DB에 insert해주는 데이터입니다. (PK: MOVIE_NO)

- MOVIE_NO: 각 영화정보의 고유한 NUMBER를 나타내며, primary key입니다. INT(5) 데이터타입을 사용합니다.
- MOVIE_NAME: 영화 제목을 나타내며, VARCHAR(50) 데이터타입을 사용합니다.
- MOVIE_GENRE: 영화 장르를 나타내며, VARCHAR(50) 데이터타입을 사용합니다.
- MOVIE_DIRECTOR: 영화의 감독을 나타내며, VARCHAR(20) 데이터타입을 사용합니다. 하나의 영화에는 한 명의 감독이 있다고 가정합니다.
- MOVIE_PLACE: 각 영화가 어떤 관에서 상영되는지를 나타내며(ex: '1관', '2관',....), 같은 영화 제목이면 같은 관에서 상영되는 것으로 가정합니다. VARCHAR(5) 데이터타입을 사용합니다.
- MOVIE_TIME: 영화의 상영 시간을 나타내며, VARCHAR(20) 데이터타입을 사용합니다. 총 4가지 상영 시간이 있다고 가정합니다. 1) 08:00~11:00 / 2) 12:00~15:00 / 3) 16:00~19:00 / 4) 20:00 ~ 23:00

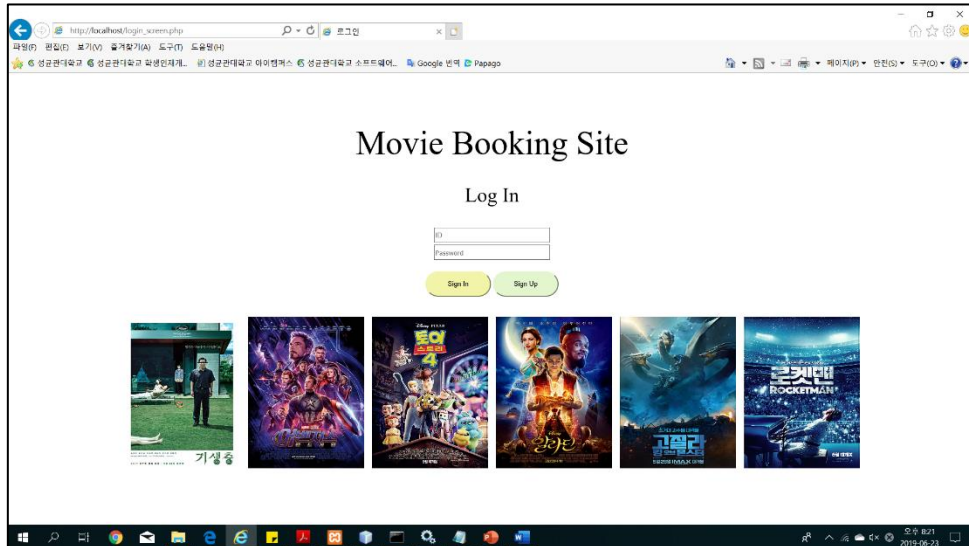
- **THEATER_INFO table:** 극장에 대한 정보를 저장하는 table입니다. 영화 예매, 예매 정보 보기 기능에 이용됩니다.

초기 DB에 insert해주는 데이터입니다. (PK: THEATER_NO)

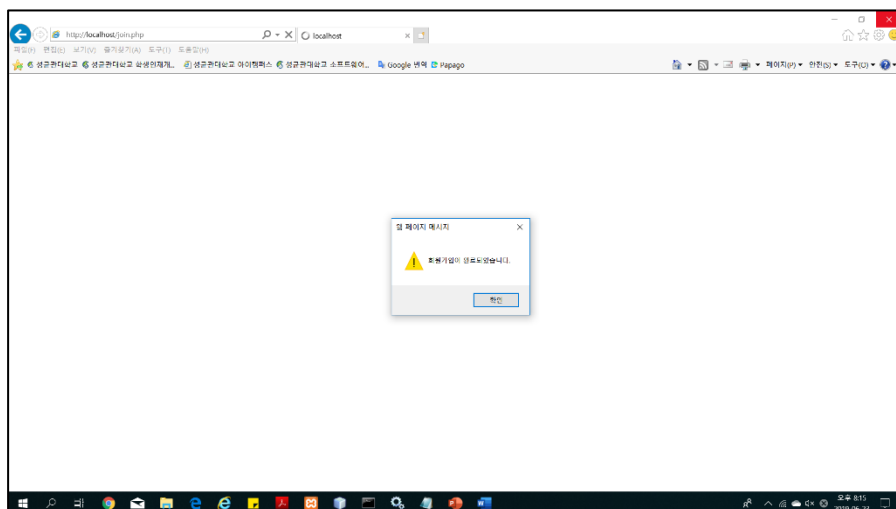
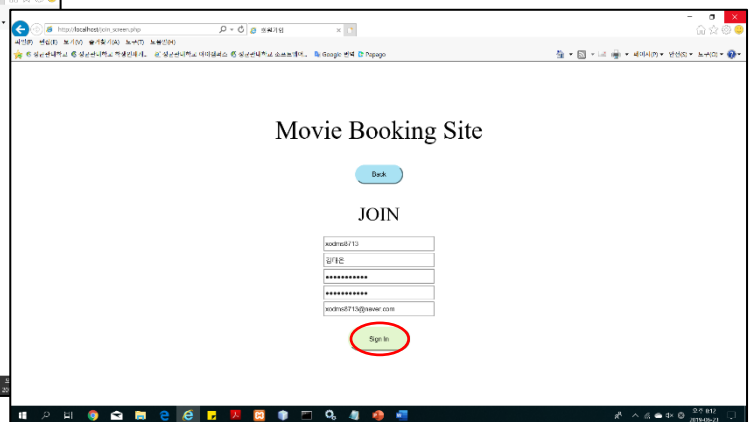
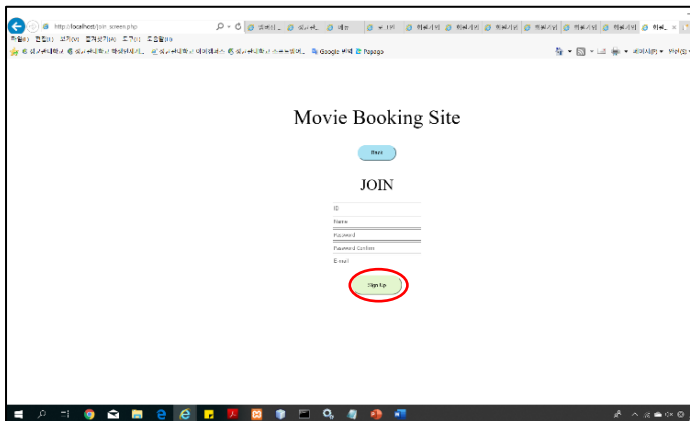
- THEATER_NO: 각 극장정보의 고유한 NUMBER를 나타내며, primary key입니다. INT(5) 데이터타입을 사용합니다.
- THEATER_NAME: 극장 이름을 나타내며, VARCHAR(20) 데이터타입을 사용합니다.
- THEATER_LOC: 극장 위치를 나타내며, VARCHAR(50) 데이터타입을 사용합니다.
- THEATER_PHONE: 극장 전화번호를 나타내며, VARCHAR(20) 데이터타입을 사용합니다.

➤ Implementation and Results

● Function 1: Join, Sign in, Sign out (회원가입, 로그인, 로그아웃)

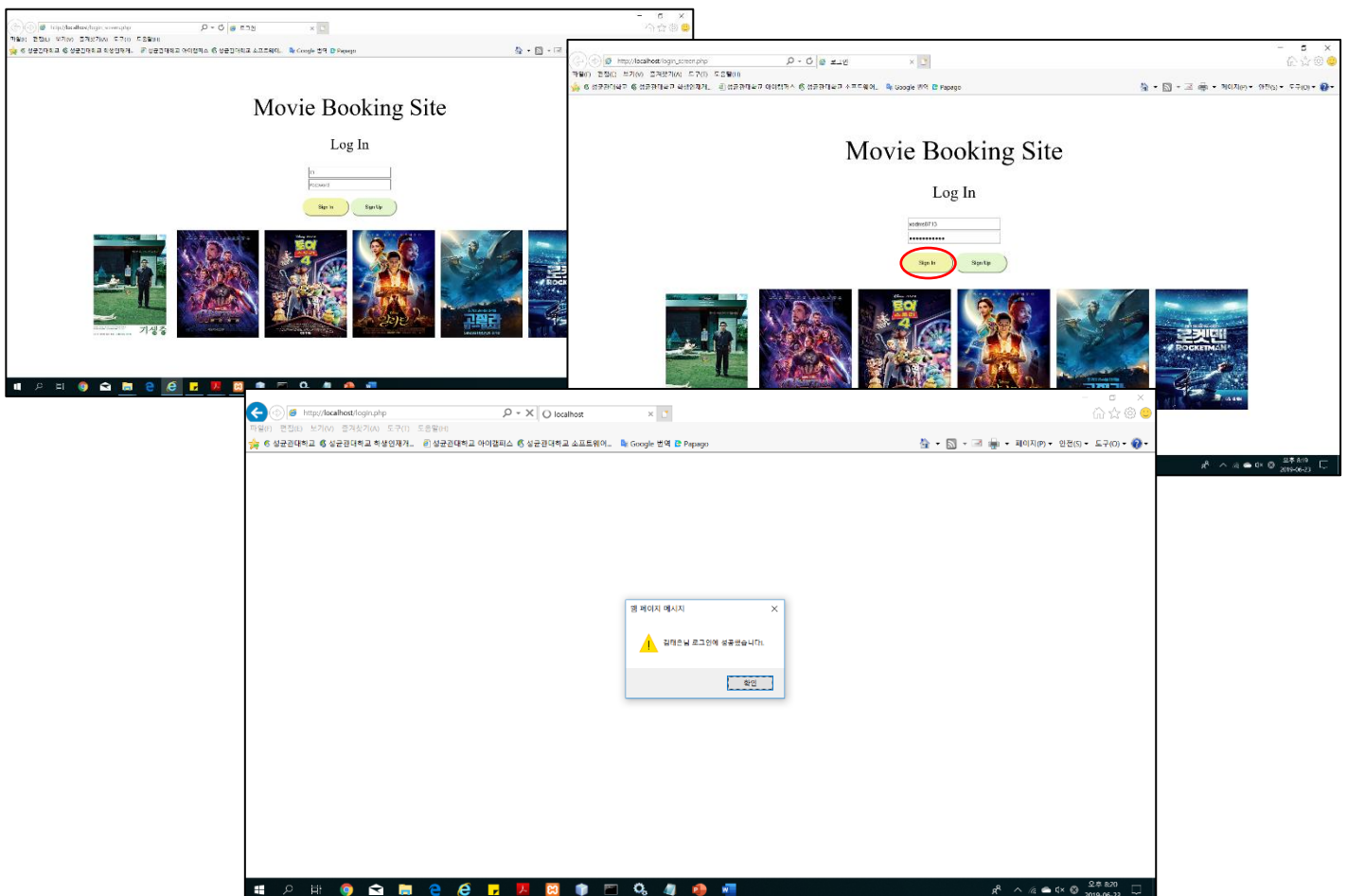


1. Join



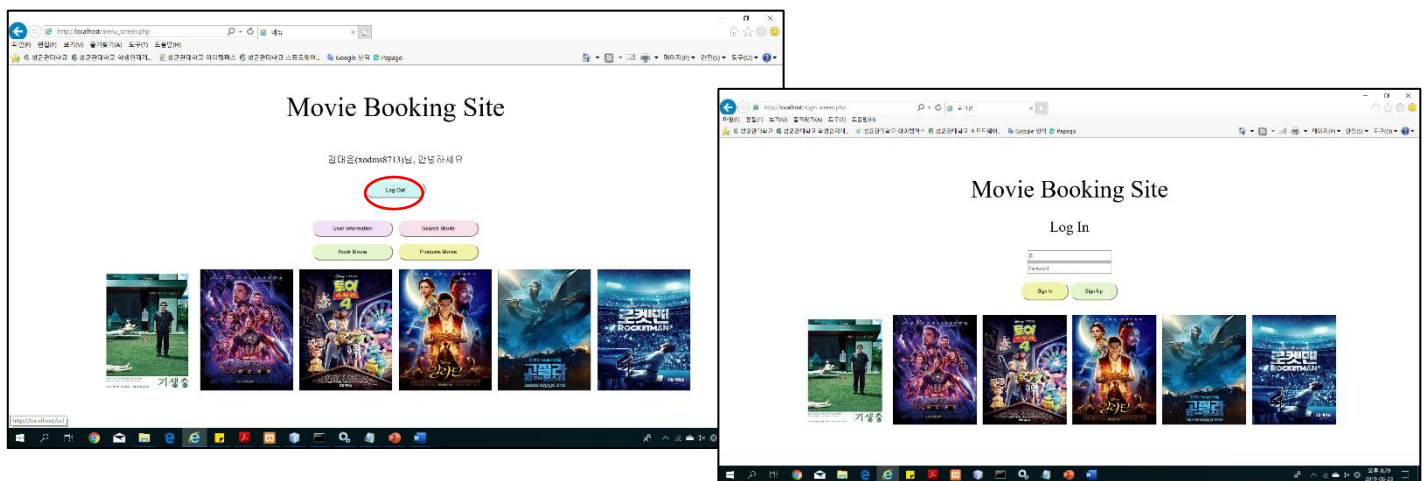
회원가입 기능: join_screen.php, join.php로 구현하였습니다. 가장 처음 화면인 로그인 창에서 sign up을 누르면 join screen으로 이동합니다. join screen에서 user_id, user_name, user_pw_check, user_email을 입력하게 됩니다. 그리고 sign up 버튼을 누르면 enroll 함수가 실행되면서 user_pw와 user_pw_check가 같은지 확인하여 같을 경우에만 join.php으로 넘어가고 다를 경우(if (frm.user_password.value != frm.user_password_check.value)) 새로 입력하도록 했습니다. join.php에서는 mysql과 연동하여 이전 창에서 입력된 정보들을 USER_INFO table에 저장합니다(INSERT INTO USER_INFO(USER_ID, USER_PW, USER_NAME, USER_EMAIL) VALUES ('\$user_id','\$user_password','\$user_name','\$user_email')). 회원가입시에 멤버십도 자동생성 합니다. 따라서 처음 가입된 회원의 경우 멤버십 포인트=0, 가입날짜를 그 시점의 날짜로 USER_MEMBERSHIP table에 저장합니다(INSERT INTO USER_MEMBERSHIP(USER_ID, MEMBERSHIP_POINT, MEMBERSHIP_DATE) VALUES ('\$user_id', 0, '\$date')). 이것까지 완료된 경우, '회원가입이 완료되었습니다.'라는 알림 창과 함께 login screen으로 이동합니다.

2. Sign in



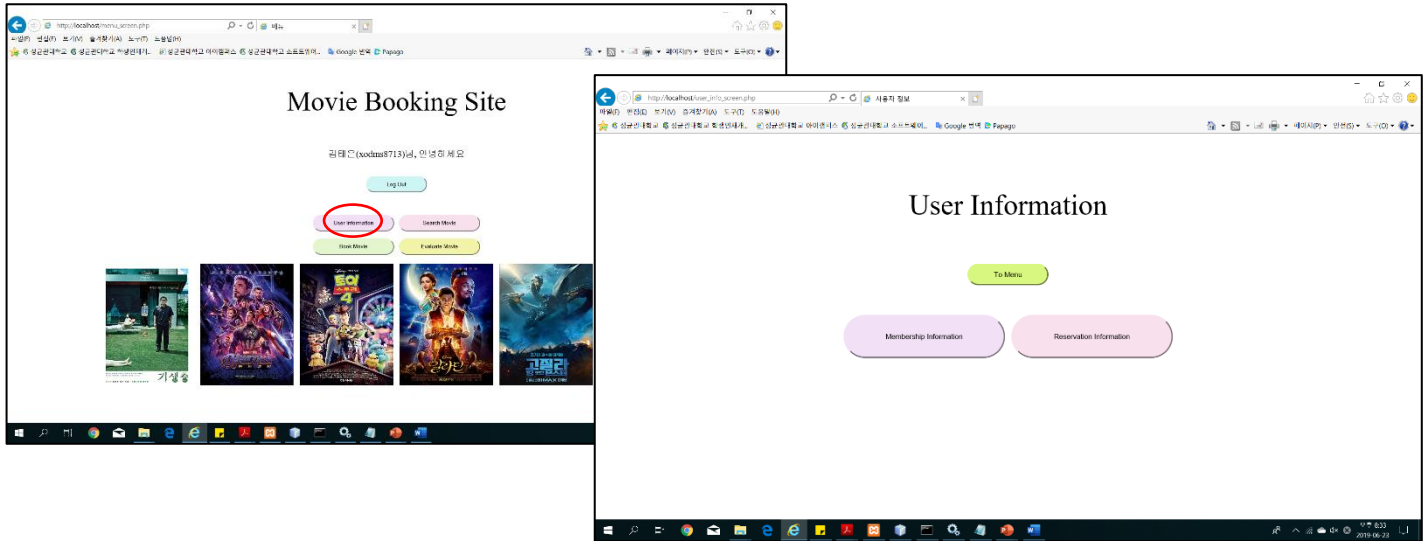
로그인 기능: login_screen.php, login.php으로 구현했습니다. login_screen에서는 사용자 아이디와 패스워드를 입력하는 칸이 있습니다. 아직 회원가입을 하지 않은 회원을 제외하고는 모두 로그인을 하게 됩니다. 아이디와 패스워드 입력 후 sign in 버튼을 누르면 login.php로 해당 정보들과 함께 이동합니다. (input type="submit") mysql을 연동하여 USER_INFO에 해당 아이디와 패스워드를 가진 tuple이 있는지 검사하고(SELECT * FROM USER_INFO WHERE USER_ID='\$user_id' AND USER_PW='\$user_password'), 없을 경우에 '아이디 혹은 비밀번호가 틀렸습니다.'라는 메시지와 함께 다시 login_screen으로 이동합니다. 해당 tuple이 존재하는 경우에는 menu_screen 으로 이동합니다.

3. Sign out



로그아웃 기능: logout.php로 구현했습니다. session_destroy(); http-equiv='refresh' 후 다시 login screen으로 이동하도록 했습니다. 메뉴 창 외에도 다양한 창에 log out 버튼을 만들어서 바로 log out 할 수 있도록 했습니다.

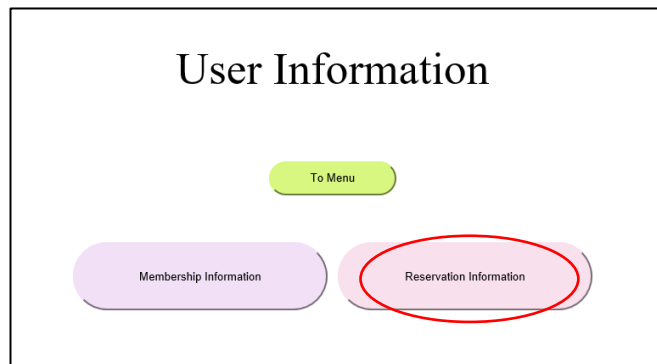
● Function 2: Show User information (사용자 정보 보기)



메뉴 창에서 user information 을 클릭하면 사용자 정보 보기 창으로 이동합니다.

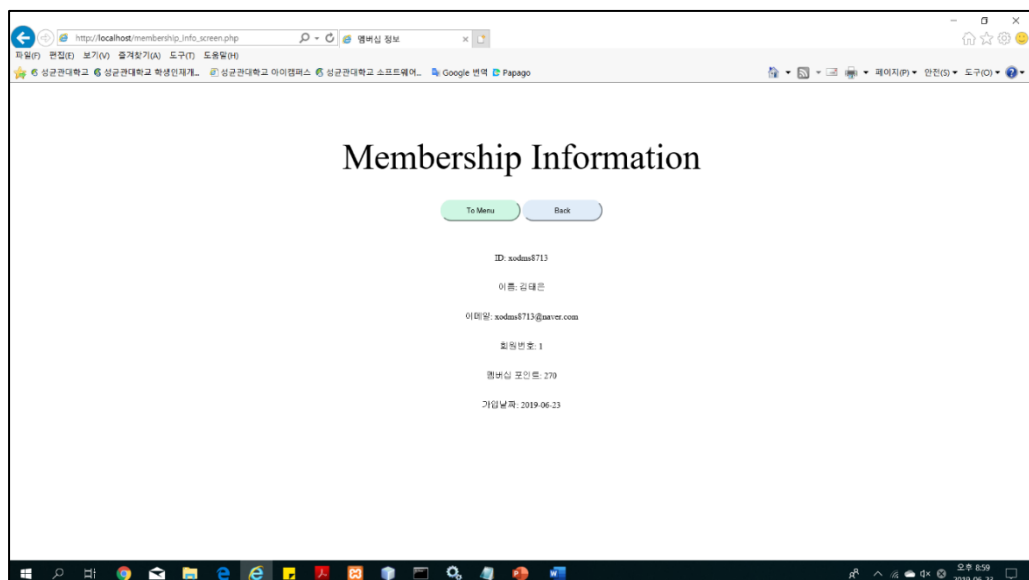
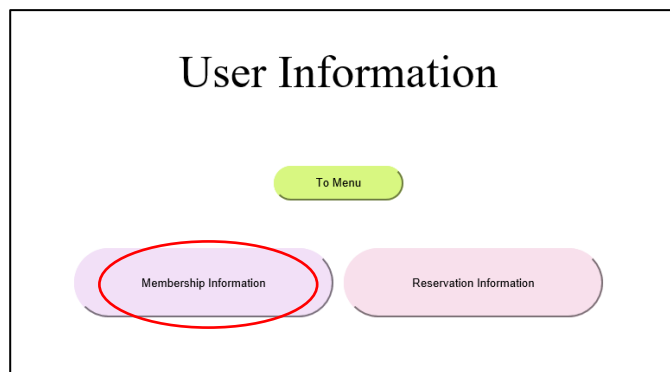
여기에서 영화 예매 정보 보기, 멤버십 정보 보기 둘 중 하나를 고를 수 있습니다.

1. movie reservation details (영화 예매 정보)



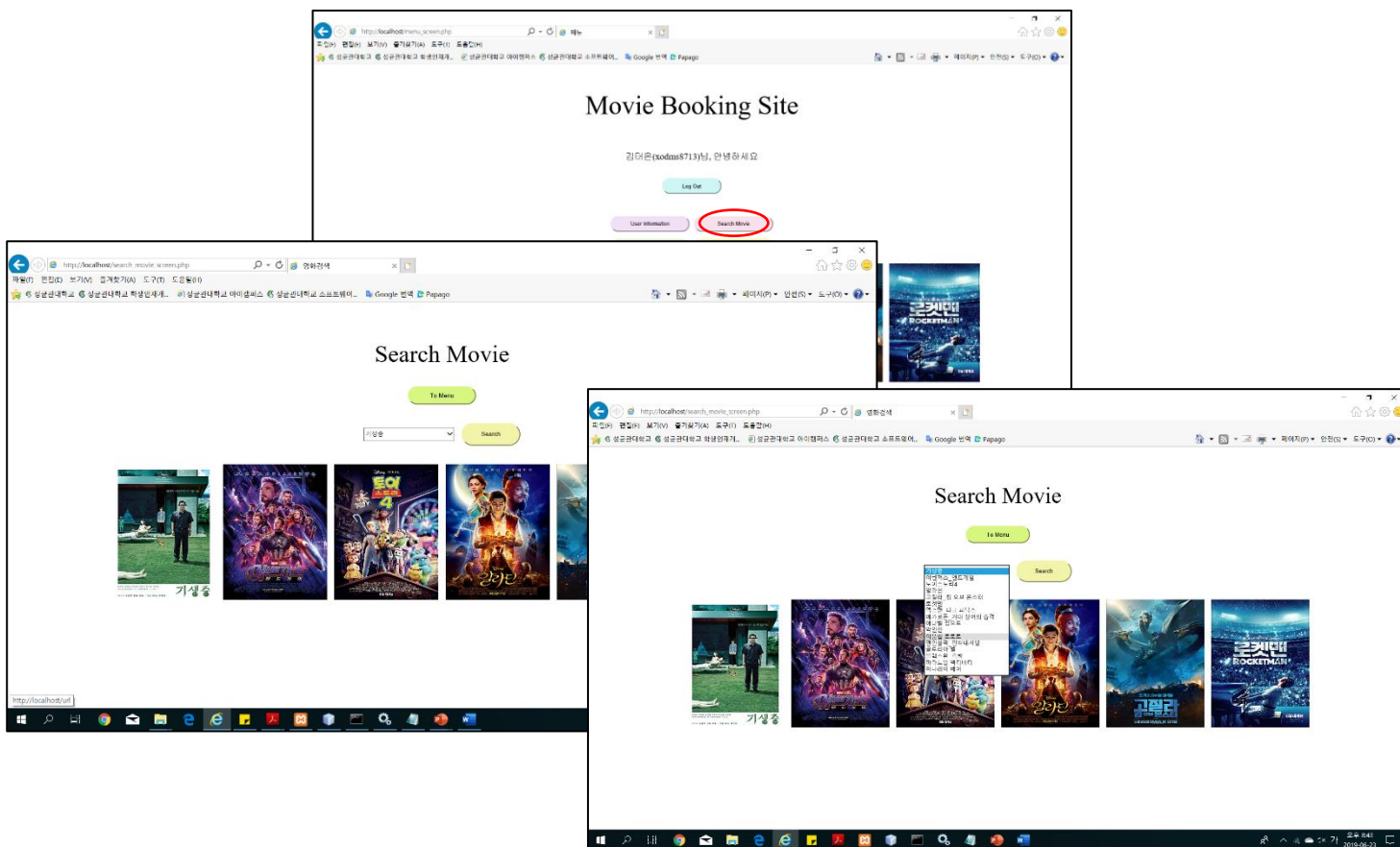
영화 예매 정보 보기 기능: user_info_screen.php, reservation_info_screen로 구현했습니다. user_info_screen에서 reservation information을 클릭하면 reservation_info_screen 창으로 이동합니다. reservation info 창에서는 mysql을 연동하여 MOVIE_RESERVATION table에서 해당 ID를 가진 사람의 예매 정보를 검색합니다(SELECT * FROM MOVIE_RESERVATION WHERE USER_ID = '{\$user_id}'). MOVIE_RESERVATION에서는 RESERVATION_NO, MOVIE_NO, THEATER_NO, SEAT의 정보만을 얻을 수 있기 때문에, 이 MOVIE_NO, THEATER_NO를 이용하여 각각 MOVIE_INFO, THEATER_INFO에서 영화, 극장에 대한 정보를 검색합니다(SELECT * FROM MOVIE_INFO where MOVIE_NO = '{\$movie_no}', SELECT * FROM THEATER_INFO where THEATER_NO = '{\$theater_no}') . 결과적으로 예매번호, 영화제목, 장르, 감독, 극장, 좌석, 상영시간의 정보를 얻을 수 있습니다.

2. membership information (멤버십 정보)



멤버십 정보 보기 기능: user_info_screen.php, membership_info_screen.php로 구현했습니다. user_info_screen에서 membership information을 클릭하면 membership_info_screen 창으로 이동합니다. membership info 창에서는 mysql을 연동하여 USER_INFO table, USER_MEMBERSHIP table에서 해당 ID를 가진 사람의 정보를 검색합니다. (SELECT * FROM USER_MEMBERSHIP WHERE USER_ID = '{user_id}', SELECT * FROM USER_INFO WHERE USER_ID = '{user_id}') USER_INFO table에서는 ID, 이름, 이메일의 정보를 얻을 수 있습니다. USER_MEMBERSHIP table에서는 회원번호, 멤버십 포인트, 가입날짜의 정보를 얻을 수 있습니다.

● Function 3: Search Movie (영화 검색)



Movie Information

To Menu

Back

*****Movie Information*****

영화제목: 애나벨 집으로

영화장르: 공포, 미스터리, 스릴러

영화감독: 게리 도버먼

*****Time Table*****

9관 08:00 ~ 11:00

10관 12:00 ~ 15:00

11관 16:00 ~ 19:00

12관 20:00 ~ 23:00

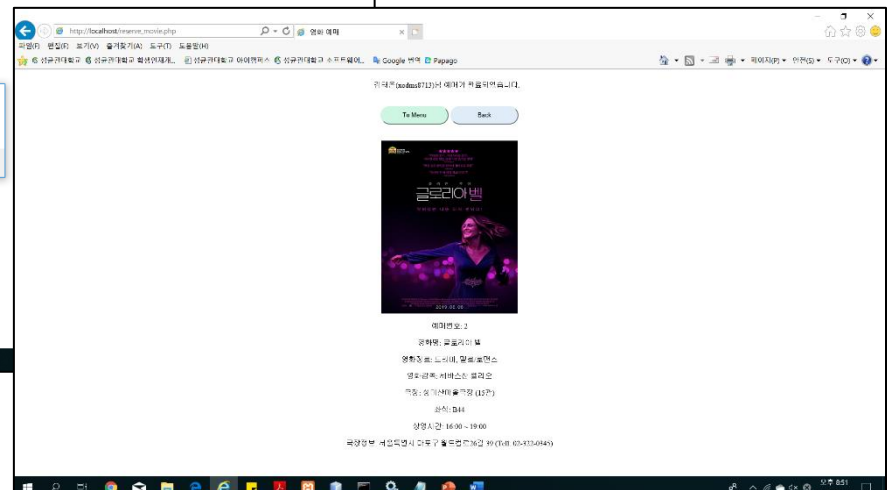
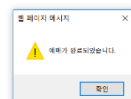
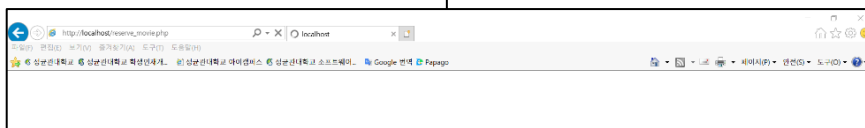
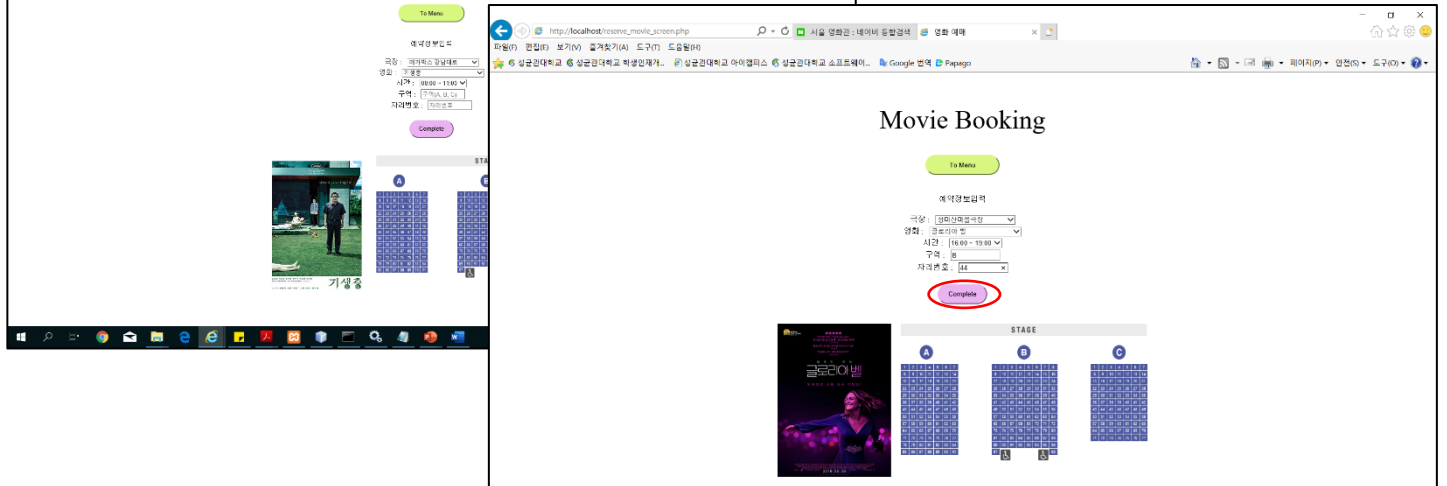
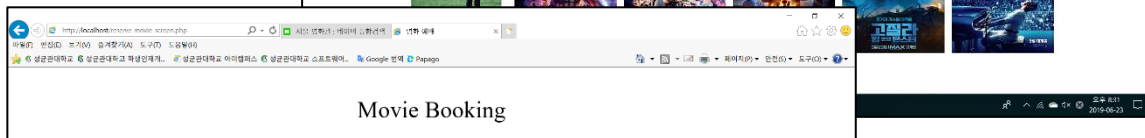
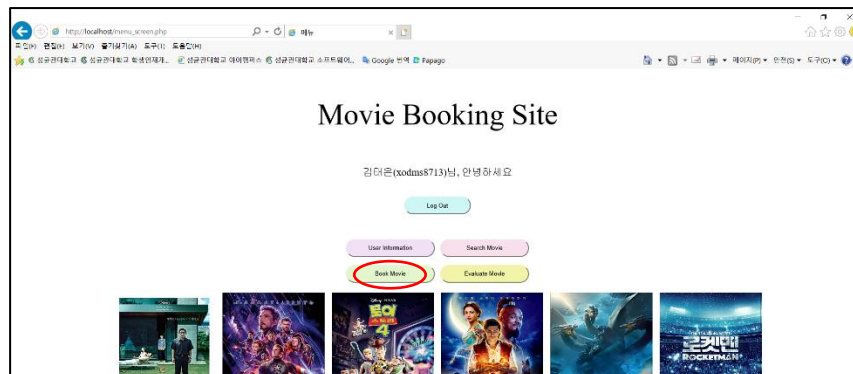
*****Movie Review*****

평가번호	작성자	명대사	평점	리뷰
1	xcodms8713	이번엔 내가 찾아갈게	9.6	재미있었다.
2	xcodms8713	애나벨, 집으로.....	4.5	너무 무섭다

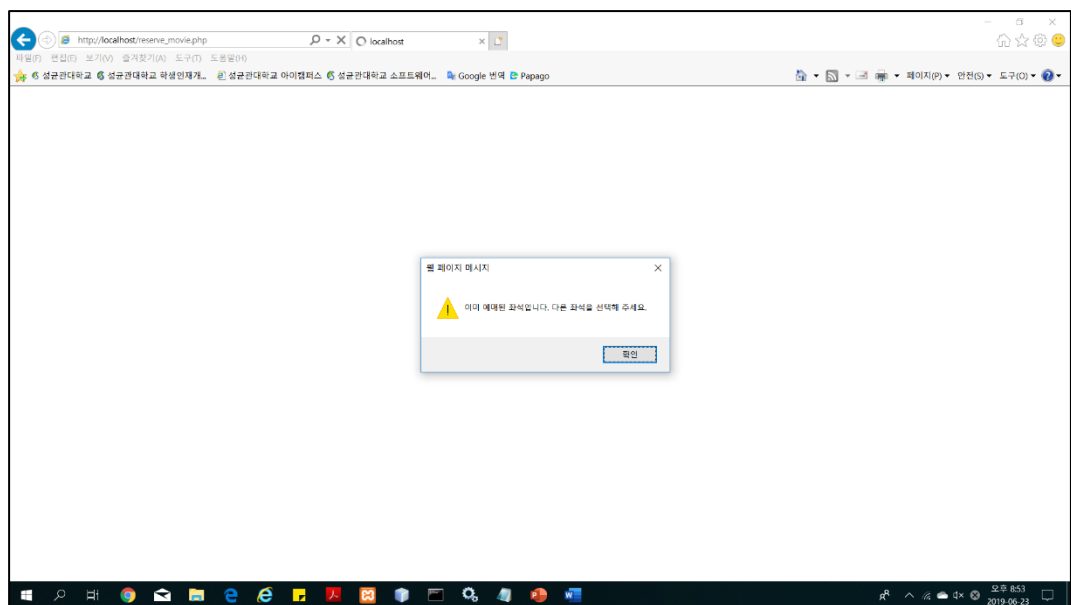
영화 검색 기능: search_movie_screen.php, search_movie.php 로 구현했습니다. MOVIE_INFO TABLE에서 서로 다른 이름을 가진 MOVIE_NAME을 SELECT하여 목록에 나열되도록 했습니다(SELECT DISTINCT MOVIE_NAME FROM MOVIE_INFO). 사용자는 이상영중인 영화 중 하나를 선택하여 영화 정보를 볼 수 있습니다. 영화를 선택한 후 search 버튼을 누르면 search_movie.php으로 이동합니다. 먼저 선택한 영화에 해당하는 영화 포스터가 등장하게 된다. " 그리고 MOVIE_INFO에서 해당 영화 이름을 가진 tuple들을 검색(SELECT * FROM MOVIE_INFO WHERE MOVIE_NAME = '{<\$_POST["movie_name"]}')하여 영화제목, 장르, 감독을 정보를 출력하고, while문을 이용하여 time table을 모두 출력할 수 있도록 했습니다. 마지막으로 영화 리뷰의 경우 MOVIE_EVALUATION table에서 해당 영화 이름을 가진 tuple들을 검색(SELECT * FROM MOVIE_EVALUATION WHERE MOVIE_NAME = '{<\$movie_name}')하여 평가 번호, 평가한 회원, 명대사, 평점, 리뷰 등이 출력될 수 있도록 했습니다.

● Function 4: Book Movie & Cancel Reservation (영화 예매, 예매 취소)

1. Book Movie

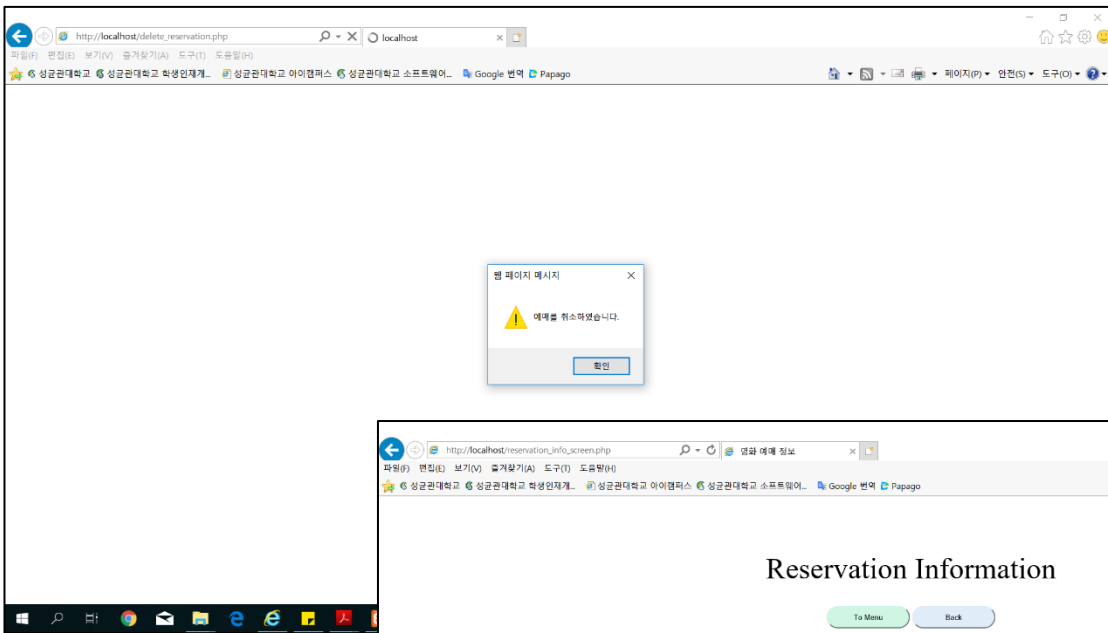
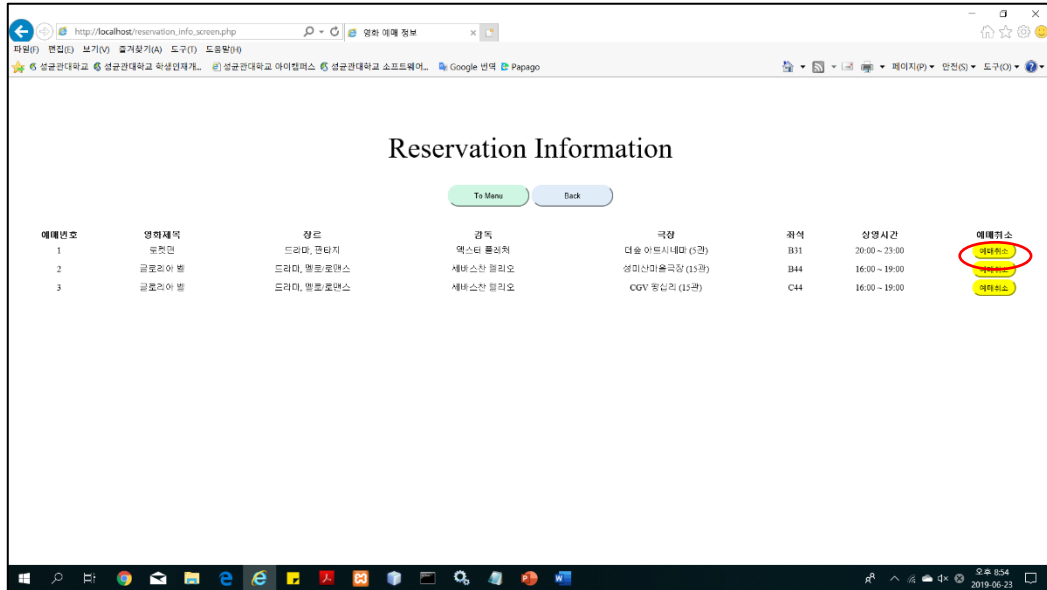


영화 예매 기능: reserve_movie_screen.php, reserve_movie.php로 구현했습니다. 극장, 영화제목, 시간, 구역, 자리번호의 정보를 받아야 합니다. 극장의 정보는 THEATER_INFO table에 미리 insert해 둔 데이터들이 목록으로 나오게끔 했습니다 (SELECT THEATER_NAME FROM THEATER_INFO). 영화 제목은 MOVIE_INFO table의 서로 다른 영화 제목들이 목록으로 나오게끔 했습니다(select distinct movie_name from movie_info). 영화 시간도 동일하게 MOVIE_INFO table의 서로 다른 영화 시간들이 목록으로 나오게끔 했습니다(select distinct movie_time from movie_info). 구역과 자리 번호는 극장 좌석 사진을 보고 사용자들이 입력할 수 있게 했습니다. 모두 입력 후 Complete 버튼을 누르면 reserve_movie.php 으로 이동합니다. MOVIE_RESERVATION table에 데이터를 insert 하기 전에 앞에서 전송된 movie_no(movie_info table을 검사하여 해당 movie_time, movie_name을 가진 tuple이 무슨 movie_no인지 위에서 확인해 둠), theater_no, seat가 모두 동일한 예약내역, 즉 중복된 예매 내역이 있는지 검사합니다. 이후 중복된 예매 내역이 없으면 MOVIE_RESERVATION table에 해당 데이터를 insert해줍니다. 또한 USER_MEMBERSHIP TABLE의 멤버십 포인트도 +100 해줍니다(UPDATE USER_MEMBERSHIP SET MEMBERSHIP_POINT=MEMBERSHIP_POINT +100). 마지막으로 평가가 완료되었다는 메시지를 띄우고 자신이 입력한 평가 정보를 확인할 수 있도록 화면에 띄워줍니다.



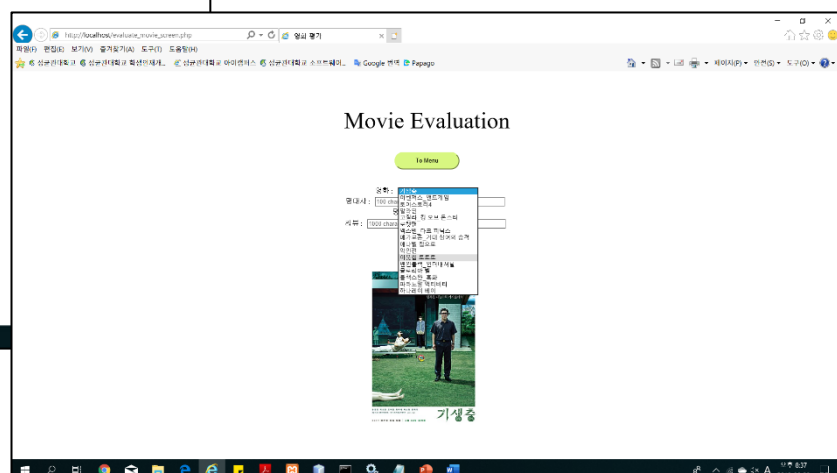
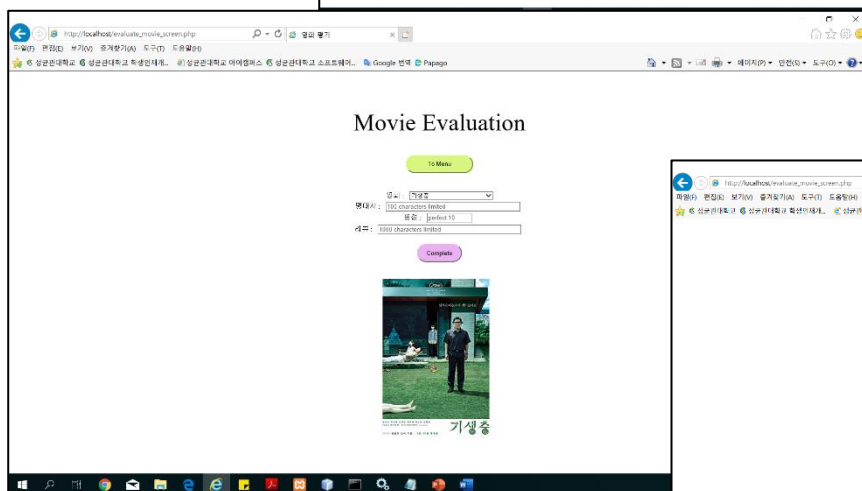
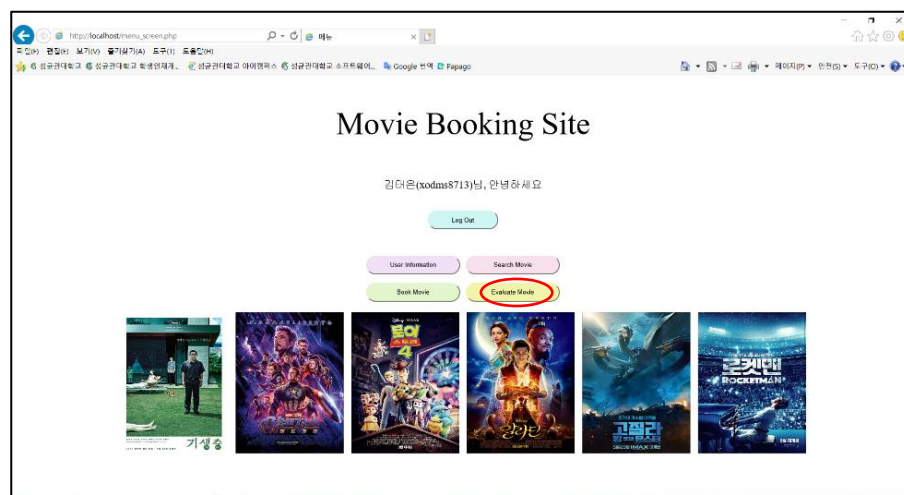
좌석 중복 확인

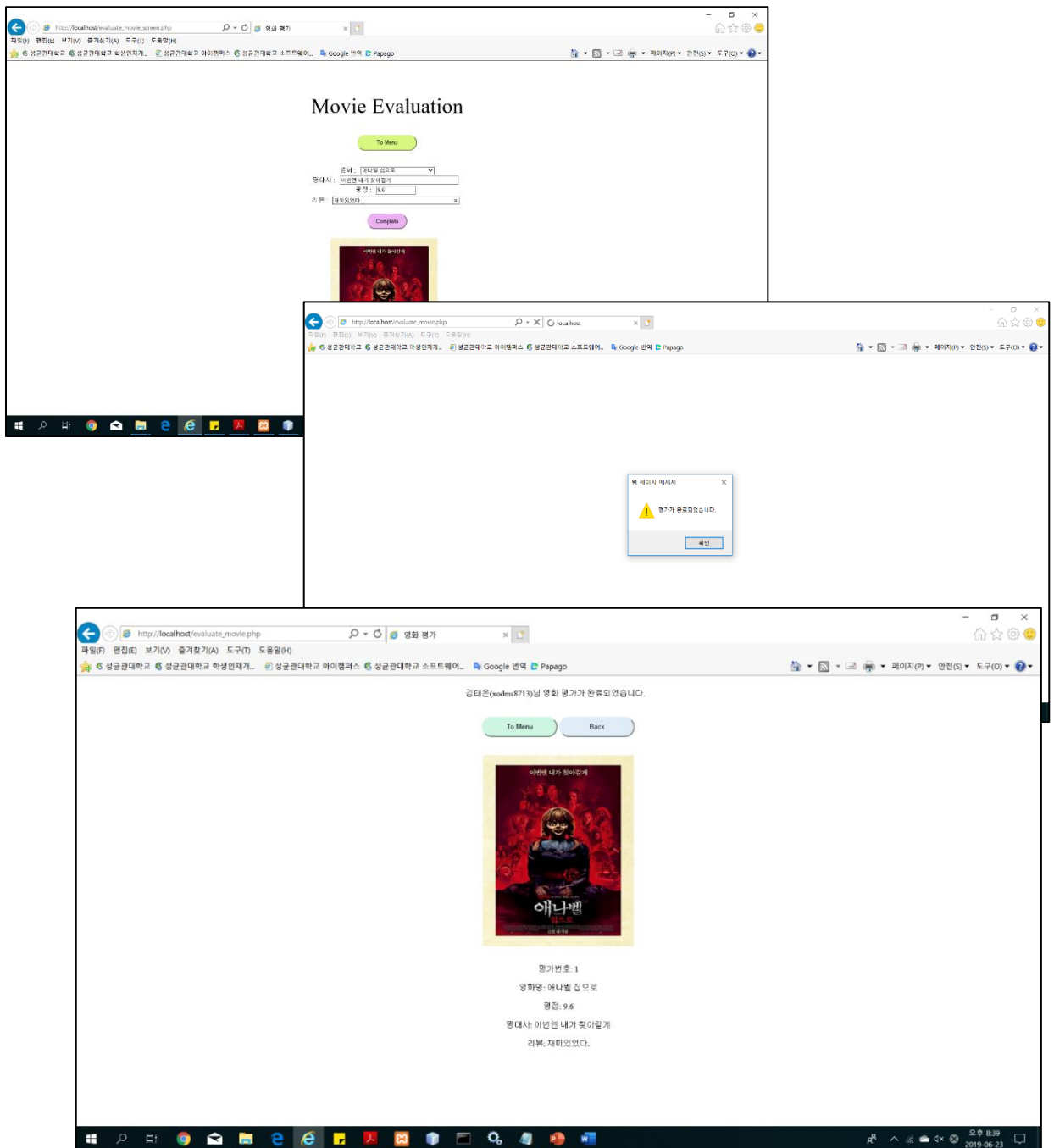
2. Cancel Reservation



예매 취소 기능: delete_reservation.php로 구현했습니다. 예매 취소 버튼은 reservation_info_screen에 등장하고, '예매 취소' 버튼을 누르면 delete_reservation.php로 이동합니다. 구현 방식은, MOVIE_RESERVATION table에서 해당 reservation_no와 같은 tuple을 delete합니다. (DELETE FROM MOVIE_RESERVATION WHERE RESERVATION_NO = '{\$reservation_no}') 또한 USER_MEMBERSHIP 에서 해당 USER의 membership point를 30점 차감합니다 (UPDATE USER_MEMBERSHIP SET MEMBERSHIP_POINT=MEMBERSHIP_POINT-30)

● Function 5: Evaluate Movie (영화 평가)





영화 평가 기능: evaluate_movie_screen.php, evaluate_movie.php로 구현했습니다. evaluate_movie_screen에서는 영화 제목, 명대사, 평점, 리뷰의 정보를 입력받게 됩니다. 영화 제목의 경우, MOVIE_INFO TABLE에서 서로 다른 이름을 가진(DISTINCT) MOVIE_NAME을 SELECT하여 목록에 나열되도록 했습니다(SELECT DISTINCT MOVIE_NAME FROM MOVIE_INFO). 또한 명대사, 평점, 리뷰를 사용자가 입력합니다. 이후 Complete 버튼을 누르면 evaluate_movie.php로 이동합니다. 모든 평가가 채

워졌는지 검사하고 빈 칸이 있을 경우(`if(empty($movie_name) || empty($movie_quote) || empty($movie_score) || empty($movie_review))`), 마저 입력하게 합니다. 그렇지 않을 경우 `MOVIE_EVALUATION` table에 해당 데이터를 insert합니다. 이후 평가가 완료되었다는 메시지를 띄우고 자신이 입력한 평가 정보를 확인할 수 있도록 화면에 띄워줍니다.