

데이터베이스 Term Project 설계서

분 반	분반		
작 품 명	Before The Trip		
개발기간	년 5 월 1 일 ~ 년 6 월 20 일		
담당교수	컴퓨터공학과	오병우 교수	
구 분	학 년	학 번	성 명
제 출 자	3학년	20190087	김건우
<p>본인은 데이터베이스 Term Project 설계서를 첨부와 같이 제출합니다. 제출한 설계서는 본인이 직접 설계하여 작성하였으며, 거짓이나 부정이 있다면 F학점을 받고 학칙에 의거하여 처벌받겠습니다 (학적부에 등재).</p> <p>2023년 05월 24일</p> <p>제 출 자 김건우</p> <p>컴퓨터공학심화프로그램</p>			

설계서

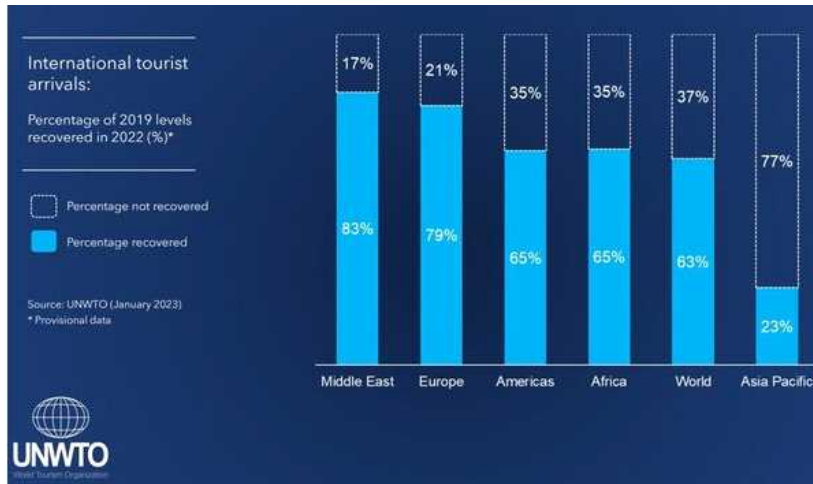
Before The Trip

1. 개요

- 개발 작품의 작품명 : Before The Trip

- 배경

이제 코로나가 잠시 종식되고 거리두기도 완화된 시점에서 최근 해외여행을 떠나는 사람들이 많아졌다.



2019년 대비 지역별 국제 여행객 회복 추이 그래프 / UNWTO

해당 통계가 2019년 코로나가 유행했을 당시에 비해 국제 여행객 회복이 점점 늘어나고 있음을 보여준다. 그렇기에 여행을 떠나려는 여행객들을 위한 사이트가 필요하다고 생각했다.

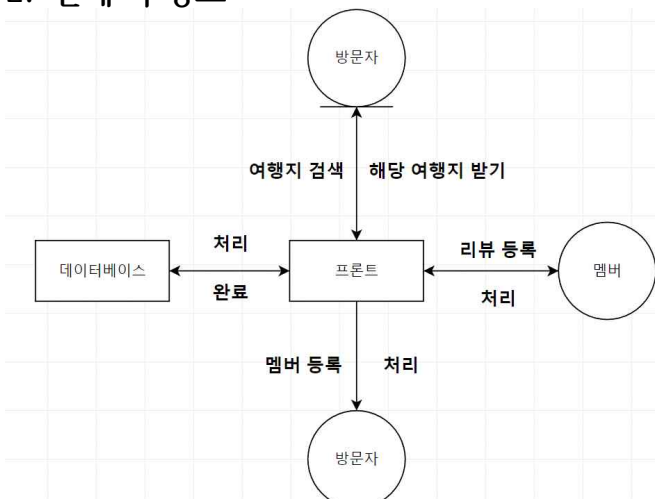
- 필요성

여행을 떠나기로 결정한 사람들이 어떤 나라로 여행을 갈지, 그 나라에서 또 어떤 지역의 어떤 관광지로 놀러 갈지 쉽게 정리되어있는 사이트가 부족하다. 그리고 해당 관광지에 대해서 미리 가본 사람들의 후기가 함께 있다면 여행을 떠날 예비 여행객들에게 많은 도움이 될 것이라 판단했다.

- 작품 설명

개발하고자 하는 작품의 이름은 'Before The Trip'이다. 이름 그대로 여행 전에 가기 전에 들리기를 원하는 마음에서 결정했다. 여행지 리스트에 추가된 여행지에 대한 정보가 있다. 해당 여행지, 국가 지역에 맞는 검색어로 검색해 원하는 여행지를 찾을 수 있다. 원하는 여행지를 선택하게 된다면 그 여행지에 등록된 리뷰 목록을 보여준다. 예비 여행객들은 해당 리뷰를 보고 여행지 선택에 있어서 참고할 수 있다. 또한 여행지를 방문한 뒤 리뷰를 남길 수도 있다. 코로나 이후 여행객들이 많아짐에 있어서 'Before The Trip' 사이트가 많은 여행객들에 의해 사용될것이라 추측 가능하다.

2. 전체 구성도



'Before The Trip'을 구현함에 있어서 구조를 설명하자면, 회원가입의 기능은 아직 정확히 배우지 않아서

따로 구현하진 않을 것이다. 그래서 방문자와 멤버로 나뉘는데 방문자에게는 여행지 목록에서 검색, 받기 정도의 기능, 멤버에 포함되어있는 사람들에게는 리뷰 작성이 가능하도록 할 것이다. 또한 데이터베이스에 접근이 필요한 경우 React로부터 클라이언트에게 받아온 정보로 Express와 DB를 연결해 요구를 수행할 것이다. React를 통해 요구 처리 여부에 맞게 상황에 맞는 화면을 띄워줄 것이다.

3. Use Case (요구 사항 분석)

페이지 별로 나눠서 해당 요구 사항을 설명했습니다.

3.1 메인 페이지

- 메인에서는 간단하게 해당 사이트에 대한 한줄 소개를 넣는다.
- 또한 멤버 목록으로 들어갈 수 있는 버튼과 여행지 리스트를 볼 수 있는 버튼을 추가한다.
- 각각의 버튼을 누르면 해당되는 사이트로 이동되게 한다.

3.2 여행지 리스트 페이지

- 등록되어있는 모든 여행지의 목록을 보여준다.
- 해당 페이지에서 여행지 검색, 추가, 수정, 삭제 기능을 추가한다.
- 해당 여행지를 누르면 그 여행지에 등록된 리뷰 목록을 보여준다.
- 정렬 기능을 추가해 최신순, 평점순으로 편하게 확인 가능하다.

3.2.1 여행지 검색

- 여행지, 국가, 지역 3개의 필드를 비교해서 해당 검색어가 들어가있다면 해당되는 모든 목록을 출력한다.
- 검색된 목록에서 최신, 평점 순으로 정렬 가능하다.

3.2.2 여행지 추가

- 추가 페이지로 이동해 이름, 국가, 지역, 설명, 이미지를 입력하고 추가 가능하다.

3.2.3 여행지 수정

- 수정은 리스트에 필드 오른쪽에 버튼을 추가해 버튼을 누르면 수정 페이지로 이동한다.
- 수정 페이지에서는 이름, 국가, 지역, 설명, 이미지가 기존에 있는 값으로 들어있다.
- 수정할 부분을 수정한 뒤, 수정 버튼을 누르면 수정이 완료된다.
- 수정된 여행지는 정렬할 때 최신에 적용되지 않는다.

3.2.4 여행지 삭제

- 삭제는 리스트에 필드 오른쪽에 버튼을 추가해 버튼을 누르면 바로 해당 필드와 안에 포함된 리뷰목록을 삭제한다.

3.3 리뷰 페이지

- 해당 여행지를 클릭하게되면, 해당 여행지에 포함된 리뷰 목록이 들어있는 리뷰 페이지가 나온다.
- 위에 어떤 여행지에 대한 리뷰인지 표시한다.
- 큰 틀은 여행지 리스트 페이지와 동일하다. 리뷰 추가, 수정, 삭제가 가능하다.
- 리뷰에서는 평점으로만 검색할 수 있다. 즉 평점별로 나누어 볼 수 있다.

3.3.1 리뷰 추가

- 추가 페이지에서는 이메일을 입력하는 부분이 있는데, 만약 해당 이메일이 멤버 목록에 존재하지 않는다면, 오류 알람을 보낸다. 즉 멤버 목록에 존재하는 이메일만 리뷰 추가가 가능하다.

3.4 멤버 페이지

- 메인에서 멤버 목록 버튼을 누르게되면 멤버 페이지로 올 수 있다.

- 멤버 페이지 또한 여행지 목록 페이지와 비슷하게 검색, 추가, 수정, 삭제 기능이 구현되어있다.

4. UI Design

Before The Trip

Before The Trip!

여행가기전 둘러보기 좋은 리뷰사이트입니다

멤버 목록 여행지 리스트

- 메인화면으로 심플하게 흰 바탕에 간단한 설명 한줄과 멤버 목록, 여행지 리스트 버튼이 있다.

Before The Trip

여행지 리스트

검색어를 입력하세요		검색	정렬						
번호	여행지	국가	최신순	설명	평점(평균)	이미지	수정	삭제	
1	런던	영국	평점순	영국의 대표적인 도시 중 하나로, 역사와 문화가 곳곳에 존재합니다.	3.25		수정	삭제	
2	시드니	호주	오세아니아	국제적인 도시에서의 해변 생활과 볼거리가 많은 도시입니다.	3.75		수정	삭제	
3	타이페이	대만	아시아	모던하고 오래된 문화가 공존하는 대만의 수도입니다.	3.75		수정	삭제	
4	코타키나발루	말레이시아	아시아	저렴한 가격과 다양한 자연경관, 문화를 느낄 수 있는 도시입니다.	3.75		수정	삭제	

- 여행지 리스트 버튼을 누르면 해당 페이지로 이동하게 된다. 검색을 통해 원하는 여행지를 검색할 수도 있고, 옆에 드롭다운을 통해 최신순 혹은 평점순으로 정렬이 가능하다.

Before The Trip

여행지 추가

- 오른쪽 위에 + 버튼을 누르면 추가 페이지로 이동한다. 필요한 정보를 모두 입력 후 추가할 수 있다.

Before The Trip

여행지 수정

여행지 이름
런던

국가
영국

지역
유럽

설명
영국의 대표적인 도시 중 하나로, 역사와 문화가 곳곳에 존재합니다.

이미지
London

수정

- 수정 버튼을 누르면 그 필드에 있는 정보가 입력된 상태인 페이지로 이동한다. 수정할 부분을 수정한 뒤 수정 버튼을 누르면 수정이 완료된다.

Before The Trip

여행지 '런던'에 대한 리뷰

번호	작성자	분점	내용	수정	삭제
1	김우현	4	다시 가고 싶은 곳입니다.	수정	삭제
2	홍길동	1	최악 중의 최악이었습니다. 절대 가지 마세요!	수정	삭제
3	강소라	4	즐거운 시간을 보냈고, 추천해요.	수정	삭제
4	이철수	4	또 가고 싶은 곳입니다.	수정	삭제

- 여행지를 누르게 되면, 해당 여행지에 등록된 리뷰 목록이 나오게 된다. 왼쪽 위에 어떤 여행지에 대한 리뷰인지 나와있고, 리뷰 또한 추가, 수정, 삭제가 구성되어있다.

Before The Trip

리뷰 추가

이메일이 존재하지 않습니다.

이메일
test@naver.com

별점 (1-5)
5

내용
good

추가

- 테스트의 추가 기능에는 이메일을 체크하는데 예시처럼 이메일이 존재하지 않는다면 알람을 띄워준다. 만약 있는 이메일이라면 해당되는 작성자의 이름으로 리뷰를 추가한다.

Before The Trip

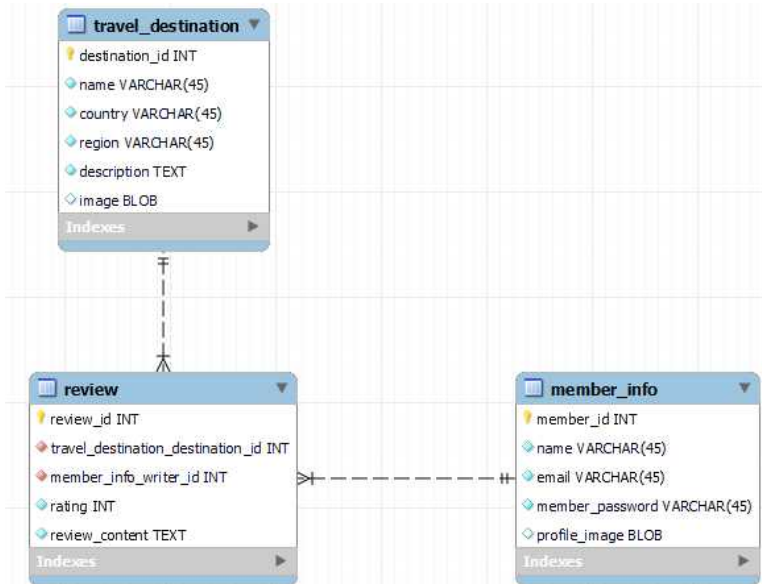
멤버 목록

리스트	이름	이메일	나이	이미지	수정	삭제
1	김건우	kwforu123@naver.com	24		수정	삭제
2	홍길동	honggildong@example.com	25		수정	삭제
3	김영희	kimyeonghee@example.com	30		수정	삭제
4	이철수	leecholsu@example.com	28		수정	삭제

- 메인에서 멤버 목록을 누르면 해당 페이지로 이동한다. 여기 또한 크게 다를 것 없이 목록을 출력하고 추가, 수정, 삭제 검색 기능을 구현한다.

5. Conceptual Design

- Crow's foot 으로 구현



여행지에 여러 개의 리뷰가 있을 수 있고, 리뷰에는 하나의 여행지가 있기 때문에 1:다 관계다. 한 멤버가 여러 개의 리뷰를 작성할 수 있고, 리뷰는 하나의 멤버만 존재하기 때문에 1:다 관계로 설정했다. 여행지와 멤버는 연 관 관계가 없다고 생각해 따로 처리해주지 않았다.

6. Logical Design

- member_info 테이블

```

-- 회원 정보 테이블 생성
CREATE TABLE member_info (
    member_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NOT NULL,
    email VARCHAR(45) NOT NULL,
    age VARCHAR(45) NOT NULL,
    profile_image BLOB,
    PRIMARY KEY (member_id)
) ENGINE = InnoDB
  DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_general_ci;
    
```

Field	Type	Null	Key	Default	Extra
member_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(45)	NO		NULL	
email	varchar(45)	NO		NULL	
age	varchar(45)	NO		NULL	
profile_image	blob	YES		NULL	

5 rows in set (0.03 sec)

- travel_destination 테이블

```

-- 여행지 테이블 생성
CREATE TABLE travel_destination (
    destination_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NOT NULL,
    country VARCHAR(45) NOT NULL,
    region VARCHAR(45) NOT NULL,
    description TEXT NOT NULL,
    image BLOB,
    PRIMARY KEY (destination_id)
) ENGINE = InnoDB
  DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_general_ci;
    
```

Field	Type	Null	Key	Default	Extra
destination_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(45)	NO		NULL	
country	varchar(45)	NO		NULL	
region	varchar(45)	NO		NULL	
description	text	NO		NULL	
image	blob	YES		NULL	

6 rows in set (0.00 sec)

- review 테이블

```
-- 리뷰 테이블 생성
CREATE TABLE review (
  review_id INT NOT NULL AUTO_INCREMENT,
  destination_id INT NOT NULL,
  writer_id INT NOT NULL,
  rating INT NOT NULL,
  review_content TEXT NOT NULL,
  FOREIGN KEY (destination_id) REFERENCES travel_destination(destination_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (writer_id) REFERENCES member_info(member_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  PRIMARY KEY (review_id)
) ENGINE = InnoDB
DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;

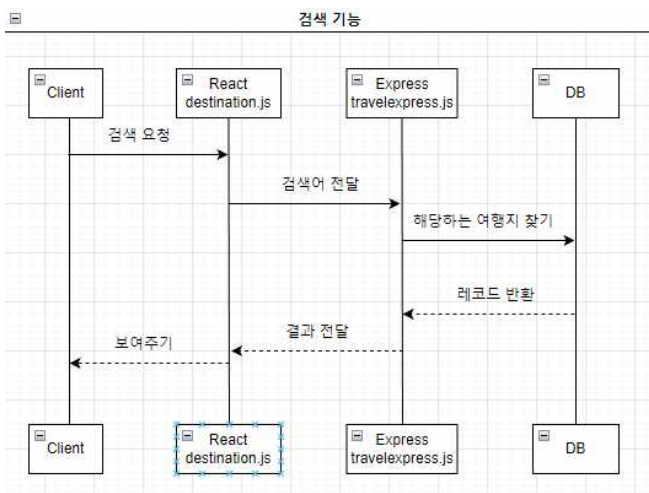
mysql> desc review;
```

Field	Type	Null	Key	Default	Extra
review_id	int(11)	NO	PRI	NULL	auto_increment
destination_id	int(11)	NO	MUL	NULL	
writer_id	int(11)	NO	MUL	NULL	
rating	int(11)	NO		NULL	
review_content	text	NO		NULL	

5 rows in set (0.00 sec)

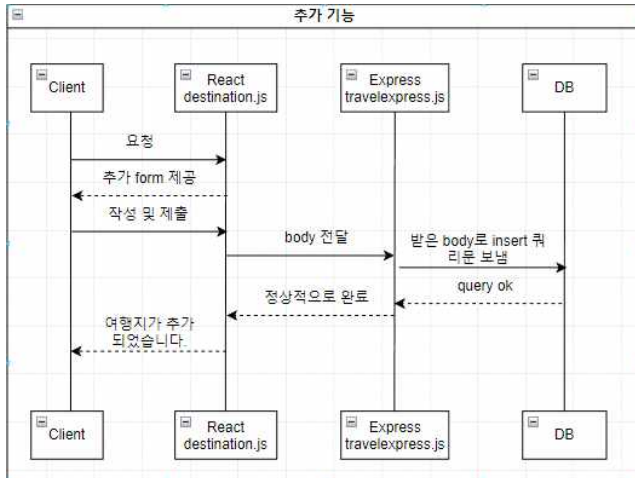
7. Sequence Diagram

7.1 여행지 검색



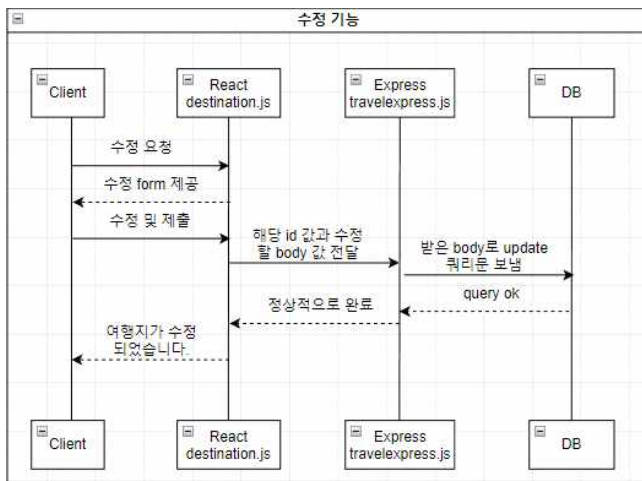
- 사용자가 여행지를 검색하는 요청할 때의 시퀀스 다이어그램이다. 사용자가 여행지를 검색하는 요청을 React의 해당 파일로 보내면 해당 키워드를 Express 파일로 전달한다. Express 파일에서 쿼리를 보내 DB 안에서 모든 대상을 검색해 구성 요소에 목록을 반환한다.

7.2 여행지 추가



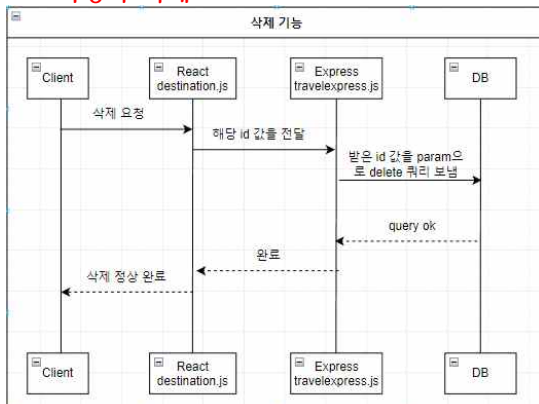
- 사용자가 여행지를 추가할 때 시퀀스 다이어그램이다. client에서 요청을 하면 React에서 해당 body를 작성할 수 있는 form을 제공한다. client에서 작성 및 제출을 하면 React에서 본문을 Express로 전달하여 해당 본문 값으로 post 방식의 insert 쿼리문을 통해 DB에 추가 가능하다.

7.3 여행지 수정



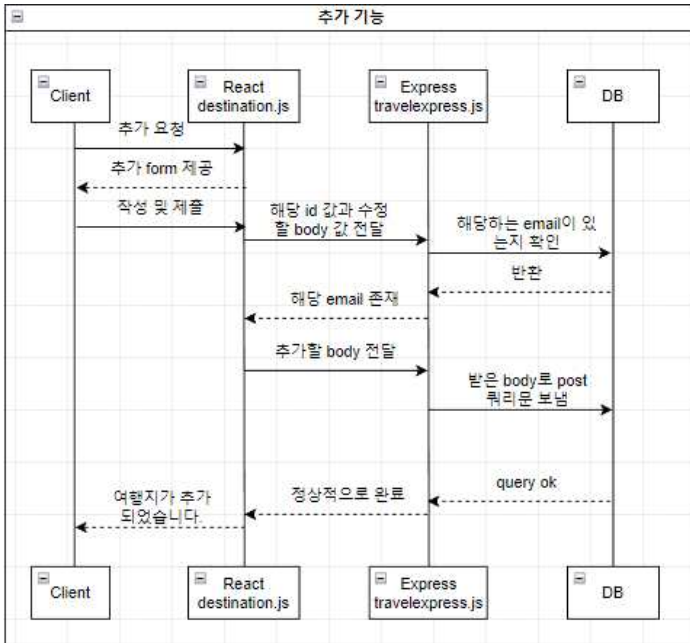
- 여행지 수정할 때 시퀀스 다이어그램은 추가와 거의 흡사하다. client에서 수정 요청 시 React에서 수정 form을 제공해 수정 및 제출한다. React 파일에서 여행지의 id 값과 수정한 body 값을 Express 파일로 전달한다. Express 파일에서 id를 params로 받고, body를 받아 update 쿼리문을 put 요청으로 DB에 보낸다.

7.4 여행지 삭제



- 삭제 시퀀스 다이어그램은 쉽게 나타낼 수 있다. 삭제 요청이 들어오면 React 파일에서 삭제할 여행지의 id를 Express 파일로 전달한다. 받은 id 값을 params으로 delete 형식의 delete 쿼리를 DB로 보낸다.

7.5 리뷰 추가



- 사용자가 리뷰를 추가할 때 시퀀스 다이어그램이다. 여행지 추가와 비슷하지만, 필드에서 email을 확인해서 member_info 테이블에 해당 email이 존재하는지 확인한다. 존재한다면 추가할 body를 다시 넘겨주고, 존재하지 않는다면 오류를 출력하고, React에서는 알람을 보낸다. body를 넘겼다면 해당 본문 값으로 post 방식의 insert 쿼리문을 통해 DB에 추가 가능하다.

- 나머지 리뷰의 수정, 삭제, 검색은 여행지 수정, 삭제, 검색과 동일하다.

- 또한 멤버의 검색, 추가, 수정, 삭제 또한 파일의 이름, 비교할 필드의 값만 다를 뿐 전체적인 시퀀스 다이어그램은 동일하다.

8. API 설계

- 여행지 (예시)

HTTP 요청 : GET <https://travel-review.run.goorm.site/destination>

응답 : 성공시 다음과 같은 응답 본문 반환 - 다음 예시처럼 모든 여행지를 반환

```
[
  {
    "destination_id": 1,
    "name": "괌",
    "country": "미국",
    "region": "서태평양",
    "description": "괌은 미국의 대표적인 해외 자치영토로서, 다양한 해양 스포츠와 관광 명소를 보유하고 있습니다.",
    "image": "Guam.jpg",
    "average_rating": 3.8
  },
  ...
]
```

HTTP 요청 : POST <https://travel-review.run.goorm.site/destination>

```
{
  "name": "산토리니",
  "country": "이탈리아",
  "region": "유럽",
  "description": "세련되고 아름다운 이탈리아의 작은 도시로 유명한 산토리니는 토스카나 지방의 중심에 있습니다.",
  "image": "image.jpg"
}
```

응답 : 성공시 다음과 같은 응답 본문 반환

```
{
  "result": "success"
}
```

HTTP 요청 : PUT <https://travel-review.run.goorm.site/destination/1>

```
{
  "name": "산토리니",
  "country": "이탈리아",
  "region": "유럽",
  "description": "세련되고 아름다운 이탈리아의 작은 도시로 유명한 산토리니는 토스카나 지방의 중
  심에 있습니다.",
  "image": null
}
```

응답 : 성공시 다음과 같은 응답 본문 반환

```
{
  "result": "success"
}
```

HTTP 요청 : DELETE <https://travel-review.run.goorm.site/destination/1>

응답 : 성공 시 Destination deleted successfully

- 리뷰 (예시)

HTTP 요청 : GET <https://travel-review.run.goorm.site/review>

응답 : 성공시 다음과 같은 응답 본문 반환 - 다음 예시처럼 모든 리뷰를 반환

```
[
  {
    "review_id": 2 ,
    "destination_id": 1 ,
    "writer_name": "홍길동",
    "rating": 4 ,
    "review_content": "너무 좋았습니다. 추천합니다."
  },
  ...
]
```

HTTP 요청 : POST <https://travel-review.run.goorm.site/destination/1/review>

```
{
  "email": "kwforu123@naver.com",
  "rating": 3,
  "review_content": "무난했습니다."
}
```

응답 : success

HTTP 요청 : PUT <https://travel-review.run.goorm.site/review/1>

```
{
  "rating": 1,
  "review_content": "별로입니다."
}
```

응답 : success

HTTP 요청 : DELETE <https://travel-review.run.goorm.site/review/1>

응답 : success

- 멤버 (예시)

HTTP 요청 : GET <https://travel-review.run.goorm.site/member>

응답 : 성공시 다음과 같은 응답 본문 반환 - 다음 예시처럼 모든 멤버를 반환

```
[
  {
    "member_id": 1 ,
    "name": "김건우",
    "email": "kwforu123@naver.com",
    "age": "24",
    "profile image": "1.jpg"
  }
]
```

```

    },
    ...
]

HTTP 요청 : POST https://travel-review.run.goorm.site/member
{
  "name": "김진우",
  "email": "kwforul23@naver.com",
  "age": 24,
  "profile_image": null
}
응답 : success

HTTP 요청 : PUT https://travel-review.run.goorm.site/member/1
{
  "name": "김진우",
  "email": "kwforul23@naver.com",
  "age": 24,
  "profile_image": null
}
응답 : success

HTTP 요청 : DELETE https://travel-review.run.goorm.site/member/1
응답 : success

```

9. 설계 구성 요소 및 제한 요소 충족 여부

<설계구성요소 및 설계제한 요소>

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/ 제작	시험/ 평가	결과 도출	성능	규격/ 표준	경제 성	미 학	신뢰성	안정성/ 내구성	환경
○		○	○				○		○	○		○

9.1 설계 구성 요소

- 목표 설정: 웹을 기반으로 자신이 좋아하는 주제의 DB를 설계하고 백엔드 및 프론트엔드 개발
- 분석: 취미 DB 분석
- 구현/제작 : 데이터베이스를 정규화 규칙 BCNF에 기반하여 제작.
- 시험/평가 : 고정아이피로 서버를 열어 다른 사람들이 웹페이지에 접속할 수 있도록 해 평가를 받는다.

9.2 설계 제한 요소

(1) 규격/표준

- 규격/표준: (React, Express, MySQL) 사용, 테이블 3개 이상 사용, Join 연산 사용
- 테이블
 - member_info : 멤버 id, 이름, 이메일, 나이, 프로필 이미지를 저장한다.
 - travel_destination : 여행지 id, 이름, 나라, 지역, 내용, 이미지를 저장한다.
 - review : 리뷰 id, 여행지 id, 멤버 id, 평점, 리뷰 내용을 저장한다.
- Join 연산은 travel_destination 과 review를 여행지 id로 join하여 여행지의 정보를 출력함에 있어서 해당 여행지의 리뷰들의 평균 평점을 구해서 정렬기능에 평점순으로 정렬 가능하게 만들 것이다.
또한 member_info와 review를 멤버 id로 join해서 리뷰 목록을 출력하는데 있어서 작성자의 이름과, 평점, 리뷰 내용을 출력할 것이다.

(2) 미학

- 미학: 미려하고 직관적인 사용자 인터페이스 설계, Image 출력 (이미지 경로 또는 DB 내의 BLOB Type 사용)

- 사용한 템플릿 명시

따로 템플릿을 사용하지 않았고, 강의에서 배운 MUI를 통해 구현했다. MUI의 구성요소 중에서 여러개를 import 해서 필요한 부분에 적절히 적용 시켰다.

ex) Button, Box, AddIcon, Select, MenuItem 등등

(3) 신뢰성

- 신뢰성 : DB 반영하여 웹페이지로부터 데이터를 삽입하는 여행지 추가, 리뷰 추가, 멤버 추가 기능과 데이터를 삭제하는 여행지 삭제, 리뷰 삭제, 멤버 삭제, 수정 연산 그리고 DB에 저장된 정보중에서 키워드를 찾아 데이터를 반환하는 검색 연산을 통해 여행지와 리뷰 관리를 할 수 있는 페이지를 개발할 것이다.

(4) 환경

- 환경: 웹 환경에서 React, Express, MySQL, JSON 사용한다.

10. 결론 및 자체 평가

- 설계 과정 요약

사용자의 정보를 저장하는 member_info, 여행지의 정보를 저장하는 travel_destination, 리뷰 정보를 저장하는 review 테이블을 만들었다. 그리고 해당 테이블끼리의 연관관계를 생각해 MySQL WorkBench를 이용해 E-R 다이어그램을 그려 개념적 설계를 했다. 개념적 설계를 바탕으로 직접 SQL 코드를 작성해서 논리적 디자인을 해보았다. 이후 UI 디자인을 했다. React를 바탕으로 강의에서 배운 MUI를 사용해서 테이블의 필드들을 목록기반으로 표시했다. 아이콘, 드롭다운 같은 MUI에서 제공하는 다양한 아이템들을 사용해서 구현했다. 해당 React 파일과 DB 구조를 바탕으로 추가, 수정, 검색, 삭제 기능이 어떻게 작동되는지 Sequence Diagram을 그려서 요청부터 완료까지 흐름대로 그려보았다. 마지막으로 API 설계를 했다. 백엔드에서 HTTP 요청과 요청에 해당하는 body를 보냈을 때 어떤 응답이 오는지 설계했다.

- 설계 과정과 결과에 대하여 어떻게 느끼고 생각하는지 적고 수행 수준을 스스로 평가하여 기록.

주제를 정하고 실제로 구현해보니 그다지 많은 테이블이 필요하지 않았다. 적절히 3개의 테이블을 사용해 연관 관계를 설정했다. 테이블을 따로 추가해볼까 했지만, 오히려 깔끔하지 못한 데이터 베이스 설계가 되는 것 같아서 굳이 추가하지 않았다. 3개의 테이블이지만 해당 테이블들을 적절한 Join을 통해 직접 원하는 결과를 도출해 보았다. AVG, LEFT JOIN, GROUP BY 등 SQL에서 제공하는 키워드를 직접 사용해보았다. 내가 원하는 주제로 처음부터 끝까지 설계를 해보니 정말 생각할 부분이 많음을 느꼈다. 이렇게 간단한 주제에 대해서도 생각해줄 부분이 많았다. 테이블 사이의 관계나 백엔드, 프론트엔드 등. 결과적으로 설계를 성공적으로 마친 후, 내 힘으로 개념적, 논리적 설계도 직접 해보고 DB 설계, 프론트 설계까지 해보니 엄청난 뿌듯함을 느꼈다. 수행 수준은 중간 이상 정도라고 생각한다. 적절한 테이블 수를 사용했으며 Join 연산을 통해 직접 원하는 결과를 도출하고, React를 통한 프론트 작업까지 완성했다.

- 과제를 설계하면서 어려웠던 점과 그 문제를 어떻게 해결했는지 적을 것

사실 모든 부분이 익숙하지 않아서 난 상대적으로 어려웠다. 처음 주제를 정하고 테이블을 생성하는 과정부터 개념적 설계, 논리적 설계, React를 이용한 페이지 생성, API 설계까지 뭐 하나

쉽게 해결한 부분이 없었다. 그중 가장 어려웠던 점은 처음 설계가 가장 어려웠다. 다른 것들은 코드를 디버깅하면서 결과를 확인하면서 해결해 나갈 수 있었다. 하지만 데이터베이스 설계는 처음이라 내가 하려는 여행 리뷰 주제에 있어서 어떤 테이블이 필요할지 생각해 먼저 테이블을 생성해 놓고, 각 테이블에 필드를 채워넣었다. 어려웠지만 결국 이 문제의 해결은 직접 해보는 것이었다. 테이블을 만들어 놓고 각 필드에 대해서 어떤 것을 Primary Key로 놓을지, null 허용 여부는 어떻게 할지, Foreign Key의 수정, 삭제에 대해서 어떤 행동을 할지 등 내가 머릿속으로 생각해서 정리해 구현하는 것이 해결 방법이었다.