

2020년도 2학기

웹 시스템 설계 Term 프로젝트 결과보고서

프로젝트 명	스터디카페 웹시스템
학번	201620931
이름	김영우

1. 개요

올해는 코로나로 인해 통학을 하지 못하고, 집에서 비대면으로 진행하는 수업이 많았습니다. 코로나로 상황이 나아지지 못하고 나빠지고 있었고, 비대면 수업은 1학기뿐 만이 아닌 2학기또한 이어졌습니다. 백신이 배포되기 전까지는 아마도 다음학기, 내년 1학기도 비대면으로 진행하게 될 확률이 큼니다. 즉 앞으로도 공부를 학교에서 할 수 있을 확률은 희박하다고 생각합니다. 모든 수업을 집에서 들으면 또 집에만 있어 익숙하다 보니 공부하기가 쉽지 않고, 집에서만 공부하기에는 불편하기 때문에, 많은 사람들이 카페에 가서 공부를 하거나, 스터디 카페로 향했습니다. 저 또한 스터디 카페를 많이 이용했는데, 현재 존재하는 카페/스터디 카페의 가장 큰 문제점은 카페를 방문하기 전까지는 카페에 자리가 얼마나 남았는지 알기가 어렵다는 것이었습니다.

따라서 저는 카페의 자리를 알기 쉽게 인터넷으로 보여주는 웹 시스템을 만들면 기존의 스터디 카페가 가지고 있는 단점을 해결할 수 있다고 생각했습니다. 따라서 이번 웹 시스템 설계 Term 프로젝트의 목표는 기존의 스터디 카페 프로그램을 이번에 학습한 React, Express.js등을 이용해 웹으로 구현해보는 것입니다.

2. 프로젝트 내용

먼저 Server-Side(백엔드)에는 Database에 접근할 수 있어야 합니다. 저는 NoSQL중 하나인 mongoDB를 이용했습니다. 아직 MySQL과 같은 RDBMS계열은 아직 데이터베이스 과목을 배우고 있는 중이고, 교수님께서 MongoDB를 수업에서 잘 설명해 주셨기 때문에, MongoDB를 쓰는 것이 더 쉽고 안전하겠다는 생각을 했습니다.

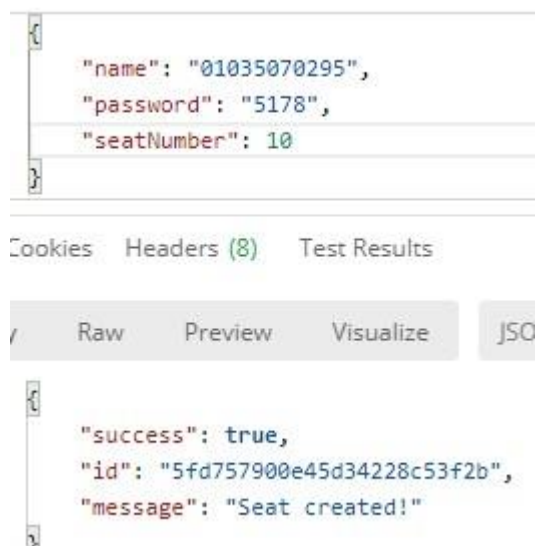
MongoDB로 쓰이는 스키마 이름은 Café로 정했고, 총 5가지의 항목을 Column으로 정했습니다. 각각 id, name, password, seatNumber, time에 해당합니다. Id는 각각의 문서들(항목들)이 가지고 있는 고유의 번호, name은 캐릭터형이 들어갑니다. 기존의 스터디 카페 프로그램은 전화번호를 입력하는 방식으로 돌아가기 때문에, 제 웹 시스템에서도 전화번호가 들어가는 것을 전제로 프로젝트를 만들었습니다. Password는 비밀번호, seatNumber는 자리의 번호가 들어갑니다. 독서실의 자리 번호가 들어간다고 생각하시면 됩니다. 마지막으로 time은 시간이 들어갑니다. 백엔드를 다 만들고 난 뒤, 이 백엔드가 제대로 돌아가는지 확인해보기 위해, mongoDB Compass와 Postman을 이용해서 확인을 했습니다. Controller에서 DELETE, POST, GET등의 asynchronous를 구현하고, router로 연결했습니다. 최종적으로 백엔드에서는 자리 추가, 자리 업데이트(자리 수정), 자리 삭제, 자리 가져오기를 구현했습니다.

프론트엔드로는 React를 이용해서 구현했습니다. 역시 교수님께서 React를 기준으로 설명을 해주셨기 때문에, React를 사용했습니다. React에서 axios를 이용해서 백엔드로 사용한 express node.js를 브라우저에서 값을 입력할 수 있게 만들었습니다. React에서 값을 쉽게 넣을 수 있게 필요한 UI도 구현을 해주었습니다. 먼저 home.js로 웹 서비스를 사용하는 방법을 적어 두었고, List.js를 만

들어서 mongoDB에 존재하는 데이터들을 여기서 출력할 수 있게 만들었습니다. Insert.js와 update.js도 만들어서 insert에는 자리를 추가하고, update는 자리를 수정할 수 있게 만들었습니다. Insert.js, update.js는 독립적인 html 웹 사이트를 만들어 주었지만, delete에는 추가적인 데이터를 넣는 것이 아니라, 오직 목적이 하나인 삭제를 하는 것이기 때문에 delete.js와 같은 파일을 만들어 주지는 않고, List.js에서 직접 삭제할 수 있게 만들었습니다.

```
_id: ObjectId("5fd734920e45d34228c53f2a")
> time: Array
  name: "01035070295"
  password: "5178"
  seatNumber: 10
  createdAt: 2020-12-14T09:46:58.241+00:00
  updatedAt: 2020-12-14T09:46:58.241+00:00
  __v: 0
```

위의 사진은 mongoDB Compass에서 값을 추가한 것입니다. 이렇게 각각의 값들이 추가가 되는 것을 알 수 있습니다.



Postman을 이용하여, 직접 axios로 연결되어 데이터 값을 넣어 볼 수 있습니다. 위의 값을 postman으로 POST를 하면, mongoDB Compass에 추가가 됩니다.

3. 프로젝트 결과

사용법 자리 현황 자리 예약하기

카페 / 스터디카페 사용법에 대해 설명합니다.

현재 페이지는 사용법 페이지입니다.

자리 현황에서는 현재 예약된 자리, 그리고 예약 번호를 알 수 있습니다. 자리 수정과 자리를 삭제하는 것도 가능합니다.

자리 예약하기는 자리를 예약하는 페이지입니다.

예약번호, 비밀번호, 자리, 이용권을 선택하고 자리를 예약할 수 있습니다.

./final_project에서 npm start를 하면 프론트엔드가 실행됩니다. ./final_project/server에서 node backend.js를 실행하면 백엔드도 실행됩니다. Npm start해서 포트 8080으로 들어가면 localhost:8080/main에서 이 화면을 확인할 수 있습니다. 여기에서 사용법을 확인할 수 있습니다. 저 화면에서 자리 예약하기를 누르면

사용법 자리 현황 자리 예약하기

자리 등록

전화번호

비밀번호

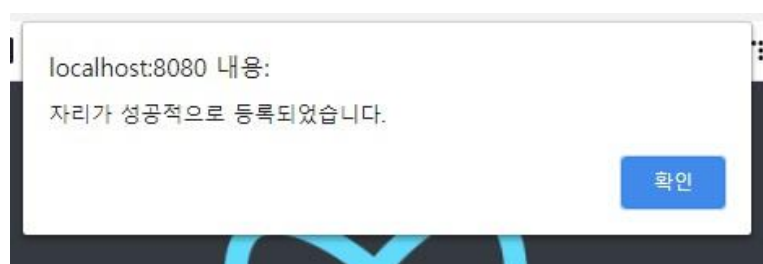
좌석번호

Time

자리 추가

취소

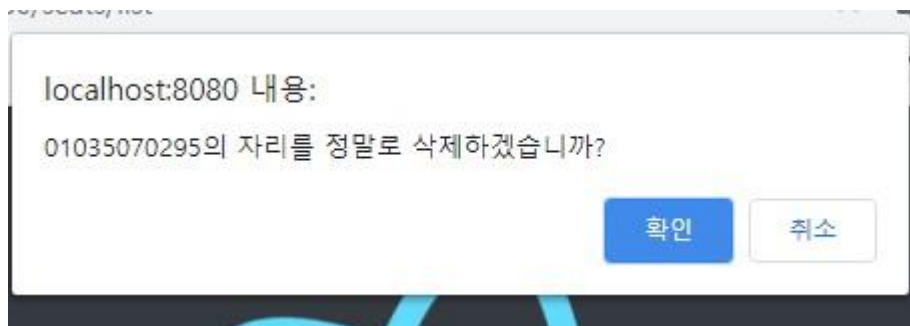
이와 같이 자리를 등록할 수 있는 웹 사이트가 나옵니다. 적절하게 전화번호, 비밀번호, 좌석번호, time을 설정해줄 수 있습니다.



이와 같이 자리를 추가하면 자리가 성공적으로 등록되었다는 알람이 뜨게 됩니다.

사용법 자리 현황 자리 예약하기				
Name	Seat	Time	삭제	수정
01035070295	10		자리 삭제	자리 수정
01021070595	17	2020-12-14 21:57	자리 삭제	자리 수정
Previous Page 1 of 1 10 rows Next				

자리 현황으로 가보면, 테이블에 내가 추가했던 값이 나오는 것을 확인할 수 있습니다. Name과 Seat의 윗부분에는 자신의 테이블을 쉽게 찾을 수 있게, 값을 넣어서 필터링을 할 수 있습니다.



테이블에서 자리 삭제를 누르면 삭제하겠냐는 알람이 뜨게 됩니다. 여기서 확인을 누르면 삭제가 되고 취소를 누르면 취소가 됩니다.

사용법 자리 현황 자리 예약하기	
자리 수정	
전화번호	
<input type="text" value="01035070295"/>	
비밀번호	
<input type="text"/>	
좌석번호	
<input type="text" value="10"/>	
Time	
<input type="text"/>	
자리 수정	취소

자리 수정을 누르면 자리 수정을 할 수 있는 곳으로 돌아오게 됩니다. 여기서 전화번호를 변경해서 타인에게 자신의 좌석번호를 양도할 수 있습니다. 다만, MongoDB에 저장되어 있는 비밀번호를 정확히 알고 있어야 자리 수정을 할 수 있습니다. 전화번호뿐만 아니라 좌석번호도 바꿀 수 있습니다. 이 경우는 자신이 자리가 마음에 들지 않아, 자리를 바꾸고 싶을 때 수정할 수 있는 것입니다.

프로젝트를 진행하면서 나왔던 해결하기 어려웠던 것들 중 하나는, 자리를 추가하거나 수정할 때, MySQL의 Key와 같은 중복되면 안되는 값들에 대한 처리였습니다. 전화번호를 사실상 ID처럼 사용하고, 좌석번호도 겹쳐서 앓을 수는 없기 때문에 이에 대해 `café_model.js`에서 `unique:true`를 이용해서 문제를 해결 할 수 있었습니다. 두번째로 어려웠던 것은 자리 수정 단계에서 전화번호와, 비밀번호가 같은지를 확인하는 문제였습니다. 이는 `getSeatById`를 이용해서 문제를 해결했습니다. `getSeatById`를 통해 id에 해당하는 값을 가져와서 이를 비밀번호에 입력한 값과 비교하면서 로그인과 같은 방식을 구현할 수 있었습니다.

4. 프로젝트 운영 환경 및 기타

본 프로젝트는 개인 노트북을 이용하고 Windows 10에서 프론트엔드와 백엔드를 모두 구현했습니다. 프론트엔드로는 리엑트를 사용했습니다. 또한 프론트엔드에 필요한 디자인을 위해 bootstrap을 적용했습니다. 백엔드에는 express를 이용하였고, node.js를 이용했습니다. 데이터베이스는 MongoDB를 사용해서 express에서 mongoose를 이용해서 MongoDB와 express를 연결했습니다.