

개인기여내용

백엔드 구조 구상 및 환경 구현 보조

- GraphQL, aws amplify, DynamoDB 등의 서비스를 조사.
- 데이터베이스를 앱 특성에 맞게 RDS로 확정.
- serverless 구조로 구축하기로 결정하고 백엔드 구조를 확정.

RDS에 데이터베이스 구축 보조

- 데이터 수집 파트와 논의하여 수집한 데이터베이스 테이블 확정.
- DBeaver를 통해서 primary key, 데이터 정렬방식 등을 지정.
- 회원가입 데이터베이스 틀을 완성하고, 추후 프론트 요구에 따라 수정 완료.

데이터베이스의 값을 client로 보내는 함수 및 구조 구축

- 주어진 과방 및 동방 번호에 따라, 그 요일의 하루치 측정데이터를 반환하는 함수 작성
- api gateway를 통하여 과방 및 동방 번호를 url 형태의 json 타입으로 받아오고, json타입의 이차원 배열로 결과값 반환
- GET 메서드를 사용하여 측정데이터를 받아올 수 있도록 api gateway 설정을 완료하고 배포

회원정보를 데이터베이스에 저장하는 함수 및 구조 구축

- 주어진 회원정보를 받아서 데이터베이스에서 아이디가 중복인지 확인하고, 중복이 아닌 경우 데이터를 DB에 저장하는 함수 작성.
- api gateway를 통하여 주어진 회원정보를 body에 넣어서 받아오고, 아이디 중복에 따른 회원가입 성공 여부 반환
- POST 메서드를 사용하여 회원정보를 보낼 수 있도록 api gateway 설정을 완료하고 배포

로그인용 함수 작성

- 주어진 아이디와 비밀번호를 받아서 데이터베이스에서 아이디와 비밀번호가 맞는지 확인하고, 확인되면 유저 데이터를 반환하는 함수 작성.
- api gateway를 통하여 주어진 로그인 정보를 body에 넣어서 받아오고, 아이디 존재 여부, 비밀번호 일치 여부에 따른 로그인 성공 여부와 유저데이터 반환
- POST 메서드를 사용하여 로그인 정보를 보낼 수 있도록 api gateway 설정을 완료하고 배포

배운 점

조별 과제를 통하여 협업을 위해 필요한 소통 능력과,
백엔드 구축을 위해 필요한 aws 서비스 사용 방법,
그리고 데이터베이스 구축 및 접근 방법에 대해 배울 수 있었고,
api 배포 및 데이터 전송 방법에 대해 확실하게 알 수 있었다.
그 외에도 백엔드를 작업하면서 신경써야 할 부분들,
프론트엔드와 어떤 점을 미리 의논해서 확정지어야 하는지,
문제가 생겼을 때 어떻게 정보를 찾고 해결하면 좋은지,
기획 부분에서 백엔드 파트에 대해서 무엇을 상의하면 좋은지 등등에 대해 배울 수 있었다.
처음에는 맨 땅에 헤딩하듯이 진행하였기에 내가 맡은 파트가 과연 구현이 되기는 하는지,
어떤 방식으로 진행해야 하는지 막막했었지만, 진행을 하면서 조사를 통해 방식을 알아가고,
수많은 시행착오를 겪으면서 기능을 온전하게 구현하는 동안 단순 지식으로서가 아니라
백엔드 개발 방법을 체득할 수 있었고, 자신감도 함께 쌓을 수 있었던 시간이 되었다.