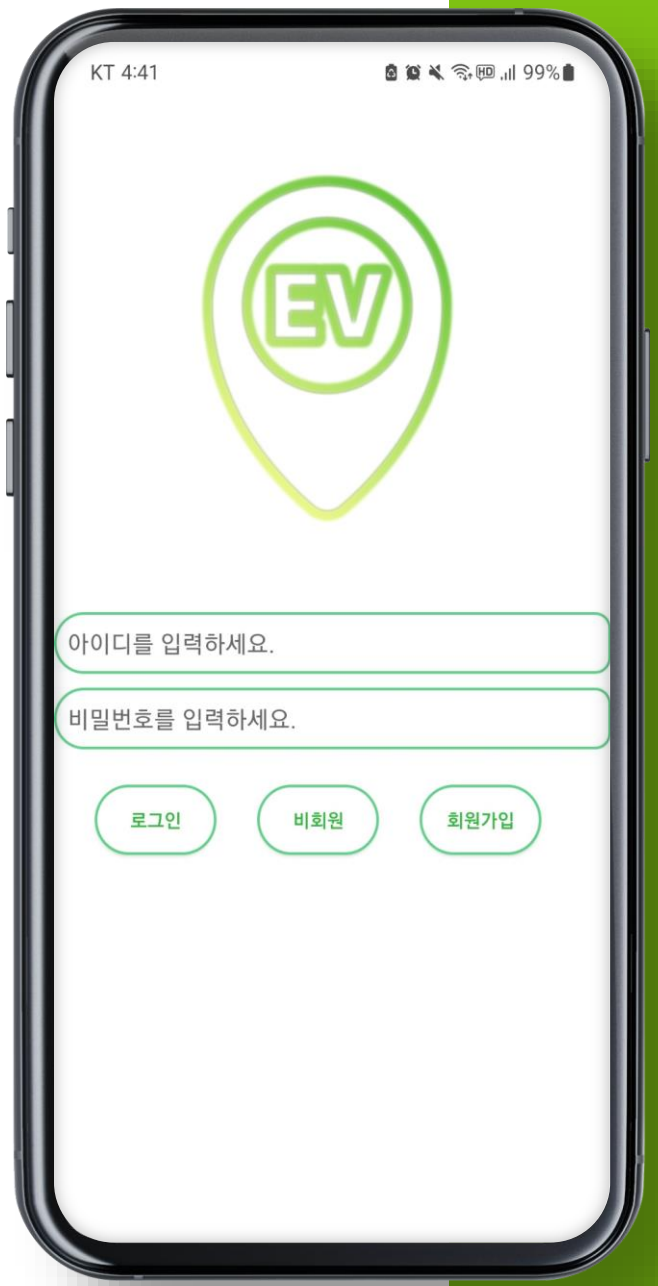
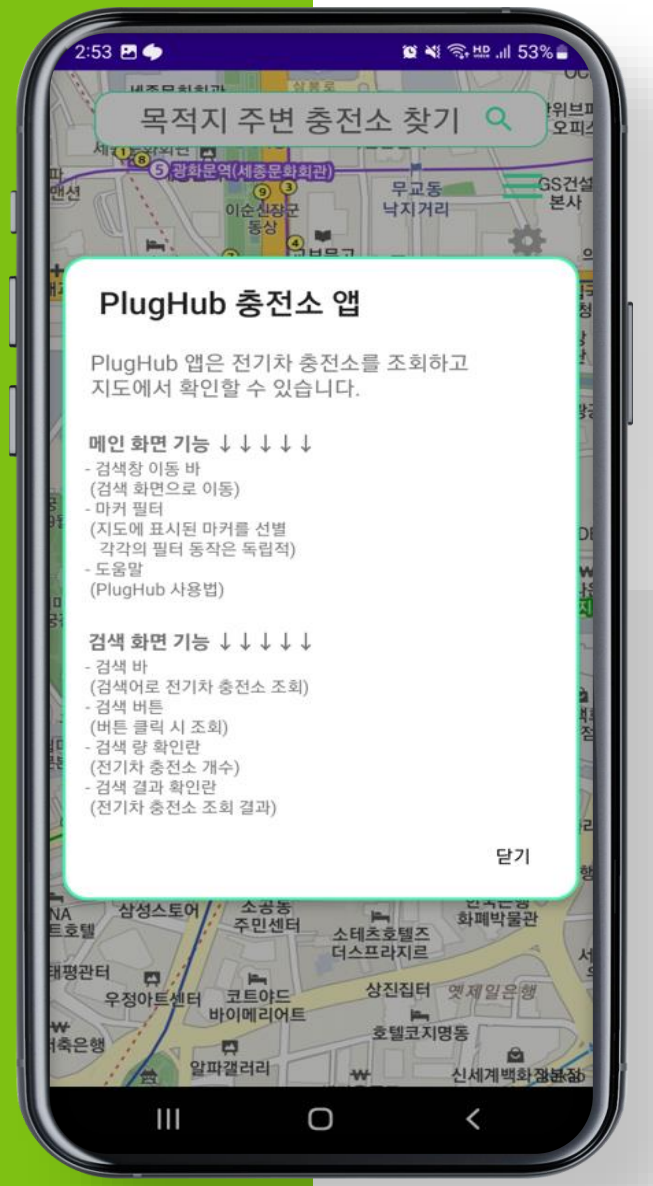
[산대특] 공공데이터
Open API기반의
앱 개발 과정



2023.05

PlugHub 앱 프로젝트

조장: 유민지 조원: 유민지, 오태현, 이규원



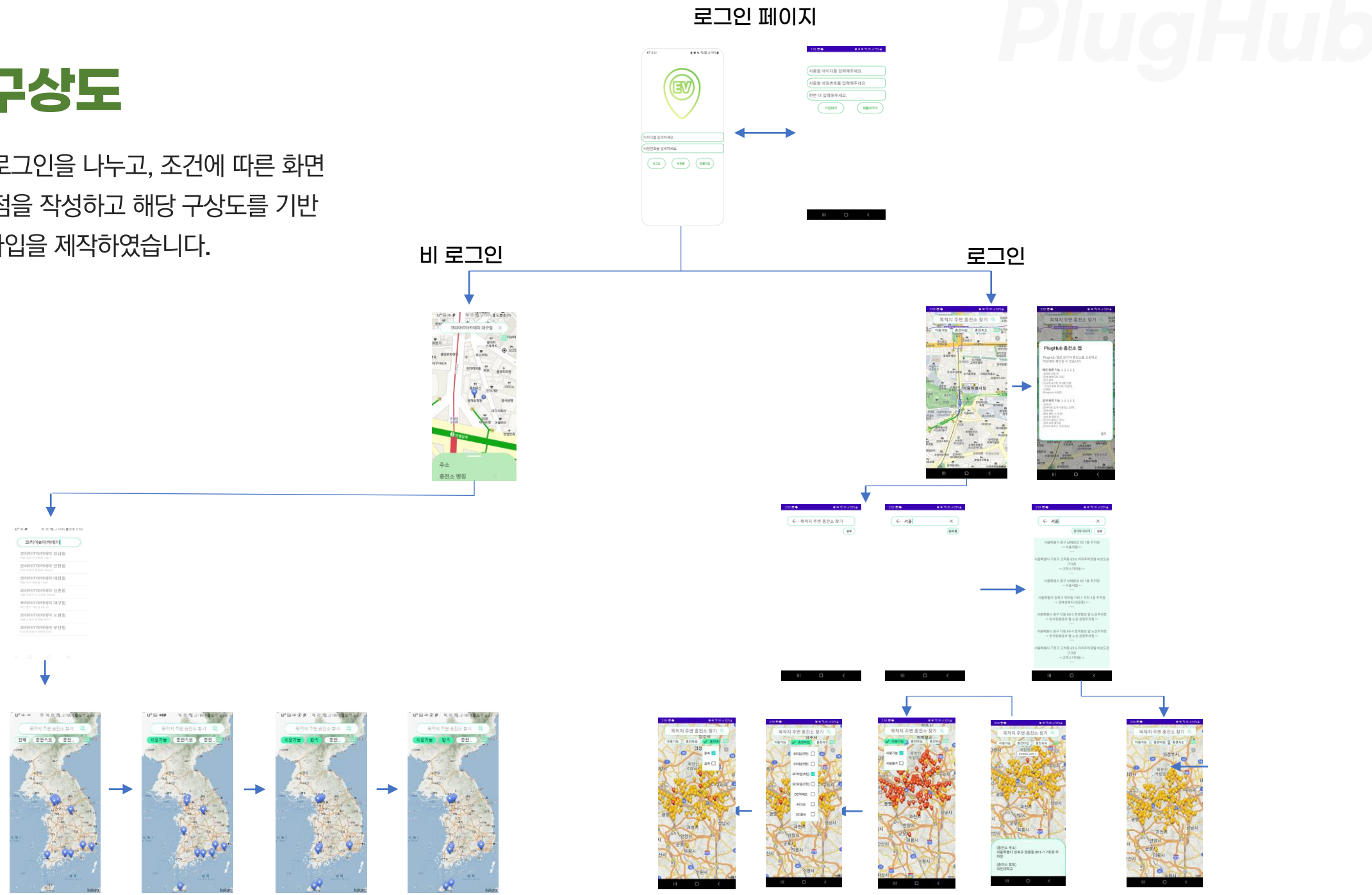
전기차 소유주 분들의 편의를 위한 **간단하고 빠른 충전소 위치 앱**

전기차의 특성들을 고려하여 다양한 정보를 제공할 것
사용하는 이용자들을 배려하여 깔끔하고 심플하게 만들 것
사용자의 편의를 높여 전기차의 수요를 높일 것

- ✓ 정확한 주소를 제공한다
- ✓ 다양한 위치 데이터를 제공한다
- ✓ 충전기 코드에 대한 데이터를 제공한다
- ✓ 사용 가능한 충전소와 가능 시간을 제공한다
- ✓ 충전 속도에 대한 데이터를 제공한다
- ✓ 로그인과 비 로그인에 차이를 주고 좀 더 다양한 내용을 제공한다

개발 구상도

로그인과 비 로그인을 나누고, 조건에 따른 화면 구성과, 차이점을 작성하고 해당 구상도를 기반으로 프로토타입을 제작하였습니다.



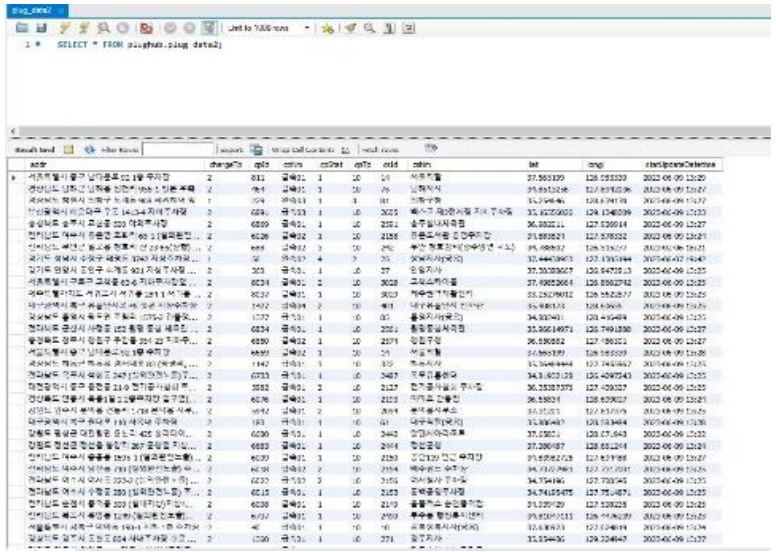
DB 구축 및 서버 구축

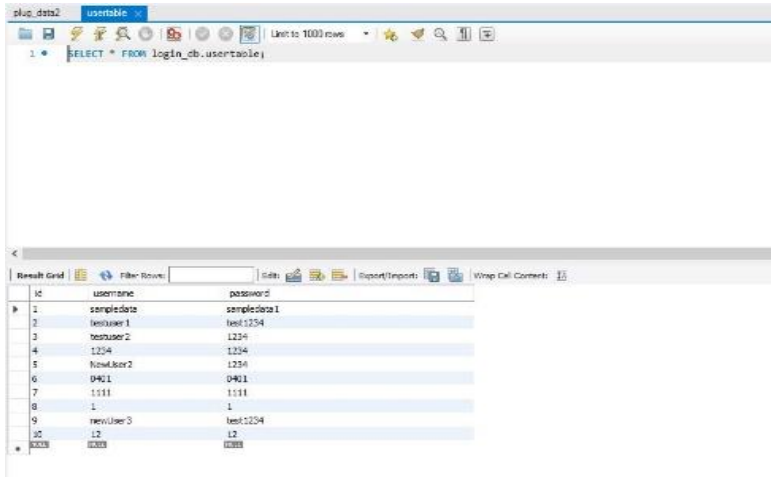
개발 구상도에 따라 로그인DB와 위치데이터DB를 구축하고 해당 DB와 연결 시키기 위해 추가적인 API를 작성, 해당 API 내에서 부족했던 부분은 팀원 분께서 보충 해주셨습니다.

```
// API 엔드포인트 정의
app.get('/plug_data2', (req, res) => {
  var using = parseInt(req.query.cpStat) || 0;
  var speed = parseInt(req.query.chargeTp) || 0;
  var charge = parseInt(req.query.cpTp) || 0;

  var query = "SELECT * FROM plug_data2";
  var where = [];
  var params = [];

  if (using != 0) {
    where.push('cpStat = ?');
    params.push(using);
  }
  if (speed != 0) {
    where.push('chargeTp = ?');
    params.push(speed);
  }
  if (charge != 0) {
    where.push('cpTp = ?');
    params.push(charge);
  }
  if (where.length > 0) {
    query += ' WHERE ' + where.join(' AND ');
  }

  console.log(query);

  // MySQL 데이터베이스에서 데이터 조회
  connection.query(query, params, (error, results) => {
    if (error) {
      console.error('Failed to fetch data from MySQL:', error);
      res.status(500).json({ error: 'Failed to fetch data from MySQL' });
    } else {
      console.log('Fetched data from MySQL:', results);
      res.json(results); // 데이터베이스 결과를 JSON 형식으로 응답
    }
  });
});
```

```
router.post('/login_process_mobile', function (request, response) {
  const id = request.body.id;
  const pw = request.body.pw;

  if (id && pw) {
    db.query(
      'SELECT * FROM usertable WHERE username = ? AND password = ?',
      [id, pw],
      function (error, results, fields) {
        if (error) {
          console.error('Database query error:', error);
          response.statusCode = 500;
          response.json({ result: false, error: 'An error occurred' });
          return;
        }

        if (results.length > 0) {
          response.statusCode = 200;
          request.session.is_logined = true;
          request.session.nickname = id;
          request.session.save(function (error) {

            if (error) {
              console.error('Session save error:', error);
              response.statusCode = 500;
              response.json({ result: false, error: 'An error occurred' });
            } else {
              response.json({ result: true });
```

```
router.post('/login_process_mobile', function (request, response) {
  const id = request.body.id;
  const pw = request.body.pw;

  if (id && pw) {
    db.query(
      'SELECT * FROM usertable WHERE username = ? AND password = ?',
      [id, pw],
      function (error, results, fields) {
        if (error) {
          console.error('Database query error:', error);
          response.statusCode = 500;
          response.json({ result: false, error: 'An error occurred' });
          return;
        }

        if (results.length > 0) {
          response.statusCode = 200;
          request.session.is_logined = true;
          request.session.nickname = id;
          request.session.save(function (error) {

            if (error) {
              console.error('Session save error:', error);
              response.statusCode = 500;
              response.json({ result: false, error: 'An error occurred' });
            } else {
              response.json({ result: true });
```





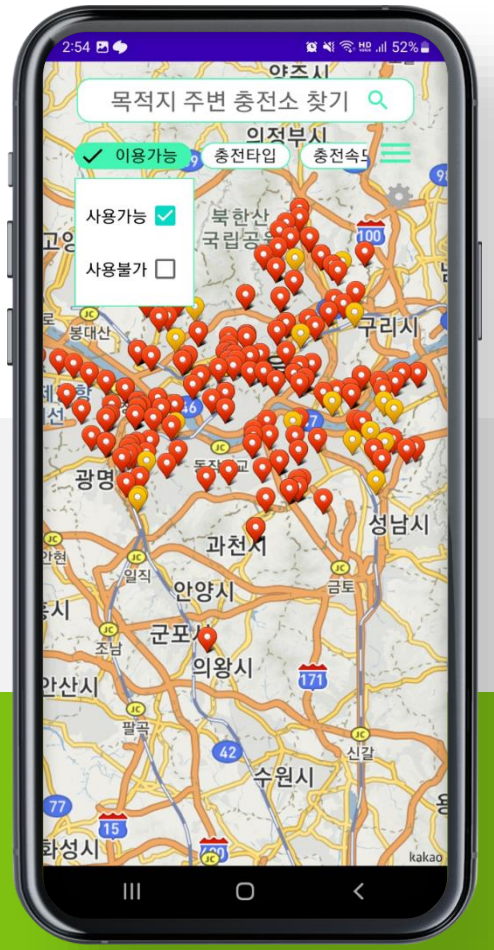
Step 1

첫 화면으로 로그인 페이지를 띄워
비 로그인과 로그인을 분리하고
로그인에 따른 화면을 다르게 출력

Step 2

비 로그인은 데이터베이스가 기반, 마커 생성 후
카카오 API를 이용, 검색어를 조회하고
원하는 위치의 마커를 사용자에게 표출

Step 3



로그인은 검색 시 모든 충전소 정보와 함께
마커 생성 가능, 필터 사용시 해당 필터에
만족할 경우에 컬러로 변화를 줌.

비 로그인

데이터베이스를 기반

OPEN API의 정보를 크롤링 하여
데이터베이스에 추가한 정보를
사용함으로서 실시간 확인이 어려움

필터를 사용하여 선별

정보에 기반한 선별이 아닌
필터를 사용하여 마커 선별이 가능

마커의 개수 제한

최대 출력 개수가 10개로 고정,
많은 양의 정보를 보기가 어려움

정보의 범위 제한

지도를 축소하였을 때 이용자의
주변에 위치한 10개의 충전소만 출력

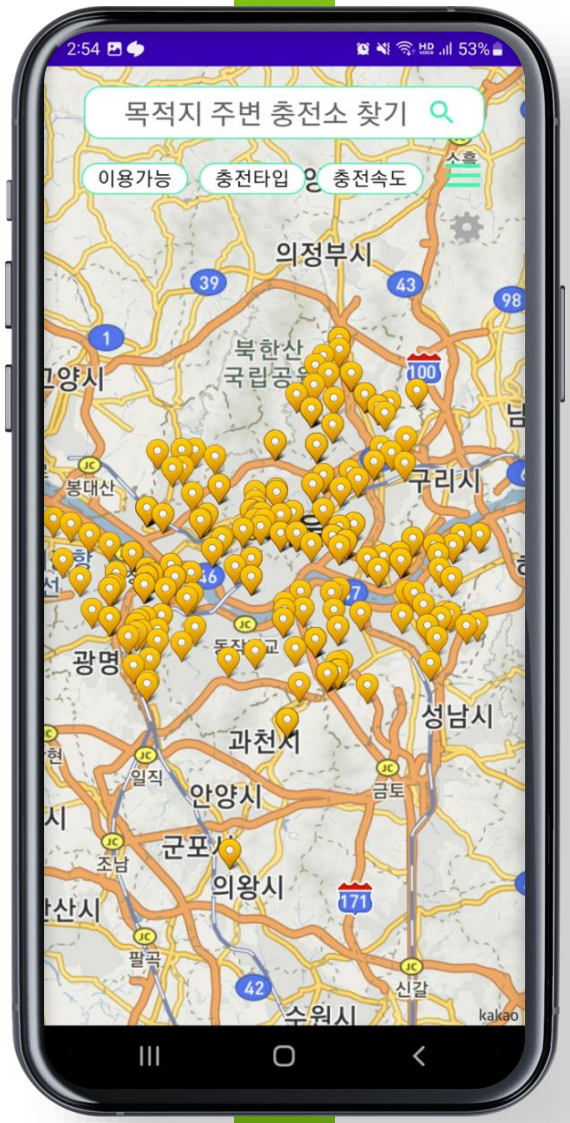
로그인

공공데이터포털을 기반

공공데이터포털을 사용하여
카카오 서버와 관계 없이
원하는 정보 확인

정보를 사용하여 선별

정보에 기반한 필터사용으로
해당하는 마커에 대해 색깔의
차이를 확인할 수 있음



마커의 개수 제한없음

공공데이터포털의 데이터를
기반하기에 더 많은 양의 마커와
충전소의 정보를 확인

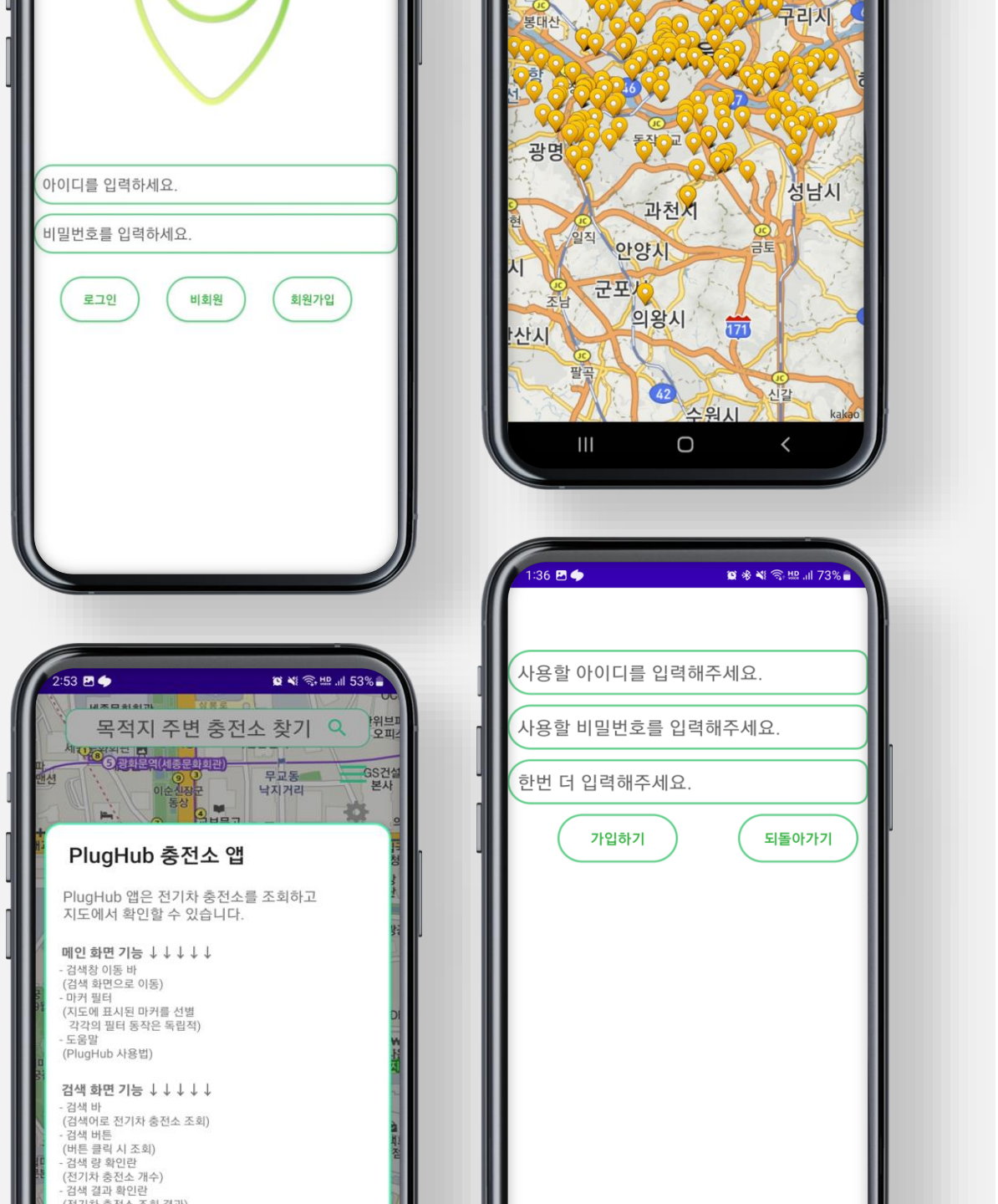
정보의 범위 제한없음

지도를 축소하였을 때 이용자의
주변이 아닌 전국에 위치한
충전소 정보를 제한없이 출력가능

프로젝트를 마치며

짧지 않은 기간 동안 수많은 검색을 통해 다양한 방법을 알게 되었고, 덕분에 많은 공부가 되었습니다.

수업의 일환으로 시작된 프로젝트 이나, 이로 인해 취업에 있어 어떤 방식으로 업무가 진행되는 것인지 간단하게 알 수 있는 계기가 되었습니다.



PlugHub

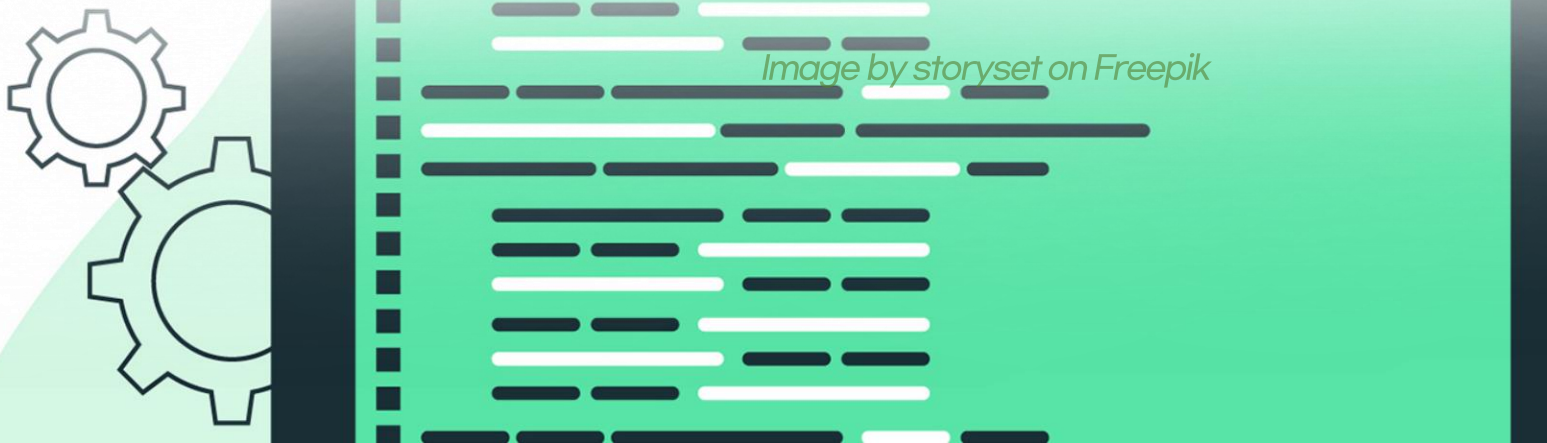
로그인 파트

프론트 엔드 오태현

바텀시트 및 마커 정보 연동, 메인 지도 화면과
필터 기능으로 마커 선별, 검색 리스트 화면 담당

처음 프로젝트를 진행하면서 복잡한 코드 작성에
시행착오가 있었지만, 코드를 간략화하기 위해
내장함수를 사용하는 방법을 발견했습니다.

검색을 통해 코드를 작성하면서 구현 방법과
작성 방향성을 익힐 수 있었습니다



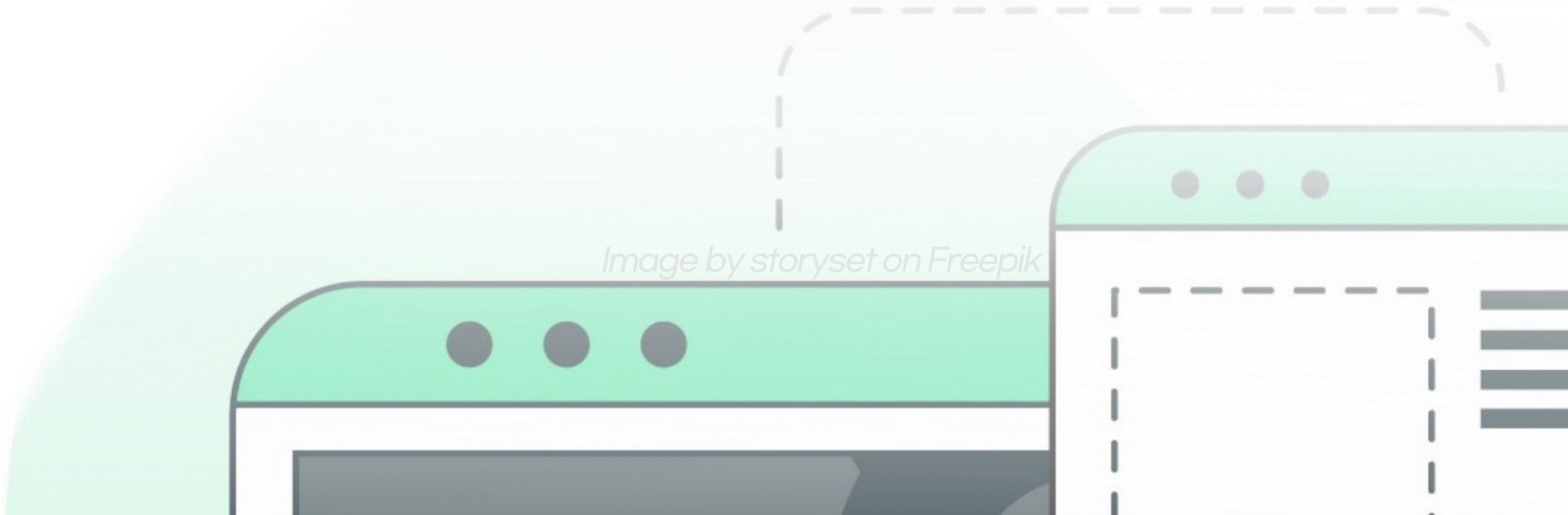
비 로그인 파트

프론트 엔드 이규원

검색 뷰, 옵션에 따른 마커 삭제, 바텀시트 연동, 서버의 추가 API

리스트를 클릭했을 때, 값을 메인 화면으로 전달해주는
otto라는 라이브러리를 알게되었고,

네트워크로부터 전달된 데이터를 우리 프로그램에서
필요한 형태의 객체로 받을 수 있는 Retrofit
라이브러리를 공부할 수 있는 좋은 계기가 되었습니다.



서버 파트

백엔드 유민지

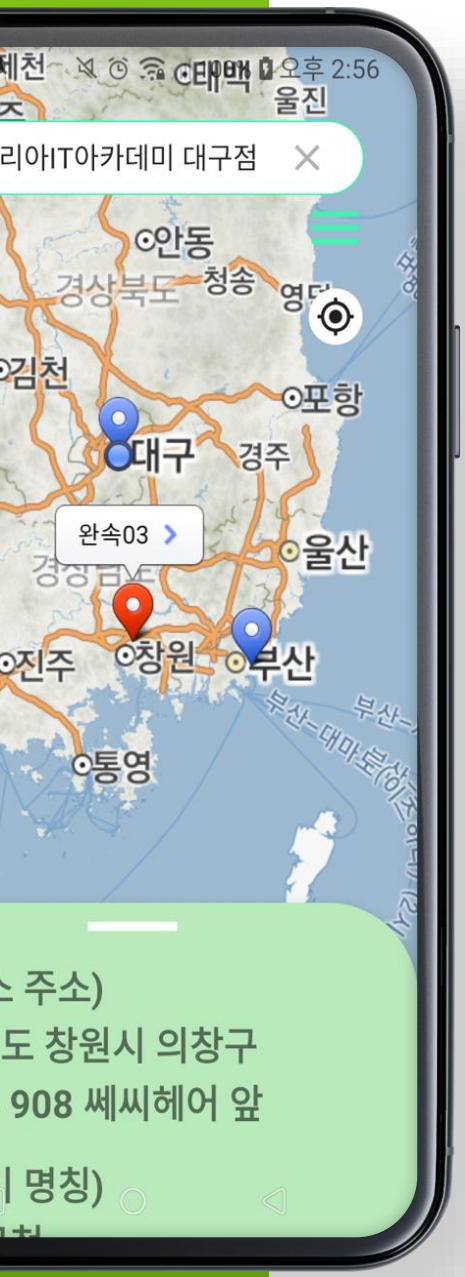
DB 및 서버 구축, REST API, Retrofit API 담당

처음에는 서버를 배우고 있는 상황이었기 때문에 Express를 사용해서
구축하고 연결만 하면 된다고 생각하고 서버 역할을 맡게 되었으나,

서버 라는 것이 단순히 연결하고 출력해주는 역할이 아니라
맵의 API나 로그인을 요청 하는 등, 수많은 일들이 함께
있는 것이라는 걸 깨달았습니다.

또한, 수많은 오류들을 마주하고, 많은 검색들을 통해 느낀 것은
담당하지 않는 부분이라고 가볍게 넘길 것이 아니라

기회가 닿는다면 그런 부분도 공부를 하면서 많은 분들과
협업함에 있어 도움이 될 수 있었으면 좋겠다고 느꼈습니다



팀원별 업무 내용 및 포지션

프론트엔드	오태현	로그인에 대한 프론트 구현
프론트엔드	이규원	비 로그인에 대한 프론트 구현
백엔드	유민지	서버 및 API 담당, 간단한 디자인 구현
추가적인 업무 내용		
서버 API	이규원	서버 API에서 부족한 부분 보충
자료 수집	오태현	부족한 자료 보충
조장	유민지	발표 및 자료 총합

감사합니다.

2023.06