**프로젝트 진행 보고서(3주차)**

**\* 프로젝트를 위해서 진행한 내용을 간단하게 정리**

1. 기능 구현 목록

|  |
| --- |
| - 현재 위치 불러오기 버튼 기능  - 위치 검색 후 해당 위치 날씨 값 불러오기 기능 (search.dart)  - 메인 화면에 미세먼지 정보 불러오기 및 UI/UX 보충 작업 (weather\_screen)  - 현재 날짜 및 시간 보여주기 |

⦁screens/weather\_screen.dart

|  |
| --- |
| …  …  …  import 'package:google\_fonts/google\_fonts.dart'; //폰트  import 'package:flutter\_svg/flutter\_svg.dart'; // svg 사용하기 위함  import 'package:intl/intl.dart'; // DateTime Format  import 'package:meonji/screens/login.dart';  import 'package:timer\_builder/timer\_builder.dart'; // 시스템 시간 불러오기 위함  import 'package:meonji/model/model.dart'; // condition에 따른 svg 표시 조건문  import 'package:firebase\_auth/firebase\_auth.dart'; // 사용자 등록/인증 관련  import 'package:meonji/data/my\_location.dart'; // 위치정보관련  import 'package:meonji/data/network.dart'; // 날씨데이터 관련  import 'package:meonji/api/key.dart'; // API Key  import 'package:meonji/screens/search.dart'; // 위치 검색 페이지  …  …  …  // 빌더를 전달받음  class WeatherScreen extends StatefulWidget {  …  …  …  dynamic parseWeatherData; // 생성자 선언을 위한 날씨 정보 파싱    dynamic parseAirPollution; // 생성자 선언을 위한 미세먼지 정보 파싱  …  …  …  // 날씨를 텍스트로 표시하는 부분  class \_WeatherScreenState extends State<WeatherScreen> {  …  …  …  late Widget icon; // 날씨 아이콘    late String des; // 날씨 설명    late Widget airIcon; // 미세먼지 아이콘    late Widget airState; // 미세먼지 설명    var date = DateTime.now(); // 오늘 날짜    late double pm10; // 미세먼지 (PM10)    late double pm2\_5; // 초미세먼지 (PM2.5)    final \_authentification =    late double currentLatitude; // 최근 위도 변수 선언    late double currentLongitude; // 최근 경도 변수 선언  …  …  …  // 날씨 업데이트    void updateData(dynamic weatherData, dynamic airData) {      Model model = Model(); // 날씨 Condition Model 객체      double temp2 = weatherData['main']['temp']; // 온도      int condition = weatherData['weather'][0]['id']; // 날씨 id      int index = airData['list'][0]['main']['aqi']; // 미세먼지 단계 (ex. 1,2,3,4,5...)      des = weatherData['weather'][0]['description']; // 날씨 설명      pm10 = airData['list'][0]['components']['pm10']; // 미세먼지 수치 불러오기      pm2\_5 = airData['list'][0]['components']['pm2\_5']; // 초미세먼지 수치 불러오기      temp = temp2.round(); // 반올림      cityName = weatherData['name']; // 도시명      icon = model.getWeatherIcon(condition); // 날씨 Condition 불러오기      airIcon = model.getAirIcon(index); // 미세먼지 아이콘 불러오기      airState = model.getAirCondition(index); // 미세먼지 상태 불러오기      print('날씨 불러오기 완료');    }    // 시스템 시간 불러오기 함수    String getSystemTime() {      var now = DateTime.now();      return DateFormat("hh:mm a").format(now);    }    // 현재 위치로 업데이트    void getCurrentLocation() async {      print('날씨 업데이트를 시작합니다.');      Mylocation myLocation = Mylocation();      await myLocation.getMyCurrentLocation();      currentLatitude = myLocation.latitude2; // 위도 변수 삽입      currentLongitude = myLocation.longitude2; // 경도 변수 삽입      print(currentLatitude);      print(currentLongitude);      Network network = Network(          'https://api.openweathermap.org/data/2.5/weather?lat=$currentLatitude&lon=$currentLongitude&appid=$apiKey&units=metric',          'https://api.openweathermap.org/data/2.5/air\_pollution?lat=$currentLatitude&lon=$currentLongitude&appid=$apiKey');      var weatherData = await network.getJsonData(); // api값을 그릇에 담기 (날씨 정보)      print(weatherData);      var airData = await network.getAirData(); // api값을 그릇에 담기 (미세먼지 정보)      print(airData);      widget.parseWeatherData = weatherData; // 새로운 날씨 데이터 삽입      widget.parseAirPollution = airData; // 새로운 공기 데이터 삽입      print('새로운 날씨, 공기 데이터 업데이트 완료');    }  @override    Widget build(BuildContext context) {  ================이하 코드부터는 아래 주소에서 확인해주세요!  https://github.com/SoonMinKwun/history\_of\_studying/blob/main/flutter/meonji/lib/screens/weather\_screen.dart |

⦁screens/search.dart

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:meonji/screens/weather\_screen.dart';  import 'package:meonji/data/network.dart'; // 날씨데이터 관련  import 'package:meonji/api/key.dart'; // API Key  import 'package:meonji/data/my\_location.dart'; // 위치정보관련  class Search extends StatefulWidget {    const Search({Key? key}) : super(key: key);    @override    State<Search> createState() => \_SearchState();  }  class \_SearchState extends State<Search> {    // 지역 텍스트 컨트롤러    TextEditingController stateController = TextEditingController();    late String state; // 지역명    late double currentLatitude; // 최근 위도 변수 선언    late double currentLongitude; // 최근 경도 변수 선언    // 지역 가져오기    void searchLocation() {      setState(        () {          state = stateController.text;          print('$state 의 날짜를 가져옵니다...');        },      );    }    // 입력한 지역의 날씨, 공기 데이터 가져와서 보내주기    void sendLocation() async {      Mylocation myLocation = Mylocation();      await myLocation.getMyCurrentLocation();      currentLatitude = myLocation.latitude2; // 위도 변수 삽입      currentLongitude = myLocation.longitude2; // 경도 변수 삽입      print(currentLatitude);      print(currentLongitude);      Network network = Network(          'https://api.openweathermap.org/data/2.5/weather?q=$state&appid=$apiKey&units=metric',          'https://api.openweathermap.org/data/2.5/air\_pollution?lat=$currentLatitude&lon=$currentLongitude&appid=$apiKey');      var weatherData = await network.getJsonData(); // api값을 그릇에 담기 (날씨 정보)      print(weatherData);      var airData = await network.getAirData(); // api값을 그릇에 담기 (미세먼지 정보)      print(airData);      // weather\_screen으로 빌더를 넘기고 weatherData값 전달해서 라우트      Navigator.push(        context,        MaterialPageRoute(          builder: (context) {            return WeatherScreen(                parseWeatherData: weatherData, parseAirPollution: airData);          },        ),      );    }    @override    Widget build(BuildContext context) {      return Scaffold(        extendBodyBehindAppBar: true,        appBar: AppBar(          backgroundColor: Colors.transparent, // 배경 없애기          elevation: 0.0,          leading: IconButton(            icon: Icon(Icons.arrow\_back\_ios\_new\_rounded),            onPressed: () {              Navigator.pop(context);            },            iconSize: 40.0,          ),        ),        // 키보드 없애기 기능        body: Builder(          builder: (context) {            return GestureDetector(              onTap: () {                FocusScope.of(context).unfocus();              },              // ignore: avoid\_unnecessary\_containers              child: Container(                child: Stack(                  children: [                    Image.asset(                      'image/background.jpg',                      fit: BoxFit.cover, // 배경 늘리기                      width: double.infinity, // 여백 없애기                      height: double.infinity, // 여백 없애기                    ),                    // 검색창 컨테이너                    Container(                      padding: EdgeInsets.all(20.0),                      margin: EdgeInsets.all(20.0),                      child: Form(                        child: Column(                          crossAxisAlignment: CrossAxisAlignment.center,                          children: [                            SizedBox(                              height: 50.0,                            ),                            // 지역 입력 텍스트 폼                            TextField(                              controller: stateController, // 컨트롤러 배정                              decoration: InputDecoration(                                hintText: '지역을 입력하세요 :)',                                filled: true,                                fillColor: Colors.white,                                enabledBorder: UnderlineInputBorder(                                  borderSide:                                      BorderSide(color: Colors.blue, width: 2.0),                                ),                                focusedBorder: UnderlineInputBorder(                                  borderSide:                                      BorderSide(color: Colors.blue, width: 3.0),                                ),                              ),                            ),                            // 검색하기 버튼 폼                            TextButton.icon(                              onPressed: () {                                searchLocation();                                sendLocation();                              },                              icon: Icon(                                Icons.location\_searching,                              ),                              label: Text('검색하기'),                            ),                          ],                        ),                      ),                    ),                  ],                ),              ),            );          },        ),      );    }  } |

위에 있는 코드는

<https://github.com/SoonMinKwun/history_of_studying/tree/main/flutter/meonji> 에서도 볼 수 있습니다.

- 관련 기술

geolocator : <https://pub.dev/packages/geolocator>

http : <https://pub.dev/packages/http>

convert : <https://pub.dev/packages/convert>

material : <https://api.flutter.dev/flutter/material/material-library.html>

OpenWeather API : <https://openweathermap.org/api>