

2025학년도 모바일프로그래밍 프로젝트 완료 보고서

친환경 실천 체크인 앱 (EcoCheckInApp)

2025년 6월 20일

컴퓨터정보과

팀명	루틴어스
팀장	조예원(202344064)
팀원	1. 김민재(202144046) 2. 투브신(202388004) 3. 용다인(202344043)



2025년도
컴퓨터정보과 모바일프로그래밍 프로젝트
수행 결과보고서
(분야 : Android 기반의 앱 개발)

연구과제명 : 친환경 실천 체크인 앱(EcoCheckInApp)

2025. 6. 20



본 과제(결과물)는 인하공업전문대학 컴퓨터정보과 모바일프로그래밍 교과목
의 프로젝트 완료보고서입니다.

제 출 문

- 분야 : Android 기반의 소프트웨어 개발
- 연구과제명 : 친환경 실천 체크인 앱(EcoCheckInApp)
- 프로젝트 책임자 : 인하공업전문대학 컴퓨터정보과 조예원

2025년도 인하공업전문대학 컴퓨터정보과
모바일프로그래밍 교과목의 프로젝트 결과물로 이 보고서를 제출합니다.

2025. 6. 20

프로젝트 책임자 : 컴퓨터정보과(202344064) 조예원 (인)

공동 참여자 : 컴퓨터정보과(202144046) 김민재 (인)
컴퓨터정보과(202388004) 투브신 (인)
컴퓨터정보과(202344043) 용다인 (인)

인하공업전문대학 컴퓨터정보과 학과장 귀하

프로젝트 요약문

분야	Android 기반의 소프트웨어 개발
프로젝트명	친환경 실천 체크인 앱(EcoCheckInApp)
프로젝트 책임자	컴퓨터정보과(202344064) 조예원

연구 내용

본 프로젝트는 사용자의 친환경 행동을 기록하고 공유함으로써 자발적인 환경 보호 실천을 유도하는 Android 기반 "친환경 실천 체크인 앱(EcoCheckInApp)"의 개발을 목적으로 진행되었다. 사용자는 일상 속에서 실천한 친환경 활동을 위치 기반으로 체크인하고, 이를 지도에 시각화하여 다른 사용자와 공유할 수 있으며, 이 과정을 통해 상호 동기 부여 및 지역사회 친환경 문화 조성에 기여하고자 하였다.

우선, Firebase Database를 활용하여 사용자 회원가입과 로그인 기능을 구성하였다. 위치 기반 실천 기록은 Google Maps API와 FusedLocationProvider를 활용해 사용자의 현재 위치를 받아오고, 입력한 장소명과 메모를 함께 체크인 기록으로 저장하였다. 각 기록은 EcoActionModel을 통해 데이터 모델로 관리되며, Firebase Realtime Database에 구조적으로 저장되었다.

사용자는 체크인 외에도 새로운 친환경 장소를 등록할 수 있으며, 등록된 장소는 지도에 마커로 표시되어 누구나 확인할 수 있다. 장소 등록 시에는 이름, 카테고리, 설명, 좌표 정보 등을 입력하며, 등록자의 ID도 함께 저장되어 데이터 관리의 명확성을 높였다.

또한, 마이페이지 기능을 통해 사용자는 자신의 실천 내역과 등록 장소 목록을 한눈에 확인할 수 있어 지속적인 실천을 장려하는 구조로 설계되었다. 실천 활동 외에, 커뮤니티 활성화를 위해 피드 기능도 구현하였다. 사용자는 사진과 텍스트로 구성된 게시글을 작성하여 친환경 팁이나 후기 등을 공유할 수 있으며, 이는 Database에 연동되어 저장된다.

개발에는 Android Studio와 Java를 사용하였으며, 외부 연동으로는 Firebase, Google Maps SDK, Glide 등을 활용하였다. 전체 시스템은 사용자, 장소, 체크인, 피드 데이터를 각각 독립적으로 관리하면서도 연동이 가능한 구조로 설계되어, 향후 유지보수와 확장성 측면에서도 안정적인 기반을 마련하였다.

이 앱은 단순한 실천 기록을 넘어 사용자 간 친환경 활동의 공유와 격려를 유도하는 플랫폼으로, 친환경 실천의 일상화를 위한 디지털 기반 모델로써 의의가 있다.

참여자 인적사항

1. 프로젝트 팀장

성명	조예원	학번	202344064
학과	컴퓨터정보과		
연락처	H.P	010-9120-9878	
	E-mail	q5556@naver.com	

2. 프로젝트 팀원

성명	김민재	학번	202144046
학과	컴퓨터정보과		
연락처	H.P	010-8512-0389	
	E-mail	202144046@itc.ac.kr	

성명	투브신	학번	202388004
학과	컴퓨터정보과		
연락처	H.P	010-8246-9396	
	E-mail	tuvshoo.enkule@gmail.com	

성명	용다인	학번	202344043
학과	컴퓨터정보과		
연락처	H.P	010-8899-7343	
	E-mail	202344043@itc.ac.kr	

목 차

1. 프로젝트 목적	1
2. 프로젝트 목표와 기대 효과	1
3. 프로젝트 진행 범위 및 방법	1
4. 프로젝트 주요 내용	2
5. 연구 결과물	3
5-1. 프로젝트명	3
5-2. 프로젝트 개요	3
5-3. 프로젝트 수행 세부 일정 및 내용	3
5-4. 프로젝트 결과	4

1. 프로젝트 목적

본 프로젝트는 사용자가 일상 속에서 친환경 루틴을 실천하고, 이를 기록 및 공유할 수 있는 Android 기반 모바일 애플리케이션을 개발하는 데 목적이 있다. 환경 문제의 심각성이 대두되는 가운데, 개개인의 행동 변화가 근본적인 해결책이 될 수 있다는 점에 착안하여, 사용자의 친환경 습관 형성과 실천을 도울 수 있는 시스템을 구축하였다. 특히, 지도 기반 실천 기록, 친환경 장소 공유, 실천 내역 통계, 커뮤니티 피드 기능을 통해 사용자 스스로 동기를 부여받고, 일상에서 지속 가능한 소비 행동을 실현할 수 있도록 설계하였다.

2. 프로젝트 목표 및 기대효과

프로젝트 목표

- 사용자의 친환경 행동을 실시간 위치 기반으로 기록할 수 있는 체크인 기능 구현
- 사용자가 알고 있는 친환경 장소(에코 핫플)를 지도 기반으로 등록하고 공유하는 기능 제공
- 개인 실천 이력을 통합하여 마이페이지에서 시각화된 형태로 제공
- 사용자 간 정보 공유를 위한 피드 및 게시글 작성 기능 구현
- Android Studio(Java) 기반의 앱 개발 및 Firebase Realtime DB, Google Maps SDK 활용

기대효과

- 개인의 친환경 실천 활동을 습관으로 형성하여 일상화 촉진
- 사용자 간 실천 사례 공유를 통한 지역 기반 친환경 커뮤니티 형성
- 실천 장소 및 행동 데이터의 집적을 통한 환경 캠페인 및 통계 자료 활용 가능성
- 청소년부터 일반 사용자까지 쉽게 접근 가능한 UX를 통한 참여 확대
- 장기적으로는 친환경 활동에 대한 사회적 관심과 참여를 높이는 기반 조성

3. 프로젝트 진행범위 및 방법

진행범위

- Android 기반 네이티브 앱 개발 (지원 OS: Android 10 이상)
- 사용자 체크인 기능, 친환경 장소 등록 및 필터링, 커뮤니티 피드 및 마이페이지 기능 포함
- Firebase Realtime Database 연동을 통한 실시간 데이터 저장 및 조회
- Google Maps API 연계를 통한 위치 기반 서비스 제공
- 관리자 권한 기능 및 데이터 구조 설계는 기획 범위로만 포함

개발 방법

- Android Studio와 Java를 기반으로 기능별 모듈화하여 개발
- MVP/단계별 개발 방식 및 기능 단위 테스트를 통해 개발 안정성 확보
- Firebase Realtime Database 구조 설계 및 마커 필터링 로직 구현
- RecyclerView, Intent, Glide, SharedPreferences 등 Android 구성 요소 적극 활용
- Google Maps 마커 클릭, 위치 허용/권한 처리 등 실시간 위치 기반 처리 구현

4. 프로젝트 주요 내용

1. 사용자 로그인

- 별도 서버 없이 사용자 입력 기반 로그인 구현

2. 체크인 기능

- 실시간 위치 기반으로 친환경 행동 기록 가능
- 장소명 및 메모를 함께 입력하여 Firebase DB에 저장
- EcoActionModel 구조를 사용하여 데이터 관리

3. 지도 기반 친환경 장소 등록 및 조회

- 사용자 직접 등록한 장소를 지도에 마커로 표시
- 마커 클릭 시 상세정보 출력
- 카테고리 필터링 기능 제공 (MapWithFilterFragment 활용)

4. 마이페이지(실천 기록 보관소)

- 사용자의 체크인 기록과 장소 등록 내역을 리스트 및 그리드로 확인
- 실천 이력을 효율적으로 관리 가능

5. 피드 게시글 기능

- 친환경 실천 후기나 팁을 게시글로 작성
- 제목, 본문, 사용자 ID 저장 (이미지 업로드 기능은 미구현)
- 게시글은 최신순으로 피드에 출력 (FeedFragment, FeedAdapter 활용)

5. 프로젝트 결과물

5-1. 프로젝트명

친환경 실천 체크인 앱(EcoCheckInApp)

5-2. 프로젝트 개요

본 프로젝트는 사용자의 친환경 행동을 기록하고 공유할 수 있는 Android 기반 모바일 앱을 개발하는 것이다. 사용자는 일상에서 실천한 친환경 활동을 위치 기반으로 체크인하고, 지도 위에 친환경 장소를 등록하거나 다른 사용자들과 실천 내용을 공유할 수 있다. 앱은 Firebase Realtime Database 와 Google Maps API를 연동하여 실시간 데이터 저장 및 지도 서비스를 제공하며, 마이페이지를 통해 개인 실천 내역을 관리할 수 있다. 이 프로젝트는 환경 보호와 디지털 기술을 결합하여 개인의 행동 변화와 지역사회 차원의 친환경 문화 조성을 동시에 목표로 한다.

5-3. 프로젝트 수행 세부일정 및 내용

■ 프로젝트 수행 세부 일정

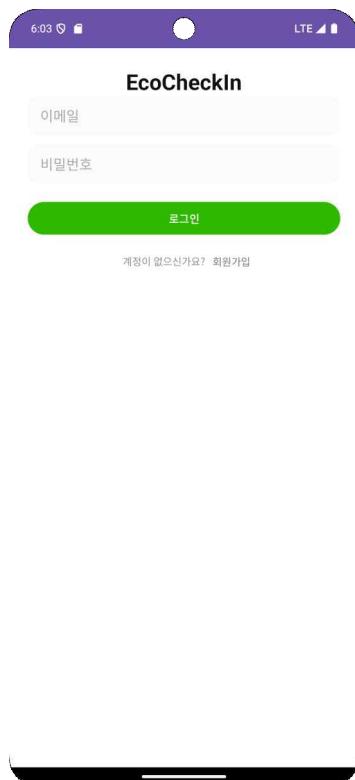
프로젝트 진행 계획(2025. 4. 21 ~ 2025. 6. 15)										
진 행 내 용	책임자	1	2	3	4	5	6	7	8	비고
프로젝트 기획 확정, 요구사항 도출	김민재	∨								
화면 설계 및 데이터베이스 구조 설계	용다인		∨							
Firebase 연동, 실시간 위치 기반 처리 구현	김민재			∨						
체크인 기능 개발 및 DB 저장 연동	조예원				∨					
친환경 장소 등록 기능, 마커 지도 출력	투브신					∨				
마이페이지 구현, 실천 기록 리스트/그리드 뷰 개발	조예원						∨			
커뮤니티 피드 및 게시글 작성 기능 구현	투브신							∨		
통합 테스트, 오류 수정, 사용자 메뉴얼 작성	용다인								∨	

■ 프로젝트 수행 내용

- Android(Java) 기반 친환경 실천 앱 기획 및 개발
- Firebase Realtime Database 기반 사용자·장소·체크인·게시글 데이터 처리 구조 설계
- Google Maps API를 이용한 위치 기반 마커 출력, 필터링, 마커 정보창 구성
- 사용자 친환경 활동(체크인) 기록 기능 개발 및 지도 기반 시각화 구현
- 에코 장소 등록 및 지도 마커 표시, 장소 카테고리별 필터 기능 제공
- 마이페이지 구성: 실천 기록 리스트 및 그리드 뷰 전환 기능 구현
- 피드 기능 구현: 게시글 작성/조회 기능 및 게시물 리스트 구성
- 사용자 간 데이터 연계 및 상호작용이 가능한 UX 설계

5-4. 프로젝트 결과

- 프로젝트 소스 (부록)
- 사용자 매뉴얼



EcoCheckIn 앱 실행 시 처음 나타나는 화면으로, 사용자가 앱을 이용하기 위해 로그인하는 기능을 제공한다.

이메일 입력

- 사용자가 가입 시 등록한 이메일 주소를 입력한다. 형식이 올바르지 않은 경우 로그인에 실패한다.

비밀번호 입력

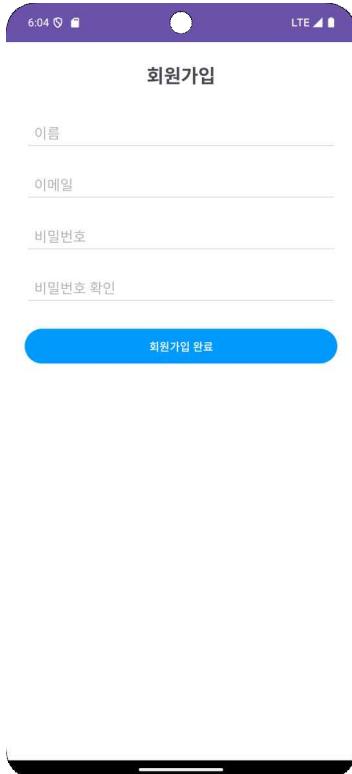
- 사용자 비밀번호를 입력한다. 입력 시 보안 처리되어 화면에는 ●●● 형태로 표시된다.

[로그인] 버튼

- 이메일과 비밀번호가 정확히 입력된 경우 로그인에 성공하며, 메인 화면으로 이동한다.
- 정보가 일치하지 않는 경우 오류 메시지를 출력한다.

[회원가입] 링크

- 아직 계정이 없는 사용자는 하단의 '회원가입'을 눌러 회원가입 화면으로 이동할 수 있다.



EcoCheckIn 앱에서 신규 사용자가 계정을 생성하기 위한 화면으로, 이름, 이메일, 비밀번호를 입력하여 회원가입을 진행한다.

이름 입력

- 사용자 이름을 입력한다. 이후 마이페이지 또는 활동 기록에 사용된다.

이메일 입력

- 로그인 시 사용할 이메일 주소를 입력한다.
- 이미 사용 중인 이메일은 등록할 수 없다.

비밀번호 입력

- 로그인에 사용할 비밀번호를 설정한다.
- 보안을 위해 화면에는 ●●● 형태로 표시된다.

비밀번호 확인 입력

- 위에서 입력한 비밀번호와 동일하게 다시 한 번 입력한다.
- 두 비밀번호가 일치해야만 가입이 가능하다.

[회원가입 완료] 버튼

- 모든 항목이 정상적으로 입력되었고, 비밀번호가 일치할 경우 회원가입을 완료한다.
- 회원가입 성공 시 로그인 화면으로 이동한다.



EcoCheckIn 앱의 메인 화면으로, 현재 위치를 기준으로 주변 친환경 장소(에코 핫플)를 지도에서 확인하고, 체크인 또는 장소 등록을 수행할 수 있도록 구성한다.

친환경 장소 지도 표시

- 사용자의 현재 위치를 중심으로 친환경 장소들이 마커로 표시된다.
- 마커 아이콘은 녹색 트리 모양으로 표시되며, 터치하면 장소 정보 확인이 가능하다.

[체크인 등록하기] 버튼

- 사용자가 현재 위치에서 실천한 친환경 행동(예: 텁블러 사용, 자전거 이용 등)을 기록한다.
- 버튼을 누르면 체크인 입력 화면으로 이동한다.

[핫플 등록하기] 버튼

- 사용자가 알고 있는 친환경 장소를 앱에 등록할 수 있다.
- 장소명, 설명, 카테고리 등을 입력한 후 지도를 통해 마커로 표시된다.

[장소 상세 보기] 버튼

- 지도에 표시된 장소 목록 중 하나를 선택하여 세부 정보를 확인할 수 있다.

하단 내비게이션 바

- 홈 탭: 현재 화면(지도 기반)
- 피드 탭: 커뮤니티 게시글 목록으로 이동
- マイ페이지 탭: 개인 실천 내역 확인
- 지도 필터 탭: 카테고리별로 마커를 필터링할 수 있는 화면으로 이동



사용자가 특정 위치에서 실천한 친환경 행동을 기록하는 화면으로, 실천 장소와 내용을 입력하여 앱에 체크인 데이터를 등록한다.

체크인 장소 입력란

- 실천을 수행한 장소명을 직접 입력한다.
- 예: “스타벅스 삼성점”, “한강공원”, “학교 매점” 등
- 해당 장소명은 마이페이지 또는 지도 마커 정보에서 함께 표시된다.

실천 내용 입력란

- 사용자가 어떤 친환경 행동을 실천했는지를 자유롭게 서술한다.
- 예: “텀블러 사용”, “종이컵 대신 개인 머그컵 이용”, “자전거로 등교” 등
- 다중 줄 입력을 지원하여 상세한 내용도 기록 가능하다.

[체크인 완료] 버튼

- 모든 입력이 완료되면 해당 버튼을 눌러 체크인 데이터를 등록한다.
- 등록된 데이터는 Firebase Realtime Database에 저장되며, 마이페이지에서 확인할 수 있다.



사용자가 직접 친환경 장소(에코 핫플)를 앱에 등록할 수 있는 화면으로, 장소명, 카테고리, 설명을 입력하여 지도를 통해 공유한다.

장소명 입력란

- 등록할 친환경 장소의 이름을 입력한다.
- 예: “초록카페”, “제로웨이스트 상점”, “리필스테이션 강남점” 등
- 입력된 이름은 지도 마커와 마이페이지에 함께 표시된다.

카테고리 선택

- 장소의 유형을 선택한다.
- 일반적으로 ‘제로웨이스트’, ‘리필스테이션’, ‘중고’, ‘생활용품’ 등의 항목으로 구성되며, 드롭다운 목록에서 선택한다.

설명 입력란

- 해당 장소가 왜 친환경적인지 설명을 작성한다.
- 예: “다회용기 사용 가능”, “일회용 플라스틱 제공 안 함”, “텀블러 할인 매장” 등
- 내용은 다른 사용자가 장소를 이해하고 참고할 수 있도록 도움을 준다.

[등록하기] 버튼

- 모든 입력이 완료되면 장소 정보를 앱에 등록한다.
- 등록된 정보는 지도에 마커로 표시되며, 사용자 전체에게 공유된다.



지도에 등록된 친환경 장소의 상세 정보를 확인할 수 있는 화면으로, 장소명, 등록 일시, 주소, 설명 등이 표시된다.

장소명

- 해당 친환경 장소의 이름이 가장 상단에 굵은 글씨로 표시된다.
- 예: “home”, “제로웨이스트 카페”, “친환경 도서관” 등

등록 일시

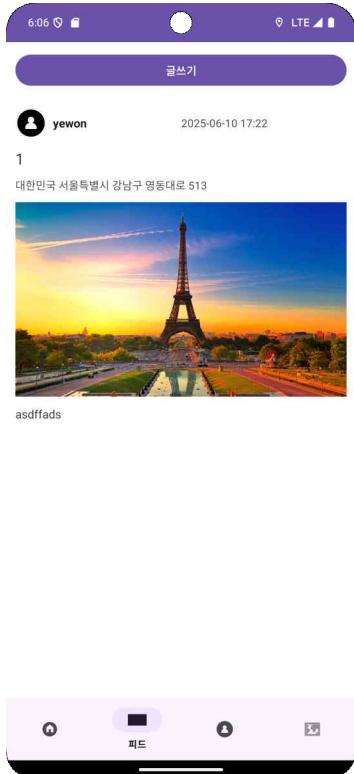
- 사용자가 이 장소를 앱에 등록한 시간 정보가 표시된다.
- 형식은 YYYY-MM-DD hh:mm:ss 형태이며, 등록 시각 기준이다.

주소 정보

- 장소의 실질적인 위치가 출력된다.
- 예: “대한민국 서울특별시 강남구 영동대로 513”

설명 내용

- 사용자가 직접 입력한 친환경 이유나 장소의 특징이 표시된다.
- 예: “텀블러 할인 제공”, “일회용품 사용 금지”, “리필 용기 사용 가능” 등



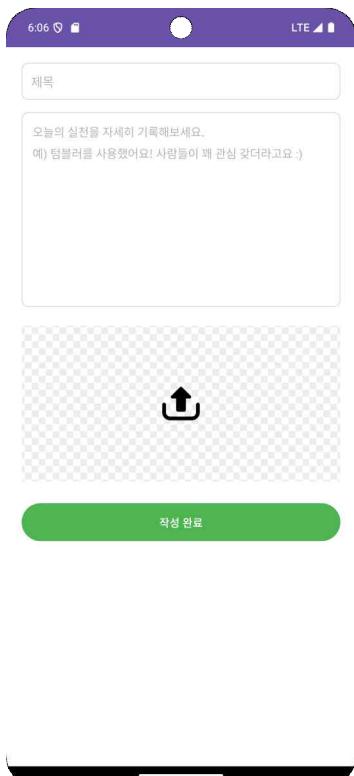
EcoCheckIn 사용자들이 공유한 친환경 실천 후기나 장소 정보를 게시글 형태로 열람할 수 있는 화면이다. 다른 사용자의 활동을 참고하고 동기를 얻을 수 있도록 구성되어 있다.

[글쓰기] 버튼

- 사용자가 새로운 게시글을 등록할 수 있도록 글쓰기 화면으로 이동한다.
- 사진, 제목, 주소, 내용 등을 입력하여 친환경 실천 후기를 공유할 수 있다.

게시글 항목

- 사용자들이 등록한 게시글이 리스트 형식으로 표시된다. 각 항목에는 다음 정보가 포함된다:
 - 작성자 닉네임: 게시글 작성자의 사용자명
 - 작성 일시: 게시글이 등록된 날짜 및 시간
 - 제목: 간단한 제목이나 장소명
 - 주소: 실천 장소 또는 추천 장소의 위치 정보
 - 사진: 사용자가 첨부한 이미지
 - 내용: 실천 후기나 장소에 대한 설명이 텍스트로 표시된다



사용자가 친환경 실천 사례나 장소에 대한 정보를 다른 사용자들과 공유하기 위해 게시글을 작성하는 화면이다. 제목, 내용, 사진을 입력하고 등록할 수 있다.

제목 입력란

- 게시글의 제목을 입력한다. 장소명이나 핵심 키워드를 간단히 작성하는 것이 좋다.
- 예: “텀블러 사용 후기”, “제로웨이스트 매장 다녀왔어요!”

내용 입력란

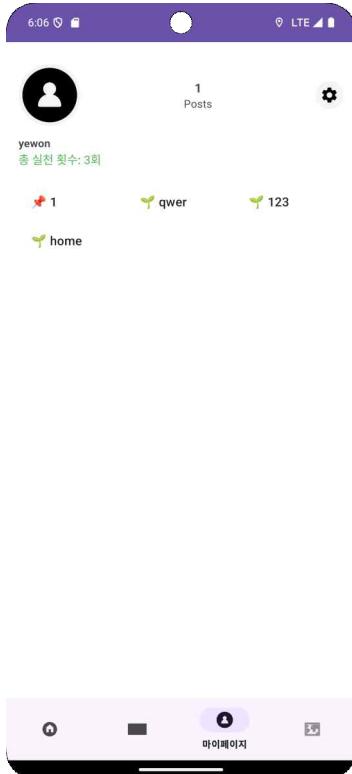
- 친환경 실천 내용을 구체적으로 서술한다.
- 예시 안내 문구: “텀블러를 사용했어요! 사람들이 꽤 관심 갖더라고요 :)”

사진 업로드 영역

- 업로드 아이콘을 눌러 사진을 선택하거나 촬영하여 첨부할 수 있다.
- 첨부한 이미지는 게시글에 함께 표시된다.

[작성 완료] 버튼

- 입력한 게시글을 등록한다. 등록 후 피드 목록에서 해당 게시글을 확인할 수 있다.



EcoCheckIn 사용자 본인의 실천 기록, 게시글 수, 등록한 장소 등을 한눈에 확인할 수 있는 화면이다. 사용자의 프로필 정보와 활동 내역을 기반으로 구성된다.

프로필 영역

- 상단에 사용자 닉네임, 프로필 이미지(기본 이미지), 체크인 총 실천 횟수, 피드 횟수가 표시된다.

게시글 수 표시

- 사용자가 작성한 피드 게시글 수가 Posts 항목에 숫자로 표시된다.

실천 장소 목록

- 사용자가 체크인한 장소가 아이콘과 함께 표시된다.
- ↗ 은 피드, ↘ 은 체크인 장소를 의미한다.

[⚙️ 설정 아이콘]

- 우측 상단의 톱니바퀴 아이콘을 누르면 앱 설정 화면으로 이동한다.



EcoCheckIn 앱의 환경설정 화면으로, 알림 수신 여부를 조정하거나 로그아웃을 수행할 수 있다.

앱 알림 뱌기 (스위치 토글)

- 사용자가 앱에서 제공하는 알림을 받을지 여부를 설정한다.
- 스위치를 켜면 리마인더 알림, 체크인 안내 등 앱 내 알림이 활성화된다.
- 스위치를 끄면 모든 알림이 중단된다.

[로그아웃] 버튼

- 현재 로그인한 계정에서 로그아웃한다.
- 로그아웃 후 로그인 화면으로 되돌아간다.



등록된 친환경 장소들을 지도 상에 표시하고, 원하는 카테고리별로 마커를 필터링하여 확인할 수 있도록 제공하는 화면이다.

Google 지도 기반 위치 표시

- 사용자 현재 위치를 중심으로 등록된 친환경 장소들이 마커로 표시된다.
- 각 장소는 아이콘 또는 마커로 시작화되어 지도 위에 나타난다.

카테고리 드롭다운 메뉴

- 지도 우측 상단에 위치한 드롭다운 버튼을 통해 장소 카테고리를 선택할 수 있다.
- 선택한 카테고리에 따라 지도에 표시되는 마커가 자동으로 필터링된다.

▶ 제공되는 카테고리:

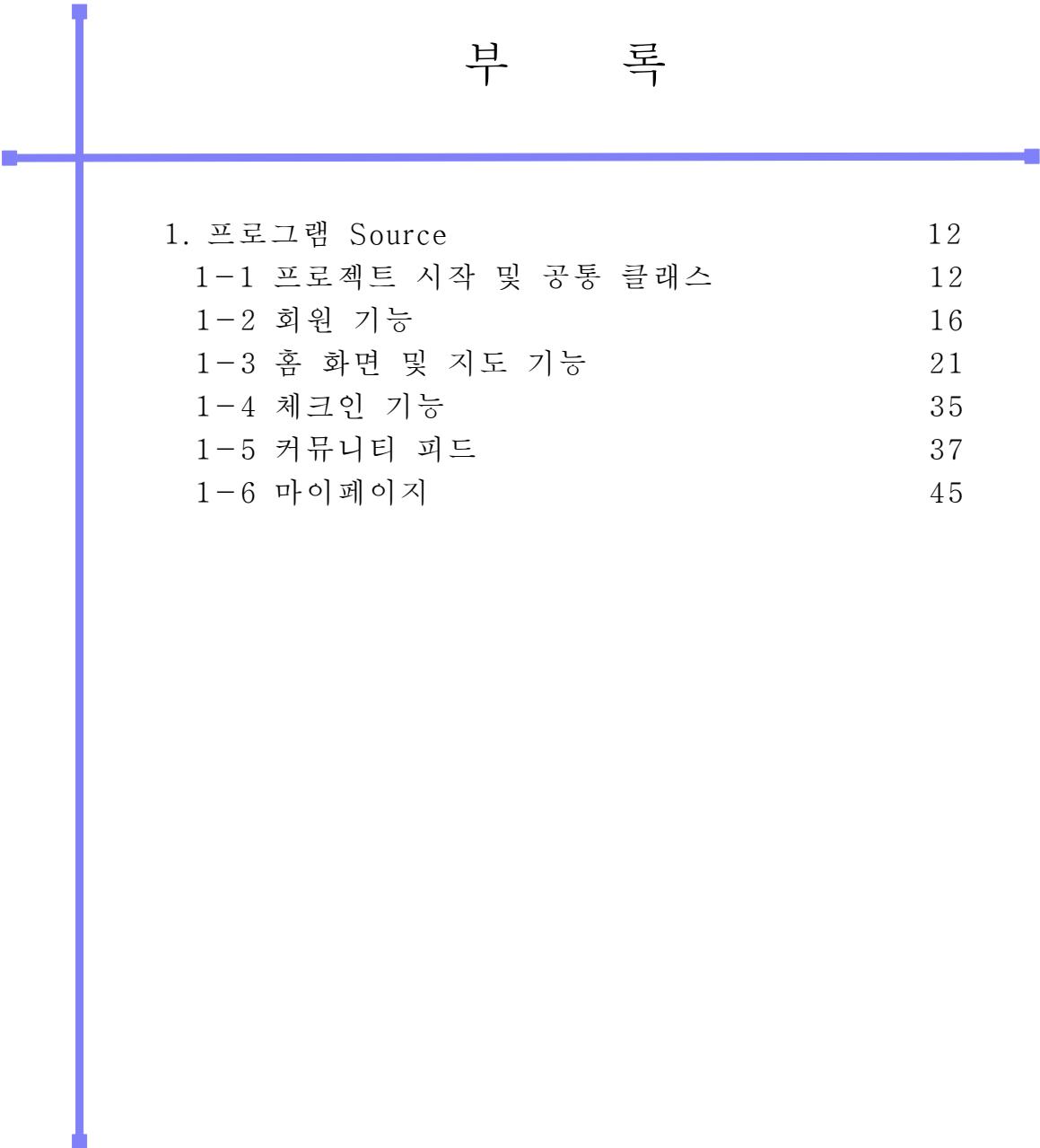
All (전체)

제로웨이스트

리필스테이션

중고

생활용품



부 록

1. 프로그램 Source	12
1-1 프로젝트 시작 및 공통 클래스	12
1-2 회원 기능	16
1-3 홈 화면 및 지도 기능	21
1-4 체크인 기능	35
1-5 커뮤니티 피드	37
1-6 마이페이지	45

1. 프로그램 Source

1-1 프로젝트 시작 및 공통 클래스

- MainActivity

```
package com.inhatc.ecocheckinapp;
```

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {
    private final Fragment homeFragment = new HomeFragment();
    private final Fragment feedFragment = new FeedFragment();
    private final Fragment myPageFragment = new MyPageFragment();
    private final Fragment mapFilterFragment = new MapWithFilterFragment();
    private Fragment activeFragment = homeFragment;
    private FragmentManager fragmentManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        UploadUtil.init(this);
        fragmentManager = getSupportFragmentManager();

        fragmentManager.beginTransaction().add(R.id.main_frame, myPageFragment,
        "3").hide(myPageFragment).commit();
        fragmentManager.beginTransaction().add(R.id.main_frame, feedFragment,
        "2").hide(feedFragment).commit();
        fragmentManager.beginTransaction().add(R.id.main_frame, mapFilterFragment,
        "4").hide(mapFilterFragment).commit();

        fragmentManager.beginTransaction().add(R.id.main_frame, homeFragment, "1").commit();
        BottomNavigationView navView = findViewById(R.id.nav_view);

        navView.setOnItemSelectedListener(item -> {
            int id = item.getItemId();
            if (id == R.id.navigation_home) {
                switchFragment(homeFragment);
                return true;
            } elseif (id == R.id.navigation_feed) {
                switchFragment(feedFragment);
                return true;
            } elseif (id == R.id.navigation_mypage) {
                switchFragment(myPageFragment);
                return true;
            }
        });
    }

    private void switchFragment(Fragment fragment) {
        fragmentManager.beginTransaction().hide(activeFragment).show(fragment).commit();
        activeFragment = fragment;
    }
}
```

```

        switchFragment(myPageFragment);
        return true;
    } elseif (id == R.id.navigation_map_filter) {
        switchFragment(mapFilterFragment);
        return true;
    }
    return false;
});

}

private void switchFragment(Fragment targetFragment) {
    fragmentManager.beginTransaction().hide(activeFragment).show(targetFragment).commit();
    activeFragment = targetFragment;
}
}

```

- UploadUtil

```
package com.inhatc.ecocheckinapp;
```

```

import android.content.Context;
import com.cloudinary.android.MediaManager;
import com.cloudinary.android.callback.ErrorInfo;
import com.cloudinary.android.callback.UploadCallback;
import java.util.HashMap;
import java.util.Map;

public class UploadUtil {
    public static void init(Context context) {
        Map<String, String> config = new HashMap<>();
        config.put("cloud_name", "dwyf8emyg");
        config.put("api_key", "562898372841627");
        config.put("api_secret", "f2ACwXPjzaR_1koqmJOMscix-4c");
        MediaManager.init(context, config);
    }

    public static void uploadImage(String filePath, UploadCallback callback) {
        MediaManager.get().upload(filePath)
            .callback(callback)
            .dispatch();
    }
}

```

- EcoActionModel

```

package com.inhatc.ecocheckinapp;

public class EcoActionModel {
    public String email;
    public String place;
    public String memo;
    public double latitude;
    public double longitude;
    public String date;

    public EcoActionModel() { }

    public EcoActionModel(String email, String place, String memo, double latitude, double longitude,
String date) {
        this.email = email;
        this.place = place;
        this.memo = memo;
        this.latitude = latitude;
        this.longitude = longitude;
        this.date = date;
    }

    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }

    public String getPlace() { return place; }
    public void setPlace(String place) { this.place = place; }

    public double getLatitude() { return latitude; }
    public void setLatitude(double latitude) { this.latitude = latitude; }

    public double getLongitude() { return longitude; }
    public void setLongitude(double longitude) { this.longitude = longitude; }

    public String getDate() { return date; }
    public void setDate(String date) { this.date = date; }

    public String getMemo() { return memo; }
    public void setMemo(String memo) { this.memo = memo; }
}

```

- EcoPlace

```
package com.inhatc.ecocheckinapp;

public class EcoPlace {
    private String name;
    private String category;
    private String description;
    private double latitude;
    private double longitude;

    public EcoPlace() { }

    public EcoPlace(String name, String category, String description, double latitude, double longitude) {
        this.name = name;
        this.category = category;
        this.description = description;
        this.latitude = latitude;
        this.longitude = longitude;
    }

    public String getName() { return name; }
    public String getCategory() { return category; }
    public String getDescription() { return description; }
    public double getLatitude() { return latitude; }
    public double getLongitude() { return longitude; }

    public void setName(String name) { this.name = name; }
    public void setCategory(String category) { this.category = category; }
    public void setDescription(String description) { this.description = description; }
    public void setLatitude(double latitude) { this.latitude = latitude; }
    public void setLongitude(double longitude) { this.longitude = longitude; }
}
```

- FeedModel

```
package com.inhatc.ecocheckinapp;

public class FeedModel {
    private String email;
    private String imageUrl;
    private String content;
    private long timestamp;
    private String title;
    private double latitude;
    private double longitude;

    public FeedModel() { }
```

```

public FeedModel(String email, String imageUrl, String content, long timestamp, String title, double
latitude, double longitude) {
    this.email = email;
    this.imageUrl = imageUrl;
    this.content = content;
    this.timestamp = timestamp;
    this.title = title;
    this.latitude = latitude;
    this.longitude = longitude;
}

public String getTitle() { return title; }
public void setTitle(String title) { this.title = title; }

public void setEmail(String email) { this.email = email; }
public String getEmail() { return email; }

public String getImageUrl() { return imageUrl; }
public void setImageUrl(String imageUrl) { this.imageUrl = imageUrl; }

public String getContent() { return content; }
public void setContent(String content) { this.content = content; }

public long getTimestamp() { return timestamp; }
public void setTimestamp(long timestamp) { this.timestamp = timestamp; }

public double getLatitude() { return latitude; }
public void setLatitude(double latitude) { this.latitude = latitude; }

public double getLongitude() { return longitude; }
public void setLongitude(double longitude) { this.longitude = longitude; }
}

```

1-2 회원 기능

- LoginActivity

```
package com.inhatc.ecocheckinapp;
```

```

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebaseio.database.DataSnapshot;

```

```

import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

public class LoginActivity extends AppCompatActivity {
    private EditText edtEmail, edtPassword;
    private Button btnLogin, btnToRegister;
    private DatabaseReference userRef;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        edtEmail = findViewById(R.id.edt_email);
        edtPassword = findViewById(R.id.edt_password);
        btnLogin = findViewById(R.id.btn_login);
        btnToRegister = findViewById(R.id.btn_to_register);
        userRef = FirebaseDatabase.getInstance().getReference("users");

        btnLogin.setOnClickListener(v -> {
            String email = edtEmail.getText().toString().trim();
            String password = edtPassword.getText().toString().trim();
            if (email.isEmpty() || password.isEmpty()) {
                Toast.makeText(this, "이메일과 비밀번호를 모두 입력하세요.",
                        Toast.LENGTH_SHORT).show();
                return;
            }
            loginUser(email, password);
        });
        btnToRegister.setOnClickListener(v ->
                startActivity(new Intent(this, RegisterActivity.class))
        );
    }

    private void loginUser(String email, String password) {
        userRef.orderByChild("email").equalTo(email)
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        if (snapshot.exists()) {
                            for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                                RegisterActivity.UserAccount user =
                                        userSnapshot.getValue(RegisterActivity.UserAccount.class);
                                if (user != null && user.password.equals(password)) {
                                    Sharedpreferences prefs = getSharedpreferences("UserPrefs",
                                           

```

```
MODE_PRIVATE);  
        prefs.edit().putString("userEmail", email).apply();  
        Toast.makeText(LoginActivity.this, "로그인 성공",  
Toast.LENGTH_SHORT).show();  
    }  
    startActivity(new Intent(LoginActivity.this, MainActivity.class));  
    finish();  
    return;  
}  
}  
Toast.makeText(LoginActivity.this, "비밀번호가 일치하지 않습니다.",  
Toast.LENGTH_SHORT).show();  
} else {  
    Toast.makeText(LoginActivity.this, "존재하지 않는 이메일입니다.",  
Toast.LENGTH_SHORT).show();  
}  
}  
}  
@Override  
public void onCancelled(@NonNull DatabaseError error) {  
    Toast.makeText(LoginActivity.this, "데이터베이스 오류: "+ error.getMessage(),  
Toast.LENGTH_SHORT).show();  
}  
});  
}  
}
```

- RegisterActivity

```
package com.inhatc.ecocheckinapp;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
public class RegisterActivity extends AppCompatActivity {  
    private EditText edtEmail, edtPassword, edtConfirm, edtName;  
    private Button btnRegister;  
    private DatabaseReference userRef;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_register);
edtEmail = findViewById(R.id.edt_email);
edtPassword = findViewById(R.id.edt_password);
edtConfirm = findViewById(R.id.edt_confirm_password);
edtName = findViewById(R.id.edt_name);
btnRegister = findViewById(R.id.btn_register);
userRef = FirebaseDatabase.getInstance().getReference("users");

btnRegister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        saveUserToDatabase();
    }
});

private void saveUserToDatabase() {
    String email = edtEmail.getText().toString().trim();
    String password = edtPassword.getText().toString().trim();
    String confirm = edtConfirm.getText().toString().trim();
    String name = edtName.getText().toString().trim();

    if (email.isEmpty() || password.isEmpty() || name.isEmpty()) {
        Toast.makeText(this, "모든 항목을 입력하세요", Toast.LENGTH_SHORT).show();
        return;
    }

    if (!password.equals(confirm)) {
        Toast.makeText(this, "비밀번호가 다릅니다.", Toast.LENGTH_SHORT).show();
        return;
    }

    userRef.orderByChild("email").equalTo(email)
        .addListenerForSingleValueEvent(new com.google.firebaseio.database.ValueEventListener()
    {

        @Override
        public void onDataChange(@NonNull com.google.firebaseio.database.DataSnapshot
dataSnapshot) {
            if (dataSnapshot.exists()) {
                Toast.makeText(RegisterActivity.this, "이미 사용 중인 이메일입니다.",
Toast.LENGTH_SHORT).show();
            } else {
                String uid = userRef.push().getKey();
                if (uid != null) {
                    UserAccount user = new UserAccount(uid, email, name, password);

```

```

        userRef.child(uid).setValue(user)
            .addOnSuccessListener(aVoid -> {
                Toast.makeText(RegisterActivity.this, "회원가입 완료",
Toast.LENGTH_SHORT).show();
            })
            .addOnFailureListener(e -> {
                Toast.makeText(RegisterActivity.this, "저장 실패: "+e.getMessage(), Toast.LENGTH_SHORT).show();
            });
        }
    }
}

@Override
public void onCancelled(@NonNull com.google.firebaseio.database.DatabaseError databaseError) {
    Toast.makeText(RegisterActivity.this, "데이터베이스 오류: "+databaseError.getMessage(), Toast.LENGTH_SHORT).show();
}
}

public static class UserAccount {
    public String uid;
    public String email;
    public String name;
    public String password;

    public UserAccount() {}

    public UserAccount(String uid, String email, String name, String password) {
        this.uid = uid;
        this.email = email;
        this.name = name;
        this.password = password;
    }
}
}

```

- SettingsActivity

```

package com.inhatc.ecocheckinapp;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Switch;

```

```

import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class SettingsActivity extends AppCompatActivity {
    private Switch switchNotification;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_settings);
        switchNotification = findViewById(R.id.switch_notification);
        switchNotification.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(this, "알림이 켜졌습니다.", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "알림이 꺼졌습니다.", Toast.LENGTH_SHORT).show();
            }
        });
    }

    findViewById(R.id.btn_logout).setOnClickListener(v -> {
        Toast.makeText(this, "로그아웃되었습니다.", Toast.LENGTH_SHORT).show();
        Intent intent =new Intent(this, LoginActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
    });
}

```

1-3 홈 화면 및 지도 기능

- HomeFragment

```
package com.inhatc.ecocheckinapp;
```

```

import android.Manifest;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.os.Looper;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```

import android.widget.Button;
import android.widget.Toast;
import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.Fragment;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.Locale;

public class HomeFragment extends Fragment implements OnMapReadyCallback {

    private GoogleMap mMap;
    private FusedLocationProviderClient fusedLocationClient;
    private Marker selectedMarker;
    private String selectedPlace, selectedAddress, selectedMemo, selectedDate;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                            @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_home, container, false);
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(requireActivity());
        SupportMapFragment mapFragment = (SupportMapFragment)
            getChildFragmentManager().findFragmentById(R.id.home_map);
        if (mapFragment != null) {
            mapFragment.getMapAsync(this);
        } else {

```

```

        Log.e("HomeFragment", "mapFragment is null. ID 확인 필요");
    }

Button btnRegisterPlace = view.findViewById(R.id.btn_register_place_test);
Button btnPlaceDetail = view.findViewById(R.id.btn_place_detail_test);
Button btnCheckIn = view.findViewById(R.id.btn_checkin);

btnRegisterPlace.setOnClickListener(v -> {
    startActivity(new Intent(getActivity(), PlaceRegisterActivity.class));
});

btnPlaceDetail.setOnClickListener(v -> {
    if (selectedMarker ==null) {
        Toast.makeText(getContext(), "마커를 먼저 선택해주세요.",
        Toast.LENGTH_SHORT).show();
        return;
    }

    Intent intent =new Intent(getActivity(), PlaceDetailActivity.class);
    intent.putExtra("title", selectedPlace);
    intent.putExtra("memo", selectedMemo);
    intent.putExtra("date", selectedDate);
    intent.putExtra("address", selectedAddress);
    startActivity(intent);
});

btnCheckIn.setOnClickListener(v -> {
    CheckInDialog dialog =new CheckInDialog();
    dialog.setOnCheckInListener((place, memo) -> {
        if (ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(getContext(), "위치 권한이 필요합니다.",
            Toast.LENGTH_SHORT).show();
            return;
        }
        fusedLocationClient.getLastLocation().addOnSuccessListener(location -> {
            if (location ==null) {
                Toast.makeText(getContext(), "위치 정보를 가져올 수 없습니다.",
                Toast.LENGTH_SHORT).show();
                return;
            }
            SharedPreferences prefs = requireContext().getSharedPreferences("UserPrefs",
Context.MODE_PRIVATE);
            String email = prefs.getString("userEmail", null);
            if (email ==null) {
                Toast.makeText(getContext(), "사용자 이메일이 없습니다.",
                Toast.LENGTH_SHORT).show();
            }
        });
    });
});

```

```

        return;
    }

    DatabaseReference usersRef = FirebaseDatabase.getInstance().getReference("users");
    usersRef.orderByChild("email").equalTo(email).limitToFirst(1)
        .get()
        .addOnSuccessListener(snapshot -> {
            if (snapshot.exists()) {
                DataSnapshot userSnap = snapshot.getChildren().iterator().next();
                String name = userSnap.child("name").getValue(String.class);
                LatLng currentLatLng =new LatLng(location.getLatitude(),
                location.getLongitude());

                String dateStr =new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.KOREA).format(new Date());
                MarkerOptions options =new MarkerOptions()
                    .position(currentLatLng)
                    .title(place)
                    .snippet("메모: "+ memo +"\n사용자: "+ name +"\n날짜:
"+ dateStr);
                mMap.addMarker(options);

                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLatLng, 15));
            } else {
                Toast.makeText(getContext(), "사용자 이름을 찾을 수 없습니다.",
Toast.LENGTH_SHORT).show();
            }
        })
        .addOnFailureListener(e -> {
            Toast.makeText(getContext(), "사용자 조회 실패: "+ e.getMessage(),
Toast.LENGTH_SHORT).show();
        });
    });

    dialog.show(getParentFragmentManager(), "CheckInDialog");
});

return view;
}

@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    mMap = googleMap;

    if (ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        requestPermissions(newString[]{}

```

```

        Manifest.permission.ACCESS_FINE_LOCATION,
        Manifest.permission.ACCESS_COARSE_LOCATION
    ), 1000);
    return;
}

mMap.setMyLocationEnabled(true);
mMap.setOnMarkerClickListener(marker -> {
    selectedMarker = marker;
    selectedPlace = marker.getTitle();
    String snippet = marker.getSnippet();
    if (snippet !=null) {
        for (String part : snippet.split(";")) {
            if (part.startsWith("memo=")) {
                selectedMemo = part.substring("memo=".length());
            } elseif (part.startsWith("date=")) {
                selectedDate = part.substring("date=".length());
            }
        }
    }
}

LatLng position = marker.getPosition();
double lat = position.latitude;
double lng = position.longitude;
try {
    Geocoder geocoder =new Geocoder(requireContext(), Locale.KOREA);
    List<Address> addressList = geocoder.getFromLocation(lat, lng, 1);
    if (addressList !=null&&!addressList.isEmpty()) {
        selectedAddress = addressList.get(0).getAddressLine(0);
    } else {
        selectedAddress ="주소 정보 없음";
    }
} catch (IOException e) {
    e.printStackTrace();
    selectedAddress ="주소 변환 오류";
}
return false;
});

LocationRequest locationRequest = LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(10000);
fusedLocationClient.requestLocationUpdates(
    locationRequest,
    new LocationCallback() {
        @Override

```

```

        public void onLocationResult(@NonNull LocationResult locationResult) {
            Location location = locationResult.getLastLocation();
            if (location != null) {
                LatLng current = new LatLng(location.getLatitude(),
                    location.getLongitude());
                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(current, 15));
            }
            fusedLocationClient.removeLocationUpdates(this);
        }
    },
    Looper.getMainLooper()
);

SharedPreferences prefs = requireContext().getSharedPreferences("UserPrefs",
    Context.MODE_PRIVATE);
String userEmail = prefs.getString("userEmail", null);
if (userEmail != null) {
    DatabaseReference checkInsRef = FirebaseDatabase.getInstance().getReference("CheckIns");
    checkInsRef.orderByChild("email").equalTo(userEmail)
        .get()
        .addOnSuccessListener(snapshot -> {
            for (DataSnapshot checkInSnap : snapshot.getChildren()) {
                Double lat = checkInSnap.child("latitude").getValue(Double.class);
                Double lng = checkInSnap.child("longitude").getValue(Double.class);
                String place = checkInSnap.child("place").getValue(String.class);
                String memo = checkInSnap.child("memo").getValue(String.class);
                String date = checkInSnap.child("date").getValue(String.class);
                if (lat != null && lng != null) {
                    LatLng position = new LatLng(lat, lng);
                    MarkerOptions marker = new MarkerOptions()
                        .position(position)
                        .title(place != null ? place : "체크인 위치")
                        .snippet("memo=" + (memo != null ? memo : "") + ";date=" +
(date != null ? date : ""));
                    mMap.addMarker(marker);
                }
            }
        })
        .addOnFailureListener(e -> {
            Toast.makeText(getContext(), "체크인 데이터 조회 실패: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
        });
}
}

@Override
public void onRequestPermissionsResult(int requestCode,

```

```

        @NonNull String[] permissions,
        @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 1000 &&
        grantResults.length > 0 &&
        grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.checkSelfPermission(requireContext(),
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
            mMap.setMyLocationEnabled(true);
            onMapReady(mMap);
        }
    } else {
        Toast.makeText(getContext(), "위치 권한이 필요합니다.", Toast.LENGTH_SHORT).show();
    }
}
}

```

- MapWithFilterFragment

```
package com.inhatc.ecocheckinapp;
```

```

import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

```

```
public class MapWithFilterFragment extends Fragment implements OnMapReadyCallback {
```

```
    private GoogleMap mMap;
```

```

private String selectedCategory ="All";

@NoArgsConstructor
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
                        @Nullable ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_map_filter, container, false);
    Spinner spinner = view.findViewById(R.id.spinner_category);
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
        requireContext(),
        R.array.place_categories,
        android.R.layout.simple_spinner_item
    );
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
    spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
            selectedCategory = parent.getItemAtPosition(position).toString();
            Toast.makeText(getApplicationContext(), "Selected: "+ selectedCategory,
                    Toast.LENGTH_SHORT).show();
            loadMarkers();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {}
    });
    SupportMapFragment mapFragment = (SupportMapFragment) getChildFragmentManager()
        .findFragmentById(R.id.map);
    if (mapFragment !=null) {
        mapFragment.getMapAsync(this);
    }
    return view;
}

@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    mMap = googleMap;
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(37.5551, 126.9102), 13));
    loadMarkers();
}

private void loadMarkers() {
    if (mMap ==null) {

```

```

        Log.w("MapFilter", "Map is null, skipping marker load.");
        return;
    }

mMap.clear();
DatabaseReference ref = FirebaseDatabase.getInstance()
    .getReference("eco_places");
ref.addListenerForSingleValueEvent(new ValueEventListener() {

    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        Log.d("MapFilter", "onDataChange called. Children count: "+snapshot.getChildrenCount());
        for (DataSnapshot placeSnapshot : snapshot.getChildren()) {
            EcoPlace place = placeSnapshot.getValue(EcoPlace.class);
            if (place !=null) {
                Log.d("MapFilter", "Fetched place: "+ place.getName() +", category: "+
place.getCategory());
                if (selectedCategory.equals("All") ||
place.getCategory().trim().equals(selectedCategory.trim())) {
                    LatLng pos =new LatLng(place.getLatitude(), place.getLongitude());
                    Log.d("MapFilter", "Adding marker at: "+ pos);
                    mMap.addMarker(new MarkerOptions()
                        .position(pos)
                        .title(place.getName())
                        .snippet("분류: "+ place.getCategory() +"\n"+
place.getDescription()));
                    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(pos, 15));
                } else {
                    Log.d("MapFilter", "Skipped: "+ place.getName() +" due to category
mismatch.");
                }
            } else {
                Log.w("MapFilter", "Null place encountered in snapshot.");
            }
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.e("MapFilter", "Failed to load data: ", error.toException());
    }
});
```

- PlaceMapFragment

```

package com.inhatc.ecocheckinapp;

import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.AdapterView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebaseio.database.*;
import java.util.ArrayList;
import java.util.List;

public class PlaceMapFragment extends Fragment implements OnMapReadyCallback {
    private GoogleMap mMap;
    private DatabaseReference databaseReference;
    private List<Marker> markerList = new ArrayList<>();
    private Spinner spinnerCategory;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                           @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_place_map, container, false);
        spinnerCategory = view.findViewById(R.id.spinner_category);

        ArrayAdapter<String> adapter = new ArrayAdapter<>(
            requireContext(),
            android.R.layout.simple_spinner_item,
            newString[]{"전체", "제로웨이스트", "리필스테이션", "중고매장", "생활용품"});

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinnerCategory.setAdapter(adapter);
        spinnerCategory.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

```

```

@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    filterMarkers(spinnerCategory.getSelectedItem().toString());
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
}
});

SupportMapFragment mapFragment = (SupportMapFragment) getChildFragmentManager()
    .findFragmentById(R.id.place_map);
if (mapFragment != null) {
    mapFragment.getMapAsync(this);
}
return view;
}

@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    mMap = googleMap;
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(37.5665, 126.9780), 11));
    databaseReference = FirebaseDatabase.getInstance().getReference("eco_places");
    databaseReference.addValueEventListener(new ValueEventListener() {

        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            markerList.clear();
            for (DataSnapshot placeSnap : snapshot.getChildren()) {
                String name = placeSnap.child("placeName").getValue(String.class);
                String category = placeSnap.child("category").getValue(String.class);
                String description = placeSnap.child("description").getValue(String.class);
                Double lat = placeSnap.child("latitude").getValue(Double.class);
                Double lng = placeSnap.child("longitude").getValue(Double.class);
                if (lat != null && lng != null) {
                    Marker marker = mMap.addMarker(new MarkerOptions()
                        .position(new LatLng(lat, lng))
                        .title(name)
                        .snippet("카테고리: " + category + "\n" + description));
                    marker.setTag(category);
                    markerList.add(marker);
                }
            }
        }
    });
}

```

```

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.e("Firebase", "DB 오류: "+ error.getMessage());
    }
});

}

private void filterMarkers(String selectedCategory) {
    for (Marker marker : markerList) {
        String markerCategory = (String) marker.getTag();
        if (selectedCategory.equals("전체") || selectedCategory.equals(markerCategory)) {
            marker.setVisible(true);
        } else {
            marker.setVisible(false);
        }
    }
}

```

- PlaceDetailActivity

```
package com.inhatc.ecocheckinapp;
```

```

import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.widget.TextView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class PlaceDetailActivity extends AppCompatActivity {
    private TextView txtTitle, txtDate, txtAddress, txtMemo;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_place_detail);
        txtTitle = findViewById(R.id.txt_title);
        txtDate = findViewById(R.id.txtDate);
        txtAddress = findViewById(R.id.txtAddress);
        txtMemo = findViewById(R.id.txt_memo);

        String title = getIntent().getStringExtra("title");
        String memo = getIntent().getStringExtra("memo");
    }
}

```

```

String date = getIntent().getStringExtra("date");
String address = getIntent().getStringExtra("address");

if (title ==null) title ="(제목 없음)";
if (memo ==null) memo ="(메모 없음)";
if (date ==null) date ="(날짜 없음)";

txtTitle.setText(title);
txtMemo.setText(memo);
txtDate.setText(date);
txtAddress.setText(address);
}

}

```

- PlaceRegisterActivity

```
package com.inhatc.ecocheckinapp;
```

```

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.Priority;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;

```

```

public class PlaceRegisterActivity extends AppCompatActivity {
    private EditText edtName, edtDescription;
    private Spinner edtCategory;
    private Button btnSubmit;
    private FusedLocationProviderClient locationClient;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_place_register);
edtName = findViewById(R.id.edt_place_name);
edtCategory = findViewById(R.id.edt_place_category);
edtDescription = findViewById(R.id.edt_place_description);
btnSubmit = findViewById(R.id.btn_register_place);
locationClient = LocationServices.getFusedLocationProviderClient(this);

btnSubmit.setOnClickListener(v -> {
    String name = edtName.getText().toString().trim();
    String category = edtCategory.getSelectedItem().toString();
    String description = edtDescription.getText().toString().trim();
    if (name.isEmpty() || category.isEmpty() || description.isEmpty()) {
        Toast.makeText(this, "모든 항목을 입력해주세요", Toast.LENGTH_SHORT).show();
        return;
    }

    if (ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{
            Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.ACCESS_COARSE_LOCATION
        }, 1001);
        Toast.makeText(this, "위치 권한이 필요합니다.", Toast.LENGTH_SHORT).show();
        return;
    }

    locationClient.getCurrentLocation(Priority.PRIORITY_HIGH_ACCURACY, null)
        .addOnSuccessListener(location -> {
            if (location !=null) {
                double lat = location.getLatitude();
                double lng = location.getLongitude();
                String date =new SimpleDateFormat("yyyy-MM-dd",
                        Locale.KOREA).format(new Date());
                HashMap<String, Object> data =new HashMap<>();
                data.put("name", name);
                data.put("category", category);
                data.put("description", description);
                data.put("latitude", lat);
                data.put("longitude", lng);
                data.put("date", date);
                DatabaseReference dbRef =
                    FirebaseDatabase.getInstance().getReference("eco_places");
                dbRef.push().setValue(data)
            }
        })
    }
}

```

```
Toast.LENGTH_SHORT).show();
        .addOnSuccessListener(unused -> {
            Toast.makeText(this, "친환경 장소가 등록되었습니다!",

Toast.LENGTH_SHORT).show();
        .addOnSuccessListener(unused -> {
            finish();
        })
        .addOnFailureListener(e -> {
            Toast.makeText(this, "등록 실패: "+ e.getMessage(),
        });
    } else {
        Toast.makeText(this, "위치를 가져올 수 없습니다.",
Toast.LENGTH_SHORT).show();
    }
});
```

1-4 체크인 기능

- CheckInDialog

```
package com.inhatc.ecocheckinapp
```

```
import android.content.Context;
import android.content.SharedPreferences;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import com.google.android.material.bottomsheet.BottomSheetDialogFragment;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

public class CheckInDialog extends BottomSheetDialogFragment {
    private EditText edtPlaceName;
    private EditText edtMemo;
    private Button btnSubmit;
```

```

private DatabaseReference checkinRef;

@NoArgsConstructor
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
                        @Nullable ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.dialog_checkin, container, false);
    edtPlaceName = view.findViewById(R.id.edt_place_name);
    edtMemo = view.findViewById(R.id.edt_memo);
    btnSubmit = view.findViewById(R.id.btn_submit);
    checkinRef = FirebaseDatabase.getInstance().getReference("CheckIns");
    btnSubmit.setOnClickListener(v -> uploadCheckin());
    return view;
}

private void uploadCheckin() {
    String place = edtPlaceName.getText().toString().trim();
    String memo = edtMemo.getText().toString().trim();
    if (place.isEmpty() || memo.isEmpty()) {
        Toast.makeText(getContext(), "모든 정보를 입력해주세요.",
        Toast.LENGTH_SHORT).show();
        return;
    }

    SharedPreferences prefs = requireContext()
        .getSharedPreferences("UserPrefs", Context.MODE_PRIVATE);
    String email = prefs.getString("userEmail", "unknown@example.com");
    LocationManager lm = (LocationManager)
    requireContext().getSystemService(Context.LOCATION_SERVICE);

    double latitude = 0;
    double longitude = 0;

    try {
        Location location = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (location != null) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    } catch (SecurityException e) {
        Toast.makeText(getContext(), "위치 권한이 필요합니다.", Toast.LENGTH_SHORT).show();
    }

    long timestamp = System.currentTimeMillis();
    String date = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault()).format(new

```

```

Date(timestamp));
EcoActionModel data =new EcoActionModel(email, place, memo, latitude, longitude, date);
checkinRef.push()
    .setValue(data)
    .addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            Toast.makeText(getApplicationContext(), "체크인 완료", Toast.LENGTH_SHORT).show();
            dismiss();
        } else {
            Toast.makeText(getApplicationContext(), "실패: "+ task.getException().getMessage(),
            Toast.LENGTH_SHORT).show();
        }
    });
}

public interface OnCheckInListener {
    void onCheckIn(String place, String memo);
}

private OnCheckInListener onCheckInListener;
public CheckInDialog setOnCheckInListener(OnCheckInListener listener) {
    this.onCheckInListener = listener;
    return this;
}
}

```

1-5 커뮤니티 피드

- FeedFragment

```
package com.inhatc.ecocheckinapp;
```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;

```

```

import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import java.util.ArrayList;
import java.util.List;

public class FeedFragment extends Fragment {
    private RecyclerView rvFeed;
    private FeedAdapter feedAdapter;
    private List<FeedModel> feedList;
    private DatabaseReference feedRef;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                            @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_feed, container, false);
        view.findViewById(R.id.btn_write_post).setOnClickListener(v -> {
            startActivity(new Intent(getActivity(), WritePostActivity.class));
        });
        rvFeed = view.findViewById(R.id.rv_feed);
        rvFeed.setLayoutManager(new LinearLayoutManager(getContext()));
        feedList =new ArrayList<>();
        feedAdapter =new FeedAdapter(feedList, getContext());
        rvFeed.setAdapter(feedAdapter);
        feedRef = FirebaseDatabase.getInstance().getReference("feeds");
        feedRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                feedList.clear();
                for (DataSnapshot snap : snapshot.getChildren()) {
                    FeedModel model = snap.getValue(FeedModel.class);
                    if (model !=null) feedList.add(model);
                }
                feedAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                }
            });
        return view;
    }
}

```

- FeedAdapter

```

package com.inhatc.ecocheckinapp;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.Locale;

public class FeedAdapter extends RecyclerView.Adapter<FeedAdapter.ViewHolder> {
    private List<FeedModel> list;
    private Context context;
    public FeedAdapter(List<FeedModel> list, Context context) {
        this.list = list;
        this.context = context;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_feed_post, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        FeedModel model = list.get(position);
        fetchUserNameFromEmail(model.getEmail(), holder.username);
        holder.caption.setText(model.getContent());
        holder.title.setText(model.getTitle());
        SimpleDateFormat sdf =new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.getDefault());
        String formattedDate = sdf.format(new Date(model.getTimestamp()));
        holder.date.setText(formattedDate);
    }
}

```

```

        Glide.with(context).load(model.getImageUrl()).into(holder.image);
        String address = getAddressFromLatLang(model.getLatitude(), model.getLongitude());
        holder.address.setText(address);
    }

    @Override
    public int getItemCount() {
        return list.size();
    }

    static class ViewHolder extends RecyclerView.ViewHolder {
        TextView username, caption, date, title, address;
        ImageView image;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            title = itemView.findViewById(R.id.txt_title);
            username = itemView.findViewById(R.id.txt_username);
            caption = itemView.findViewById(R.id.txt_caption);
            date = itemView.findViewById(R.id.txt_date);
            address = itemView.findViewById(R.id.txt_address);
            image = itemView.findViewById(R.id.img_post);
        }
    }

    private String getAddressFromLatLang(double lat, double lng) {
        Geocoder geocoder =new Geocoder(context, Locale.KOREA);
        try {
            List<Address> addresses = geocoder.getFromLocation(lat, lng, 1);
            if (addresses !=null&&!addresses.isEmpty()) {
                return addresses.get(0).getAddressLine(0);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "주소를 가져올 수 없음";
    }

    private void fetchUserNameFromEmail(String email, TextView usernameView) {
        DatabaseReference usersRef = FirebaseDatabase.getInstance().getReference("users");
        usersRef.orderByChild("email").equalTo(email)
            .limitToFirst(1)
            .get()
            .addOnSuccessListener(snapshot -> {
                if (snapshot.exists()) {
                    for (DataSnapshot userSnap : snapshot.getChildren()) {
                        String name = userSnap.child("name").getValue(String.class);

```

```

        usernameView.setText(name !=null ? name : "(이름 없음)");
        break;
    }
} else {
    usernameView.setText("(할 수 없음)");
}
})
.addOnFailureListener(e -> {
    usernameView.setText("(불러오기 실패)");
});
}
}
}

```

- WritePostActivity

```
package com.inhatc.ecocheckinapp;
```

```

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import com.google.firebaseio.storage.FirebaseStorage;
import com.google.firebaseio.storage.StorageReference;
import com.inhatc.ecocheckinapp.FeedModel;
import org.json.JSONObject;
import java.io.DataOutputStream;
import java.io.InputStream;

```

```

import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Scanner;
import java.util.UUID;
import android.Manifest;

public class WritePostActivity extends AppCompatActivity {
    private static final int PICK_IMAGE_REQUEST = 101;
    private ImageView imgPreview, btnUploadImage;
    private EditText edtTitle, edtContent;
    private Button btnSubmit;
    private Uri imageUri;
    private StorageReference storageRef;
    private DatabaseReference feedRef;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_write_post);
        imgPreview = findViewById(R.id.img_preview);
        edtTitle = findViewById(R.id.edt_title);
        edtContent = findViewById(R.id.edt_content);
        btnUploadImage = findViewById(R.id.btn_upload_image);
        btnSubmit = findViewById(R.id.btn_submit);
        feedRef = FirebaseDatabase.getInstance().getReference("feeds");

        btnUploadImage.setOnClickListener(v -> {
            Intent intent = new Intent();
            intent.setType("image/*");
            intent.setAction(Intent.ACTION_GET_CONTENT);
            startActivityForResult(Intent.createChooser(intent, "이미지 선택"),
                    PICK_IMAGE_REQUEST);
        });
        btnSubmit.setOnClickListener(v -> {
            uploadPost();
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null && data.getData() != null) {
            imageUri = data.getData();
            imgPreview.setImageURI(imageUri);
        }
    }
}

```

```

}

private void uploadPost() {
    String title = edtTitle.getText().toString().trim();
    String content = edtContent.getText().toString().trim();
    if (imageUri ==null|| TextUtils.isEmpty(content) || TextUtils.isEmpty(title)) {
        Toast.makeText(this, "이미지와 제목, 내용을 모두 입력해주세요.",
        Toast.LENGTH_SHORT).show();
        return;
    }

    ProgressDialog pd =new ProgressDialog(this);
    pd.setMessage("업로드 중...");
    pd.setCancelable(false);
    pd.show();

    new Thread(() -> {
        try {
            String boundary ="----WebKitFormBoundary"+System.currentTimeMillis();
            URL url =new URL("https://api.cloudinary.com/v1_1/dwyf8emyg/image/upload");
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);
            conn.setUseCaches(false);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type", "multipart/form-data; boundary="+ boundary);
            DataOutputStream request =new DataOutputStream(conn.getOutputStream());
            String uploadPreset ="eco_upload";
            request.writeBytes("--"+ boundary +"\r\n");
            request.writeBytes("Content-Disposition: form-data; name=\"upload_preset\"\r\n\r\n");
            request.writeBytes(uploadPreset +"\r\n");
            InputStream inputStream = getContentResolver().openInputStream(imageUri);
            byte[] buffer =newbyte[1024];
            int bytesRead;
            request.writeBytes("--"+ boundary +"\r\n");
            request.writeBytes("Content-Disposition: form-data; name=\"file\"";
            filename="image.jpg\r\n");
            request.writeBytes("Content-Type: image/jpeg\r\n\r\n");
            while ((bytesRead = inputStream.read(buffer)) !=-1) {
                request.write(buffer, 0, bytesRead);
            }
            request.writeBytes("\r\n--"+ boundary +"--\r\n");
            inputStream.close();
            request.flush();
            request.close();
            Scanner scanner =newScanner(conn.getInputStream());
            StringBuilder response =new StringBuilder();

```

```

        while (scanner.hasNext()) {
            response.append(scanner.nextLine());
        }
        scanner.close();
        JSONObject json = new JSONObject(response.toString());
        String imageUrl = json.getString("secure_url");
        long timestamp = System.currentTimeMillis();
        Sharedpreferences prefs = getSharedpreferences("UserPrefs",
Context.MODE_PRIVATE);
        String userEmail = prefs.getString("userEmail", "사용자");
        double[] latLngHolder = newdouble[2];
        FusedlocationProviderClient locationClient =
LocationServices.getFusedlocationProviderClient(this);
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            runOnUiThread(() -> {
                Toast.makeText(this, "위치 권한이 필요합니다.",
Toast.LENGTH_SHORT).show();
            });
            pd.dismiss();
            return;
        }
        locationClient.getLastlocation().addOnSuccessListener(location -> {
            if (location !=null) {
                latLngHolder[0] = location.getLatitude();
                latLngHolder[1] = location.getLongitude();
            }
            runOnUiThread(() -> {
                DatabaseReference usersRef =
FirebaseDatabase.getInstance().getReference("users");
                usersRef.orderByChild("email").equalTo(userEmail).limitToFirst(1)
                    .addlistenerForSingleValueEvent(new ValueEventListener() {
                        @Override
                        publicvoid onDataChange(@NonNull DataSnapshot snapshot) {
                            if (snapshot.exists()) {
                                for (DataSnapshot userSnap : snapshot.getChildren()) {
                                    FeedModel post =new FeedModel(userEmail,
imageUrl, content, timestamp, title,
latLngHolder[0], latLngHolder[1]);
                                }
                            }
                        }
                    })
                    .push().setvalue(post).addOnCompleteListener(task -> {
                        pd.dismiss();
                        if (task.isSuccessful()) {
                            Toast.makeText(WritePostActivity.this,
                            "게시물이 업로드되었습니다.", Toast.LENGTH_SHORT).show();
                            finish();
                        }
                    });
            }
        });
    }
}

```

```
        } else {
            Toast.makeText(WritePostActivity.this,
                           "업로드 실패", Toast.LENGTH_SHORT).show();
        }
    });
    break;
}
} else {
    pd.dismiss();
    Toast.makeText(WritePostActivity.this, "사용자 정보를
찾을 수 없습니다.", Toast.LENGTH_SHORT).show();
}
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
    pd.dismiss();
    Toast.makeText(WritePostActivity.this, "에러: "+error.getMessage(), Toast.LENGTH_LONG).show();
}
});
});
}
});
}
} catch (Exception e) {
    runOnUiThread(() -> {
        pd.dismiss();
        Toast.makeText(this, "에러: "+e.getMessage(), Toast.LENGTH_LONG).show();
    });
}
}).start();
}
}
```

1-6 마이페이지

- MyPageFragment

```
package com.inhatc.ecocheckinapp
```

```
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.GridView;
import android.widget.ImageButton;
import android.widget.TextView;
```

```

import android.widget.ArrayAdapter;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import java.util.ArrayList;
import java.util.List;

public class MyPageFragment extends Fragment {
    private TextView txtUserName, txtPostCount, txtTotalStats;
    private GridView gridPosts;
    private final ArrayList<String> titleList = new ArrayList<>();
    private ArrayAdapter<String> adapter;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                            @Nullable ViewGroup container,
                            @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_mypage, container, false);
        ImageButton btnSettings = view.findViewById(R.id.btn_settings);

        btnSettings.setOnClickListener(v -> {
            Intent intent = new Intent(getActivity(), SettingsActivity.class);
            startActivity(intent);
        });

        txtUserName = view.findViewById(R.id.user_realname);
        txtPostCount = view.findViewById(R.id.post_count);
        txtTotalStats = view.findViewById(R.id.txt_total_stats);
        gridPosts = view.findViewById(R.id.grid_posts);

        adapter = new ArrayAdapter<>(requireContext(),
                android.R.layout.simple_list_item_1, titleList);
        gridPosts.setAdapter(adapter);
        SharedPreferences prefs = requireContext().getSharedPreferences("UserPrefs",
                Context.MODE_PRIVATE);

        String userEmail = prefs.getString("userEmail", null);
        if (userEmail == null) return view;
        DatabaseReference usersRef = FirebaseDatabase.getInstance().getReference("users");
        usersRef.orderByChild("email").equalTo(userEmail).get().addOnSuccessListener(userSnapshot -> {
            if (userSnapshot.exists()) {

```

```

        DataSnapshot userSnap = userSnapshot.getChildren().iterator().next();
        String name = userSnap.child("name").getValue(String.class);
        txtUserName.setText(name !=null ? name : "이름 없음");
        FirebaseDatabase.getInstance().getReference("feeds")
            .orderByChild("email").equalTo(userEmail)
            .get()
            .addOnSuccessListener(feedSnapshot -> {
                int postCount =0;
                for (DataSnapshot postSnap : feedSnapshot.getChildren()) {
                    String title = postSnap.child("title").getValue(String.class);
                    if (title !=null) {
                        titleList.add("📌 "+ title);
                        postCount++;
                    }
                }
                txtPostCount.setText(String.valueOf(postCount));
                adapter.notifyDataSetChanged();
            });
    }

    FirebaseDatabase.getInstance().getReference("CheckIns")
        .orderByChild("email").equalTo(userEmail)
        .get()
        .addOnSuccessListener(checkSnapshot -> {
            int checkinCount =0;
            for (DataSnapshot checkSnap : checkSnapshot.getChildren()) {
                String place = checkSnap.child("place").getValue(String.class);
                if (place !=null) {
                    titleList.add("📍 "+ place);
                    checkinCount++;
                }
            }
            txtTotalStats.setText("총 실천 횟수: "+ checkinCount +"회");
            adapter.notifyDataSetChanged();
        });
    }
);
return view;
}
}

```