

5/13 회의록

≡ 태그	팀회의
📅 날짜	@2025년 5월 13일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

회의 안건

1. 발표자료 피드백
2. 중간보고서 피드백

제안 발표 질문 피드백 반영

질문 내용	지난 답변
투표의 기준이 어떻게 되나요?	좋아요 수/ 싫어요 비율
재난 지역 리스트 선정 프로세스	GPS 기반 운영자 중심 선정
가게 선정 기준의 초반 운영 어려움	이용자 수를 고려한 선정 기준 조절, 추후 MAU 반영 확장 예정
서비스 운영 초반 가게 리스트의 부족 문제	초반에 선한 영향력 가게 Api를 이용해 선행 가게 리스트를 운영자가 미리 올려 놓을 예정
포인트 가중 이벤트 기간 선정 방법	운영자의 모니터링으로 이벤트 기간 조절
포인트 현금 전환 과정 중 현금 확보 문제	웹페이지 내 광고 노출을 통해 수익금 창출 및 기부처 전달

회의 결과

1. 발표자료 반영 및 이클래스, Github 제출

다음 회의

일정: 202Y년 MM월 DD일 00:00

준비: [다음 회의 때 준비해야 할 사항]

5/19 회의록

☰ 태그	팀회의
📅 날짜	@2025년 5월 19일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

지난 질문 피드백

▼ 사장님한테 연락 가게 하기 (플랫폼에 올라왔어요) / 부정적인 피드백이 있을 때 속상해 할 사장님의 입장

착한 가게 등록 시 유선 상으로 연락, 요청 시 언제든지 등록된 가게(리뷰) 블라인드 처리 (삭제)

▼ JWT 리프레쉬 토큰

JWT 액세스 토큰만 쓰고, 리프레쉬 토큰 없이 구현함. 개발 초기라 복잡도 줄이고, 개발 중 팀 내 로그인 세션 유지 요구가 크지 않아서

▼ 임시 휴무일 반영?

사장에게도 낮은 단계의 Admin 권한 부여, 자기 가게 정보 업데이트 가능하게 추후 확장 고려

▼ UI에서 청소년 소비 불가 가게(술집) 표시할지?

미성년자 가리는 건 가게의 책임, 물론 술집이라는 정보는 표기할 것

▼ 재난지역의 범위를 잡는 기준?

관리자가 뉴스 등 언론 보도를 참고해서 수동으로 결정

- 공신력 있는 자료를 근거로 하기 때문에 임의성이 적고,
- 자동화 API로는 커버 안 되는 지역별/상황별 특수성 반영 가능
- 실제로 공공기관도 뉴스·행정명령 등 공식 안내를 기준으로 지정하는 경우 많음

▼ 프론트 반응형 모바일 말고 다른 화면에서 캡처한거 있는지

(최종보고서에 넣기)

▼ Sns 연동 로그인 기능 구현 여부

작업량 많고 우선순위 밀려 추후 확장 고려

▼ 사용자 지속 사용, 기부를 하기 위한 요소들이 있는지?

지속 사용 유도를 위해 귀여운 캐릭터 성장형 게이미피케이션과 커뮤니티 기능을 도입함.

단순히 '기부'만 하면 반복성이 떨어지지만, 기부/착한소비 활동이 쌓이면 캐릭터가 성장하고 커뮤니티에서 다른 사람들과 소통(인증, 응원, 정보 공유 등)할 수 있음

캐릭터 성장으로 동기 부여 + 커뮤니티에서 인정받고 연결되는 경험 두 가지가 합쳐져서 단순 기부보다 훨씬 더 지속 사용성을 높임

▼ 착한가게에서 가게의 선행활동만 볼것인지? 친절도 맛 등 기본적 요소들도 반영할것인지?

등록 시에는 사용자들의 투표로 결정, 등록 이후 만족도가 낮으면 별점 낮아지고 자연스레 내려갈 것

▼ 앱 스토어에 출시 계획 있는지?

다양한 기기 접근성, 설치 과정 필요없다는 장점으로 반응형 웹 결정.

앱은 작업량 많고 우선순위 밀려 추후 확장 고려

▼ 초반에는 커뮤니티가 제대로 구성 되지 않을텐데 어떻게 할 생각인지

구조적으로 많은 서비스가 공유하는 문제, 당근마켓의 사례처럼 초반에는 공격적으로 홍보

깃허브 업데이트 설명 - 시연

src 폴더에 FE, BE 넣음

일정 정리

5월 26일 : 구현 마무리

5월 26일 ~ 6월 4일 : FE, BE 연동

6월 4일 - 최종 테스트

6월 4일 ~ 11일 : 문서 작성 및 발표 준비

최종발표

▼ (병진) 주제들

프론트 기술스택 보고서에 반영 - 프론트엔드쪽 다루기

최종보고서 노션 동시공유 - 한글에 최대한 양식하고, 노션에 업데이트

최종발표 시간관리 - 연습 때도 실전처럼 천천히 하고, 상호 간 시간 공유되게 하기

주요기능에 집중

회의 결과

1. 가게 db에 사진 추가 (제보 시)
 2. 다음주 월요일까지 구현 완료!
-

다음 회의

일정: 202Y년 MM월 DD일 00:00

준비: [다음 회의 때 준비해야 할 사항]

최종 발표 - 시간관리

README, Src, 최종보고서, 제품구성배포운영자료(1장정도), 시연동영상, OSS프로젝트 결과 및 참여보고서,팀구성원이 함께 찍은 사진 1장(Tnn_팀명_팀사진.jpg), 작품설명용 사진 1~2장, 최종발표_동료팀평가표, 개인별 평가표

<https://wikidocs.net/232570>

질문 리스트

5/21 회의록

☰ 태그	팀회의
📅 날짜	@2025년 5월 21일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

회의 안건

1. 백엔드 역할 분배
2. 진행 발표
3. 프론트엔드 진행 상황 회의

디자인 / 프론트엔드

디자인 진행 상황 보고 - 돈쫌 인증서 발급 페이지, 리뷰상세페이지(댓글), 검색결과화면, 더보기 화면, 예외상, 커뮤니티 게시판 화면

디자인 추가 요구 사항 - 기부 상세 페이지, 기부 포인트 모달 구현

프론트엔드 - 기부 게이미피케이션 일요일까지 구현 완료할 것

백엔드

역할 분배 - ocr 연동 (병진님에게 토스 : 가장 어려운 파트)

시연 - 파일 구조 파악의 어려움

백엔드 완료된 부분 - 회원가입, 로그인, 가게제보로직, 투표기능, OCR 기능(환경구성)

백엔드 해야할 부분 - 나머지 전부 다

진행 발표- 교수님 피드백

자기팀의 최종 발표 : 전체 범위 설정 :→ 분석설계 → 실제기능구현(핵심 기능 설정)

프론트엔드 - 기능 - 작성자 설명서

시퀀스 다이어그램

고유 핵심 기능 - originality 강조

카테고리 재분류

→ 중식, 일식, 패스트푸드, 해산물, 족발/보쌈, 분식, 카페/디저트, 한식, 고기, 양식, 아시안, 기타음식점, 의료시설, 스포츠시설, 생활, 의류/잡화, 미용, 취미/여가, 학원

회의 결과

1. 이번주 일요일까지 개발 완성
 2. 시연 - (백엔드) 마이페이지 / 연동 전담
-

다음 회의

일정: 2025년 5월 28일 17:00

준비: [다음 회의 때 준비해야 할 사항]

개발 완료 및 연동 진행

5/28 회의록

☰ 태그	팀회의
📅 날짜	@2025년 5월 28일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

회의 안건

1. 개발 현황 (프론트, 백엔드)
2. 연동 현황, 배포
3. 문서 작성 역할 분배

연동 진행 상황

회원가입/로그인

프론트에서 찍은 이미지 백엔드에서 ocr 돌려서 소비 인증하고 사업자번호 db에 있으면 착한가게 소비 화면 / 없으면 새로운 착한가게 제보 화면으로 넘어가게 하기

기부 페이지 연동을 어떻게 하나?

- ▼ (희정) 기부 게임 캐릭터 백엔드 연동 (DB구조 확인 필요)

주의 사항

- `userId = 1` 부분은 실제 로그인 사용자 ID로 바뀌어야 함
- 백엔드 API가 `/api/users/{id}/points` GET/POST 로 연결돼 있어야 정상 작동함

```
@GetMapping("/api/users/{id}/points")
public ResponseEntity<Map<String, Integer>> getPoints(@PathVariable Long id)
```

```
@PostMapping("/api/users/{id}/points")
public ResponseEntity<Map<String, Integer>> addPoints(@PathVariable Long id, @RequestBody Map<String, Integer> points)
```


백엔드 DB 포인트 값 수정 로직 (현재 백엔드 환경에 맞게 다시 수정하셔야할 수도)

```
@PostMapping("/api/users/{id}/points")
public ResponseEntity<Map<String, Integer>> addPoints(@PathVariable Long id, @RequestBody Map<String, Integer> body) {
    int delta = body.get("delta"); // 프론트에서 보낸 +10 또는 -10000

    User user = userRepository.findById(id).orElseThrow();
    user.setPoints(user.getPoints() + delta); // 명시적으로 포인트 수정!
    userRepository.save(user); // DB에 저장

    return ResponseEntity.ok(Map.of("updatedPoints", user.getPoints()));
}
```

▼ (희정) 지도 백엔드 연동

목표

1. 백엔드에서 선행 가게 리스트(위도/경도 포함)를 받아오기
2. 카테고리별로 지도 위에 마커를 띄우는 작업

프로세스

- DB에 저장된 가게들을 가져온다
- 카테고리를 선택적으로 쿼리 파라미터로 받아서 필터링한다
- 위도(lat), 경도(lng), 이름(name), 카테고리(category)를 반환한다

Entity: Store.java

```
package com.ossproj.donjjul.domain;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.NoArgsConstructor;

@Entity
@Getter
@NoArgsConstructor
public class Store {
```

```

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

private String name;
private double lat;
private double lng;
private String category;
}

```

Repository: StoreRepository.java

```

package com.ossproj.donjjul.repository;

import com.ossproj.donjjul.domain.Store;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface StoreRepository extends JpaRepository<Store, Long> {
    List<Store> findByCategory(String category); // 카테고리 필터용 메소드
}

```

Controller: StoreController.java

```

package com.ossproj.donjjul.controller;

import com.ossproj.donjjul.domain.Store;
import com.ossproj.donjjul.repository.StoreRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController

```

```

@RequiredArgsConstructor
@RequestMapping("/api/stores")
public class StoreController {

    private final StoreRepository storeRepository;

    @GetMapping
    public List<Store> getStores(@RequestParam(required = false) String category) {
        if (category != null && !category.isEmpty()) {
            return storeRepository.findByCategory(category);
        } else {
            return storeRepository.findAll();
        }
    }
}

```

문서 작성

- 3_1_OSSProj_팀번호_팀명_최종보고서.pdf (다이어그램, 결과물 (각자 작성))
- 3_2_OSSProj_팀번호_팀명_최종발표자료.ppt (결과물 각자 작성)
- 3_3_OSSProj_팀번호_팀명_회의록.pdf → 회정
- 결과 참여 보고서 → 회정
- 3_4_OSSProj_팀번호_팀명_제품구성배포운영자료.pdf
- 3_5_OSSProj_팀번호_팀명_시연동영상.mpeg
- read_me 다시 작성
- 오픈소스 라이선스

회의 결과

1. netrify : 호스트 주소 상세하게 해서 1조와 같지 않게 하기 꼭 !!!!

다음 회의

일정: 202Y년 MM월 DD일 00:00

준비: [다음 회의 때 준비해야 할 사항]

6/2 회의록

☰ 태그	팀회의
📅 날짜	@2025년 6월 2일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

회의 안건

- 1.
- 2.
- 3.

포인트 가져오는 연동 코드 질문

1. fetch로 포인트 연동 시 문제점 : fetch는 비동기 API라 초반에 stage에 맞는 햄스터가 안 뜨는 상황
2. 일단 제가 기본으로 애기햄스터가 나오게끔 코딩 push 해놔서 이제는 포인트에 상관없이 초반엔 무조건 애기 햄스터가 뜨는 상황
3. 포인트도 fetch 결과 값을 가져올 때까지 0포인트로 뜨다가 갱신
4. 대안 : 그대로 갑시다!!

배포

프론트엔드 - netrify

백엔드 - AWS

남은 문서 작업

- 3_1_OSSProj_팀번호_팀명_최종보고서.pdf (다이어그램, 결과물) → 각자 작성))
- 3_2_OSSProj_팀번호_팀명_최종발표자료.ppt (결과물 각자 작성) → 각자 작성
- 3_3_OSSProj_팀번호_팀명_회의록.pdf → 회정

- 결과 참여 보고서 → 병진
- 3_4_OSSProj_팀번호_팀명_제품구성배포운영자료.pdf → 시연, 병진
- 3_5_OSSProj_팀번호_팀명_시연동영상.mpeg →수요일에 다같이 만들기
- read_me 다시 작성 → 희정
- 오픈소스 라이선스 작성 → 희정

→ 역할 분배해야할 것

설후 : 최종보고서 정리, 최종발표자료 최종 완성본 제작, 프론트엔드 수정, 최종발표 프론트엔드(보고서, ppt)

희정 :

- ☒ ~~프론트 배포~~
- ☒ ~~프론트엔드 수정~~
- ☐ readme 재작성
- ☐ 오픈소스 라이선스 작성
- ☒ ~~제품구성배포운영자료 작성 (배포 절차를 작성해라 : 양식x)~~
- ☐ 회의록
- ☒ ~~최종 발표 보고서 작성~~
- ☒ ~~최종 발표 ppt~~
- ☒ ~~slack github 연동,~~

병진

- ☒ ~~작품설명용 개념 설계 사진 그리기(서버, 네트워크(지구본), ...)~~
- ☒ ~~결과 참여 보고서 작성(양식대로)~~
- ☒ ~~최종보고서 백엔드~~
- ☐ 최종발표 백엔드
- ☒ ~~ERD, 클래스 다이어그램~~
- ☒ ~~유스케이스~~
- ☒ ~~시스템 아키텍처~~
- ☒ ~~블록다이어그램~~

시연 : 연동(최우선), 백엔드 배포(최우선), 제품구성배포운영자료 작성 (배포 절차를 작성해
라 : 양식x)

시연동영상

<최종보고서에서 작성해야하는 부분>

- OSS프로젝트 결과 및 참여보고서 (이클래스 제공 양식 참조, 참여학생 및 교수 서명 후 제출)
- 사진자료 : 팀구성원이 함께 찍은 사진 1장(Tnn_팀명_팀사진.jpg), 작품설명용 사진 1~2장(파일명: eg. Tnn_팀명_4_개념설계.jpg) - 시스템 다이어그램 /

▼ 최종보고서에 작성되어 있는 부분

1. 팀원이 함께 찍은 사진 1장 jpg파일 별도제출 (파일명: eg. Tnn_팀명_팀사진.jpg)
2. 작품설명용 사진 1~2장 jpg파일 별도 제출 (파일명: eg. Tnn_팀명_4_개념설계.jpg)

시연 영상

회의 결과

1. (희정) 프론트엔드 수정
 - ☐ 지도 마커 색깔 수정
 - ☒ 프론트엔드 배포 완료
 - ☒ 애기햄스터, 햄스터 누끼 따서 올라가
 - ☒ 카부 상세 페이지 만들가
 - ☒ 카부 페이지 라우팅
 - ☒ 프론트엔드 배포 시작
 2. (설후) 프론트엔드 수정 : 가게 리스트 라우팅
 3. (시연) 수요일까지 연동 및 배포 완료
 4. (병진) 결과 참여 보고서 작성
-

다음 회의

일정: 202Y년 MM월 DD일 00:00

준비: [다음 회의 때 준비해야 할 사항]

6/4 회의록

☰ 태그	팀회의
📅 날짜	@2025년 6월 4일

향후 발전사항에 선행을 받은 사람들이 등록 및 인증 할 수 있는거 만들기

가게 제보 페이지에 돈쫌 내줘야하는 이유 << 선행 단어 넣어주기 ('선행 사유'와 '후기'를 분리)

프론트엔드 설치 방법

1. 터미널에 `npm -v` 작성 : 버전이 제대로 뜨면 정상적으로 설치된 것
2. frontend 폴더에서 `npm install`
3. `npm start`

시연님 연동 상황 (수~목)

- ✓ 가게제보 후, 투표게시판에 제보 내역 연동
- ✓ 착한가게 > 상세정보 연동

배포

프론트엔드 - npm build 중 오류 수정 중

백엔드 - 연동 후 마저 작업

일정

< 수,목 >

프론트엔드 배포 완료

백엔드 연동 및 배포 완료

<금>

희정, 병진 : 문서 작업 완료

6/9 회의록

☰ 태그	팀회의
📅 날짜	@2025년 6월 9일

▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

회의 안건

1. 백엔드 배포 완료
2. 시연 영상 촬영
3. 발표 대본 작성

백엔드 배포 과정

프론트엔드에서 HTTPS로 로드된 페이지가 백엔드의 HTTP 자원(
`http://13.125.255.84`)에 접근하면서 발생하는 Mixed Content 오류가 지속적으로 발생함.

초기에는 Netlify 프록시 설정을 통해 임시적으로 우회하였으나, 이미지 업로드 시 보안 정책이 강화되며 브라우저 차단이 심화되었음.

이에 따라 근본적인 해결을 위해 백엔드 서버에 HTTPS 설정을 적용하는 것이 필요하다는 결론에 도달함.

```
✖ ▼ Mixed Content: The page at 'page-47892abd8e01b688.js:1'
https://donzzul.netlify.app/camera' was loaded over HTTPS, but requested an
insecure resource 'http://13.125.255.84/api/receipt/process'. This request
has been blocked; the content must be served over HTTPS.

c @ page-47892abd8e01b688.js:1
iX @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
(anonymous) @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
nS @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
i2 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
s7 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
s5 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1

✖ ▼ 업로드 에러: TypeError: Failed to fetch 684-ba1f03b97491ba43.js:1
at c (page-47892abd8e01b688.js:1:2246)
at iX (4bd1b696-9a99fb4dbb7f9f2a.js:1:132394)
at 4bd1b696-9a99fb4dbb7f9f2a.js:1:138472
at nS (4bd1b696-9a99fb4dbb7f9f2a.js:1:18812)
at i2 (4bd1b696-9a99fb4dbb7f9f2a.js:1:133627)
at s7 (4bd1b696-9a99fb4dbb7f9f2a.js:1:159677)
at s5 (4bd1b696-9a99fb4dbb7f9f2a.js:1:159499)
push.6905.window.console.error @ 684-ba1f03b97491ba43.js:1
c @ page-47892abd8e01b688.js:1

await in c
iX @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
(anonymous) @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
nS @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
i2 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
s7 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1
s5 @ 4bd1b696-9a99fb4dbb7f9f2a.js:1

[NEW] Explain Console errors by using Copilot in Edge:
click ⓘ to explain an error. Learn more Don't show again
```

도메인 구매

처음에 시도 → 반복되는 오류로 포기하고 도메인 구매하기로 결심

가비아에서 donzzul.store 구매 완료

Nginx 설치, Nginx 서버 블록 등록 완료

AWS ec2에 Nginx 설치 완료 → 활성화 확인

Nameserver 설정 및 도메인 등록 완료

이름
donzzul.store

A AAAA ANY CAA CNAME DNSKEY DS MX **NS** PTR SOA SRV TLSA TSIG TXT

TTL:
6 hours
TARGET:
ns-1553.awsdns-02.co.uk.

TTL:
6 hours
TARGET:
ns-327.awsdns-40.com.

NS
TTL:
6 hours
TARGET:
ns-1405.awsdns-47.ors.

Let's Encrypt를 통한 SSL 인증서를 발급 완료

그러나 nginx 연결 문제로 5시간 동안 골머리...

https 인증을 포기하고 다시 proxy로 우회

EC2환경을 만지는 과정에서 연결 문제가 해결 되었는지 기존 문제 해결!

컴퓨터에서 정상적으로 작동!

모바일 상에선 인식 실패모달만 뜨게 되었음

회의 결과

1. 컴퓨터 상에선 배포 연결 성공!
2. 시연 - 시연 동영상 촬영, 배포운영자료 작성

다음 회의

일정: 2025년 06월 11일 17:00

준비: [다음 회의 때 준비해야 할 사항]

발표 분배 및 대본 준비