

# 4/21 회의록

☰ 태그	팀회의
📅 날짜	@2025년 4월 21일

## ▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

## 회의 안건

1. 희정 - 기부 게이미피케이션 프로세스 구체화
2. 병진 - AI 리포트 프로세스 구체화
3. 시연 - SNS 웹페이지 상 공유 프로세스 구체화 (기술적으로)
4. 설후 - 해시 텍스트 비교 프로세스 구체화

어떤 라이브러리, 어떤 툴을 써야하는지, 간단하게 툴 사용 속지!

## ▼ (희정) 기부 게이미피케이션 프로세스 구체화

툴	역할	MySQL과 연동 가능?	백엔드 부담	코멘트
Lottie	애니메이션 표시	✅ 필요 없음	💎 없음	단순 프론트 전용, JSON만 렌더링
Framer Motion	터치/애니메이션 UI	✅ 필요 없음	💎 없음	프론트만 쓰면 돼서 부담 X
Zustand / Redux	상태 관리 (포인트, 캐릭터 등)	✅ 프론트 전용	💎 없음	API만 만들어두면 프론트에서 자체 관리
React (Vite + Tailwind)	모바일 웹 UI 구현	✅ 백엔드 연결 쉬움	💎 없음	Rest API만 잘 만들어두면 됨

## ▼ 백엔드 DB

### 1. 백엔드 처리 API

- `POST /point/add` : 포인트 추가
- `GET /character` : 현재 캐릭터 상태 조회

- `POST /donate` : 기부 처리 (10,000점 도달 시)

## 2. DB 테이블 구조

- 사용자

```
CREATE TABLE users (  
  id INT PRIMARY KEY,  
  nickname VARCHAR(50),  
  total_point INT DEFAULT 0,  
  current_character_stage ENUM('baby', 'teen', 'adult') DEFAULT 'baby'  
);
```

- 포인트 히스토리

```
CREATE TABLE point_log (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  user_id INT,  
  change_amount INT,  
  reason VARCHAR(255),  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

- 기부 로그

```
CREATE TABLE donations (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  user_id INT,  
  amount INT,  
  character_stage_at_donation ENUM('baby', 'teen', 'adult'),  
  donated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

▼ 프론트 예시 코드

### ✨ 사용법 간단 요약

#### 1. 설치 명령어 (터미널)

```
bash  
복사편집
```

```
npm i lottie-react framer-motion zustand tailwindcss
```

## 2. `animations/hamster_adult.json` 위치에 Lottie JSON 파일 추가

- <https://lottiefiles.com/search?q=hamster> 에서 귀여운 캐릭터 애니메이션 다  
운 가능

## 3. 캐릭터 클릭 → 500pt 누적 → 10,000pt 넘으면 기부 UI 표시

```
// 필요한 라이브러리 설치
// 터미널에서 아래 명령어 실행
// npm i lottie-react framer-motion zustand tailwindcss

import { useEffect, useState } from 'react';
import { motion } from 'framer-motion';
import Lottie from 'lottie-react';
import adultHamster from './animations/hamster_adult.json'; // 애니메이션 j
import { create } from 'zustand';

// Zustand 상태 저장소
const useStore = create((set) => ({
  point: 0,
  stage: 'baby',
  addPoint: (val) => set((state) => ({
    point: state.point + val,
    stage: state.point + val >= 10000 ? 'adult' : state.stage
  })),
  reset: () => set({ point: 0, stage: 'baby' })
}));

export default function DonationHamster() {
  const { point, stage, addPoint, reset } = useStore();
  const [showDonate, setShowDonate] = useState(false);

  useEffect(() => {
    if (point >= 10000) {
      setShowDonate(true);
    }
  }, [point]);
```

```




return (
  <div className="flex flex-col items-center justify-center min-h-screen">
    <p className="text-lg font-semibold mb-2">기부 포인트: {point} pt</p>

    <motion.div
      whileTap={{ scale: 0.9 }}
      onClick={() => addPoint(500)}
      className="w-52 h-52 cursor-pointer"
    >
      <Lottie animationData={adultHamster} loop={true} />
    </motion.div>

    {showDonate && (
      <motion.div
        className="bg-white p-4 rounded-xl shadow-md mt-4"
        initial={{ opacity: 0 }}
        animate={{ opacity: 1 }}
      >
        <p className="mb-2">10,000pt 도달! 자동 기부 진행 중 🎉</p>
        <button
          className="bg-yellow-400 px-4 py-2 rounded-full text-white"
          onClick={() => {
            // TODO: 기부 API 연결
            reset();
            setShowDonate(false);
          }}
        >기부 완료</button>
      </motion.div>
    )}
  </div>
);
}

```

### 포함된 기술 요소

-  **Lottie**로 캐릭터 애니메이션 렌더링
-  **Framer Motion**으로 터치 애니메이션 ( `whileTap` )
-  **Zustand**로 포인트 상태 관리

-  **Tailwind CSS**로 스타일링 (모바일 친화적 UI)

## 각 툴 / 프레임워크 설명

### 1. Lottie: 애니메이션을 웹에 넣는 가장 쉬운 방법

- 디자이너가 만든 **After Effects** 애니메이션을 **JSON**으로 저장해서 웹/앱에서 실행할 수 있게 해주는 라이브러리.
- `.gif` 나 `.mp4` 보다 훨씬 가볍고 선명하게 동작함.

<https://lottiefiles.com>>> 여기서 원하는 애니메이션 파일을 json 형태로 다운!

#### 언제 써야 해?

- 캐릭터 성장, 응원 메시지, 버튼 누른 후 동작 등에 **귀여운 움직임**을 넣고 싶을 때!

 근데 Lottie 파일에 있는 파일로만 사용할 수 있음

좀 안 이쁨 ㅊ


그래서 **PNG 그대로 쓰고 애니메이션 효과만 줄 수 있는 방법?**

그럴 땐 ➤ **Framer Motion**으로 애니메이션 주기!

정적 이미지를 흔들거나, 점프시키거나, 투명도 조절하는 등 "가짜 애니메이션"은 가능해.

```
jsx
복사편집
import { motion } from 'framer-motion';
import hamster from './images/baby_hamster.png';

<motion.imgsrc={hamster}
  alt="Baby Hamster"
  animate={{ rotate: [0, 5, -5, 0] }}
  transition={{ repeat: Infinity, duration: 2 }}
  className="w-40"
/>
```

 위 코드는 햄스터 이미지가 살짝 좌우로 흔들리는 효과를 줘!

→ Lottie는 아니지만, 사용자 입장에서는 "움직이는 햄스터처럼" 보임 🐹✨

## 2. Framer Motion: 부드러운 터치·움직임 애니메이션

- React 컴포넌트에 **자연스러운 애니메이션을 간단히 적용**할 수 있는 라이브러리.
- 드래그, 클릭, 페이드 인/아웃 등을 손쉽게 구현 가능

### 📌 언제 써야 해?

- 캐릭터 터치, 버튼 누르기, 부드러운 화면 전환 등 UX를 높이고 싶을 때!

▼ 예시

### "먹이주기 버튼" → 밥 먹는 애니메이션 실행

#### 🔄 상태 전환을 통해 컴포넌트 변경하기

```
import { useState } from 'react';
import { AnimatePresence, motion } from 'framer-motion';
import Lottie from 'lottie-react';
import idleHamster from './animations/hamster_idle.json';
import eatingHamster from './animations/hamster_eating.json';

export default function FeedHamster() {
  const [isEating, setIsEating] = useState(false);

  const handleFeed = () => {
    setIsEating(true);
    setTimeout(() => setIsEating(false), 3000); // 3초 후 원상복귀
  };

  return (
    <div className="flex flex-col items-center">
      <AnimatePresence mode="wait">
        {isEating ? (
          <motion.div
            key="eating"
            initial={{ opacity: 0 }}
            animate={{ opacity: 1 }}
            exit={{ opacity: 0 }}
          >
```

```

    >
    <Lottie animationData={eatingHamster} loop={false} />
  </motion.div>
) : (
  <motion.div
    key="idle"
    initial={{ opacity: 0 }}
    animate={{ opacity: 1 }}
    exit={{ opacity: 0 }}
  >
    <Lottie animationData={idleHamster} loop={true} />
  </motion.div>
)}
</AnimatePresence>

<button
  onClick={handleFeed}
  className="mt-4 bg-orange-400 text-white px-4 py-2 rounded-full
  >
    🍲 먹이주기
  </button>
</div>
);
}

```

## ✅ 위 코드가 하는 일

- `isEating` 상태가 `true` 이면 밥먹는 애니메이션( `eatingHamster` ) 실행
- 3초 후 자동으로 다시 기본 상태( `idleHamster` )로 복귀
- `Framer Motion` 의 `AnimatePresence` 가 부드러운 전환을 도와줘

## 🔄 확장 아이디어

- 먹이 줄 때 포인트도 +500점
- 밥먹는 애니메이션 끝나면 말풍선 `"냠냠 고마워!"` 등장
- 먹이주기 횟수에 따라 다른 애니메이션 재생

## 📦 준비물

- 밥먹는 Lottie 애니메이션 ( `.json` )

- idle 상태 애니메이션 (or 정적 이미지)

필요하면 Lottie 파일도 추천해줄게! 직접 넣고 싶으면 어떤 동물(고양이/햄스터/강아지 등) 어떤 스타일 원해? 🐱🐹🐶

## 3. Zustand: 초간단 상태관리 (Redux보다 100배 간단)

### 💡 이게 뭐야?

- React 앱에서 데이터를 공유하고 관리하는 도구야.
- Redux보다 훨씬 가볍고 설정이 필요 없어!

### 📌 언제 써야 해?

- 포인트 저장, 캐릭터 성장 상태, 기부 여부 같은 **공통된 상태를 관리**하고 싶을 때!
- 프론트엔드 계의 데이터베이스. RAM 느낌

항목	백엔드 DB (MySQL 등)	Zustand (프론트 상태 저장소)
위치	서버 (백엔드)	브라우저 (프론트엔드 메모리)
역할	데이터 영구 저장	사용자 인터랙션 기반 일시적 저장
예시	회원정보, 기부기록 등	현재 포인트, 캐릭터 상태 등
접근 방법	API (REST, GraphQL 등)	<code>useStore()</code> 혹은 바로 접근
지속성	페이지 새로고침해도 유지	새로고침하면 날아감 (기본 설정)

### 🎯 왜 Zustand를 '프론트 DB'처럼 쓰는 거야?

- 매번 백엔드 호출하면 느림 → 자주 바뀌는 값은 클라이언트에서 바로 관리
- 예: 버튼 누르면 500pt 증가 → 서버 안 가고 바로 보여줌 → UX 향상
- 나중에 백엔드 연동은 `POST /point` 로 한 번에 처리

## 4. Tailwind CSS: 클래스만 쓰면 예쁜 UI가 똑딱

### 💡 이게 뭐야?

- CSS를 직접 쓰지 않고, **HTML에 클래스를 붙여서 바로 스타일링**하는 방법이야.
- ex: `text-xl`, `bg-pink-100`, `rounded-full`, `shadow-md` 등

### 📌 언제 써야 해?

- 반응형 UI, 모바일 웹에서 빠르게 예쁜 UI 만들고 싶을 때 최고!



요소	일반 CSS	Tailwind
캐릭터 UI	직접 클래스 스타일 정의 필요	<code>rounded-full</code> , <code>bg-yellow-200</code> 같은 조합만 쓰면 바로 적용
점수 배지	위치, 폰트, 색 따로 지정해야 함	<code>text-sm text-white bg-pink-400 px-2 py-1 rounded</code> 한 줄로 끝
반응형(모바일 대응)	미디어 쿼리 직접 작성해야 함	<code>md:text-lg</code> , <code>sm:hidden</code> 등 클래스 하나로 처리
애니메이션 적용	keyframes + 클래스 따로 정의	<code>animate-bounce</code> , <code>animate-ping</code> 등 바로 사용 가능

## 다음에 할 일

- ☐ 기부 API( `POST /donate` ) 백엔드 연동
- ☐ 캐릭터 단계 따라 JSON 교체 (baby, teen, adult)
- ☐ 포인트 쌓일 때마다 성장 연출 추가

→ lottie 버리고, 버튼 프로세스 구축하기 (토스 고양이 키우기 기능 어떤식으로 구현했는지 조사)

## ▼ (병진) AI 개인 종합 리포트

### 리포트 개념 (개인 종합 리포트)

사용자의 누적 소비 인증을 바탕으로 착한 소비 성향, 사회적 가치 기여도, 테마별 소비 패턴 등을 분석해서 보여주는 방식

## 흐름

### 1. 마이페이지에 등록된 소비 인증들 전부 가져오기

- 인증 텍스트/사진/위치/가게명 등

### 2. AI 분석 처리

- 인증 텍스트 기반 키워드 및 가치 추출
- 전체 소비 중 "사회적 가치가 높은 소비" 비율
- 테마 분류: 예) 지역상권 / 친환경 / 사회적 기업 등

### 3. 리포트 생성

- 전체 기여 요약 문장
- 가치 분포 차트
- 대표 키워드 클라우드
- 소비 스타일 요약 (ex. "지역 소상공인 중심의 실천가 타입")

## 라이브러리/툴 정리 (업데이트)

목적	툴	설명
NLP 분석	KeyBERT , transformers	키워드 추출, 요약, 감성분석
분류	scikit-learn 또는 rule-based	소비를 카테고리별로 분류
시각화	matplotlib , plotly , wordcloud	바 차트, 키워드 클라우드 등
리포트 UI	React	마이페이지 내 리포트 렌더링

## 예시 구성 (UI 상)

| 마이페이지 > 나의 소비 리포트

- 나의 총 소비 인증 수: 15회
- 가치별 분포 (바 차트)
  - 지역상권: 9회
  - 친환경: 3회
  - 사회적기업: 2회
- AI 요약 문장:

| “당신은 주로 지역 상권에서의 긍정적 소비 경험을 바탕으로 사회적 가치를 실천하는 소비자입니다.”

<기능후보로 잡고 넘어가기>

1. 기준 명확화 : 착한 선한 영향력 (결식 아동) / 지역 몇 번 활성화했어요

▼ (설후) 해시 프로세스

- phash : 이미지를 해시값으로 변환하여 유사도 검증하는 방법
- 비교대상인 "대표영수증"이 필요 → 사용자가 최초 제보할때 등록하는 영수증을 "대표 영수증"으로 설정
- 대표 영수증도 검증방식이 필요함 ! → OCR + 상호명과 결제일시 직접 입력 비교 방법 사용 → 제보시기와 결제일시의 차이가 크다면 영수증 통과 x →
- 제보시기와 결제일시 비교 어떻게 ? → OCR에서 결제일시 텍스트 추출 (정규식으로 파싱), 제보시기는 `datetime.now()`로 바로 저장 하여 시간차 계산 → 시간 기준은 별도로 지정해야할것 같다. 예를들면 일주일 이내 ?
- 대표 영수증 인증이 끝났으면 db에 저장한다.
- 추후 구매 영수증에 대한 인증 방법 (phash)
  - 프로세스
    - 사용자가 영수증 이미지 업로드
    - OCR →사업자 번호 추출 → 정규식으로 파싱
    - 사업자 번호(가게ID)를 기준으로 db에서 이미지 phash값 가져오기
    - 새롭게 등록되는 영수증 이미지에서 pHash 계산
    - 두 영수증 해시 거리 비교
    - 기준 거리 이하이면 → 인증 성공  
이상이면 → 보류 또는 재검토
  - 유사도 판단 기준
    - 5 이하면 같은 영수증
    - 6 ~10 이상이면 다른 영수증으로 판단.
  - 해시로 통과된 영수증은 다시한번 OCR로 결제일시 검증
- 영수증 phash db 예시
  - DB 구조 예시

사업자번호(가게ID)	가게명	지점	기준 pHash
123-45-67890	착한김밥	강남점	3a8fcf09d65b1a12
678-90-12345	선행카페	본점	f0e7ab3cd8123d21

1. 제보시기 기준 정하기 : 소비 (3일내) / 최초 (일주일 or 한 달)
2. 유사도 판단할 때, 같은 가게지만 메뉴나 주문일자가 달라졌을 때 유사도에 미치는 정도를 확인

## ▼ (시연) 공유 프로세스

html→png 변환, js로 이미지 자동 다운로드(download.js), 클립보드 복사 (navigator.clipboard.writeText), 반응형 디자인 모두 프론트엔드에 들어가는 기술들 (js, html)

1. 사용자가 포인트 누적으로 기부 완료  
↓
2. 프론트에서 기부 완료 메시지 + 시각 요소(캐릭터/트리 등) 보여줌  
↓
3. [html2canvas]로 해당 화면(DOM)을 이미지로 캡처  
↓
4. [download.js] 또는 <a download>로 이미지 자동 다운로드  
↓
5. [navigator.clipboard.writeText()]로 링크 자동 복사 (선택)  
↓
6. 사용자에게 "이미지 저장됨! 공유하기 버튼" 제공  
↓
7. 사용자가 SNS 버튼 클릭 → 공유창 오픈 (카카오톡, 트위터, 인스타 등)

## ✅ 예시 UI 구성 ( #donation-card )

html

코드 복사

```
<div id="donation-card" class="p-4 rounded-lg shadow-md bg-white w-[300px]">
  <h2 class="text-xl font-bold">기부 완료 🎉</h2>
  <p>캐릭터: <span class="font-semibold">산돌이</span></p>
  <p>누적 포인트: 10,000P</p>
  
</div>
```

## ✅ JS 예시 코드 흐름

```

js
코드 복사
// 1. 이미지로 캡처
html2canvas(document.getElementById("donation-card")).then(canvas =>
{
  const imgData = canvas.toDataURL("image/png");

  // 2. 다운로드
  const a = document.createElement("a");
  a.href = imgData;
  a.download = "donjjul-donation.png";
  a.click();

  // 3. 링크 복사 (선택)
  navigator.clipboard.writeText("https://donjjul.com/my-donation").then(() => {
    alert("공유 링크가 복사되었어요! 이미지는 저장되었습니다.");
  });
});

```

## SNS 공유 버튼 (카카오톡 예시)

```

html
코드 복사
<a href="https://twitter.com/intent/tweet?text=돈쫄! 착한소비 기부 완료🙌&url=https://donjjul.com/my-donation" target="_blank">
  <button class="btn-share">트위터에 공유하기</button>
</a>

```

- 착한 소비 공유버튼 :
- 소비 기록 공유버튼 : DB에서 가져온 정보가 이미지템플릿을 자동으로 생성할 수 있게 하면 좋을 것 같다 -상장 느낌 : 글씨만 바꿔주면 됨!

## 1. 이미지 템플릿 디자인만들기 (

### 중간발표 : 수행계획서와의 차이점

목표, 선행 목표 - > 무게를 가볍게 두냐마냐 : 구현 정도 차이

1. 개발 내용
2. 프로세스 구체화
3. 데이터 베이스 설계서
2. 플로우차트, 시퀀스 다이어그램 확장
3. 페이지 구조도
4. 디자인 작업 more

백엔드 : 파이프라인 구축, 알고리즘 개발, api 땡겨오기

---

### 회의 결과

1. 회정 : lottie 버리고, 버튼 프로세스 구축하기 (토스 고양이 키우기 기능 어떤식으로 구현했는지 조사)
2. 병진 : 리포트 기능 구현은 구현을 초기 구현 완성 후 재검토
3. 설후 : 유사도 판단할 때, 같은 가게지만 메뉴나 주문일자가 달라졌을 때 유사도에 미치는 정도를 확인
4. 시연 : 이미지 템플릿 디자인만들기
5. 기본 등록 가게 : 착한 영향력 가게 vs 소상공인 데이터셋 비교 (inner join or merge)

---

### 다음 회의

일정: 2025년 4월 23일 17:00

준비: [다음 회의 때 준비해야 할 사항]

시연 : OCR + 해시 + 카메라 공부

병진 : 백엔드 개발 시작(API)

희정 : 발표자료 (+ 회원가입 화면 : 디자인)

설후 : 데이터 전처리

페이지 구조도 (디자인에게 맡기기)