

HEALING FOREST

SSAFY FINAL PROJECT

# Healing Forest

PJT-FINAL-K-HY-GT

박희연

박건택

발표일 | Date. 2024.05.24





# 목차 소개

Table of Contents

01 | 기획 배경 및 목표

02 | 타임라인

03 | 서비스 아키텍처

04 | API 및 DB 설계

05 | 개발 결과

06 | 트러블 슈팅

07 | 프로젝트 진행 후기



## 01. 기획 배경 및 목표

사용자가 꿈고 싶은 중독 요소를 제어하고 목표 달성을 도와주는 서비스

01

**꿈고싶은 중독을 등록하고 목표를 설정**

시작한 날짜로부터 경과한 시간과 목표 달성률 트래킹합니다.

02

**연속으로 금욕한 일 수에 따라 배지 획득**

배지와 점수를 통한 자기보상으로 동기를 부여합니다.

03

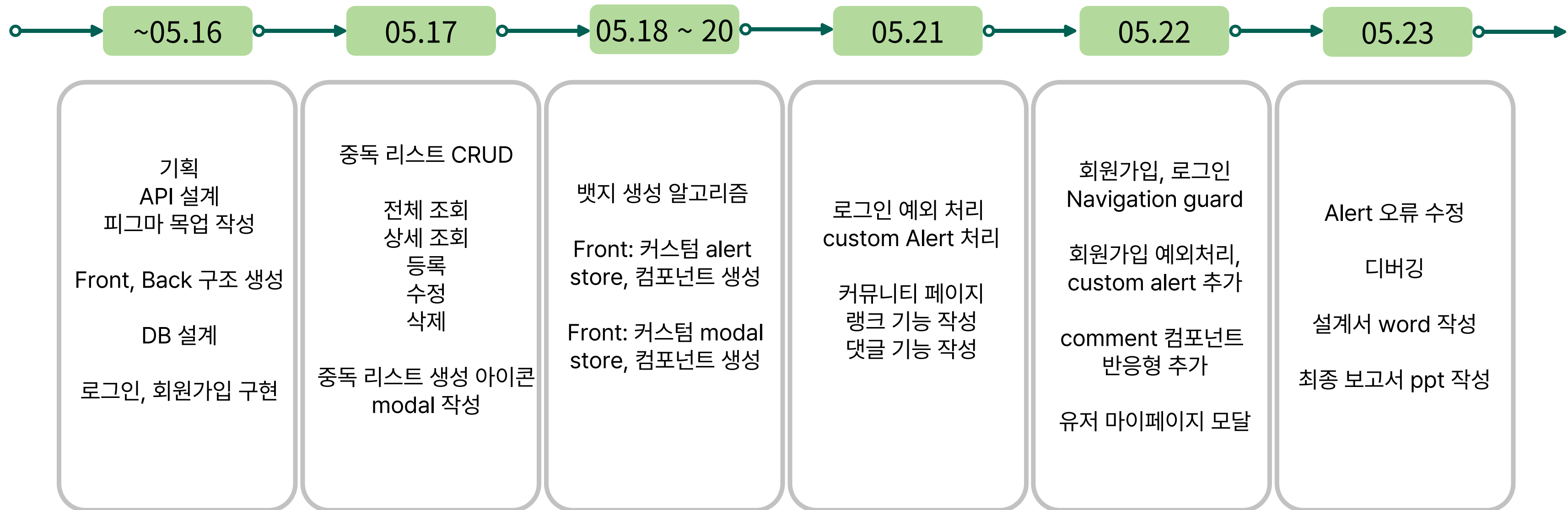
**다른 유저와 소통, 랭킹 시스템을 통한 의지 강화**

상위 랭크와 자신의 랭크를 조회하고, 의견을 주고받으며 의지를 개선시킵니다.



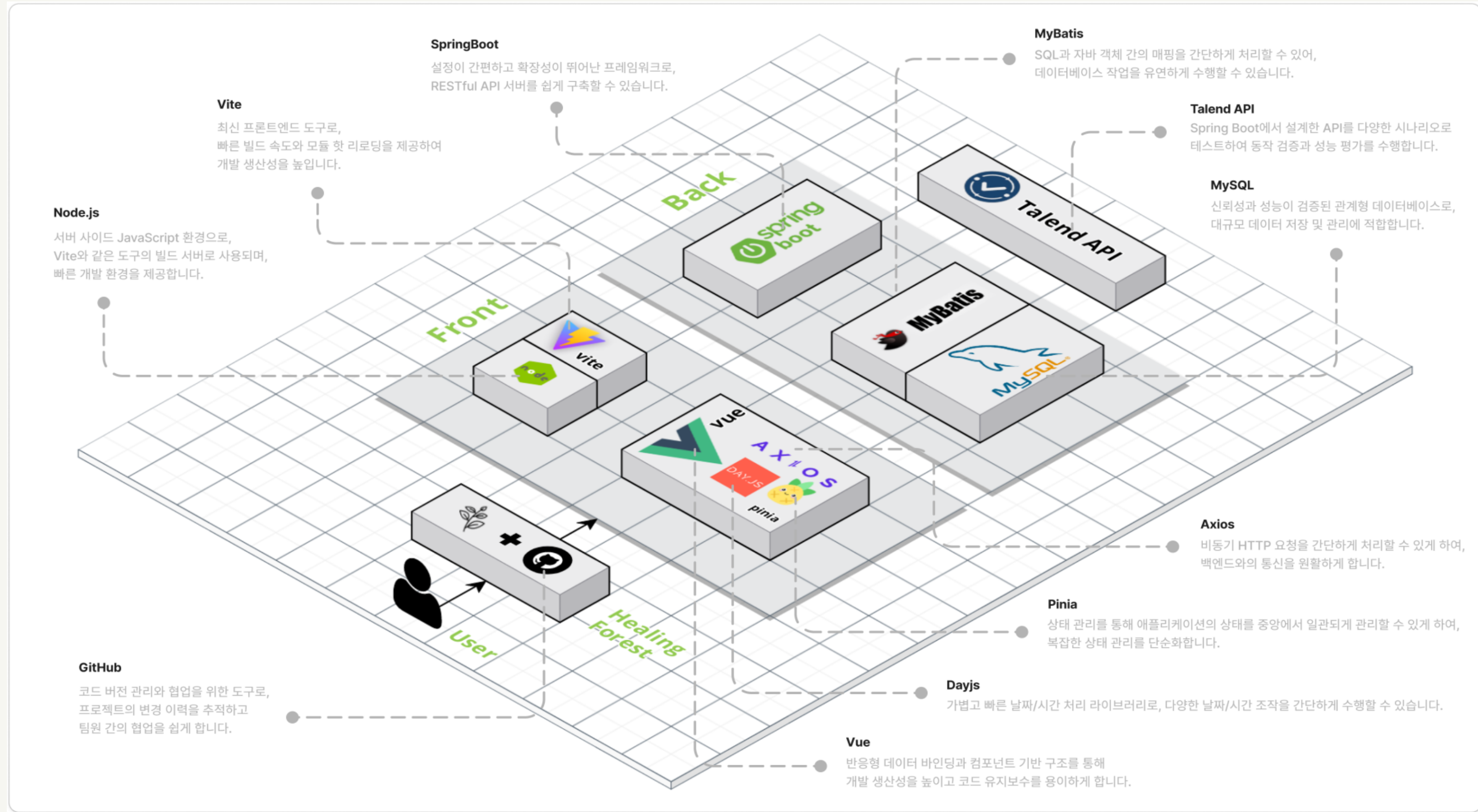
## 02. 타임라인

날짜별 개발 진행 사항



# 03. 서비스 아키텍처

## 개발 환경 및 기술 스택





# 04. API 및 DB 설계

## API 설계

### API

Aa 분류	≡ 항목	≡ router	≡ API URL	⌵ method
home	최초 화면	/	/	GET
user	회원가입	/signup	/api-user/signup	POST
	로그인	/login	/api-user/login	GET
	로그아웃			GET
addiction_list	등록	/addiction/create	/api-addiction/addiction	POST
	전체 조회	/addiction	/api-addiction/addiction	GET
	상세 조회	/addiction/:id	/api-addiction/addiction/:id	GET
	수정	/addiction/update/:id	/api-addiction/addiction/:id	PUT
	삭제		/api-addiction/addiction/:id	DELETE
badge	조회	/addiction/:id	/api-badge/badge/:id	GET
	등록	/addiction/:id	/api-badge/badge/:id	POST
community	rank 조회	/community	/api-community/rank	GET
	내 점수 조회	/community	/api-community/myscore	GET
	댓글 전체 조회	/community	/api-community/comment	GET
	댓글 등록	/community	/api-comunity/comment	POST
	댓글 수정	/community	/api-community/comment/:id	PUT
	댓글 삭제	/community	/api-community/comment/:id	DELETE

## 04. API 및 DB 설계

DB 설계

user			
🔑	user_id	varchar(20)	NOT NULL
	password	varchar(30)	NOT NULL
	nickname	varchar(15)	NOT NULL
	user_image_url	varchar(255)	NOT NULL
	user_score	int	NOT NULL

addiction_list			
🔑	addiction_id	int	NOT NULL
🔑	user_id	varchar(20)	NOT NULL
	title	varchar(30)	NOT NULL
	start_time	timestamp	DEFAULT now()
	target_time	int	NOT NULL
	end_time	timestamp	NOT NULL
	addiction_score	int	DEFAULT 0
	success	boolean	DEFAULT false

comment			
🔑	comment_id	int	AUTO_INCREMENT
🔑	user_id	varchar(100)	NOT NULL
	nickname	varchar(20)	NOT NULL
	content	varchar(100)	NOT NULL
	createDate	timestamp	DEFAULT now()

badge			
🔑	badge_id	int	AUTO_INCREMENT
🔑	addiction_id	int	NOT NULL
🔑	user_id	varchar(20)	NOT NULL
	badge_date	int	NOT NULL
	badge_score	int	NOT NULL
	badge_image_url	varchar(255)	NOT NULL



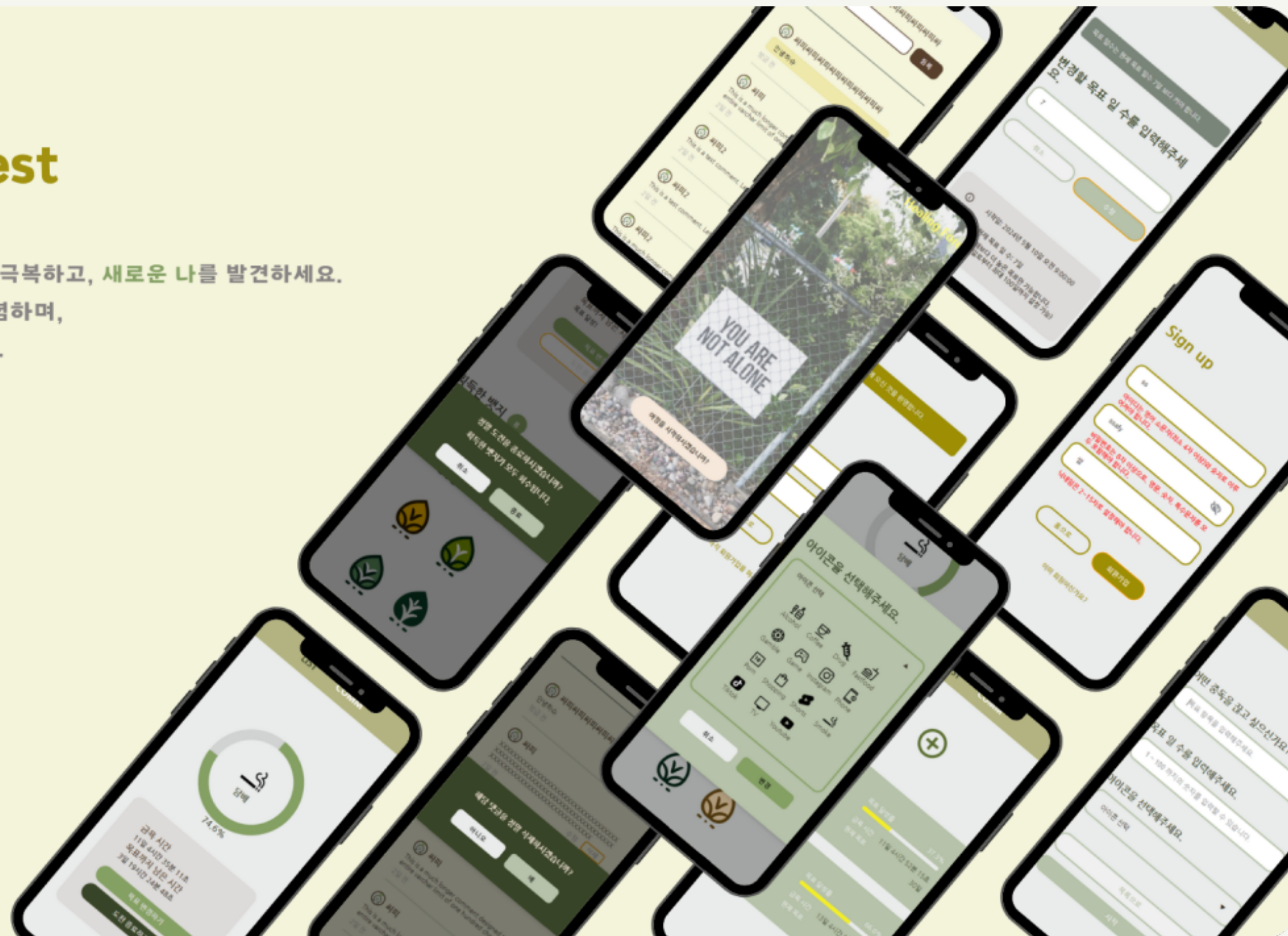
## 05. 개발 결과

Mock-up

### Healing Forest

힐링 포레스트에서 **중독**을 극복하고, **새로운 나**를 발견하세요.  
매일의 성공을 배지로 기념하며,  
**성장**을 확인할 수 있습니다.

함께 시작해볼까요?







# 05. 개발 결과

핵심 기능

## I. 중독 관리 리스트 CRUD

LIST

COMM

+

알코올

목표 달성률

57%

금속 시간

4일 1시간 13초

현재 목표

7일

핸드폰

목표 달성률

30%

금속 시간

7시간 12분

현재 목표

1일

LIST

COMM

어떤 중독을 끊고 싶으신가요 ?

목표 일 수를 입력해주세요.

1 ~ 100 까지의 숫자를 입력할 수 있습니다.

아이콘을 선택해주세요

목록으로

시작

LIST

COMM

변경할 목표 일 수를 입력해주세요.

7일

취소

수정

① 시작일: 2024년 5월 8일 오후 2시 55분 40초

시작일로부터 최대 100일까지 설정 가능하므로, 현재 95일까지 변경 가능합니다.

LIST

COMM

알코올

57%

획득한 배지 4

금속 시간

목표까지 남은 시간

4일 1시간 13초

2일 22시간 59분 47초

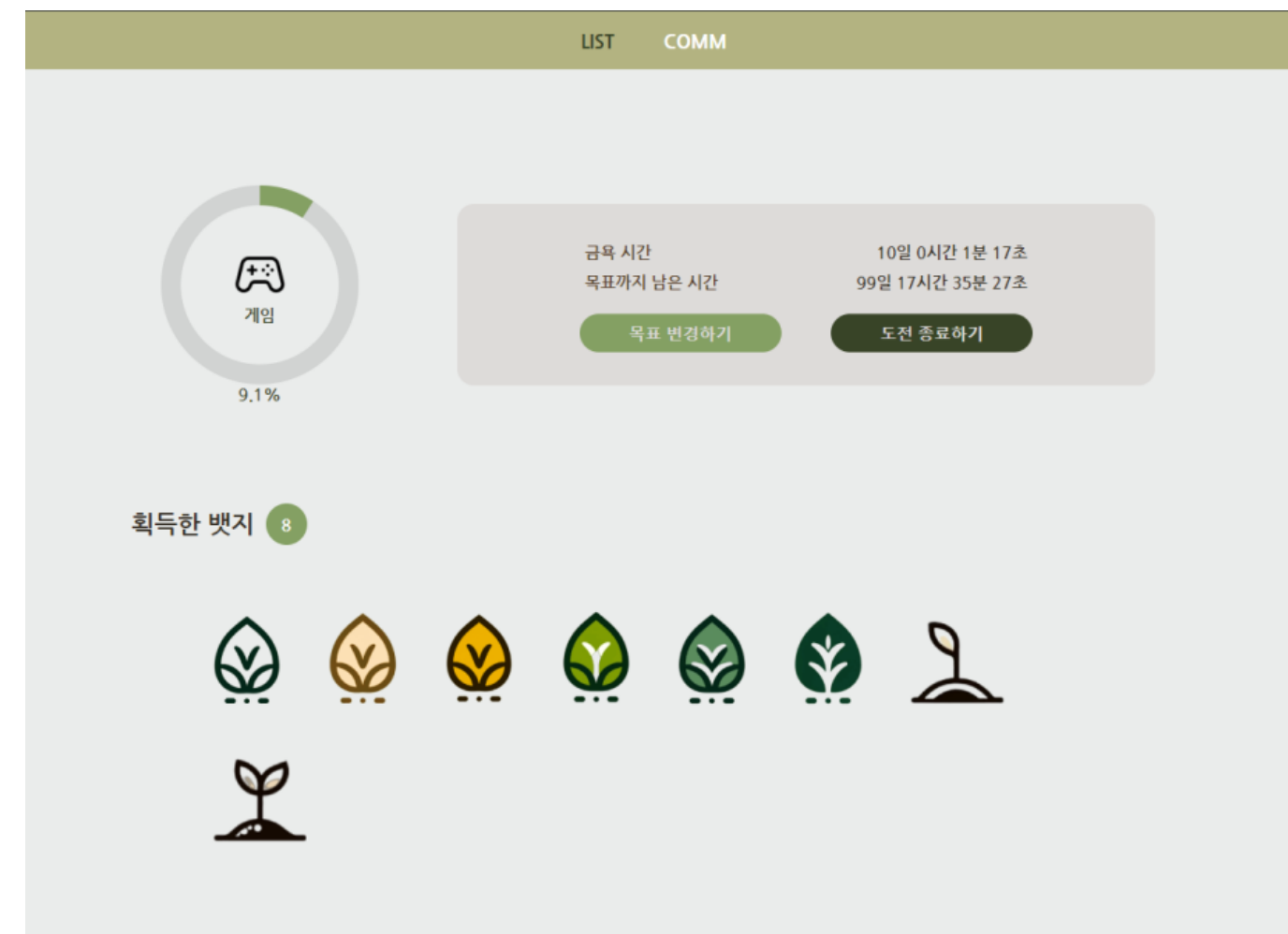
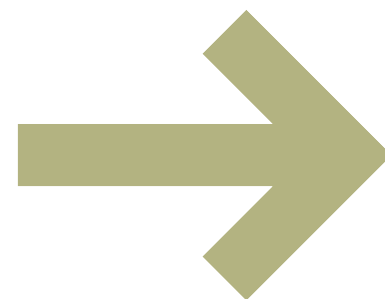
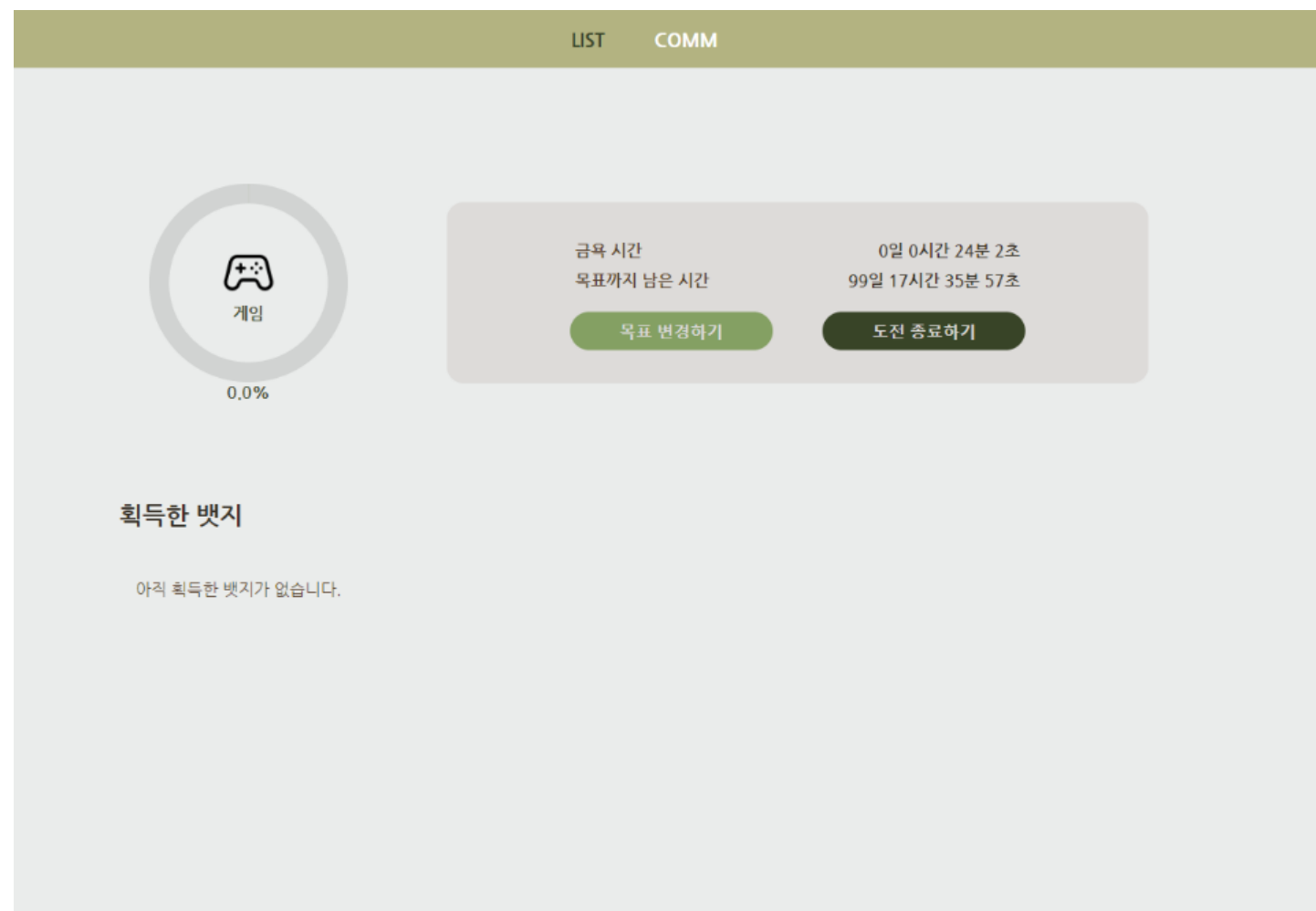
목표 변경하기

도전 종료하기

# 05. 개발 결과

핵심 기능

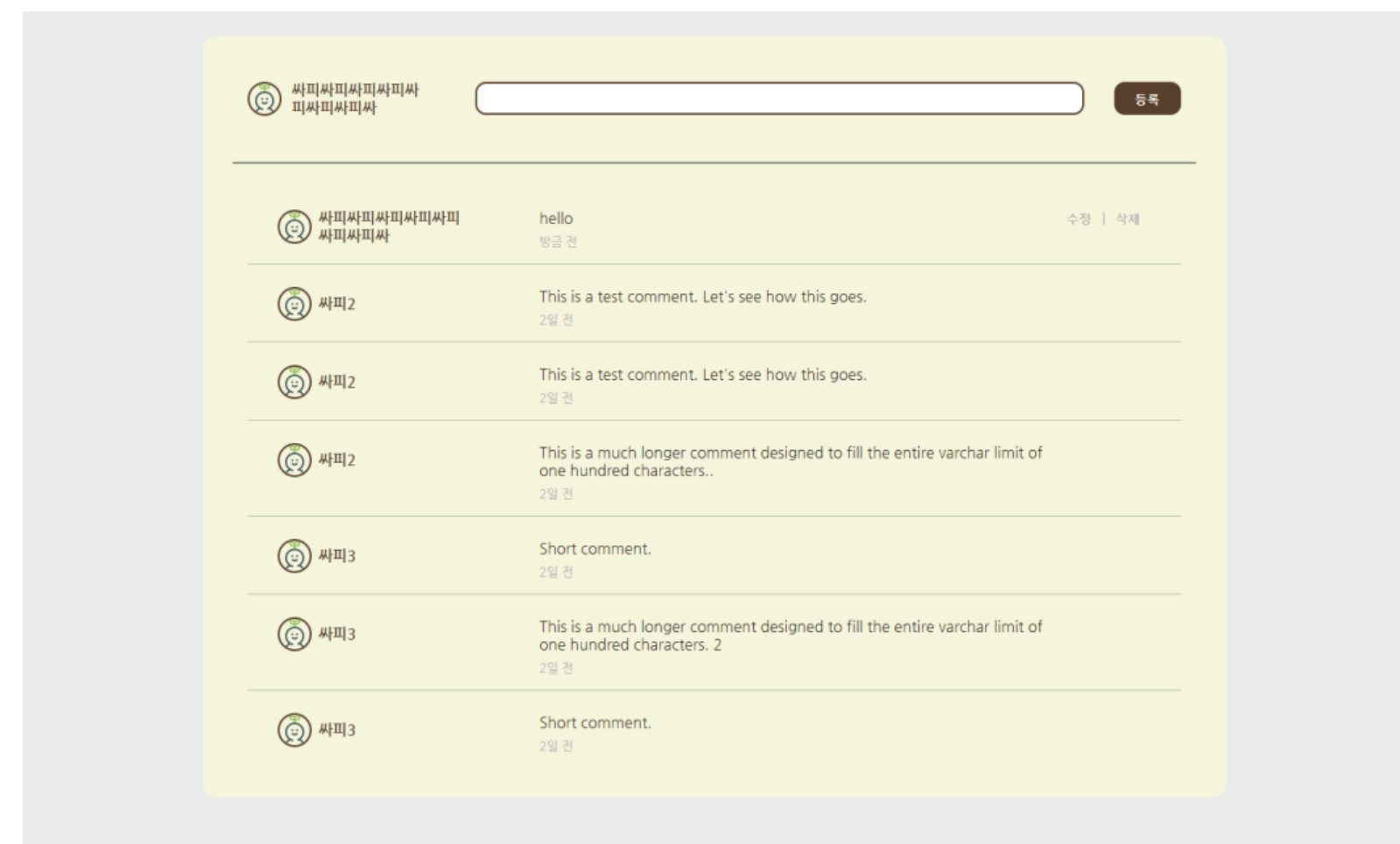
## II. 목표 달성 시 배지 생성



## 05. 개발 결과


핵심 기능

### III. 랭킹 조회 및 글 등록





## 06. 트러블 슈팅

### Trouble 1: Badge 생성을 위한 DB Table 설계

user			
	user_id	varchar(100)	NOT NULL
	password	varchar(100)	NOT NULL
	nickname	varchar(100)	NOT NULL
	user_image_url	varchar(255)	NOT NULL
	user_score	int	NOT NULL

?

badge			
	badge_name	varchar(50)	NOT NULL
	badge_date	int	NOT NULL
	badge_score	int	NOT NULL
	badge_image_url	varchar(255)	NOT NULL

addiction_list			
	item_id	int	NOT NULL
	user_id	varchar(100)	NOT NULL
	title	varchar(100)	NOT NULL
	start_time	timestamp	NOT NULL
	target_time	int	NOT NULL
	item_score	int	NOT NULL
	badge1	boolean	NOT NULL
	badge2	boolean	NOT NULL
	badge3	boolean	NOT NULL
	badge4	boolean	NOT NULL
	badge5	boolean	NOT NULL
	badge6	boolean	NOT NULL
	badge7	boolean	NOT NULL



addiction\_list 아이템 항목 하나하나 당 badge를 boolean 값으로 직접 관리하려 했음



addiction 테이블과 badge 테이블 간  
외래 키 연결이 애매해짐

각 badge 마다  
개별 정보(name, date, score, image) 테이블을  
모두 생성해야 하기 때문

즉, 불필요하고 비효율적인 테이블 생성



badge별 목표 일 수 기준을 충족할 때마다 해당 user의 해당 addiction 아이템에 badge record 추가

## 댓글, 좋아요 방식으로 구현

## 06. 트러블 슈팅

### Trouble 2: 공통 Alert 사용에 따른 전역 상태 공유

```
import { defineStore } from "pinia";
import { ref } from "vue";

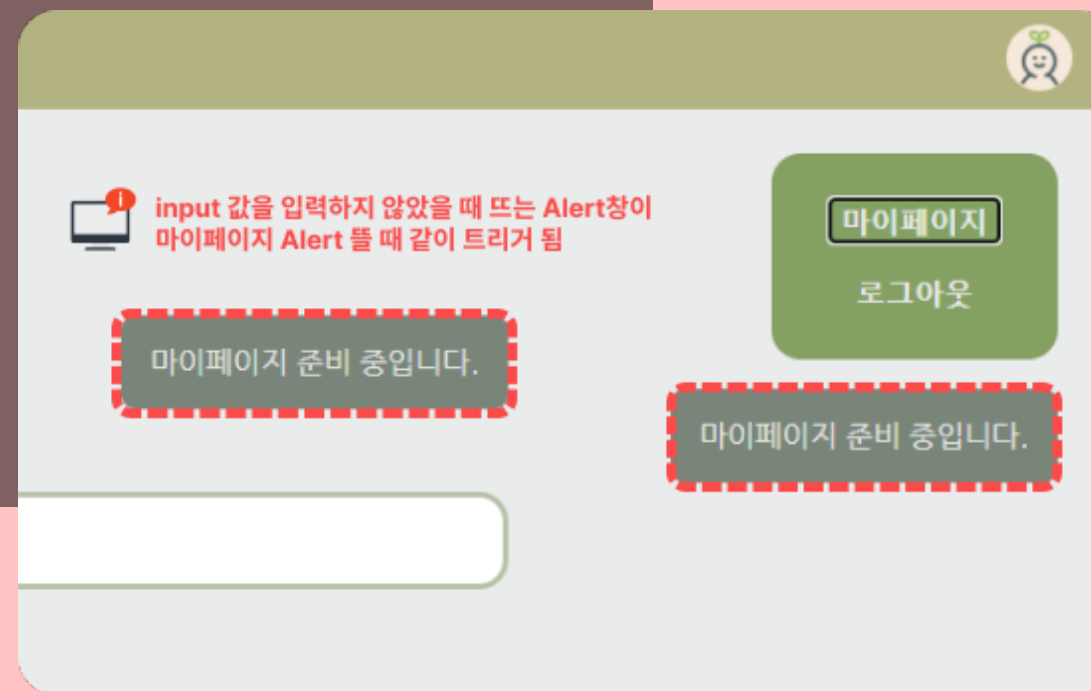
export const useAlertStore = defineStore("alert", () => {
  // state
  const message = ref("");
  const type = ref("");

  // action
  const showAlert = (msg, alertType, duration = 1000) => {
    message.value = msg;
    type.value = alertType;

    setTimeout(() => {
      clearAlert();
    }, duration);
  };

  const clearAlert = () => {
    message.value = "";
    type.value = "";
  };
});
```

단순히 message, type 값을 받아서 활용했을  
어디에서 어떤 Alert를 어떤 순서로 호출했는지  
알 방법이 없음



마이페이지 모달을 클릭하면 '준비 중' alert가 뜨는데,  
등록 및 수정 페이지에서  
input 값을 입력하지 않았을 때 뜨는 alert가  
함께 트리거 되어 두 개의 메세지 창이 뜨게 됨

여러 컴포넌트 간에 같은 Alert 상태를 공유하고 있기  
때문에 발생하는 문제



## 06. 트러블 슈팅

### Solve 2: alertQueue 생성 및 componentId 추가하여 props 전달

```
export const useAlertStore = defineStore("alert", () => {
  // state
  const alertQueue = ref([]); // 알림 큐
  const currentAlert = ref(null); // 현재 알림

  // action
  const setAlert = (msg, alertType, componentId, duration = 1000) => {
    alertQueue.value.push({ msg, alertType, componentId, duration });
    if (!currentAlert.value) {
      showAlert();
    }
  };

  const clearAlert = () => {
    currentAlert.value = null;
    if (alertQueue.value.length > 0) {
      showAlert();
    }
  };

  const showAlert = () => {
    if (alertQueue.value.length > 0) {
      const nextAlert = alertQueue.value.shift();
      currentAlert.value = nextAlert;

      setTimeout(clearAlert, nextAlert.duration);
    }
  };
});
```

💡 alertQueue 및 currentAlert 변수 설정

💡 alert 메시지 setting 시 componentId 설정해두고, 원하는 컴포넌트에서 Alert 넣어줄 때 props로 해당 componentId 전달

💡 alertQueue에서 shift로 먼저 들어온 alert부터 꺼내서 showAlert로 보여줌

✅ 공통 AlertModal 관리하는 store - alertQueue 생성하여 관리

```
<template>
  <div class="modal-overlay" @click.self="closeModal">
    <Alert id="alert" componentId="ProfileModal" />
    <div class="modal-content">
      <button @click="readyAlert">마이페이지</button>
      <button @click="logout">로그아웃</button>
    </div>
  </div>
</template>
```

💡 props로 해당 componentId 내려보냄

```
// 마이페이지 준비중
const readyAlert = () => {
  alertStore.setAlert("마이페이지 준비 중입니다.", "mypage", "ProfileModal");
};
```

💡 alertModal에 해당 componentId 세팅

✅ Alert 띄우고자 하는 컴포넌트 - props로 componentId 전달



## 07. 프로젝트 진행 후기

### 박희연

- 프로젝트에서 가장 많은 공을 들인 게 반응형 디자인이었는데, 레이아웃 잡는 게 정말 어려운 것 같다..
- 공통 컴포넌트에 필요한 정보를 분기하여 구현하는 것이 처음에는 복잡하지만, 재사용 할 때 정말 편하고 좋았다.
- front / back로 나누어 각자 구현하니 코드에 일관성이 있어 좋았으나, 다른 사람의 코드를 보며 고민하고 협업하는 경험도 중요한 것 같다.
- 사용자 측면에서의 편의나, 어떤 오류 사항이 있을 지 찾아내고 수정하는 과정이 나름 보람찼다.
- Auth에 JWT, 보안 관련 설정을 도전해보고 싶고, 다음 프로젝트에서는 새로운 API를 활용해보고 싶다.

### 박건택

- 설계를 탄탄히 하는 것이 중요하다는 것을 깨달았습니다. 주요 기능, 화면 구성, 시간 계획을 기획 단계에서 설정하고, API, 데이터베이스 설계를 탄탄히 해놓고 각자 분배한 역할을 하며 프로젝트를 진행하니 큰 문제 없이 수월하게 진행할 수 있었습니다.
- 프로젝트를 하며 배운 내용을 바탕으로 응용을 하며 개발을 하니 실력이 어느정도 늘은 것 같아 뿌듯하였습니다. 동시에 어려움도 많이 느껴서 부족함을 깨닫고 노력을 해야겠다는 생각도 들었습니다.
- 내부적인 로직이 부족한 점이 무엇인지 알고 싶어졌고, 보안적인 측면을 고려해보고 싶습니다. 또한 처음에 생각한 다양한 API를 사용해보지 못한 것이 아쉽습니다.

