

적금생활을 더욱 다채롭게 • 적금통 키우기

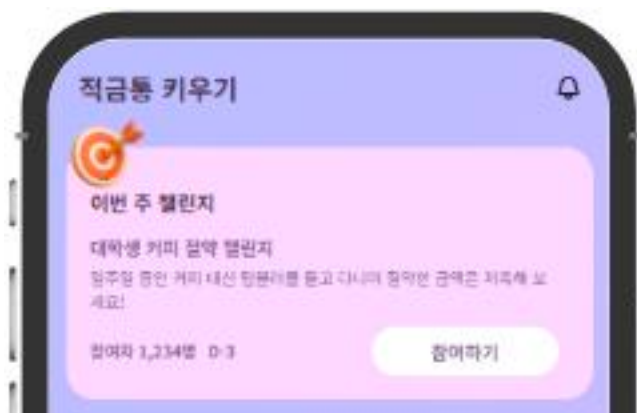
2025년 신한은행 해커톤 with SSAFY 본선 진출 프로젝트

Team 저쪽 신사분께서 보내셨습니다

Role 백엔드 개발자, 팀장

Work 2025. 08 ~ 2025. 09(1개월)





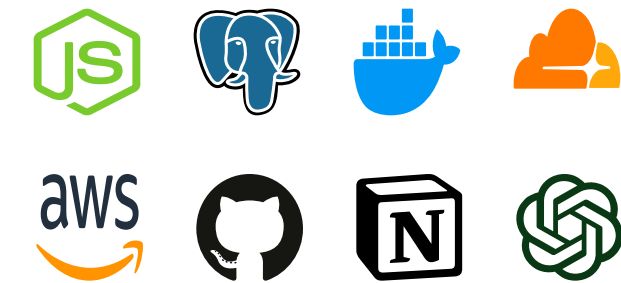
Period

2025.08 ~ 2025.09

Link

https://github.com/Strangekim/saving_box_challenge_backend
<https://github.com/footballSquare/backend>

Skills



Member

김연호, 백엔드 개발자, 팀장

명민주, 백엔드 개발자, UX/UI

송영주, 프론트엔드 개발자, PL

김현우, 프론트엔드 개발자

조예림, 프론트엔드 개발자

개인 기여


- ▶ 해커톤 팀 빌딩
- ▶ API 명세서, 요구사항 명세서 작성
- ▶ DB 스키마 설계
- ▶ 외부 API 호출 추상화
- ▶ 업적, 적금통, 가방, 알림 API 구현
- ▶ docker 컨테이너 셋업
- ▶ Cloud Flare HTTPS 연결
- ▶ 외부 서버 DB 동기화 배치 처리
- ▶ OpenAI GPT-4o-mini로 일일 데이터 분석, 로깅 및 Notion 자동 업로드

Prepare

▶ 해커톤 예선 심사

해커톤에 참여하기 위하여 예선 심사 과정을 거쳐야 했습니다.

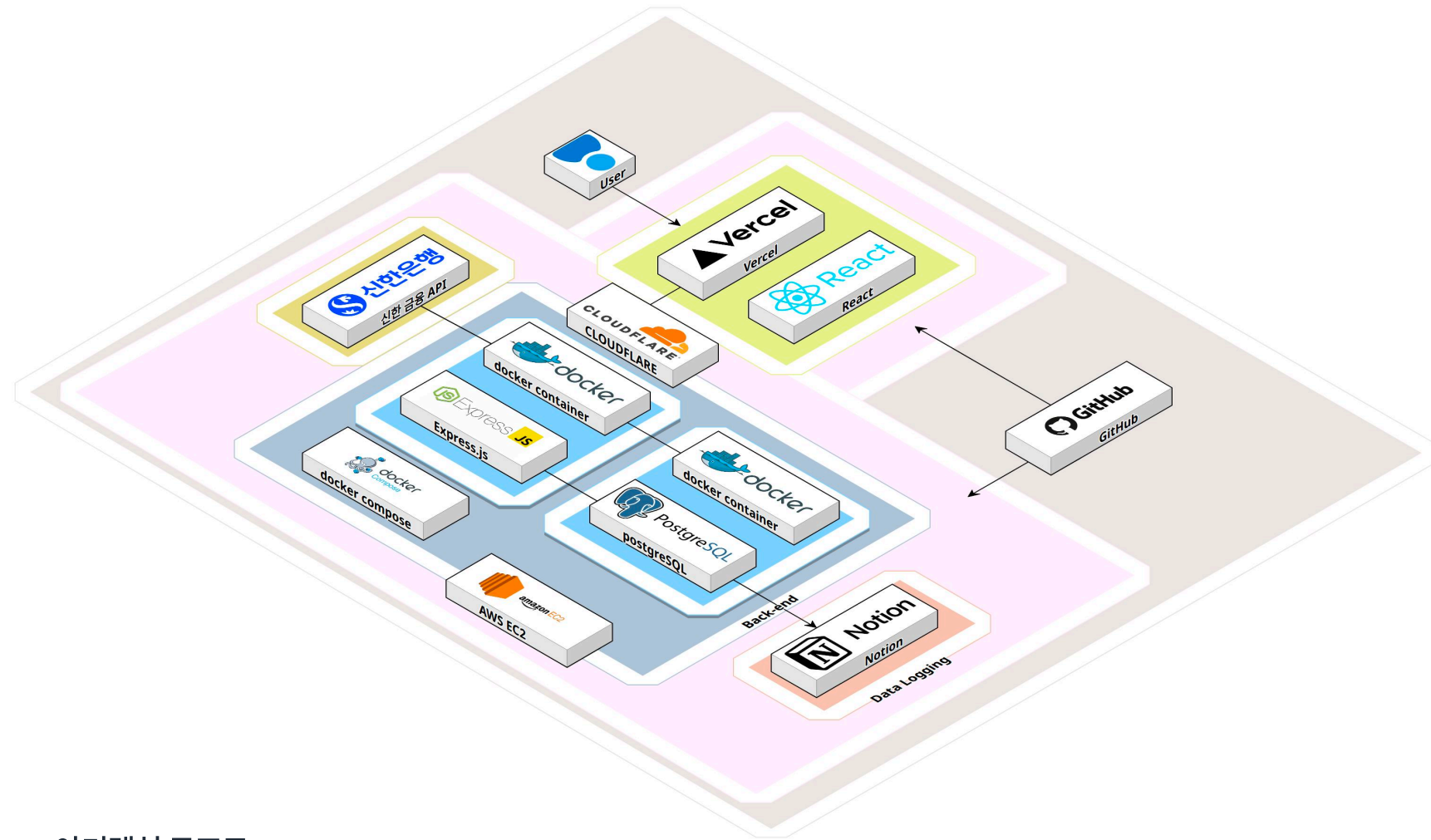
꼼꼼한 기획과 문서 작성을 바탕으로 10:1 이상의 경쟁률을 뚫고 해커톤 본선에 참여할 수 있었습니다.

캠퍼스 생활을 편리하고, 더 재미있게 withSSAFY 2025 Shinhan Hackathon 개발 기획서	
1. 요약서	
팀 명	 저쪽 신사분께서 보내셨습니다.
팀 장	김연호
팀 원	명민주, 송영주, 조예민, 김현우
프로젝트 명	적금통 키우기
프로젝트 개요	자신만의 적금통을 만들어 성장시키고 공유하는, 성장형 소셜 적금 플랫폼
2. 제안 내용	
기획 배경	<p>대학생활 시기는 사회로 나아가기 전 금융 생활의 기초를 다지는 중요한 시기이지만, 많은 학생들이 금융 상품에 대한 관심이나 경험이 부족한 것이 현실입니다.</p> <p>특히 '정기 예,적금'과 같은 기초 금융 습관은 중요함에도 불구하고, 자발적인 참여를 유도할 수 있는 동기나 재미 요소가 부족합니다. 이에 따라 '적금통 키우기'는 단순한 금융 상품 가입을 넘어, 게이미피케이션, 커뮤니티 소셜 요소, 챌린지, 행킹 시스템을 결합하여 재미있게 금융 습관을 형성할 수 있는 새로운 방식의 서비스를 기획하였습니다.</p> <p>사용자는 적금 사연을 공유하고, 친구들과 행킹을 겨루며, 마치 게임처럼 자신만의 적금통이 성장하는 금융 경험을 하게 됩니다. 신한은행의 금융 신뢰도와 대학생의 생활 친화적 접근을 결합하여, 재미 + 습관 + 커뮤니티 + 금융을 융합한 차세대 대학생 맞춤 금융 커뮤니티를 목표로 합니다.</p>
핵심 기능 설명	<ol style="list-style-type: none"> 마이 프로필 및 성장 시스템 <ul style="list-style-type: none"> 사용자의 적금 참여 이력, 포인트, 캐릭터 성장 등을 시각화합니다. 사용자마다 다른 성장시스템으로 나만의 적금통을 가질 수 있습니다. 단순 금액 관리가 아닌 "나만의 금융 캐릭터 성장 경험"으로 사용자 몰입도 향상 적금통 만들기 기능 <ul style="list-style-type: none"> 연결된 적금 상품 리스트 중 선택 -> 목표 금액 및 기간 입력 -> 적금통 이름, 목적, 사연 등 입력 적금통 공유 및 커뮤니티 <ul style="list-style-type: none"> 사용자가 자신의 적금 목적, 사연, 목표 금액 등을 공유하고, 좋아요 / 댓글 기능을 통해 커뮤니티 기반의 상호작용 제공. 금융 활동에 사회적 인정과 공감을 더해 지속적인 참여 유도 적금 챌린지

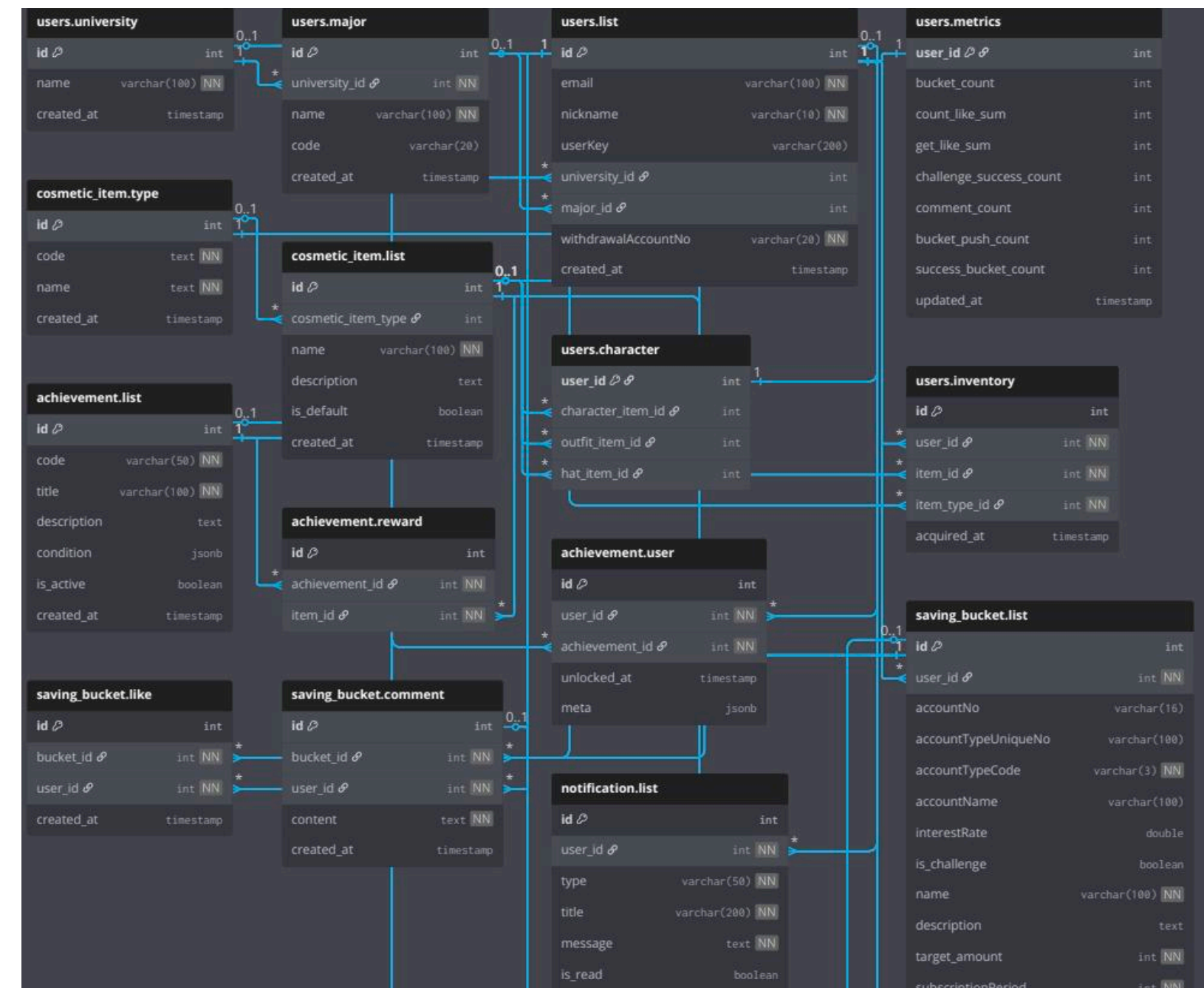
아이디어 차별성	<ul style="list-style-type: none"> 신한은행 또는 캠퍼스 주권의 단기 챌린지형 적금 이벤트 기간, 조건, 보상이 명확한 챌린지를 통해 게임처럼 재미있게 저축 습관 형성 <p>5. 행킹 시스템</p> <ul style="list-style-type: none"> 개인, 친구, 학교 단위로 다양한 행킹을 제공하여 경쟁을 통한 동기 부여 강화 실적 중심의 명확한 피드백 구조로 반복 참여 유도
아이디어 실현 및 구체화 방안	<p>1. 재테크 + 소셜 + 게임의 융합</p> <ul style="list-style-type: none"> 단순한 적금 가입이 아니라, 챌린지 참여 → 행킹 경쟁 → 커뮤니티 공유 → 캐릭터 성장으로 이어지는 루프를 통해 재미있는 금융 경험 제공 <p>2. 대학생 특화 설계</p> <ul style="list-style-type: none"> 대학생의 관심사(인정욕구, 공유문화, 짧은 기간 집중 등)에 맞춰, 목표 기반 적금, 단기 이벤트형 챌린지, SNS 형 게시물 공유 등 실질적 참여 동기를 부여 <p>3. 게이미피케이션 기반의 금융 습관 형성</p> <ul style="list-style-type: none"> 캐릭터 성장, 포인트, 레벨, 배지 등의 요소로 성과에 대한 즉각적인 피드백을 제공, 진입장벽이 높은 금융 활동을 몰입 가능한 경험으로 전환 <p>4. 금융 데이터의 감정화</p> <ul style="list-style-type: none"> 금액 수치만 나열하는게 아닌, "이 적금은 내 여행을 위한 거야" 같은 개인의 서사와 감정을 담아 감성적 연결이 가능한 구조 <p>5. 학교 단위 경쟁</p> <ul style="list-style-type: none"> 개인 뿐만 아니라 학교 간 행킹 시스템을 도입하여, 집단 소속감을 자극하고 자연스러운 확산 효과 기대 <p>1. 핵심 서비스 단계별 MVP 개발</p> <ul style="list-style-type: none"> 핵심 기능 우선 구현 <ul style="list-style-type: none"> 적금통 만들기 및 공유 기능 챌린지 참여 및 기본 행킹 시스템 마이페이지에서 나의 적금 현황, 적금통 리스트 보기 좋아요, 댓글 등 커뮤니티 상호 작용 확장 기능 추가 <ul style="list-style-type: none"> 대학별 행킹 세부 페이지 캐릭터 성장, 포인트 및 레벨 시스템 구체화 적용 SNS 공유 및 사용자 태그 기능 <p>2. 신한은행 예,적금 API 연동</p> <p>3. 소셜 커뮤니티 시스템 경량화</p> <ul style="list-style-type: none"> 적금통 게시물 공유 -> 좋아요 / 댓글 기능 Rest API 기반 서버 사이드 단순 구현 사용자간 관계 (친구, 같은 학교 등) 은 추후 확장을 고려해 기초적인 관계 매핑 구조만 도입 <p>4. 챌린지 / 행킹 시스템 간소화</p> <ul style="list-style-type: none"> 행킹은 "적립률" 또는 "참여수" 기준으로 우선 간단하게 계산 향후 기간별, 챌린지별, 태그별 행킹 등으로 확장 <p>5. 캐릭터 및 성장 시스템 최소 단위 도입</p> <ul style="list-style-type: none"> 포인트 누적에 따라 단계별 일러스트 변화만 우선 적용 이후 경험치 시스템, 커스터마이징 요소 추가 가능하도록 데이터 구조 분리

기대 효과	<ol style="list-style-type: none"> 대학생 사용자 측면 <ul style="list-style-type: none"> 재미있는 금융 첫 경험 제공 : <ul style="list-style-type: none"> 단순한 금융 상품 가입이 아닌, 캐릭터 성장, 스토리 공유, 행킹 경쟁 등 게임처럼 즐기는 재테크 경험 자발적인 금융 습관 형성 유도 <ul style="list-style-type: none"> 챌린지 및 행킹 시스템으로 통해 반복적이고 지속적인 적금 참여 동기 부여 금융 커뮤니티 경험 확대 <ul style="list-style-type: none"> 친구나 같은 대학 친구들의 적금 사연을 보며 공감과 자극을 동시에 경험하고, 함께 저축하는 문화 조성 신한은행 측면 <ul style="list-style-type: none"> MZ 세대 대상 금융 유입 창구 확보 <ul style="list-style-type: none"> 기존 금융 앱이 도달하기 어려운 대학생들에게 놀이 기반 유입 구조 제공 신규 적금 상품 가입 증가 기대 <ul style="list-style-type: none"> 챌린지를 통한 특정 상품 참여 유도 -> 목표 기반 단기 적금 판매 확대 장기 고객으로의 전환 기반 마련 <ul style="list-style-type: none"> 대학생 시절부터 신한 브랜드에 익숙해지며 자연스러운 고객 충성도 형성
활용할 금융 API (다중 선택 가능) * 금융 서비스 개발을 선택한 경우 작성	<ul style="list-style-type: none"> 모든 예금 관련 API 모든 적금 관련 API 계좌 입출금 API
3. 개발 내용	
사용 언어 및 프레임워크	<p>프론트엔드 : React, Vercel</p> <p>백엔드 : Node.js, Express, PostgreSQL</p> <p>기타 : AWS EC2, AWS S3, Git Hub</p>
개발 세부내용 (플로우차트, ERD, 간트 차트 등 이미지 첨부 가능)	<ul style="list-style-type: none"> 화면 설계서

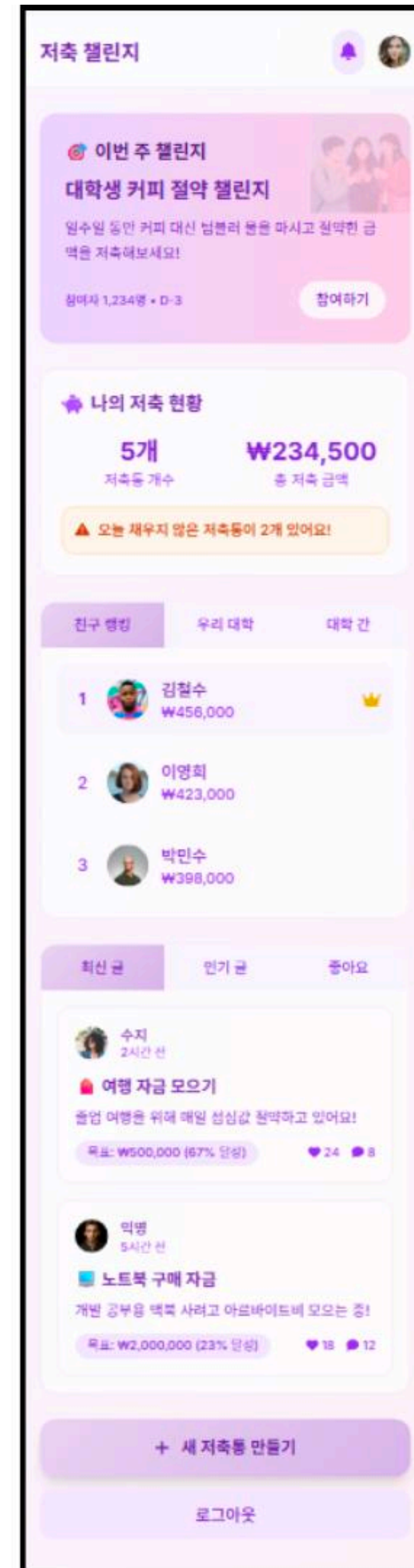
▶ 예선 개발 기획서



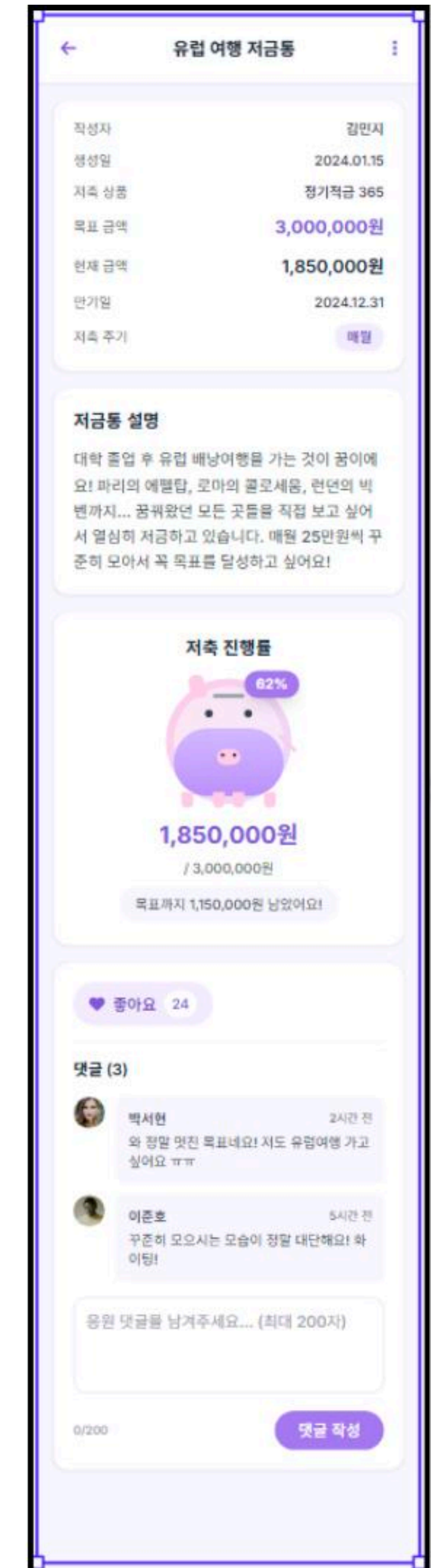
아키텍처 구조도



DB ERD



화면 설계서



Problem

▶ 대학생들의 낮은 저축률과 금융 무관심

20대 초반 대학생들은 수입이 적고 소비 욕구가 높아 **저축 습관 형성이 어렵다**는 문제를 사전조사에서 파악하였습니다. 또한, 기존의 금융 서비스는 딱딱하고 UX가 불편한 부분이 많아 MZ세대의 관심을 끌기 어려웠습니다. 특히 적금 상품들은, 단순히 돈을 넣고 이자를 받는 일방향적 서비스로 최근 젊은 세대의 트렌드인 **즉각적인 보상과 피드백**과 같은 니즈를 반영하기 어렵다는 한계가 있습니다.

Solution

1 나만의 성장하는 적금통 기능

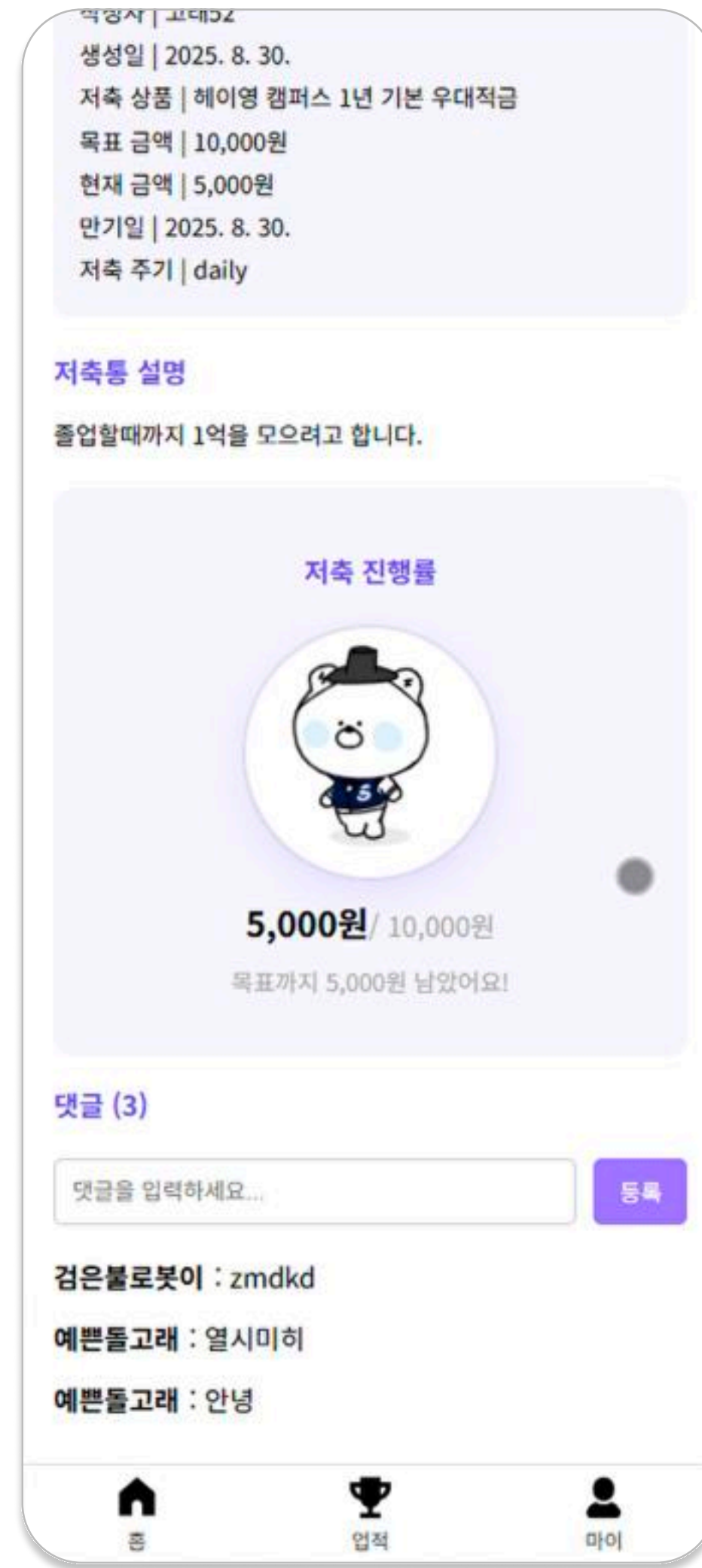
- 사용자의 적금 참여 이력, 포인트, 캐릭터 성장 등을 **시각화** 합니다.
- **사용자마다 다른 성장 시스템**으로, 나만의 적금통을 가질 수 있습니다.
- “나만의 캐릭터 성장 경험”으로 **몰입도를 향상** 시킵니다.

2 SNS 형 적금통 공유 및 커뮤니티

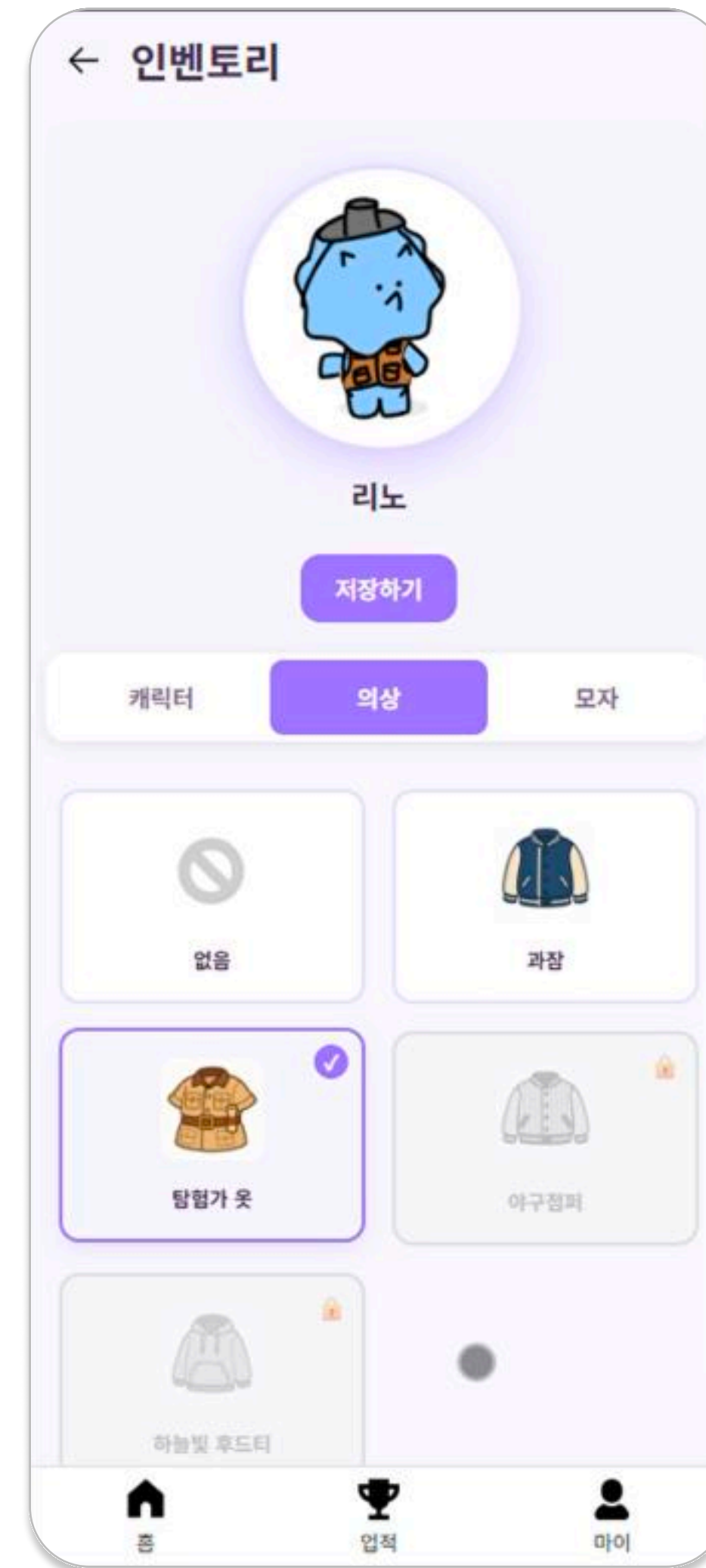
- 자신의 적금 목적, 사연 등을 대학교 친구들과 공유하며, **커뮤니티 기반의 상호작용**을 제공 합니다.
- 금융 활동에 사회적 인정과 공감을 더해 **지속적인 참여**를 유도합니다.

3 게이미피케이션 기반의 금융 습관 형성

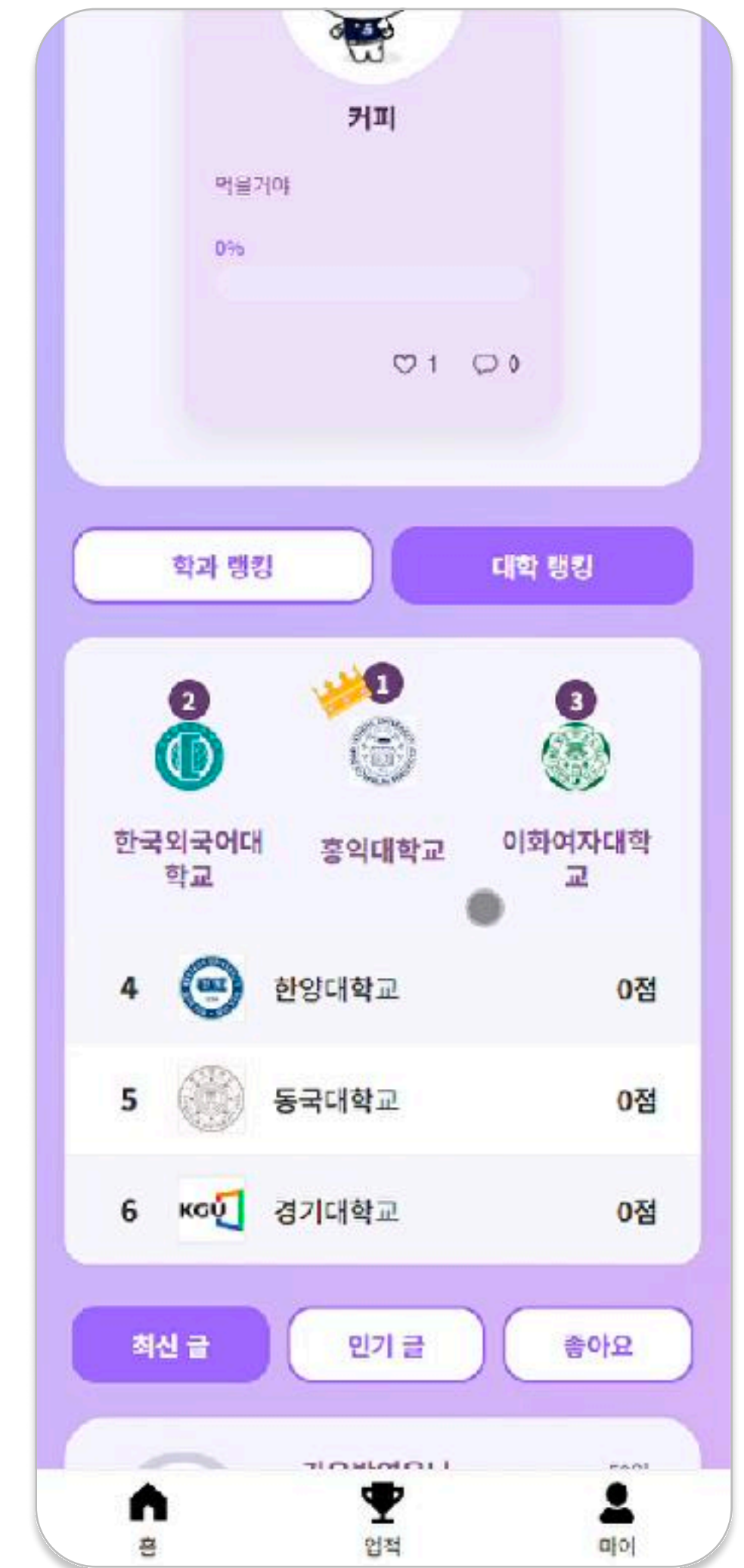
- 캐릭터 성장, 포인트, 레벨, 배지 등의 요소를 도입하여 성과에 **즉각적인 피드백**을 제공, 몰입 가능한 경험을 제공합니다.
- 개인 뿐 아니라 학교 간 랭킹, 챌린지 시스템 도입으로 **집단 소속감과 자연스러운 확산 효과**를 기대합니다.



▶적금통 상세보기 페이지



▶캐릭터 및 인벤토리 페이지



▶랭킹 페이지

Solution

▶ 미들웨어 기반 업적 시스템 추상화

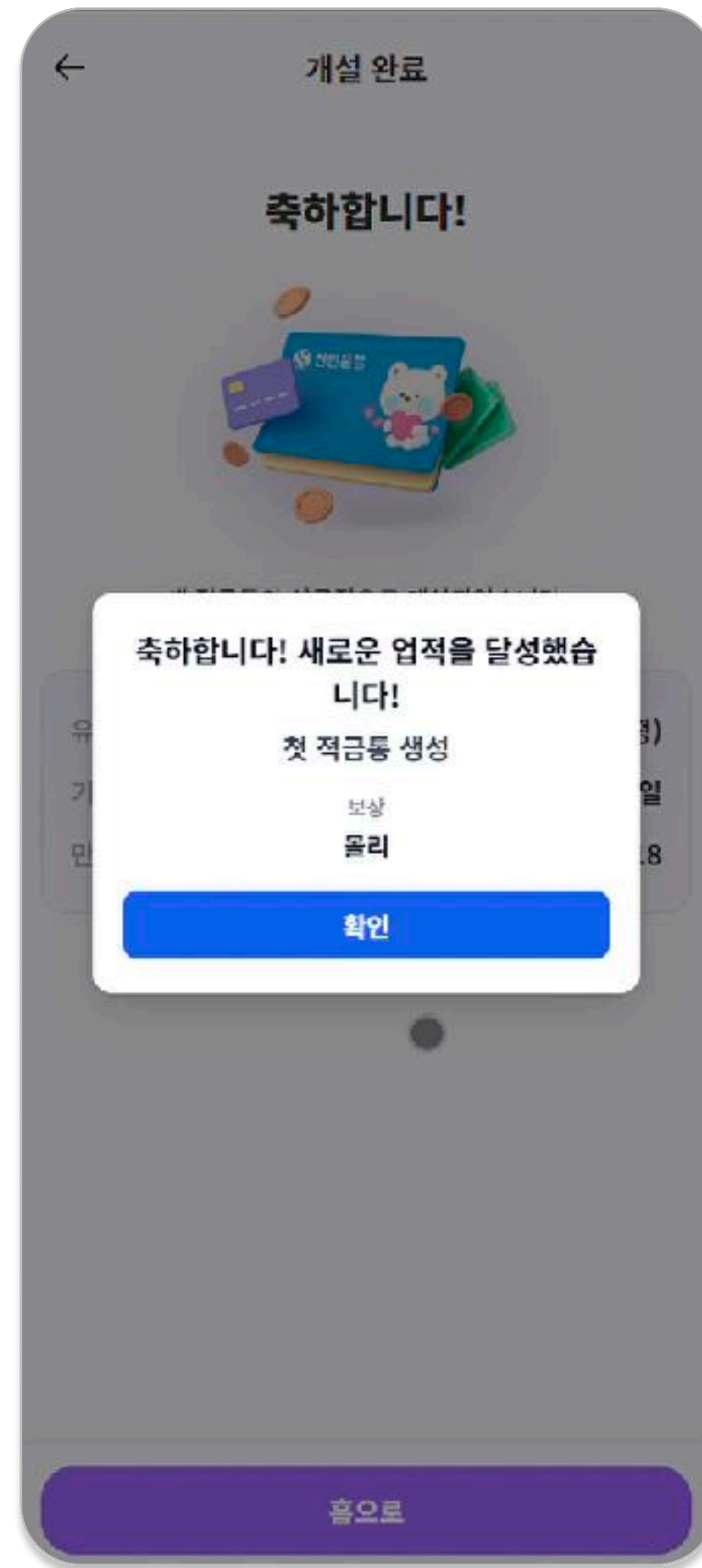
서비스의 모든 사용자 액션(적금통 생성, 좋아요, 댓글 등)에서 업적 처리 로직이 반복되는 문제를 해결하기 위해 공통 미들웨어 패턴을 설계하였습니다.

핵심 구조 :

- achievementService.js : 업적 조건 확인 및 보상 지급 로직
- achievementMiddleware.js : 각 액션별 헬퍼 함수로 컨트롤러 계층 추상화
- 각 컨트롤러에서 handelXXXAchievement() 함수 호출만으로 업적 처리 완료

효과 :

- 8개의 액션 타입에 대한 업적 처리를 단일 함수로 통합
- DB 업데이트 → 조건 확인 → 보상 지급 → 알림 전송 파이프라인 자동화
- 컨트롤러 코드량 60% 이상 감소, 업적 추가 시 서비스 계층만 수정



▶ 업적 달성 예시 화면

```
// ===== 업적 처리 후 응답 가로채기 함수 =====
export const handleAchievementResponse = async (req, res, actionType, actionData = {}, originalResponseData = null) => {
  // 이미 응답이 전송되었는지 확인
  if (res.headersSent) {
    return false;
  }

  // 로그인된 사용자만 업적 처리
  if (!req.session?.userId) {
    return false;
  }

  try {
    const userId = req.session.userId;

    // 업적 처리
    const achievementResult = await processUserAction(userId, actionType, actionData);

    if (achievementResult.newAchievements.length > 0) {
      // 각 업적마다 알림 생성 (읽음 처리된 상태로)
      for (const unlock of achievementResult.newAchievements) {
        await notifyAchievement(userId, {
          achievementId: unlock.achievement.id,
          achievementTitle: unlock.achievement.title
        });
      }
      // 다시 한 번 응답 상태 확인
      if (res.headersSent) {
        return false;
      }
    }
    console.log("actionData", actionData)
    // 기본 응답 데이터와 업적 정보를 합친 응답
    const achievementResponse = {
      // 원래 응답 데이터가 있으면 포함
      ...(originalResponseData || { success: true }),
      type: 'achievement_unlocked',
      message: '축하합니다! 새로운 업적을 달성했습니다!',
      achievements: {
        count: achievementResult.newAchievements.length,
        totalRewards: achievementResult.newAchievements.reduce(
          (total, ach) => total + ach.rewards.length, 0
        )
      }
    };
  }
}
```

▶ achievementService.js

```
// 8. 업적 처리 및 응답 가로채기 시도 (기존 데이터 포함)
const achievementHandled = await handleBucketCreationAchievement(req, res, savedBucket, responseData);

if (!achievementHandled) {
  console.log('일반 응답 전송 시도 (201)');
```

▶ 각 컨트롤러 업적 처리 함수 호출 예시

Solution

▶ 크론 기반 적금통 동기화

자체 서버 DB 를 신한은행 DB와의 적금 납입 상태 동기화를 위하여 수백 개의 적금 납입 내역을 **매일 자동으로 동기화**하고, 만료 / 실패 처리 및 알림을 전송하는 배치 시스템을 구축 하였습니다.

핵심 구조 :

- 매일 08:00 KST 스케줄링
- 배치 단위 병렬 처리 + 딜레이 제어
- 신한 API 호출 → 데이터 파싱 → DB 업데이트 → 알림 / 업적 처리 파이프라인

효과 :

- 성능 최적화 : 배치 처리로 N 개 적금통을 3개씩 병렬 실행 (전체 시간 1/3 단축)
- 운영 자동화 : 만료 / 실패 처리, 알림 생성, 업적 달성 판정 모두 자동
- 모니터링 : 상세 로그 + 실행 요약

▶ 동기화 → AI 분석 → Notion 리포트 파이프라인 구축

매일 적금통 동기화 작업 후, 자동으로 서비스 운영 데이터를 분석하고 AI 리포트를 생성하여 Notion에 업로드하는 자동화 데이터 로깅 시스템을 구축하였습니다.

파이프라인 :

08:00 크론 작동 → 적금통 동기화 → 8개의 DB 쿼리로 운영 데이터 수집 → Open AI 로 분석 리포트 생성 → Notion 블록 생성 후 업로드

```
try {
  // 1. 활성 적금통 조회
  const activeBuckets = await getActiveBuckets();
  totalBuckets = activeBuckets.length;

  console.log(`📁 Found ${totalBuckets} active buckets to sync`);

  if (totalBuckets === 0) {
    console.log('✅ No active buckets to sync');
    return;
  }

  // 2. 배치 처리 설정
  const BATCH_SIZE = 3;
  const BATCH_DELAY = 500; // 500ms

  // 3. 배치별 처리
  for (let i = 0; i < activeBuckets.length; i += BATCH_SIZE) {
    const batch = activeBuckets.slice(i, i + BATCH_SIZE);
    const batchNumber = Math.floor(i / BATCH_SIZE) + 1;
    const totalBatches = Math.ceil(activeBuckets.length / BATCH_SIZE);

    console.log(`📁 Processing batch ${batchNumber}/${totalBatches} (${batch.length} buckets)`);

    // 배치 내 병렬 처리
    const batchPromises = batch.map(bucket => syncSingleBucket(bucket));
    const batchResults = await Promise.allSettled(batchPromises);
  }
}
```

▶ 크론 기반 적금통 동기화

```
// ===== 크론 스케줄러 설정 =====
export const setupCronJobs = () => {

  // 매일 오전 8시에 실행 (한국시간)
  cron.schedule('0 8 * * *', async () => {
    const timestamp = new Date().toLocaleString('ko-KR', { timeZone: 'Asia/Seoul' });
    console.log(`[${timestamp}] Daily bucket sync started`);

    try {
      await syncAllBuckets();

      // 동기화 완료 후 AI 리포트 생성
      console.log('AI 리포트 생성 시작...');
      const response = await fetch('http://localhost:3000/report/generate-ai-report', {
        method: 'POST'
      });

      if (response.ok) {
        console.log('✅ AI 리포트 생성 완료');
      } else {
        console.error('AI 리포트 생성 실패:', response.status);
      }
    } catch (error) {
      console.error('Daily sync cron failed:', error);
    }
  });
}
```

▶ AI 리포트 생성 스케줄러

일일 리포트 - 2025. 8. 30.

🤖 AI 생성 리포트 | 생성 시간: 2025. 8. 30. 오전 4:02:17

🐷 적금통 키우기 일일 리포트 - 2025. 8. 30.

핵심 지표

- ****총 사용자 수:**** 10명
- ****진행 중인 적금통:**** 11개
- ****성공한 적금통:**** 0개
- ****실패한 적금통:**** 2개
- ****신규 가입자:**** 0명
- ****신규 적금통:**** 0개 (일반 0개, 챌린지 0개)
- ****새로운 좋아요:**** 0개
- ****새로운 댓글:**** 0개

전일 대비 변화

어제(2025-08-28)와 비교했을 때, 신규 활동이 전혀 없었습니다. 신규 가입자, 적금통 생성, 좋아요 및 댓글이 모두 0으로 기록되었습니다. 전체 사용자 수와 진행 중인 적금통 수치 역시 변화가 없습니다.

특이사항 및 주목할 점

현재 진행 중인 적금통 11개 중 성공한 적금통이 0개라는 점이 우려스럽습니다. 실패한 적금통이 2개로, 이는 전체 진행 중인 적금통의 약 18%에 해당합니다. 이러한 상황은 사용자들의 참여와 지속적인 활동이 저조하다는 신호일 수 있습니다.

개선 필요 사항 및 제안

1. ****사용자 참여 유도:**** 신규 가입자와 활동을 증가시키기 위해 프로모션이나 이벤트를 고려해 볼 필요가 있습니다. 예를 들어, 신규 가입자에게 첫 적금통 생성 시 인센티브를 제공하거나, 기존 사용자에게 친구 추천 시 보상을 주는 방안을 제안합니다.

▶ 생성된 AI 리포트

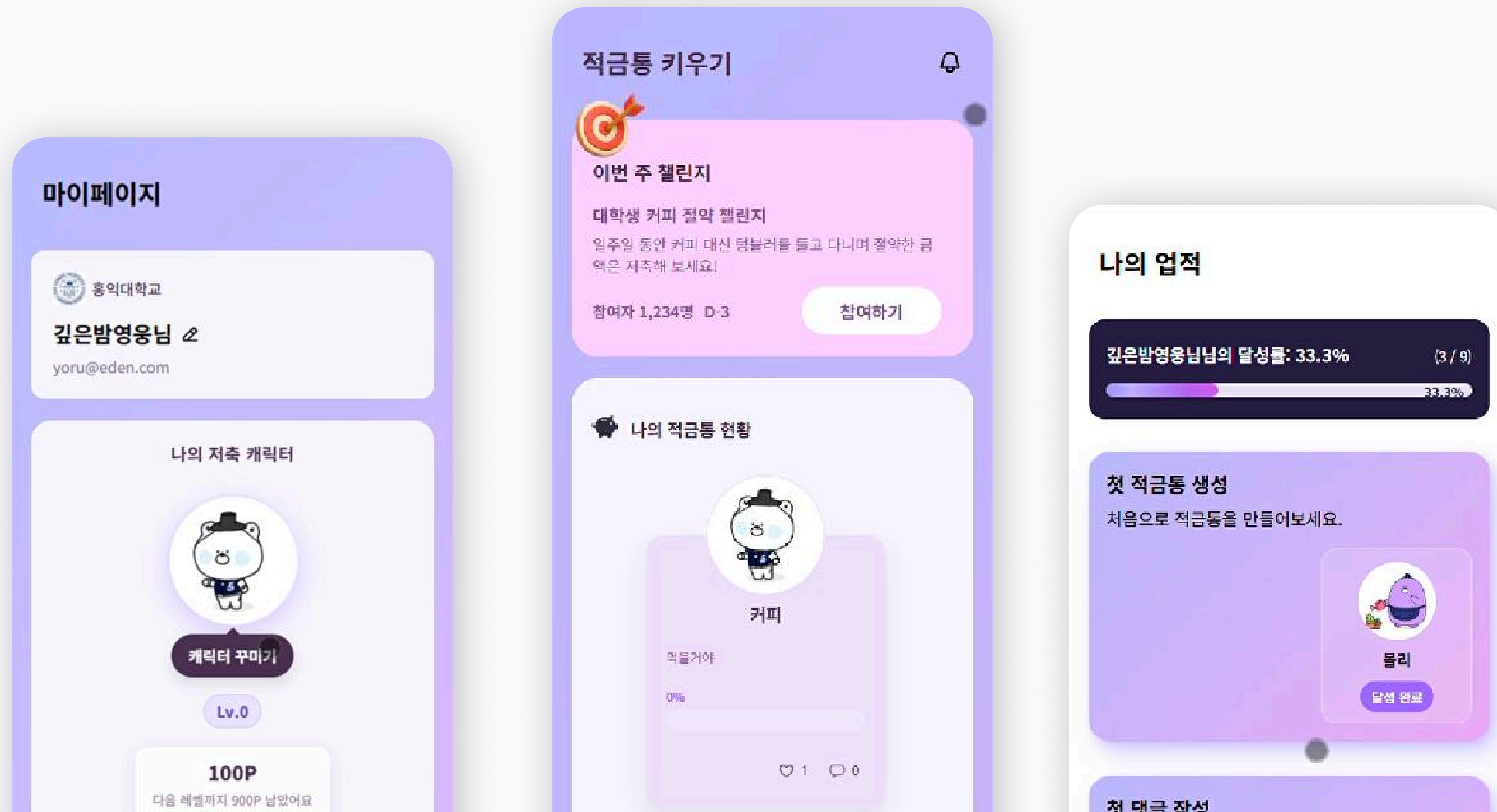
Lesson Learn

비록 수상에는 실패하였지만, 첫 해커톤에서 성공적으로 프로젝트를 빌드업하고 완성해낸 것에 의의를 두고 있습니다.

팀원들과 짧은 기간안에 의견을 취합하고, 합리적인 의사결정을 내리는 **과정 자체가 값진 경험**이었습니다.

수상팀들의 작업물을 살펴보며 프로페셔널한 개발이 무엇인지 깨달았습니다.

결과보다 과정에서 더 많은 것을 배운 경험이었던 만큼 다음 도전에서는 이번 교훈을 바탕으로 더 나은 결과를 만들어낸 자신이 생겼던 시간이었습니다



Thank you

새로운 것을 개발하고,
코드 작성하는걸 사랑하는 개발자, 김연호

roger345@naver.com

010 • 3324 • 8711