

급PAY??

급하게 아르바이트생을 구할 수 있는 플랫폼

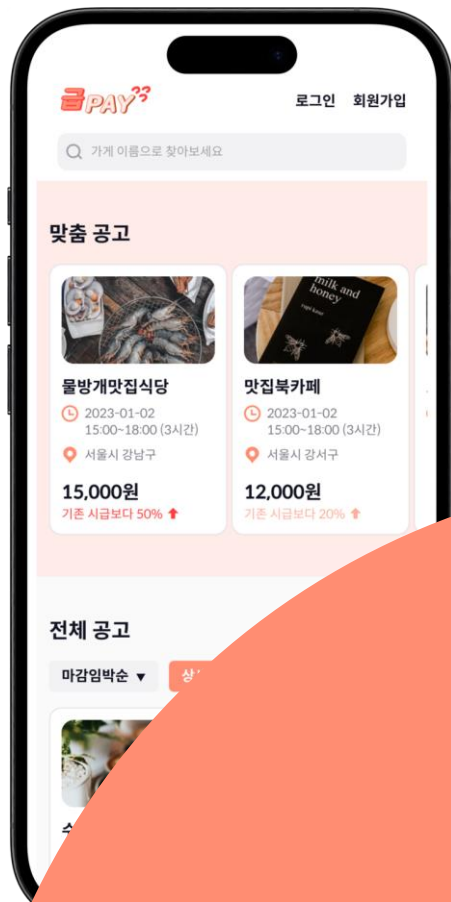


Part3. 9팀)

김보미 : 이승현 : 임진조 : 채종민 : 홍서하

8 목차

1. 프로젝트 목표
2. 팀원소개
3. 프로젝트 상세
 - 프로젝트 관리
 - 기술 스택
 - 프로젝트 구성도
 - 기능 소개
4. 프로젝트 시연
5. Q&A



프로젝트 목표

📌 프로젝트 선정 이유

✏️ 사장과 아르바이트생 두 시점에서의 필요성

- 사장 시점

급하게 일손이 필요한 상황에서 신속하게 높은 시급을 제시함으로써 필요한 순간에 빠르게 일손을 구할 수 있다.

- 아르바이트생 시점

아르바이트생들은 유연한 근무 조건과 더 높은 시급을 제공하는 단기 일자리를 선호하는 경향이 있다. 이 서비스는 아르바이트생들이 자신의 일정과 선호에 맞는 일자리를 쉽게 찾을 수 있도록 한다.

✏️ 실제로 외주가 들어올 법한 주제

실제로 많은 소상공인과 아르바이트생이 직면하는 문제를 해결하는 실용적인 솔루션을 가지고 있다. 서비스의 상업적 가치가 높아 다양한 비즈니스 모델로 확장할 수 있는 잠재력이 있다.

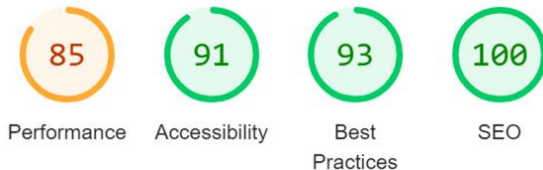
✏️ 다양한 커스터마이징 가능성

이 서비스는 고용주와 아르바이트생 모두의 필요와 선호에 따라 다양한 기능을 커스터마이징할 수 있다.

예를 들어, 고용주는 특정 시간대나 업무에 맞는 인력을 찾을 수 있는 필터링 기능을 사용할 수 있으며, 아르바이트생은 자신의 선호도에 맞는 일자리를 쉽게 찾을 수 있는 맞춤형 검색 기능을 사용할 수 있다.

📌 프로젝트 목표

✏️ SEO를 지키는 개발



✏️ DX를 신경 쓰는 개발 (공통 컴포넌트 jsDocs 적극 활용)

```
const han
  setup
    @example
    <Form className="w-[500px]"> // 넓이는 부모 요소 100%
    <CustomTextInput(
      label="비밀번호"
      placeholder="비밀번호를 입력하세요"
      register={register("password")}
      displayRequiredMarker={true} // 생략하면 * 없음
      type="password" // 생략하면 text
      errorMessage={errors.password?.message}
    ) />
    <form>
    <div>
    <CustomTextInput
      label="이름"
      placeholder="가게 이름을 입력해 주세요"
      register={register("name")}
      displayRequiredMarker={true}
      errorMessage={errors.name?.message}
    />
  />
```

✏️ UX를 신경 쓰는 개발 (서버 컴포넌트 적극 활용)

✏️ 공식 문서를 준수하며 앱 라우터 활용

팀원 소개



김보미



채종민



이승현



임진조



홍서하

프로젝트 상세

1) 프로젝트 관리

깃허브 주소

GitHub - 902z/gheupPAY: 급하게 일손이 필요한 자리에 더 많은 시금...

902z/gheupPAY

급하게 일손이 필요한 자리에 더 많은 시금을 제공해서 아르바이트생을 구할 수 있는 서비스. Contribute to 902z/gheupPAY development by creating an account on <https://github.com/902z/gheupPAY>

배포 링크

배포 링크

프로젝트

프로젝트

작업

기술 스택

요구사항 리스트

피그마 시간

우리 팀 피그마 시간

useflow

api 명세

API 상세

트라블 이슈

규칙

폴더 구조

Git 브랜치

코드 컨벤션

커밋 컨벤션

프로젝트 순서

일정

팀미팅 14시

멘토링 화, 금 15시

코어타임 14시~17시

데일리 스크럼

멘토링

기타

API

The-julge

여기러 오세요

회고

팀 회고

활동

회의록

데일리 문제

프로젝트 발표

07/01

프로젝트 관리

한정된 권한을 가진 사용자에 대한 접근 제한

프로젝트 관리

한정된 권한을 가진 사용자에 대한 접근 제한

프로젝트 관리

작업

작업

작업

작업

노션

노션

1) 데일리 스크럼

- 최대 15분으로 빠르게 끝내기
- Done, Todo, 어려웠던 점을 매일 정리해서 공유하기

2) 작업장

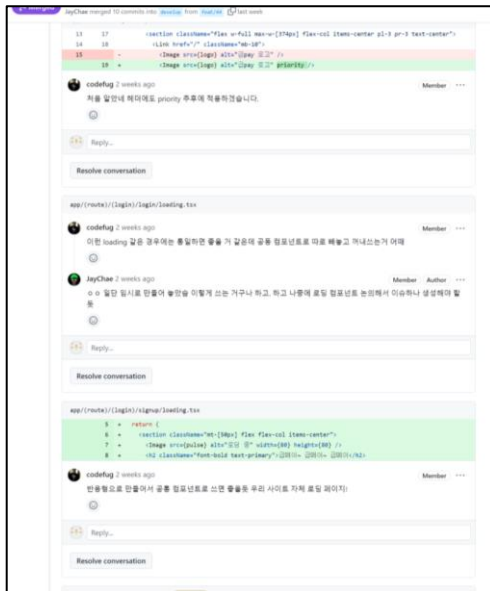
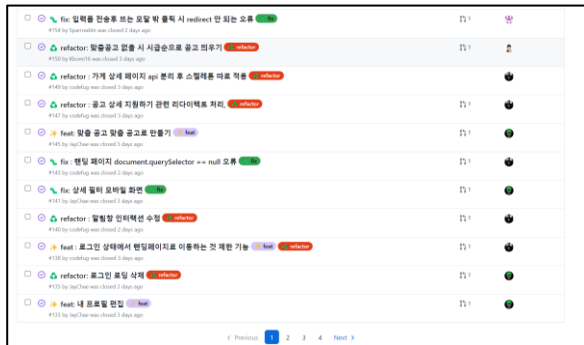
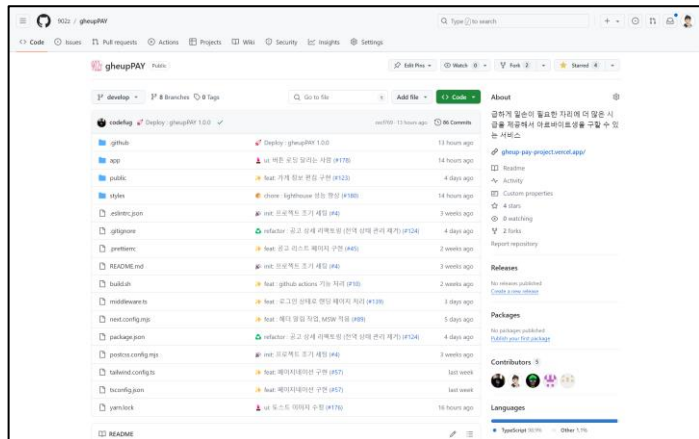
- 작업 만들어서 개발 정리하기

3) 멘토링

- 멘토링 하면서 프로젝트에 대한 수정사항 정리하기

프로젝트 상세

1) 프로젝트 관리



1) 브랜치 전략

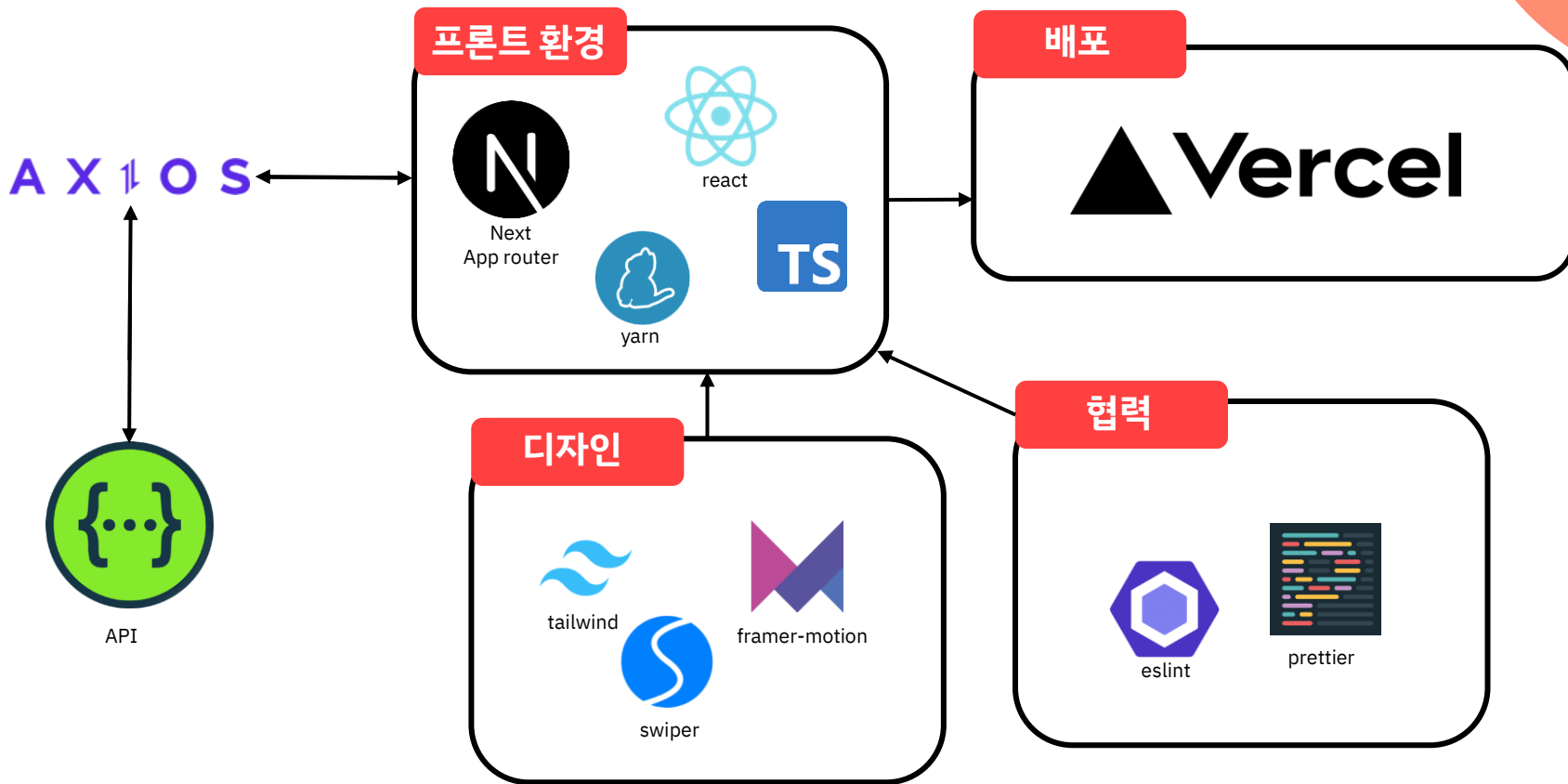
- 이슈만들고, 이슈 토대로 브랜치 파기
 - 기능에 따른 브랜치 타입명 붙이기
- 예) feat/12, refactor/158

2) 코드리뷰

- 팀원들이 올린 코드는 꼭! 보고, 리뷰하기
- 2명 승인 후, squash and merge하기

프로젝트 상세

2) 기술 스택



프로젝트 상세

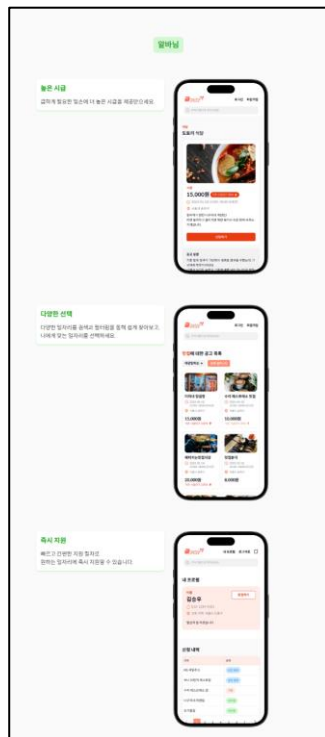
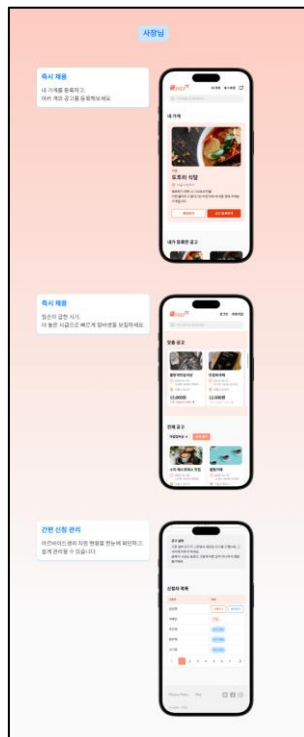
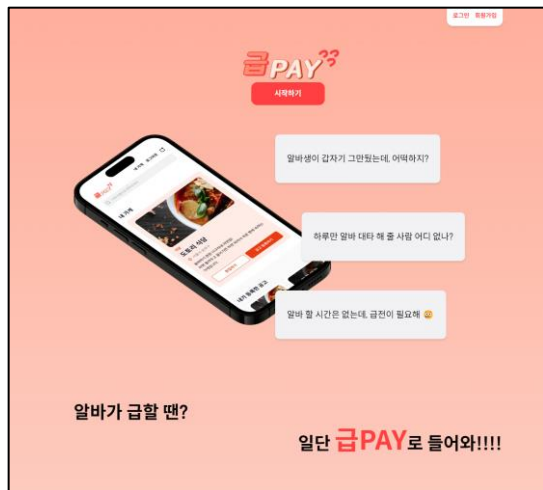
3) 프로젝트 구성도



프로젝트 상세

4) 기능소개

✨ 랜딩 페이지



framer-motion, aos
라이브러리를 사용해 애니메이션 구현

프로젝트 상세

4) 기능소개

✨ 로그인/회원가입 페이지

- 로컬 vs 주스탄스 vs 쿠키
 - 쿠키 사용
 - 서버 컴포넌트는 로컬 접근 불가

- 헤더를 서버컴포넌트로 제작 가능
 - 렌더링 시점에 로그인 상태 바로 반영
 - “로그인”에서 “내 가게”로 깜빡하면서 바뀌지 않음

```
"use server";

import { cookies } from "next/headers";

export async function setCookie(name: string, value: string) {
  const oneDay = 24 * 60 * 60 * 1000; //일단 만료 기한은 하루로 할게요.
  cookies().set(name, value, { expires: Date.now() + oneDay });
}

export async function deleteCookie(name: string) {
  cookies().delete(name);
}

export async function getCookie(name: string) {
  return cookies().get(name)?.value;
}
```

프로젝트 상세

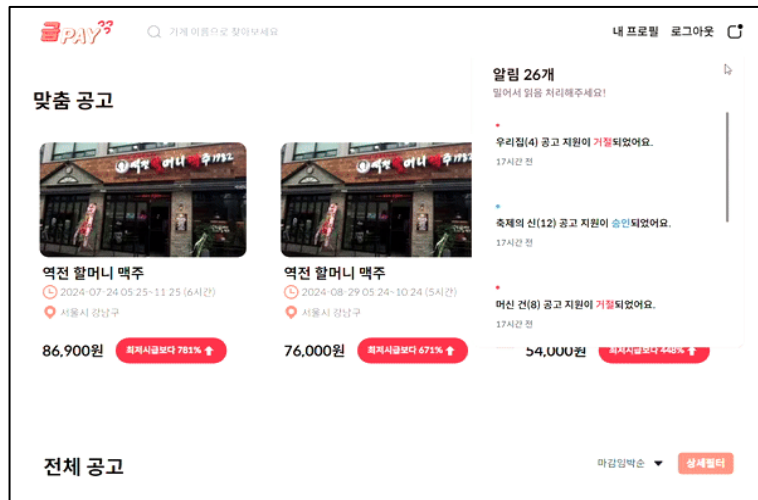
4) 기능소개

✨헤더 - 알림창



✎ 쿠키를 활용한 조건부 렌더링

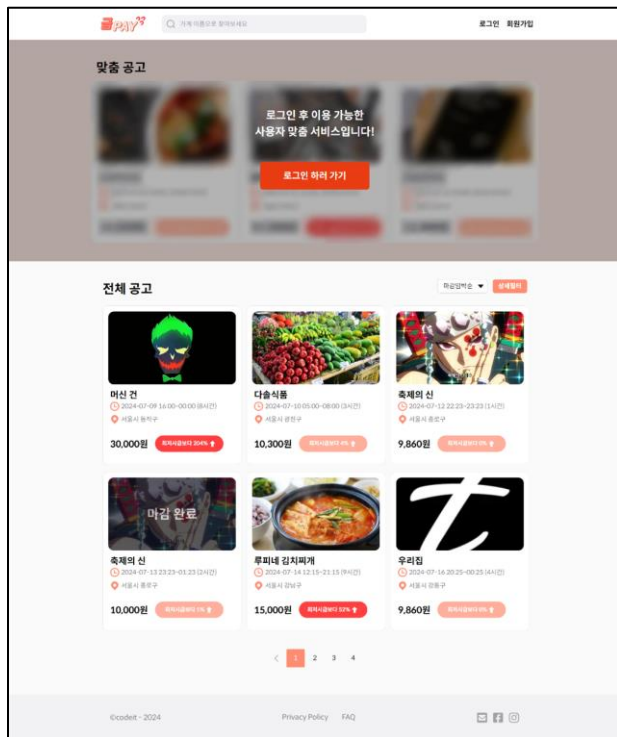
✎ SSR로 초기값 렌더링 후, CSR (무한 스크롤 구현)



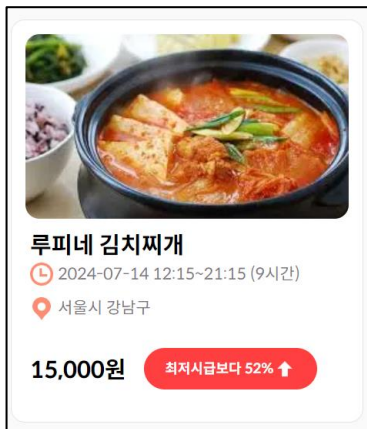
프로젝트 상세

4) 기능소개

🌟 공고 리스트 페이지(비로그인)



- 로그인 후 이용 가능한 맞춤 공고 서비스
맞춤 공고는 사용자의 선호 주소 기반이기 때문에 로그인 후 이용 가능함.



```
import { MINIMUM_HOURLY_WAGE } from "../constants/hourly-wage";  
  
export const calculateWagePercentage = (hourlyWage: number) => {  
  if (hourlyWage < MINIMUM_HOURLY_WAGE) {  
    return 0;  
  }  
  return Math.round(  
    ((hourlyWage - MINIMUM_HOURLY_WAGE) / MINIMUM_HOURLY_WAGE) * 100,  
  );  
};
```

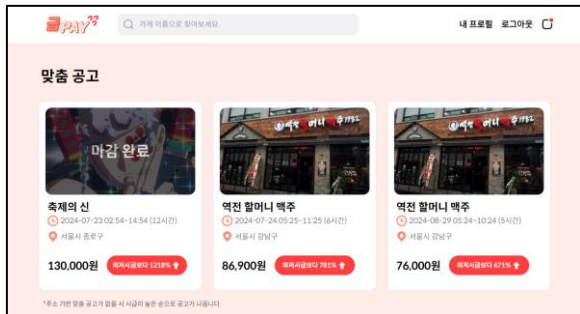
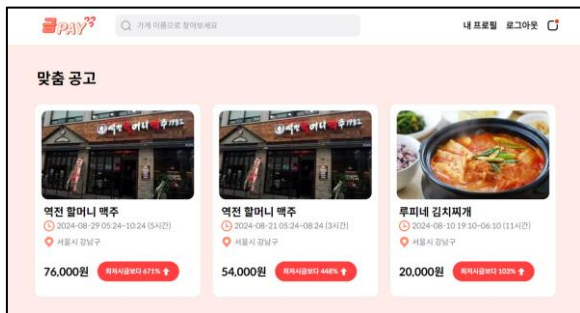
```
const getColorByPercentage = (percent: number) => {  
  if (percent >= 50) return "md:bg-red-40";  
  if (percent >= 30) return "md:bg-red-30";  
  if (percent >= 0) return "md:bg-red-20";  
};
```

- 주어진 시급이 최저시급 대비 몇 퍼센트인지 계산 후,
퍼센트에 따른 라벨 배경색 지정

프로젝트 상세

4) 기능소개

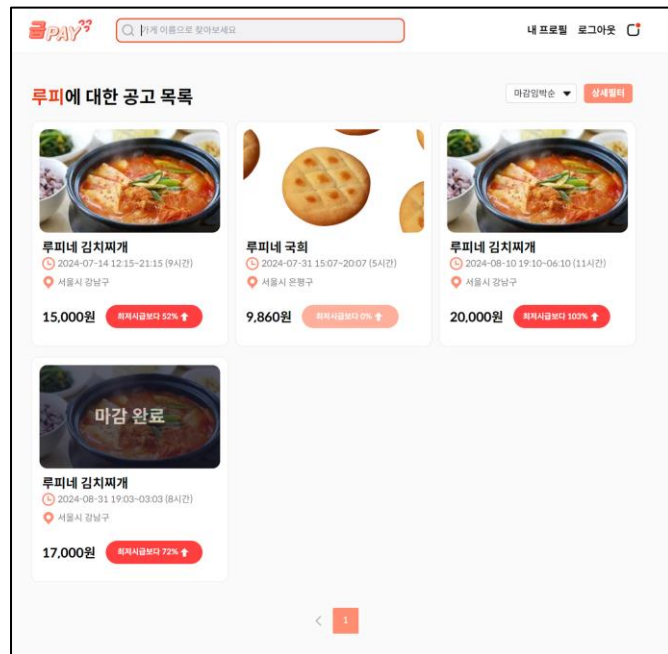
🌟 광고 리스트 페이지(로그인)



주소에 해당하는 광고가 없으면
시급이 높은 순으로 맞춤 광고 대체

가게 이름 관련 검색 시
전체 광고가
관련 키워드에 대한 광고 목록으로 대체

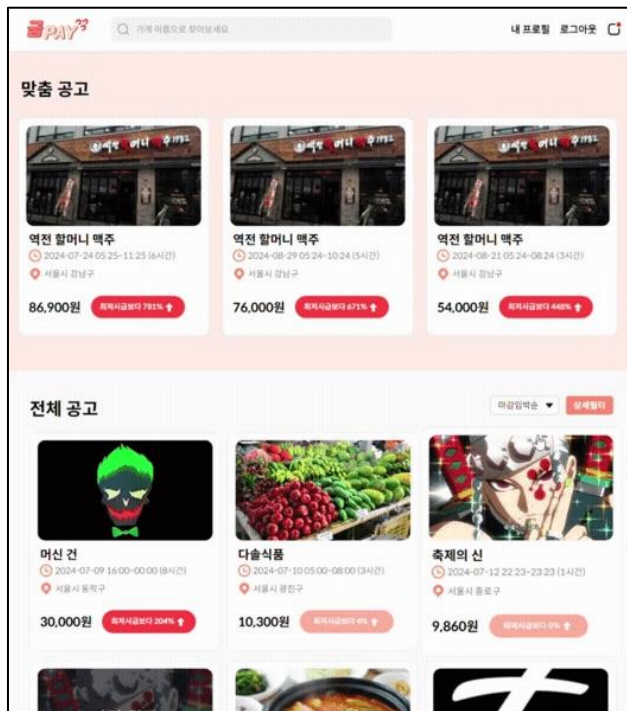
로그인 시
랜딩페이지 보여주지 않고 바로
광고 리스트로 이동



프로젝트 상세

4) 기능소개

🌟 공고 리스트 페이지 - 상세 필터, 공고 정렬



📌 드롭다운을 통해 공고 정렬(마감임박순, 시급높은순, 시간많은순, 가나다순)

📌 ux를 위해 적용된 필터링 저장

📌 검색어 입력시 정렬과 모든 필터링 초기화

```
<input
  className="hidden"
  type="checkbox"
  name="address"
  value={address}
  checked
  onChange={() => handleAddressClick(address)}
/>
```

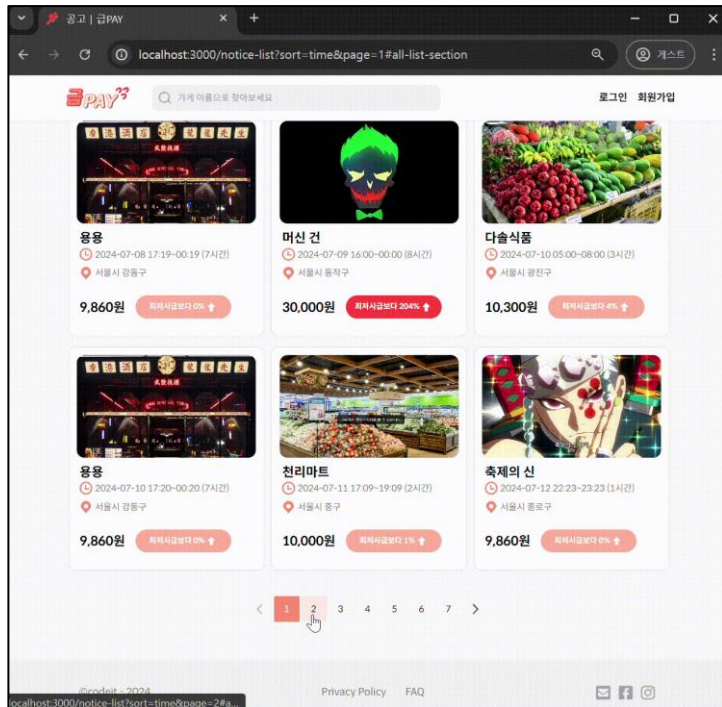
📌 form 형식의 특성상 input으로 값 처리

📌 배열에 저장된 데이터는 입력 불가능 이슈
→ 선택된 주소: checked checkbox로 처리

프로젝트 상세

4) 기능소개

🌟 광고 리스트 페이지 - 페이지네이션



- 🌟 프론트엔드 서버에서 fetch하는 방식
- 🌟 페이지가 많아지면 중앙값에 활성화 한 페이지 정렬
- 🌟 현재 페이지가 중앙값이 되도록 계속해서 리스트를 생성함
- 🌟 기능 구현을 위해 api에서 주는 페이지 정보를 쓰지 않고 계산해서 구현

```
const totalPages = Math.ceil(totalItemsCount / itemsCountPerPage);

const nextPage = totalPages < activePage + 1 ? activePage : activePage + 1;
const prevPage = activePage > 1 ? activePage - 1 : 1;

const midPoint = Math.floor(pageRangeDisplayed / 2);

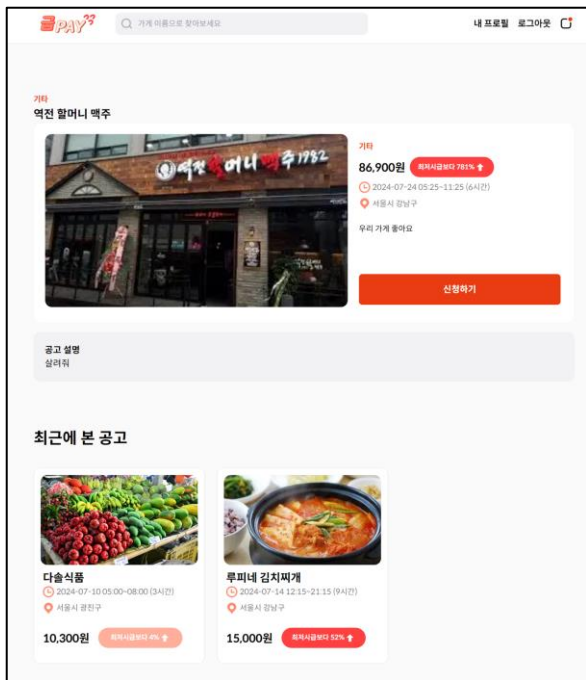
const startPage = Math.max(
  1,
  Math.min(activePage - midPoint, totalPages - pageRangeDisplayed + 1),
);

const endPage = Math.min(
  totalPages,
  Math.max(activePage + midPoint, pageRangeDisplayed),
);
```

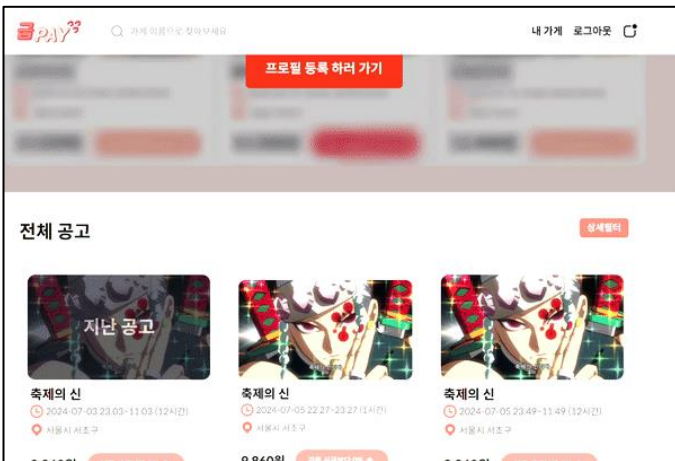
프로젝트 상세

4) 기능소개

🌟 광고 상세 페이지



- 📌 최근 본 광고 구현을 위해 큐 자료구조 선택
- 📌 중복된 데이터 조회시 순서 변경



- 📌 카드 핸들러에 set 함수 설정후
광클시 **사이트 폭파** 이슈 발견
- 📌 광고 상세에서
setCookie를 위한
클라이언트 컴포넌트 제작하여 해결


프로젝트 상세

4) 기능소개

🌟 공고 상세 페이지

📍 (사장) 다른 사장의 공고를 볼 때

기타
다슬식품



기타

10,300원 최저시급보다 4% ↑

🕒 2024-07-10 05:00-08:00 (3시간)

📍 서울시 광진구

식품 납품 업체입니다.


알바생만 신청 가능합니다

공고 설명

들어온 식품 가게에 옮기면 됩니다.

📍 (사장) 내 가게의 공고를 볼 때

한식
루피네 김치찌개



한식

15,000원 최저시급보다 52% ↑

🕒 2024-07-14 12:15-21:15 (9시간)

📍 서울시 강남구

한 입 먹으면 느끼한 속이 싸~악 군침이 났 도는 김치찌개집입니다.

편집하기

공고 설명

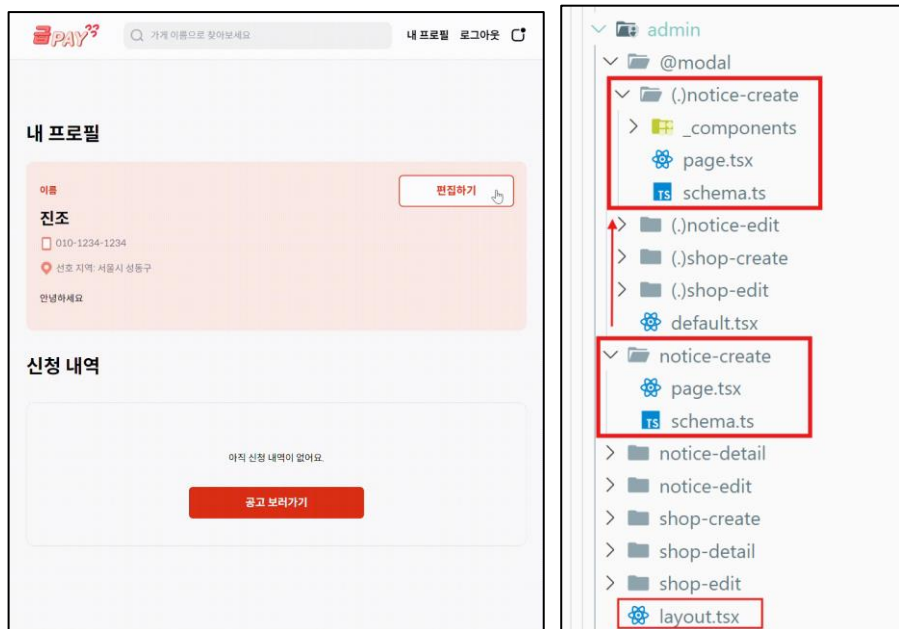
급하게 주방에서 일 할 친구 구해요~!!!! 식대 제공.

프로젝트 상세

4) 기능소개

✨ (알바) 프로필 페이지 - 프로필 등록 및 편집

📍 parallel routes를 사용하여 모달창 구현

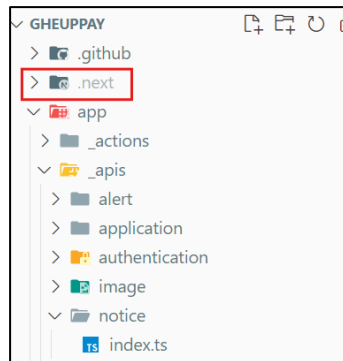


※ parallel routes란? 페이지를 컴포넌트처럼 사용할 수 있다.

💡 폴더링은 정상적으로 들어 갔지만
데이터 캐시 때문에 폴더 이름이 2개 동일해 지면서 나는 오류

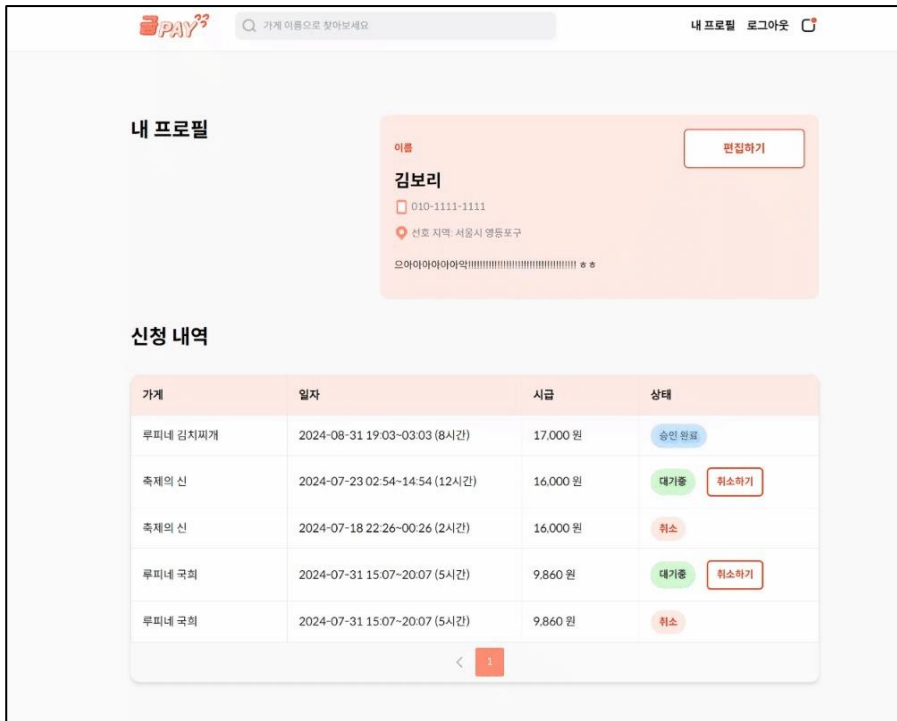
Application error: a client-side exception has occurred (see the browser console for more information).

해결)
.next 파일을 삭제하고 재실행하여 해결



프로젝트 상세 4) 기능소개

🌟 (알바) 프로필 페이지 - 공고 신청 목록



- 유틸리티 타입 Record를 사용해
4가지 상태(pending, accepted, rejected, canceled)에 따른 라벨링 구현

+ 추가기능)

대기중 라벨 옆에 있는 취소하기 버튼 클릭 시, 모달 띄우고 취소할 수 있음

신청 내역	
가게	일자
루피네 국회	2024-07-17 16:50~21:50 (5시간)
루피네 김치찌개	2024-07-14 12:15~21:15 (9시간)
루피네 김치찌개	2024-08-31 19:03~03:03 (8시간)
축제의 신	2024-07-23 02:54~14:54 (12시간)
축제의 신	2024-07-18 22:26~00:26 (2시간)

< 1 2

- 모바일, 테블릿일 때)
가게 컬럼 고정 후
나머지 컬럼 스크롤

프로젝트 상세

4) 기능소개

✨ (사장) 내 가게 페이지 - 내 가게 등록 및 편집

ts.vercelapp/admin/shop-create

PAY

가게 이름으로 찾아보세요

내 가게 로그아웃

9.860 원

가게 이미지 *

GS26

가게 설명

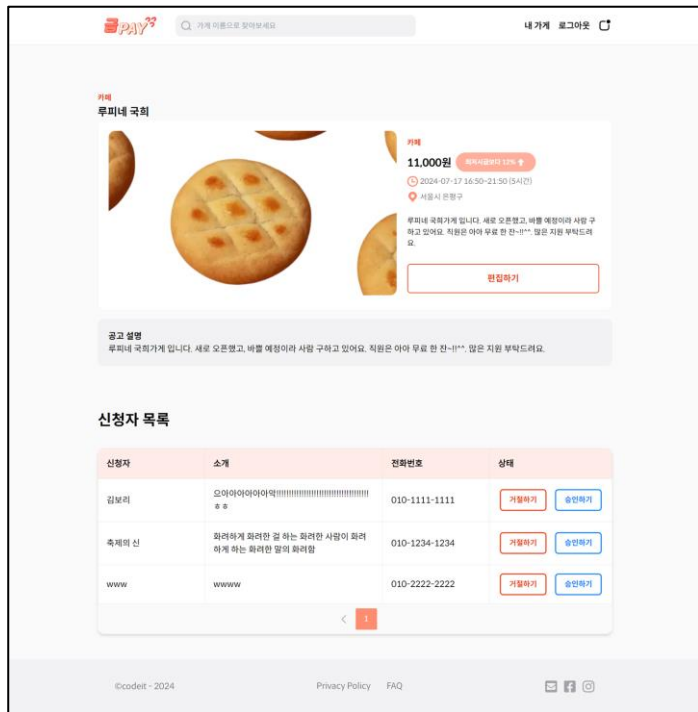
26시간 영업합니다~

📌 parallel routing 활용하여 구현

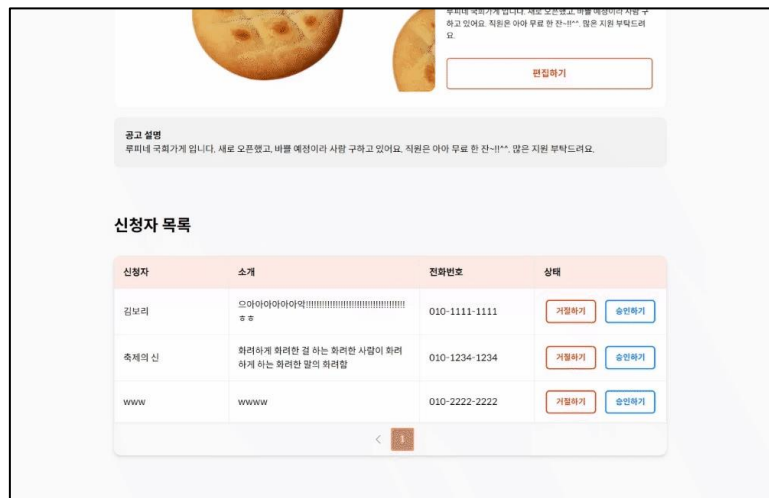
📌 등록 후 뒤로가시기 다시 입력 form으로 오지 않게 replace

프로젝트 상세 4) 기능소개

🌟 (사장) 내 가게 특정 공고 페이지 - 알바생 신청자 목록



- 유틸리티 타입 Record를 사용해
4가지 상태(pending, accepted, rejected, canceled)에 따른 라벨링 구현
- 신청자 대기중일때 거절, 승인 버튼으로 신청자 관리



프로젝트 회고



K

- 📌 끝까지 완성
- 📌 코드 리뷰 퀄리티
- 📌 넥스트 활용
- 📌 시맨틱 태그
- 📌 케이스폴딩

P

- 📌 소통의 오류
- 📌 폴더구조 설계 문제
- 📌 path name restFul
- 📌 쿠키 남발

T

- 📌 초기 설계 단계에서 일어날 수 있는 문제점을 예상해보기
- 📌 path name