



Javascript / SCSS / jQuery / HTML / Ajax

UI/UX PROJECT 003 / PORTFOLIO.

WEB PUBLISHING EDUCATION.

YOO HEE WON.

—

: Open API 활용하기

영풍문고

PORTFOLIO

Please enter the contents here

CONTENTS

1. 주제 선정 - 주제선정

2. 기획 - 페르소나, 와이어프레임, 프로토타입

3. 구현 - COLOR SYS, 코드 구현

4. 마무리 - 평가, 소감

주제 선정

API 를 이용한 프로젝트 목적을 가지고 여러
서점 사이트를 서치 한 결과
영풍 문고가 가장 직관적인
카테고리로 책 정보를 전달 하고 있었고
레이아웃 형식도 심플한 형태라

Open API 를 이용한 프로젝트 라는 취지에
가장 잘 어울릴거라 생각해 영풍 문고 를 참고 하게
되었습니다.

페르소나



이름 : 유은혜 나이 : 28

직업 : 도서 평론가 / 신간 큐레이터

디지털 성향 : 뉴스레터 구독, 웹 기반 독서 플랫폼 적극 이용

PAIN

- 리스트는 많은데 정보가 너무 무질서하고 기계적
- 책이 너무 많고, 무슨 책이 주목받는지도 헷갈림.

NEEDS

- 신간 큐레이션의 정확성과 속도
- 자신의 비평 노출 공간 확보
- 책 선정에 드는 시간 절감

OPEN API

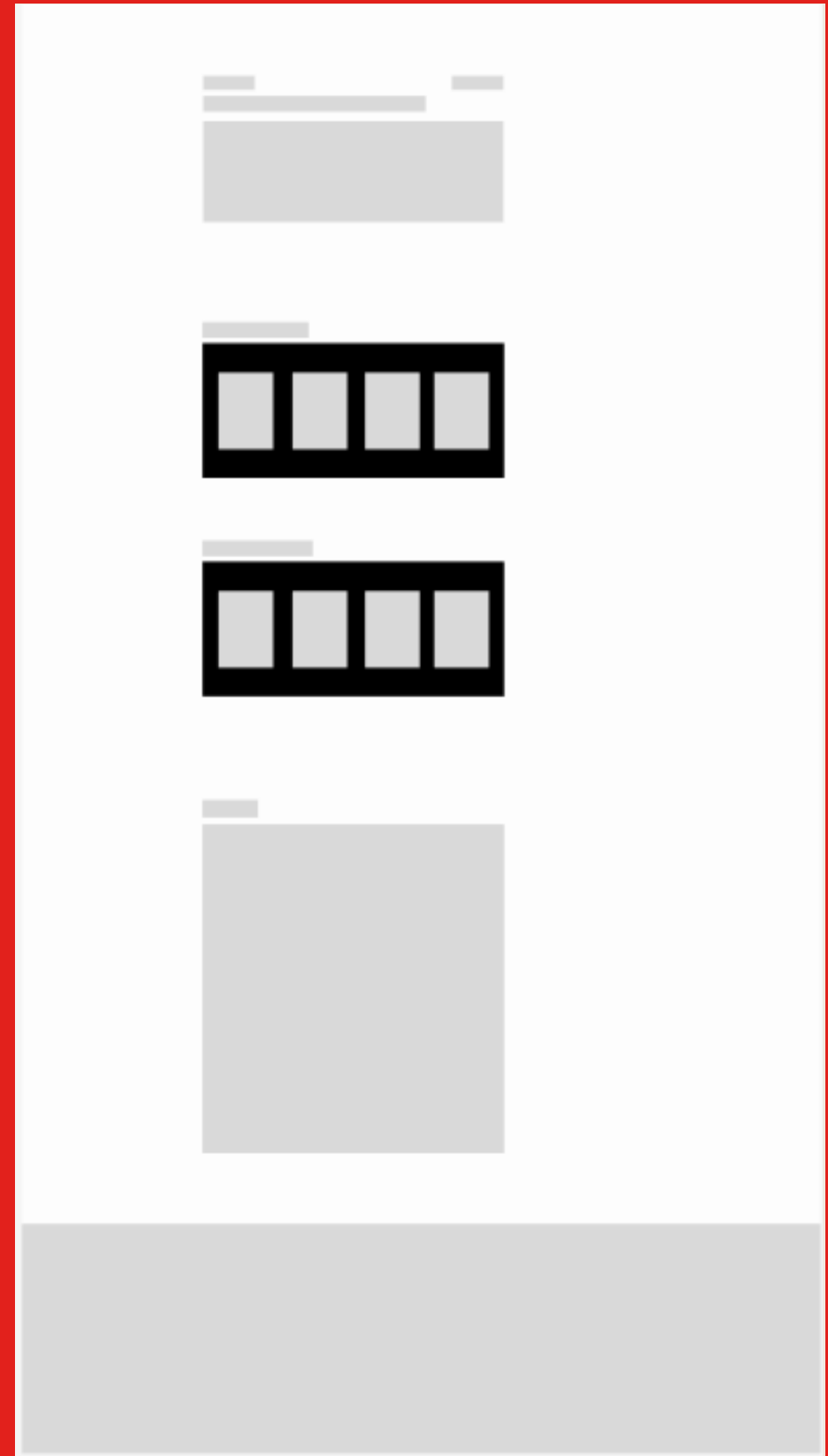
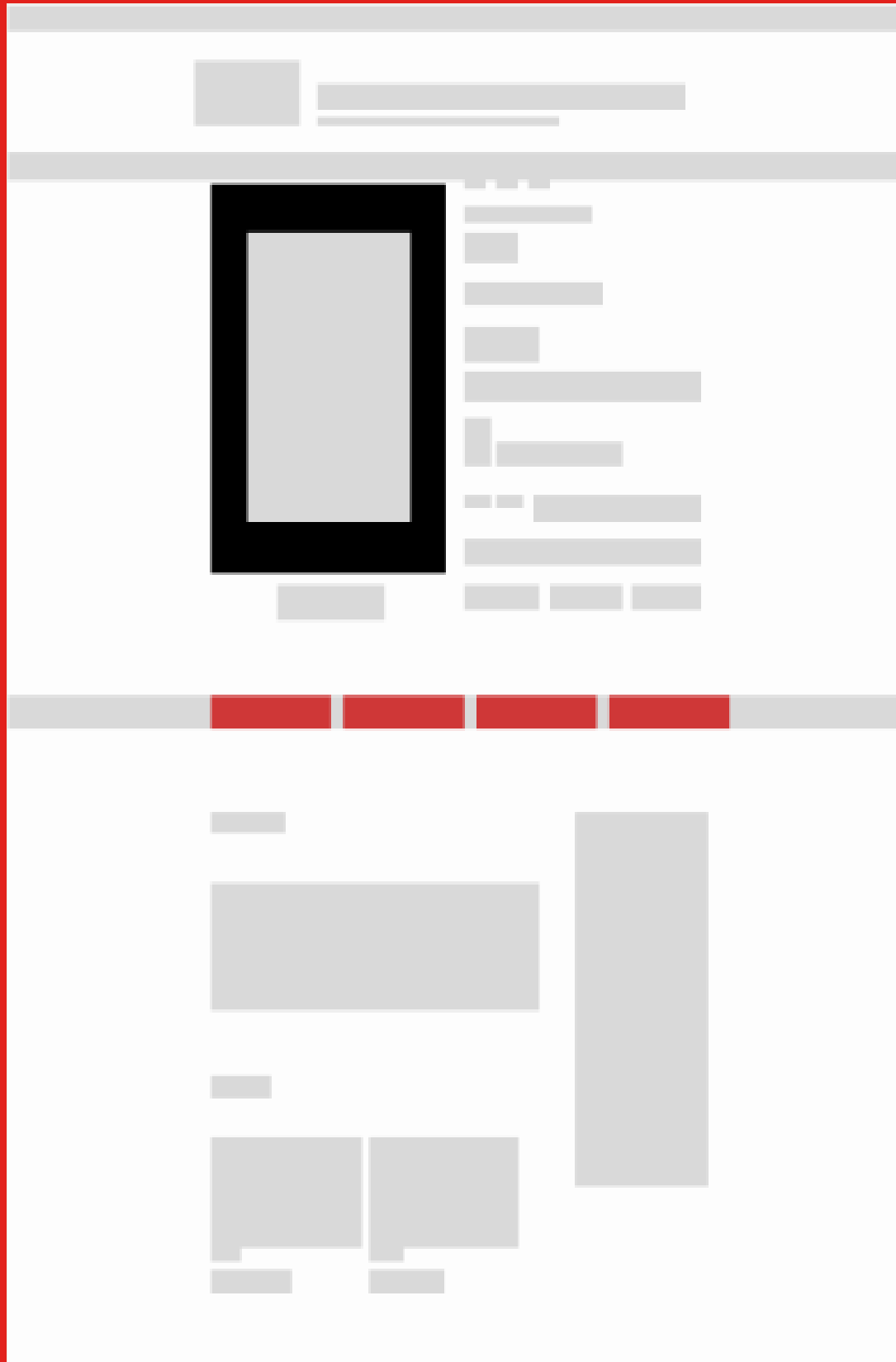
PROJECT

WIREFRAME

OPEN API

PROJECT

WIREFRAME



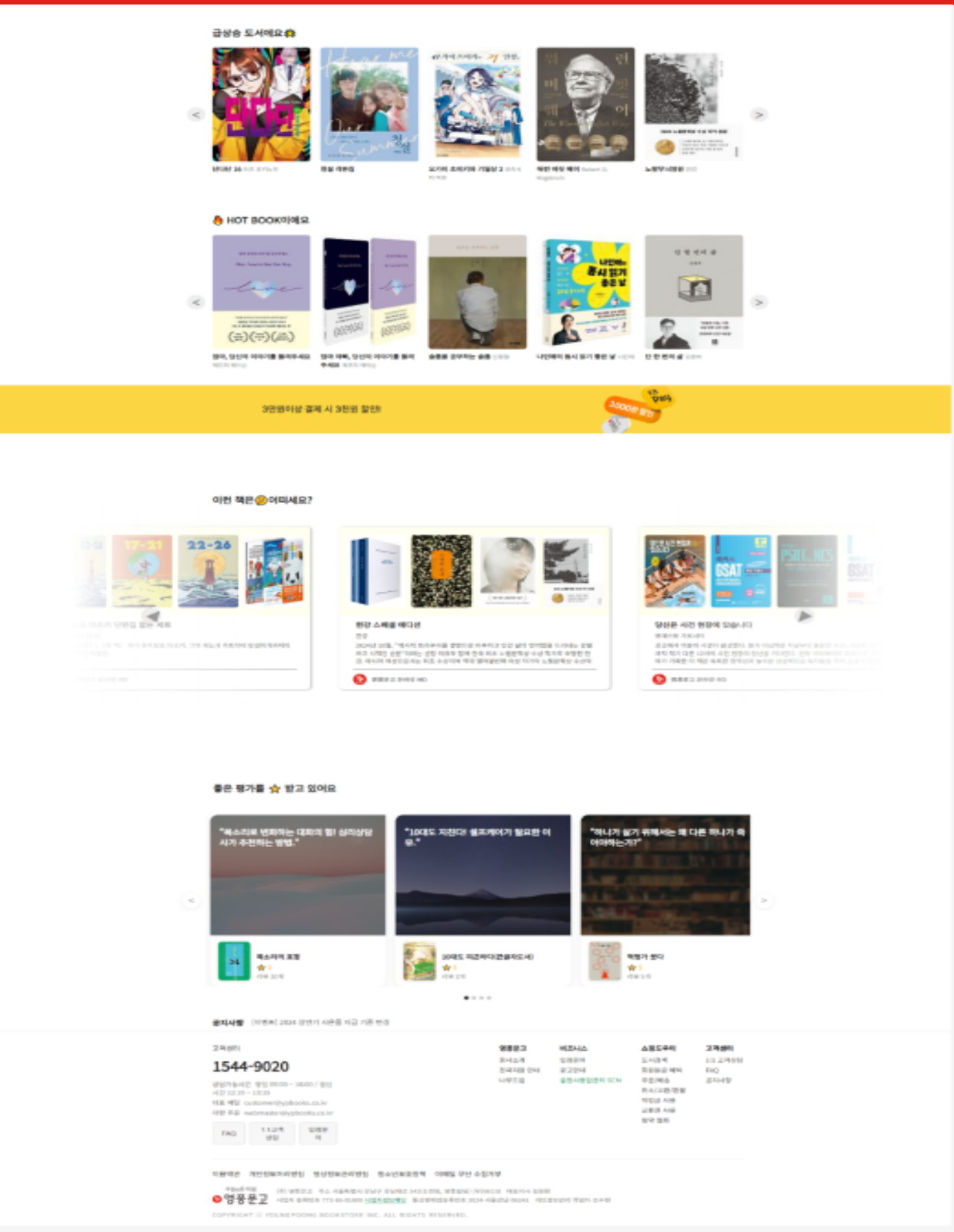
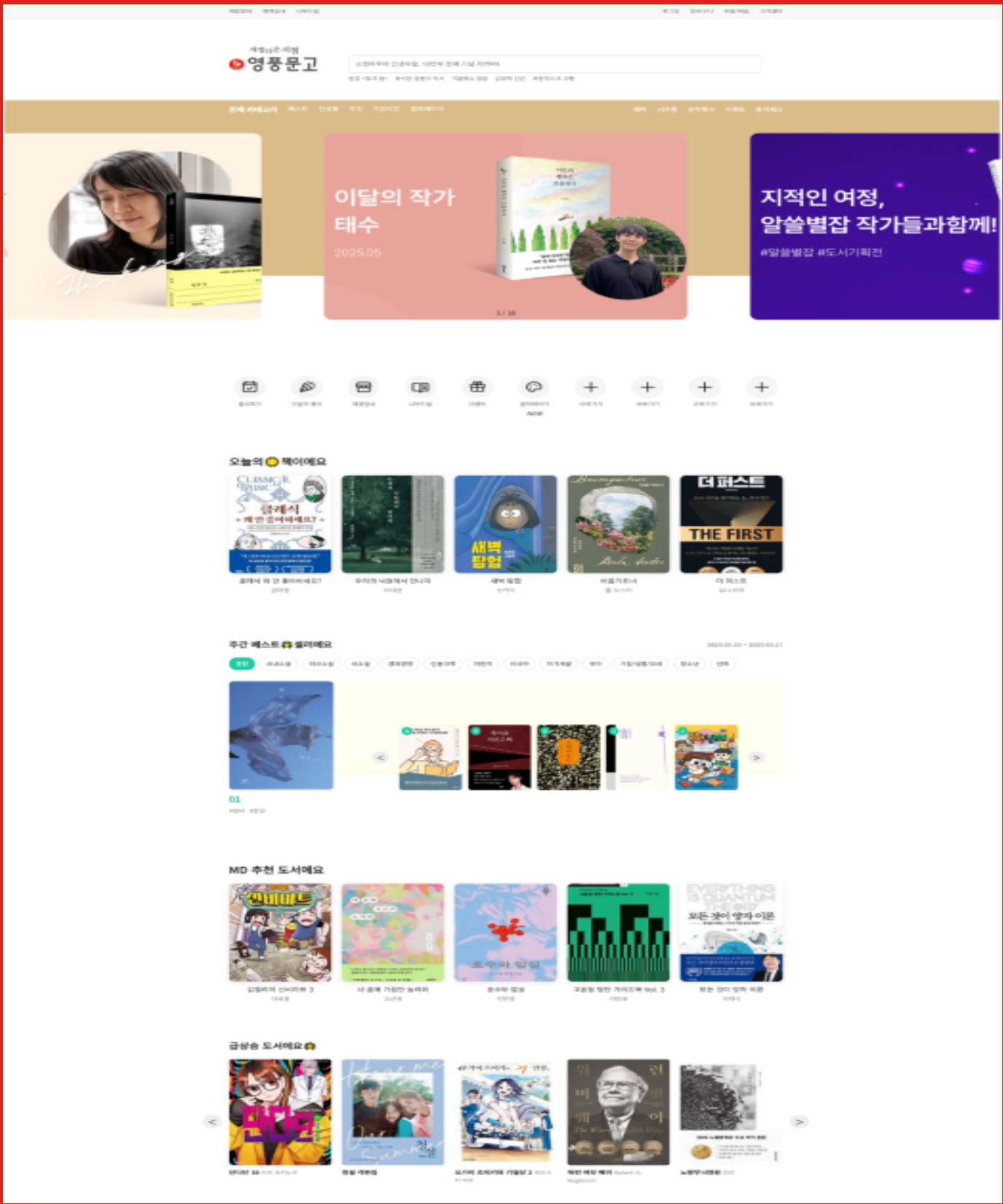
프로토타입 메인 페이지

01

OPEN API

PROJECT

PROTOTYPE



프로토타입

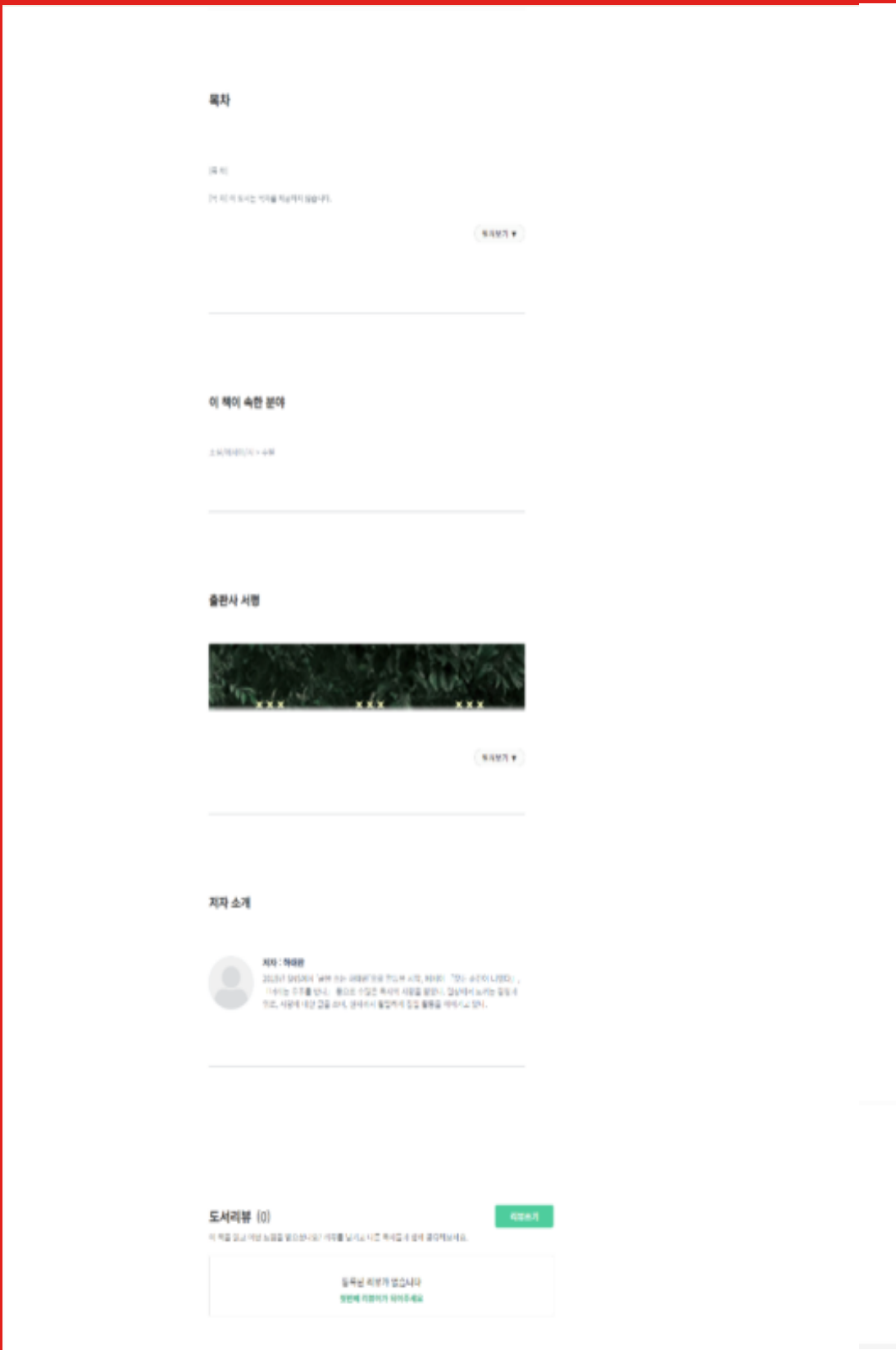
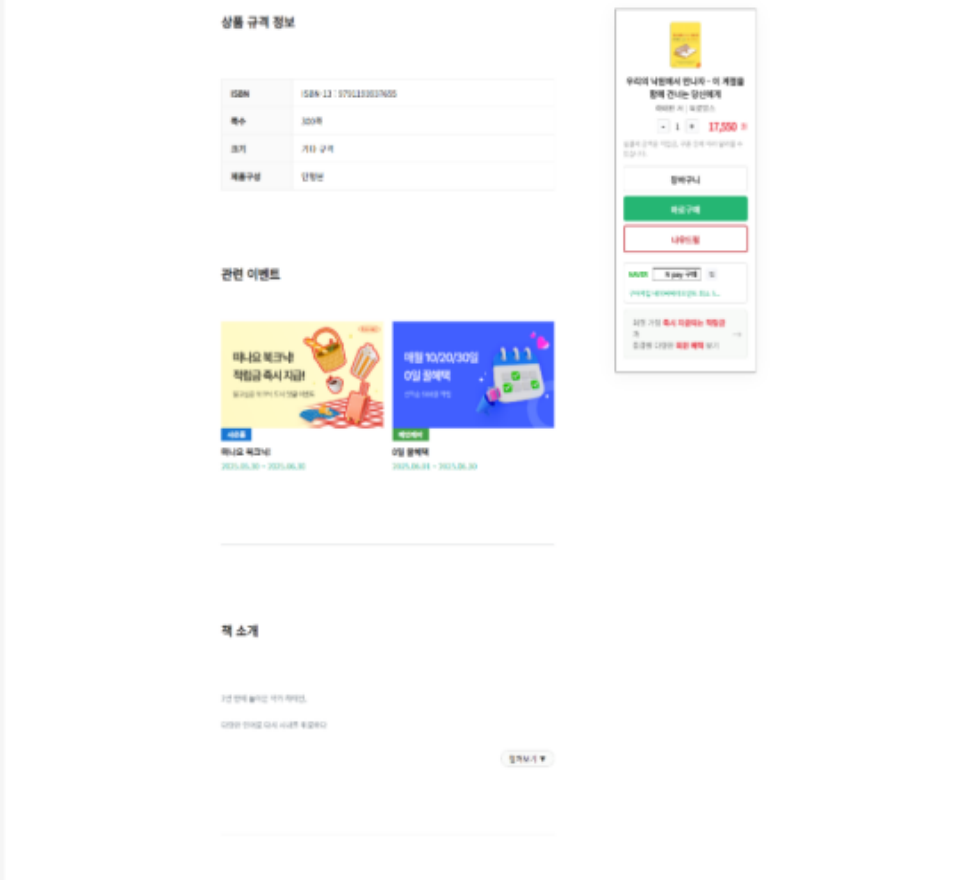
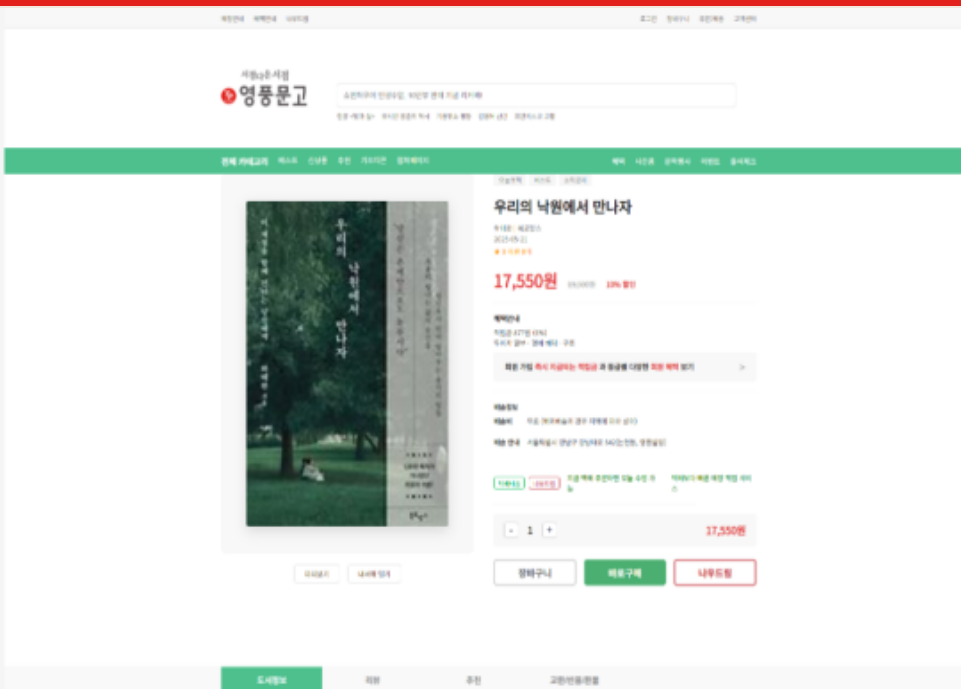
서브 페이지

02

OPEN API

PROJECT

PROTOTYPE



COLOR SYS

영풍문고 와 최대한 비슷하게 구현해야
했기 때문에 영풍문고 로고에 있던 색을
참고 했으며 메인 로고 속 붉은 색상인
#E2211C 를 메인 컬러로 선정하고
작업하였습니다.

YP
BOOKS

해당 스크립트는 GNB(Global Navigation Bar)의 드롭다운 메뉴를 마우스 호버 기반으로 제어하는 로직입니다. 사용자의 움직임에 따라 메뉴가 자연스럽게 노출되거나 닫히도록 구성되어 있으며, UX 측면에서 흔히 발생하는 깜빡임 현상을 방지하도록 설계되었습니다.

코드리뷰

```
document.addEventListener('DOMContentLoaded', () => {
  const dropdownTrigger = document.querySelector('.has-dropdown');
  const dropdownMenu = document.querySelector('.gnb_dropdown');

  if (!dropdownTrigger || !dropdownMenu) {
    console.warn("⚠️ 드롭다운 대상 요소가 없습니다.");
    return;
  }

  dropdownTrigger.addEventListener('mouseenter', () => {
    dropdownMenu.style.display = 'block';
  });

  dropdownTrigger.addEventListener('mouseleave', () => {
    setTimeout(() => {
      if (!dropdownMenu.matches(':hover')) {
        dropdownMenu.style.display = 'none';
      }
    }, 100);
  });

  dropdownMenu.addEventListener('mouseenter', () => {
    dropdownMenu.style.display = 'block';
  });
});
```

이 코드는 Kakao Book API를 활용하여 특정 키워드 배열에 따라 책 데이터를 비동기적으로 요청하고, 썸네일과 설명이 유효한 도서만 필터링한 후, UI 슬라이드 요소에 자동 렌더링하는 기능을 구현합니다.

```
// 이런책 불러오기
async function fetchBooks(query) {
  const params = new URLSearchParams({
    target: "title",
    query,
  });
  const url = `https://dapi.kakao.com/v3/search/book?${params}`;

  const response = await fetch(url, {
    method: 'GET',
    headers: {
      Authorization: `KakaoAK 7dc8a40298c87972a469f758f14bd142`
    }
  });

  if (!response.ok) {
    throw new Error(`HTTP 오류: ${response.status}`);
  }

  return response.json();
}
```

```
async function bookDataMd() {
  try {
    const querys = ['후지모토', '한강', '추리', '바르가스', 'AI', '티니핑', '리그오브', '사계절', '경제', '트럼프'];

    querys.forEach(async (query, i) => {
      const data = await fetchBooks(query);

      //썸네일이 빈 문자열인것은 제외
      const origin = data.documents;
      let book = origin.filter((val) => {
        return val.thumbnail != '' && val.contents != '';
      });

      const divs = $('#md_slide').find('.md_list').eq(i);
      divs.append(`
        <div class="md_list_img">
          <img src=${data.documents[0].thumbnail}/>
          <img src=${data.documents[1].thumbnail}/>
          <img src=${data.documents[2].thumbnail}/>
          <img src=${data.documents[3].thumbnail}/>
        </div>
        <div class="md_list_text">
          <h3>${data.documents[0].title}</h3>
          <p>${data.documents[0].authors}</p>
          <span>${data.documents[0].contents}</span>
        </div>
        <div class="md_list_logo">
          
          <span>영풍문고 온라인 MD</span>
        </div>
      `);
    });
  } catch (error) {
    console.log('에러발생', error);
  }
}
```

해당 스크립트는 서브페이지 하단의 탭 콘텐츠(책소개, 목차, 출판사 서평 등)를 사용자가 직접 접거나 펼칠 수 있도록 구현한 토글 기능입니다.

버튼 클릭 시 콘텐츠 영역의 클래스가 전환되며, 이에 따라 콘텐츠가 열리거나 닫히고, 버튼 문구 역시 상태에 맞게 동적으로 변경됩니다.

콘텐츠 전환

```
//탭 콘텐츠 영역 (서브페이지 하단)
// (접고 펼치기)
document.addEventListener('DOMContentLoaded', function() {
  function setupToggle(boxId, btnId) {
    const box = document.getElementById(boxId);
    const btn = document.getElementById(btnId);
    btn.addEventListener('click', function() {
      const isFold = box.classList.contains('fold');
      if (isFold) {
        box.classList.remove('fold');
        box.classList.add('open');
        btn.textContent = '접어보기 ▲';
      } else {
        box.classList.remove('open');
        box.classList.add('fold');
        btn.textContent = '펼쳐보기 ▼';
      }
    });
  }
  setupToggle('descBox', 'descToggleBtn'); // 책소개
  setupToggle('indexBox', 'indexToggleBtn'); // 목차
  setupToggle('reviewBox', 'reviewToggleBtn'); // ★ 출판사 서평 추가!
});
```

해당 스크립트는 서브페이지 하단 탭 네비게이션 클릭 시, 연결된 콘텐츠 섹션으로 부드럽게 스크롤 이동시키는 기능을 수행합니다.

고정된 상단 네비게이션 높이를 자동 계산하여 콘텐츠가 가려지지 않도록 보정하고, 탭 선택 상태는 *active* 클래스로 시각적으로 구분됩니다.

data-target 속성을 기반으로 동작하며, 항목 수에 관계없이 유연하게 확장 가능한 구조로 설계되어 있습니다.

코 드 구 현

```
// 하단 카테고리
document.querySelectorAll('.detail_nav_item').forEach(btn => {
  btn.addEventListener('click', function() {
    const targetId = btn.dataset.target;
    const targetEl = document.getElementById(targetId);

    if (targetEl) {
      const navHeight = document.querySelector('.detail_nav_wrap').offsetHeight;
      // 스틱 네비 높이만큼 보정 (탭바가 섹션 가리지 않게!)
      const top = targetEl.getBoundingClientRect().top + window.scrollY - navHeight;
      window.scrollTo({ top, behavior: 'smooth' });
    }
    // 탭 active 스타일 처리
    document.querySelectorAll('.detail_nav_item').forEach(el => el.classList.remove('active'));
    btn.classList.add('active');
  });
});
```

프로젝트를 마치며

이번 프로젝트를 통해 API를 활용한 데이터 연동 방식과 자바스크립트를 기반으로 한 기능 구현에 대해 깊이 있게 접근할 수 있는 계기가 되었습니다.

특히, 간단한 기능을 구현하더라도 자바스크립트가 다양한 방식으로 조합되어야 한다는 점에서 많은 시행착오를 겪었고, 이를 통해 기능 단위로 코드를 구조화하는 사고 방식을 익히게 되었습니다.

API의 개념과 흐름을 이해하는 데에는 다소 시간이 걸렸지만, 반복적인 학습을 통해 조금씩 익숙해졌으며, 이제는 사이트를 구성할 때 전체적인 틀과 섹션 구조가 머릿속에 자연스럽게 떠오를 정도로 기획과 구현 간의 연계가 어느 정도 정립되었다고 느낍니다.

3번째 프로젝트까지 완수하며 기본적인 퍼블리싱과 동적 기능 구현에 대한 자신감은 다소 생겼지만, 아직도 디테일한 부분에서는 미숙함이 남아 있고, 실전 감각과 깊이 있는 코드 설계 능력은 더 많은 수련이 필요하다고 생각합니다.

다음으로 진행할 React 기반 프로젝트에서는 이번 경험을 바탕으로 보다 구조적이고 완성도 높은 결과물을 만들 수 있도록 노력하겠습니다.

OPEN API

PROJECT

END

PORTFOLIO

2025-06-10

THANK YOU!

YOO HEE WON

UI / UX

OPEN API