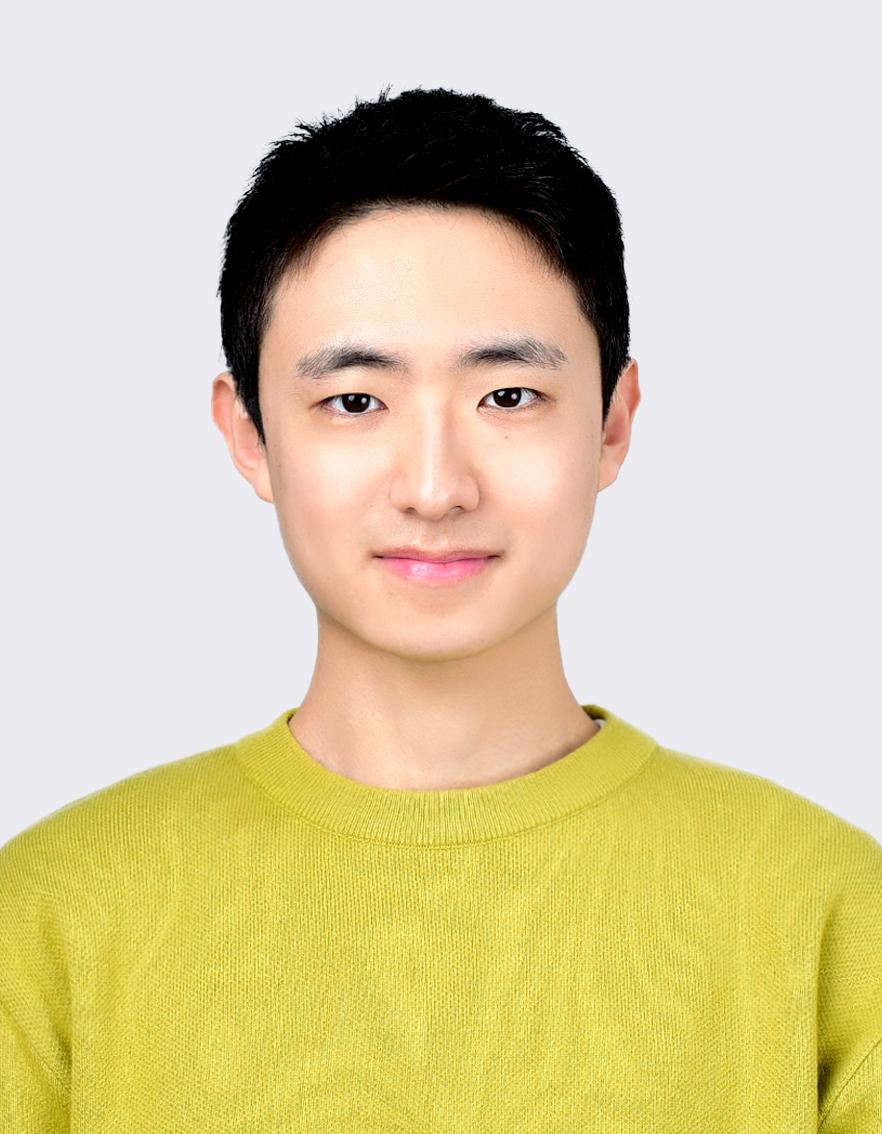
임동민

Frontend Engineer

**phone : 010-7194-5308**

**Email : dongmin11566@gmail.com**

**Github :** [**https://github.com/dongmin115**](https://github.com/dongmin115)

**Blog :** [**https://velog.io/@dongmin115/posts**](https://velog.io/@dongmin115/posts)

# Summary

| API 요청 시 반복되는 쿼리 파라미터 처리 로직을 개선하기 위해, 값이 존재하는 경우에만 파라미터를 추가하는 유틸리티 함수를 생성하여 코드 중복을 제거하고, 유지 보수성과 확장성을 높여 팀 전체의 개발 효율성을 향상시켰습니다.  이미지 성능 최적화를 위해 다양한 방안을 분석하고 개발자 도구의 네트워크 탭을 통해 로딩 시간을 비교하며 최선의 방법을 선별하여 적용하였습니다.  기획부터 배포까지 개발의 모든 과정에 직접 참여하고, 다양한 팀원들과 협업하여 프로젝트를 주도한 경험이 있습니다. | |
| --- | --- |

# Project

| **Dream Vault : 음악 공유 웹 플랫폼**  기술스택 : Next, Typescript, Tanstack-query, TailwindCSS, MSW | 2024.03 ~ 06 |
| --- | --- |
| 성능 최적화와 공통 컴포넌트 개발로 코드 재사용성 및 유지보수성 개선을 경험한 프로젝트   * 무한 스크롤 공통 컴포넌트 개발   + 프로젝트 내 여러 페이지에서 무한 스크롤 기능이 필요하다는 것을 파악하여 이를 공통 컴포넌트로 추상화하여 개발   + **Tanstack-query**의 **useInfiniteQuery**와 **Intersection Observer API**를 결합하여 스크롤 위치를 감지하여 다음 데이터를 자동으로 로드하는 로직을 컴포넌트 내부에 구현   + **queryKey, queryFn, renderItem, getNextPageParam, dataPath**를컴포넌트의 **props**로 두어 다양한 데이터 구조와 API 엔드포인트에 대응할 수 있도록 설계 * 성능 최적화   + 사용자와 상호작용 할 필요가 없는 컴포넌트들을 서버 컴포넌트로 분리해 **서버 사이드 렌더링**을 적용하여 초기 로딩 시간 개선   1. 음악 플랫폼 특성상 많은 앨범 커버 이미지를 다루어야 했기 때문에 Next.js의 **Image** 컴포넌트와 **Webp** 형식의 이미지를 활용하여 **이미지 로딩시간 3초 -> 1.8초로 단축**  1. 액세스 토큰 만료 대응 로직 구현    1. **액세스 토큰의 만료로 인한 사용자 재로그인 불편을 해결하기 위해**, 모든 API 요청 전에 Axios **요청 인터셉터**를 통해 액세스 토큰을 Authorization 헤더에 자동으로 추가하도록 설정    2. 응답에서 401 Unauthorized 오류 발생 시, **리프레시 토큰을 사용하여 새로운 액세스 토큰을 발급받고 실패한 요청을 재시도하도록** 응답 인터셉터를 설정했으며, 토큰 재발급에도 실패하면 사용자에게 재로그인이 필요하다는 안내를 제공하도록 예외 처리를 구현    3. 이를 통해 **토큰 만료 시 자동으로 새로운 액세스 토큰을 발급받고, 실패한 요청을 재시도하는 로직을 구현**하여 문제를 해결 2. 팀 내 개발 프로세스 개선    1. **Mock Service Worker**를 활용하여 API 엔드포인트를 모의 구현하여 백엔드 API 개발에 의존하지 않고 독립적으로 개발을 진행할 수 있게 되어 팀 내 개발 일정 관리의 효율성을 향상 | |
| **링크** : [Github](https://github.com/DreamVault-2024/dreamvalut-frontend) | |

# 

| **Brain Washer : 독한 말 서비스**  기술스택 : React, Typescript, TailwindCSS | 2024.06 ~ 2024.08 |
| --- | --- |
| 오디오와 텍스트를 사용자에게 실시간으로 스트리밍하여 사용자 경험을 개선한 프로젝트   * 스트림 데이터 파싱 오류 해결 [[blog]](https://velog.io/@1mdongmin/%ED%8C%8C%EC%8B%B1-%ED%8A%B8%EB%9F%AC%EB%B8%94-%EC%8A%88%ED%8C%85)   + **문제 발생**: 서버에서 텍스트와 오디오 데이터를 스트림 방식으로 실시간으로 받아오는 과정에서, 데이터 파싱 중 오류 발생. 스트림 데이터가 여러 줄로 나뉘어 있을 때, data:를 기준으로 텍스트를 파싱하는 과정에서 특정 청크에 여러 개의 data: 필드가 포함되어 있어 빈번하게 오류 발생   + **해결 방법**: 로깅을 통해 문제 발생 지점을 파악한 후, 스트림이 여러 줄에 걸쳐 전달될 때 부분 메시지를 병합하고, 데이터가 완전히 수신될 때까지 기다렸다가 처리하는 방식으로 개선. 불완전한 청크 데이터를 유지하는 변수를 도입하여, 완전한 데이터가 수신된 후에만 이를 파싱하는 로직으로 수정   + **결과**: 텍스트와 오디오 데이터가 정확하게 파싱되어 실시간 스트리밍 기능을 안정적으로 제공할 수 있게 됨 * 오디오 마지막 부분 재생 오류 해결 [[blog]](https://velog.io/@1mdongmin/%EC%98%A4%EB%94%94%EC%98%A4-%EC%8A%A4%ED%8A%B8%EB%A6%AC%EB%B0%8D-%EB%A7%88%EC%A7%80%EB%A7%89-%EC%B2%AD%ED%81%AC-%EB%88%84%EB%9D%BD-%EC%9D%B4%EC%8A%88)   1. **문제 발생**: 오디오 스트리밍 중 마지막 청크 데이터가 재생되지 않는 문제가 발생. 서버로부터 받은 오디오 청크 데이터(31개)와 실제 오디오 플레이어 컴포넌트에 전달된 데이터(82개) 간 불일치가 발생함   2. **해결 방법**: 오디오 플레이어 초기화 로직 추가 및 마지막 청크 수신 후 재생 완료 테스트를 진행했으나 문제 해결되지 않음. 추가 분석 결과, 마지막 청크 데이터 형식이 다른 데이터와 다르다는 것을 발견하고, 이를 백엔드 팀에 전달하여 백엔드 로직 수정   3. **결과**: 오디오 스트리밍의 품질과 신뢰성을 개선하여, 사용자에게 지연 없이 오디오와 텍스트를 실시간으로 제공할 수 있게 됨 | |
| **링크** : [Github](https://github.com/2024-SummerBootcamp-Team) | |

# 

| **Wish Tree : 소원 공유 서비스**  기술스택 : Nest, React, Typescript, TailwindCSS | 2024.08.24 ~ 2024.08.25 |
| --- | --- |
| 하루 동안 요구사항을 만족하는 서비스를 개발한 해커톤 경험 프로젝트   * 유닛 테스트 코드 작성 및 코드 안정성 향상   + 서비스의 주요 메서드에 대한 유닛 테스트 코드를 작성하여 코드의 안정성과 신뢰성을 높임   + **NestJS**의 **TestingModule**과 **jest**를 활용하여 서비스와 리포지토리를 모킹(mocking)하고, 다양한 시나리오에 대한 테스트를 수행   + 소원을 찾을 수 없을 때 **NotFoundException**을 발생시키는 예외 처리 로직에 대한 테스트를 추가하여 코드의 견고성 높임 * 함수 호출의 가독성 및 사용성 개선   + 모든 소원을 불러오는 **findAll** 함수에서 선택적 매개변수인 **page**와 **limit**을 함수 호출 시마다 모두 입력해야 하는 불편함 발생   + 구조 분해 할당과 기본값 설정을 활용하여 매개변수를 객체로 받아오도록 수정하여 **page**와 **limit**을 선택적으로 입력할 수 있게 되어 함수 호출의 가독성과 사용성 향상 | |
| **링크** : [Back-end](https://github.com/techeer-sv/Good-Night-3rd-Hackathon-Backend/pull/5) / [Front-end](https://github.com/techeer-sv/Good-Night-3rd-Hackathon-Frontend/pull/1) | |

|  | |
| --- | --- |
| **On My Desk : 데스크탑 셋업 공유 웹 플랫폼**  기술스택 : React, Typescript, TailwindCSS, Zustand | 2024.03 ~ 2024.06 |
| 시각적으로 매력적인 사용자 인터페이스에 대한 관심으로 시작되어, 머릿속 아이디어를 실제로 구현하는 쾌감을 통해 프론트엔드의 매력에 본격적으로 빠지게 된 프로젝트   1. React Hook Form과 Zod를 활용한 폼 관리 및 유효성 검증 [[blog]](https://velog.io/@dongmin115/React-Hook-Form%EA%B3%BC-zod%EB%A5%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EC%97%AC-%ED%9A%A8%EC%9C%A8%EC%A0%81%EC%9C%BC%EB%A1%9C-%ED%8F%BC-%EA%B4%80%EB%A6%AC%ED%95%98%EA%B8%B0)    1. 불필요한 리렌더링을 방지하기 위해서 **React Hook Form**의 **useForm** 훅을 사용하여 mode를 **onSubmit**으로 설정하여 비제어 컴포넌트로 구현    2. **Zod**를 사용하여 유효성 검사 스키마를 정의하고 **React Hook Form**과 결합하여 폼 필드의 유효성 검사 결과를 바탕으로 자동적으로 에러 메시지를 생성하고 관리하여 에러 핸들링 구현 2. 핫 리로딩 오류 해결    1. **문제 발생**: 보일러 템플릿을 사용해 개발 중, TailwindCSS 스타일 변경 사항이 새로고침 시 적용되지 않고, 빌드를 다시 해야만 반영되는 문제가 발생    2. **해결 방법**: 보일러플레이트의 초기 코드를 분석한 결과, 빌드를 통해 생성되는 **tailwindcss-output.css** 파일이 **index.html** 에 연결되어 있는 것을 발견하여 개발 모드에서 변경 사항을 즉시 확인할 수 있도록 **input.css** 파일로 연결을 변경하여 문제를 해결    3. **결과**: 빌드 없이도 새로고침만으로 TailwindCSS 스타일 변경 사항이 즉시 반영되도록 개선 | |
| **링크** : [Github](https://github.com/Team-OMD/Frontend) | |

| **개인 포트폴리오 프로젝트**  기술스택 : React, Typescript, Three.js, TailwindCSS, DaisyUI | 2024.02 ~ 2024.02 |
| --- | --- |
| 시각적으로 매력적인 사용자 인터페이스에 대한 관심으로 시작되어, 머릿속 아이디어를 실제로 구현하는 쾌감을 통해 프론트엔드의 매력에 본격적으로 빠지게 된 프로젝트   1. 3D 인터페이스    1. **Three.js** 와 **GLTFLoader** 를 사용하여 3D 모델 ( 배경 및 리액트 로고 ) 을 웹사이트에 통합    2. **WebGL 렌더러**를 사용하여 고해상도의 3D 콘텐츠를 효율적으로 렌더링하고 브라우저의 프레임 레이트와 호환되도록 성능을 최적화하여 부드러운 사용자 인터페이스를 제공    3. **Three.js** 의 조명 기능을 활용하여 리얼리즘을 높이고, 3D 객체에 그림자를 추가하여 깊이감을 증가 2. 마우스 호버 효과    1. **pointermove** 이벤트를 활용하여 사용자의 마우스 위치에 따라 오버레이 요소의 스타일을 실시간으로 변경하고 **CSS 변수 (--x, --y)** 를 동적으로 업데이트하여 마우스 위치에 따른 시각적 효과 제공    2. 다양한 디스플레이에서도 일관된 경험을 할 수 있도록 보장하기 위해 각 카드의 크기 변화를 감지하여 해당 크기에 맞게 오버레이 카드의 크기를 동적으로 조정 3. 랜덤 그라데이션 색상 변경 효과    1. **jQuery** 와 순수 **JavaScript** 를 사용하여 그라데이션 색상이 시간에 따라 부드럽게 변화하는 애니메이션 효과를 구현    2. **requestAnimationFrame** 을 활용하여 브라우저의 리프레시 레이트에 맞춰 애니메이션을 업데이트 함으로써 성능 저하 없이 부드러운 사용자 경험을 제공    3. **CSS** 의 **-webkit-gradient** 와 **-moz-linear-gradient** 속성을 이용해 크로스 브라우징 호환성을 확보 | |
| **링크** : [Github](https://github.com/dongmin115/resume) | |

| | Skill | | | | | --- | --- | --- | --- | | **Framework & Library** | **Language** | **Styling** | **State Management** | | React.js  Next.js  Nest.js | Typescript  Javascript | Tailwind CSS  styled-components | Tanstack-Query  Zustand |  Activity | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Good Night Hackathon**  프론트엔드와 백엔드를 1인 개발로 완성하는 해커톤 | | 2024.08.24 ~ 2024.08.25 |
| **[티타임즈 x Techeer] 실리콘밸리 SW 부트캠프**  실리콘밸리 엔지니어가 운영하는 End-to-End 프로젝트 개발 부트캠프 리더 | | 2024.06 ~ 2024.08 |
| **Techeer IDEATHON 2024**  아이디어 해커톤 | 2024.05.31 ~ 2024.06.01 | |
| **Techeer**  실리콘밸리 기술 기반 SW 개발자 커리어 그룹 | 2024.02 ~ 활동 중 | |
| **[티타임즈 x Techeer] 실리콘밸리 SW 부트캠프**  실리콘밸리 엔지니어가 운영하는 End-to-End 프로젝트 개발 부트캠프 | | 2023.12 ~ 2024.02 |
| **Techeer Partners**  개발 스터디 활동 | | 2023.09 ~ 2023.12 |

| Education | |
| --- | --- |
| 성결대학교 - 정보통신공학과 | 2019.03 ~ 2025.02 ( 졸업예정 ) |

# 