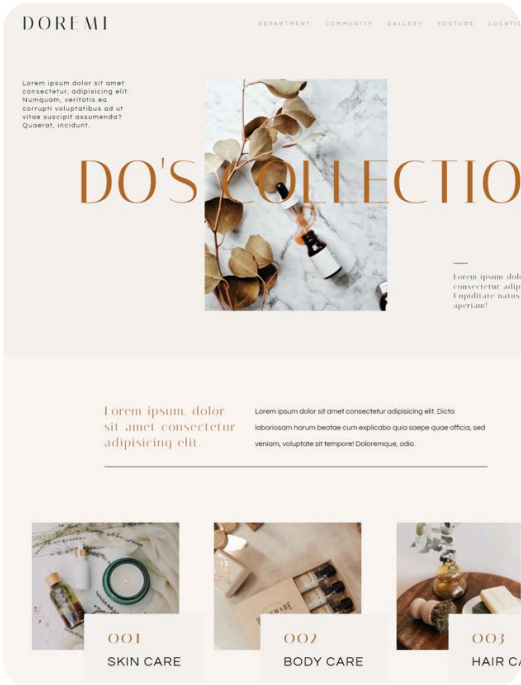


웹표준과 접근성을 고려한 UI 구현에 강점이 있는 프론트엔드 개발자 박민혜입니다.

박민혜 / minhyepark.dev@gmail.com



📌 프로젝트명 : 개인 프로젝트 기업형 웹사이트 (사이트 바로가기)

📅 진행기간 : 2022.01 ~ 2022.02

👤 역할 : 기획자, 디자이너, 프론트엔드 개발자

🔧 기술스택 : React, JavaScript, CSS, SASS(SCSS), Redux

🌐 프로젝트 소개

이 프로젝트는 리액트를 사용하여 기업형 웹사이트를 구현한 개인 프로젝트입니다. 회사 소개, 커뮤니티, 갤러리, 유튜브, 오시는 길, 회원가입 등의 메뉴를 포함하여, 다양한 비즈니스 요구 사항을 반영한 웹사이트를 개발했습니다. 기획부터 디자인, 개발까지 전 과정을 혼자 진행했으며, 반응형 디자인을 적용해 다양한 디바이스에서도 최적의 사용자 경험을 제공하도록 설계했습니다.

🚀 담당 역할 및 기여

1. 기획

- 프로젝트의 전체 기획을 맡아 요구 사항을 수집하고, 사이트 맵 및 기능 요구사항을 정의
- 각 페이지의 목표 및 사용자 흐름을 설계하여 웹사이트 구조 결정

2. 디자인

- UI/UX 디자인을 혼자 진행하며, 사용자 경험을 고려한 인터페이스 설계 및 디자인 시스템 구축
- 공통 스타일 가이드와 색상, 폰트, 레이아웃 등을 정의하여 일관된 디자인 유지

3. 프론트엔드 개발

- React를 사용하여 각 페이지 및 기능을 구현
- Redux를 활용하여 상태 관리 및 API 연동 처리
- CSS/SASS(SCSS)를 사용하여 스타일링 및 반응형 웹 디자인 구현
- YouTube, Flickr, Kakaomap API를 연동하여 기능 추가
- 쿠키 및 setInterval을 활용한 사용자 경험 개선 기능 구현

🎯 주요 성과

- ✅ 혼자서 기획, 디자인, 개발까지 진행하며 전체적인 프로젝트 흐름을 파악하고 관리하는 경험을 쌓음
- ✅ 반응형 웹 디자인을 통해 다양한 디바이스에서 최적화된 사용자 경험을 제공
- ✅ API 응용 및 기능 개발을 통해 실용적인 기능을 구현하고, 사용성 향상
- ✅ 크로스 브라우징을 고려한 개발로, 다양한 브라우저에서 안정적인 동작 보장

🕒 어려웠던 부분

- React를 처음 접했기 때문에, 데이터가 변경될 때 화면이 자동으로 리렌더링된다는 개념 자체가 어려웠다. 이로 인해 상태 변화를 다루는 방식과 렌더링 최적화 방법을 이해하는 데 시간이 걸렸다.
- Redux의 동작 원리를 처음에는 잘 이해하지 못했으며, 상태를 어떻게 관리하고 업데이트해야 하는지 익숙하지 않아, 상태 구조를 설계하고 활용하는 데 어려움을 겪었다.
- 전역 상태와 컴포넌트별 상태 관리의 차이점을 명확하게 이해하지 못해, 불필요한 상태 변경으로 인해 예상치 못한 리렌더링이 발생하는 문제를 겪기도 했다.

📝 회고

이 프로젝트를 통해 리액트를 처음 접하며 상태 관리와 렌더링 동작을 깊이 이해하는 계기가 되었다.

처음에는 데이터 변경 시 자동으로 화면이 갱신되는 개념이 익숙하지 않아 리렌더링이 언제, 왜 발생하는지 이해하는 데 어려움이 있었다. 또한, Redux를 사용하면서 전역 상태 설계와 데이터 흐름을 제대로 파악하지 못해 시행착오를 겪었다. React의 렌더링 방식과 상태 업데이트 흐름을 명확히 이해한 후에는 전역 상태와 컴포넌트별 상태를 적절히 분리하고 최적화할 수 있게 되었다.

그 결과, 전역 상태를 꼭 사용해야 하는 경우와 로컬 상태로 관리하는 것이 적절한 경우를 구분하며 개발하는 습관을 가지게 되었다. 이 경험을 바탕으로 이후 프로젝트에서는 렌더링 최적화를 신경 쓰고, 상태를 체계적으로 관리하는 방식을 고민하며 개발할 수 있었다.