

자기주도 PJT

**Create Next App + React + Typescript + next.js 로
프론트엔드 프로젝트 토대 쌓기**

목차

1. 과제 개요.....	3
2. 과제 목표.....	5
3. 필수 지식 학습.....	6
4. 기능 명세.....	7
5. 산출물 제출.....	10
6. 추천 학습 방법.....	오류! 책갈피가 정의되어 있지 않습니다.

1. 과제 개요

왜 React 인가?

React 는 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리입니다. 국내에서는 이미 Angular 나 Vue.js 보다 시장 지배적인 위치를 차지했고 대부분의 개발 회사에서는 프론트엔드 개발을 위해서 React 를 선택하는 분위기이기 때문에 취업을 준비하는 입장에서는 React 를 하지 않을 이유가 없습니다. React 는 입문자에게는 학습의 난이도가 있는 편이지만 공식 문서를 학습하면서 천천히 따라간다면 프로젝트에 사용할 만큼 이해하는 것이 충분히 가능합니다.

CSR 과 SSR

SPA(Single Page Application)이 프론트엔드 개발의 대세가 되면서 CSR(Client Side Rendering)을 활용한 서비스들이 등장하기 시작했는데 여러 가지 장점들로 인해서 기존의 SSR 을 완전하게 대체할 것처럼 보였습니다. 하지만 많은 회사들이 실제로 서비스를 운영해보니 CSR 의 치명적인 문제를 알게 되었는데, 바로 대부분의 검색 엔진들이 CSR 로 개발된 페이지를 제대로 인식하지 못한다는 것이었습니다. 회사들의 입장에서는 서비스를 홍보하기 위해서 막대한 마케팅 비용을 지출하게 되는데, 구글이나 네이버 등의 검색엔진을 통한 노출은 비용 지출 없이도 자연적으로 유저의 유입이 발생하는 수단이었습니다. 하지만 CSR 로 개발한 페이지들은 검색 엔진이 분석할 수 없었기 때문에 next.js 와 같은 대체적인 방법을 활용하게 되었습니다. 최근 검색 엔진의 성능이 조금씩 좋아져서 CSR 로 구현된 페이지도 조금씩 분석이 가능해지고 있지만 여전히 최상위 페이지에 노출되기에는 부족한 수준입니다.

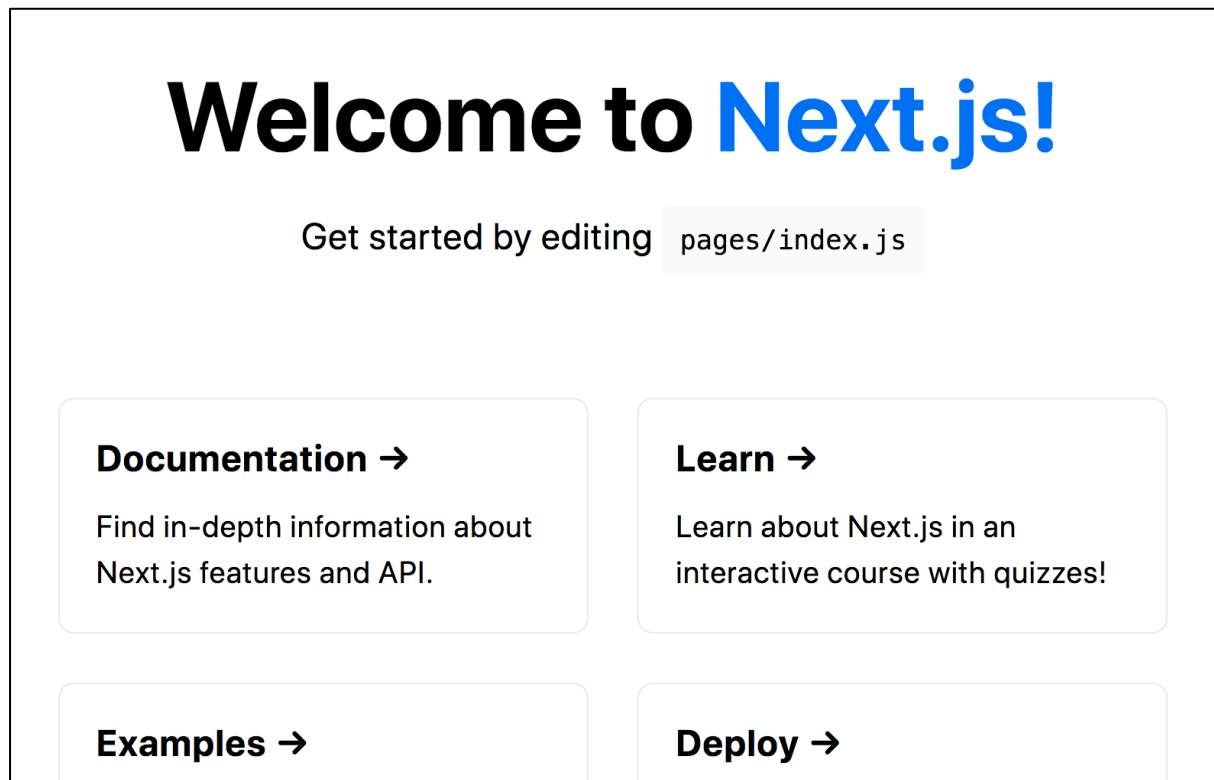
TypeScript

TypeScript 는 JavaScript 를 기반으로 정적 타입 문법을 추가한 프로그래밍 언어입니다. JavaScript 는 동적 타입의 인터프리터 언어로 런타임에서야 오류를

발견할 수 있었던 반면, TypeScript 는 정적 타입의 컴파일 언어이며 컴파일러를 통해서 JavaScript 코드로 변환됩니다. TypeScript 는 코드 작성 단계에서 타입을 체크해서 오류를 확인할 수 있기 때문에 실행하기 전에 많은 오류들을 잡아낼 수 있다는 장점이 있습니다. JavaScript 로 개발하는 대부분의 회사에서 TypeScript 를 사용하고 있기 때문에 프론트엔드로 취업을 준비하는 개발자라면 반드시 익혀야 하는 기술 스택입니다.

React + next.js + TypeScript

이 명세서의 목표는 Create a Next.js App 을 이용해서 아래의 화면을 보는 것입니다. 단, 설정 과정에서 TypeScript 가 포함되어야 한다는 것을 기억하세요. 이후에 시간이 충분하다면 심화 과정으로 1 학기 때 진행했던 관통 프로젝트를 이 기술 스택을 기반으로 리팩토링 해보세요.



2. 과제 목표

- 1) 공식 문서를 보면서 React 의 기본 개념과 철학을 익힌다
- 2) next.js 를 공부하면서 SSR(Server Side Rendering)의 중요성을 익힌다
- 3) TypeScript 를 사용하면서 프론트엔드의 정적 타입 문법을 사용해본다
- 4) 개발에 필요한 환경을 구축해보고 기초적인 셸 사용법을 익힌다

3. 필수 지식 학습

오늘 학습할 내용들은 기초적인 내용들이지만, 다루고 있는 기술 스택들을 완벽하게 이해할 때까지는 많은 시간이 필요합니다. 너무 조급하게 생각하지 말고 공식 문서의 튜토리얼과 instruction 들을 하나씩 따라 해보세요. 문서를 읽으면서 라이브러리의 구성 요소들을 배워야 하고, 실제로 프로젝트를 구현해보면서 손으로 익혀야 하는 것들도 많습니다. 앞으로 진행하게 될 3 개의 프로젝트를 위한 토대라고 생각하고 천천히 따라 해봅시다.

참고 자료

구분	제목	링크
이해	React 공식 문서	https://ko.reactjs.org/
이해	Next.js 공식 문서	https://nextjs.org/
이해	TypeScript 공식 문서	https://www.typescriptlang.org/

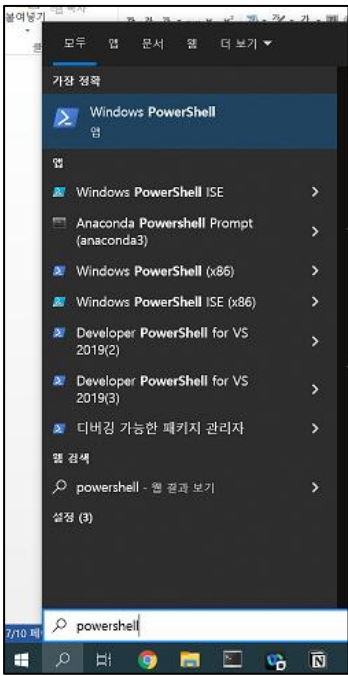
4. 기능 명세

1. 기능/과제 목록

Req.	Category
1	환경 구축
2	Node.js 설치
3	create-next-app 을 이용한 Next.js 설치
4	웹서버 실행 후 결과 확인
5	프로젝트 폴더 분석과 오류 해결

2. 기능/과제 상세

Req. 1. 환경 구축

Req. 1-1	환경 구축
기능 상세	<p>셸을 사용할 수 있는 환경을 구축합니다. AWS 무료 인스턴스를 활용해서 우분투 리눅스 환경을 구축하거나, 윈도우 PC 에 git bash 혹은 파워셸을 설치합니다. 이미 구축이 되어 있다면 다음 단계로 넘어갑니다. 윈도우 사용자라면 이미 파워셸이 설치되어 있는 경우가 많을 것입니다.</p> 

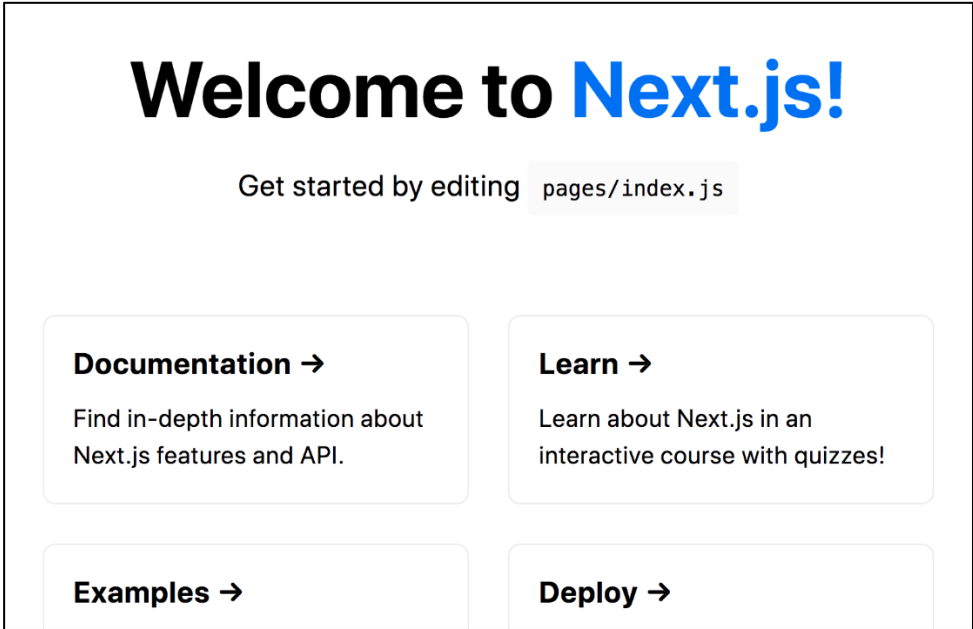
Req. 2. Node.js 설치

Req. 2-1	Node.js 설치
기능 상세	node.js 를 설치합니다. node.js 의 경우에는 LTS 버전을 설치하면 됩니다. 이미 node.js 가 설치되어 있는 경우 이 단계를 건너뛰니다. 설치가 완료되면 <code>node --version</code> 명령어를 통해서 설치된 버전을 확인합니다.

Req. 3. create-next-app 을 이용한 Next.js 설치

Req. 3-1	create-next-app 을 이용한 Next.js 설치
기능 상세	<p>create-next-app 을 활용해서 프로젝트를 시작합니다. 프로젝트 폴더명을 지정합니다.</p> <pre>npx create-next-app@latest --ts # or yarn create next-app --typescript</pre>

Req. 4. 웹서버 실행 후 결과 확인

Req. 4-1	웹서버 실행 후 결과 확인
기능 상세	<p>프로젝트의 루트 폴더로 이동해서 웹서버를 실행하고 브라우저에서 정상적으로 동작하는지 확인합니다.</p> <pre>npm run dev</pre> 

Req. 5. 프로젝트 폴더 분석과 오류 해결

Req. 5-1	프로젝트 폴더 분석과 오류 해결
기능 상세	<p>IDE 를 활용해서 프로젝트 폴더를 열어봅니다. 운이 좋다면 아무런 오류 없이 동작할 수도 있겠지만, 많은 경우에는 오류에 직면하게 될 것입니다. IDE 에서 프로젝트를 편집하고 실행할 수 있도록 프로젝트 설정을 변경합니다. 굉장히 심플한 프로젝트의 폴더 구조를 살펴봅니다.</p> <pre> TypeError: this.cliEngineCtor is not a constructor at ESLintPlugin.invokeESLint (C:\Program Files\JetBrains\IntelliJ IDEA\plugins\eslint\lib\eslint-plugin.js:10:1) at ESLintPlugin.getErrors (C:\Program Files\JetBrains\IntelliJ IDEA\plugins\eslint\lib\eslint-plugin.js:20:1) at ESLintPlugin.onMessage (C:\Program Files\JetBrains\IntelliJ IDEA\plugins\eslint\lib\eslint-plugin.js:30:1) at Interface.<anonymous> (C:\Program Files\JetBrains\IntelliJ IDEA\plugins\eslint\lib\eslint-plugin.js:40:1) at Interface.emit (events.js:375:28) at Interface._onLine (readline.js:364:10) at Interface._normalWrite (readline.js:509:12) at Socket.ondata (readline.js:216:10) at Socket.emit (events.js:375:28) </pre> <p>이런 오류들은 흔하게 자주 볼 수 있습니다.</p>

5. 산출물 제출

- <https://lab.ssafy.com/s10-study/self-project/> 의 "산출물 제출 가이드".docx 참조
- 작성 내용

산출물 제출

생성한 프로젝트를 gitlab 에 제출한다. README.md 에 실행한 프로젝트의 스크린샷을 포함한다.

6. 심화 학습

이제 next.js의 바다로 나아갈 준비는 다 되었습니다. 공식 문서의 튜토리얼과 예제들을 따라하면서 하나하나 필요한 것들을 익혀보세요. 어디서부터 시작해야 할 지 막막하다면 Router(<https://nextjs.org/docs/api-reference/next/router>)부터 해보는 건 어떨까요?