**고급 웹 프로그래밍 중간고사 과제**

2021113490 이현서

**목차**

1. 과제 목표
2. 개발 환경
3. 개발 내용 및 구조
   1. Root Directory
   2. Src Directory
   3. Components Directory
      1. Introduce Page
      2. Experience Page
      3. MyProduceCard Page
      4. Footer Page
      5. ScrollTopButton Page
4. 프로젝트 배포
   1. 배포 사이트 주소
5. 프로젝트 진행 후 느낀 점

**과제 목표**

이번에 제가 구현할 MVP(Minimum Viable Product)는 저를 소개하는 자기소개 포트폴리오 입니다. 과제를 이로 선정한 이유는 앞으로 개발자의 길을 걸으면서, 나를 소개하고 같이 협업할 수 있는 개발자에게 나를 어필 할 수 있는 자료를 만들고 싶었기 때문입니다. 더불어, 기초적인 디자인 실력을 늘리고자 했기 때문입니다.

포트폴리오의 세부적인 내용은 구현하지 않고 디자인과, 기본적인 애니메이션, 문구들을 mock up하기로 했습니다. 그 이유는 시간적인 문제가 가장 컸습니다. 저를 소개하는 문구, 중요한 틀만 잡아 놓고, 세부적인 내용은 구현하는데 시간적인 한계가 존재했습니다.

추후에 시간적인 여유가 있는 때에, 다양한 오픈 소스를 활용해서 세부기능을 추가할 예정입니다. 또한 백 엔드와 연계해서 사람들이 소통할 수 있는 블로그와 같은 소통의 장을 직접 만들어서 포트폴리오에 링킹하고 싶습니다.

**개발 환경**

고급 웹 프로그래밍에서 배운 것을 토대로 포트폴리오를 구성하려고 노력하였습니다. 우선 Visual Studio Code을 활용하여 개발하였습니다. 그리고 Next.js로 전체적인 틀을 짰지만, Server-Side-Rendering은 사용하지 않았습니다.

**개발 내용 및 구조**

**< root Directory >**

우선 Next.js는 React의 Framework입니다. 따라서 Component단위로 포트폴리오의 구조를 서술하겠습니다. root directory를 아래와 같이 구성했습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

핵심적인 디렉토리만 설명하겠습니다.

1. node\_modules >> dependency file
2. public >> static file serving
3. src >> page component && next.js api module
4. .babelrc >> styled-component transpile
5. .eslintrc.json >> syntax error highlighting
6. Jsconfig.json >> for absolute path serving and autocomplete in project
7. next.config.js >> gsap application

중요한 pages들의 component들은 src/pages안에 있습니다. 핵심 구조를 파악하기 위해 src폴더의 구조를 뜯어보겠습니다.

**< src Directory >**

텍스트이(가) 표시된 사진

자동 생성된 설명

src의 폴더 안에는 page의 component를 관리하는 components폴더, page rendering에 필요한 \_app.js와 index.js가 있는 pages폴더, globalStyle( CSS-IN-JS에서 전역 스타일링을 하기 위해서는 이렇게 해야함 ), responsive한 웹을 위해 변수들을 저장해 놓은 theme파일이 styles폴더 안에 존재합니다. 자주 쓰는 색들을 colors에 저장해 놓았고, 변수들을 constant에 저장했습니다.

**< components Directory >**

이제 배포된 포트폴리오를 보면서 디자인과 구조에 대해 구체적으로 살펴보겠습니다. 페이지 렌더링에 필요한 src/pages/index.js를 보겠습니다. <Section>이라는 box안에 <Introduce />, <Experience />, <Footer />, <ScrollTopButton />이 있습니다. 이름에서 각각의 역할을 유추할 수 있습니다.

**Components - Introduce page**

텍스트, 사람이(가) 표시된 사진

자동 생성된 설명

저를 소개하는 **Introduce Page** 입니다. Introduce컴포넌트는 src/components/Introduce안에 있습니다. 저는 모든 Component폴더 안에 index.jsx, styled.jsx를 넣어서 스타일링 하는 것을 선호합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이 안의 index.jsx코드를 분석하는 것은 내용상의 한계가 존재하므로 핵심적인 내용만 보겠습니다.

위의 Introduce Page도 다시 SideBar, Main 컴포넌트로 나뉩니다. 이는 CSS-In-JS중 하나인 **Styled-component**로 스타일링을 하였습니다. Styled-component스타일링의 대부분은 styled.jsx에서 진행되었습니다.

또한 Introduce Page에서 다양한 애니메이션 효과도 적용했습니다. 컴포넌트가 렌더링 되었을 때, 밑줄도 그어지고 시간 차를 두고 하나씩 렌더링 되게 했습니다. 이는 gsap framework를 통해 구현하였습니다. 아래와 같이 src/components/gsap에서 관리하였으며, 우리가 애니메이션을 적용하고 싶은 컴포넌트에 HOC로 감싸주면 적용되는 간단한 원리입니다. 애니메이션 효과는 시연에서 보실 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

gsap컴포넌트에는 FadeInBodyText부터 FadeInContent, FadeInHeader, index, Underline (.jsx)파일들이 있는데 index.js는 하나를 묶어서 export하기 위한 파일입니다. 각각의 파일들을 리펙토링하면 하나로 줄일 수 있겠지만, 시간상의 문제로 여기까지는 하지 못했습니다. Introduce Page의 Components들을 렌더링할 때 애니메이션을 적용 시켜 주기 위한 gsap애니메이션이 적용된 HOC컴포넌트들이라고 할 수 있겠습니다.

텍스트, 사람이(가) 표시된 사진

자동 생성된 설명

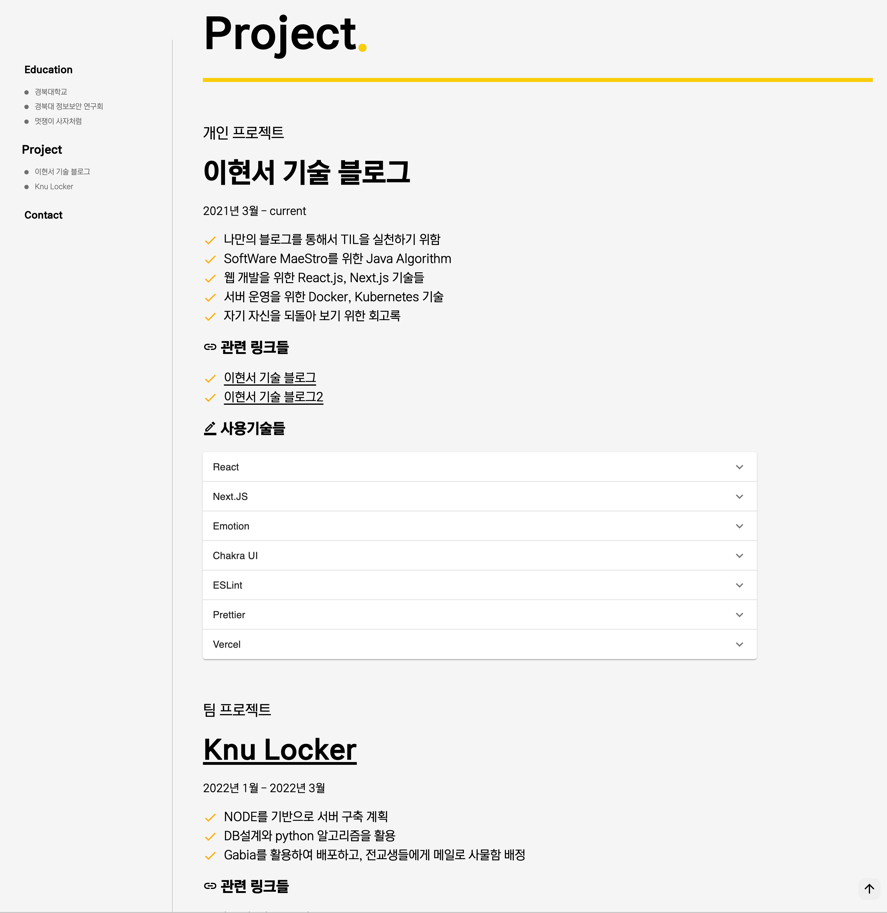
또한 Responsive Web Design을 구현하기 위해서 view port의 크기에 따라 미디어 쿼리를 사용해서 글자의 크기와 flex-body의 direction을 조절하 화면의 layout을 변경해 주었습니다. 이는 mobile에서도 적절히 작동함을 확인했습니다.

**Components - Experience page**

Introduce Page 다음으로는 저를 소개하고, 저의 경험을 서술하는 페이지가 위의 Introduce페이지에서 Scroll Down하면 나타나게 됩니다.

텍스트이(가) 표시된 사진

자동 생성된 설명



텍스트이(가) 표시된 사진

자동 생성된 설명

Experience Page의 큰 틀로는 Navbar가 있고, 저를 구체적으로 서술하는 Component가 옆에 존재하게 됩니다. 또한, 저를 구체적으로 소개하는 페이지는 크게 Education, Projects Part로 나뉩니다. 이를 스타일링하기 위해 저는 아래와 같이 src/components/Experience폴더를 작성했습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 Experience안에 Components에서 쓰일 공통 layout은 common폴더 안에, education, projects컴포넌트는 contents안에 있는 edcations, projects폴더에 작성했습니다. Educations의 내용으로는 제 학교와 관련된 내용인 **Kert, Likelion, University**와 관련된 내용을 mock up하였으며, Projects의 내용으로는 제가 학교에 들어와서 했던 프로젝트인 **KnuLocker, Tec-Blog**를 간단히 mock up하였습니다.

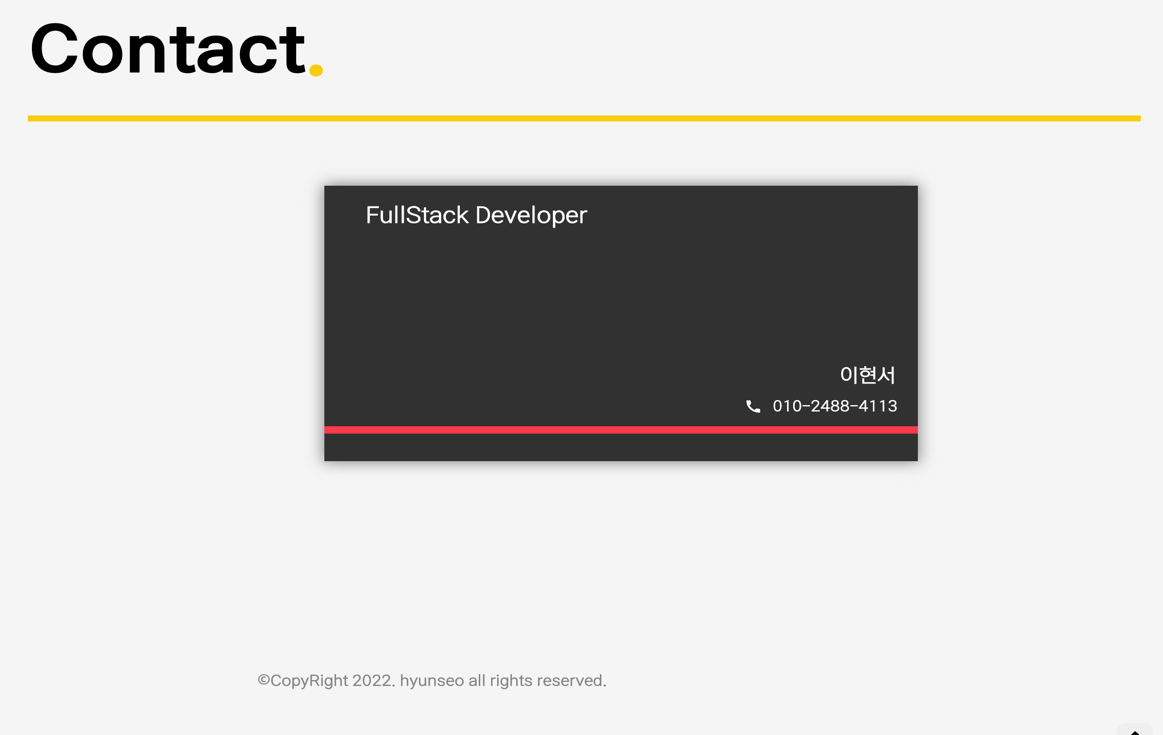
Experience옆에 있는 Navbar는 **Intersection Observer API**를 사용해서 지금 화면이 존재하는 곳을 계산하여 지금 내가 있는 곳에 특별한 효과를 부여되게끔 하였습니다. 추가로 Navbar에서 각각의 콘텐츠를 클릭하면 **smooth-scroll browser api**로 해당 Experience content로 이동하게끔 개발했습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

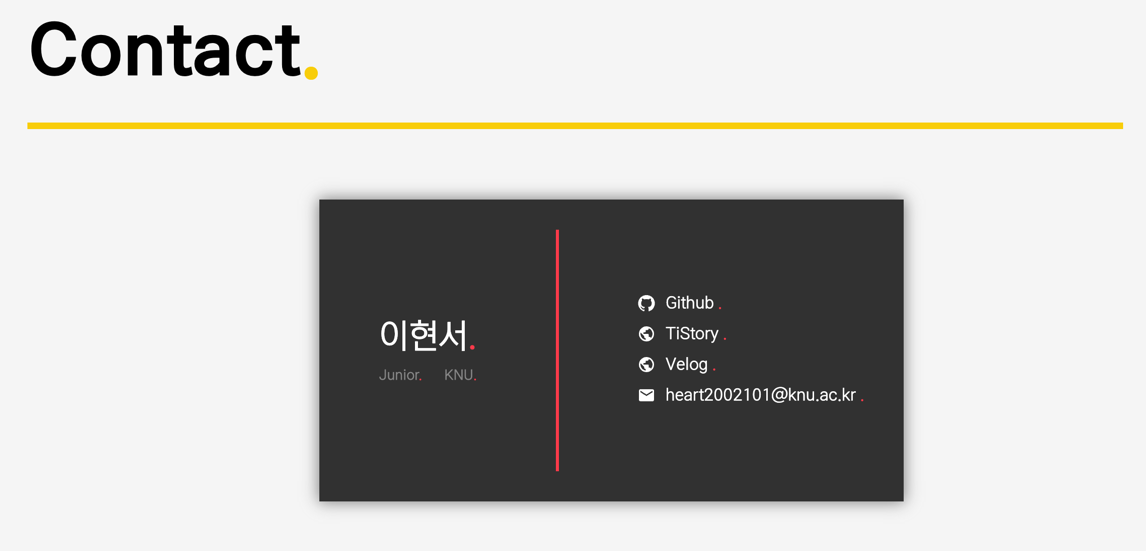
Navbar컴포넌트는 src/components/TableOfContents폴더 안에 스타일링 하였습니다.

**Components - MyProduceCard page**

****

텍스트, 스크린샷, 명함이(가) 표시된 사진

자동 생성된 설명



Contact Card는 제 포트폴리오의 디자인이 심심한 것 같아서 재밌는 포인터를 주고자 명함 형식으로 저의 정보를 표현한 것입니다.

마우스를 카드에 hover하면 카드가 3D형태로 기울게 되고, 카드를 클릭하면 카드가 horizontal 방향으로 돌아가게 되어, 다른 정보를 카드에 보여주게끔 했습니다.

이를 위해서 **react-card-flip**을 사용하였으며, **react-parallax-tilt**를 사용하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

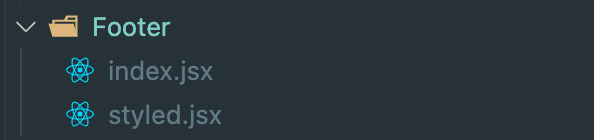
myProduceCard컴포넌트는 src/components/MyProduceCard폴더 안에 스타일링 했습니다.

**Components – Footer page**

텍스트이(가) 표시된 사진

자동 생성된 설명

이는 모든 페이지에 존재하는 Footer페이지 입니다. 그냥 간단한 저작권 표시를 하였습니다.



footer는 src/components/Footer폴더에 만들어서 스타일링 하였습니다.

**Components – ScrollTopButton page**

텍스트이(가) 표시된 사진

자동 생성된 설명

모든 페이지의 위에 부유하게 position속성을 absolute로 해서 맨 위로 페이지를 smooth하게 올려주는 버튼을 만들었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

ScrollTopButton은 src/components/ScrollTopButton을 만들어서 스타일링 하였습니다.

**프로젝트 배포**

<https://github.com/LE123123/MyResumeProject>

우선 제 GitHub에 먼저 commit하는 과정을 거쳤습니다.

기존 사이트들은 서버에서 직접 구동해서 전세계에 뿌려줬었습니다. 이렇게 되면 서버의 CPU, RAM, SSD등등 구입 및 관리해야 할 것이 매우 많고, 주기적으로 업데이트도 해줘야 했습니다. 그런데, GitHub의 저장소에 있는 파일들을 받아서, 스스로 빌드한 뒤 전세계 CDN에 무료로 뿌려주는 서비스가 등장했기 때문입니다.

알려진 서비스로는 netlify가 있습니다. 하지만 CDN서버가 한국에 있지 않아 느릴 수 있다는 점입니다. 그래서 Vercel을 통해서 이번 프로젝트를 빌드하고, 배포하였습니다. 돈도 전혀 받지 않고, 트래픽 비용, 데이터 저장 비용도 받지 않기 때문입니다. 또한 서버가 한국, 캐나다, 미국에 특화되어 있어 netlify보다 빠릅니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 Vercel에서 배포하기 위해서 노드 환경설정을 통해 build명령어를 작성해 주었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그리고 Vercel에서 GitHub 최종 commit을 delpoy해 주어 배포를 완료했습니다. 그리고 custom Domain 설정을 통해서 제 사이트 이름을 hyunseo-resume.vercel.app으로 설정해 주었습니다.

**< 배포 도메인 >**

<https://hyunseo-resume.vercel.app/>

**프로젝트 진행 후 느낌 점**

고급 웹 프로그래밍 수업에서 기초적인 html, CSS, JS과 관련된 지식을 배웠습니다. 하지만 웹 개발을 한지 반년이 지났음에도, 기초적인 웹 지식이 많이 부족함을 느낄 수 있었습니다. 저는 이번 프로젝트를 하기에 앞서 기초적인 지식의 기반을 다지고자 하는 목표가 컸습니다.

이를 잘 적용시킬 수 있는 Frontend Framework를 선정하고 스타일링을 하고, 나만의 개발 스타일을 수립해 갔습니다. 아직 리팩토링 부분과 Opensource 읽는 능력이 많이 부족함을 느낄 수 있었습니다. 그리고 제가 세웠던 목표인 기초적인 지식을 전반적으로 다 다루어 스타일링 하며, 실력이 다져진 것 같아 매우 뿌듯합니다.

일주일 동안 안되는 수정하고 git에 push하는 과정을 거치니 진정한 개발자의 고뇌에 대해 느낄 수 있었습니다. 또한, 여기서 성장을 멈추는 것이 아니라, 다른 CS 지식을 많이 습득해서 추후에 이를 발전시켜 더욱 완성도 있는 포트폴리오를 만들고 싶다고 느꼈습니다.