

12팀

이상한 나라의 개발자

- 프론트엔드 : 김채정 이정민
- 백엔드 : 엄혜진 배주영 김현서

01.DEMO

<http://elice-kdt-ai-4th-team12.elicecoding.com>

02. 기능 추가 및 개선

기능 개선

학력, 수상이력, 자격증, 프로젝트 삭제

비밀번호 변경

유저 이메일 수정 불가

프론트: 편집 시 ReadOnly props를 설정
백: 필드의 immutable 속성을 true로 설정

사용자 탈퇴

네트워크에서 본인 프로필 제외

[MongoDB 쿼리]- \$eq로 본인 프로필과 불일치한 것만 추출해서 반환

버튼 UI 개선

[react-icons] - 버튼에서 텍스트를 아이콘으로 대체

기능 추가

비밀번호 초기화

[nodemailer]

사용자의 이름과 이메일이 DB에서 일치하는 경우 임시 비밀번호 생성

유저 프로필 사진 설정

프론트: [FileReader] - 업로드한 이미지 파일 저장 및 미리보기

백: [AWS Bucket] - 사진 파일을 저장하고, DB에는 이미지 링크를 저장
이모지 링크(👤)를 default 값으로 설정해 사진을 업로드하지 않는 경우 처리

네트워크 무한스크롤

[React.forwardRef] - 엘리먼트의 위치 확인

[IntersectionObserver API]

엘리먼트의 위치와 뷰포트의 겹치는 정도를 감지하여,
최하단 엘리먼트가 뷰포트에 보이는 순간 다음 페이지 데이터를 요청

아카이빙 링크

input에서 url 파싱을 이용해 'https://' 또는 'http://' 유무를 검사

Refresh Token

refresh token을 쿠키에 저장했다가 access token이 유효 기간이 만료된 경우,
재발급받아 다시 로그인할 필요 없이 서비스 이용 가능

다크 모드

[Styled-Component]

create global style을 도입해 전역으로 적용하는 css 사용

기능 추가

프로필 좋아요

프론트:

클릭 이벤트에 좋아요 누른 유저 id 목록(likes)을 업데이트 하는 함수 연결
useEffect를 통해 변경된 likes에 따른 좋아요 수, 좋아요 여부 state 값을 변경

백: [MongoDB의 \$pull, \$push]

req.userId가 해당 User Collection에 있다면 제거하고,
없다면 추가하는 방식으로 좋아요 데이터 관리

소셜 로그인

구글: [React>react-google-login]
[Node.js>google-auth-library]

깃허브: [GitHub OAuth App]

이미 해당 이메일로 가입된 데이터가 있다면 해당 계정으로 자동 로그인,
데이터가 없다면 GitHub 유저 정보를 불러와 새 계정 생성

메시지 알림창

회원 가입 시/로그인 시/인증 완료/비밀번호 변경 완료 시/회원 탈퇴 완료 시 :

[react-alert] - success alert을 적용

삭제 시도/탈퇴 시도 시:

[react-confirm-alert] - 삭제 확인/탈퇴 확인 알림창 적용

로그인 실패/회원 가입 실패 시:

[react-alert] - error alert을 적용

내용 변경 시 알림

백: [MongoDB .watch() 함수] - Project Collection 변경 시
변경 내용을 Alert Collection에 기록. 프론트엔드는 구현되어있지 않음.

유효한 이메일 인증

03. 주요 이슈

01

현서

깃허브 로그인 + CORS, DB 저장 오류 (*CORS: Cross-Origin Resource Sharing)

1. 가입한 계정이 있는 경우, 해당 계정으로 자동 로그인,
계정이 없는 경우, GitHub 유저 정보를 바탕으로 새로운 계정 생성
2. 처음 Oauth 설정을 백엔드 서버로 하다보니, CORS 에러 발생
3. Oauth 설정을 프론트엔드 서버로 바꾼 후, 프론트엔드에서 백엔드 서버로 요청을 하고,
그 백엔드 서버 안에서 GitHub API 서버로 요청을 해서 정보를 처리하는 방식으로 바뀌서 해결 가능
4. GitHub에서 비공개 설정 등에 의해 name을 받아올 수 없는 경우가 생겼는데,
계정 생성 시 필수값인 name이 존재하지 않아 로그인이 정상적으로 진행되지 않는 오류 발생
5. name이 없는 경우, GitHub의 닉네임으로 대체하는 방법으로 문제 해결

02

주영

Refresh Token

1. access token이 만료된 경우,
2. 모든 request에서 오류가 날 수 있기 때문에
 axios.interceptor를 사용해서 전체에 적용하고
 자동으로 로그인이 연장되도록 설정
3. 쿠키에 저장된 refresh token의 보안을 위해
 httpOnly 속성을 적용
4. 로그아웃 시 백엔드에 요청을 보내 refresh token 삭제

03

혜진

무한스크롤

1. 커스텀 컴포넌트를 ref할 수 없기 때문에
 React.forwardRef를 이용해서 커스텀 컴포넌
 트를 ref 하게 함
2. React.forwardRef를 이용해서 엘리먼트의
 위치를 확인하고 IntersectionObserver API 를
 이용해서 그 엘리먼트의 위치와 뷰포트의 겹치는
 정도를 감지하여, 가장 아래의 엘리먼트가
 뷰포트에 보이는 순간 다음 페이지 데이터를
 요청함.

04. 소감

채정

“프로젝트를 하면서 최선을 다했던 것 같은데, 배포하니깐 아쉬운 부분들이 너무 많이 보입니다.”

현서

“실제 개발자가 일하는 것처럼 팀원과 함께 웹 애플리케이션을 만드는 과정에서 많은 것을 배웠습니다. 후반부에 깃허브 로그인에서 막혀 부족한 부분들을 스스로 채우지 못한 것 같아서 아쉽고, 프론트엔드를 더 알았다면 다른 팀원분들과의 소통이 더 편했을 것 같은데, 그런 지점들에서 나의 부족함을 점점 채워나가야겠다는 생각이 들었습니다.”

주영

“배포하고 나서 부족한 게 많이 보여서 왜 프로젝트 배포가 시작이라고 하는지 알겠어요. 그리고 각종 문제가 생겼을 때 팀원 분들이 잘 도와주시고 해결해주셔서 감사했습니다.”

정민

“열정이 넘치는 팀원분들과 함께 해서 배울 점이 많았고, 실제 협업을 하며 깃을 써보면서 깃과 친해지는 시간이 되었습니다.”

혜진

“많이 부족하다는 것을 스스로 깨달은 프로젝트였던 것 같습니다. 나중에 이 결과물보고 부딪힘 보다는 ‘귀엽다’라는 생각이 들었으면 좋겠습니다.”

05. 배운 점 & Tip!

10시 스크럼

하루를 일찍 시작해
더욱 생산적인 협업 가능

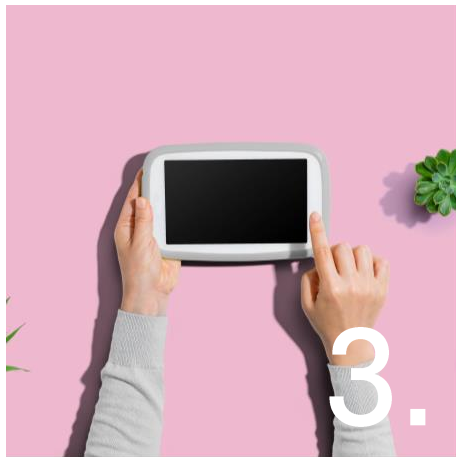



협업 툴 세팅

Notion
Postman API
GitLab


소통의 중요성

GitLab 이슈, 디스코드 채팅 등
협업에서 소통을 꾸준히 하게




 김채정


#프론트엔드 #이슈대장

 이정민


#프론트엔드 #좋아요장인

 김현서

#백엔드 #스피드레이서

 배주영

#백엔드 #토큰마스터

 엄혜진

#백엔드 #배포도사

THANK
YOU

12팀 이상한 나라의 개발자 🐼