

1. 프로젝트 목적

- Node.js 설치/ Socket.io 설치 / MongoDB 설치 해보기
- Node.js + Socket.io + MongoDB 를 이용한 "웹 어플리케이션" 만들기
- React, Bootstrap 이용해보기

2. 요구 기능

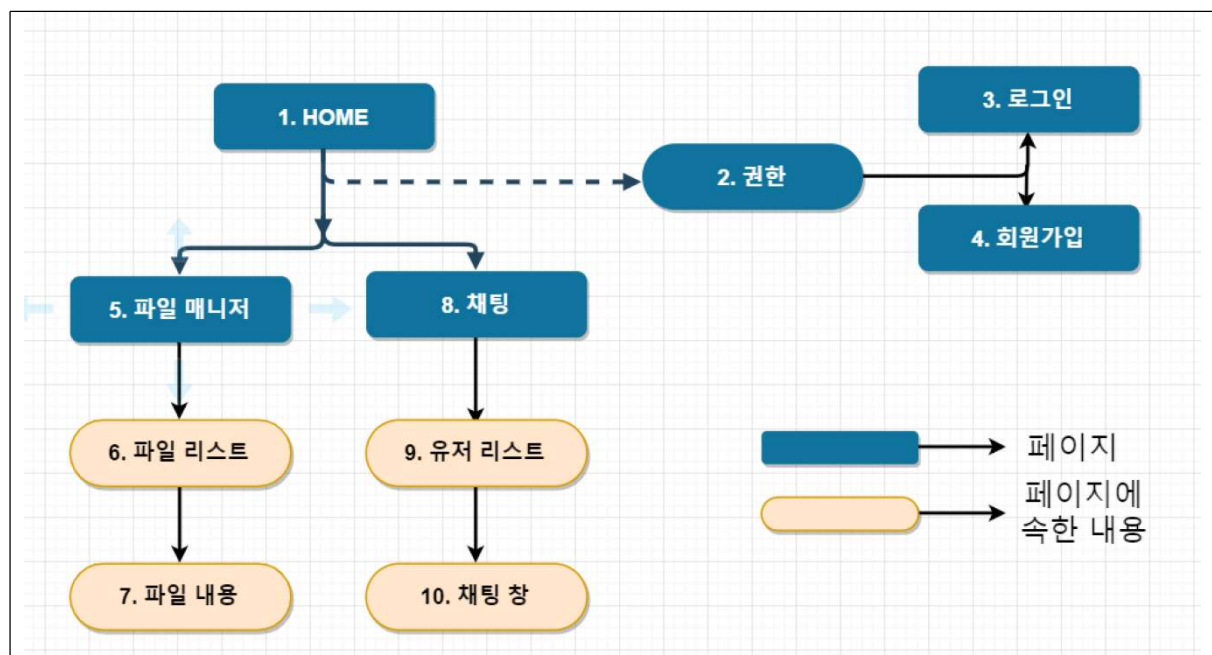
- 회원가입/ 로그인 / 로그아웃
 - ID/ PW 방식으로 구현
 - 세션을 생성하여 로그인 유지
 - 로그아웃을 하면 세션 제거
- 파일 매니저 기능
 - 파일/폴더 구조를 가진 프로젝트를 업로드 합니다.
 - zip, tar 를 지원
 - 업로드 된 프로젝트를 풀어 ul 과 li 를 이용해 리스트를 만듭니다.
 - 디렉토리 구조는 [폴더 1]/[폴더 2]/파일과 같이 하여 리스트업
 - 해당 디렉토리 밑에 있는 파일에 대해 [읽기/쓰기] 가능하도록 합니다.
 - 읽기 : 파일명을 클릭하면 textarea 에 파일 내용이 보입니다.
 - 쓰기 : textarea 에서 내용을 편집하고 저장버튼을 눌러 저장
- 채팅 기능
 - 채팅을 할 수 있습니다.
 - 소켓을 이용하여 실시간으로 메시지를 주고 받습니다.
 - 새로고침이나 재접속을 하여도 채팅 내역은 남습니다.
 - 귓속말을 통해 1:1 메시지 교환도 가능 합니다.
 - 귓속말 구현시 모든 상대방에게 보내면 안됩니다. 반드시 <받는사람>만 내용을 받아야합니다.

3. 프로젝트 주요 스펙

- Front-end : React + Bootstrap + Reactstrap + React-router
- Back-end : Node.js + Socket.io + MongoDB + Redis

[Front-end]

4. 페이지 구성도



(1) Home : 첫 페이지

(2) 권한 : 중복로그인 방지 기능

- 동일한 계정으로 로그인 시 동일 계정 접속 Alert 출력 및
(접속한 계정, 접속해있던 계정)로그아웃 처리
- 세션 만료시 로그아웃 처리

(3) 로그인 : 페이지에 로그인 할 수 있는 페이지

- ID/PW 모두 입력하지 않았을 경우 : 모두 입력해주세요 Alert 출력
- ID 틀렸을 경우 : 없는 아이디입니다. Alert 출력
- PW 틀렸을 경우 : 비밀번호가 틀립니다. Alert 출력
- ID/PW 맞았을 경우 : 로그인 처리

(4) 회원가입 : 회원가입을 할 수 있는 페이지

- 모두 입력하지 않았을 경우 : 모두 입력해주세요 Alert 출력
- 아이디 혹은 이메일이 중복된 경우 : 중복된 아이디 혹은 이메일입니다. Alert 출력
- PW 와 PW 확인이 다를 경우 : PW 가 다릅니다. Alert 출력.

(5) 파일 매니저 : 파일을 관리할 수 있는 페이지

(6.)파일 리스트 : 파일을 업로드하고 ul, li 로 출력하는 컴포넌트

- tar.. zip 파일만 선택 할 수 있는 기능.
- 압축 폴더 이름 중복 검사 후 업로드 여부 파악
- 페이지 업로드 상황을 Progress Bar 로 출력
- ul, li 를 이용하여 디렉토리구조 출력
- 폴더를 클릭했을 경우 하위 디렉토리 출력
- 파일을 클릭했을 경우 해당 파일 내용을 불러옴.

(7) 파일 내용 : 파일을 수정할 수 있는 컴포넌트

- 파일리스트(5)에서 클릭한 파일에 내용을 출력
- 파일에 내용을 수정 후 저장버튼 클릭시 해당 내용 저장

(8) 채팅 : 실시간으로 접속한 유저들과 채팅할 수 있는 페이지

(9) 유저 리스트 : 접속한 유저를 출력하는 컴포넌트

- 접속한 유저 출력
- 리스트 클릭시 해당 사용자와 대화(귓속말) 가능

(10) 채팅 창 : 전체 채팅 및 귓속말 채팅을 보낼 수 있는 컴포넌트

- 유저리스트에서 [전체] 클릭시 전체 채팅 및 이전 대화내용 출력
- 유저리스트에서 [ID] 클릭시 귓속말 채팅 및 이전 대화내용 출력

[추가기능] : 메시지 알림 기능

예) A 사용자가 접속한 B 사용자에게 귓속말 채팅 시 B 사용자가 채팅 페이지에 없다면 B 사용자 웹페이지에 메시지 알림 출력

[Back-end]

5. 기능 구현

회원가입, 로그인, 로그아웃 기능
<u>사용한 모듈</u> <ul style="list-style-type: none">- passport, passport-local : 회원가입, 로그인,, 로그아웃 세션 관리를 쉽게 해주기 위한 모듈- bcrypt-nodejs : 비밀번호를 암호화 복호화 하기위한 모듈- express-session : passport 모듈과 같이 사용하여 세션을 저장하기위한 모듈- connect-redis : 서버가 꺼지면 없어지는 express-session 과 같이 사용하여 세션을 저장하기 위한 모듈
<u>회원가입 순서(성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> ID/Email 존재 검사 -> PW 암호화 -> DB 저장
<u>로그인 순서 (성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> ID 검사 -> PW 복호화 및 검사 -> DB 최근 접속 시간 수정 -> 세션 저장 및 로그인 처리

파일 매니저 기능
<u>사용한 모듈</u> <ul style="list-style-type: none">- multer : 파일을 업로드하기위해 사용하는 모듈- tar-fs, unzip : tar, zip 파일을 각각 압축해제하기위해 사용하는 모듈- directory-tree : 해당 경로에 속한 모든 서브디렉토리와 파일들의 데이터를 tree 구조로 구조화 하기위해 사용하는 모듈
<u>중복 파일 검사 (성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> 로그인 유무 검사 -> 압축 파일 이름과 DB 에서 유저가 가지고있는 파일 검사 -> 존재한다면 해당 파일 id 반환 존재하지않다면 null 반환
<u>업로드 순서 (성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> 로그인 유무 검사 -> zip, tar 확장자 검사 -> 압축해제 -> path 값 DB 저장 -> fileData DB 저장
<u>파일 읽기 순서 (성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> 로그인 유무 검사 -> 파일 읽기
<u>파일 수정 순서 (성공한 경우만)</u> <ul style="list-style-type: none">- 요청 -> 로그인 유무 검사 -> 파일 수정

채팅 기능
<p><u>사용한 모듈</u></p> <ul style="list-style-type: none"> - socket.io : 실시간 양방향 통신을 하기위해 사용하는 모듈
<p><u>참여한 유저리스트 가져오기</u></p> <ul style="list-style-type: none"> - 참여한 소켓 아이디와 DB 에 저장되어있는 참여 유저 DB 와 비교하여 데이터 리턴
<p><u>전체 채팅 가져오기</u></p> <ul style="list-style-type: none"> - 전체 채팅 내역이 저장되어있는 DB 에서 값을 리턴
<p><u>전체 채팅 보내기</u></p> <ul style="list-style-type: none"> - 메시지 정보를 전체 채팅 DB 에 저장 -> 전체에게 메시지 보내기
<p><u>귓속말 채팅 가져오기</u></p> <ul style="list-style-type: none"> - 보낸사람과 받는사람의 데이터를 귓속말 채팅 DB 에서 검사 -> 리턴
<p><u>귓속말 보내기</u></p> <ul style="list-style-type: none"> - 상대방이 존재하는 유저인지 검사 -> 연결되어있는 유저인지 검사 -> DB 에 내역 저장 -> 메시지 보내기

6. 스키마 (옵션은 생략)

1. 유저(회원정보) 모델
<p>Username - 회원 ID</p> <p>Password - 회원 Password (해시화하여 저장)</p> <p>Email - 회원 Email</p> <p>Files - 회원이 업로드한 파일 _id (mongoose.schema.Types.ObjectId) 배열</p> <p>Sex - 회원 성별</p> <p>CreatedDate - 회원가입 시간</p> <p>CurrentDate - 최근 접속 시간</p>
2. 업로드한 파일 모델
<p>Filename, - 파일의 이름</p> <p>Path - 저장된 파일의 경로</p> <p>CreatedDate - 업로드된 시간</p> <p>OriginalName - 압축파일 원본 이름</p> <p>Size - 파일의 크기</p>

3. 채팅 참여자 모델

Username - 접속한 유저의 아이디
SocketId - 접속한 유저의 소켓 아이디
UserId - 유저 모델의 _id (유저 모델을 참조하고 있음.)
AccessTime - 접속한 시간

4. 전체 채팅 내역 모델

Sender - 보낸 유저 _id (유저모델을 참조하고 있음)
Username - 보낸 유저의 아이디
Message - 메시지 내용
CreateDate - 메시지를 보낸 시간

5. 귓속말 채팅 내역 모델

Sender - 보낸 유저 _id (유저 모델을 참조하고 있음)
Username - 보낸 유저의 아이디
Message - 메시지 내용
Receiver - 받는 유저 _id (유저 모델을 참조하고 있음)
ReceiverName - 받는 유저의 아이디
CreateDate - 메시지를 보낸 시간