

회의록					
교과목명	팀 프로젝트1	학년/학기	2/2	팀명/팀번호	사조참치/4조
회의일시	2021.10.02 (금) 22:00 ~23:00			장소	google meet
참 석 자	김윤기, 김명비, 박소민, 안선영				
안 건	프로젝트 진행 상황 검토, 앞으로 해야할 일 논의, 이슈사항				
회의내용	<p><프로젝트 진행상황 검토></p> <p>-회원가입 페이지, 로그인, 즐겨찾기 디자인 완료</p> <p>-로그인 페이지 회원가입 페이지 별도의 브랜치 추가 실질적인 컴포넌트 디자인 완료</p> <p>-최소 비용선택지(시간, 거리, 비용)중 임의로 하나를 선택 해 검색을 할 때 검색 결과로 해당 선택지에 대한 최소 비용 뿐만 아니라 그 최소비용에 대응되는 나머지 두개의 비용역시 검색 결과에 보여주어야 하므로 각 최소 비용 선택지에 해당하는 나머지 두개의 데이터를 초기 엑셀파일에 형태의 데이터에서 sql문으로 가공후 이 데이터를 insert할 테이블 제작후 insert완료.</p> <p>-검색 결과를 보여주는 모달창 구현 완료</p> <p>-클라이언트로 응답하는 데이터에 단위 추가(비용: 원, 시간: 시, 분, 초, 거리: km, m).</p> <p>-검색 결과 응답 API에 min_value와 other_values 키로 데이터를 나타내는 대신 cost, distance, time이라는 3개의 키를 써 각각의 데이터 명시.(기존 거리 검색 요청에 대한 응답 API는 3개의 최소 비용 선택지 중 사용자가 선택한 최소비용에 대한 데이터를 min_value라는 키값으로, 나머지 2개에 값은 other_values로 돌려주도록 설계를 했지만 이렇게 데이터를 응답하면 클라이언트에서 오직 키값으로 데이터의 위치를 판단하기 어렵다는 결론)</p> <p>-하나의 sql문을 제작하는 함수를 sql문 별로 나누어서 테이블 이름을 인자로 받는 방식으로 바꿈. (기존 데이터를 sql문으로 파싱하기 위한 플로이드와샬을 사용한 코드는 sql문이 모두 main함수 안에 존재 했고 테이블 이름을 변수가 아닌 스트링으로 명시했다. 따라서 데이터를 넣어야 한다면 해당 스트링을 모두 바꾸어야 하는 방식으로 진행되었으나 이 방식이 너무 복잡하다는 결론)</p> <p><앞으로 해야할 일 논의></p> <p>-다음주까지 회원가입이랑 로그인 완료하기</p> <p>-모달 디자인 경유버튼 추가</p>				
첨부자료					



회의록					
교과목명	팀 프로젝트1	학년/학기	2/2	팀명/팀번호	사조참치/4조
회의일시	2021.10.15 (금) 22:00 ~23:00			장소	google meet
참 석 자	김윤기, 김명비, 박소민, 안선영				
안 건	프로젝트 진행 상황 검토, 앞으로 해야할 일 논의, 이슈사항				
회의내용	<p><프로젝트 진행상황 검토></p> <p>-회원가입(이메일 입력 칸, 비밀번호, 비밀번호 확인) 기능 구현, 모든 입력 칸은 30자 내</p> <p>-이메일, 비밀번호는 정규식을 활용해서 조건에 맞는 이메일인지 비밀번호인지 체크</p> <p>비밀번호에 대한 조건은 8글자 특수문자, 영어, 숫자 최소 1개씩 포함</p> <p>이메일, 비밀번호에 대한 검증은 사용자가 웹사이트를 거치지 않고 검증을 요청할 때를 대비해서 서버 쪽에서 한번 더 검증</p> <p>이메일 유효 여부는 단순히 정규식을 통해서 검증을 했을 때는 이메일 형식만 검증이 가능하기 때문에 실제 존재하는 이메일인지를 확인하기 위해서 유저가 본인이 입력한 이메일로 들어가서 유효여부 검증 메일을 클릭해야 회원가입이 완료된다.</p> <p>이 이메일의 유효기간은 1시간이기 때문에 1시간이 지난 후에 유저가 클릭하면 메일 인증이 되지 않아서 다시 인증을 받아야 함</p> <p>-위 내용을 구현하기 위해서 user 테이블에 boolean type을 가진 checkedEmail 추가</p> <p>사용자가 이메일 검증을 위해서 이메일에서 회원가입 완료하기 버튼을 눌렀을 때 실제로 그 사용자가 눌렀는지를 확인하기 위해서 유저 테이블에 저장된 사용자에 대한 record를 대변하는 pk와 랜덤으로 생성된 문자열을 저장하는 테이블이 필요해서 그에 대한 테이블도 생성함.</p> <p><앞으로 해야할 일 논의></p> <p>-개념 설계 보고서 작성 준비</p> <p>-로그인 기능 완료하기</p>				
첨부자료					

회의록					
교과목명	팀프로젝트1	학년/학기	2/2	팀명/팀번호	사조참치/4조
회의일시	2021.10.22.(금) 22:00 ~23:00			장소	google meet
참 석 자	김윤기, 김명비, 박소민, 안선영				
안 건	진행상황 확인				
회의내용	<p><진행상황 확인></p> <p>1. 로그인창 구현 완료 -모든 페이지 헤더부분에 있는 로그인 버튼과 회원가입 버튼이 각각 마이페이지 버튼과 로그아웃 버튼으로 바뀌는 기능 구현 완료.</p> <p>2. 회원가입 창 -이메일 인증 창으로 가는 링크 추가</p> <p>3. 이메일 재발송 창 -사용자가 회원가입을 한 후 한시간 이내에 인증 이메일을 인증하지 않았을 경우 이메일 재발송을 위해 이메일을 입력하고 이메일을 재발송하는 버튼 구현 완료</p> <p>3. jwt를 통한 로그인 구현 완료 -accesstoken의 만료기한을 한시간으로 설정. accesstoken이 한시간이 지났을 경우 refreshtoken을 통해 accesstoken 재발급. -프론트를 리액트로 구현했기 때문에 통상적인 상황에서는 데이터를 state에 저장을 해야한다. 하지만 jwt로 로그인을 한 후 accesstoken을 state에 보관을 할 경우 페이지가 refresh되면 모든 state가 날라가는 문제로 인해 유저의 로그인 여부를 판단할 수 없다. 따라서 accesstoken을 브라우저 로컬 스토리지에 저장을 했다. -refreshtoken을 쿠키에 저장을 했고 accesstoken을 쿠키에 저장하지않은 이유는 accesstoken 관련된 쿠키가 탈취당했을시 탈취를 한 유저가 다른 사람 개인정보에 접근할 수 있기 때문이다.</p> <p>4. 코드 수정 -회원가입이랑 로그인 회원인증메일 발송, 컴포넌트 세 개 ->컴포넌트 하나로 수정해서 코드 좀 더 깔끔해짐. 회원가입 ,로그인, 회원인증 메일에 따라 다르게 보여야하는 부분은 별도로 props로 넘겨서 처리했다.</p> <p>5. 로그아웃 기능 완료 -사용자가 로그아웃 시 백엔드에서는 쿠키와 token table에 저장한 refreshtoken 삭제. 프론트에서는 브라우저 로컬 스토리지에 저장한 accesstoken 삭제. -로그아웃을 백엔드에서 진행하기전에 사용자가 실제로 로그인을 한 후 로그아웃을 누르는건지 확인을 하기 위해 verifyToken이라는 미들웨어를 통해 accesstoken 검증. 이 미들웨어는 사용자의 개인 데이터를 요구하는 모든 라우터에 미들웨어로 들어갈 예정.</p>				
첨부자료					