
데이터베이스 조별 프로젝트

최종보고서



소주제 1 박성현

소주제2 서기원

송낙원

이의동

유현석

최희란

I. 팀원 역할 및 소개

1. 소주제1 팀원

박성현 2015147565, 송낙원 2015147561, 유현석 2015147533

2. 소주제2 팀원

서기원 2015147531, 이의동 2015147562, 최희란 2015147550

3. 중간까지의 역할

ER-Diagram 및 Relation Schema 및 Script 생성 : 서기원, 박성현, 이의동

시스템 설계 문서 : 최희란, 송낙원

보고서 작성 : 유현석

4. 중간 이후의 역할

프론트엔드 to 백엔드: 서기원, 유현석, 박성현, 최희란, 송낙원

백엔드 to DB : 이의동

II. 단계별 프로젝트 진행 과정

1. 전체적인 프로젝트 방향성에 대한 회의

→ github를 사용하여 팀 프로젝트를 진행하기로 함, 날짜별 대략적인 진도를 정함

2. 역할 배정

→ 소주제에 맞게 프론트엔드 역할을 배정하고 DB가 설치된 팀원이 DB를 맞기로 함

3. 각자 맡은 역할 진행

→ 각자 맡은 역할에 맞는 프로젝트를 개발하고 현황을 공유하면서 진행

4. 프론트엔드 및 백엔드 하나로 합치기

→ 맡은 역할이 대략적으로 마무리 되었을 때 하나로 합침, 합치는 과정에서 생긴 오류 해결 및 서로에게 피드백

5. DB 연결

→ 그전까지는 백엔드에서 하드코딩을 통해 기능을 확인했지만 하나로 합친 후 DB가 설치된 팀원이 DB를 연결

6. 최종 점검

-> DB를 연결한 후 최종 점검 및 수정

7. 회의를 통해 보완

-> 수정할 내용을 마무리하고 디자인 통일 및 CSS 마무리

8. 마무리

-> 서버에 설치 및 보고서 제출

III. 기능별 구체적 구현 내용

1. 로그인 및 로그아웃

1-1 프론트엔드

먼저 vuex를 이용하여 현재 로그인된 정보를 저장하였습니다. vuex에는 store.js 파일에서 로그인 될 때 백엔드에서 넘어오는 유저 정보를 저장하고 로그아웃할 때는 다시 초기화 시켜줍니다.

로그인 페이지에서는 로그인된 후 해당 직책에 따라서 1. 평가자 2. 제출자 3. 관리자 이렇게 3개의 독립된 페이지로 이동하게 설정하였습니다.

만약 로그인이 되지 않는 상태라면 로그인 페이지로만 이동하고 로그인이 되지 않으면 이동을 못하게 구현을 하였습니다. 로그인하는 과정에서는 먼저 입력된 정보를 백엔드로 넘겨서 1. 아이디가 존재 여부 2. 해당 아이디에 맞는 비밀번호인지 이렇게 2가지 경우를 확인 후 해당 경우에 맞는 경고문을 출력합니다. 또한, 아이디가 없을 경우를 대비하여 회원가입 버튼이 존재하여 그 버튼을 누를 시 회원가입 창으로 이동하게 하였습니다.

- Header.vue : sign_out을 클릭할 시 로그아웃이 가능하게 구현되어있음
- Home.vue : 로그인 페이지로서 앞에서 언급한 경우에 따른 경고문 출력과 로그인 정보를 입력 후 성공하면 아래의 IndexPage로 이동, 회원가입 페이지로 이동 가능
- IndexPage.vue : 로그인이 될 시 해당 직책에 맞는 페이지로 이동시켜주는 역할 로그인이 되어있지 않으면 Home으로 다시 이동하여 로그인이 성공할 때만 이동 가능하게 설정
- store.js : 현재 로그인 되어 있는 정보를 저장하는 역할 및 로그아웃시 정보를 초기화 시키는 역할

1-2 백엔드

로그인 및 로그아웃을 구현하기 위해서 passport를 사용하였습니다. 이 passport에서는 현재 로그인된 유무를 파악하고 DB에 저장된 정보를 토대로 로그인을 하게 합니다. 또한 로그아웃을 할 시 기존에 로그인할 때 저장된 정보 및 로그인된 상태를 기존의 초기 상태로 돌려줍니다. 로그인하는 과정에서는 프론트에서 넘어온 정보를 토대로 1. 아이디가 존재 여부 2. 해당 아이디에 맞는 비밀번호인지 이렇게 2가지를 확인하여 해당 경우에 맞는 에러를 프론트로 전송합니다.

- passport/index.js : 쿠키 관리 및 로그인과 로그아웃을 관리해주는 기능
- login.js : 로그인 호출시 passport를 통해서 로그인 여부 및 성공 실패를 확인 후 프론트로 전송
- logout.js : 로그아웃 호출시 passport를 통해서 기존에 정보를 다 초기화 및 로그인 전 상태로 회귀

2. 회원 가입 및 정보변경, 탈퇴

2-1 회원가입 페이지

회원가입 페이지에서는 user database 저장에 필요한 정보를 입력 받습니다. 입력 받은 정보는 database의 domain 규칙 혹은 추가적인 규칙에 따라 유효성 검사를 거치게 됩니다. user database의 Key인 ID는 중복 검사를 실시하며, 유효성 검증 완료 이후 입력 받은 정보를 제출하면 database에 저장합니다.

- components/SignUpPage.vue : 회원가입 페이지입니다. 회원 가입에 필요한 정보 입력(Form 사용) 및 각각의 요소의 유효성을 검사합니다. 특히 key인 ID의 중복체크는 POST 요청을 통해 합니다. 모

든 요소의 유효성 검사를 마치면 form을 제출하고 로그인 페이지로 이동합니다.

- check_id.js : ID를 매개변수로 받아 DB에 같은 아이디의 존재 유무를 검색합니다.
- sign_up.js : form에서 제출한 정보를 DB에 저장하기 위해 정보를 전달합니다.

2-2 정보 변경 및 탈퇴 페이지

정보 변경 및 탈퇴는 한 페이지에 구현 하였습니다. store에 저장된 ID, password, 이름, 권한을 불러옵니다. ID, 이름, 권한, 생일 정보는 변경 불가능 하고 password, phone_number, address 의 정보는 수정 가능합니다. 정보 수정과 탈퇴를 위해서는 store에서 불러온 password와 입력한 password의 일치 가 필요합니다.

- ProfilePage.vue : 정보변경 및 탈퇴 페이지입니다. store에 저장된 정보를 미리 출력하며 수정 가능한 요소들의 입력을 받을 수 있습니다. 회원가입 페이지와 마찬가지로 입력된 정보들은 유효성 검사를 거친 뒤 form을 제출하고 로그인 페이지로 이동합니다. 탈퇴 기능은 기존 비밀번호만 입력하면 form을 제출하며 회원정보가 삭제 되며 역시 로그인 페이지로 이동합니다.
- profile.js : ID, password, phone_number, address를 매개변수로 받으며 비밀번호, 전화번호, 주소 정보를 변경하게 합니다.
- withdraw.js : ID를 매개변수로 받으며 회원정보를 삭제 합니다.

3. 평가자 기능

3-1 배당 파일 목록 페이지

로그인한 사용자가 승인된 평가자라면 배당 파일 목록 파일 페이지로 넘어오게 됩니다. 평가자의 아이디를 parameter로 하는 GET 요청을 서버로 전송하여 사용자에게 할당되었으나 사용자가 아직 평가하지 않은 파싱 데이터 시퀀스 파일 목록을 받아옵니다. 이 목록이 존재하면 사이트에 목록을 나타내고, 목록이 존재하지 않다면 “할당된 파일이 없습니다.”라는 문구를 띄웁니다. 표에서 평가하고자 하는 파일 정보가 나타나는 행을 클릭하면 그 파일을 평가할 수 있는 페이지로 넘어갑니다.

- views/ToDoListView.vue : 페이지 구조를 정의하는 파일입니다. 서버로 get요청을 보내서 배당 파일 목록을 받아와서 AllocatedFileList.vue에 넘겨줍니다.
- components/EvalMenu.vue : 평가자가 사용하는 페이지의 왼쪽에 나타나는 메뉴 바입니다. 배당 파일 목록과 평가 완료 파일 목록을 누르면 해당 페이지로 이동합니다.
- components/AllocatedFileList.vue : 페이지의 오른쪽에 나타나는 부분입니다. ToDoListView.vue에서 넘겨준 배당 파일 목록을 띄워주는 역할을 합니다. vue.js에서 제공하는 v-for 문을 통하여 각 tuple들이 나타날 수 있도록, 각각 tuple의 정보를 AllocatedFile.vue에 보내줍니다.
- components/AllocatedFile.vue : AllocatedFileList.vue에서 각 tuple의 정보를 받아와서 표시하는 역할을 합니다.

3-2 파일 평가 페이지

배당 파일 목록 페이지에서 파일이 나타난 행을 클릭했을 때 나타나는 페이지입니다. 해당 파일의 id를 GET 요청을 통해 서버에 전달하여 관련 데이터를 받아옵니다. 해당 파싱 데이터 시퀀스 파일에 대한 정보들이 나타나고 그 파일을 다운로드 할 수 있는 링크도 나타납니다. 또한 페이지의 하단에 파일 점

수와 저장 여부를 입력할 수 있는 폼이 있습니다. 입력한 후에 제출 버튼을 누르면 post 요청을 통해 데이터를 백엔드로 전송하고, 전송이 성공하면 성공했다는 메시지가 나타난 후에 해당 파일 목록 페이지로 넘어가게 됩니다.

- view/EvalTaskEvalView.vue : 페이지 구조를 정의하는 파일입니다.
- component/EvalTodoTask.vue : 평가할 파싱 데이터 시퀀스 파일에 대한 정보를 보여줄 컴포넌트입니다. 이 파일에서 GET요청을 서버로 보내서 해당 파싱 데이터 시퀀스 파일에 대한 정보를 받아옵니다. 파싱 데이터 시퀀스 파일의 아이디, 테스트, 타입, 회차, 기간 정보와 전체 튜플 수, 중복 튜플 수, column별 Null 값 비율, 파일 다운로드 링크를 보여줍니다. 또한 파일을 평가할 수 있는 폼이 있습니다. 그 폼을 작성한 후 제출 버튼을 누르면 평가 정보가 전송됩니다.

3-3 평가 완료 파일 목록 페이지

평가자 페이지들의 왼쪽 메뉴바에서 평가 완료 파일 목록 버튼을 누르면 나타나는 페이지입니다. 평가자의 아이디를 GET 요청을 통해 서버에 전송하면, 서버에서 평가자가 평가한 파일 목록을 보내줍니다. 평가한 파일이 없다면 “평가한 파일이 없습니다”라는 문구가 페이지에 나타나고, 있으면 목록이 나타납니다.

- view/EvaluatedListView.vue : 페이지 구조를 나타내는 파일입니다. 여기서 서버로 평가 완료 파일 목록을 받는 요청을 보내서 해당 목록을 받고, EvaluatedFileList.vue로 목록을 넘겨줍니다.
- components/EvaluatedFileList.vue : 파일 목록을 받아서 표 형태로 목록들을 보여주는 파일이다. 각 파일에 대한 tuple을 v-for를 통하여 EvalutedFile.vue에 넘겨줌으로써 표를 생성합니다.
- components/EvaluatedFile.vue : EvalutedFileList.vue에서 각 파일에 대한 tuple을 받아서 행을 출력합니다.

4. 제출자 기능

로그인한 사용자가 제출자일 경우 IndexPage.vue에서 분기가 되어 제출자 페이지로 라우팅 됩니다.

4-1 메인화면

제출자의 메인 페이지(Submitter.vue)는 Welcome! {{회원이름}}을 중앙에 표시해 회원의 이름 정보를 보여주었고 직관적으로 “TASK 신청” 페이지와 “DATA 제출” 페이지를 선택할 수 있도록 버튼 2개를 구현했습니다.

- components/user/Submitter.vue : 버튼 2개를 통해 GetTask.vue또는 SubmitData.vue로 라우팅

4-2 TASK 신청

먼저 “TASK 신청” 버튼을 클릭해 GetTask.vue로 넘어가면 본인의 id 정보를 가지고 신청 가능한 Task 목록, 신청 대기 중인 목록을 불러옵니다. 기본적으로 task는 task 이름, 최소 업로드 주기, 설명 속성들을 보여주도록 설정했고 task 이름은 버튼으로 구현해 task 신청을 위해 선택할 수 있습니다. task를 선택하면 바로 밑 Selected Task : 부분에 선택한 task 이름이 출력됩니다.

신청 대기 중인 목록은 아직 신청은 했지만 허가 나지 않은 task 목록입니다.

task를 선택하고 개인정보 이용동의서에 체크해야 task 신청이 됩니다. 만일 둘 중 하나라도 안 할 경우 알림창이 뜨며 신청 접수가 되지 않습니다. task 신청을 성공하면 “{{task 이름}}가 신청되었습니다.”라는

알림창이 뜨고 신청 task 이름을 백엔드로 전달하며 접수가 완료됩니다.

뒤로 가기를 클릭하면 다시 제출자 메인 페이지로 돌아갑니다.

- components/user/GetTask.vue : 신청 가능 task 목록, 신청 대기 중인 task 목록 확인, 개인정보 이용 동의서 체크, task 선택 및 신청 기능

4-3 DATA 제출

다시 메인 페이지에서 “DATA 제출” 버튼을 클릭하면 DATA 제출 페이지로 넘어갑니다. 가장 위에서 자신의 평가 점수를 확인할 수 있습니다. 그 밑으로 데이터 제출이 가능한 task 목록이 보여지고 task 이름을 클릭해 선택할 경우, Selected Task : 에 해당 task가 출력됩니다.

그 후 Original Data Type : 옆의 옵션을 열면 해당 task의 원본 데이터 타입 목록들을 확인할 수 있습니다. 원본 데이터 타입을 선택하면 옆에 해당 데이터 타입 이름이 출력되면서 밑에 해당 원본 데이터 타입에 제출한 파일들의 목록들이 테이블로 생성됩니다. 기본적으로 제출된 파일 이름, 회차, pass여부를 출력합니다.

제출된 파일 테이블 밑엔 지금까지 제출자가 제출한 총 파일 수와 pass된 총 tuple 수를 출력합니다.

그 밑에 파일을 제출할 수 있는 업로드 창이 있고 file를 선택해 올리면 Selected file : 에 이름이 출력됩니다. 그 밑으로 회차와 기간을 입력할 수 있는 input 창을 구현했고 위 모든 것을 선택, 입력, 업로드를 하면 SUBMIT 제출이 가능해집니다. 모든 요구사항을 충족하지 못할 경우 알림창이 뜨게 됩니다. 성공적으로 제출되면 제출되었다는 알림창이 뜨며 백엔드에 파일, 회차, 기간, task 이름, 원본 데이터 타입 이름 정보가 넘어갑니다.

- components/user/SubmitData.vue : 평가 점수, 신청 허가된 task 목록, task 별 원본 데이터 타입 목록, 원본 데이터 타입 별 제출한 파일 정보, 파일 제출 및 해당 파일 정보 입력 기능

5. 관리자 기능

- 기능별 폴더를 활용하여 파일들을 관리
- Backend/routes/task.js : TASK관리에 관련된 프론트-백 통신을 담당
- Backend/routes/member.js : 회원관리에 관련된 프론트-백 통신을 담당

5-1 메인화면

관리자 계정으로 로그인 시 TASK 관리 페이지와 회원 관리 페이지를 선택하여 해당 페이지로 이동할 수 있습니다.

- AdmMain.vue : 관리자 계정 메인 화면. TASK 선택 시 TASK 관리 화면으로 이동. 회원 관리 선택 시 회원 관리 화면으로 이동
- Task/TaskMenu.vue : TASK 관리 페이지의 좌측 메뉴
- Member/Menu.vue : 회원 관리 페이지의 좌측 메뉴

5-2 TASK 통계

우선 현재 생성된 TASK의 이름, TASK의 간략한 설명, 최소업로드 주기, TASK 데이터 테이블 이름과 원본 DATA 타입의 이름이 표를 통해 한 눈에 볼 수 있게 페이지 하나로 표현하였습니다. 각 TASK를 클릭하여 선택하게 되면 해당 TASK의 상세 정보 페이지로 이동하게 됩니다. TASK의 상세 정보 페이지

지에서는 이전 페이지의 표에서 보여주었던 TASK 이름, TASK 설명, TASK에 지금까지 제출된 전체 파일의 개수, 기준에 부합하여 통과된 튜플의 수, 원본 DATA 타입 수준의 튜플 수를 상단에서 확인할 수 있습니다. 현재 DB에 저장된 튜플들에 대한 정보의 다운로드를 관리자가 희망할 때 생성 버튼을 클릭하여 정보를 담은 CSV 파일을 생성하고 다운로드 버튼을 재차 클릭하여 다운 받아 데이터를 확인할 수 있습니다. TASK DATA TABLE의 스키마를 이 화면에서 확인할 수 있고 바로 아래에서 현재 TASK에 설정된 원본 DATA 타입들의 스키마 정보를 확인할 수 있습니다. 다음 표에서는 TASK에 현재 참여 중인 제출자의 명단을 명시하였습니다.

- ShowInfo.vue : TASK 목록을 보여주는 페이지의 최상위 부모 컴포넌트.
- TaskInfo.vue : TASK 리스트를 나타내는 컴포넌트.
- Task.vue : TASK 목록 row의 데이터 형식을 나타내는 컴포넌트. TASK 클릭 시 TASK 상세 정보 페이지로 이동.
- TaskDetail.vue : TASK 상세 정보 페이지를 나타내는 최상위 부모 컴포넌트.
- UserView.vue : TASK에 참여하는 제출자 목록을 나타내는 컴포넌트.
- User.vue : 제출자 목록 row의 데이터 형식을 담은 컴포넌트. 제출자 클릭 시 회원 상세 정보 페이지로 이동.

5-3 TASK 생성

TASK 이름, TASK에 대한 간략한 설명, 최소업로드 주기, 제출 파일에 대한 PASS/NON PASS 기준을 text input 형식으로 입력할 수 있게 구현하였습니다. Extra Credit 요소 중의 하나인 TASK DATA 수집 종료를 구현하기 위해 TASK 생성 시 TASK DATA 수집 시작 날짜와 수집 종료 날짜를 설정할 수 있게 하였습니다. TASK 데이터 테이블의 스키마를 정의할 시 관리자가 SQL을 사용할 것인지를 결정할 수 있습니다. 사용하게 된다면 위와 마찬가지로 SQL을 text input 형식으로 입력할 수 있고 이는 string 형태로 백으로 전달됩니다. SQL을 사용하지 않으면 표의 형태로 속성별 이름과 변수 타입을 지정할 수 있게 하였습니다. 관리자가 TASK 생성 시 원본 DATA 타입을 한 가지 지정해야 하는데 이를 위해 원본 DATA 타입 이름을 설정할 수 있는 text input 빈칸과 원본 DATA 타입의 속성 이름, 변수 타입과 TASK DATA 테이블 스키마 중 매핑할 정보를 입력할 빈칸을 TABLE 형태로 구현하였습니다. 위의 정보 중 입력되지 않은 정보가 존재하면 TASK 생성 버튼을 누를 때 TASK 생성이 되지 않고 경고 문구가 나오게 설정하였습니다.

- TaskCreate.vue : TASK 생성 페이지의 최상위 부모 컴포넌트.
- CreateForm.vue : TASK 생성 시 필요한 정보 입력하는 format을 나타내는 컴포넌트.

5-4 TASK 관리

TASK 관리 메뉴를 들어가게 되면 먼저 현재 생성된 TASK 목록이 나오고 관리할 TASK를 선택하게 되면 해당 TASK의 관리 페이지로 넘어가게 됩니다. 우선 해당 TASK에 참가를 신청한 제출자의 목록이 ID, 이름, 평가 점수의 형태로 표로 나옵니다. 평가 점수를 바탕으로 참가를 승인 여부를 관리자가 정하면 해당 이름을 선택하여 참가 승인을 합니다. 정상적으로 승인이 되었을 경우 팝업창을 띄웁니다. TASK의 데이터 파일 PASS 기준은 드래그를 통해 재설정을 할 수 있게 설정하였습니다. 제대로 설정이 되었을 경우 마찬가지로 팝업창이 나옵니다. 원본 DATA 타입 추가는 TASK 생성 시 사용했던 원본 DATA 타입 생성 형태와 똑같이 구현하였습니다.

- TaskManage.vue : TASK 관리 페이지의 최상단 부모 컴포넌트.

- TaskList.vue : 현재 생성된 TASK의 목록을 나타내는 컴포넌트.
- ShowList.vue : TASK 목록의 row 데이터 정보를 나타내는 컴포넌트. 해당 row 선택 시 해당 TASK 관리 페이지로 이동.
- ManageView.vue : TASK 신청자 승인, PASS 기준 재설정, 원본 DATA 타입 추가 생성을 화면에 띄우는 컴포넌트.
- Register.vue/RegisterUser.vue : 해당 TASK에 참가 신청을 한 제출자 목록을 나타내는 컴포넌트.
- ResetStandard.vue : TASK의 데이터 PASS 기준을 재설정 부분을 담당하는 컴포넌트.
- AddType.vue : TASK의 원본 DATA 타입을 추가 생성하는 부분을 담당하는 컴포넌트.

5-5 회원 정보 확인

회원 관리 페이지를 접속하게 되면 현재까지 가입한 전체 회원 리스트를 확인할 수 있습니다. 회원 리스트는 회원 번호와 회원 이름을 확인할 수 있습니다. 리스트에서 한 명을 클릭하게 되면 해당 회원의 상세 정보 페이지로 이동하게 됩니다. 상세 정보 페이지에서는 회원 ID, 이름, 역할, 성별이 나타나고 제출자일 경우 참가하고 있는 TASK의 간략한 정보 목록이 나오게 되고 평가자일 경우 지금까지 평가한 제출 파일들의 ID, 점수와 PASS 여부가 표로 나타나게 됩니다.

- ShowMember.vue : 가입한 회원 리스트를 보여주는 페이지의 최상위 부모 컴포넌트.
- MemberInfo.vue/Member.vue : 회원ID, 회원 이름으로 구성된 회원 리스트를 보여주기 위한 컴포넌트. 특정 회원을 선택 시 해당 회원 상세 정보 페이지로 이동.
- UserInfo.vue : 회원 상세 정보 페이지를 담당하는 컴포넌트. 제출자일 경우 불러와야 하는 참여 TASK 정보를 담당하는 TaskInfo 컴포넌트와 평가자일 경우 불러와야 하는 평가한 FILE 정보를 담당하는 FileInfo 컴포넌트를 포함.
- UserTask.vue/Task.vue : 제출자가 참여 중인 TASK의 정보를 보여주는 컴포넌트. 평가자일 경우 참여 중인 TASK가 없으므로 “평가자입니다” 문구를 대신 출력.
- EvalFile.vue/File.vue : 평가자가 평가한 FILE 정보를 보여주는 컴포넌트. 제출자일 경우 평가한 FILE이 없으므로 “제출자입니다” 문구를 대신 출력.

5-6 회원 검색

검색 기준은 “TASK 이름, 회원 번호, 성별, 나이, 역할”으로 설정하였고, 검색 시 해당 조건에 부합하는 회원 리스트가 하단에 나타나게 됩니다. 하단 리스트에서 회원을 클릭할 시 위와 마찬가지로 해당 회원 상세 정보 페이지로 넘어가게 됩니다.

- Search.vue : 회원 검색 페이지를 나타내는 컴포넌트. 사용자가 입력하는 검색어들을 통해 DB에 질의한 후 부합하는 회원 목록을 받아 MemberInfo 컴포넌트로 전달. 이는 위에서 회원 목록을 보여주는 컴포넌트와 동일.

6. 백엔드 및 디비 기능

백엔드의 경우 javascript에서 제공하는 mariaDB 핸들을 관리해주는 API를 이용해 백엔드와 DB를 연결해 주었습니다. Nodejs는 비동기 처리방식인데, 외부 리소스인 DB를 사용하기 위해선 응답을 기다리는 순차적 처리가 필수적입니다. 이를 해결하기 위해 Promise 객체를 적극 활용하였습니다. 또한 중간 보고서에서 제시한 ERD와 크게 벗어나지 않는 DB를 설계하였고, 무결성 유지를 위해 백엔드쪽에서 프론트엔드로 DB status를 보내주어 DB의 처리가 완료되기 전에 프론트엔드가 다른 일을 하지 않도록 제약해 주었습니다.

다. 백엔드에서 DB를 콜하기 위한 함수는 backend/scripts/Functions에 구현하였고, 나머지는 routes의 라우터 파일들에서 처리해주었습니다.

6-1. 로그인

프론트엔드 메인화면에서 json 형태로 넘겨진 id, password 정보를 이용해 DB의 USER 테이블과 검사하여 튜플을 리턴해 주었습니다. 해당하는 튜플이 없을 시 에러 메시지를 전송하여 프론트에서 유저에게 경고 메시지를 줄 수 있도록 해주었습니다.

6-2. 회원가입, 정보변경 및 탈퇴

회원가입의 경우 프론트에서 넘겨진 유저 정보 json 데이터를 이용해 아이디 중복여부 등을 DB에서 체크한 후 프론트에 메시지를 전송하도록 해주었습니다. 탈퇴의 경우 USER 테이블에서 튜플을 DELETE하여 처리해 주었습니다. 정보변경의 경우 UPDATE SET문을 활용해 처리해 주었습니다.

6-3. 제출자 기능

6-3-1. 파일 제출

파일 제출의 경우 Node에서 Python child process를 생성하여 처리해 주었습니다. 유저가 제출한 파일 아이디는 유저ID+현재시각의 정보로 유니크한 키로 매핑됩니다. 제출한 파일은 다음과 같은 순서로 순차적 처리로 DB에 저장됩니다. 1) 제출자가 제출한 원본 데이터 타입의 스키마 정보를 가져옵니다. 2) 원본 데이터 타입의 스키마에서 속성 목록과 그에 해당하는 data type 목록을 가져옵니다. 3) 파싱된 데이터 테이블인 parsing_data_sequence_file에서, 가장 배당된 데이터가 적은 평가자의 ID를 SQL을 통해 가져옵니다. 4) scripts/parse_csv.py를 콜하여 제출한 데이터가 원본 데이터 타입과 일치하는지 검사하고, javascript에서 처리할 수 있는 형태로 DB의 튜플 값들을 변환합니다. 5) 올바른 형식의 데이터라면, javascript 문을 이용해 튜플들을 원본 데이터 타입 테이블에 넣습니다.

6-3-2. 태스크 신청

DB의 APPLY 테이블에 튜플을 INSERT 해주는 방식으로 구현하였습니다.

그 외 화면에 여러 가지 UI 정보를 출력해주기 위해 SQL 문들을 활용하였습니다.

6-4. 관리자 기능

6-4-1. 태스크 생성

태스크 생성의 경우 페이지에서 제공하는 편리한 UI를 따라 몇가지 조건으로 태스크 스키마를 자동으로 생성하거나, 관리자가 직접 CREATE TABLE문을 입력하여 테이블을 생성할 수 있게끔 구현하였습니다.

6-4-2. 신청자 관리

APPLY 테이블에서 튜플을 DELETE하고, PARTICIPATE 테이블에 튜플을 INSERT 해주는 방식으로 구현하였습니다.

6-4-3. 태스크 통계, 회원 관리 및 나머지 UI들

Promise 객체들을 활용한 순차적 DB콜들과, SQL문의 활용으로 구현하였습니다.

6-5. 평가자 기능

6-5-1. 파일 평가 페이지

PARSING_DATA_SEQUENCE_FILE, FILE, SOURCE_DATA_TYPE 등의 테이블들을 활용해 UI를 구현하였으며, NULL 비율, 튜플 개수 등의 값들을 SQL과 Promise 객체들을 활용해 Real-time으로 계산하게끔 구현 해주었습니다.

6-5-2. 배당 파일 목록 & 평가 완료 파일 목록

SQL과 Promise 객체들을 활용해 구현 해주었습니다.

6-6. 다운로드 기능

평가자와 관리자에서 현재 태스크의 DB를 CSV파일로 저장하거나, 제출자의 파일을 저장할 수 있도록 구현하였습니다. 구현은 Javascript에서 Python child process를 생성하여 pandas 모듈을 활용해 구현하였고, Promise 객체들을 이용해 이상한 값이 프론트에 전달되지 않도록 해주었습니다.

IV. 시스템에 사용된 DB

```

CREATE TABLE USER(
  ID VARCHAR(10) NOT NULL UNIQUE,
  NAME VARCHAR(50) NOT NULL,
  PASSWORD VARCHAR(12) NOT NULL,
  PHONE_NUMBER VARCHAR(11) NOT NULL,
  ADDRESS TEXT,
  BIRTHDAY DATE NOT NULL,
  TYPE ENUM('ADMINISTRATOR', 'SUBMITTER', 'EVALUATOR') NOT NULL,
  SCORE INT DEFAULT 0, CONSTRAINT VALID_SCORE CHECK((SCORE >= 0 AND SCORE <= 10)),
  PRIMARY KEY(ID));

CREATE TABLE TASK(
  NAME VARCHAR(200) NOT NULL UNIQUE,
  DESCRIPTION TEXT NOT NULL,
  START_PERIOD DATE NOT NULL,
  END_PERIOD DATE NOT NULL,
  MIN_SUBMIT_PERIOD INT NOT NULL,
  STANDARD_OF_PASS INT, CONSTRAINT VALID_STANDARD_OF_PASS CHECK((STANDARD_OF_PASS >= 0 AND
STANDARD_OF_PASS <= 100)),
  DATA_TABLE_NAME TEXT NOT NULL UNIQUE,
  PRIMARY KEY(NAME));

CREATE TABLE MANAGE(
  MANAGER_ID VARCHAR(10) NOT NULL,
  TASK_NAME VARCHAR(200) NOT NULL,
  PRIMARY KEY(MANAGER_ID, TASK_NAME),
  FOREIGN KEY(MANAGER_ID) REFERENCES USER(ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(TASK_NAME) REFERENCES TASK(NAME)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

CREATE TABLE PARTICIPATE(
  PARTICIPANT_ID VARCHAR(10) NOT NULL,
  TASK_NAME VARCHAR(200) NOT NULL,
  PRIMARY KEY(PARTICIPANT_ID, TASK_NAME),
  FOREIGN KEY(PARTICIPANT_ID) REFERENCES USER(ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(TASK_NAME) REFERENCES TASK(NAME)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

CREATE TABLE APPLY(
  APPLICANT_ID VARCHAR(10) NOT NULL,
  TASK_NAME VARCHAR(200) NOT NULL,
  PRIMARY KEY(APPLICANT_ID, TASK_NAME),
  FOREIGN KEY(APPLICANT_ID) REFERENCES USER(ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(TASK_NAME) REFERENCES TASK(NAME)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

CREATE TABLE SOURCE_DATA_TYPE(

```

```

TASK_NAME VARCHAR(200) NOT NULL,
TABLE_NAME VARCHAR(200) NOT NULL,
SCHEMA_INFO TEXT NOT NULL,
PRIMARY KEY(TASK_NAME, TABLE_NAME),
FOREIGN KEY(TASK_NAME) REFERENCES TASK(NAME)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

```

CREATE TABLE FILE(
    ID VARCHAR(30) NOT NULL,
    NUMBER INT,
    START_PERIOD DATE,
    END_PERIOD DATE,
    TYPE ENUM('CSV', 'TAB'),
    TABLE_NAME TEXT NOT NULL,
    PRIMARY KEY(ID));

```

```

CREATE TABLE SUBMIT(
    SUBMITTER_ID VARCHAR(10) NOT NULL,
    FILE_ID VARCHAR(30) NOT NULL,
    PRIMARY KEY(SUBMITTER_ID, FILE_ID),
    FOREIGN KEY(SUBMITTER_ID) REFERENCES USER(ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY(FILE_ID) REFERENCES FILE(ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```

```

CREATE TABLE PARSING_DATA_SEQUENCE_FILE(
    FILE_ID VARCHAR(30) NOT NULL,
    TYPE ENUM('CSV') NOT NULL,
    P_NP ENUM('EMPTY', 'PASS', 'NONPASS') DEFAULT 'EMPTY',
    SCORE INT, CONSTRAINT VALID_SCORE CHECK((SCORE >= 0 AND SCORE <= 10)),
    DUPLICATED_TUPLE_NUMBER INT NOT NULL,
    SUBMIT_TIME DATE NOT NULL,
    ALL_TUPLE_NUMBER INT NOT NULL,
    EVALUATOR_ID VARCHAR(10) NOT NULL,
    PRIMARY KEY(FILE_ID),
    FOREIGN KEY(EVALUATOR_ID) REFERENCES USER(ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```

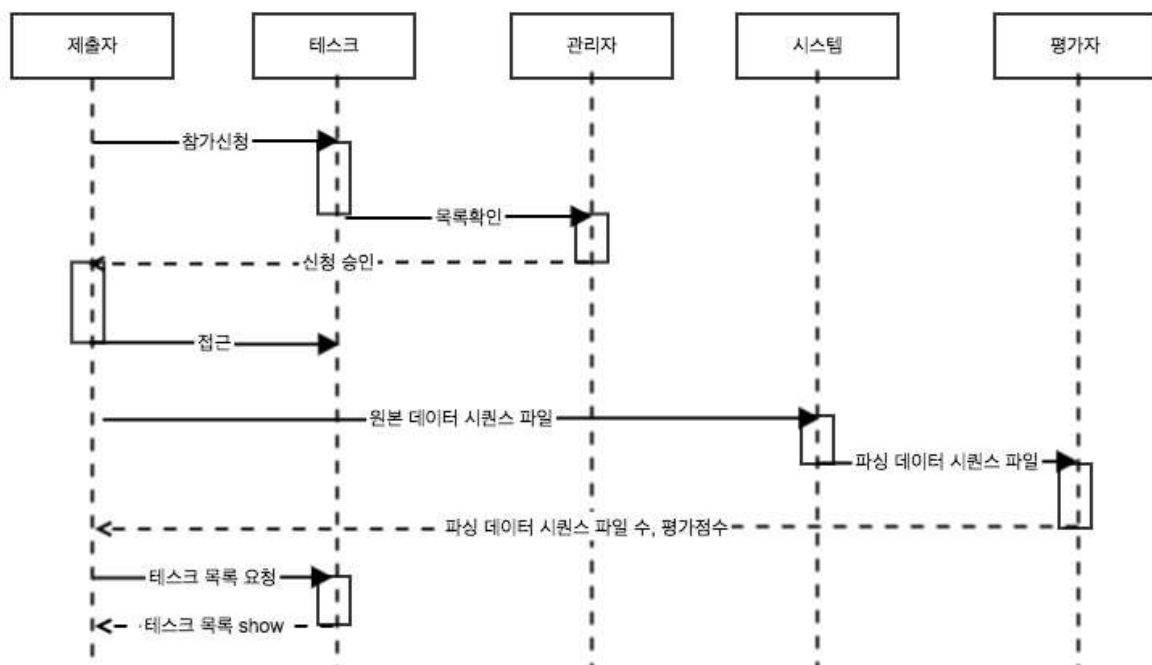
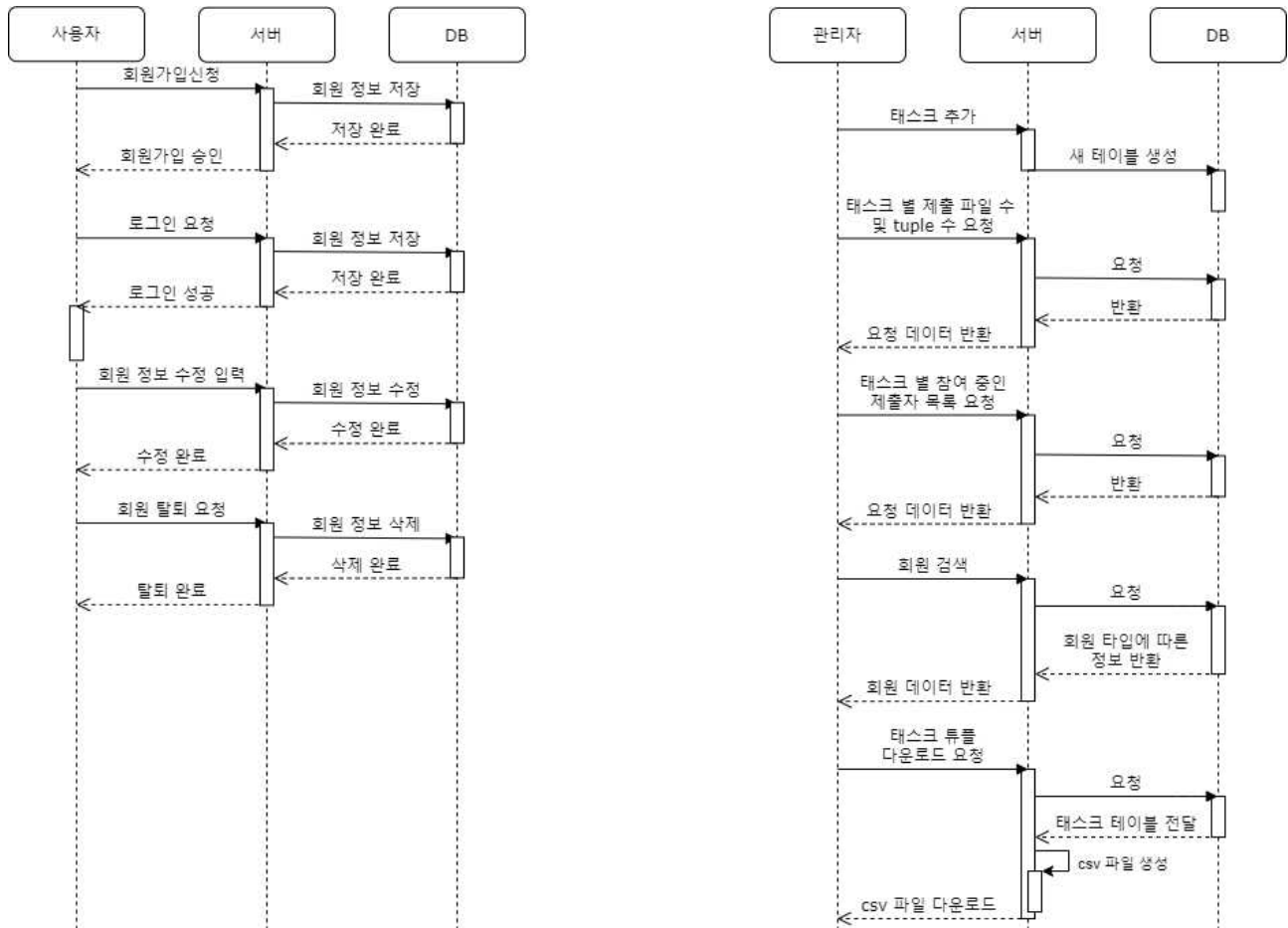
```

INSERT INTO USER(ID, NAME, PASSWORD, PHONE_NUMBER, BIRTHDAY, TYPE)
VALUES('admin', '관리자', 'admin', '01029876311', '1996-07-02', 'ADMINISTRATOR');

```

V. 변경/추가된 내용

1. 기존



TEAM1 변경사항

1. 기존 설계에는 할당된 파일 목록을 보는 부분 없이 바로 파싱 데이터 시퀀스 파일을 받아오도록 설계
이 부분을 할당된 파싱 데이터 시퀀스 파일 목록을 먼저 받아오고, 그 목록에서 파싱 데이터 시퀀스 파
일을 선택했을 때 평가할 수 있는 페이지로 넘어오도록 변경
2. 중간 설계에는 없던 평가자가 자신이 평가한 파싱 데이터 시퀀스 파일을 볼 수 있는 부분이 추가

TEAM2 변경사항

1. 기존에는 로그인 시 DB에만 정보를 저장하였지만 프론트엔드에도 해당 정보를 저장하는 방식으로 수정
2. 원본 데이터 타입을 DB에서 관리하기 위한 원본 데이터 타입 이름을 설정
3. 원본 데이터 타입을 정해진 TASK의 데이터 테이블 스키마와 매핑 형태를 딕셔너리 형태로 수정

VI. 추가적인 기능 및 팀만의 장점

1. 로그인이 되어야만 기능을 사용할 수 있게 하였습니다.
2. 관리자가 태스크 데이터 수집 기간을 설정할 수 있습니다.
3. // 추가기능 구현 : 제출 파일 다양화, 평가 기능 강화(평가자 여러 명에게 배정, 평가 시간에 제한두
기)
4. 팀 로고 및 개발자 소개 페이지를 만들었습니다.
5. 메인 화면 버튼을 클릭시 직책별 초기 페이지로 이동합니다.
6. 해당 작업 실패 시 알림 문구가 나와 사용자가 작업이 제대로 이루어지지 않았다는 것을 쉽게 알 수
있습니다.