

완료

관리자 화면에서 일반사용자 등록시 SSL 에러가 발생하는 문제가 있어서 accept\_send함수 호출을 주석처리하여 해결. 개발 PC상의 문제일 수 있으므로 향후 고객PC에서 주석해제하여 다시 확인 필요.

system\_manager\views\user\_management\_views.py

@login\_required(login\_url="/user/login/")

def register\_user(request, pk):

    if not request.user.is\_system\_manager:  # 시스템 매니저만 접근 가능

        return redirect("main:home")

    if request.method == "POST":

        user = CustomUser.objects.get(pk=pk)

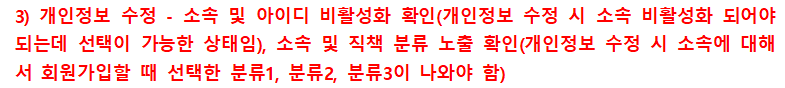
        user.register = True

        # accept\_send(user.phone)

        user.save()

        return redirect("system\_manager:user\_management")

    return redirect("main:home")



완료

templates\auth\editUser.html 수정

빨간 글씨 부분 추가

<div class="col">

<select name="class1" class="col-12 text-center" maxlength="60" id="id\_class1">

<option {% if user.class1 == "종합건설업체" %} selected {% endif %} value="종합건설업체">종합건설업체</option>

<option {% if user.class1 == "감리 및 CM 업체" %} selected {% endif %} value="감리 및 CM 업체">감리 및 CM 업체</option>

</select>

</div>

<div class="col">

<select name="class2" class="col-12 text-center" required="required" id="id\_class2">

<option {% if user.class2 == "일반 관리자" %} selected {% endif %} value="일반 관리자">일반 관리자</option>

<option {% if user.class2 == "현장 대리인" %} selected {% endif %} value="현장 대리인">현장 대리인</option>

</select>

</div>

<div class="col">

<select name="class3" class="col-12 text-center" maxlength="60" id="id\_class3">

<option {% if user.class3 == "시공 관리자" %} selected {% endif %} value="시공 관리자">시공 관리자</option>

<option {% if user.class3 == "품질 관리자" %} selected {% endif %} value="품질 관리자">품질 관리자</option>

<option {% if user.class3 == "안전 관리자" %} selected {% endif %} value="안전 관리자">안전 관리자</option>

<option value="해당 사항 없음">해당 사항 없음</option>

</select>

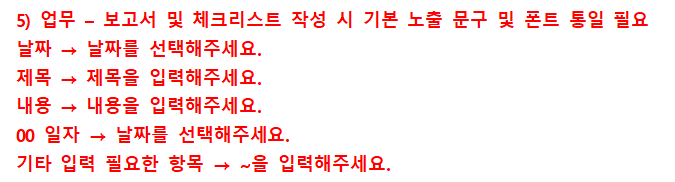
</div>

아래 코드 추가

$('#id\_class1 option').not(":selected").attr("disabled", "disabled");

$('#id\_class2 option').not(":selected").attr("disabled", "disabled");

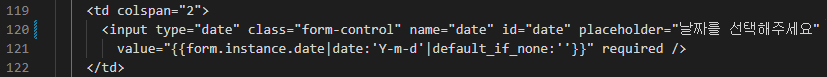
$('#id\_class3 option').not(":selected").attr("disabled", "disabled");

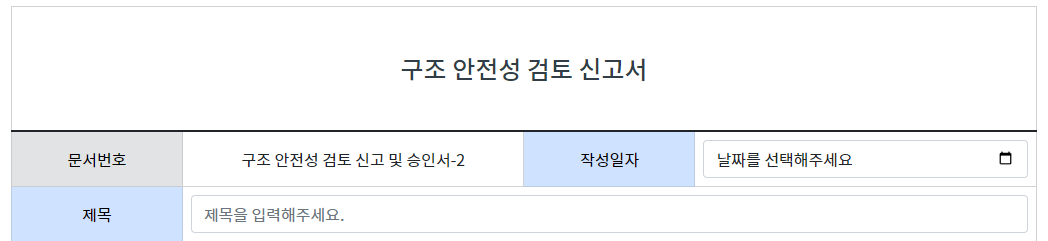


완료

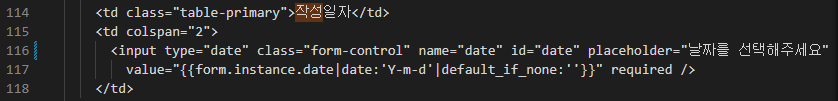
\templates\work\safety\create\_safety\_general.html

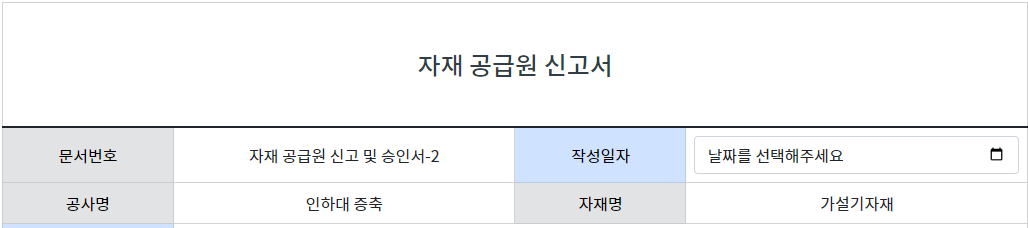
placeholder수정

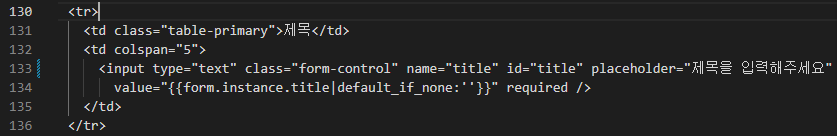


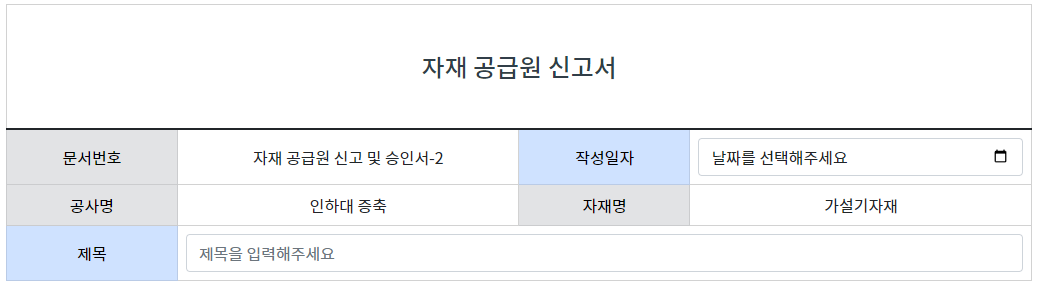


\templates\work\material\create\_material.html

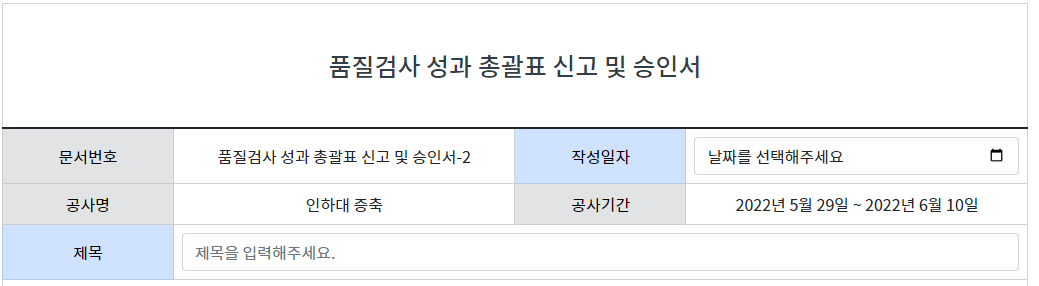
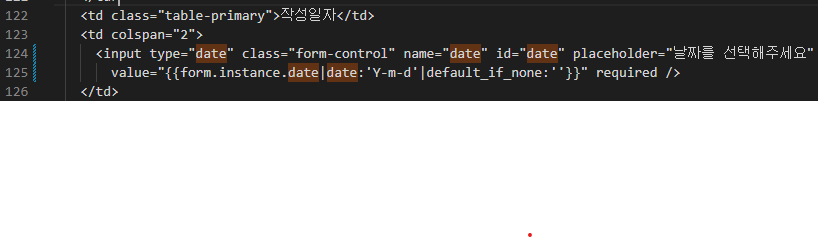


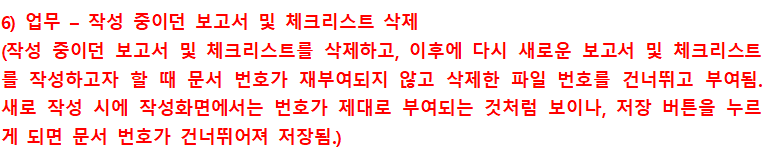






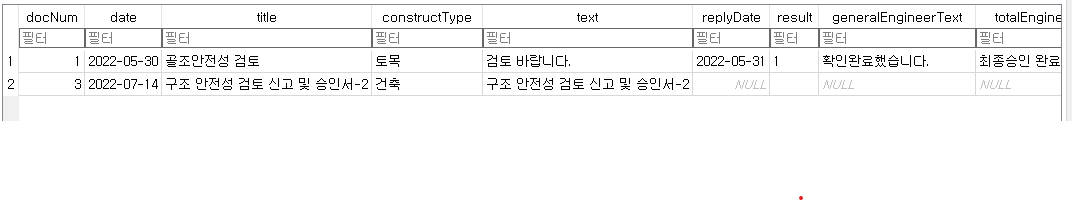
\templates\work\quality\quality\_report\create\_quality\_report.html





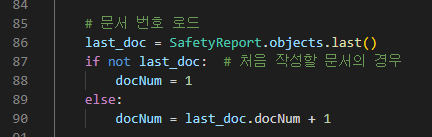
보류

저장중인 문서를 삭제하고 문서를 새로 생성하면 DB에서 docNum 필드를 통해 번호가 부여되는데, 이 필드는 Auto Increment 속성을 가지고 있기 때문에 이전의 문서가 지워졌다고 해서 다시 같은 번호가 부여되지 않기 때문임. docNum 필드는 고유값이기 때문에 한번 생성되면 변할 수 없음. 고객에 대해서는 설득이 필요함. 문서 1 생성 🡪 문서 2 생성(작성중) 🡪 문서 3 생성 후 승인 완료 🡪 문서 2 삭제 이렇게 진행한 경우, 문서 3의 번호는 그대로 문서 3을 유지하게 되는데, 이것은 문서3 생성 전에 문서2를 삭제했다고 해도 동일하게 동작함.

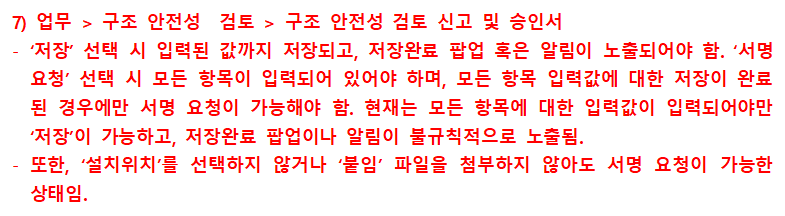


문서 생성시 번호가 잘못 부여되는 것은 아래 코드로 인함. 1번 문서 생성 🡪 2번 문서 생성 후 삭제 🡪 3번 문서 생성의 순으로 할 때 3번 문서 생성시에는 1번 문서의 번호(docNum)인 1에 1을 더해서 표시하기 때문에 2를 표시하는데 DB에 저장될 때는 Auto Increment 속성에 의해서 3이 저장됨.

\work\services\safety\_services.py

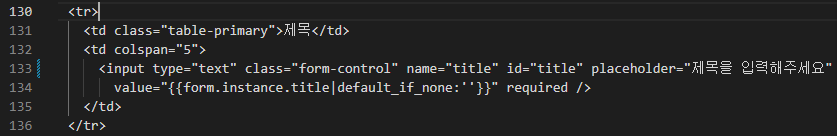


사실 문서가 생성(DB에 저장)되기 전에 문서의 번호를 미리 아는 것은 불가능합니다. 문서1생성🡪사용자1이 문서2생성하고 진행중 🡪 사용자2가 문서3생성하고 진행중 🡪 사용자2가 문서를 먼저 저장 이러한 경우에 사용자1이 부여받은 문서번호는 DB에 저장시 다른 번호를 부여받게 됩니다. 기술적으로 이것을 해결하는 방법은 문서를 처음 생성할 때 DB에 저장하고 primary key를 부여받는 방식인데 제가 구현하기에는 어려울 것 같습니다.



보류

작성일자의 경우, <input> 태그를 사용하는데 required가 없으면 placeholder문구가 표시되지 않는 문제가 있어서 required가 필요함. required를 지정하면 내용을 입력하지 않을 경우 저장이 불가능함. 따라서 이 기능을 구현하기 어려움.

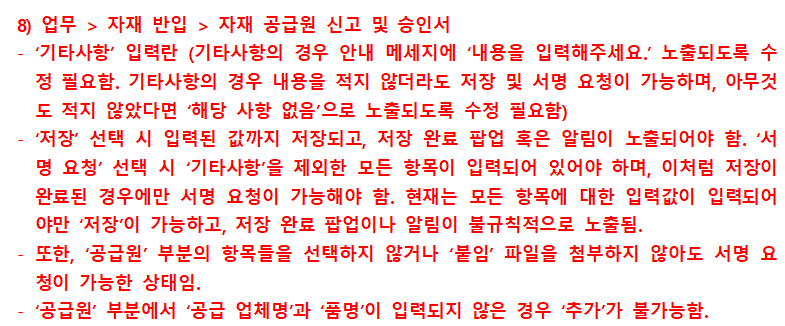


작성일자 이외의 내용은 required속성을 없애도 문제가 없을지는 확인해 보지는 않았습니다.

내용이 없이 저장하려면 DB 테이블의 필드 속성에서 NOT NULL이 없어야 하는데 이 부분도 확인이 필요합니다. 아래 DB를 보면 NOT NULL이 대부분 사용되고 있어서 기능을 구현하려면 DB를 수정해야 합니다.

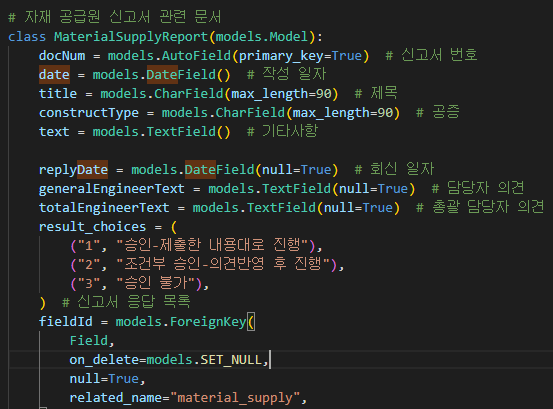
CREATE TABLE "work\_safetyreport" ("docNum" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "date" date NOT NULL, "title" varchar(90) NOT NULL, "constructType" varchar(90) NOT NULL, "text" text NOT NULL, "replyDate" date NULL, "result" varchar(10) NOT NULL, "generalEngineerText" text NULL, "totalEngineerText" text NULL, "isSaveManager" bool NOT NULL, "isSaveAgent" bool NOT NULL, "isSaveGeneralEngineer" bool NOT NULL, "isSaveTotalEngineer" bool NOT NULL, "isCheckManager" bool NOT NULL, "isCheckAgent" bool NOT NULL, "isCheckGeneralEngineer" bool NOT NULL, "isSuccess" bool NOT NULL, "checklistDate" date NULL, "checklistConstructType" varchar(90) NULL, "checklistTitle" varchar(90) NULL, "writerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "agentId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "generalEngineerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "totalEngineerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "created\_at" datetime NOT NULL, "updated\_at" datetime NOT NULL, "requested\_at" datetime NULL)

모든 내용을 기입하고 저장하는 것으로 고객과 협의가 필요할 것 같습니다.



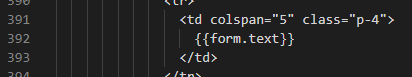
불가

기타사항은 DB에서 NOT NULL 속성으로 되어 있어서 수정이 불가합니다.

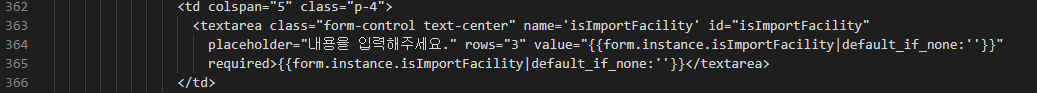


CREATE TABLE "work\_materialsupplyreport" ("docNum" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "date" date NOT NULL, "title" varchar(90) NOT NULL, "constructType" varchar(90) NOT NULL, "text" text NOT NULL, "replyDate" date NULL, "generalEngineerText" text NULL, "totalEngineerText" text NULL, "fieldId\_id" bigint NULL REFERENCES "system\_manager\_field" ("id") DEFERRABLE INITIALLY DEFERRED, "result" varchar(10) NOT NULL, "isSaveManager" bool NOT NULL, "isSaveAgent" bool NOT NULL, "isSaveGeneralEngineer" bool NOT NULL, "isSaveTotalEngineer" bool NOT NULL, "isCheckManager" bool NOT NULL, "isCheckAgent" bool NOT NULL, "isCheckGeneralEngineer" bool NOT NULL, "isSuccess" bool NOT NULL, "writerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "agentId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "generalEngineerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "totalEngineerId\_id" bigint NULL REFERENCES "user\_customuser" ("id") DEFERRABLE INITIALLY DEFERRED, "created\_at" datetime NOT NULL, "updated\_at" datetime NOT NULL)

아래와 같이 되어 있어서 placeholder를 지정할 수 없음.

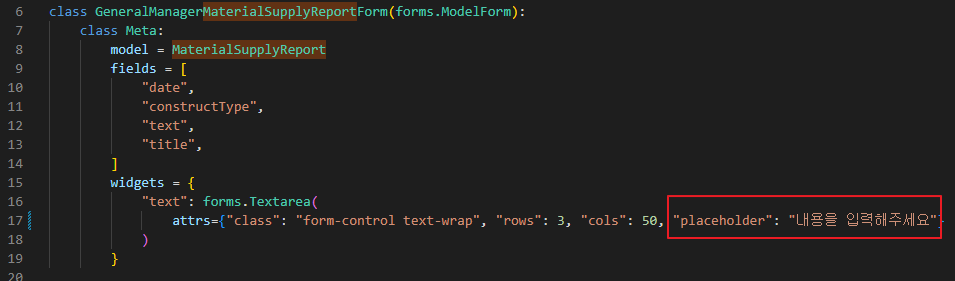


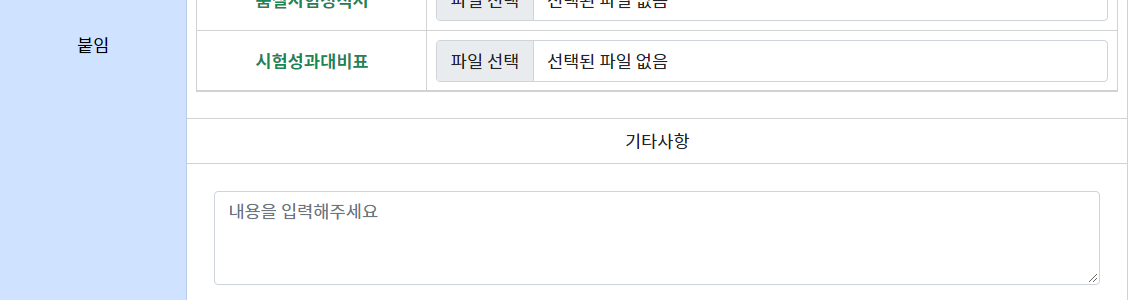
품질검사의뢰서의 경우 textarea 태그를 직접사용하고 있음.



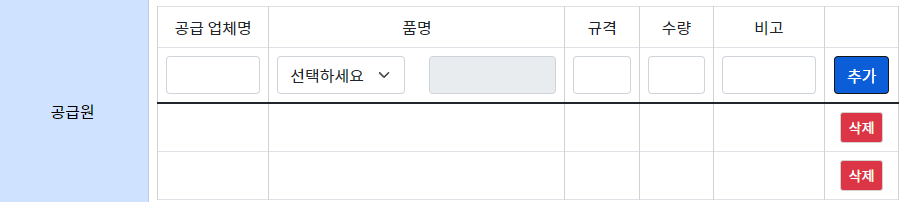
widgets에서 속성을 통해 지정하면 placeholder 내용 지정 가능

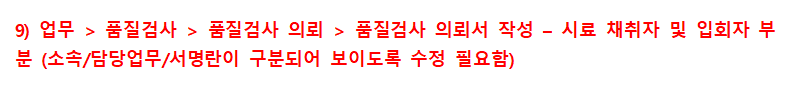
work\forms\material\_forms.py





공급원 부분에서 추가 버튼이 추가가 됩니다.





완료

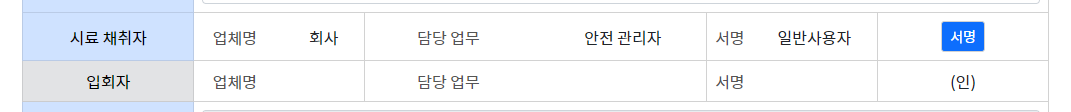
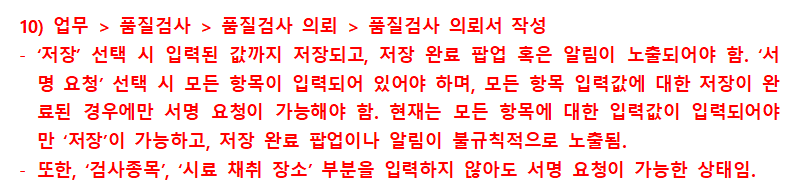


table 태그로 되어 있고 한 줄이 6의 colspan을 사용하고 있어서 업체명과 담당업무, 서명을 3등분으로 분할할 수 없습니다. 업체명부터 서명까지는 colspan값이 4인데 colspan은 소수점이 있는 숫자를 지원하지 않습니다. 그래서 담당업무에는 colspan 2를 부여하고 나머지에는 1을 부여하였습니다.



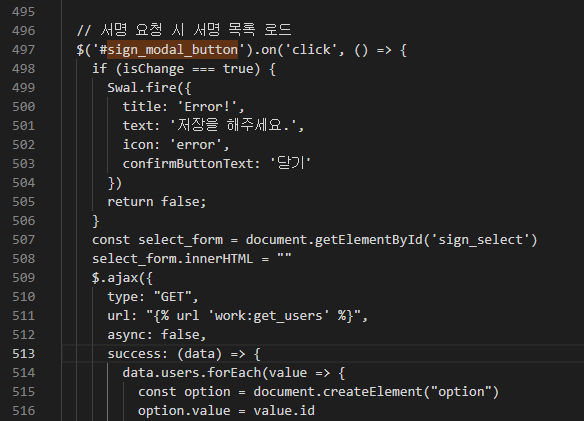
일부 불가

이전 수정사항과 유사한 이유. DB에 NOT NULL 속성으로 되어 있어서 입력하지 않은 상태에서 저장 불가.

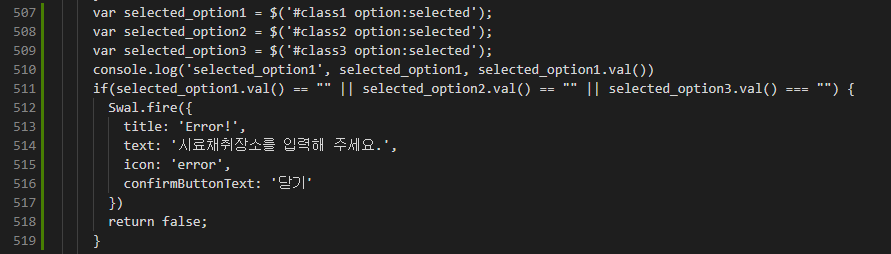
검사종목은 라디오버튼이 아니고 체크 박스라서 모두 체크가 안되도 상관이 없을 거 같음

서명요청시에 시료 채취 장소를 입력했는지 확인 필요

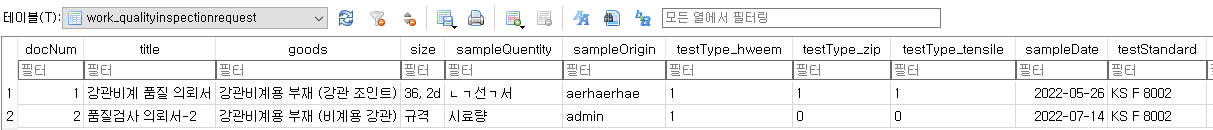
templates\work\quality\quality\_request\update\_quality\_request.html

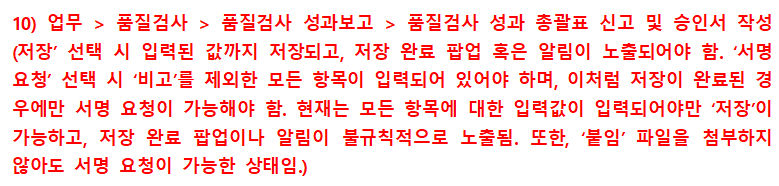


아래의 코드를 추가함.



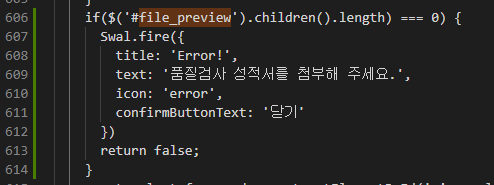
시료채취장소는 저장을 해도 이전 값이 표시가 안되는데 DB에 해당 필드가 없는 것으로 보임.

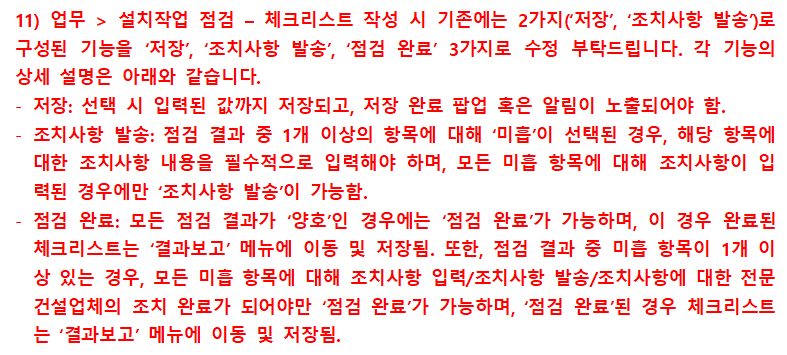




일부 수정

붙임 파일을 첨부하지 않으면 서명 요청이 안되도록 수정. 품질검사 성적서 첨부 안할시 에러가 뜨도록 수정함.





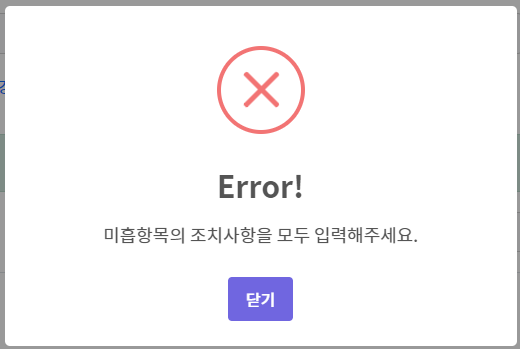
일부 수정

[현재 동작 확인]

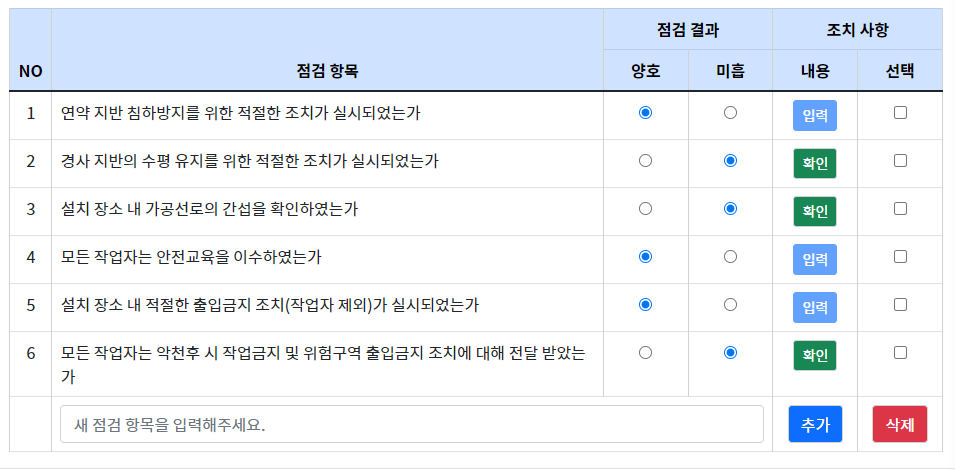
현재는 미흡항목에 대해서 내용을 입력한 경우에만 저장이 가능함. 아래와 같이 미흡이 선택된

경우에는 저장시 에러가 발생.



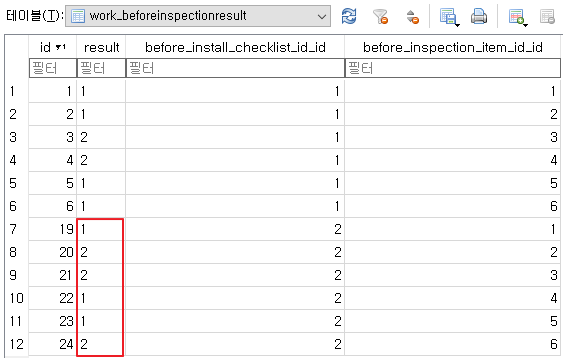


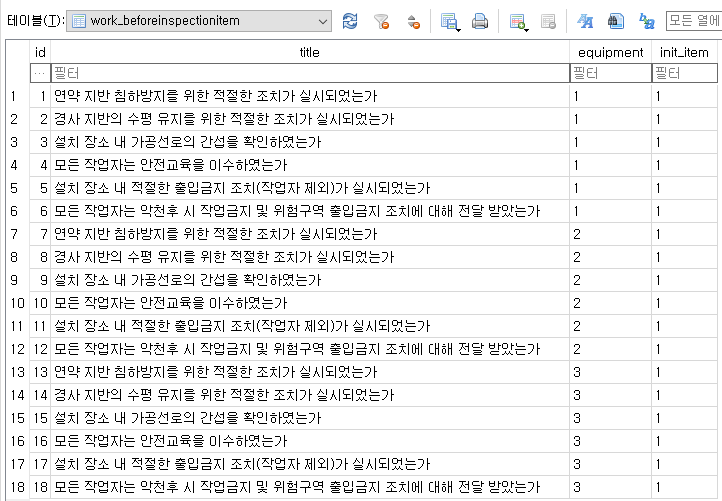
[분석]



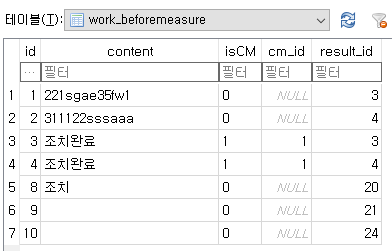
점검결과는 work\_beforeinspectionresult 테이블의 result 컬럼에 저장됨.1 = 양호, 2 = 미흡.

before\_install\_checklist\_id\_id는 체크리스트 문서의 번호를 나타냄. before\_inspection\_item\_id\_id는 work\_beforeinspectionresult 테이블의 id를 나타냄.

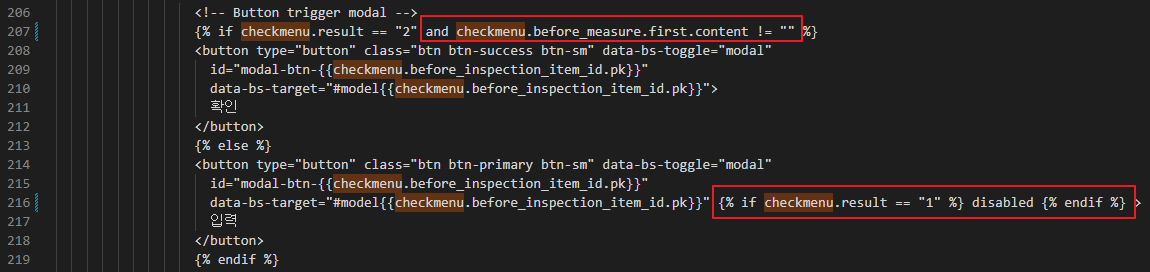


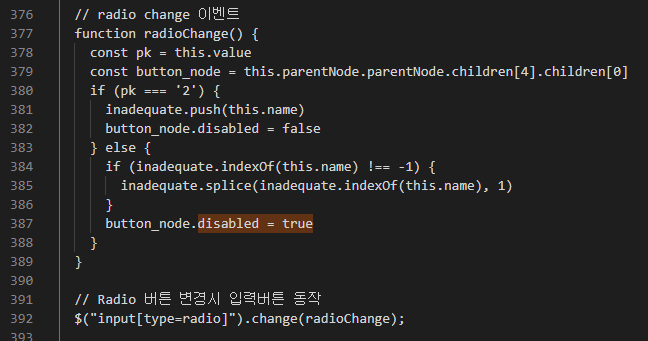


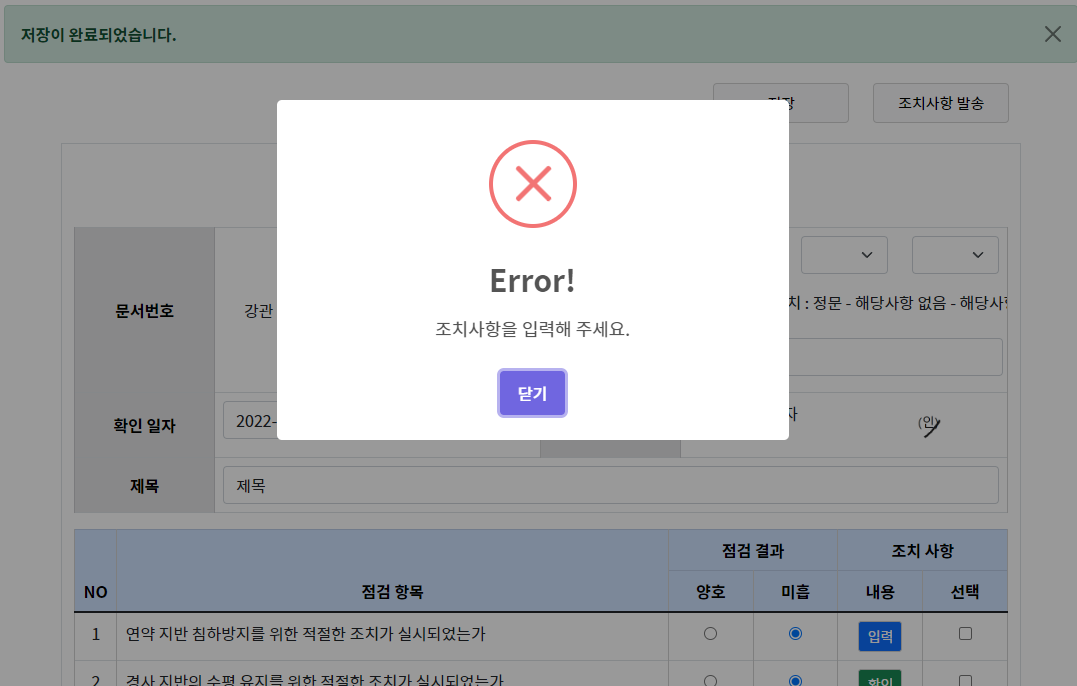
조치한 내용은 work\_beforemeasure 테이블에 저장된다.



조건 추가. 미흡(2)인 경우에 내용 입력이 되어 있어야만 확인버튼 표시. 미흡인 경우, 입력이 활성화되도록 조건 추가.

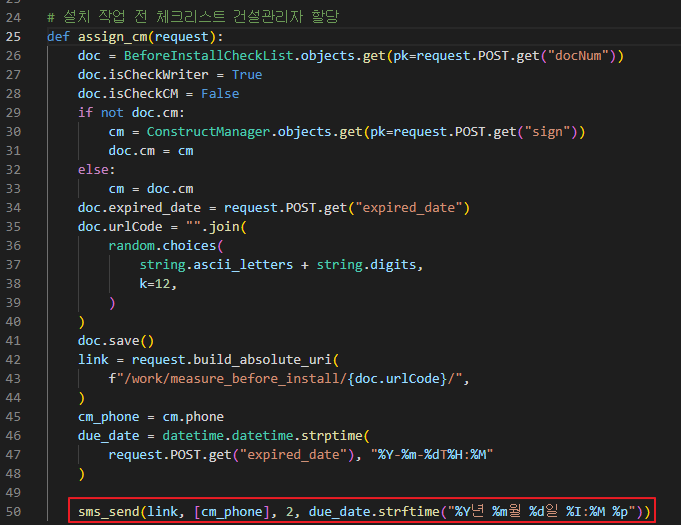






조치사항 전달시 SSL에러 발생.

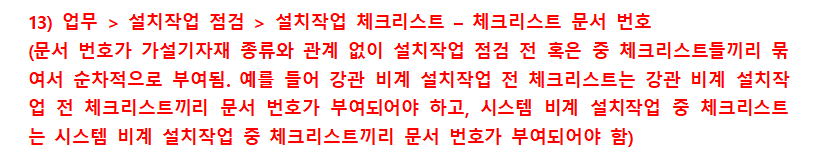
work\services\before\_install\_services.py



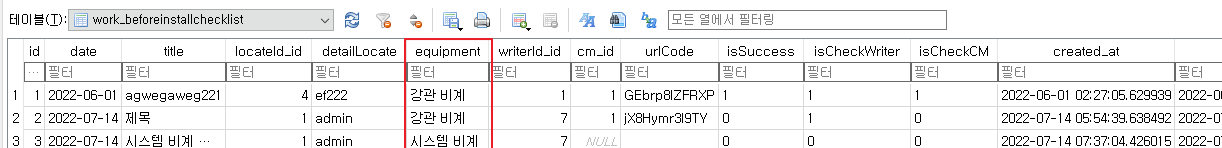
assign\_cm함수를 주석처리하고 실행.

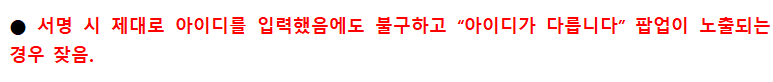
결과보고에 조치사항 문서가 잘 저장되어 있음을 확인.



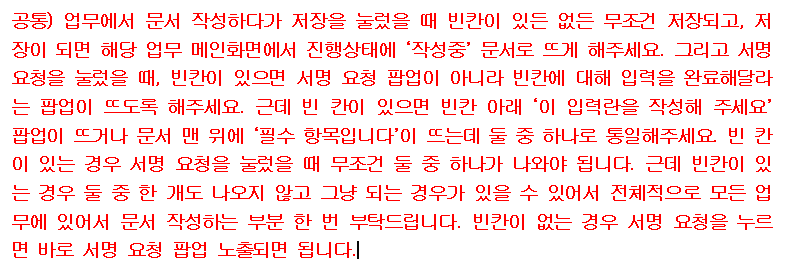


강관 비계, 시스템 비계, 시스템 동바리 등은 모두 같은 테이블에 저장되어 자동으로 번호가 부여되기 때문에 수정이 어려움.





서명시 한글키나 대문자키가 켜져 있지 않은지 확인하고, 엔터키를 누르면 안되고 마우스로 확인을 클릭해야 에러가 안남.



현재 DB구조에서 NOT NULL로 되어 있는 부분은 필수 입력이기 때문에 입력이 안된 상태에서 저장할 수 없도록 되어 있음.