

Introduction to Databases (Spring 2020)

DB Project Report

Student ID 2015313754

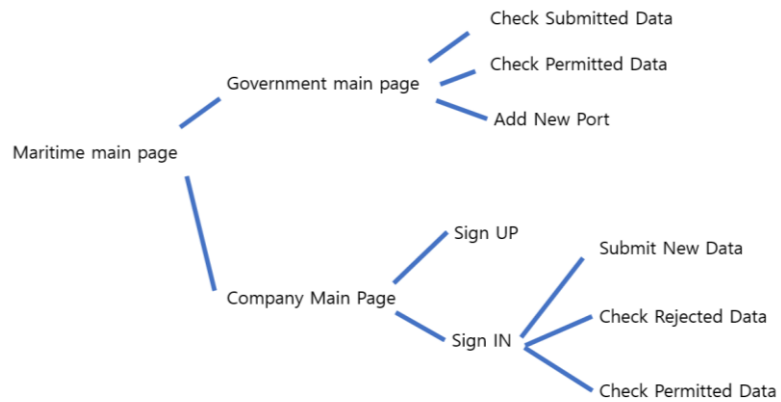
Name Taehyung Gil(길태형)

A. Overview of Project

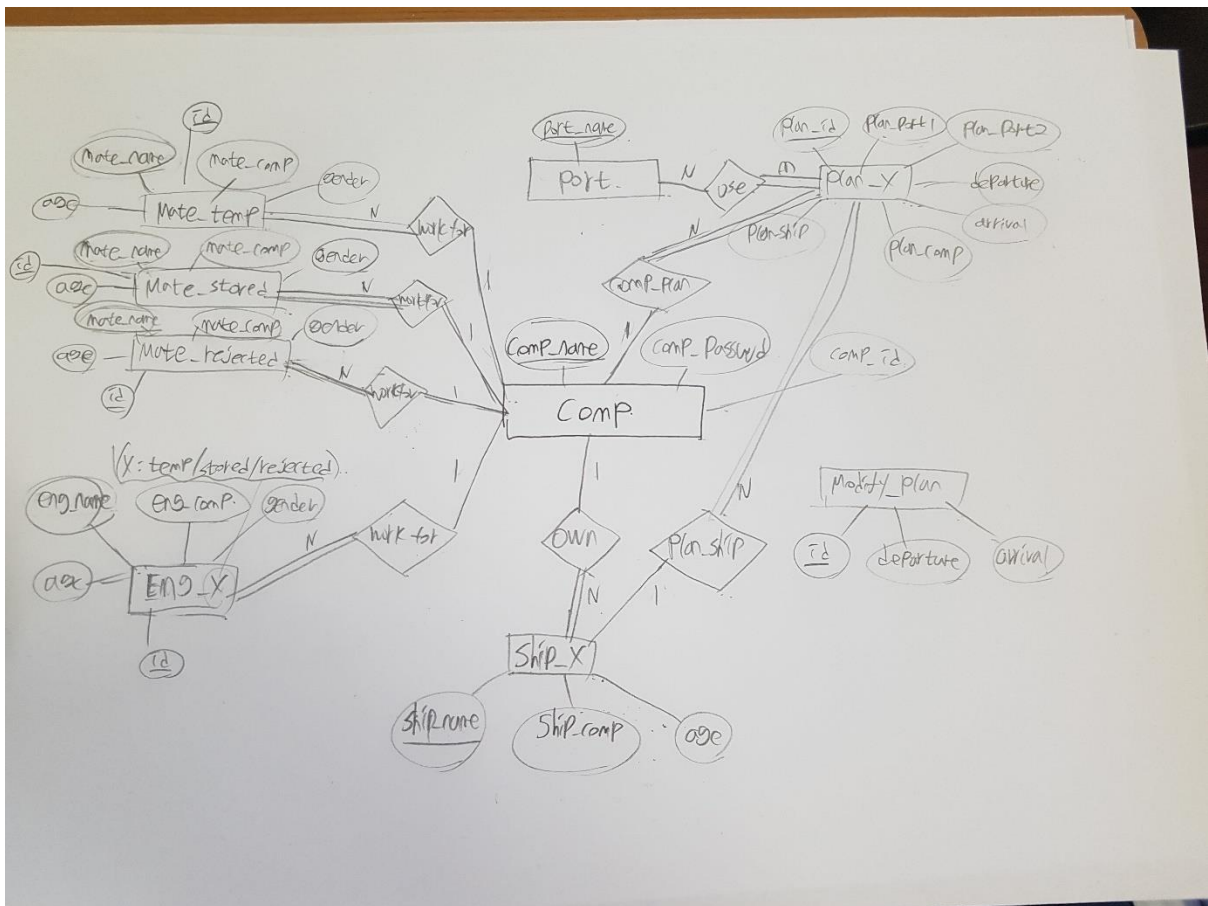
- Target
 - 해운회사와 정부의 Data 관리 시스템
- Objective
 - 해운회사가 회사 소속의 항해사, 기관사, 배, 항해 계획을 제출하면, 정부가 승인을 하거나, 미승인을 할 수 있도록 하며, 정부와 해운회사가 데이터를 열람할 수 있는 시스템을 구축한다.
- 기능
 1. Sign Up / Sign In / Sign out
 2. Check All permitted Data / Government & Company
 3. Add new Data / Government & Company
 4. Permit or Reject submitted Data / Government
 5. Check Rejected Data or Modify Data / Company

회사가 보유하고 있는 데이터의 종류는 항해사, 기관사, 배, 항해 계획으로 table로는 mate_stored, eng_stored, ship_stored, plan_stored가 있습니다. 정부에게 새로운 Data를 제출(Submit)하고, 미승인된 데이터를 다시 볼 수 있도록 X_temp, X_rejected (ex. Mate_temp, mate_rejected) 와 같은 테이블들을 별도로 만들었습니다. 정부에서는 모든 회사가 제출한 Data들을 열람할 수 있고, Permit을 하거나, Reject을 할 수 있음으로써 X_temp에 있는 데이터를 X_stored에 추가시키거나, X_rejected에 추가하게 됩니다. 또한, Plan table의 경우, 정부가 Reject을 했을 경우, Modify할 수 있도록 함으로써, Plan을 수정하여 다시 제출할 수 있도록 구현하였습니다.

<페이지 구성도>



B. ER Diagram



Description of ER Diagram

#Comp

-comp_name(Varchar type & Primary Key): 회사의 이름을 저장합니다.

-comp_id(Varchar type) : 회사의 Log IN ID를 저장합니다.

-comp_passwd(varchar type) : 회사의 Log IN Password를 저장합니다.

#Mate_X (X:temp, stored, rejected)

Temp: 회사가 새로 등록하기 위해 정부에 제출한 데이터를 저장하는 Table입니다.

Stored: 정부가 승인(permit)을 해서 정식 Data로 저장된 Table입니다.

Rejected: 회사가 제출했지만, 정부가 미승인한 Data들을 저장하는 Table입니다.

-id(int type & Primary Key): 해당 항해사의 고유키 입니다.

-mate_name(Varchar Type) : 해당 항해사의 이름을 저장합니다.

-mate_comp(Varchar Type & Foreign Key referencing Comp): 항해사가 소속된 회사의 이름을 저장합니다.

-gender(Varchar Type): 항해사의 성을 저장합니다.

-age(int): 항해사의 나이를 저장합니다.

#Eng_X (X:temp, stored, rejected)

-id(int type & Primary Key): 해당 기관사의 고유키 입니다.

-eng_name(Varchar Type) : 해당 기관사의 이름을 저장합니다.

-eng_comp(Varchar Type & Foreign Key referencing Comp): 기관사가 소속된 회사의 이름을 저장합니다.

-gender(Varchar Type): 기관사의 성을 저장합니다.

-age(int): 기관사의 나이를 저장합니다.

#ship_X (X:temp, stored, rejected)

- ship_name(varchar type & primary key): 해당 선박의 이름을 저장합니다.

- age(int): 해당 선박이 얼마나 오래됐는지 저장합니다.

- ship_comp(Varchar Type & Foreign Key Referencing Comp Table): 선박이 소속된 회사의 이름을 저장합니다.

#plan_X (X:temp, stored, rejected)

- plan_id(int & Primary Key): 해당 계획의 고유키 입니다.

- plan_port1(Varchar Type & Foreign Key Referencing Port Table): 선박이 항해를 시작하는 항구를 저장합니다.

- plan_port2(Varchar Type & Foreign Key Referencing Port Table): 선박이 항해를 끝내는 항구를 저장합니다.
- departure(dateType) : 선박이 항해를 시작하는 날을 저장합니다.
- arrival(dateType): 선박이 항해를 끝내는 날을 저장합니다.
- plan_ship(Varchar Type & Foreign Key Referencing Ship Table): 항해하는 선박을 저장합니다.
- plan_comp(Varchar Type & Foreign Key Referencing Comp Table): 선박이 소속된 회사를 저장합니다.

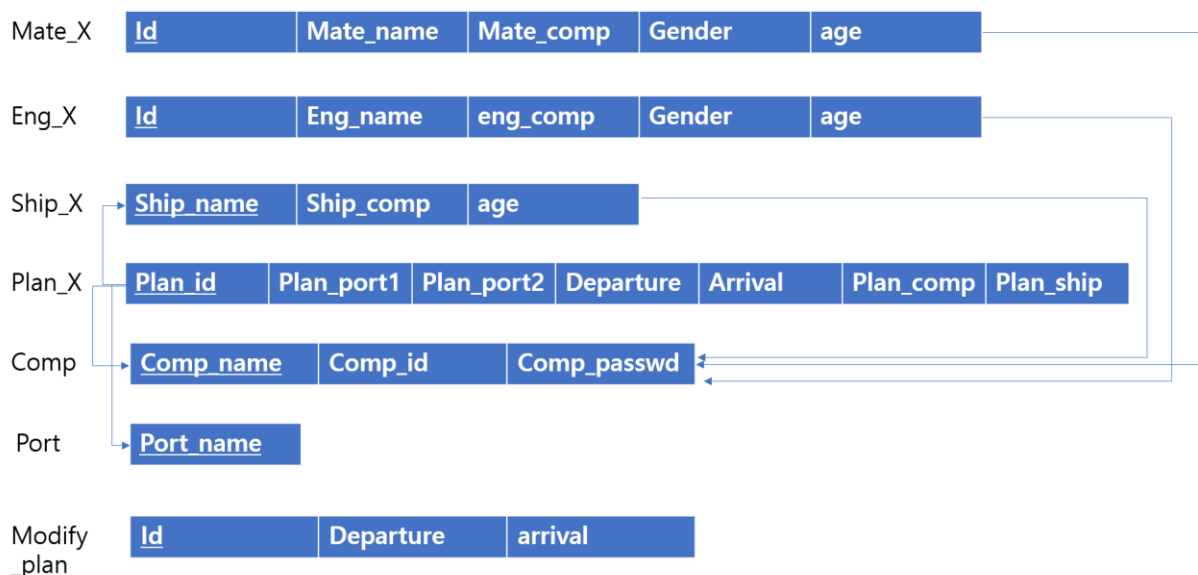
#port

- port_name(Varchar Type & Primary Key): 항구의 이름을 저장합니다.

#Modify_plan

- id(int & Primary Key): 변경된 항해 계획을 저장하는 Record의 고유키 입니다.
- departure(dateType): 변경된 항해 계획의 출발 일자를 저장합니다.
- arrival(dateType): 변경된 항해 계획의 도착 일자를 저장합니다.

C. Relational Model



Description of Relational Model

#Mate_X(X: temp/stored/rejected) :

회사에서 처음 등록을 하면, Mate_temp에 저장이 되고, 정부에서 승인(Permit)을 하면, Id를 제외한 속성 그대로 Mate_stored에 Record를 추가해서 저장합니다. 정부에서 미승

인(reject)을 하면, Id를 제외한 속성 그대로 Mate_rejected에 Record를 추가해서 저장합니다.

Primary Key : Id & Foreign Key : Mate_comp(Referencing Comp)

항해사 마다 고유의 키(Id)가 있고, 소속된 회사의 이름을 저장하는 Mate_comp column이 있습니다. 그 외에, 항해사의 이름을 저장하는 Mate_name, 성을 저장하는 Gender, 나이를 저장하는 age Column이 있습니다.

#Eng_X(X: temp/stored/rejected) :

회사에서 처음 등록을 하면, Eng_temp에 저장이 되고, 정부에서 승인(Permit)을 하면,

Id를 제외한 속성 그대로 Eng_stored에 Record를 추가해서 저장합니다. 정부에서 미승인(reject)을 하면, Id를 제외한 속성 그대로 Eng_rejected에 Record를 추가해서 저장합니다.

Primary Key : Id & Foreign Key : Eng_comp(Referencing Comp)

엔지니어 마다 고유의 키(Id)가 있고, 소속된 회사의 이름을 저장하는 Eng_comp column이 있습니다. 그 외에, 엔지니어의 이름을 저장하는 Eng_name, 성을 저장하는 Gender, 나이를 저장하는 age Column이 있습니다.

#Plan_X(X: temp/stored/rejected) :

회사에서 처음 등록을 하면, Plan_temp에 저장이 되고, 정부에서 승인(Permit)을 하면,

Id를 제외한 속성 그대로 Plan_stored에 Record를 추가해서 저장합니다. 정부에서 미승인(reject)을 하면, Id를 제외한 속성 그대로 Plan_rejected에 Record를 추가해서 저장합니다.

Primary Key : plan_id & Foreign Key : Plan_comp(Referencing Comp), Plan_ship(Referencing Ship), Plan_port1,2(Referencing Port)

항해 계획 마다 고유의 키(Id)가 있고, 소속된 회사의 이름을 저장하는 Plan_comp column, 항해하는 배를 저장하는 Plan_ship, 항해의 출발과 도착 항구를 저장하는 Plan_port1,2 가 있습니다. Departure과 Arrival Column은 출발과 도착 일자를 저장합니다.

#Ship_X(X: temp/stored/rejected) :

회사에서 처음 등록을 하면, Ship_temp에 저장이 되고, 정부에서 승인(Permit)을 하면,

속성 그대로 Ship_stored에 Record를 추가해서 저장합니다. 정부에서 미승인(reject)을 하면, 속성 그대로 Ship_rejected에 Record를 추가해서 저장합니다.

Primary Key : Ship_name & Foreign Key : Ship_comp(Referencing Comp)

배 마다 고유의 키(Ship_name, 배의 이름)이 있고, 소속된 회사의 이름을 저장하는 Ship_comp Column, 배가 얼마나 오래됐는지 저장하는 Age Column이 있습니다.

#Comp :

Primary Key : Comp_name

회사의 이름이 고유키가 됩니다. 그 외에, Sign in에 필요한 Comp_id, Comp_passwd를 저장합니다.

#Modify_Plan :

Primary Key : Id

정부가 미승인(Reject)한 Plan을 수정할 때 사용하는 Table입니다.

D. Implementation Results

<Functions>

- A. Sign Up / Sign In / Sign out
- B. Check All permitted Data / Government & Company
- C. Add new Data / Government & Company
- D. Permit or Reject submitted Data / Government
- E. Check Rejected Data or Modify Data / Company

A. Sign Up / Sign In / Sign out

#Sign UP a new Company (url pattern : /maritime/signup/)

name of new Company:

name of Company

ID of new Company:

log in ID of Company

Password of new Company:

log in password of Company

Go to Comp Main Page

<구현 코드>

Urls.py의 해당 코드

```
path('signup/', views.signup, name='signup')
```

views.py의 해당 코드

```
def signup(request):  
    form = SigUpForm()  
    return render(request, 'accounts/signup.html', {'form': form})
```

forms.py의 SignUpForm()코드

```
class SigUpForm(ModelForm):  
    class Meta:  
        model = Comp  
        fields = ['comp_name', 'comp_id', 'comp_passwd']  
        labels = {  
            'comp_name': _('name of new Company'),  
            'comp_id': _('ID of new Company'),  
            'comp_passwd': _('Password of new Company'),  
        }
```

해당 Form 이 아래의 Attribute 들을 받아옵니다.

Field	Type	Null	Key	Default	Extra
comp_name	varchar(20)	NO	PRI	NULL	
comp_passwd	varchar(20)	NO		NULL	
comp_id	varchar(20)	NO		NULL	

account/signup.html 코드

```
<form action = "/maritime/comp_signup/" method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <button type="submit">제출</button>
</form>
<hr>
<form action = "/maritime/comp_main_page/" method="get">
    {% csrf_token %}
    <h1>
        Go to Comp Main Page
    </h1>
    <button type="submit">Go</button>
</form>
<hr>
```

Urls.py 의 Signup_comp url action

path('comp_signup/', views.signup_comp, name='Comp_signup'),

views.py의 signup_comp 함수

```
def signup_comp(request):
    form = SigUpForm(request.POST)
    if form.is_valid():
        new_item = form.save()
        new_item.save()
        data = form.cleaned_data
    else:
```



```
class Comp_LogInForm(ModelForm):
    class Meta:
        model = Comp
        fields = ['comp_id', 'comp_passwd']
        labels = {
            'comp_id': _('ID of Company'),
            'comp_passwd': _('Password of Company'),
        }
```

accounts/login.html의 코드

```
<form action = "/maritime/connect_to_main/" method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <button type="submit">제출</button>
</form>
<hr>
<form action = "/maritime/comp_main_page/" method="get">
    {% csrf_token %}
    <h1>
        Go to Comp Main Page
    </h1>
    <button type="submit">Go</button>
</form>
<hr>
```

Urls.py의 해당 코드

```
path('connect_to_main/', views.connect_to_main, name='to_main'),
```

views.py의 해당 코드

```
def connect_to_main(request):
    form = Comp_LogInForm(request.POST)
    if form.is_valid():
        data = form.cleaned_data
        return
    HttpResponseRedirect(reverse('maritime:comp_main', args=(data['comp_id'],)))
```

해당 Sql문

```
SELECT comp_id FROM Comp WHERE comp_id=data['comp_id'];
```

#Sign Out

Company Main Home Page

Sign up as a new company

Check

Log in as a company

Check

Go to Maritime Main Page

Go

해당 form 코드

```
<hr>
<form action = "/maritime/maritime_main/" method="get">
  {% csrf_token %}
  <h1>
    Go to Martime Main Page
  </h1>
  <button type="submit">Check</button>
</form>
```

Maritime_main 페이지로 이동하게 되어, Sign Out의 효과를 가집니다.

B. Check All permitted Data / Government & Company

#Check a company's Permitted Data (url pattern : {{ comp_id }}/comp_check_permitted/)

Company name : pow

Mate

- Name : TH2 Gender : male Age : 23

Engineer

- Name : TH Gender : female Age : 23

Ship

- Name : GTH Age : 10

Plan

- Departure port : Beijing Arrival port : tokyo Departure Date : Jan. 3, 2020 Arrival port : March 1, 2020 Ship : king

Go to Company Main Page

Go

Urls.py

```
path('<str:comp_name>/comp_check_permitted/', views.comp_check_permitted,
name='comp_check_permitted'),
```

views.py

```
def comp_check_permitted(request, comp_name):
    comp = get_object_or_404(Comp, pk=comp_name)
    all_mates = Mate_stored.objects.filter(mate_comp=comp)
    all_engs = Eng_stored.objects.filter(eng_comp=comp)
    all_ships = Ship_stored.objects.filter(ship_comp=comp)
    all_plans = Plan_stored.objects.filter(plan_comp=comp)
    return render(request, 'comp/comp_check_permitted.html', {'comp': comp,
'all_mates': all_mates, 'all_engs': all_engs, 'all_ships': all_ships, 'all_plans':
all_plans} )
```

해당 SQL 문

SELECT * FROM mate_sotred WHERE mate_comp=comp_name;

SELECT * FROM eng_sotred WHERE eng_comp=comp_name;

SELECT * FROM ship_sotred WHERE ship_comp=comp_name;

SELECT * FROM plan_sotred WHERE plan_comp=comp_name;

Comp/check_permitted.html 코드

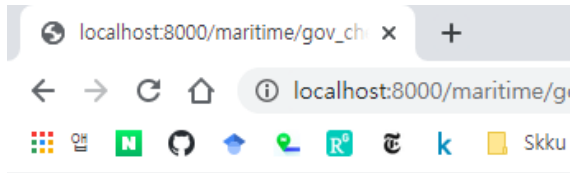
```
<h1>
    Company name : {{ comp.comp_name }}
</h1>
<!-- mates -->
{% if not all_mates %}
<h1>
    Mate Empty
</h1>
{% else %}
<h1>
    Mate
</h1>
<ul>
{% for mate in all_mates %}
    <li>
        Name : {{ mate.mate_name }}    Gender : {{ mate.gender }}    Age :
        {{ mate.age }}
    </li>
{% endfor %}
</ul>
{% endif %}
<!-- engineers -->
{% if not all_engs %}
<h1>
    Engineer Empty
</h1>
```

```

{% else %}
<h1>
    Engineer
</h1>
<ul>
{% for eng in all_engs %}
    <li>
        Name : {{ eng.eng_name }}    Gender : {{ eng.gender }}    Age :
        {{ eng.age }}
    </li>
{% endfor %}
</ul>
{% endif %}
<!-- ships -->
{% if not all_ships %}
<h1>
    Ship Empty
</h1>
{% else %}
<h1>
    Ship
</h1>
<ul>
{% for ship in all_ships %}
    <li>
        Name : {{ ship.ship_name }}    Age : {{ ship.age }}
    </li>
{% endfor %}
</ul>
{% endif %}
<!-- plans -->
{% if not all_plans %}
<h1>
    Plan Empty
</h1>
{% else %}
<h1>
    Plan
</h1>
<ul>
{% for plan in all_plans %}
    <li>
        Departure port : {{ plan.plan_port1.port_name }} Arrival port :
        {{ plan.plan_port2.port_name }}
        Departure Date : {{ plan.departure }} Arrival port : {{ plan.arrival }}
        Ship : {{ plan.plan_ship.ship_name }}
    </li>
{% endfor %}
</ul>
{% endif %}
<hr>
<form action = "/maritime/{{ comp.comp_name }}/comp_main/" method="get">
    {% csrf_token %}
    <h1>
        Go to Company Main Page
    </h1>
    <button type="submit">Go</button>
</form>

```

#Check all Permitted Data (url pattern : gov_check_permitted/)



Mate

Name	gender	age
sailor	male	23
aaa	male	23
ss	male	22
qwe123	male	22
TH2	male	23

Engineer

Name	gender	age	Comp
jh	male	20	sm
qweqwee	male	32	sm
ewq	female	32	rr
TH	female	23	pow

Ship

Name	age	Comp
aaa	23	gth
aaaa	32	sm
asdasdd	23	q23
bassa	231	rr
GTH	10	pow
king	20	sm
rew	22	sm

Plan

Departure	Arrival	Port1	Port2	Ship	Comp
Jan. 2, 2020	March 2, 2020	seoul	seoul	king	sm
March 1, 2020	April 1, 2020	seoul	seoul	king	sm
March 1, 2020	April 1, 2020	seoul	seoul	king	gth
March 1, 2020	April 1, 2020	seoul	seoul	aaa	sm
March 1, 2020	May 1, 2020	seoul	seoul	king	sm
Jan. 3, 2020	March 1, 2020	Beijing	tokyo	king	pow

Port

Port Name
asdasd
Beijing
Brooklyn
NY
Polo
seoul
tokyo

Go to Gov Main Page

[Go](#)

Urls.py

```
path('gov_check_permitted/', views.gov_check_permitted,
name='gov_check_permitted'),
```

views.py

```
def gov_check_permitted(request):
    all_mates = Mate_stored.objects.all()
    all_engs = Eng_stored.objects.all()
    all_ships = Ship_stored.objects.all()
    all_plans = Plan_stored.objects.all()
    all_ports = Port.objects.all()
    return render(request, 'gov/gov_check_permitted.html', {'all_mates':
all_mates, 'all_engs': all_engs, 'all_ships': all_ships, 'all_plans': all_plans,
'all_ports': all_ports})
```

해당 SQL 문

SELECT * FROM mate_stored;

SELECT * FROM eng_stored;

SELECT * FROM ship_stored;

SELECT * FROM plan_stored;

Gov_check_permitted.html

```
<!-- mates -->
{% if not all_mates %}
<h1>
    Mate Empty
</h1>
{% else %}
<h1>
    Mate
</h1>

<table class="table_mates_rejected">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>gender</B></td>
        <td><B>age</B></td>
    </tr>
    </thead>
    <tbody>
    {% for mate in all_mates %}
    <tr>
```



```

        <td> {{ mate.mate_name }}</td>
        <td> {{ mate.gender }} </td>
        <td> {{ mate.age }} </td>
    </tr>
{% endfor %}
</tbody>
</table>
{% endif %}
<!-- engineers -->
{% if not all_engs %}
<h1>
    Engineer Empty
</h1>
{% else %}
<h1>
    Engineer
</h1>

<table class="table_engineers_rejected">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>gender</B></td>
        <td><B>age</B></td>
        <td><B>Comp</B></td>
    </tr>
    </thead>
    <tbody>
    {% for eng in all_engs %}
    <tr>
        <td> {{ eng.eng_name }}</td>
        <td> {{ eng.gender }} </td>
        <td> {{ eng.age }} </td>
        <td> {{ eng.eng_comp.comp_name }} </td>
    </tr>
    {% endfor %}
    </tbody>
</table>
{% endif %}

<!-- ships -->
{% if not all_ships %}
<h1>
    Ships Empty
</h1>
{% else %}
<h1>
    Ship
</h1>

<table class="table_ships_submitted">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>age</B></td>
        <td><B>Comp</B></td>
    </tr>
    </thead>

```

```

<tbody>
{% for ship in all_ships %}
<tr>
<td> {{ ship.ship_name }}</td>
<td> {{ ship.age }} </td>
<td> {{ ship.ship_comp.comp_name }} </td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}

<!-- plans -->
{% if not all_plans %}
<h1>
Plan Empty
</h1>
{% else %}
<h1>
Plan
</h1>

<table class="table_ships_submitted">
<thead>
<tr>
<td><B>Departure</B></td>
<td><B>Arrival</B></td>
<td><B>Port1</B></td>
<td><B>Port2</B></td>
<td><B>Ship</B></td>
<td><B>Comp</B></td>
</tr>
</thead>
<tbody>
{% for plan in all_plans %}
<tr>
<td> {{ plan.departure }}</td>
<td> {{ plan.arrival }}</td>
<td> {{ plan.plan_port1.port_name }}</td>
<td> {{ plan.plan_port2.port_name }}</td>
<td> {{ plan.plan_ship.ship_name }}</td>
<td> {{ plan.plan_comp.comp_name }}</td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}

{% if not all_ports %}
<h1>
Port Empty
</h1>
{% else %}
<h1>
Port
</h1>

<table class="table_ships_submitted">

```

```

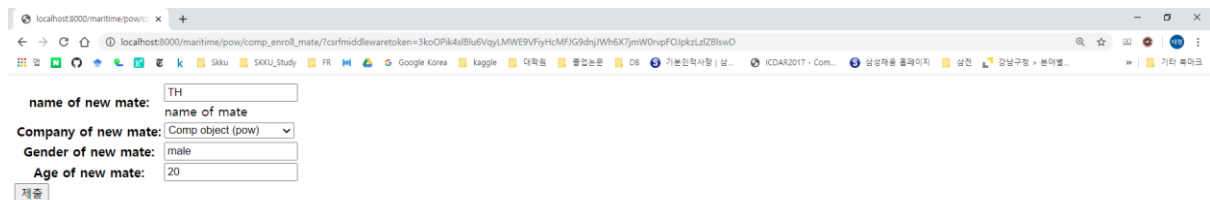
<thead>
<tr>
    <td><B>Port Name</B></td>
</tr>
</thead>
<tbody>
{% for port in all_ports %}
<tr>
    <td> {{ port.port_name }}</td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}

<hr>
<form action = "/maritime/gov_main_page/" method="get">
    {% csrf_token %}
    <h1>
        Go to Gov Main Page
    </h1>
    <button type="submit">Go</button>
</form>
<hr>

```

C. Add new Data / Government & Company

#Add Company's a new mate



Go to Company Main Page

Go

Urls.py

```

path('<str:comp_name>/comp_enroll_mate/', views.comp_enroll_mate,
name='comp_enroll_mate'),

```

views.py

```

def comp_enroll_mate(request, comp_name):
    if request.method == 'POST':
        form = MateForm(request.POST)

```

```

        if form.is_valid():
            new_item = form.save()
            new_item.save()
            return HttpResponseRedirect(reverse('maritime:comp_main',
args=(comp_name,)))
        mate_form = MateForm()
        return render(request, 'comp/comp_enroll_mate.html', {'form': mate_form,
'comp_name': comp_name})

```

forms.py

```

class MateForm(ModelForm):
    class Meta:
        model = Mate_temp
        fields = ['mate_name', 'mate_comp', 'gender', 'age']
        labels = {
            'mate_name': _('name of new mate'),
            'mate_comp': _('Company of new mate'),
            'gender': _('Gender of new mate'),
            'age': _('Age of new mate')
        }

```

comp_enroll_mate.html

```

<form action = "/maritime/{% comp_name %}/comp_enroll_mate/" method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <button type="submit">제출</button>
</form>
<hr>
<form action = "/maritime/{% comp_name %}/comp_main/" method="get">
    {% csrf_token %}
    <h1>
        Go to Company Main Page
    </h1>
    <button type="submit">Go</button>
</form>

```

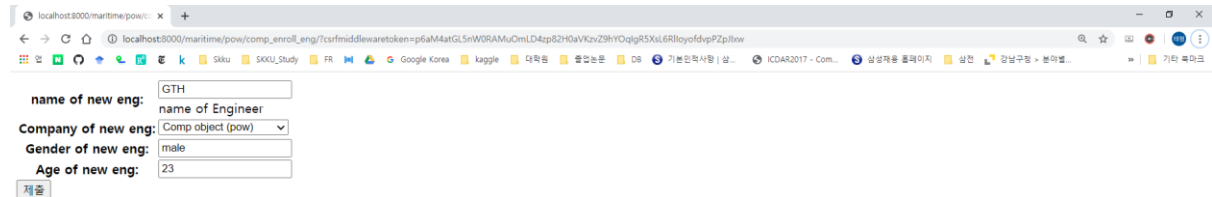
해당 SQL문

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
mate_name	varchar(20)	NO		NULL	
gender	varchar(10)	NO		NULL	
age	int	NO		NULL	
mate_comp_id	varchar(20)	NO	MUL	NULL	

```
INSERT INTO mate_temp(mate_name,gender,age,mate_comp_id) VALUES(data['mate_name'],
```

```
data['gender'], data['age'], data['mate_comp_id']));
```

#Add Company's a new Engineer

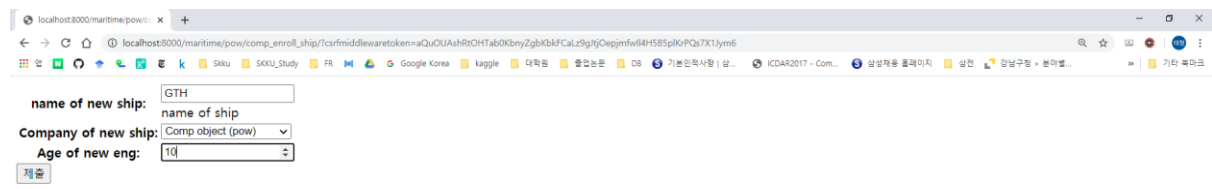


A web browser window showing a form titled 'name of new eng:'. The form includes a text input for 'name of Engineer' with the value 'GTH', a dropdown for 'Company of new eng:' with 'Comp object (pow)' selected, a dropdown for 'Gender of new eng:' with 'male' selected, and a text input for 'Age of new eng:' with the value '23'. A '제출' (Submit) button is at the bottom.

Go to Company Main Page

Go

#Add Company's a new Ship

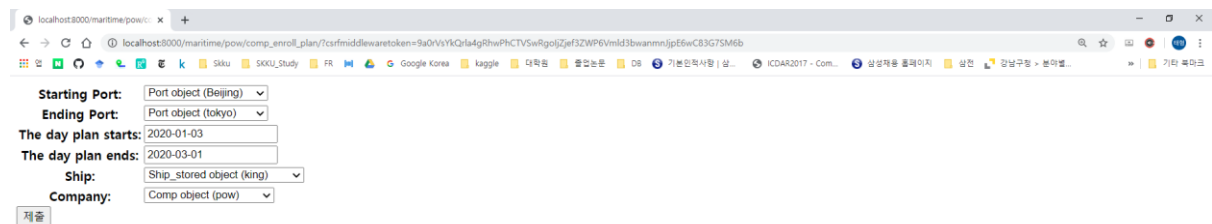


A web browser window showing a form titled 'name of new ship:'. The form includes a text input for 'name of ship' with the value 'GTH', a dropdown for 'Company of new ship:' with 'Comp object (pow)' selected, and a dropdown for 'Age of new eng:' with '10' selected. A '제출' (Submit) button is at the bottom.

Go to Company Main Page

Go

#Add Company's a new Plan



A web browser window showing a form titled 'Starting Port:'. The form includes dropdowns for 'Starting Port:' (Port object (Beijing)) and 'Ending Port:' (Port object (tokyo)), text inputs for 'The day plan starts:' (2020-01-03) and 'The day plan ends:' (2020-03-01), a dropdown for 'Ship:' (Ship_stored object (king)), and a dropdown for 'Company:' (Comp object (pow)). A '제출' (Submit) button is at the bottom.

Go to Company Main Page

Go

Add a new Engineer SQL문

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
eng_name	varchar(20)	NO		NULL	
gender	varchar(10)	NO		NULL	
age	int	NO		NULL	
eng_comp_id	varchar(20)	NO	MUL	NULL	

```
INSERT INTO eng_temp(eng_name,gender,age, eng_comp_id) VALUES(data['eng_name'],
```

```
data['gender'], data['age'], data['eng_comp_id']));
```

Add a new Ship SQL문

Field	Type	Null	Key	Default	Extra
ship_name	varchar(20)	NO	PRI	NULL	
age	int	NO		NULL	
ship_comp_id	varchar(20)	NO	MUL	NULL	

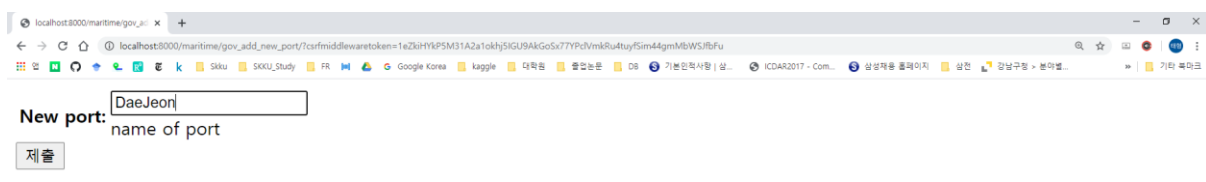
```
INSERT INTO ship_temp(ship_name,age,ship_comp_id) VALUES(data['ship_name'], data['age'], data['ship_comp_id']);
```

Add a new Plan SQL문

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
departure	date	NO		NULL	
arrival	date	NO		NULL	
plan_port1_id	varchar(30)	NO	MUL	NULL	
plan_port2_id	varchar(30)	NO	MUL	NULL	
plan_ship_id	varchar(20)	NO	MUL	NULL	
plan_comp_id	varchar(20)	NO	MUL	NULL	

```
INSERT INTO plan_temp(departure,arrival,plan_port1_id,plan_port2_id,plan_ship_id,plan_comp_id) VALUES(data['departure'],data['arrival'], data['plan_port1_id'], data['plan_port2_id'], data['plan_ship_id'], data['plan_comp_id']));
```

#Add a new Port (Government) (url pattern: gov_add/new_port)



Go to Gov Main Page

Ulr.py

```
path('gov_add_new_port/', views.gov_add_new_port, name = 'gov_add_new_port'),
```

views.py

```
def gov_add_new_port(request):
    if request.method == 'POST':
        form = AddNewPortForm(request.POST)
        if form.is_valid():
            new_item = form.save()
            new_item.save()
            return redirect(reverse('maritime:gov_main_page'))
            # return HttpResponseRedirect(reverse('maritime:gov_main_page'))
        form = AddNewPortForm()
    return render(request, 'gov/gov_add_new_port.html', {'form': form})
```

forms.py

```
class AddNewPortForm(ModelForm):
    class Meta:
        model = Port
        fields = ['port_name']
        labels = {
            'port_name': _('New port'),
        }
```

gov_add_new_port.html

```
<form action = "/maritime/gov_add_new_port/" method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <button type="submit">제출</button>
</form>
<hr>
<form action = "/maritime/gov_main_page/" method="get">
    {% csrf_token %}
    <h1>
        Go to Gov Main Page
    </h1>
    <button type="submit">Go</button>
</form>
```

해당 SQL문

Field	Type	Null	Key	Default	Extra
port_name	varchar(30)	NO	PRI	NULL	

INSERT INTO PORT VALUES(data['port_name']);

D. Permit or Reject submitted Data / Government

#Government Check submitted (url pattern:

localhost:8000/maritime/gov_check_submitted/7c9fmsddlewaretoken=a1b5WSKFAxRqYHkaxobshymWEFCDLv5gCanQwHH5IzrqBgfoidk5ScsdX1cyP

Mate

Name gender age company Check Deny

TH2 male 23 pow

Engineer

Name gender age Comp Check Deny

TH female 23 pow

Ship

Name age Comp Check Deny

GTH 10 pow

Plan

Departure Arrival Port1 Port2 Ship Comp Check Deny

Jan. 3, 2020 March 1, 2020 Beijing tokyo king pow

Go to Gov Main Page

Urls.py

```
path('gov_check_submitted/', views.gov_check_submitted,
name='gov_check_submitted'),
```

views.py

```
def gov_check_submitted(request):
    all_mates = Mate_temp.objects.all()
    all_engs = Eng_temp.objects.all()
    all_ships = Ship_temp.objects.all()
    all_plans = Plan_temp.objects.all()
    return render(request, 'gov/gov_check_submitted.html', {'all_mates':
all_mates, 'all_engs': all_engs, 'all_ships': all_ships, 'all_plans': all_plans} )
```

해당 SQL 문

```
SELECT * FROM mate_temp;
```

```
SELECT * FROM eng_ temp;
```

```
SELECT * FROM ship_ temp;
```

```
SELECT * FROM plan_ temp;
```


Gov_check_submitted.html

```
<!-- mates -->
{% if not all_mates %}
<h1>
    Mate Empty
</h1>
{% else %}
<h1>
    Mate
</h1>

<table class="table_mates_rejected">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>gender</B></td>
        <td><B>age</B></td>
        <td><B>company</B></td>
        <td><B>Check</B></td>
        <td><B>Deny</B></td>
    </tr>
    </thead>
    <tbody>
    {% for mate in all_mates %}
    <tr>
        <td> {{ mate.mate_name }}</td>
        <td> {{ mate.gender }} </td>
        <td> {{ mate.age }} </td>
        <td> {{ mate.mate_comp.comp_name }} </td>
        <td>
            <form action = "/maritime/{{mate.id}}/gov_checking_submitted_mate/"
method="post">
                {% csrf_token %}
                <button name="choice" value="{{mate.id}}">선택</button>
            </form>
        </td>
        <td>
            <form action = "/maritime/{{mate.id}}/gov_deny_submitted_mate/"
method="post">
                {% csrf_token %}
                <button name="choice" value="{{mate.id}}">선택</button>
            </form>
        </td>
    </tr>
    {% endfor %}
    </tbody>
</table>
{% endif %}
<!-- engineers -->
{% if not all_engs %}
<h1>
    Engineer Empty
</h1>
{% else %}
<h1>
    Engineer
```

```

</h1>

<table class="table_engineers_rejected">
  <thead>
    <tr>
      <td><B>Name</B></td>
      <td><B>gender</B></td>
      <td><B>age</B></td>
      <td><B>Comp</B></td>
      <td><B>Check</B></td>
      <td><B>Deny</B></td>
    </tr>
  </thead>
  <tbody>
    {% for eng in all_engs %}
    <tr>
      <td> {{ eng.eng_name }}</td>
      <td> {{ eng.gender }} </td>
      <td> {{ eng.age }} </td>
      <td> {{ eng.eng_comp.comp_name }} </td>
      <td>
        <form action = "/maritime/{{eng.id}}/gov_checking_submitted_eng/"
method="post">
          {% csrf_token %}
          <button name="choice" value="{{eng.id}}">선택</button>
        </form>
      </td>
      <td>
        <form action = "/maritime/{{eng.id}}/gov_deny_submitted_eng/"
method="post">
          {% csrf_token %}
          <button name="choice" value="{{eng.id}}">선택</button>
        </form>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
{% endif %}

<!-- ships -->
{% if not all_ships %}
<h1>
  Ships Empty
</h1>
{% else %}
<h1>
  Ship
</h1>

<table class="table_ships_submitted">
  <thead>
    <tr>
      <td><B>Name</B></td>
      <td><B>age</B></td>
      <td><B>Comp</B></td>
      <td><B>Check</B></td>
      <td><B>Deny</B></td>
    </tr>
  </thead>
  <tbody>
    {% for ship in all_ships %}
    <tr>
      <td> {{ ship.ship_name }}</td>
      <td> {{ ship.age }}</td>
      <td> {{ ship.ship_comp.comp_name }}</td>
      <td>
        <form action = "/maritime/{{ship.id}}/gov_checking_submitted_ship/"
method="post">
          {% csrf_token %}
          <button name="choice" value="{{ship.id}}">선택</button>
        </form>
      </td>
      <td>
        <form action = "/maritime/{{ship.id}}/gov_deny_submitted_ship/"
method="post">
          {% csrf_token %}
          <button name="choice" value="{{ship.id}}">선택</button>
        </form>
      </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
{% endif %}

```

```

</tr>
</thead>
<tbody>
{% for ship in all_ships %}
<tr>
<td> {{ ship.ship_name }}</td>
<td> {{ ship.age }} </td>
<td> {{ ship.ship_comp.comp_name }} </td>
<td>
<form action =
"/maritime/{{ ship.ship_name }}/gov_checking_submitted_ship/" method="post">
{% csrf_token %}
<button name="choice" >선택</button>
</form>
</td>
<td>
<form action =
"/maritime/{{ ship.ship_name }}/gov_deny_submitted_ship/" method="post">
{% csrf_token %}
<button name="choice" >선택</button>
</form>
</td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}

<!-- plans -->
{% if not all_plans %}
<h1>
Plan Empty
</h1>
{% else %}
<h1>
Plan
</h1>

<table class="table_ships_submitted">
<thead>
<tr>
<td><B>Departure</B></td>
<td><B>Arrival</B></td>
<td><B>Port1</B></td>
<td><B>Port2</B></td>
<td><B>Ship</B></td>
<td><B>Comp</B></td>
<td><B>Check</B></td>
<td><B>Deny</B></td>
</tr>
</thead>
<tbody>
{% for plan in all_plans %}
<tr>
<td> {{ plan.departure }}</td>
<td> {{ plan.arrival }}</td>
<td> {{ plan.plan_port1.port_name }}</td>
<td> {{ plan.plan_port2.port_name }}</td>

```

```

        <td> {{ plan.plan_ship.ship_name }}</td>
        <td> {{ plan.plan_comp.comp_name }}</td>
        <td>
            <form action = "/maritime/{{ plan.id }}/gov_checking_submitted_plan/"
method="post">
                {% csrf_token %}
                <button name="choice" >선택</button>
            </form>
        </td>
        <td>
            <form action = "/maritime/{{ plan.id }}/gov_deny_submitted_plan/"
method="post">
                {% csrf_token %}
                <button name="choice" >선택</button>
            </form>
        </td>
    </tr>
    {% endfor %}
</tbody>
</table>
{% endif %}

```

```

<hr>
<form action = "/maritime/gov_main_page/" method="get">
    {% csrf_token %}
    <h1>
        Go to Gov Main Page
    </h1>
    <button type="submit">Go</button>
</form>
<hr>

```

구현 방식이 동일하므로, mate의 경우로 한정해서 작성하겠습니다.

permit mate_temp

Urls.py

```

path('<int:mate_id>/gov_checking_submitted_mate/',
views.gov_checking_submitted_mate,name='gov_checking_submitted_mate'),

```

views.py

```

def gov_checking_submitted_mate(request,mate_id):
    mate = Mate_temp.objects.filter(id=mate_id).first()
    new_mate = Mate_stored(mate_name=mate.mate_name, mate_comp=mate.mate_comp,
gender=mate.gender, age=mate.age)
    new_mate.save()
    mate.delete()
    return HttpResponseRedirect(reverse('maritime:gov_check_submitted'))

```

```
mysql> desc maritime_mate_stored;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
mate_name	varchar(20)	NO		NULL	
gender	varchar(10)	NO		NULL	
age	int	NO		NULL	
mate_comp_id	varchar(20)	NO	MUL	NULL	

해당 SQL 문 ([advanced](#))

```
INSERT INTO mate_stored (mate_name, gender, age, mate_comp_id)
```

```
SELECT mate_name,gender,age,mate_comp_id FROM mate_temp WHERE id=mate_id;
```

```
DELETE FROM mate_temp WHERE id=mate_id;
```

reject mate_temp

Urls.py

```
path('<int:mate_id>/gov_deny_submitted_mate/',
views.gov_deny_submitted_mate,name='gov_deny_submitted_mate'),
```

views.py

```
def gov_deny_submitted_mate(request,mate_id):
    mate = Mate_temp.objects.filter(id=mate_id).first()
    new_mate = Mate_rejected(mate_name=mate.mate_name, mate_comp=mate.mate_comp,
gender=mate.gender, age=mate.age)
    new_mate.save()
    mate.delete()
    return HttpResponseRedirect(reverse('maritime:gov_check_submitted'))
```

해당 SQL 문 ([advanced](#))

```
mysql> desc maritime_mate_rejected;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
mate_name	varchar(20)	NO		NULL	
gender	varchar(10)	NO		NULL	
age	int	NO		NULL	
mate_comp_id	varchar(20)	NO	MUL	NULL	

```
INSERT INTO mate_rejected (mate_name, gender, age, mate_comp_id)
```

```
SELECT mate_name,gender,age,mate_comp_id FROM mate_temp WHERE id=mate_id;
```

```
DELETE FROM mate_temp WHERE id=mate_id;
```

E. Check Rejected Data or Modify Data / Company

#Check Rejected (url pattern : comp_name/mate_id/mate_checking_rejected/)

(mate,eng,ship,plan 모두 동일한 방식으로 구현했으므로, mate에 대해서만 작성합니다.)

Company name : pow

Mate

Name gender age Check
JW MALE 22

Engineer

Name gender age Check
JW FEMALE 100

Ship

Name age Comp Check
DB/JW 200 pow

Plan

Departure Arrival Port1 Port2 Ship Comp Check Modify
Jan. 2, 2020 April 3, 2020 Polo tokyo king sm

Comp_check_rejected.html

```
<h1>
    Company name : {{ comp.comp_name }}
</h1>
<!-- mates -->
{% if not all_mates %}
<h1>
    Mate Empty
</h1>
{% else %}
<h1>
    Mate
</h1>

<table class="table_mates_rejected">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>gender</B></td>
        <td><B>age</B></td>
        <td><B>Check</B></td>
    </tr>
    </thead>
    <tbody>
    {% for mate in all_mates %}
    <tr>
        <td> {{ mate.mate_name }}</td>
        <td> {{ mate.gender }}</td>
        <td> {{ mate.age }}</td>
        <td>
            <form action =
```

```

"/maritime/{{comp.comp_name}}/{{mate.id}}/mate_checking_rejected/" method="post">
    {% csrf_token %}
    <button name="choice" value="{{mate.id}}">선택</button>
</form>
</td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}
<!-- engineers -->
{% if not all_engs %}
<h1>
    Engineer Empty
</h1>
{% else %}
<h1>
    Engineer
</h1>

<table class="table_engineers_rejected">
    <thead>
    <tr>
        <td><B>Name</B></td>
        <td><B>gender</B></td>
        <td><B>age</B></td>
        <td><B>Check</B></td>
    </tr>
    </thead>
    <tbody>
    {% for eng in all_engs %}
    <tr>
        <td> {{ eng.eng_name }}</td>
        <td> {{ eng.gender }} </td>
        <td> {{ eng.age }} </td>
        <td>
            <form action =
"/maritime/{{comp.comp_name}}/{{eng.id}}/eng_checking_rejected/" method="post">
                {% csrf_token %}
                <button name="choice" value="{{eng.id}}">선택</button>
            </form>
        </td>
    </tr>
    {% endfor %}
    </tbody>
</table>
{% endif %}
<!-- ships -->
{% if not all_ships %}
<h1>
    Ships Empty
</h1>
{% else %}
<h1>
    Ship
</h1>

<table class="table_ships_submitted">

```

```

<thead>
<tr>
    <td><B>Name</B></td>
    <td><B>age</B></td>
    <td><B>Comp</B></td>
    <td><B>Check</B></td>
</tr>
</thead>
<tbody>
{% for ship in all_ships %}
<tr>
    <td> {{ ship.ship_name }}</td>
    <td> {{ ship.age }} </td>
    <td> {{ ship.ship_comp.comp_name }} </td>
    <td>
        <form action =
"/maritime/{{ comp.comp_name }}/{{ ship.ship_name }}/ship_checking_rejected/"
method="post">
            {% csrf_token %}
            <button name="choice" >선택</button>
        </form>
    </td>
</tr>
{% endfor %}
</tbody>
</table>
{% endif %}
<!-- plans -->
{% if not all_plans %}
<h1>
    Plan Empty
</h1>
{% else %}
<h1>
    Plan
</h1>
<table class="table_ships_submitted">
<thead>
<tr>
    <td><B>Departure</B></td>
    <td><B>Arrival</B></td>
    <td><B>Port1</B></td>
    <td><B>Port2</B></td>
    <td><B>Ship</B></td>
    <td><B>Comp</B></td>
    <td><B>Check</B></td>
    <td><B>Modify</B></td>
</tr>
</thead>
<tbody>
{% for plan in all_plans %}
<tr>
    <td> {{ plan.departure }}</td>
    <td> {{ plan.arrival }}</td>
    <td> {{ plan.plan_port1.port_name }}</td>
    <td> {{ plan.plan_port2.port_name }}</td>
    <td> {{ plan.plan_ship.ship_name }}</td>

```



```

        <td> {{ plan.plan_comp.comp_name }}</td>
        <td>
            <form action =
"/maritime/{{ comp.comp_name }}/{{ plan.id }}/plan_checking_rejected/"
method="post">
                {% csrf_token %}
                <button name="choice" >선택</button>
            </form>
        </td>
        <td>
            <form action =
"/maritime/{{ comp.comp_name }}/{{ plan.id }}/modify_plan/" method="get">
                {% csrf_token %}
                <button name="choice" >선택</button>
            </form>
        </td>
    </tr>
    {% endfor %}
</tbody>
</table>
{% endif %}
<hr>
<form action = "/maritime/{{ comp.comp_name }}/comp_main/" method="get">
    {% csrf_token %}
    <h1>
        Go to Company Main Page
    </h1>
    <button type="submit">Go</button>
</form>

```

Urls.py

```

path('<str:comp_name>/<int:mater_id>/mater_checking_rejected/',
views.mater_checking_rejected, name='mater_checking_rejected'),

```

views.py

```

def mater_checking_rejected(request,comp_name, mater_id):
    Mater_rejected.objects.filter(id=mater_id).delete()
    return
HttpResponseRedirect(reverse('maritime:comp_check_rejected',args=(comp_name,)))

```

해당 SQL문

```

mysql> desc maritime_mater_rejected;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| mater_name | varchar(20)   | NO   |     | NULL    |                 |
| gender     | varchar(10)   | NO   |     | NULL    |                 |
| age       | int           | NO   |     | NULL    |                 |
| mater_comp_id | varchar(20) | NO   | MUL | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

DELETE FROM mate_rejected where id =mate_id;

#Modify Rejected (url pattern : comp_name/plan_id/modify_plan/)

Urls.py

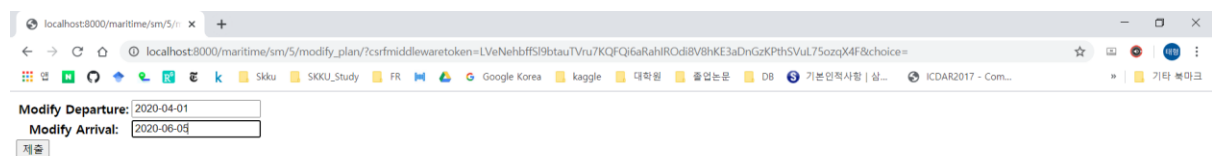
```
path('<str:comp_name>/<int:plan_id>/modify_plan/',  
views.modify_plan,name='modify_plan'),
```

views.py

```
def modify_plan(request,comp_name,plan_id):  
    if request.method == 'POST':  
        form = ModifyPlanForm(request.POST)  
        if form.is_valid():  
            plan_rejected = Plan_rejected.objects.get(id=plan_id)  
            data = form.cleaned_data  
            plan_rejected.departure=data['departure']  
            plan_rejected.arrival=data['arrival']  
            new_plan = Plan_temp(plan_port1=plan_rejected.plan_port1,  
plan_port2=plan_rejected.plan_port2, departure=plan_rejected.departure,  
                                arrival=plan_rejected.arrival,  
plan_ship=plan_rejected.plan_ship, plan_comp=plan_rejected.plan_comp)  
            new_plan.save()  
            plan_rejected.delete()  
            return HttpResponseRedirect(reverse('maritime:comp_main',  
args=(comp_name,)))  
        form = ModifyPlanForm()  
        return render(request,'comp/comp_modify_plan.html', {'form': form,  
'comp_name': comp_name, 'plan_id': plan_id})
```

forms.py

```
class ModifyPlanForm(ModelForm):  
    class Meta:  
        model = Modify_Plan  
        fields = ['departure', 'arrival']  
        labels = {  
            'departure': _('Modify Departure'),  
            'arrival': _('Modify Arrival')  
        }  
}
```



Go to Company Main Page

Comp_modify_plan.html

```

<form action = "/maritime/{{ comp_name }}/{{ plan_id }}/modify_plan/"
method="post">
    {% csrf_token %}
    <table>
        {{ form.as_table }}
    </table>
    <button type="submit">제출</button>
</form>
<hr>
<form action = "/maritime/{{ comp_name }}/comp_main/" method="get">
    {% csrf_token %}
    <h1>
        Go to Company Main Page
    </h1>
    <button type="submit">Go</button>
</form>

```

해당 SQL 문 (advanced)

```

mysql> desc maritime_modify_plan;
+-----+-----+-----+-----+-----+-----+
| Field      | Type | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int  | NO   | PRI | NULL    | auto_increment |
| departure  | date | NO   |     | NULL    |                 |
| arrival    | date | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc maritime_plan_rejected;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int       | NO   | PRI | NULL    | auto_increment |
| departure  | date      | NO   |     | NULL    |                 |
| arrival    | date      | NO   |     | NULL    |                 |
| plan_comp_id | varchar(20) | NO   | MUL | NULL    |                 |
| plan_port1_id | varchar(30) | NO   | MUL | NULL    |                 |
| plan_port2_id | varchar(30) | NO   | MUL | NULL    |                 |
| plan_ship_id | varchar(20) | NO   | MUL | NULL    |                 |
+-----+-----+-----+-----+-----+-----+

```

```

UPDATE plan_rejected SET departure=data['departure'], arrival=data['arriaval'] WHERE
id=plan_id;

```

```

INSERT INTO plan_temp (departure, arrival, plan_comp_id, plan_port1_id, plan_port2_id,
plan_ship_id) SELECT departure, arrival, plan_comp_id, plan_port1_id, plan_port2_id,
plan_ship_id FROM plan_rejected WHERE id=plan_id;

```

```

DELET FROM maritime_plan_rejected wher id = plan_id;

```