

## ■ 게시판

### ● 필요 기술 / 라이브러리

- HTML Design
- Database
- Model
  - class User / Article
- Django View
- Django Template

## ■ 게시판 - 작성 (Create)

### ● templates/write.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <form method="post" action="/article/write/">
      {% csrf_token %}
      <div class="form-group">
        <label for="title">Title:</label>
        <input type="text" class="form-control" name="title" id="title">
      </div>
      <div class="form-group">
        <label for="content">Content:</label>
        <textarea class="form-control" rows="5" name="content" id="content"></textarea>
      </div>
      <button type="submit" id="complete" class="btn btn-primary btn-block">
        글쓰기
      </button>
    </form>
  </div>
</div>

{% endblock %}
```

Title:

Content:

글쓰기

## ■ 게시판 - 작성 (Create)

### ● templates/write\_success.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>글 작성이 완료되었습니다.</h1>
  </div>
</div>

{% endblock %}
```

### ● templates/write\_fail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>로그인 후 사용해주세요.</h1>
  </div>
</div>

{% endblock %}
```






## ■ 게시판 - 작성 (Create)

### ● article/models.py

```
class Article(models.Model):  
    title = models.CharField(max_length=100)  
    content = models.CharField(max_length=1000)  
    user = models.ForeignKey(User, on_delete=models.CASCADE)
```

### ● migration

- python manage.py makemigrations
- python manage.py migrate

 article_article		CREATE TABLE "article_article" ("id" integer NOT NULL PRIMARY KEY AUTOINCR
 id	integer	"id" integer NOT NULL PRIMARY KEY AUTOINCR
 title	varchar(100)	"title" varchar(100) NOT NULL
 content	varchar(1000)	"content" varchar(1000) NOT NULL
 user_id	integer	"user_id" integer NOT NULL

## ■ 게시판 - 작성 (Create)

### ● board/views.py

```
def write(request):
    if request.method == 'POST':
        title = request.POST.get('title')
        content = request.POST.get('content')

    try:
        email = request.session['email']

        # select * from user where email = ?
        user = User.objects.get(email=email)

        # insert into article (title, content, user_id) values (?, ?, ?)
        article = Article(title=title, content=content, user=user)
        article.save()
        return render(request, 'write_success.html')
    except:
        return render(request, 'write_fail.html')

return render(request, 'write.html')
```

#### 세션 정보가 없는 경우 예외 발생

KeyError at /article/write/

'email'

Request Method: POST

Request URL: http://localhost:8000/article/write/

Django Version: 2.2.5

Exception Type: KeyError

Exception Value: 'email'

Exception Location: C:\Users\GGoreb\Miniconda3\lib\site-packages\django\contrib\sessions\backends\base.py in  
\_getitem\_, line 54

Python Executable: C:\Users\GGoreb\Miniconda3\python.exe

## ■ 게시판 - 작성 (Create)

### ● board/urls.py

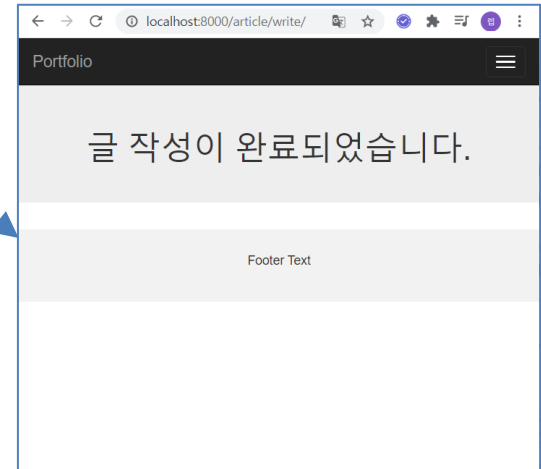
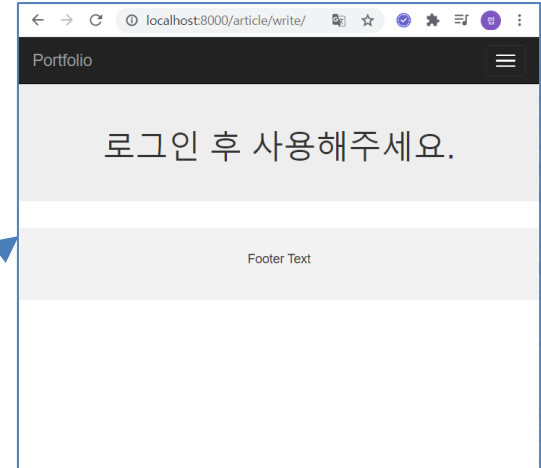
```
from django.contrib import admin
from django.urls import path
import board.views as views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', views.index),
    path('signup/', views.signup),
    path('signin/', views.signin),
    path('article/write/', views.write),
]
```

## ■ 게시판 - 작성 (Create)

● <http://localhost:8000/article/write/>

The screenshot shows a web browser window with the URL `localhost:8000/article/write/`. The page has a dark header with the word "Portfolio" and a hamburger menu icon. Below the header, there is a form with two sections: "Title:" with a text input field containing the placeholder "제목", and "Content:" with a larger text area containing the placeholder "내용". Below these fields is a blue button labeled "글쓰기". At the bottom of the page, there is a light gray footer area with the text "Footer Text".



id	title	content	user_id
필터	필터	필터	필터
1	제목	내용	1

## ■ 게시판 - 목록 조회 (Read)

### ● templates/list.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <table class="table table-hover">
      <thead>
        <tr>
          <th>번호</th><th>제목</th><th>작성자</th>
        </tr>
      </thead>
      <tbody>
        {% for article in article_list %}
        <tr>
          <td>{{ article.title }}</td>
          <td>{{ article.content }}</td>
          <td>{{ article.user.name }}</td>
        </tr>
        {% endfor %}
      </tbody>
    </table>
  </div>
</div>

{% endblock %}
```

번호

제목

작성자



## ■ 게시판 - 목록 조회 (Read)

### ● board/views.py

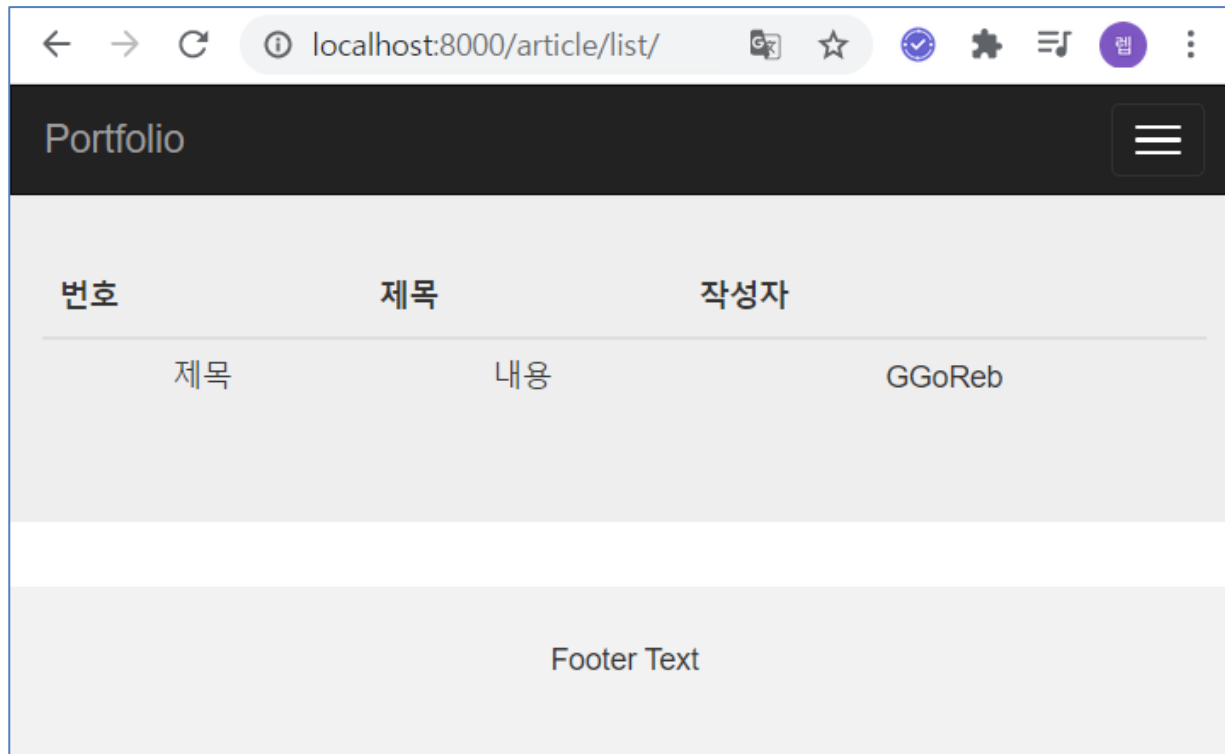
```
def list(request):  
    # select * from article order by id desc  
    article_list = Article.objects.order_by('-id')  
    context = {  
        'article_list' : article_list  
    }  
    return render(request, 'list.html', context)
```

### ● board/urls.py

```
...  
  
urlpatterns = [  
    ...  
    path('article/list/', views.list),  
]
```

## ■ 게시판 - 목록 조회 (Read)

● <http://localhost:8000/article/list/>



The screenshot shows a web browser window with the address bar displaying `localhost:8000/article/list/`. The page has a dark header with the word "Portfolio" and a hamburger menu icon. Below the header is a table with three columns: "번호" (Number), "제목" (Title), and "작성자" (Author). The first row of data shows "제목" (Title), "내용" (Content), and "GGoReb" (Author). Below the table is a white row, and at the bottom is a light gray footer area with the text "Footer Text".

번호	제목	작성자
제목	내용	GGoReb
Footer Text		

## ■ 게시판 - 상세 조회 (Read)

### ● templates/detail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="container" style='margin-top:30px;'>
  <div class="panel-group">
    <div class="panel panel-default">
      <div class="panel-heading" style='background-color: beige;'>
        {{ article.id }}번
      </div>
      <div class="panel-heading">{{ article.title }}</div>
      <div class="panel-body">{{ article.content }}</div>
      <div class="panel-footer">{{ article.user.name }}</div>
    </div>
  </div>
</div>
<div class="container text-right">
  <a href='' class="btn btn-default" id="list">목록</a>
  <a href='' class="btn btn-default" id="update">수정</a>
  <a href='' class="btn btn-default" id="delete">삭제</a>
</div>

{% endblock %}
```

번호

목록 수정 삭제

## ■ 게시판 - 상세 조회 (Read)

### ● board/views.py

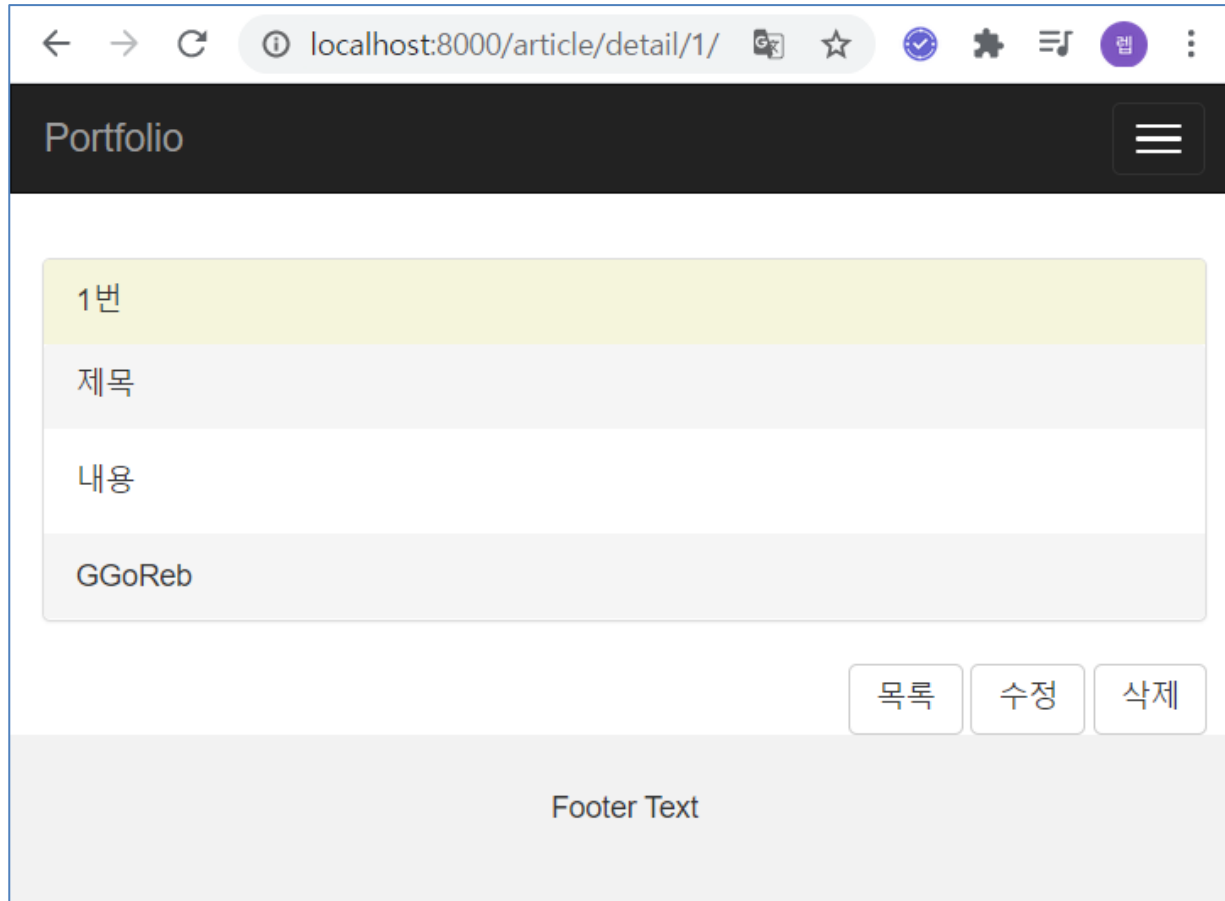
```
def detail(request, id):  
    # select * from article where id = ?  
    article = Article.objects.get(id=id)  
    context = {  
        'article' : article  
    }  
    return render(request, 'detail.html', context)
```

### ● board/urls.py

```
...  
  
urlpatterns = [  
    ...  
    path('article/detail/<int:id>/', views.detail),  
]
```

## ■ 게시판 - 상세 조회 (Read)

● <http://localhost:8000/article/detail/1/>



## ■ 게시판 - 수정 (Update)

### ● templates/update.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <form method="post" action="/article/update/{{ article.id }}/">
      {% csrf_token %}
      <div class="form-group">
        <label for="title">Title:</label>
        <input type="text" class="form-control" name="title"
          id="title" value="{{ article.title }}">
      </div>
      <div class="form-group">
        <label for="content">Content:</label>
        <textarea class="form-control" rows="5" name="content"
          id="content">{{ article.content }}</textarea>
      </div>
      <button type="submit" id="complete" class="btn btn-primary">
        수정
      </button>
    </form>
  </div>
</div>

{% endblock %}
```



Title:

Content:

글쓰기

## ■ 게시판 - 수정 (Update)

### ● templates/update\_success.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>글 수정이 완료되었습니다.</h1>
  </div>
</div>

{% endblock %}
```

### ● templates/update\_fail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>잘못된 요청입니다.</h1>
  </div>
</div>

{% endblock %}
```

## ■ 게시판 - 수정 (Update)

### ● board/views.py

```
def update(request, id):
    # select * from article where id = ?
    article = Article.objects.get(id=id)

    if request.method == 'POST':
        title = request.POST.get('title')
        content = request.POST.get('content')
        try:
            # update article set title = ?, content = ? where id = ?
            article.title = title
            article.content = content
            article.save()
            return render(request, 'update_success.html')
        except:
            return render(request, 'update_fail.html')
    context = {
        'article' : article
    }
    return render(request, 'update.html', context)
```



## ■ 게시판 - 수정 (Update)

### ● board/urls.py

```
...  
  
urlpatterns = [  
    ...  
    path('article/update/<int:id>/', views.update),  
]
```

## ■ 게시판 - 수정 (Update)

● <http://localhost:8000/article/update/1/>

Portfolio

Title:

제목

Content:

내용

수정

Footer Text



Portfolio

글 수정이 완료되었습니다.

Footer Text

id	title	content	user_id
필터	필터	필터	필터
1	제목수정	내용수정	1

## ■ 게시판 - 삭제 (Delete)

### ● templates/detail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="container" style='margin-top:30px;'>
  <div class="panel-group">
    <div class="panel panel-default">
      <div class="panel-heading" style='background-color: beige;'>
        {{ article.id }}번
      </div>
      <div class="panel-heading">{{ article.title }}</div>
      <div class="panel-body">{{ article.content }}</div>
      <div class="panel-footer">{{ article.user.name }}</div>
    </div>
  </div>
</div>
<div class="container text-right">
  <a href='' class="btn btn-default" id="list">목록</button>
  <a href='' class="btn btn-default" id="update">수정</button>
  <a href='/article/delete/{{ article.id }}/' class="btn btn-default" id="delete">
    삭제
  </a>
</div>

{% endblock %}
```

## ■ 게시판 - 삭제 (Delete)

### ● templates/delete\_success.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>글 삭제가 완료되었습니다.</h1>
  </div>
</div>

{% endblock %}
```

### ● templates/delete\_fail.html

```
{% extends 'base.html' %}
{% block content %}

<div class="jumbotron">
  <div class="container text-center">
    <h1>잘못된 요청입니다.</h1>
  </div>
</div>

{% endblock %}
```

## ■ 게시판 - 삭제 (Delete)

### ● board/views.py

```
def delete(request, id):
    try:
        # select * from article where id = ?
        article = Article.objects.get(id=id)
        article.delete()
        return render(request, 'delete_success.html')
    except:
        return render(request, 'delete_fail.html')
```

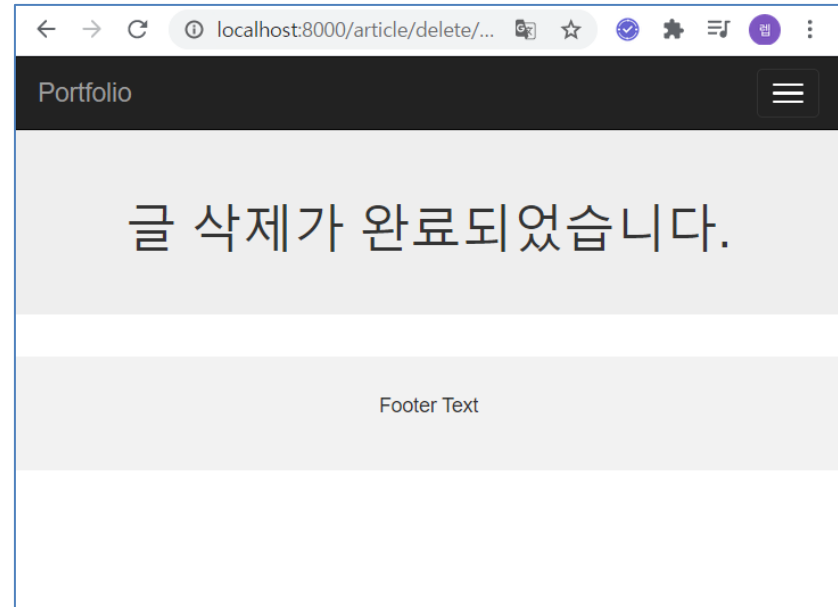
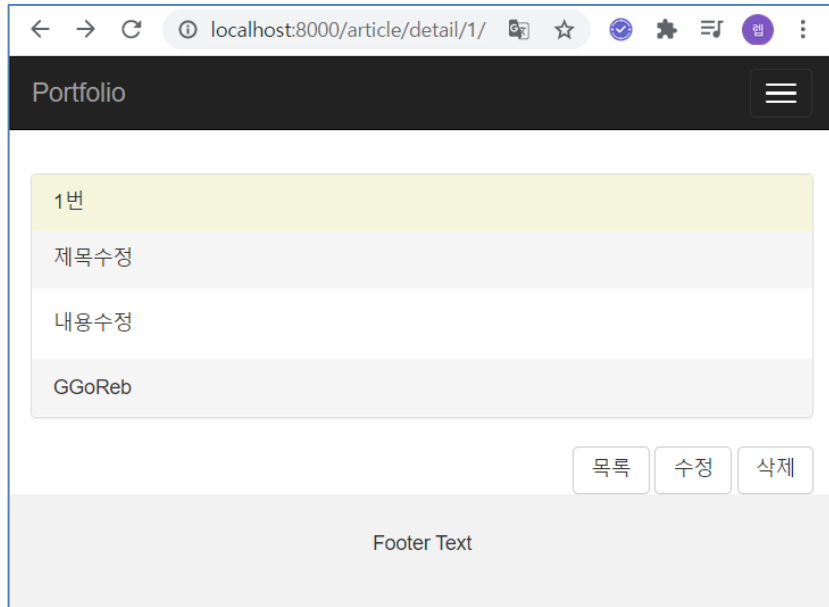
### ● board/urls.py

```
...

urlpatterns = [
    ...
    path('article/delete/<int:id>/', views.delete),
]
```

## ■ 게시판 - 삭제 (Delete)

● <http://localhost:8000/article/delete/1/>



id	title	content	user_id
필터	필터	필터	필터