

■ 회원가입

● 필요 기술 / 라이브러리

- HTML Design
- Database (SQLite)
- Model
 - class User
- Django View
- Django Template

app 생성

settings.py 설정

■ 회원가입

● App 생성

- python manage.py startapp article

● board/settings.py

```
...  
# Application definition  
INSTALLED_APPS = [  
    'article',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]  
...
```

▼ article

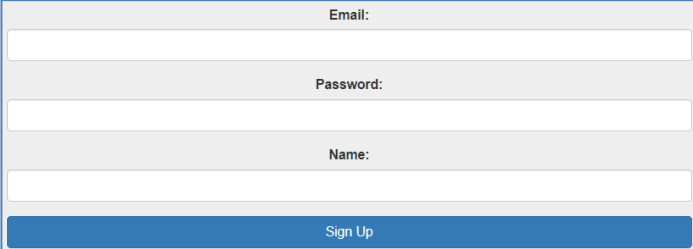
- > __pycache__
- > migrations
- 🔗 __init__.py
- 🔗 admin.py
- 🔗 apps.py
- 🔗 models.py
- 🔗 tests.py
- 🔗 views.py
- > board
- > templates
- ≡ db.sqlite3

■ 회원가입

● templates/signup.html

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

<div class="jumbotron">
  <div class="container text-center">
    <form method="post" action="/signup/">
      {% csrf_token %}
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="text" class="form-control" id="email" name="email">
      </div>
      <div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control" id="pwd" name="pwd">
      </div>
      <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" class="form-control" id="name" name="name">
      </div>
      <button class="btn btn-primary btn-block" id="signup">Sign Up</button>
    </form>
  </div>
</div>
{% endblock %}
```

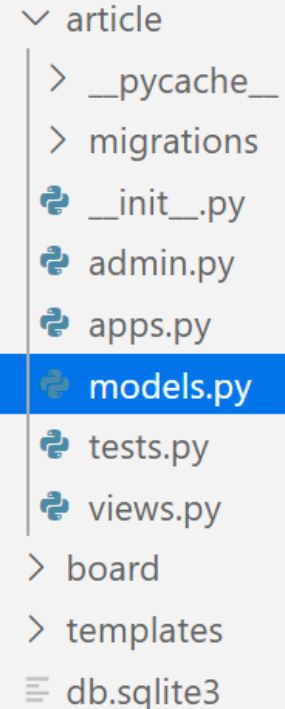


■ 회원가입

● article/models.py

```
from django.db import models
```

```
class User(models.Model):  
    email = models.CharField(max_length=50)  
    pwd = models.CharField(max_length=100)  
    name = models.CharField(max_length=10)
```



```
▼ article  
  > __pycache__  
  > migrations  
  🔄 __init__.py  
  🔄 admin.py  
  🔄 apps.py  
  🔄 models.py  
  🔄 tests.py  
  🔄 views.py  
  > board  
  > templates  
  ≡ db.sqlite3
```

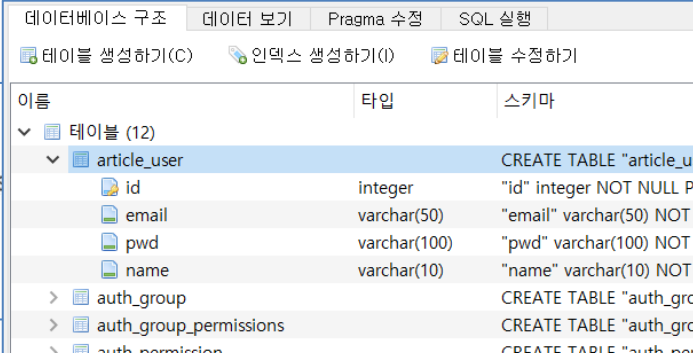
● migration

- python manage.py makemigrations

```
PS F:\dev\python_django\board> python manage.py makemigrations  
Migrations for 'article':  
  article\migrations\0001_initial.py  
    - Create model User  
PS F:\dev\python_django\board> █
```

- python manage.py migrate

```
PS F:\dev\python_django\board> python manage.py migrate  
Operations to perform:  
  Apply all migrations: admin, article, auth, contenttypes, sessions  
Running migrations:  
  Applying article.0001_initial... OK  
PS F:\dev\python_django\board> █
```



이름	타입	스키마
▼ 테이블 (12)		
▼ article_user		CREATE TABLE "article_us
id	integer	"id" integer NOT NULL P
email	varchar(50)	"email" varchar(50) NOT
pwd	varchar(100)	"pwd" varchar(100) NOT
name	varchar(10)	"name" varchar(10) NOT
> auth_group		CREATE TABLE "auth_gro
> auth_group_permissions		CREATE TABLE "auth_gro
> auth_permission		CREATE TABLE "auth per

■ 회원가입

● board/views.py

```
from django.http import HttpResponseRedirect
from article.models import User

def signup(request):
    if request.method == 'POST':
        # 회원정보 저장
        email = request.POST.get('email')
        name = request.POST.get('name')
        pwd = request.POST.get('pwd')
        user = User(email=email, name=name, pwd=pwd)
        user.save()
        return HttpResponseRedirect('/index/')

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

■ 회원가입

● 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),
]
```

■ 회원가입

● <http://localhost:8000/signup/>

Portfolio

Email:

seorab@naver.com

Password:

.

Name:

GGoReb

Sign Up

Footer Text



Portfolio

My Portfolio

Some text that represents "Me"...

Some of my Work

Some text..

테이블(T): article_user

	id	email	pwd	name
필터	필터	필터	필터	필터
1	1	seorab@naver.com	1	GGoReb

■ 로그인

● 필요 기술 / 라이브러리

- HTML Design
- Database (SQLite)
- Model
 - class User
- Django View
- Django Template

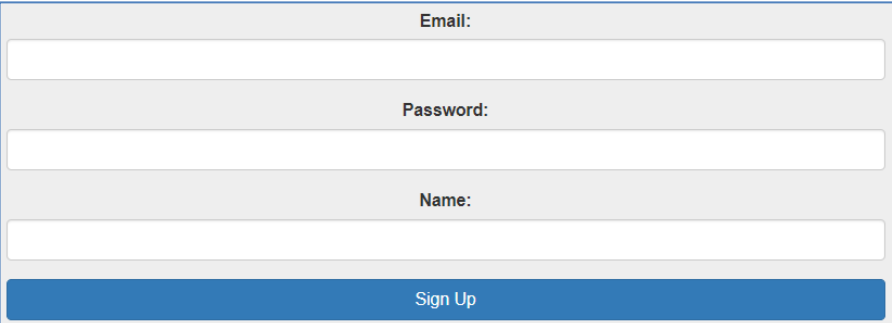
■ 로그인

● templates/signin.html

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

<div class="jumbotron">
  <div class="container text-center">
    <form method="post" action="/signin/">
      {% csrf_token %}
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="text" class="form-control" id="email" name="email">
      </div>
      <div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control" id="pwd" name="pwd">
      </div>
      <button class="btn btn-primary btn-block" id="signin">Sign in</button>
    </form>
  </div>
</div>

{% endblock %}
```



The image shows a visual rendering of the HTML template code provided. It features a sign-in form with three input fields: 'Email:', 'Password:', and 'Name:'. Each field is preceded by its label and followed by a text input box. The 'Email' and 'Password' fields are on the first two rows, and the 'Name' field is on the third row. Below these fields is a blue 'Sign Up' button. The form is styled with light gray headers for each input section and a blue footer for the button.

■ 로그인

● templates/signin_success.html

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

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

{% endblock %}
```

● templates/signin_fail.html

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

<div class="jumbotron">
  <div class="container text-center">
    <h1>이메일과 비밀번호를 확인해주세요.</h1>
  </div>
</div>

{% endblock %}
```

■ 로그인

● board/views.py

```
def signin(request):  
    if request.method == 'POST':  
        # 회원정보 조회  
        email = request.POST.get('email')  
        pwd = request.POST.get('pwd')  
  
        try:  
            # select * from user where email=? and pwd=?  
            user = User.objects.get(email=email, pwd=pwd)  
            return render(request, 'signin_success.html')  
        except:  
            return render(request, 'signin_fail.html')  
  
    return render(request, 'signin.html')
```

조회 정보가 없는 경우 예외 발생

DoesNotExist at /signin/

User matching query does not exist.

Request Method: POST

Request URL: http://localhost:8000/signin/

Django Version: 2.2.5

Exception Type: DoesNotExist

Exception Value: User matching query does not exist.

Exception Location: C:\Users\GGoReb\Miniconda3\lib\site-packages\django\db\models\query.py in get, line 408

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

■ 로그인

● board/urls.py

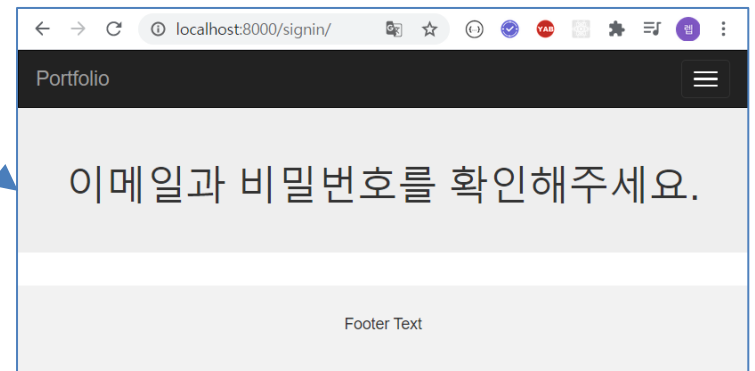
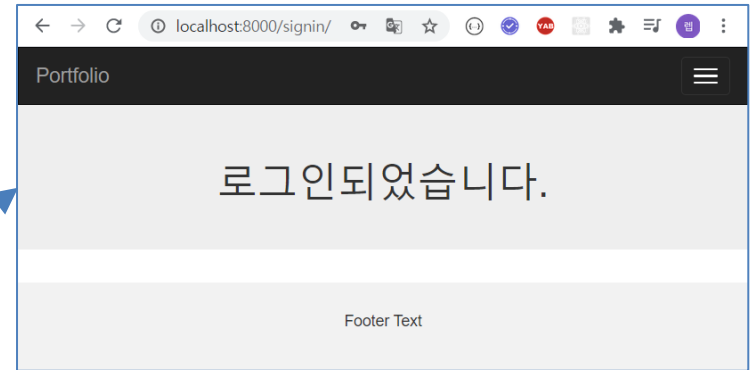
```
from django.contrib import admin
from django.urls import path
import board.views as views

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

■ 로그인

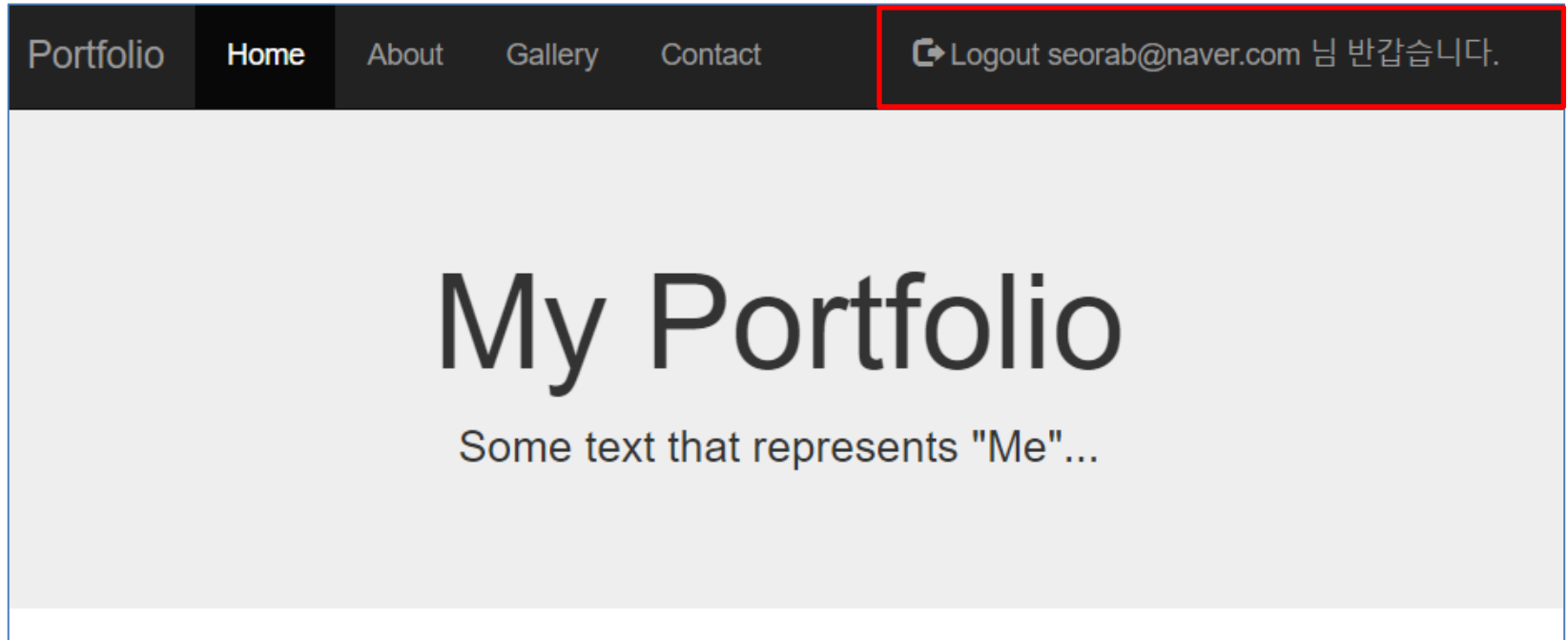
● <http://localhost:8000/signin/>

A screenshot of a web browser showing the login page at localhost:8000/signin/. The page has a dark header with the word "Portfolio" and a hamburger menu icon. Below the header, there is a form with two input fields: "Email:" containing "seorab@naver.com" and "Password:" containing a single dot. A blue "Sign in" button is positioned below the password field. At the bottom of the page, there is a light gray footer area with the text "Footer Text".



■ session

● 로그인 후 네비게이션 영역에 사용자 정보 출력



● 로그아웃 시 session에 저장된 정보 삭제

● Django Template session 표현식

- {{ request.session.세션저장명 }}

■ session

● board/views.py (로그인)

```
def signin(request):
    if request.method == 'POST':
        # 회원정보 조회
        email = request.POST.get('email')
        pwd = request.POST.get('pwd')

        try:
            # select * from user where email=? and pwd=?
            user = User.objects.get(email=email, pwd=pwd)
            request.session['email'] = email
            return render(request, 'signin_success.html')
        except:
            return render(request, 'signin_fail.html')

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

■ session

● templates/base.html

```
...
<nav class="navbar navbar-inverse">
  ...
  <div class="collapse navbar-collapse" id="myNavbar">
    ...
    <ul class="nav navbar-nav navbar-right">
      {% if request.session.email %}
      <li>
        <a href="/signout/">
          <span class="glyphicon glyphicon-log-out"></span> Logout
          {{ request.session.email }} 님 반갑습니다.
        </a>
      </li>
      {% else %}
      <li>
        <a href="/signin/">
          <span class="glyphicon glyphicon-log-in"></span> Login
        </a>
      </li>
      {% endif %}
    </ul>
  </div>
</div>
</nav>
...
```


■ session

● board/views.py (로그아웃)

```
def signout(request):  
    del request.session['email'] # 개별 삭제  
    request.session.flush() # 전체 삭제  
  
    return HttpResponseRedirect('/index/')
```

● 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('signout/', views.signout),  
]
```

Login

Logout seorab@naver.com 님 반갑습니다.