

```
$ django-admin startapp crud
$ cd crud
$ python manage.py startapp articles
```

settings.py

```
INSTALLED_APPS = [
    'articles',
]
TEMPLATES = [
    'DIRS': [BASE_DIR / 'templates',]]
```

templates폴더 생성 후 base.html 생성

```
<div class="container">
    {% if request.user.is_authenticated %}
        <h3>Hello, {{ user }}</h3>
        <form action="{% url 'accounts:logout' %}" method="POST">
            {% csrf_token %}
            <input type="submit" value="Logout">
        </form>
        <a href="{% url 'accounts:update' %}">회원정보수정</a>
        <form action="{% url 'accounts:delete' %}" method="POST">
            {% csrf_token %}
            <input type="submit" value="회원탈퇴">
        </form>
    {% else %}
        <a href="{% url 'accounts:login' %}">Login</a>
        <a href="{% url 'accounts:signup' %}">Signup</a>
    {% endif %}

    {% block content %}

    {% endblock content %}
</div>
```

crud/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('articles/', include('articles.urls')),
    path('accounts/', include('accounts.urls')),
]
```

articles/admin.py

```
from django.contrib import admin
from .models import Article

class ArticleAdmin(admin.ModelAdmin):
    list_display = ('pk', 'title', 'content', 'created_at', 'updated_at',)

admin.site.register(Article, ArticleAdmin)
```

articles/models.py

```
from django.db import models

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.title
```

모델 값 변경 후

```
$ python manage.py makemigrations
$ python manage.py migrate
$ python manage.py sqlmigrate articles 0001
$ python manage.py showmigrations
```

articles/forms.py

```
from django import forms
from .models import Article

class ArticleForm(forms.ModelForm):
    title = forms.CharField(
        label='제목',
        widget=forms.TextInput(
            attrs = {
                'class': 'my-title form-control',
                'placeholder': 'Enter the title',
                'maxlength': 10,
            }
        ),
    )
    content = forms.CharField(
        label = '내용',
        widget = forms.Textarea(
```

```

        attrs = {
            'class': 'my-content form-control',
            'placeholder': 'Enter the content',
            'rows': 5,
            'cols': 50,
        }
    ),
    error_messages = {
        'required': 'please enter your content'
    }
)
class Meta:
    model = Article
    fields = '__all__'

```

articles/urls.py

```

from django.urls import path
from . import views

app_name = 'articles'
urlpatterns = [
    path('', views.index, name='index'),
    path('create/', views.create, name='create'), # GET / POST
    path('<int:pk>/', views.detail, name='detail'),
    path('<int:pk>/delete/', views.delete, name='delete'),
    path('<int:pk>/update/', views.update, name='update'), # GET / POST
]

```

articles/views.py

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import render, redirect, get_object_or_404
from django.views.decorators.http import require_http_methods, require_POST,
require_safe
from .models import Article
from .forms import ArticleForm

# Create your views here.
@require_safe
def index(request):
    articles = Article.objects.order_by('-pk')
    context = {
        'articles': articles,
    }
    return render(request, 'articles/index.html', context)

@login_required
@require_http_methods(['GET', 'POST'])
def create(request):
    if request.method == 'POST':
        form = ArticleForm(request.POST)

```

```

        if form.is_valid():
            article = form.save()
            return redirect('articles:detail', article.pk)
    else:
        form = ArticleForm()
    context = {
        'form': form,
    }
    return render(request, 'articles/create.html', context)

@require_safe
def detail(request, pk):
    article = get_object_or_404(Article, pk=pk)
    context = {
        'article': article,
    }
    return render(request, 'articles/detail.html', context)

@require_POST
def delete(request, pk):
    if request.user.is_authenticated:
        article = get_object_or_404(Article, pk=pk)
        article.delete()
    return redirect('articles:index')

@login_required
@require_http_methods(['GET', 'POST'])
def update(request, pk):
    article = get_object_or_404(Article, pk=pk)
    if request.method == 'POST':
        form = ArticleForm(request.POST, instance=article)
        if form.is_valid():
            article = form.save()
            return redirect('articles:detail', article.pk)
    else:
        form = ArticleForm(instance=article)
    context = {
        'article': article,
        'form': form,
    }
    return render(request, 'articles/update.html', context)

```

templates/articles/index.html

```

{% extends 'base.html' %}

{% block content %}
<h1>Articles</h1>
{% if request.user.is_authenticated %}
<a href="{% url 'articles:create' %}">CREATE</a>
{% else %}

```

```

    <a href="{% url 'accounts:login' %}">[새 글을 작성하려면 로그인 하세요]</a>
{% endif %}
<hr>
{% for article in articles %}
    <p>글 번호: {{ article.pk }}</p>
    <p>글 제목: {{ article.title }}</p>
    <p>글 내용: {{ article.content }}</p>
    <a href="{% url 'articles:detail' article.pk %}">DETAIL</a>
    <hr>
{% endfor %}
{% endblock content %}

```

templates/articles/detail.html

```

{% extends 'base.html' %}

{% block content %}
    <h1>DETAIL</h1>
    <h3>{{ article.pk }}번째 글</h3>
    <hr>
    <p>제목 : {{ article.title }}</p>
    <p>내용 : {{ article.content }}</p>
    <p>작성 시각 : {{ article.created_at }}</p>
    <p>수정 시각 : {{ article.updated_at }}</p>
    <hr>
    <a href="{% url 'articles:update' article.pk %}">수정</a>
    <form action="{% url 'articles:delete' article.pk %}" method="POST">
        {% csrf_token %}
        <input type="submit" value="삭제">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}

```

templates/articles/create.html

```

{% extends 'base.html' %}

{% block content %}
    <h1>CREATE</h1>
    <hr>
    <form action="{% url 'articles:create' %}" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}

```

templates/articles/update.html

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

{% block content %}
<h1>UPDATE</h1>
<hr>
<form action="{% url 'articles:update' article.pk %}" method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit">
</form>
<a href="{% url 'articles:detail' article.pk %}">back</a>
{% endblock content %}
```

```
$ python startapp accounts
```

crud/settings.py

```
INSTALLED_APPS = [
    'articles',
    'accounts',
    'django_extensions',
]
```

accounts/forms.py

```
from django.contrib.auth.forms import UserChangeForm
from django.contrib.auth import get_user_model

class CustomUserChangeForm(UserChangeForm):
    class Meta:
        model = get_user_model() # User
        fields = ('email', 'first_name', 'last_name',)
```

accounts/urls.py

```

from django.urls import path
from . import views

app_name = 'accounts'
urlpatterns = [
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
    path('signup/', views.signup, name='signup'),
    path('delete/', views.delete, name='delete'),
    path('update/', views.update, name='update'),
    path('password/', views.change_password, name='change_password'),
]

```

accounts/views.py

```

from django.contrib.auth import login as auth_login
from django.contrib.auth import logout as auth_logout
from django.contrib.auth import update_session_auth_hash
from django.contrib.auth.decorators import login_required
from django.contrib.auth.forms import (
    AuthenticationForm,
    UserCreationForm,
    PasswordChangeForm,
)
from django.views.decorators.http import require_http_methods, require_POST
from django.shortcuts import render, redirect
from .forms import CustomUserChangeForm

# Create your views here.
@require_http_methods(['GET', 'POST'])
def login(request):
    if request.user.is_authenticated:
        return redirect('articles:index')

    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            # 로그인
            auth_login(request, form.get_user())
            return redirect(request.GET.get('next') or 'articles:index')
    else:
        form = AuthenticationForm()
        context = {
            'form': form,
        }
        return render(request, 'accounts/login.html', context)

@require_POST
def logout(request):
    if request.user.is_authenticated:
        auth_logout(request)
        return redirect('articles:index')

```

```

@require_http_methods(['GET', 'POST'])
def signup(request):
    if request.user.is_authenticated:
        return redirect('articles:index')

    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth_login(request, user)
            return redirect('articles:index')
    else:
        form = UserCreationForm()
    context = {
        'form': form,
    }
    return render(request, 'accounts/signup.html', context)

@require_POST
def delete(request):
    if request.user.is_authenticated:
        # 반드시 회원탈퇴 후 로그아웃 함수 호출
        request.user.delete()
        auth_logout(request)
    return redirect('articles:index')

@login_required
@require_http_methods(['GET', 'POST'])
def update(request):
    if request.method == 'POST':
        form = CustomUserChangeForm(request.POST, instance=request.user)
        if form.is_valid():
            form.save()
            return redirect('articles:index')
    else:
        form = CustomUserChangeForm(instance=request.user)
    context = {
        'form': form,
    }
    return render(request, 'accounts/update.html', context)

@login_required
@require_http_methods(['GET', 'POST'])
def change_password(request):
    if request.method == 'POST':
        form = PasswordChangeForm(request.user, request.POST)
        if form.is_valid():
            # user = form.save()
            # update_session_auth_hash(request, user)
            form.save()
            update_session_auth_hash(request, form.user)
            return redirect('articles:index')
    else:
        form = PasswordChangeForm(request.user)
    context = {

```



```
        'form': form,
    }
    return render(request, 'accounts/change_password.html', context)
```

accounts/templates/accounts/login.html

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

{% block content %}
    <h1>로그인</h1>
    <hr>
    <form action="" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}
```

accounts/templates/accounts/signup.html

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

{% block content %}
    <h1>회원가입</h1>
    <hr>
    <form action="{% url 'accounts:signup' %}" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}
```

accounts/templates/accounts/update.html

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

{% block content %}
    <h1>회원정보수정</h1>
    <hr>
    <form action="{% url 'accounts:update' %}" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}
```

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

{% block content %}
    <h1>비밀번호변경</h1>
    <hr>
    <form action="{% url 'accounts:change_password' %}" method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
    <a href="{% url 'articles:index' %}">back</a>
{% endblock content %}
```