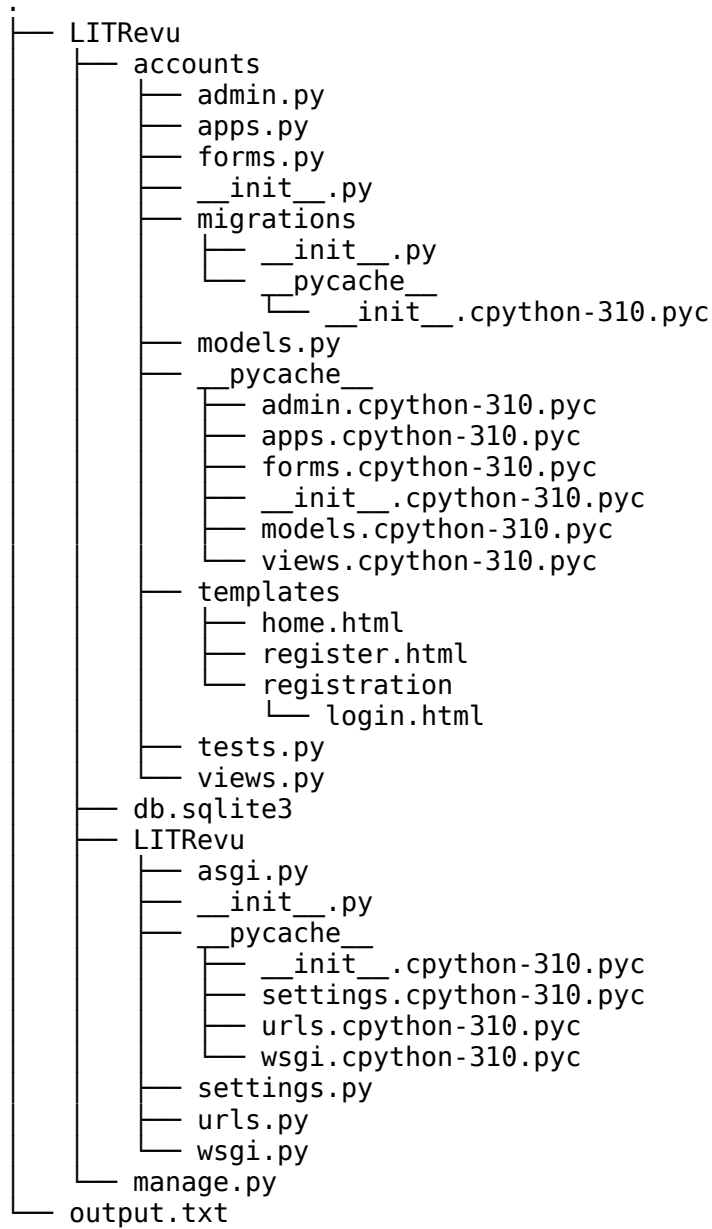


Structure du projet



9 directories, 30 files

#Fichier /Projet/accounts/admin.py

```
from django.contrib import admin
```

```
# Register your models here.
```

Fichier /Projet/accounts/apps.py

```
from django.apps import AppConfig
```

```
class AccountsConfig(AppConfig):  
    default_auto_field = 'django.db.models.BigAutoField'  
    name = 'accounts'
```

Fichier /Projet/accounts/forms.py

```
# accounts/forms.py
```

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm
```

```
class NewUserForm( UserCreationForm):
    email = forms.EmailField(required=True)
    first_name = forms.CharField(required=True)
    last_name = forms.CharField(required=True)
    birth_date = forms.DateField(required=True, input_formats=['%d/%m/%Y'])

    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'birth_date', 'email', 'password1', 'password2')

    def save(self, commit=True):
        user = super(NewUserForm, self).save(commit=False)
        user.email = self.cleaned_data['email']
        user.first_name = self.cleaned_data['first_name']
        user.last_name = self.cleaned_data['last_name']
        user.birth_date = self.cleaned_data['birth_date']
        if commit:
            user.save()
        return user
```

Fichier /Projet/accounts/models.py

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

```
# Create your models here.
```

Fichier /Projet/accounts/tests.py

```
from django.test import TestCase
```

```
# Create your tests here.
```

Fichier /Projet/accounts/views.py

```
from django.shortcuts import render, redirect
from .forms import NewUserForm # Importez le formulaire personnalisé
```

```
def home(request):
    return render(request, 'home.html')
```

accounts/views.py

```
def register(request):
    if request.method == 'POST':
        form = NewUserForm(request.POST) # Utilisez le formulaire personnalisé
        if form.is_valid():
            form.save()
            return redirect('login')
    else:
        form = NewUserForm() # Utilisez le formulaire personnalisé
    return render(request, 'register.html', {'form': form})
```

Fichier Projet/accounts/templates/home.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Accueil</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <h1>Bienvenue sur notre site !</h1>
  <a href="/login">Se connecter</a> ou <a href="/register">Créer un compte</a>
</body>
</html>
```


Fichier /Projet/accounts/templates/register.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Inscription</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <h1>Inscription</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class="btn btn-primary">Inscription</button>
  </form>
</body>
</html>
```

Fichier /Projet/accounts/templates/registration/login.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Connexion</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <h1>Connexion</h1>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Connexion</button>
  </form>
</body>
</html>
```

```
# Fichier /Projet/LITRevu/manage.py
```

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault("DJANGO_SETTINGS_MODULE", "LITRevu.settings")
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == "__main__":
    main()
```

```
# Fichier /Projet/LITRevu/LITRevu/urls.py
```

```
"""
```

```
URL configuration for LITRevu project.
```

```
The `urlpatterns` list routes URLs to views. For more information please see:
```

```
https://docs.djangoproject.com/en/4.2/topics/http/urls/
```

```
Examples:
```

```
Function views
```

```
1. Add an import:  from my_app import views
```

```
2. Add a URL to urlpatterns:  path('', views.home, name='home')
```

```
Class-based views
```

```
1. Add an import:  from other_app.views import Home
```

```
2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
```

```
Including another URLconf
```

```
1. Import the include() function: from django.urls import include, path
```

```
2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
```

```
"""
```

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from accounts.views import home, register
```

```
from django.contrib.auth import views as auth_views
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    path('home/', home, name='home'),
```

```
    path('login/', auth_views.LoginView.as_view(), name='login'),
```

```
    path('register/', register, name='register'),
```

```
]
```

Fichier /Projet/LITRevu/LITRevu/settings.py

```
"""
Django settings for LITRevu project.

Generated by 'django-admin startproject' using Django 4.2.4.

For more information on this file, see
https://docs.djangoproject.com/en/4.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.2/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = "django-insecure-465!9uvvi-yrucv__ytxw+)^c@5jcq-llk#6#=#-kn@opm=__m"

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "accounts",
]

MIDDLEWARE = [
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
```

```
"django.contrib.auth.middleware.AuthenticationMiddleware",
"django.contrib.messages.middleware.MessageMiddleware",
"django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "LITRevu.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [],
        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.debug",
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]

WSGI_APPLICATION = "LITRevu.wsgi.application"

# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator",
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
    },
]
```

```
{
    "NAME": "django.contrib.auth.password_validation.CommonPasswordValidator",
},
{
    "NAME": "django.contrib.auth.password_validation.NumericPasswordValidator",
},
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/
```

```
LANGUAGE_CODE = "fr"
```

```
TIME_ZONE = "Europe/Paris"
```

```
USE_I18N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/
```

```
STATIC_URL = "static/"
```

```
# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field
```

```
DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"
```

```
LOGIN_REDIRECT_URL = 'accounts/login.html'
```