

NB : Pour des raisons de sécurité, le projet ne peut pas être poussé directement sur GitHub tant qu'il contient la clé secrète Twilio, car Git bloque le push. »

```
PS C:\Users\HP\Desktop\projet> .\venv\Scripts\activate  
>> █
```

```
>>  
(venv) PS C:\Users\HP\Desktop\projet> cd myproject█
```

```
(venv) PS C:\Users\HP\Desktop\projet> cd myproject  
(venv) PS C:\Users\HP\Desktop\projet\myproject> pip install django█
```

```
et\myproject> python -m venv venv█
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> pip install python█
```

```
> django-admin startproject mon_projet█
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> python manage.py createsuperuser  
Username: nago
```

```
Error: That username is already taken.
```

```
Username: nago1
```

```
Email address: nago1998@gmail.com
```

```
Password: █
```

```
superuser created successfully.
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> django-admin startproject myproject█
```

```
superuser created successfully.
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> python manage.py startapp accounts█
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> python manage.py makemigrations accounts  
>> █
```

```
superuser created successfully.
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> pip install twilio
```

```
>> █
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> pip install --upgrade urllib3  
>> █
```

```
superuser created successfully.
```

```
(venv) PS C:\Users\HP\Desktop\projet\myproject> curl -v https://api.twilio.com
```

```
>> █
```

```

from twilio.rest import Client
from django.conf import settings
def send_sms(to, body):
    """
    Envoie un SMS via l'API Twilio.

    :param to: Numéro de téléphone du destinataire (format international, ex. '+221776774799')
    :param body: Contenu du message
    :return: SID du message ou exception en cas d'erreur
    """
    try:
        client = Client(settings.TWILIO_ACCOUNT_SID, settings.TWILIO_AUTH_TOKEN)
        message = client.messages.create(
            body=body,
            from_=settings.TWILIO_PHONE_NUMBER,
            to=to
        )
        return message.sid
    except Exception as e:
        raise Exception(f"Erreur lors de l'envoi du SMS : {e}")

```

```

def login_view(request):
    if request.method == "POST":
        username = request.POST.get('username')
        password = request.POST.get('password')

        # Authentifier l'utilisateur
        user = authenticate(request, username=username, password=password)

        if user is not None:
            # Générer un code de vérification aléatoire
            verification_code = random.randint(1000, 9999)
            request.session['verification_code'] = verification_code # Stocker le code en session
            request.session['user_id'] = user.id # Stocker l'ID utilisateur

            # Envoyer le code de vérification par SMS
            try:
                send_sms("+221776774799", f"Votre code de vérification est : {verification_code}")
            except Exception as e:
                return render(request, 'accounts/login.html', {
                    'error': f"Impossible d'envoyer le SMS. Veuillez vérifier la configuration : {e}"
                })

            # Rediriger vers la page de vérification
            return redirect('verify_code')
        else:
            return render(request, 'accounts/login.html', {'error': "Nom d'utilisateur ou mot de passe invalide."})

    return render(request, 'accounts/login.html')

```

```
from django.shortcuts import render, redirect
from django.contrib.auth import login
from django.contrib.auth.models import User

def verify_code_view(request):
    if request.method == "POST":
        entered_code = request.POST.get('code')
        verification_code = request.session.get('verification_code')
        user_id = request.session.get('user_id')

        if str(entered_code) == str(verification_code): # Comparer les codes
            user = User.objects.get(id=user_id) # Récupérer l'utilisateur
            login(request, user) # Connecter l'utilisateur
            return redirect('home') # Rediriger vers la page principale
        else:
            return render(request, 'accounts/verify_code.html', {'error': 'Code invalide.'})

    return render(request, 'accounts/verify_code.html')
```

```
1  [% load static %]
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Login </title>
8      <style>
9          body {
10              margin: 0;
11              font-family: Arial, sans-serif;
12              height: 100vh;
13              display: flex;
14              justify-content: center;
15              align-items: center;
16              background-color: #f0f2f5;
17          }
18          .main-container {
19              display: flex;
20              width: 90%;
21              max-width: 1200px;
22              height: 100%;
23              box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
24              border-radius: 10px;
25              overflow: hidden;
26              background-color: #fff;
27          }
28      .main-container img {
29          flex: 1;
30          object-fit: cover; /* L'image couvre la zone sans déformation */
31          height: 100%;
32          background-color: #fff;
33          transform: scale(1.3);
34      }
35      .form-container {
36          flex: 1;
37          display: flex;
38          flex-direction: column;
39          justify-content: center;
40          align-items: center;
41          padding: 10px;
42          background-color: #f9f9f9;
43          width: 400px; /* Réduit la largeur du conteneur */
44          height: 250px; /* Réduit la hauteur du conteneur */
45          margin-top: 200px; /* Déplace vers le bas */
46          margin-left: 45px; /* Déplace vers la droite */
47          border: 2px solid #ccc; /* Bordure de 2px grise */
48          border-radius: 10px; /* Coins arrondis */
49          box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Ombre douce pour un effet de fenêtre */
50          padding: 20px; /* Espace interne */
51      }
```

```
52 |     .logo {
53 |         font-size: 32px;
54 |         font-weight: bold;
55 |         color: black;
56 |         margin-bottom: 30px;
57 |
58 |     }
59 |     form {
60 |         width: 100%;
61 |         max-width: 350px;
62 |         display: flex;
63 |         flex-direction: column;
64 |
65 |     }
66 |     label {
67 |         font-size: 14px;
68 |         font-weight: bold;
69 |         margin-bottom: 5px;
70 |
71 |     }
72 |     input {
73 |         margin-bottom: 20px;
74 |         padding: 10px;
75 |         font-size: 16px;
76 |         border: 1px solid #ccc;
77 |         border-radius: 5px;
78 |
79 |     }
80 |     .button-container {
81 |         display: flex;
82 |         justify-content: space-between;
83 |
84 |     }
85 |     button {
86 |         padding: 10px 20px;
87 |         font-size: 16px;
88 |         border: none;
89 |         border-radius: 5px;
90 |         cursor: pointer;
91 |         width: 30%;
92 |
93 |     }
94 |     .btn-ok {
95 |         background-color: #4CAF50;
96 |         color: white;
97 |
98 |     }
99 |     .btn-cancel {
100 |         background-color: #f44336;
101 |         color: white;
102 |
103 |     }
104 |     .btn-emergency {
105 |         background-color: #ffa500;
106 |         color: white;
107 |
108 |     }
109 |     .footer-text {
110 |         margin-top: 20px;
111 |         font-size: 12px;
112 |         color: #666;
113 |         text-align: center;
114 |         width: 250px; /* Réduit la largeur du conteneur */
115 |         height: 350px; /* Réduit la hauteur du conteneur */
116 |
117 |     }
```

```

108     </style>
109 </head>
110 <body>
111     <div class="main-container">
112         <!-- Image dans le conteneur principal -->
113         
114
115
116
117         <!-- Conteneur pour la partie connexion -->
118         <div class="form-container">
119             <div class="logo">Neusoft</div>
120             <form action="{% url 'login' %}" method="post">
121                 {% csrf_token %}
122                 <label for="login">Login</label>
123                 <input type="text" id="username" name="username" placeholder="Enter your username" required>
124
125                 <label for="password">Password</label>
126                 <input type="password" id="password" name="password" placeholder="Enter your password" required>
127
128                 <div class="button-container">
129                     <button type="button" class="btn-cancel" onclick="window.location.href='/'>Cancel</button>
130                     <button type="button" class="btn-emergency" onclick="alert('Emergency action triggered!')>Emergency<b></b></button>
131                     <button type="submit" class="btn-ok">OK</button>
132                 </div>
133             </form>
134             <div class="footer-text">NeuViz 64 In 1.0.7.4164370</div>
135         </div>
136
137     </div>
138 </body>
139 </html>

```

myproject > accounts > templates > accounts > verify_code.html > ...

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Vérification du code</title>
5  </head>
6  <body>
7      <h1>Vérification du code</h1>
8      <form method="POST">
9          {% csrf_token %}
10         <label>Code de vérification :</label>
11         <input type="text" name="code" required>
12         <button type="submit">Vérifier</button>
13     </form>
14     {% if error %}
15         <p style="color: red;">{{ error }}</p>
16     {% endif %}
17 </body>
18 </html>
19

```

```
myproject > accounts > urls.py > ...
1  from django.urls import path
2  from . import views
3  from .views import create_user_view
4
5  urlpatterns = [
6      path('create_user/', create_user_view, name='create_user'),
7      # Ajoutez d'autres URL ici
8
9      path('login/', views.login_view, name='login'),
10     path('verify-code/', views.verify_code_view, name='verify_code'),
11 ]
12
```

```
myproject > myproject > urls.py > ...
1  """
2  URL configuration for myproject project.
3
4  The `urlpatterns` list routes URLs to views. For more information please see:
5      https://docs.djangoproject.com/en/5.1/topics/http/urls/
6  Examples:
7  Function views
8      1. Add an import: from my_app import views
9         2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12         2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15         2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path
19 from django.urls import path, include
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('accounts/', include('accounts.urls')), # Inclure les URLs de l'application
24
25 ]
26
27 # Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.1/howto/static-files/
28
STATIC_URL = '/static/'
29
STATICFILES_DIRS = [
30     BASE_DIR / 'static', # Le répertoire 'static' est correctement référencé
31 ]
32
AUTH_USER_MODEL = 'accounts.CustomUser'
```

```
# Database
# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

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

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'accounts/templates'],
        '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',
            ],
        },
    },
]

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

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-gihg^m4=r+_u7(cx8xjti+5ln&e%7*tfgr5b)v-wkc'

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

ALLOWED_HOSTS = ['localhost', '127.0.0.1']
```

