Absolutely! You've already done a great job setting up the foundational structure for your Django project. Let's turn what you've built into a **clean, beginner-friendly documentation** that explains:

* What was done ✅
* Why it was done 🧠
* Where each part lives 🗂️
* And how everything connects 🔗

**📝 Django Project: schoolapp**

**📌 Goal: Implement login and registration functionality using the users app.**

**🧱 PROJECT STRUCTURE OVERVIEW**

schoolapp/

├── manage.py

├── schoolapp/ # Main project settings & config

│ ├── \_\_init\_\_.py

│ ├── settings.py # Django settings (important!)

│ ├── urls.py # Root URL dispatcher

│ └── wsgi.py # WSGI entrypoint (used in production)

│

├── users/ # App handling user login/register

│ ├── \_\_init\_\_.py

│ ├── admin.py

│ ├── apps.py

│ ├── forms.py # Custom forms (UserCreation)

│ ├── models.py # Models (not used yet)

│ ├── urls.py # App-level URL dispatcher

│ ├── views.py # Business logic for login/register

│ ├── migrations/

│ └── templates/

│ └── users/

│ ├── base.html

│ ├── login.html

│ └── register.html

└── templates/ # Optional global templates folder (added in settings)

**🚀 STEP-BY-STEP DOCUMENTATION**

**1️⃣ Project Initialization**

Create the main project:

django-admin startproject schoolapp

cd schoolapp

Create the users app:

python manage.py startapp users

Register users in INSTALLED\_APPS inside schoolapp/settings.py:

INSTALLED\_APPS = [

...

'users', # Our custom app

]

**2️⃣ Templates Setup**

**Why?**

Django needs to know **where to look for HTML files**. By default, it only searches inside each app's templates/ folder (thanks to APP\_DIRS=True). But we also added a global templates/ directory for shared templates.

**Configuration (settings.py):**

from pathlib import Path

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

TEMPLATES = [

{

...

'DIRS': [BASE\_DIR / 'templates'], # Look for global templates

'APP\_DIRS': True, # Also search inside app templates

...

},

]

**3️⃣ Create Base HTML Layout (users/templates/users/base.html)**

This is the **main layout** that other pages will inherit from:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>School App</title>

</head>

<body>

<h1>Welcome</h1>

{% block content %}

<!-- Page-specific content will be inserted here -->

{% endblock %}

</body>

</html>

**4️⃣ Create Login and Register Pages**

**📄 users/templates/users/register.html**

{% extends 'users/base.html' %}

{% block content %}

<h2>Register</h2>

<!-- Form will be added later -->

{% endblock %}

**📄 users/templates/users/login.html**

{% extends 'users/base.html' %}

{% block content %}

<h2>Login</h2>

<!-- Form will be added later -->

{% endblock %}

**5️⃣ Views: Render Login & Register Pages**

**📄 users/views.py**

from django.shortcuts import render

def register\_view(request):

return render(request, 'users/register.html')

def login\_view(request):

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

📌 **Why?**

* render() is used to return an HTML response.
* Template paths are relative to the templates/ folder.

**6️⃣ App-Level URLs**

**📄 users/urls.py**

from django.urls import path

from . import views

urlpatterns = [

path('register/', views.register\_view, name='register'),

path('login/', views.login\_view, name='login'),

]

📌 **Why?**

* We keep URL patterns inside the app for modularity.
* The name= allows reverse URL lookups in templates.

**7️⃣ Hook App URLs to Project**

**📄 schoolapp/urls.py**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

path('', include('users.urls')), # Delegate to the users app

]

📌 **Why?**

* This links the root URL (/) to the users app.
* You can also scope it with path('users/', include(...)) later for clarity.

**8️⃣ Authentication Redirect Settings**

In schoolapp/settings.py, configure post-login and post-logout redirect URLs:

LOGIN\_REDIRECT\_URL = '/'

LOGOUT\_REDIRECT\_URL = '/login/'

**9️⃣ Create Admin User (Superuser)**

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser

Fill in:

Username: admin

Email: admin@example.com

Password: \*\*\*\*\*\*\*\*

Then go to:  
👉 http://127.0.0.1:8000/admin/

**✅ What You Have So Far**

| **Feature** | **Status** | **Template** | **View Function** | **URL** |
| --- | --- | --- | --- | --- |
| Register Page | ✅ Renders | register.html | register\_view | /register/ |
| Login Page | ✅ Renders | login.html | login\_view | /login/ |
| Base Layout | ✅ Created | base.html | Used via {% extends %} | N/A |
| Admin Panel | ✅ Functional | Built-in | Superuser login | /admin/ |

**💡 What's Next?**

Once this is tested and working, the next steps for the users app could be:

1. **Add Registration Form using UserCreationForm**
2. **Add Login using Django’s built-in AuthenticationForm**
3. **Add Logout view**
4. **Protect pages using @login\_required**
5. **Show messages like “Registered Successfully”**

Would you like the **next version of this doc** including form handling and validation (e.g., functional register/login forms)?