📘Django Project Documentation: `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 Guide

✂️Project Initialization

Commands:

```
django-admin startproject schoolapp
cd schoolapp
python manage.py startapp users
```

Why? - `schoolapp` is the main project containing settings & configurations. - `users` app will handle login and registration logic.

Where? - `schoolapp/settings.py` → register `users` in `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    ...
    'users',  # Our custom app
]
```

✒️Templates Setup

Why? - Django needs to know where to look for HTML templates. - Default: app-level `templates/` (via `APP_DIRS=True`). - Added global `templates/` for shared layouts.

Where? 📄 `schoolapp/settings.py`

```
from pathlib import Path
BASE_DIR = Path(__file__).resolve().parent.parent

TEMPLATES = [
    {
        ...
        'DIRS': [BASE_DIR / 'templates'],  # Global templates
        'APP_DIRS': True,                  # App templates
        ...
    },
]
```

🛏️Base Layout

📄 `users/templates/users/base.html`

```
<!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>
```

Why? - DRY principle: one layout reused across all pages.

🌂 Login & Register Templates

📄 `users/templates/users/register.html`

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

{% block content %}
<h2>Register</h2>
<!-- Registration form will go here -->
{% endblock %}
```

📄 `users/templates/users/login.html`

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

{% block content %}
<h2>Login</h2>
<!-- Login form will go here -->
{% endblock %}
```

🌡️ Views

📄 `users/views.py`

```python
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()` maps templates to views. - Keeps business logic in one place.

🎒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? - Each app manages its own URL patterns. - `name=` allows template lookups with `{% url 'register' %}`.

🎓Hook into Project URLs

📄 `schoolapp/urls.py`

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('users.urls')),  # Delegates to users app
]
```

Why? - Keeps project modular. - You can later scope routes (e.g., `path('users/', include(...))`).

🎩Authentication Redirects

📄 `schoolapp/settings.py`

```
LOGIN_REDIRECT_URL = '/'
LOGOUT_REDIRECT_URL = '/login/'
```

Why? - Tells Django where to redirect after login/logout.

🎽Create Admin User

```
python manage.py makemigrations
python manage.py migrate
python manage.py createsuperuser
```

Fill in credentials: - Username: `admin` - Email: `admin@example.com` - Password: `********`

🔔Access: `http://127.0.0.1:8000/admin/`

🔗Current Progress

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	via `{% extends %}`	N/A
Admin Panel	🔗Ready	Built-in	Superuser login	/admin/

💡 Next Steps

1. Add **Registration Form** (`UserCreationForm`).
2. Add **Login** (`AuthenticationForm`).
3. Implement **Logout View**.
4. Secure pages with `@login_required`.
5. Display success/error messages (Django messages framework).

📌This concludes **Phase 1 Documentation** for `schoolapp`. Phase 2 will cover **functional forms and authentication logic**.