

Building a Django Sign-Up/Sign-In and Dashboard Application with Microservices

June 15, 2025

Introduction

This document provides step-by-step instructions to build a Django application with separate apps for authentication (sign-up/sign-in) and a dashboard. The UI design is inspired by the provided screenshot, featuring a clean sign-in page and a dashboard with a left-side sliding menu. The project uses a microservices-inspired approach by splitting functionality into two apps: `auth_app` and `dashboard_app`. All code and steps are included below.

1 Step 1: Set Up the Django Project

Create a virtual environment, install Django, and set up the project structure.

```
1 # Create a virtual environment
2 python -m venv venv
3 source venv/bin/activate  % On Windows: venv\Scripts\activate
4
5 # Install Django
6 pip install django
7
8 # Create the Django project
9 django-admin startproject myproject
10 cd myproject
11
12 # Create two apps: auth_app and dashboard_app
13 python manage.py startapp auth_app
14 python manage.py startapp dashboard_app
```

2 Step 2: Configure Project Settings

Update `myproject/settings.py` to include the apps and configure templates.

```
1 INSTALLED_APPS = [
2     'django.contrib.admin',
3     'django.contrib.auth',
4     'django.contrib.contenttypes',
5     'django.contrib.sessions',
6     'django.contrib.messages',
7     'django.contrib.staticfiles',
8     'auth_app',
9     'dashboard_app',
10 ]
11
12 TEMPLATES = [
13     {
14         'BACKEND': 'django.template.backends.django.DjangoTemplates',
15         'DIRS': ['templates'],
```

```

16     'APP_DIRS': True,
17     'OPTIONS': {
18         'context_processors': [
19             'django.template.context_processors.debug',
20             'django.template.context_processors.request',
21             'django.contrib.auth.context_processors.auth',
22             'django.contrib.messages.context_processors.messages',
23         ],
24     },
25 },
26 ]
27
28 STATICFILES_DIRS = [
29     'static',
30 ]
31
32 STATIC_URL = '/static/'

```

Create the following directories in the project root:

- `templates/` (for HTML templates)
- `static/` (for CSS and JavaScript files)

3 Step 3: Set Up URLs

Configure the project and app URLs.

3.1 Project URLs (`myproject/urls.py`)

```

1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('auth_app.urls')),
7     path('dashboard/', include('dashboard_app.urls')),
8 ]

```

3.2 Auth App URLs (`auth_app/urls.py`)

Create `authapp/urls.py`:

```
urlpatterns = [ path('', views.signin, name='signin'), path('signup/', views.signup, name='signup'),
path('logout/', views.logoutview, name =' logout'),]
```

3.3 Dashboard App URLs (`dashboard_app/urls.py`)

Create `dashboardapp/urls.py`:

```
urlpatterns = [ path('', views.dashboard, name='dashboard'), ]
```

4 Step 4: Implement Authentication in `auth_app`

Create views and templates for sign-up and sign-in.

4.1 Views (auth_app/views.py)

```
1 from django.shortcuts import render, redirect
2 from django.contrib.auth import authenticate, login, logout
3 from django.contrib.auth.models import User
4 from django.contrib import messages
5
6 def signin(request):
7     if request.method == 'POST':
8         email = request.POST['email']
9         password = request.POST['password']
10        try:
11            user = User.objects.get(email=email)
12            user = authenticate(request, username=user.username, password=password)
13            if user is not None:
14                login(request, user)
15                return redirect('dashboard')
16            else:
17                messages.error(request, 'Invalid email or password.')
18        except User.DoesNotExist:
19            messages.error(request, 'Invalid email or password.')
20    return render(request, 'auth/signin.html')
21
22 def signup(request):
23     if request.method == 'POST':
24         email = request.POST['email']
25         password = request.POST['password']
26         username = email.split('@')[0] # Simple username derivation
27         try:
28             user = User.objects.create_user(username=username, email=email,
29                                             password=password)
30             user.save()
31             login(request, user)
32             return redirect('dashboard')
33         except Exception as e:
34             messages.error(request, 'Error creating account.')
35     return render(request, 'auth/signup.html')
36
37 def logout_view(request):
38     logout(request)
39     return redirect('signin')
```

4.2 Templates

Create the following templates under `templates/auth/`.

4.2.1 Sign-In Page (templates/auth/signin.html)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Sign In</title>
7     <link rel="stylesheet" href="/static/css/styles.css">
8 </head>
9 <body>
10    <div class="auth-container">
11        <div class="auth-box">
12            <h2>Sign in</h2>
13            {% if messages %}
14                {% for message in messages %}
```

```

15         <div class="error-message">{{ message }}</div>
16     {% endfor %}
17 {% endif %}
18 <div class="social-login">
19     <button class="social-btn google">Sign in with Google</button>
20     <button class="social-btn facebook">Sign in with Facebook</button>
21     <button class="social-btn github">Sign in with GitHub</button>
22     <button class="social-btn sso">Sign in with SSO</button>
23 </div>
24 <div class="divider">Or continue with email</div>
25 <form method="POST">
26     {% csrf_token %}
27     <label>EMAIL</label>
28     <input type="email" name="email" placeholder="Enter your email
29         address" required>
30     <label>PASSWORD</label>
31     <input type="password" name="password" placeholder="*****"
32         required>
33     <a href="#" class="forgot-password">Forgot password</a>
34     <button type="submit" class="auth-btn">Sign in</button>
35 </form>
36     <p class="signup-link">Don't have an account? <a href="{% url 'signup'
37         %}">Sign up</a></p>
38 </div>
39 </div>
40 </body>
41 </html>

```

4.2.2 Sign-Up Page (`templates/auth/signup.html`)

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Sign Up</title>
7     <link rel="stylesheet" href="/static/css/styles.css">
8 </head>
9 <body>
10    <div class="auth-container">
11        <div class="auth-box">
12            <h2>Sign up</h2>
13            {% if messages %}
14                {% for message in messages %}
15                    <div class="error-message">{{ message }}</div>
16                {% endfor %}
17            {% endif %}
18            <div class="social-login">
19                <button class="social-btn google">Sign up with Google</button>
20                <button class="social-btn facebook">Sign up with Facebook</button>
21                <button class="social-btn github">Sign up with GitHub</button>
22                <button class="social-btn sso">Sign up with SSO</button>
23            </div>
24            <div class="divider">Or continue with email</div>
25            <form method="POST">
26                {% csrf_token %}
27                <label>EMAIL</label>
28                <input type="email" name="email" placeholder="Enter your email
29                    address" required>
30                <label>PASSWORD</label>
31                <input type="password" name="password" placeholder="*****"
32                    required>

```

```

31         <button type="submit" class="auth-btn">Sign up</button>
32     </form>
33     <p class="signup-link">Already have an account? <a href="{% url 'signin' %}">Sign in</a></p>
34   </div>
35 </div>
36 </body>
37 </html>

```

5 Step 5: Implement Dashboard in dashboard_app

Create the dashboard view and template with a sliding menu.

5.1 Views (dashboard_app/views.py)

```

1 from django.shortcuts import render, redirect
2 from django.contrib.auth.decorators import login_required
3
4 @login_required
5 def dashboard(request):
6     return render(request, 'dashboard/dashboard.html', {'user': request.user})

```

5.2 Template (templates/dashboard/dashboard.html)

Create the dashboard template under templates/dashboard/.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Dashboard</title>
7     <link rel="stylesheet" href="/static/css/styles.css">
8 </head>
9 <body>
10    <div class="container">
11        <div class="sidebar" id="sidebar">
12            <div class="sidebar-header">
13                
14                <span>{{ user.email }}</span>
15            </div>
16            <ul class="sidebar-menu">
17                <li><a href="#">Dashboard</a></li>
18                <li><a href="#">Agents</a></li>
19                <li><a href="#">Call History</a></li>
20                <li><a href="#">Knowledge Base</a></li>
21                <li><a href="#">Phone Numbers</a></li>
22                <li><a href="#">Outbound</a></li>
23                <li><a href="#">Settings</a></li>
24                <li><a href="{% url 'logout' %}">Logout</a></li>
25            </ul>
26        </div>
27        <div class="main-content">
28            <button class="menu-toggle" onclick="toggleSidebar()"> </button>
29            <h1>Welcome, {{ user.email }}</h1>
30            <div class="dashboard-stats">
31                <div class="stat-box">
32                    <p>Number of calls</p>
33                    <h3>0</h3>

```

```

34         </div>
35         <div class="stat-box">
36             <p>Average duration</p>
37             <h3>0:00</h3>
38         </div>
39         <div class="stat-box">
40             <p>Total cost</p>
41             <h3>0 credits</h3>
42         </div>
43         <div class="stat-box">
44             <p>Average cost</p>
45             <h3>0 credits/call</h3>
46         </div>
47         <div class="stat-box">
48             <p>Total LLM cost</p>
49             <h3>US\$0.00</h3>
50         </div>
51     </div>
52     <p class="no-metrics">No metrics for the specified period.</p>
53 </div>
54 </div>
55 <script src="/static/js/scripts.js"></script>
56 </body>
57 </html>

```

6 Step 6: Add Styling

Create static/css/styles.css to style the pages like the screenshot.

```

1 body {
2     font-family: Arial, sans-serif;
3     margin: 0;
4     padding: 0;
5     background-color: #f5f5f5;
6 }
7
8 .auth-container {
9     display: flex;
10    justify-content: center;
11    align-items: center;
12    height: 100vh;
13    background-color: #f5f5f5;
14 }
15
16 .auth-box {
17     background: white;
18     padding: 40px;
19     border-radius: 10px;
20     box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
21     width: 400px;
22     text-align: center;
23 }
24
25 .auth-box h2 {
26     margin-bottom: 20px;
27     font-size: 24px;
28 }
29
30 .social-login {
31     display: flex;
32     flex-direction: column;
33     gap: 10px;

```

```

34 }
35
36 .social-btn {
37     padding: 10px;
38     border: 1px solid #ddd;
39     border-radius: 5px;
40     background: none;
41     cursor: pointer;
42     font-size: 16px;
43 }
44
45 .social-btn.google { background: url('/static/images/google-icon.png') no-repeat 10
46   px center; padding-left: 40px; }
46 .social-btn.facebook { background: url('/static/images/facebook-icon.png') no-
47   repeat 10px center; padding-left: 40px; }
47 .social-btn.github { background: url('/static/images/github-icon.png') no-repeat 10
48   px center; padding-left: 40px; }
48 .social-btn.sso { background: url('/static/images/sso-icon.png') no-repeat 10px
49   center; padding-left: 40px; }
50
50 .divider {
51     margin: 20px 0;
52     color: #888;
53 }
54
55 form label {
56     display: block;
57     text-align: left;
58     font-size: 12px;
59     color: #888;
60     margin-top: 10px;
61 }
62
63 form input {
64     width: 100%;
65     padding: 10px;
66     margin: 5px 0;
67     border: 1px solid #ddd;
68     border-radius: 5px;
69     font-size: 14px;
70 }
71
72 .forgot-password {
73     display: block;
74     text-align: right;
75     font-size: 12px;
76     color: #888;
77     text-decoration: none;
78     margin-top: 5px;
79 }
80
81 .auth-btn {
82     width: 100%;
83     padding: 10px;
84     background-color: #ddd;
85     border: none;
86     border-radius: 5px;
87     font-size: 16px;
88     cursor: pointer;
89     margin-top: 20px;
90 }
91
92 .signup-link {

```

```

93     margin-top: 20px;
94     font-size: 14px;
95   }
96
97 .signup-link a {
98   color: #007bff;
99   text-decoration: none;
100 }
101
102 .error-message {
103   color: red;
104   font-size: 14px;
105   margin-bottom: 10px;
106 }
107
108 .container {
109   display: flex;
110   height: 100vh;
111 }
112
113 .sidebar {
114   width: 250px;
115   background: white;
116   padding: 20px;
117   box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
118   transition: transform 0.3s ease;
119 }
120
121 .sidebar-hidden {
122   transform: translateX(-250px);
123 }
124
125 .sidebar-header {
126   display: flex;
127   align-items: center;
128   gap: 10px;
129   margin-bottom: 20px;
130 }
131
132 .user-icon {
133   width: 40px;
134   height: 40px;
135   border-radius: 50%;
136 }
137
138 .sidebar-menu {
139   list-style: none;
140   padding: 0;
141 }
142
143 .sidebar-menu li {
144   margin: 10px 0;
145 }
146
147 .sidebar-menu a {
148   text-decoration: none;
149   color: #333;
150   font-size: 16px;
151 }
152
153 .main-content {
154   flex: 1;
155   padding: 20px;

```

```

156     background-color: #f5f5f5;
157 }
158
159 .menu-toggle {
160     display: none;
161     font-size: 24px;
162     background: none;
163     border: none;
164     cursor: pointer;
165 }
166
167 .dashboard-stats {
168     display: flex;
169     gap: 20px;
170     margin-top: 20px;
171 }
172
173 .stat-box {
174     background: white;
175     padding: 20px;
176     border-radius: 10px;
177     flex: 1;
178     text-align: center;
179     box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
180 }
181
182 .stat-box p {
183     margin: 0;
184     font-size: 14px;
185     color: #888;
186 }
187
188 .stat-box h3 {
189     margin: 10px 0 0;
190     font-size: 24px;
191 }
192
193 .no-metrics {
194     margin-top: 20px;
195     color: #888;
196 }
197
198 @media (max-width: 768px) {
199     .sidebar {
200         position: absolute;
201         height: 100%;
202         z-index: 1000;
203         transform: translateX(-250px);
204     }
205     .sidebar-visible {
206         transform: translateX(0);
207     }
208     .menu-toggle {
209         display: block;
210     }
211 }

```

7 Step 7: Add JavaScript for Sidebar Toggle

Create `static/js/scripts.js` for the sidebar toggle functionality.

```

1 function toggleSidebar() {

```

```
2 const sidebar = document.getElementById('sidebar');
3 sidebar.classList.toggle('sidebar-hidden');
4 sidebar.classList.toggle('sidebar-visible');
5 }
```

8 Step 8: Run Migrations and Test

Run migrations to set up the database and start the server.

```
1 python manage.py makemigrations
2 python manage.py migrate
3 python manage.py runserver
```

Visit <http://127.0.0.1:8000/> to see the sign-in page. Sign up or sign in to access the dashboard.

9 Notes

- The social login buttons (Google, Facebook, GitHub, SSO) are placeholders. Implementing them requires additional OAuth setup.
- Replace the icon URLs in `styles.css` with actual paths to icons if needed.
- The dashboard stats are static for now but can be made dynamic by integrating with a database.