

## ----- DJANGO TUTORIAL 8 -----

Django comes with both user authentication and new user signup built in! It is very important to point out that as long as a developer does not do anything like creating raw queries Django is the most secure web framework with ASP.NET coming in at a close 2nd.

NodeJS along with Express is definitely less secure because they require outside modules for security. I have personally seen many projects that used malicious modules that broke the security of the site. With that being said let's jump into coding!

# 1. We will be using the LoginView which requires a **login.html** page stored in a directory named registration in the templates folder

```
{% extends 'base.html' %}
{% block content %}
<h3>Log In</h3>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit" class="btn btn-primary btn-lg">Log In</button>
</form>
{% endblock content %}
```

# 2. Add accounts URL to the project\_3 urls.py file. It is the one with the admin link in it.

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path('', include('articles.urls')),
]
```

# 3. Set the redirect page after a login and log out in settings.py

```
LOGIN_REDIRECT_URL = 'home'
LOGOUT_REDIRECT_URL = 'home'
```

# 4. Verify if the user is logged in base.html

```
<div class="wrapper">
  <!-- Check if user is logged in and display message -->
  {% if user.is_authenticated %}
    <p>Hi {{ user.username }}!</p>
  {% else %}
    <p>You are not logged in <a href="{% url 'login' %}" class="btn btn-primary" btn-lg>Log
In</a></p>
  {% endif %}
{% block content %}
```

# 5. Add a link in the toolbar that allows users to log out

```
<ul>
  <li><a href="{% url 'home' %}">Home</a></li>
  <li><a href="{% url 'add_article' %}">Add Article</a></li>
  {% if user.is_authenticated %}
    <li><a href="{% url 'logout' %}">Log Out</a></li>
  {% endif %}
</ul>
```

## ----- NEW USER SIGNUP -----

# 1. Django helps with new user sign ups with UserCreationForm

# We'll create a new app that handles new user signup

python manage.py startapp accounts

# 2. Add the app to setting.py

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

# 3. Add a URL path to project\_3/urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path('accounts/', include('accounts.urls')),
    path('', include('articles.urls')),
]
```

# 4. Create the accounts/urls.py file

from django.urls import path

from .views import SignUpView

```
urlpatterns = [
    path('signup/', SignUpView.as_view(), name='signup'),
]
```

```
# 5. Create accounts/views.py
from django.contrib.auth.forms import UserCreationForm
from django.urls import reverse_lazy
from django.views import generic
```

```
class SignUpView(generic.CreateView):
    # Using the built-in user creation form
    form_class = UserCreationForm
    # Redirect user to login page after registration
    success_url = reverse_lazy('login')
    template_name = 'signup.html'
```

```
# 6. Add templates/signup.html
{% extends 'base.html' %}
{% block content %}
<h2>New User Sign Up</h2>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Sign Up</button>
</form>
{% endblock content %}
```

```
# 7. Update base.html to add the sign up button
```

```
<div class="wrapper">
    <!-- Check if user is logged in and display message -->
    {% if user.is_authenticated %}
        <p>Hi {{ user.username }}!</p>
    {% else %}
        <p>You are not logged in <a href="{% url 'login' %}" class="btn btn-primary" btn-lg>Log
In</a>
        <a href="{% url 'signup' %}" class="btn btn-primary" btn-lg>Sign Up</a></p>
    {% endif %}
{% block content %}
```