

Django login views

Django has its own class based views to deal with authentication

```
from django.contrib.auth import views as auth_views
from django.urls import path
from . import views

urlpatterns = [
    # old login path
    #path('login/', views.user_login, name='login'),
    path('login/', auth_views.LoginView.as_view(), name='login'),
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),
]
```

Create Login.html templates/registration

```
{% extends "base.html" %}

{% block title %}Log-in{% endblock %}

{% block content %}
<h1>Log-in</h1>
{% if form.errors %}
<p>
    Your username and password didn't match
    Please try again.
</p>
{% else %}
<p>Please, use the following form to log-in</p>
{% endif %}
<div class="login-form">
    <form action="{% url 'login' %}" method="post">
        {{ form.as_p }}
        {% csrf_token %}
        <input type="hidden" name="next" value="{{ next }}" />
        <p><input type="submit" value="Log-in"></p>
    </form>
</div>
{% endblock %}
```

We have created a login template and a logout template using django's authenticationForm in `django.contrib.auth.forms`

- `{% if form.errors %}` checks if credentials are wrong

Create logged_out.html templates/registration

```
{% extends "base.html" %}

{% block title %}Logged out{% endblock %}

{% block content %}
```

```

<h1>Logged out</h1>
<p>
    You have been successfully logged out.
    You can <a href="{% url 'login' %}">log-in again</a>
</p>
{% endblock %}

```

- Displayed after logout

Inside view.py add

```
from django.contrib.auth.decorators import login_required
```

```

@login_required
def dashboard(request):
    return render(
        request,
        'account/dashboard.html',
        {'section': 'dashboard'}
    )

```

- Created the dashboard view
- Applied login_required decorator
 - o Checks whether the user is authenticated
- If authenticated executes the decorated view
- If not authenticated directs to login url as a GET parameter named next
- By doing this the login view redirects to the URL that they were trying to access after the successfully logged in.
 - o We added a hidden <input> named next in login template for this purpose

Add to account/urls.py

```
path("", views.dashboard, name='dashboard'),
```

Add login to setting.py

```

LOGIN_REDIRECT_URL = 'dashboard'
    - Tells django which url to redirect user to after login
LOGIN_URL = 'login'
    - Redirect the user to login
LOGOUT_URL = 'logout'
    - The url to redirect the user to log out

```

Add a link to login URL and button to log out to the base template.

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <title>{% block title %}{% endblock %}</title>
    <link href="{% static 'css/base.css' %}" rel="stylesheet">
</head>
<body>

```

```

<div id="header">
  <span class="logo">Bookmarks</span>
  {% if request.user.is_authenticated %}
    <ul class="menu">
      <li {% if section == "dashboard" %}class="selected"{% endif %}>
        <a href="{% url 'dashboard' %}">My dashboard</a>
      </li>
      <li {% if section == "images" %}class="selected"{% endif %}>
        <a href="#">Images</a>
      </li>
      <li {% if section == "people" %}class="selected"{% endif %}>
        <a href="#">People</a>
      </li>
    </ul>
  {% endif %}
  <span class="user">
    {% if request.user.is_authenticated %}
      Hello {{ request.user.first_name|default:request.user.username }},
      <form action="{% url 'logout' %}" method="post">
        <button type="submit">Logout</button>
        {% csrf_token %}
      </form>
    {% else %}
      <a href="{% url 'login' %}">Log-in</a>
    {% endif %}
  </span>
</div>
<div id="content">
  {% block content %}
  {% endblock %}
</div>
</body>
</html>

```