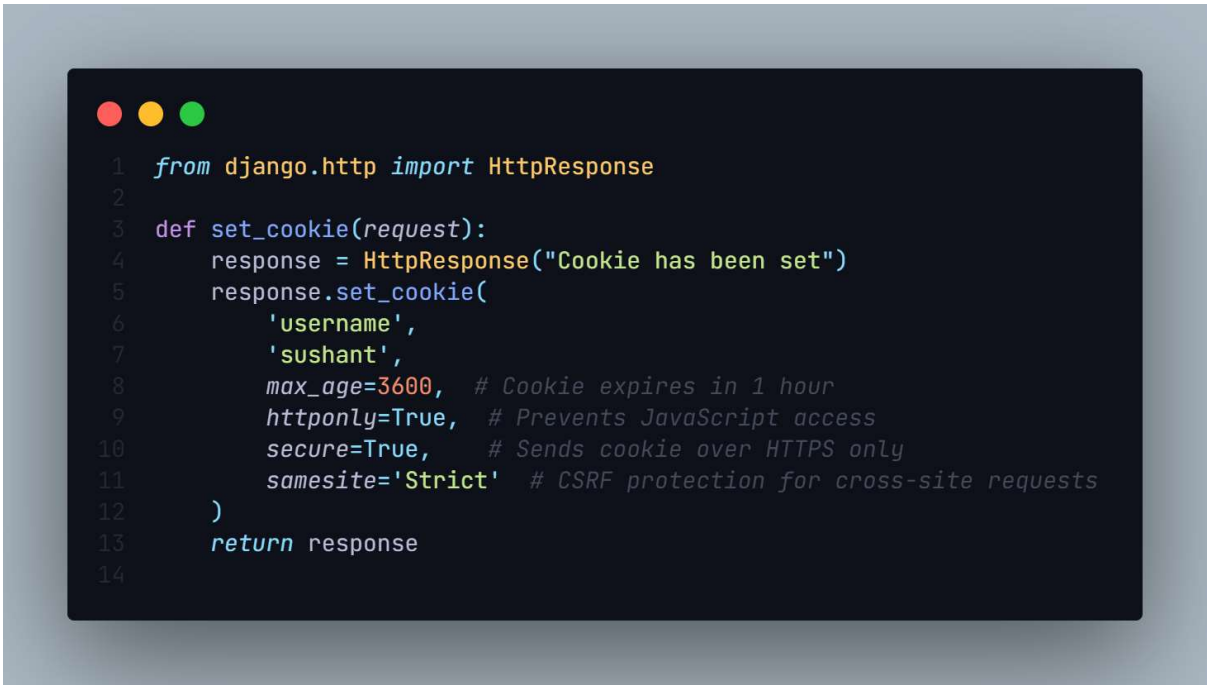
How do you set a cookie in Django using the HttpResponse object?

In Django, you can set cookies using the set_cookie() method on an HttpResponse object.

Example of Setting a Cookie:



```python
from django.http import HttpResponse

def set_cookie(request):
    response = HttpResponse("Cookie has been set")
    response.set_cookie(
        'username',
        'sushant',
        max_age=3600,  # Cookie expires in 1 hour
        httponly=True,  # Prevents JavaScript access
        secure=True,    # Sends cookie over HTTPS only
        samesite='Strict'  # CSRF protection for cross-site requests
    )
    return response
```

Example of Reading a Cookie:

```python
def get_cookie(request):
    username = request.COOKIES.get('username', 'Guest')
    return HttpResponse(f"Hello, {username}")
```

What are Django sessions, and how do they work?

Django Sessions store user-specific data on the server-side and use a session ID stored in the browser's cookies to track the session.

How Django Sessions Work:

1. **Session Creation:** When a user accesses the website, Django assigns them a session ID.
2. **Session Storage:** Data is stored on the server (in the database, cache, or file system).
3. **Cookie with Session ID:** The user's browser stores only the session ID.
4. **Session Retrieval:** On each request, Django uses the session ID to retrieve user data from the server.

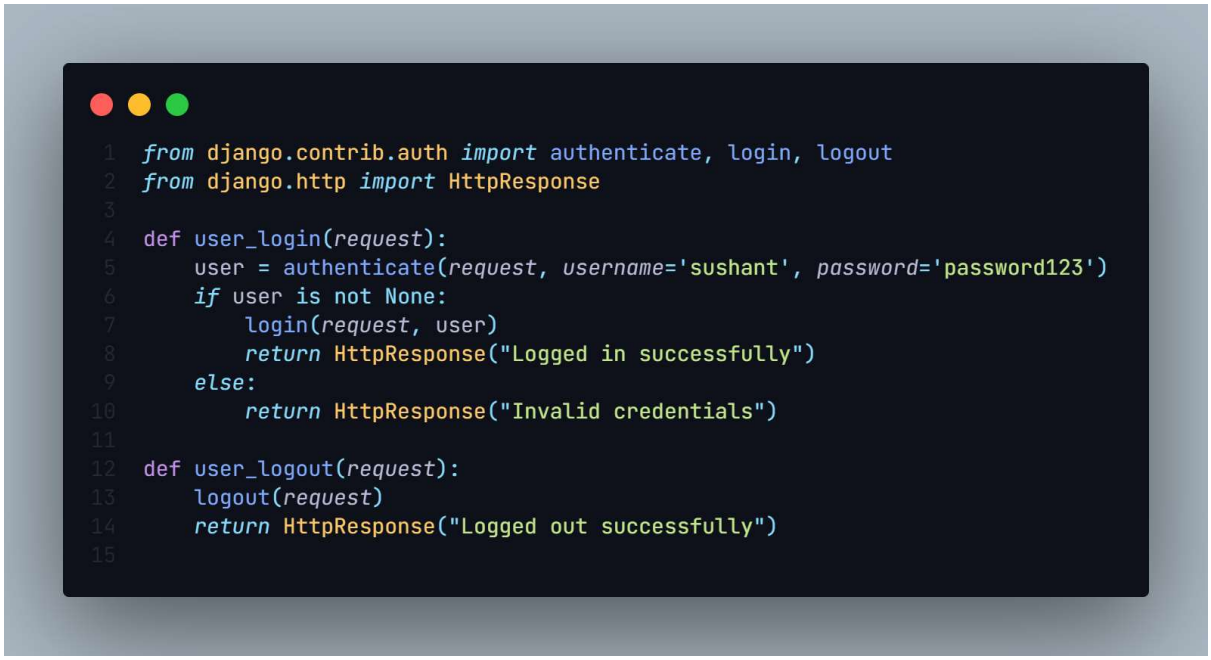
Example of Setting and Getting a Session:

```python
# Set session
def set_session(request):
    request.session['username'] = 'sushant'
    request.session['email'] = 'sushant@example.com'
    return HttpResponse("Session has been set")

# Get session
def get_session(request):
    username = request.session.get('username', 'Guest')
    email = request.session.get('email', 'Not provided')
    return HttpResponse(f"Hello {username}, your email is {email}")

# Delete session
def delete_session(request):
    request.session.flush()  # Clears all session data
    return HttpResponse("Session has been deleted")
```

What is the difference between authentication and authorization in Django?

FEATURE	AUTHENTICATION	AUTHORIZATION
DEFINITION	Verifies who the user is (identity).	Determines what a user can do (permissions).
PURPOSE	Confirms user identity via login credentials.	Grants or denies access to resources.
IMPLEMENTATION	Handled by Django's built-in authenticate() and login() methods.	Managed through permissions, groups, and is_staff or is_superuser flags.
EXAMPLE	Logging in with a username and password.	Allowing only admins to access the admin panel.
EXAMPLE	Logging in with a username and password.	Allowing only admins to access the admin panel.
DJANGO MODULES	django.contrib.auth for login/logout.	django.contrib.auth with groups, permissions, and decorators (@permission_required, @login_required).

Example of Authentication (Login/Logout):



```python
from django.contrib.auth import authenticate, login, logout
from django.http import HttpResponse

def user_login(request):
    user = authenticate(request, username='sushant', password='password123')
    if user is not None:
        login(request, user)
        return HttpResponse("Logged in successfully")
    else:
        return HttpResponse("Invalid credentials")

def user_logout(request):
    logout(request)
    return HttpResponse("Logged out successfully")
```