What is CSRF protection in Django, and how does it work?

CSRF (Cross-Site Request Forgery) is a security measure that prevents malicious websites from making unauthorized requests on behalf of authenticated users.

How CSRF Protection Works in Django:

- **CSRF Token:** Django generates a unique, secret token for each user session.
- **Token Verification:** When a user submits a form (e.g., logging in, updating a profile), the form must include this token.
- **Middleware Check:** Django's CSRF middleware checks the token from the form against the token stored in the session. If they match, the request is allowed. Otherwise, it is rejected.

Using CSRF in Django:

- Add {% csrf_token %} inside forms in templates:

What are Django cookies, and how do they differ from sessions?

FEATURE	COOKIES	SESSIONS
DEFINITION	Small pieces of data stored in the user's browser.	Server-side storage mechanism for user data.
STORAGE LOCATION	Stored in the user's browser.	Stored on the server (database or cache).
PURPOSE	Track user preferences (e.g., remember me).	Store user-specific information (e.g., login status).
SECURITY	Less secure (stored on client-side).	More secure (data is stored on the server).
EXPIRATION	Can be set to expire after a specific time.	Usually tied to the user's session and expires automatically.
SIZE LIMIT	Limited (usually 4KB).	Can store more data since storage is on the server.

```python
# Example of Setting a Cookie:
def set_cookie(request):
    response = HttpResponse("Cookie Set")
    response.set_cookie('username', 'sushant', max_age=3600)  # Expires in 1 hour
    return response



# Example of Setting a Session:
def set_session(request):
    request.session['username'] = 'sushant'
    return HttpResponse("Session Set")

```