1. What is CSRF Protection in Django, and how does it work?

CSRF (Cross-Site Request Forgery) is an attack where someone tricks a logged-in user into submitting a request (like form submission) to a website **without their knowledge**.

Simple Example:

Imagine you're logged into your bank, and someone sends you a fake link. If you click it, it could try to transfer money **without your permission**.

✅ **Django's CSRF Protection:**

Django protects you using a **CSRF token**.

- When you submit a form, Django includes a **hidden token** (a special key).
- This token is checked on the server when the form is submitted.
- If the token is **missing or wrong**, Django **blocks the request**.

How to use it in templates:

```
<form method="post">
 {% csrf_token %}
 <!-- your form fields -->
</form>
```

---

2. What are Django Cookies, and how do they differ from Sessions?

🔸 **Cookies:**

- Small pieces of data stored in the **user's browser**.
- Django can set cookies using the response.
- Example: storing theme color, language preference, etc.

```
response.set_cookie('user_name', 'Alice')
```

🔸 **Sessions:**

- Stores data **on the server**.
- A unique **session ID** is stored in the browser (as a cookie).
- Django uses this ID to find the data on the server.

request.session['user_id'] = 101

🆚 **Cookie vs Session (Simple Table):**

Feature	Cookie	Session
Storage	In user's browser	On Django server
Size	Limited (few KB)	Can store more data
Security	Less secure	More secure
Use Case	Small, non-sensitive data	User login info, cart, etc.