**What is CSRF?**

**CSRF** stands for **Cross-Site Request Forgery**.

**Let’s see following scenario:**

You're logged into your bank’s website in one browser tab, maybe checking your balance.

Now you visit a "Trick Website"—maybe it looks like a funny meme site or a fake blog.

But secretly, that Trick Website has some hidden code that tries to send a request to your bank like this:

“Transfer ₹10,000 to account XYZ”

Since you’re already logged in, your browser might unknowingly send that request, and the bank thinks you sent it.

That’s what **CSRF** is — a Trick Website forging a request to another website on your behalf, without your permission.

In this scenario, since you're already logged in, your browser might send that request **with your session** (cookies), and the bank thinks **you approved it**.

That’s CSRF – someone **forges a request** from your side to another site you're logged into.

**What is a CSRF Token?**

To protect users, web apps add a **special hidden code** (called a **CSRF token**) to each form.

Think of it as a **secret handshake** between your form and the server.

* When the form is loaded, the server generates a **unique token** and sends it with the form.
* When the form is submitted, it **must include that same token**.
* If the token is **missing** or **doesn’t match**, the server says:  
  “Nope, this request might be fake!”

Only someone who **actually visited your form page** can get the correct token.

**In Django Forms**

When you create a form in Django and submit it using POST, Django expects a CSRF token for protection.

That’s why you always see:

<form method="POST">

{% csrf\_token %}

<!-- form fields here -->

</form>

If you **forget {% csrf\_token %}**, Django will block the request and show:

“403 Forbidden – CSRF verification failed.”

**Why It’s Important**

* Prevents **hackers** or **malicious websites** from making fake form submissions on behalf of users.
* Helps protect **logins, payments, settings**, or **any sensitive action** in your app.

**Real-Life Analogy**

Imagine you're placing an order at a store. The cashier gives you a **token number** (receipt).  
You must return with that token to receive your items.

If someone else tries to collect your order without the token, they’ll be denied.  
Same with CSRF — it confirms **you’re the real one making the request**.