Django Cookies

- We Know about Cookies and a purpose of cookies even though cookies are useful, there are some problems with cookies.
  1. An attacker can modifie the contents of cookie that can break the aplication easily.
  2. We cannot store sensitive data. (password)
  3. We can only store a limited amount of data in cookies.
  4. Most browsers dont allow a cookie to store more than 4kb of data.
  5. When we have more data, this data can be break into multiple cookies, but multiple cookies will cause too much overload in each request.
  6. Further you cont even depend on a no of cookies allowed by the browser.
  7. We can overcome these problems by using session.

Django Session

1. Whenever we use sessions then the data will not store directly in the browsers, it will store in the server.
2. Django creates a unique random string which is called sessionid or SID and associated this SID with the data.
3. The server sends a cookie named sessionid or SID as a value to the browser.
4. On requesting a page, the browser sents the request along with SID to the server.
5. Django uses this SID to retrive session data and makes it accesible in your code.
6. SID is a 32 characters long random string, so it is almost imposible to guess by an attacker.
7. SID is generated by Django.

- In Cookies concept, we can use set_cookie(name , value , max_age) for creating new cookies and delete_cookie(cookieName) for deleting cookies.
- In Sessions concept, we can use **set_test_cookie()** to create session and **delete_test_cookie()** to delete the sessions and **test_cookie_worked()** is used to test the session weather it is there or not.

  For example:

```
def viewName(request):
   if request.session.test_cookie_worked():
       request.session.delete_test_cookie()
       return HttpResponse('sessionid deleted')
    else:
       return HttpResponse('sessionid not available')
```