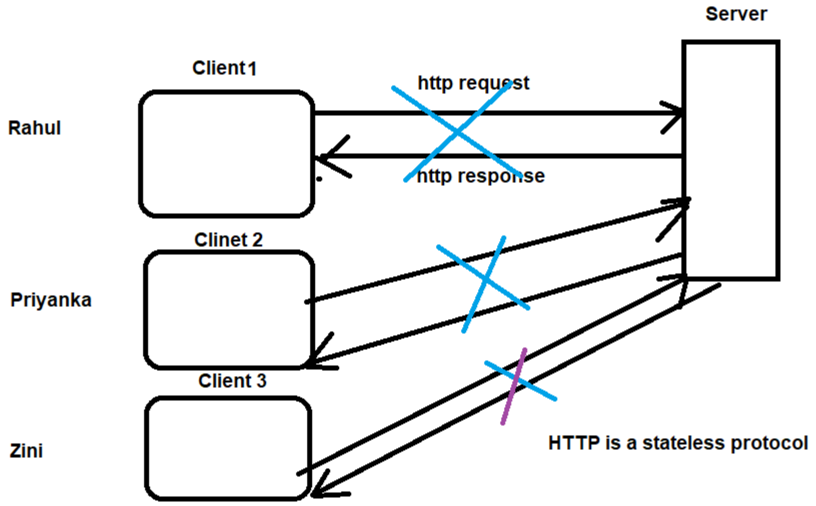
DAy: 30, Date: 20/07/2024
Chapter: 7, Name: State Management

- Normally whenever we are sending any request to server those requests are HTTP type.
- HTTP is a stateless protocol.
- 

- HTTP is a stateless protocol. It means once the response is delivered by the server then entire page information will be destroyed. This is the drawback of HTTP protocol.
- Stateless protocol means unable to remember the client information.
- If the server is unable to remember the client information, then that server will treat each request as a new request.
- To overcome the drawback of stateless mechanisms we must use state management techniques.
- State management is a process to maintain the state of the client or webpage.
- Here the meaning of the state is username, password, email address, etc.
- State value is maintained by either client/server.
- Some of the popular state management techniques are:
    - **COOKIES**
    - **SESSION**
    - **URL rewriting**
    - **Hidden Form Field**
    - **Cache etc.**
- State management is also called as Session management.

Chapter: 7.1(COOKIES)

- It is a state management technique.
- Cookies are a small amount of information.
- Server will create the cookie and client will maintain cookie.
- Maintain means store.
- Cookies can store 4KB of information which is equivalent to 4096 bytes.
- In the cookie data stores in the form of plain text.
- For a website, a browser can allow maximum 20 cookies.
- If it cross more than 20 cookies, then automatically it will deleted the old one.(21 –> ~~1~~, 22- >~~2~~ , 23->~~3~~).

Types of Cookies

- In-memory cookies
- Persistent cookies

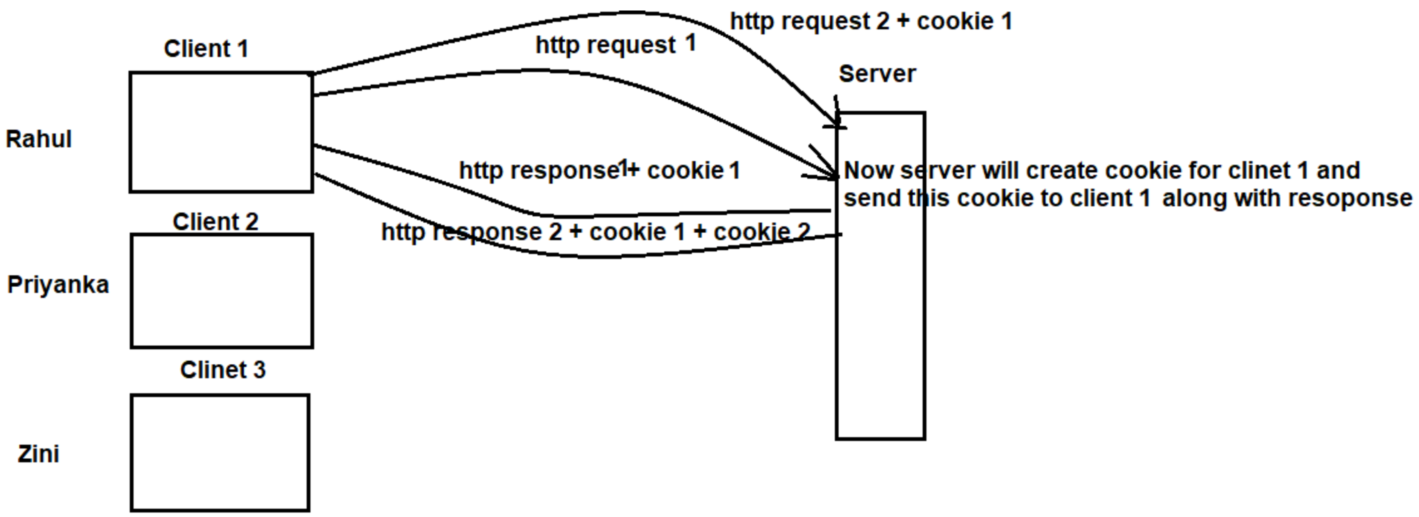
In-memory cookies:

- In in-memory cookies the cookie will be stored in browser memory, when the browser will close then cookie will be destroyed.

Persistent cookies:

- Persistent cookies have expiry date and time.
- If the browser close, then still it hold cookie value inside the browser memory.

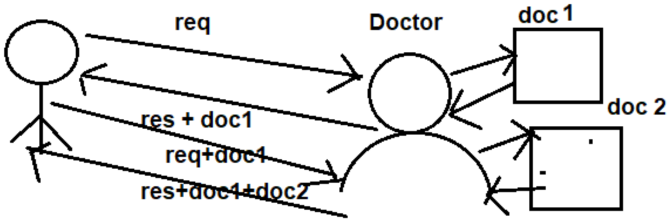
Diagram of client -server communication with cookie



Now server will create cookie for clinet 1 and send this cookie to client 1 along with resoponse

Client server communication with cookies

Here in this diagram client 1 sends http request 1 to server. Now the server receives the request from client 1. If the server wants to remember the client information, then the server will create a cookie object then server will send the cookie object along with the response to client 1. When client 1 sends more requests to the server, server will not treat as a new request.



Here
Patient : User
Doctor : server
Doc1 and Doc 2 :
Cookie

Here doctor(server) created the cookie and patient(client) maintain the cookie.

Real life example

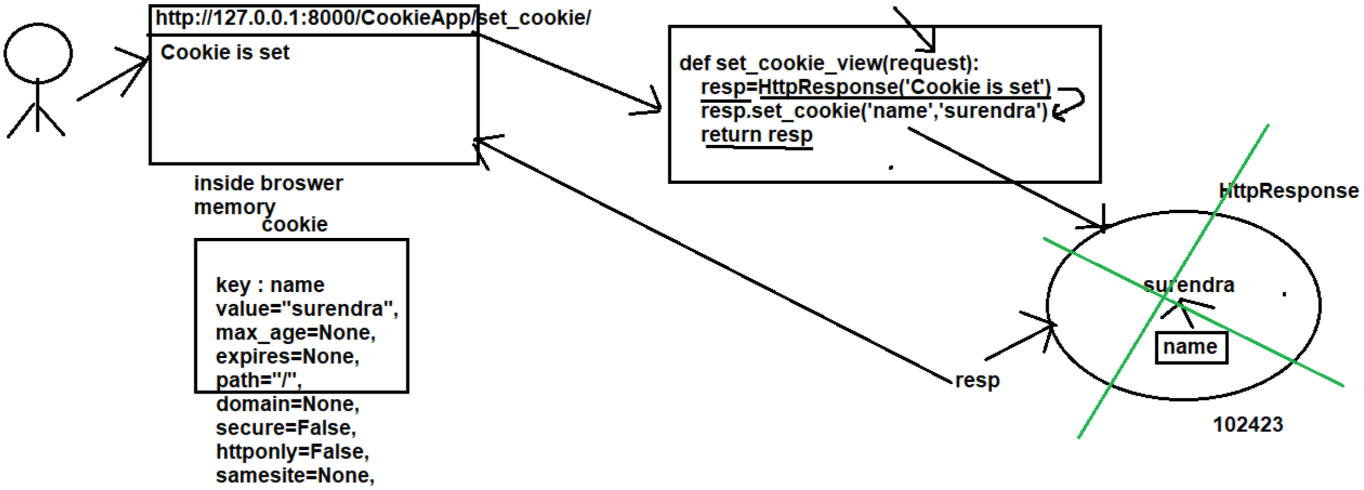
Day: 31 and Date: 21/07/2024

Topic: Implementation of Cookies

a) How to create the cookie?

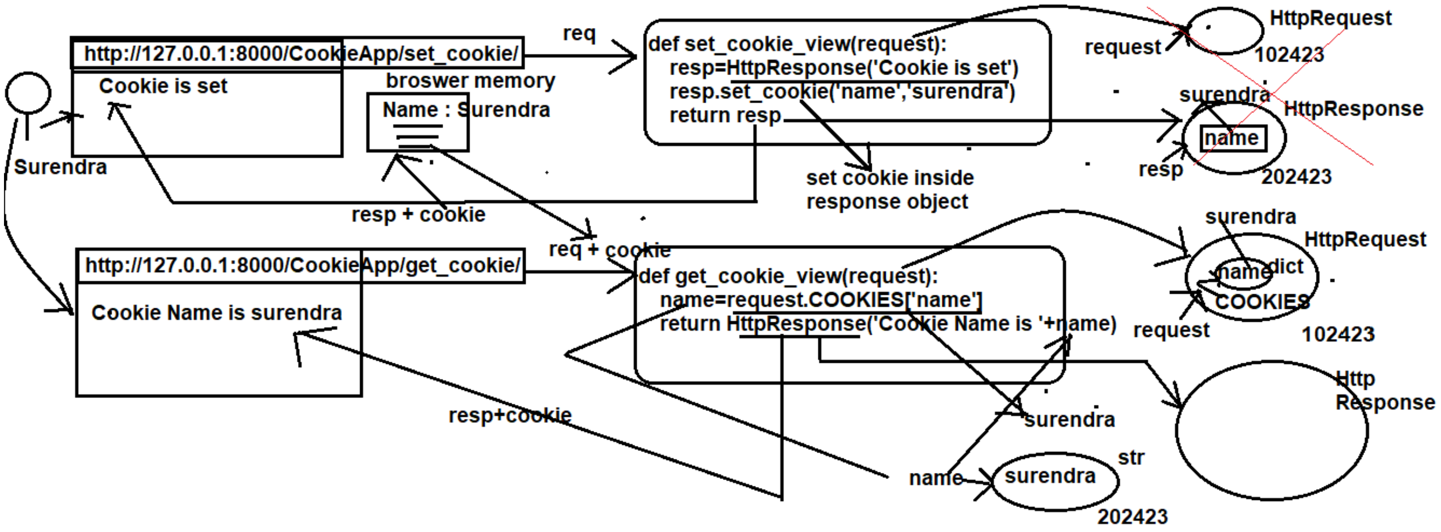
b) How to get the cookie?
c) How to update the cookie?
d) How to delete the cookie?

a) How to create the cookie:



b) How to set,get and delete the cookie

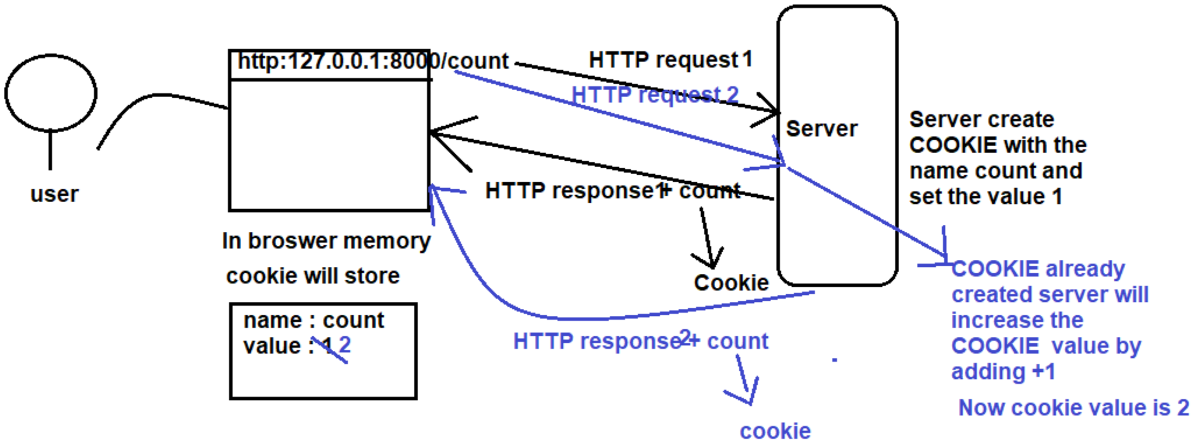
```
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def set_cookie_view(request):
    resp=HttpResponse('Cookie is set')
    resp.set_cookie('name','surendra')
    return resp

def get_cookie_view(request):
    name=request.COOKIES['name']
    return HttpResponse('Cookie Name is '+name)

def delete_cookie_view(request):
    resp=HttpResponse('Cookie deleted')
    resp.delete_cookie('name')
    return resp
```

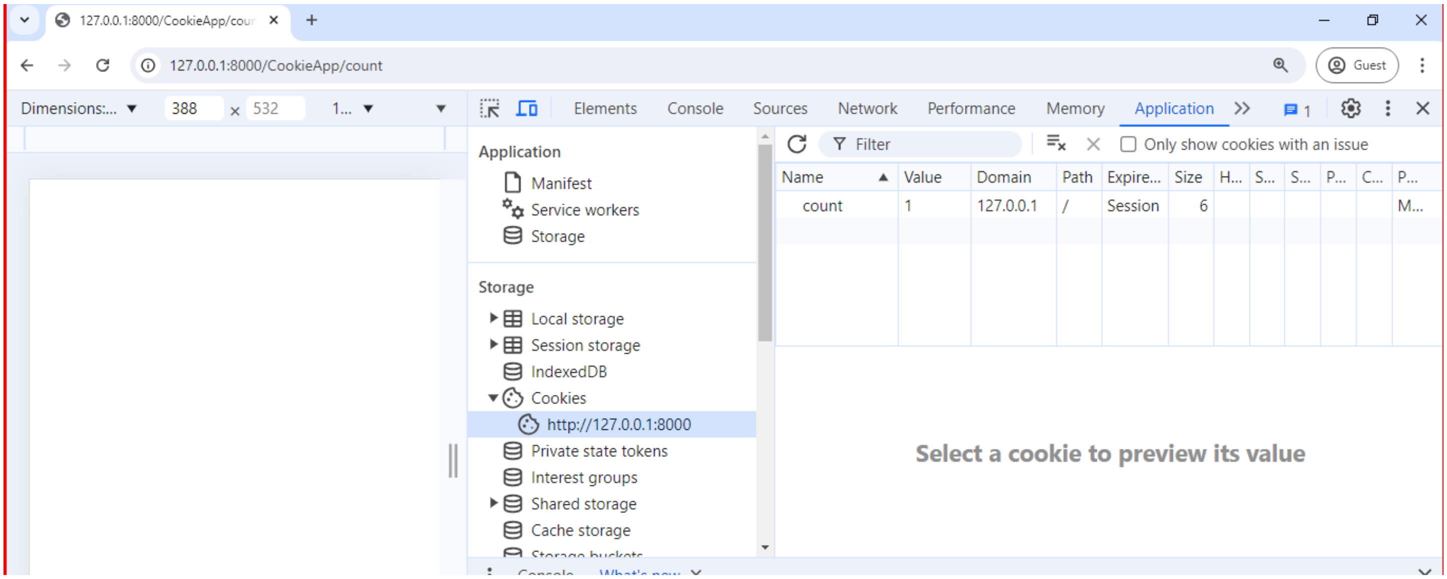


Day: 32 and Date: 29/07/2024
Topic: Implementation of Cookies

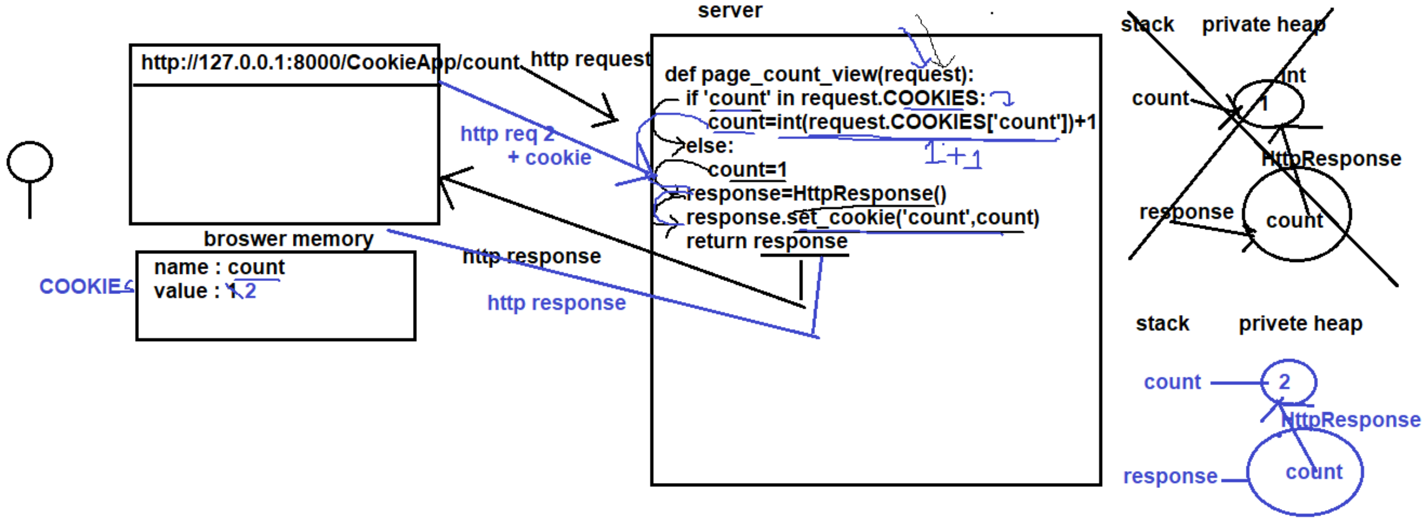
P034

Problem Statement: Develop page count application using COOKIE.





Name	Value	Domain	Path	Expire...	Size	H...	S...	S...	P...	C...	P...
count	1	127.0.0.1	/	Session	6						M...



```
def page_count_view(request):
  if 'count' in request.COOKIES:
    count=int(request.COOKIES['count'])+1
  else:
    count=1
  response=HttpResponse()
  response.set_cookie('count',count)
  return response
```

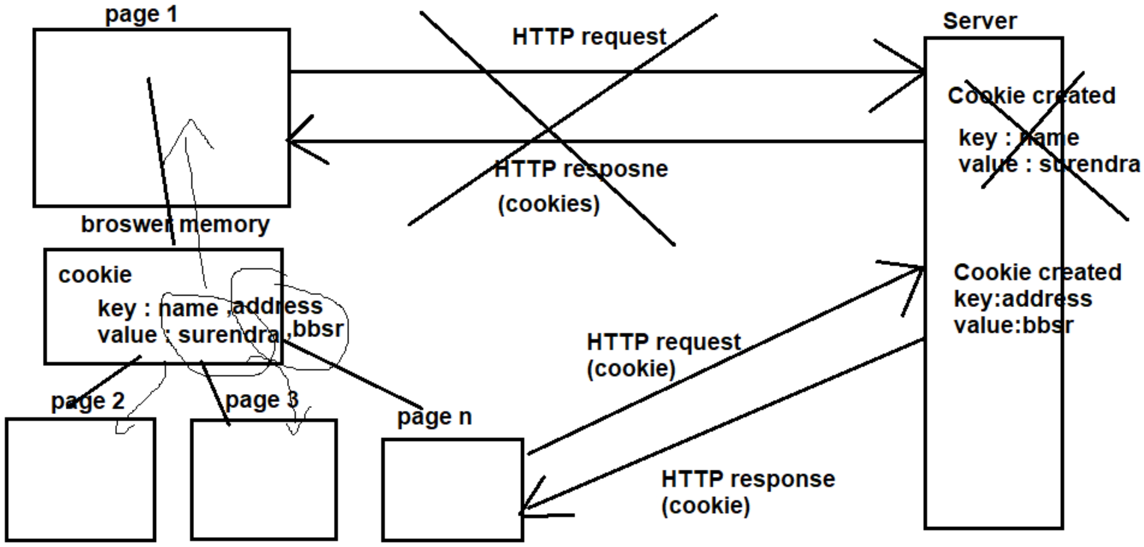
Views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def page_count_view(request):
    if 'count' in request.COOKIES:
        count=int(request.COOKIES['count'])+1
    else:
        count=1
    response=HttpResponse()
    response.set_cookie('count',count)
    return response
```

Day: 33 and Date: 30/07/2024

Topic: Implementation of Cookies

Cookie is used to transfer the data in between pages.



Transfer data in bettwen pages

Education_form.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container mt-5">
        <div class="mb-3">
            <form action="{% url 'personal_info' %}">
            <h3>Education Form</h3>
            <label for="college" class="form-label">College</label>
            <input type="text" class="form-control"  name="college">
            <label for="degree" class="form-label">Degree</label>
            <input type="text" class="form-control"  name="degree">
            <label for="mark" class="form-label">Mark</label>
            <input type="number" class="form-control"  name="mark">
            <div class="col-auto">
                <button type="submit" class="btn btn-primary mt-2">Next</button>
            </div>
            </form>
          </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
</html>
```

Personal_form.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container mt-5">
        <div class="mb-3">
            <form action="{% url 'extra_cur_info' %}">
            <h3>Persoanl Form</h3>
            <label for="name" class="form-label">Name</label>
            <input type="text" class="form-control"  name="name">
            <label for="address" class="form-label">Address</label>
            <input type="text" class="form-control" name="address">
            <label for="email" class="form-label">Email</label>
            <input type="email" class="form-control"  name="email">
            <div class="col-auto">
                <button type="submit" class="btn btn-primary mt-2">Next</button>
            </div>
            </form>
          </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
</html>
```

Extra_cur.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container mt-5">
        <div class="mb-3">
            <form action="{% url 'data' %}">
            <h3>Extra_Cur Form</h3>
            <label for="fav_sports" class="form-label">Fav Sports</label>
            <input type="text" class="form-control"  name="fav_sports">
            <label for="fav_actor" class="form-label">Fav Actor</label>
            <input type="text" class="form-control"  name="fav_actor">
            <div class="col-auto">
                <button type="submit" class="btn btn-primary mt-2">Next</button>
            </div>
            </form>
          </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
```

```
</html>
```

Data.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <a href="{% url 'all_records' %}">Click here to see all the cookie data</a>
</body>
</html>
```

All_records.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3>{{all_records}}</h3>
</body>
</html>
```

Views.py

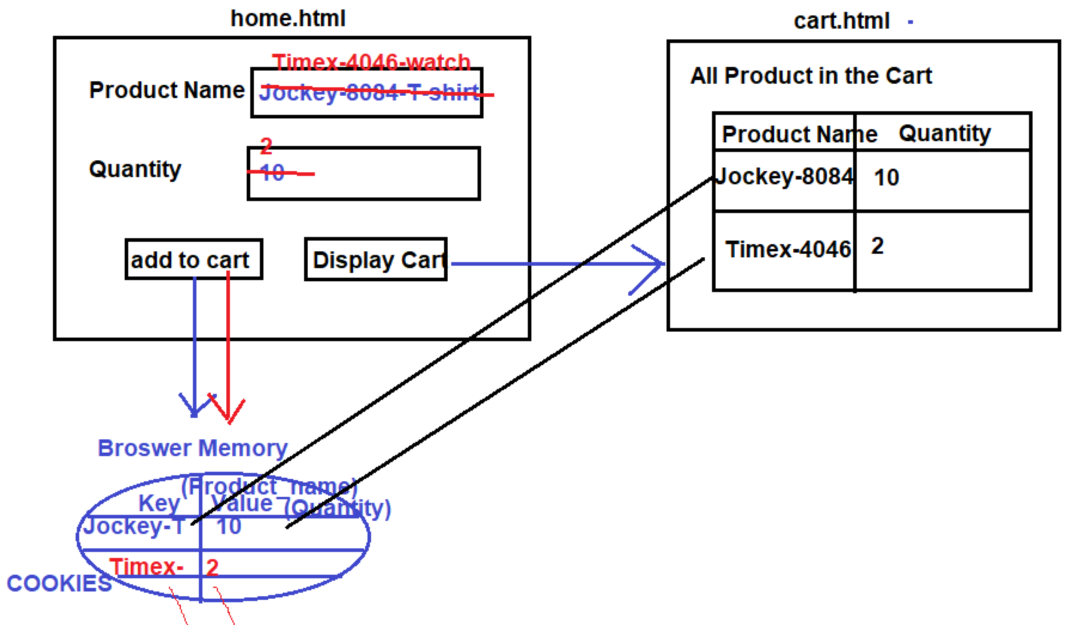
```
from django.shortcuts import render,HttpResponse

# Create your views here.
def education_form_view(request):
    return render(request,'StudentInfoApp/education_form.html')
def personal_form_view(request):
    college=request.GET['college']
    degree=request.GET['degree']
    mark=request.GET['mark']
    response=render(request,'StudentInfoApp/personal_form.html')
    response.set_cookie('college',college)
    response.set_cookie('degree',degree)
    response.set_cookie('mark',mark)
    return response
def extra_cur_form_view(request):
    name=request.GET['name']
    address=request.GET['address']
    email=request.GET['email']
    response=render(request,'StudentInfoApp/extra_cur.html')
    response.set_cookie('name',name)
    response.set_cookie('address',address)
    response.set_cookie('email',email)
    return response
def data_view(request):
    fav_sports=request.GET['fav_sports']
    fav_actor=request.GET['fav_actor']
    response = render(request,'StudentInfoApp/data.html')
    response.set_cookie('fav_sports',fav_sports)
    response.set_cookie('fav_actor',fav_actor)
    return response

def all_records_view(request):
    all_records=request.COOKIES
    context={
        'all_records':all_records
    }
    return render(request,'StudentInfoApp/all_records.html',context)
```

urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('education_info/',views.education_form_view,name='education_info'),
    path('personal_info/',views.personal_form_view,name='personal_info'),
    path('extra_info/',views.extra_cur_form_view,name='extra_cur_info'),
    path('data/',views.data_view,name='data'),
    path('all_records/',views.all_records_view,name='all_records'),
]
```

Day: 34 and Date: 01/08/2024
Topic: Implementation of Cookies

Add product and quantity in cart by using COOKIE.



Product Name	Quantity
Jockey-8084	10
Timex-4046	2

Key	Value
Jockey-T	10
Timex-	2

If COOKIE does not contain any product then in cart it will display there is no product in your cart.

Step 1: Common steps (project and app creation, template folder creation and configuration etc.)

Step 2: Define form class inside forms.py file

Forms.py

```
        from django import forms

        class Add_Product_Form(forms.Form):
           Product_Name=forms.CharField(max_length=30)
           Quantity= forms.IntegerField()
```

Step 3: views.py

```
from django.shortcuts import render,HttpResponse,redirect
from .forms import Add_Product_Form
# Create your views here.
def home_view(request):
   if request.method=='POST':
      form=Add_Product_Form(request.POST)
      if form.is_valid():
         product_name=form.cleaned_data['Product_Name']
         quantity=form.cleaned_data['Quantity']
         response=redirect('home')
         response.set_cookie(product_name,quantity)
   else:
      form=Add_Product_Form()
      response=render(request,'CartApp/home.html',{'form':form})
   return response

def display_cart_view(request):
   all_record=request.COOKIES
   context={
      'all_record':all_record,
   }
   return render(request,'CartApp/cart.html',context)
```

Step 4 : home.html

```
<!doctype html>
<html lang="en">
  <head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Bootstrap demo</title>
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
   <div class="container mt-3">
      <form action="" method="post">
         {% csrf_token %}
```

```
            <table>
                {{ form.as_table }}
                <tr>
                    <td><input type="submit" value="Add to cart"></td>
                </tr>
            </table>
        </form>
        <a href="{% url 'cart' %}">Display Cart</a>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
</html>
```

Step 5:cart.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container mt-3">
        <table class="table">
            <thead>
              <tr>
                <th scope="col">Product Name</th>
                <th scope="col">Quantity</th>
              </tr>
            </thead>
            <tbody>
              {% for key,value in all_record.items %}
              <tr>
                <th scope="row">{{key}}</th>
                <td>{{value}}</td>
              </tr>
              {% endfor %}
            </tbody>
          </table>
          <a href="{% url 'home' %}">Go back to Home</a>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  </body>
</html>
```

Step 6: url patten

Urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('home/', views.home_view,name='home'),
    path('cart/',views.display_cart_view,name='cart')
]
```

Step 7:
Run server and send HTTP request

Day: 35 and Date: 03/08/2024

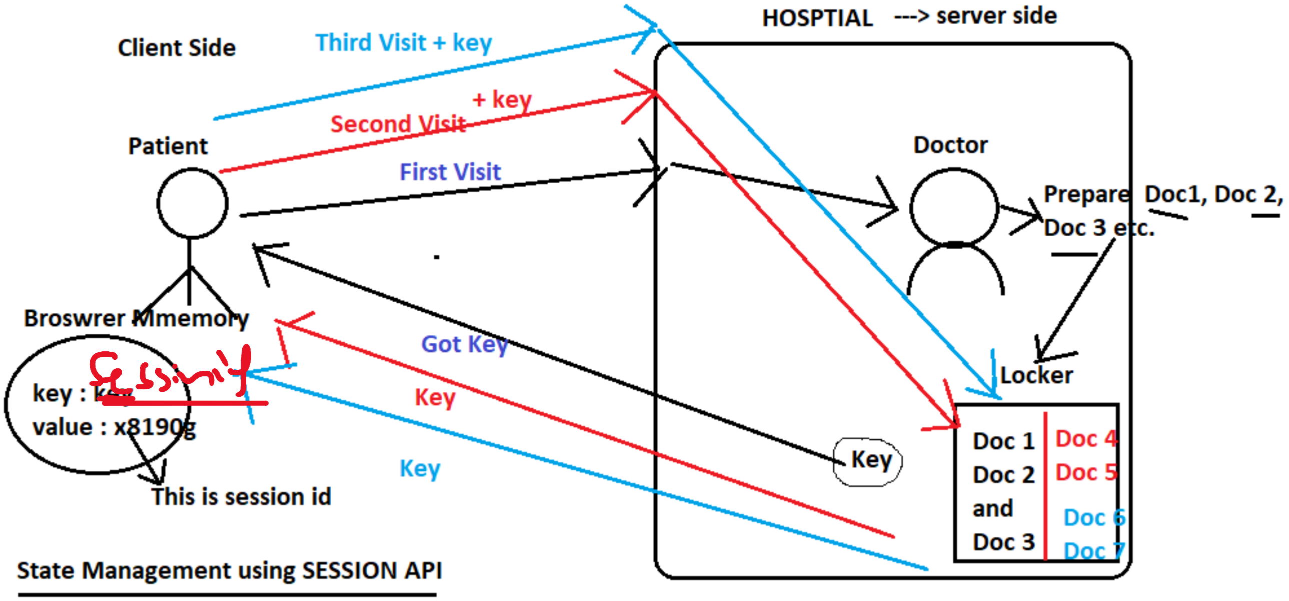
Chapter: 7.2(SESSION)/ State Management using SESSION API

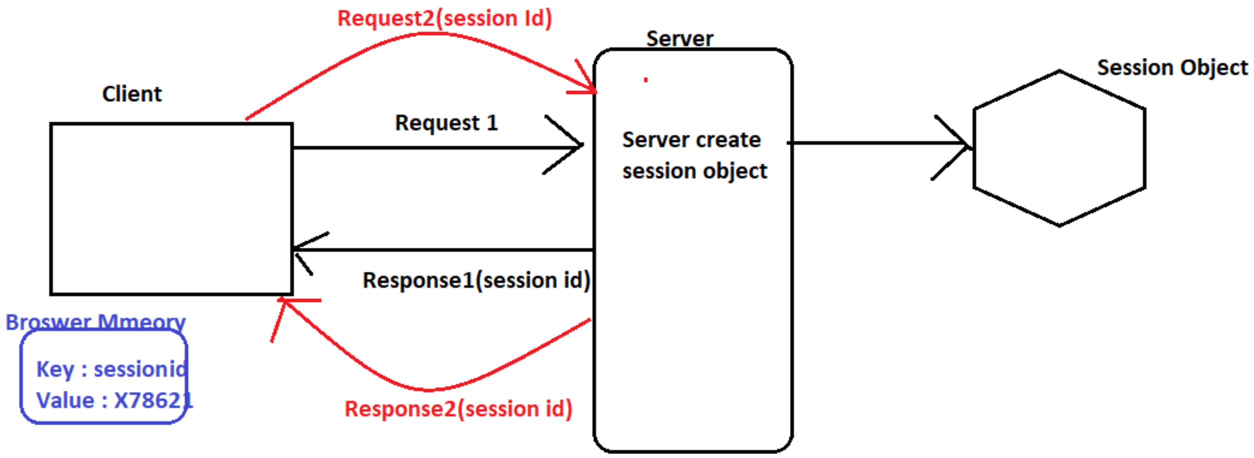
Drawback of cookies:

- Cookies hold less amount of information(4KB).
- Cookies can store only specific types of data (only str type).
- Cookies are not secure because they store data in plain text format.
- Any one can see the cookie and edit, delete the cookie.
- Every time along with request, cookies will go to the server that means if we send 1000 request then 1000-time cookies will be go along with request. (Cookie is always travel with request).

SESSION:

- **COOKIE** and **SESSION** both are state management techniques. But SESSION is a server-side state management technique.
- **SESSION** created by server and maintained by server but only the **session id** maintained by client.





What is session id?

A unique id available for each session object this is nothing but session id.

- The client is responsible to store the session id.
- Server is responsible to store the session object.
- Server always send session id as the part of response to the client. When a client send request to sever then that session id always travel with request and server can recognize user based on this session id.
- Session information can be stored inside files, database and cache.

Session Method

request.session.get()
request.session.keys()
request.session.items()
request.session.set_expiry()
request.session.get_ expire _date()
request.session.get_ expire _age()
request.session.get_expire_at_broswer_close()

P037
Project: Implementation of SESSION

Views.py

```
from django.shortcuts import render,HttpResponse

# Create your views here.
def create_session(request):
    request.session['name']='surendra'
    request.session['mark']=79
    request.session['course']='MTech'
    return HttpResponse('Session is set')
```

Note : To store session data we have to execute makemigrations and migrate command. Otherwise we will get no such table : django-session.

Views.py

```
from django.shortcuts import render,HttpResponse

# Create your views here.
def create_session(request):
    request.session['name']='surendra'
    request.session['mark']=79
    request.session['course']='MTech'
    return HttpResponse('Session is set')

def get_session(request):
    name=request.session['name']
    mark=request.session['mark']
    items=request.session.items()
    keys=request.session.keys()
    exp_date=request.session.get_expiry_date()
    exp_age=request.session.get_expiry_age()
    context={
        'name':name,
        'mark':mark,
        'items':items,
        'keys':keys,
        'exp_date':exp_date,
        'exp_age':exp_age,
    }
    return render(request,'SessionApp/get_session.html',context)

def delete_session(request):
    del request.session['mark']
    return HttpResponse('Session - mark deleted')

def page_count(request):
    count=request.session.get('count',0)
    newcount=count+1
    c=request.session['count']=newcount
    return HttpResponse(c)
```

get_session.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Session Information</h1>
    <p>Name : {{name}} </p>
    <p>Mark : {{mark}} </p>
    <p>Items: {{items}}</p>
    <p>Keys : {{keys}} </p>
    <p>Exp_date : {{exp_date}} </p>
    <p>Exp_age : {{exp_age}}</p>
    <p>----------------------</p>
</body>
</html>
```

urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('set_session/', views.create_session),
    path('get_session/',views.get_session),
    path('delete_session/',views.delete_session),
    path('page_count/',views.page_count),
]
```