Apr 20

- **views.py**

```
def display_items_view(request):
   return render(request,'testapp/displayitems.html')
```

- **displayitems.html**

```
<body>
   <h1>Shopping Cart Items</h1>
   {% if request.COOKIES %}
   <table border="3">
     <thead>
       <th>Item Name</th>
       <th>Quantity</th>
     </thead>
     {% for key,value in request.COOKIES.items %}
     {% if key != 'csrftoken'%}
     <tr>
       <td>{{key}}</td>
       <td>{{value}}</td>
     </tr>
     {% endif %}
     {% endfor %}
   </table>
   {% else %}
   <p>No Items in the shopping cart</p>
   {% endif %}
 </body>
```

- **urls.py**

```
path('displayitems/',views.display_items_view)
```

Limitations of Cookies:

1.By using cookies we can store very less amount of information. The size of cookies is fixed. Hence if we want to store huge amount of information then cookies is not best choice.

2.Cookies can hold only string information. If we want to store non-string objects we cannot use cookies.

3.Cookies information is stored on client side and hence there is no security.

4.Every time with every request, browser will send all cookies related to that application, which creates network traffic problems.

5.There is a limit on max number of cookies supported by browser.

To overcome these limitations, we should go for Session Framework.

Django provides in-built support for session api

```
INSTALLED_APPS = [
   'django.contrib.sessions',
]
```

```
MIDDLEWARE = [
            'django.contrib.sessions.middleware.SessionMiddleware',
]
```

Session Management by using session API(Django Session Framework)

Coockies---->RMP Doctor
Session Framework--->Bank Locker | Temple Chappal Stand | Super speciality hospitals

Useful methods for session management:

1). To add data to the session

request.session['key'] = value

2). To get the data from the session

value = request.session['key']

3).request.session.set_expiry(seconds)

set expiry time for the session.

4).request.session.get_expiry_age()

Returns expiry age in seconds(The number of seconds until this session expires)

5).request.session.get_expiry_date()

Return the date on which session will be expired.

App:

```
D:\Django_20MAR_7PM>django-admin startproject sessionproject5
D:\Django_20MAR_7PM>cd sessionproject5
D:\Django_20MAR_7PM\sessionproject5>py manage.py startapp testapp
```

-->Add app in settings.py
-->Makemigrations and migrate

- **views.py**

```
def page_count_view(request):
   print(request.COOKIES)
   count = request.session.get('count',0)
   count += 1
   request.session['count'] = count
   request.session.set_expiry(120)
   print(request.session.get_expiry_age())
   print(request.session.get_expiry_date())
   return render(request,'testapp/pagecount.html',{'count':count})
```

- **pagecount.html**

```
<body>
   <h1>The Page Count:<span>{{count}}</span></h1>
</body>
```

- **urls.py**

```
path('count/',views.page_count_view)
```