

Template

Home.html

<html>

<body>

    <form action="/product">

        <h2>

            ProductName <input type="text" name="t1"><br>

            Qty <input type="text" name="t2"><br>

            <input type="submit" value="add" name="b">

            <input type="submit" value="remove" name="b">

            <input type="submit" value="view" name="b">

            <input type="submit" value="update" name="b">

        </h2>

        {{output}}

    </form>

</body>

</html>

Views.py

from django.shortcuts import render

# Create your views here.

def productview(request):

    b=request.GET.get("b")

    pname=request.GET.get("t1")

    qty=request.GET.get("t2")

    if b=="add":

        request.session[pname]=qty

        msg="Product Added to Cart"

    elif b=="update":

        request.session[pname]=qty

        msg="Product Updated"

    elif b=="remove":

        request.session.\_\_delitem\_\_(pname)

        msg="Product Deleted"

    elif b=="view":

        msg=""

        for pname,qty in request.session.items():

            msg=msg+pname+" "+qty+"\n"

    return render(request,"home.html",context={"output":msg})

def home(request):

    return render(request,"home.html")

settings.py

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'app1',

]

import os.path

template\_path=os.path.join(BASE\_DIR,"templates")

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [template\_path],

        'APP\_DIRS': True,

        'OPTIONS': {

**Cookies**

A cookie is **a small piece of information which is stored in the client browser**. It is used to store user's data in a file permanently (or for the specified time). Cookie has its expiry date and time and removes automatically when gets expire. Django provides built-in methods to set and fetch cookie.

**HttpRequest.COOKIES**

A dictionary containing all cookies. Keys and values are strings.

**HttpResponse.set\_cookie**(***key***, ***value='')***

Sets a cookie

Templates

<html>

<body>

    <form action="/product">

        <h2>

            ProductName <input type="text" name="t1"><br>

            Qty <input type="text" name="t2"><br>

            <input type="submit" value="add" name="b">

            <input type="submit" value="remove" name="b">

            <input type="submit" value="view" name="b">

            <input type="submit" value="update" name="b">

        </h2>

        {{output}}

    </form>

</body>

</html>

Views.py

from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.

def productview(request):

    b=request.GET.get("b")

    pname=request.GET.get("t1")

    qty=request.GET.get("t2")

    response=render(request,"home.html")

    if b=="add":

        response.set\_cookie(pname,qty)

    elif b=="update":

        response.set\_cookie(pname,qty)

    elif b=="remove":

        response.set\_cookie(pname,qty,max\_age=0)

    elif b=="view":

        msg=""

        for pname,qty in request.COOKIES.items():

            msg=msg+pname+" "+qty+"\n"

        response=render(request,"home.html",context={"output":msg})

    return response

def home(request):

    return render(request,"home.html")

urls.py

from django.contrib import admin

from django.urls import path

from app1 import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('product/',views.productview),

    path('home/',views.home)

]

Settings.py

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'app1',

]

mport os.path

template\_path=os.path.join(BASE\_DIR,"templates")

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [template\_path],

        'APP\_DIRS': True,

        'OPTIONS': {