Puis dans static>js>main.js

console.log('Hello word!!!')

maintenant si vous acceder la console du site vous trouver le message hello word

modifier dans boutique.html

                    <button data-produit="{{produit.id}}" data-action="add" class="btn btn-outline-secondary add-btn update-chariot">Ajouter au panier</button>

Dans main.html ajouter script

<head>

  <script type="text/javascript">

    var user = '{{request.user}}'

  </script>

</head>

Modifier main.js

var updateBtns = document.getElementsByClassName('update-chariot')

*for*(i = 0; i < updateBtns.length; i++){

    updateBtns[i].addEventListener('click', function(){

        var produitId = this.dataset.produit

        var action = this.dataset.action

        console.log('produitId:', produitId, 'Action:', action)

        console.log('USER:', user)

*if* (user == 'AnonymousUser'){

            console.log('Utilisateur non authentifié')

        }*else*{

            console.log('Utilisateur authentifié....')

        }

    })

}

Dans boutique>views.py ajouter

*from* django.http *import* JsonResponse

def updateItem(request):

*return* JsonResponse('Article ajouté', safe=False)

dans boutique>urls.py ajouter

    path('update\_item/', views.updateItem, name="update\_item"),

puis acceder a <http://127.0.0.1:8080/update_item/>

ajouter dans main.js

function updateUserCommande(produitId, action){

    console.log('Utilisateur est authentifié, envoie des données ....')

    var url = '/update\_item/'

    fetch(url, {

        method:'POST',

        headers:{

            'Content-Type':'application/json'

        },

        body:JSON.stringify({'produitId': produitId, 'action':action})

    })

    .then((response) =>{

*return* response.json()

    })

    .then((data) =>{

        console.log('data:', data)

    })

}

Si ouvrer la console vous trouver error token

Dans la documentation de django chercher : The CSRF token cookie is named csrftoken

Et vous trouver le code

Puis modifier main.html

   <head>

…

        <script type="text/javascript">

            var user = '{{request.user}}'

            function getCookie(name) {

                  var cookieValue = null;

*if* (document.cookie && document.cookie !== '') {

                      var cookies = document.cookie.split(';');

*for* (var i = 0; i < cookies.length; i++) {

                          var cookie = cookies[i].trim();

*// Does this cookie string begin with the name we want?*

*if* (cookie.substring(0, name.length + 1) === (name + '=')) {

                              cookieValue = decodeURIComponent(cookie.substring(name.length + 1));

*break*;

                          }

                      }

                  }

*return* cookieValue;

            }

            var csrftoken = getCookie('csrftoken');

          </script>

    </head>

Maintenant l’erreur a disparue

Modifier le nom de la fonction dans main.html

function getToken(name) {

et

var csrftoken = getToken('csrftoken');

modifier dans boutique>views.py

*import* json

def updateItem(request):

     data = json.loads(request.body)

     produitId = data['produitId']

     action = data['action']

     print('Action:', action)

     print('produitId:', produitId)

     client = request.user.client

     produit = Produit.objects.get(id=produitId)

     commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

     articlecom, created = ArticleCom.objects.get\_or\_create(commande=commande, produit=produit)

*if* action == 'add':

          articlecom.quantite = (articlecom.quantite + 1)

*elif* action == 'remove':

          articlecom.quantite = (articlecom.quantite - 1)

     articlecom.save()

*if* articlecom.quantite <= 0:

          articlecom.delete()

*return* JsonResponse('article est ajouté', safe=False)

dans main.js modifier

.then((data) =>{

        console.log('data:', data)

        location.reload()

    })

Dans boutique>views.py modifier

def boutique(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

          chariotItems = commande.get\_chariot\_items

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0}

          chariotItems = commande['get\_chariot\_items']

     produits = Produit.objects.all()

     context = {'produits':produits, 'chariotItems':chariotItems}

*return* render(request, 'boutique/boutique.html', context)

Modifier main.html

     <div class="form-inline my-2 my-lg-0">

                    <a href="{% url 'chariot' %}">

                      <img  id="chariot-icon" src="{% static 'images/cart.png' %}">

                    </a>

                    <p id="chariot-total">{{chariotItems}}</p>

                    <li><i class='bx bx-moon change-theme' id="theme-button">Bismillah</i></li>

                </div>

Modifier chariot.html

<div style="flex:1"><p>{{item.get\_total|floatformat:2}}$</p></div>

modifier chariot.html

 <div class="quantite">

                                    <img  data-produit={{item.produit.id}} data-action="add" class="chg-quantite update-chariot" src="{% static  'images/arrow-up.png' %}">

                                    <p class="quantite">Kilo</p>

                                    <img data-produit={{item.produit.id}} data-action="remove" class="chg-quantite update-chariot" src="{% static  'images/arrow-down.png' %}">

                                </div>

Dans models.py modifier

class Commande(models.Model):

    …

    @property

    def shipping(self):

        shipping = False

        article\_coms = self.articlecom\_set.all()

*for* i *in* article\_coms:

*if* i.produit.digital == False:

                shipping = True

*return* shipping

dans views.py modifier> boutique

def boutique(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

          chariotItems = commande.get\_chariot\_items

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0, 'shipping':False}

          chariotItems = commande['get\_chariot\_items']

     produits = Produit.objects.all()

     context = {'produits':produits, 'chariotItems':chariotItems}

*return* render(request, 'boutique/boutique.html', context)

def chariot(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

          chariotItems = commande.get\_chariot\_items

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0, 'shipping':False}

          chariotItems = commande['get\_chariot\_items']

     context = {'items':items, 'commande':commande, 'chariotItems':chariotItems}

*return* render(request, 'boutique/chariot.html', context)

def caisse(request):

*if* request.user.is\_authenticated:

          client = request.user.client

          commande, created = Commande.objects.get\_or\_create(client=client, etat=False)

          items = commande.articlecom\_set.all()

          chariotItems = commande.get\_chariot\_items

*else*:

          items = []

          commande = {'get\_chariot\_total':0, 'get\_chariot\_items':0, 'shipping':False}

          chariotItems = commande['get\_chariot\_items']

     context = {'items':items, 'commande':commande, 'chariotItems':chariotItems}

*return* render(request, 'boutique/caisse.html', context)