Code Documentation

Login/signin :

def signin(request):

    if request.method == "POST":

        username = request.POST['username']

        pass1 = request.POST['pass1']

        user = authenticate(username=username,password=pass1)

        if user is not None:

            login(request,user)

            fname = user.first\_name

            products = Product.objects.all()

            return  redirect('index')# render(request, "index.html",{'fname':fname,'products':products})

        else :

            messages.error(request,"Bad Credentials!")

            return redirect('index')

    return render(request,"login.html")

Explanation : If method is get then Login form Template is rendered ,if method is post then first User is authenticated then Redirected to the Index page.

Signup :

def signup(request):

    if request.method == "POST":

        username = request.POST['username']

        fname = request.POST['fname']

        lname = request.POST['lname']

        email = request.POST['email']

        pass1 = request.POST['pass1']

        pass2 = request.POST['pass2']

        if User.objects.filter(username = username ):

            messages.error(request,"Username already exist! please try anothrt one")

            return redirect('index')

        if User.objects.filter(email=email):

            messages.error(request,"Email already exist! please try anothrt one")

            return redirect('index')

        if len(username) >10 :

            messages.error(request,"Username must be under 10 charecters")

            #return redirect('home')

        if pass1 != pass2 :

            messages.error(request,"password is not match !")

        if not username.isalnum():

            messages.error(request,"Username must be Alpha-Numaric")

            return redirect('index')

        myuser = User.objects.create\_user(username,email,pass1)

        myuser.first\_name = fname

        myuser.last\_name = lname

        #myuser.is\_active = False

        myuser.is\_active = True

        myuser.save()

        messages.success(request,"your account has been successfully created. wer have send you a cnformation email please confirm your email ")

        return redirect("signin")

    return render(request,"signup.html")

Explanation: If method is get then signup Template/form is rendered for user Signup. if method is post then new user is created.

Signout :

def signout(request):

    logout(request)

    messages.success(request,"logged out successfully!")

    return redirect('index')

Explanation: When the signout endpoint is called the current authenticated user is will be logout.

Checkout :

def checkout(request):

    if request.method=="POST":

        user = request.user.username

        items\_json = request.POST.get('itemJson','')

        mylist = []

        items\_json1 = json.loads(items\_json)

        print(items\_json)

        for item in items\_json1:

            mylist.append(items\_json1[item])

        mylist = json.dumps(mylist)

        name = request.POST.get('name','')

        email = request.POST.get('email','')

        address = request.POST.get('address1','') + " " + request.POST.get('address2','')

        city = request.POST.get('city','')

        state = request.POST.get('state','')

        zip\_code = request.POST.get('zip','')

        phone = request.POST.get('phone','')

        grand\_total = request.POST.get('grand\_total','')

        order = Order(items\_json=mylist,name=name,email=email,address=address,city=city,state=state,zip\_code=zip\_code,phone=phone,user=user,grand\_total=grand\_total)

        order.save()

        thank = True

        return render(request,'checkout.html',{'thank':thank})

    return render(request,"checkout.html",{})

Explanation: if the method is get then checkout template is rendered. If post then order is stored in database with all the data User’s data.

Product Model :

# Create your models here.

class Product (models.Model):

    product\_id = models.AutoField

    product\_name = models.CharField(max\_length=50,default="")

    desc = models.CharField(max\_length=50,default="")

    p\_img\_1 = models.ImageField(upload\_to='static/images/',default="")

    product\_price = models.IntegerField(default=0)

    date\_added = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return self.product\_name

    class Meta:

        ordering = ['-date\_added']

Explanation: Class model for Product which have fields product id, product Name, description, image, product price, and date added at which date the product is added.

Model Order :

class Order (models.Model):

    order\_id = models.AutoField(primary\_key=True)

    user = models.CharField(max\_length=50)

    name = models.CharField(max\_length=50)

    email = models.CharField(max\_length=50)

    address = models.CharField(max\_length=120)

    city = models.CharField(max\_length=50)

    state = models.CharField(max\_length=50)

    zip\_code = models.CharField(max\_length=50)

    phone = models.CharField(max\_length=50)

    grand\_total = models.CharField(max\_length=50)

    items\_json = models.CharField(max\_length=120)

    date\_added = models.DateTimeField(auto\_now\_add=True)

    class Meta:

        ordering = ['-date\_added']

Explanation: Class model for Order which have fields order id, user, name, email, address, state, city, zip code, phone total, date added at which date the product is added.