1) Lo primero que haremos es modicar nuestro models.py y agregamos la librerías del modelo de usuario:

from django.contrib.auth.models import User

y ademas, agregamos la relacion 1 a 1:

*class* costumer(*models*.*Model*):

    user= models.OneToOneField(User, null, *on\_delete*=models.CASCADE)

    name = models.CharField(*max\_length*=100,*null*=True)

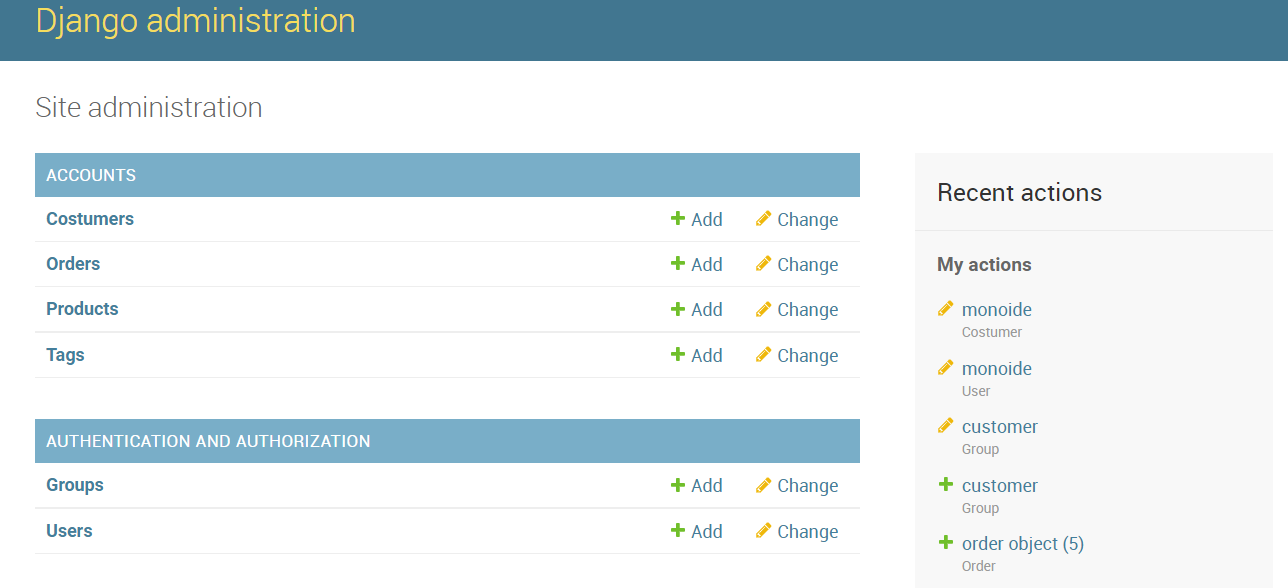
    phone = models.CharField(*max\_length*=100, *null*=True)

    email = models.EmailField(*max\_length*=100, *null*=True)

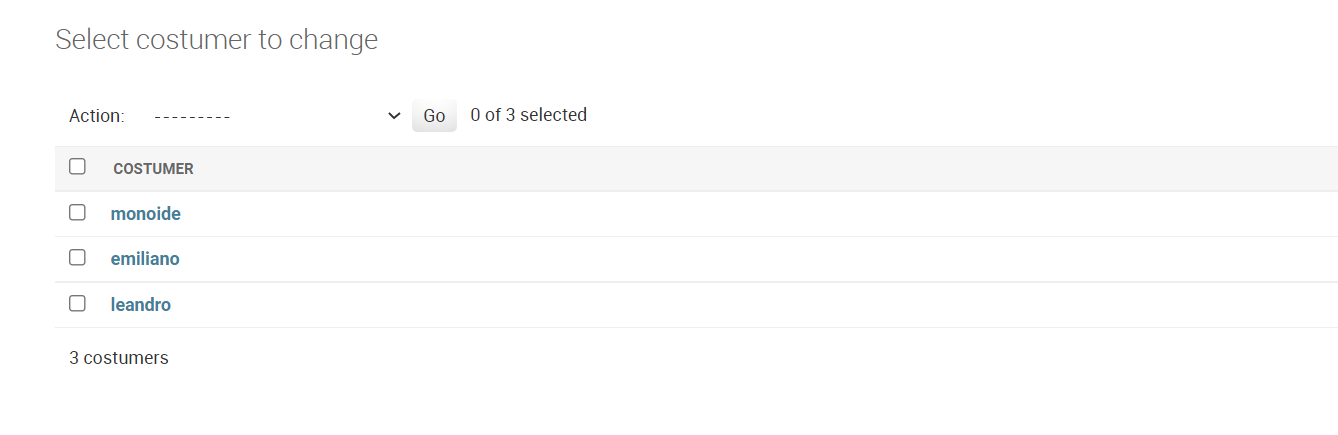
    date\_created = models.DateField(*auto\_now\_add*=True)

ejecutamos con makemigration y migrate, luego corremos le servidor.

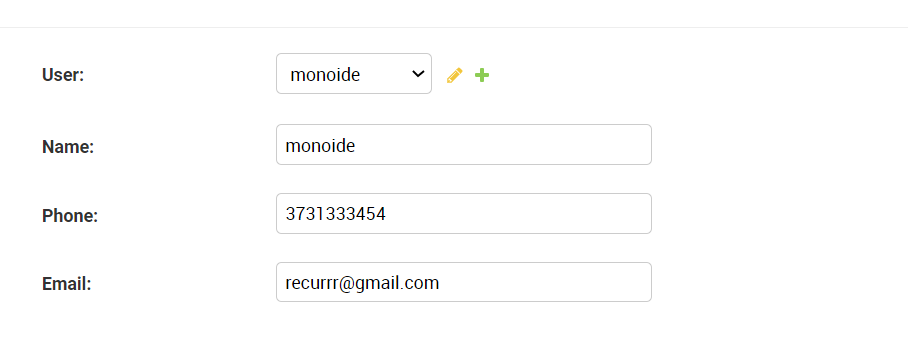
2) Ahora nos vamos al panel de admin y nos logueamos como administrador, nos aparece la siguiente información:



Nos vamos a los costumer, ya que ahí fue donde modificamos nuestro modelo a ser una relación de 1 a 1. Nos mostrara los siguiente información:

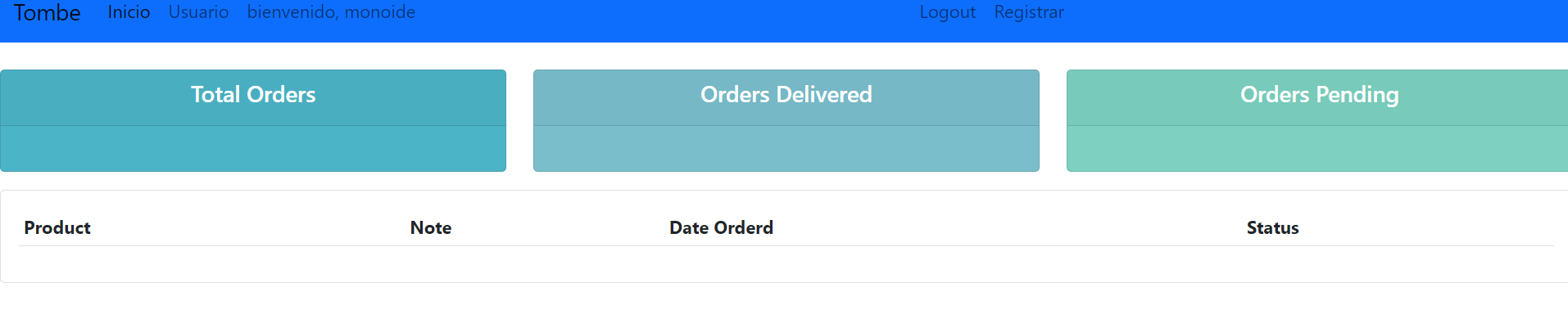


Dentro del ejemplo, pasamos a modificar el cliente monoide.



Y dentro de monoide, ponemos como user ‘monoide’ que asi fue como creamos nuestro usuario. Ahora con esta información lo hacemos es que nos muestre solamente su estadísticas en cuenta a las ordenes creadas.

3) Creamos un html llamada user.html que refleje solamente las características del usuario en si, una vez obtenido esto. Podemos crear una solapa de usuario con la siguiente información



Lo que nos falta ahora mismo, es modificar la función de userpage, que hace referencia a la pagina del usuario y agregar lo siguiente:

*def* userPage(*request*):

    orders= request.user.costumer.order\_set.all()

    print('ORDERS:', orders)

    total\_orders = orders.count()

    pending= orders.filter(*status*='Pending').count()

    delivered= orders.filter(*status*='Delivered').count()

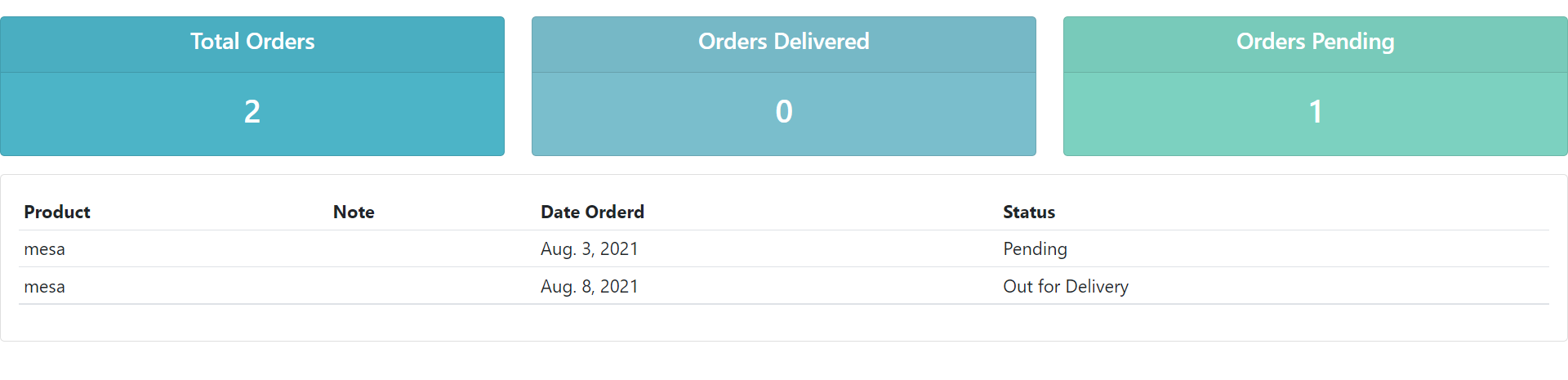
    context ={'orders': orders, 'total\_orders': total\_orders,'pending': pending, 'delivered': delivered}

    return render(request,'user.html',context)

La consulta orders del principcio de la funcion nos muestra el listado de solamente el usuario registrado, y no de todos como se definio anteriormente en la pagina de estadísticas, por esa razón tuvimos que crear una nueva pagina llamada user.html

Después con total\_orders, obtenemos las ordenes totales, con pending las ordenes pendientes y deliveres las ordenes que fueron entegradas.

Lo que nos muestra, será la siguiente info dentro de los visual en html:



4) Si queremos agregar al cliente como un usuario cliente ni bien se registra, lo que podemos hacer es modificar la función registro y agregar la siguiente información:

*def* registerPage(*request*):

    form = CreateUserForm()

    if request.method=='POST':

        form = CreateUserForm(request.POST)

        if form.is\_valid():

            user =form.save()

            username = form.cleaned\_data['username']

            group = Group.objects.get(*name*='prueba')

            user.groups.add(group)

           costumer.objects.create(*user*=user,*name*=user.username,*email*=user.email)

campo agregado, pasando las intancias de username y email

            messages.success(request, 'la carga ha sido exitosa' + ' ' + username)

            return redirect('login')

    context ={'form':form}

    return render(request,'register.html',context)

pasamos el user definido en cliente como la relacion 1 a 1