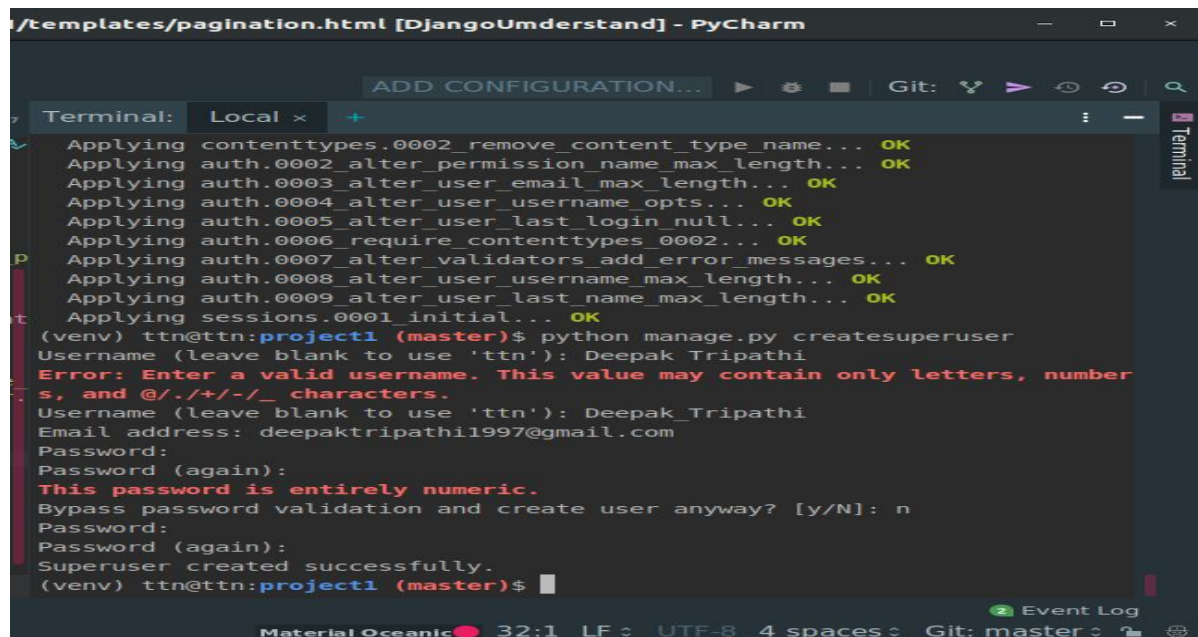
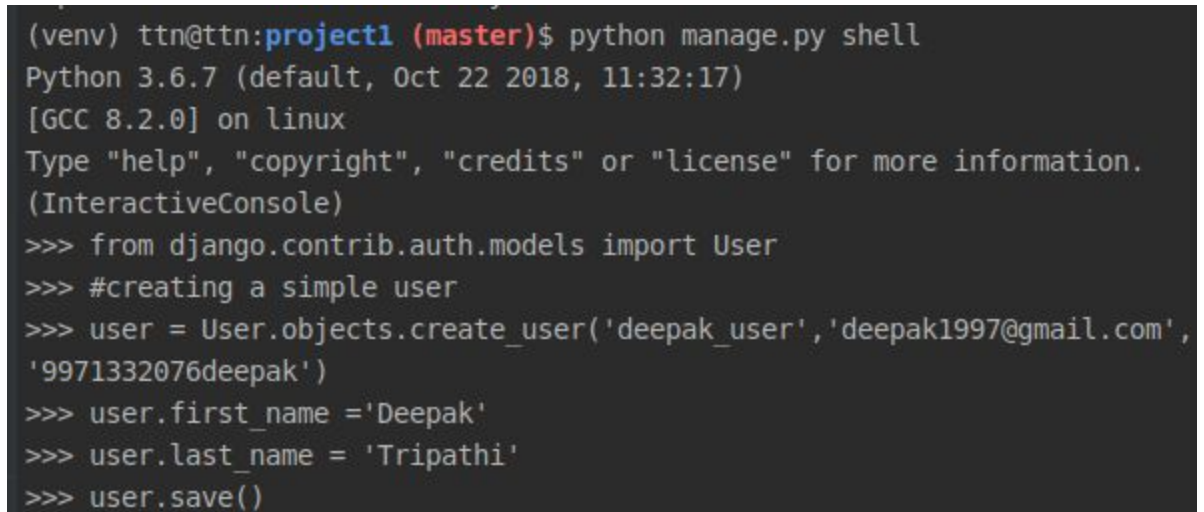


Q1 Create 2 users in django app (by any way either through command line) or Django Admin. One Of them should be SuperUser and Other Should Be Staff user.



```
/templates/pagination.html [DjangoUmderstand] - PyCharm
ADD CONFIGURATION...
Terminal: Local x +
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying sessions.0001_initial... OK
(venv) ttn@ttn:project1 (master)$ python manage.py createsuperuser
Username (leave blank to use 'ttn'): Deepak Tripathi
Error: Enter a valid username. This value may contain only letters, number
s, and @/./+/-/_ characters.
Username (leave blank to use 'ttn'): Deepak_Tripathi
Email address: deepaktripathi1997@gmail.com
Password:
Password (again):
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: n
Password:
Password (again):
Superuser created successfully.
(venv) ttn@ttn:project1 (master)$
```

SUPERUSER

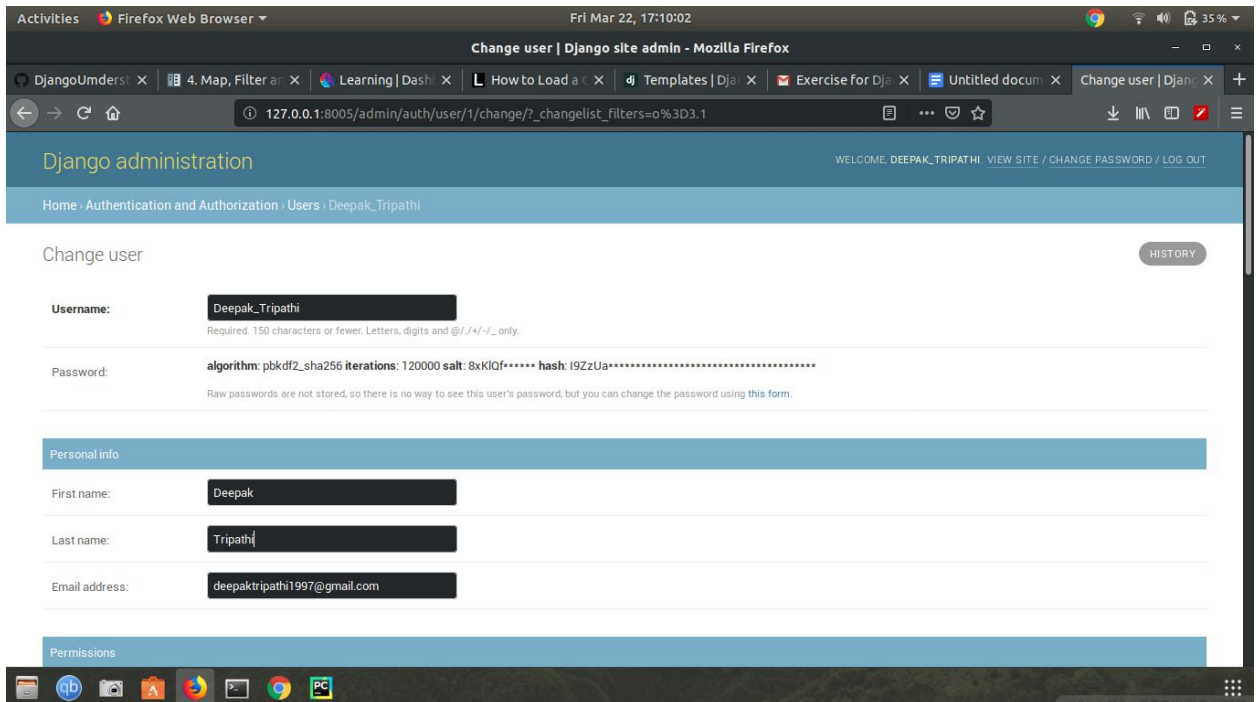


```
(venv) ttn@ttn:project1 (master)$ python manage.py shell
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from django.contrib.auth.models import User
>>> #creating a simple user
>>> user = User.objects.create_user('deepak_user','deepak1997@gmail.com',
'9971332076deepak')
>>> user.first_name = 'Deepak'
>>> user.last_name = 'Tripathi'
>>> user.save()
```

Non Staff User

```
>>> user.__dict__
{'_state': <django.db.models.base.ModelState object at 0x7f6f38d77cc0>, 'id': 2, 'password': 'pbkdf2_sha256$120000$tf25cYD9qrEY$AbestSd5z85z2Eia4lEV78nP8IFv8Ewy204YmIO5MRg=', 'last_login': None, 'is_superuser': False, 'username': 'deepak_user', 'first_name': 'Deepak', 'last_name': 'Tripathi', 'email': 'deepak1997@gmail.com', 'is_staff': False, 'is_active': True, 'date_joined': datetime.datetime(2019, 3, 22, 9, 6, 17, 731184, tzinfo=<UTC>), '_password': None}
>>> user.is_staff = True
>>> user.save()
```

Staff User but not Superuser

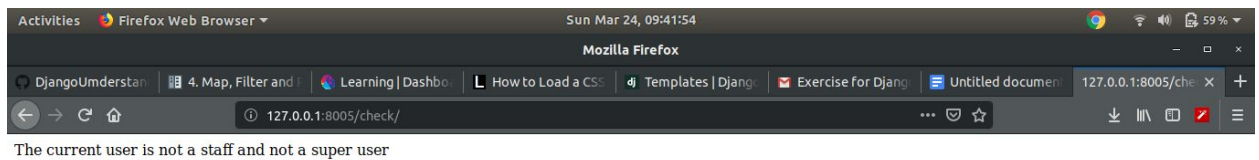


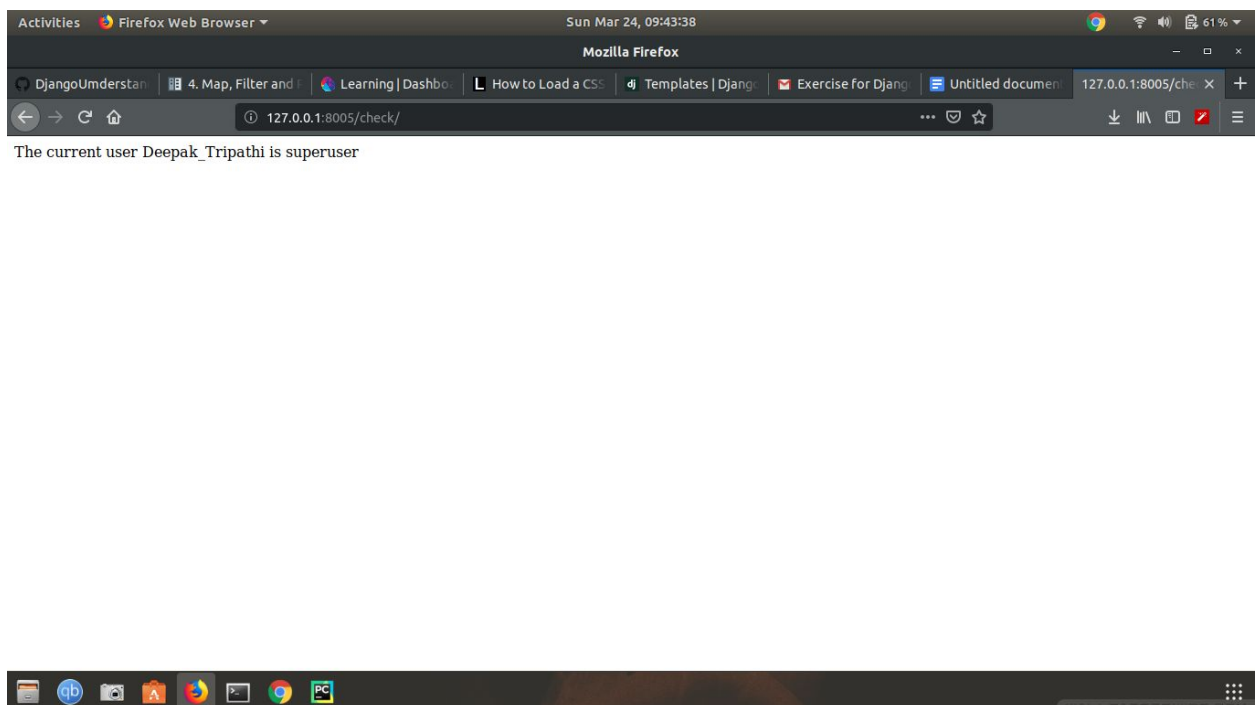
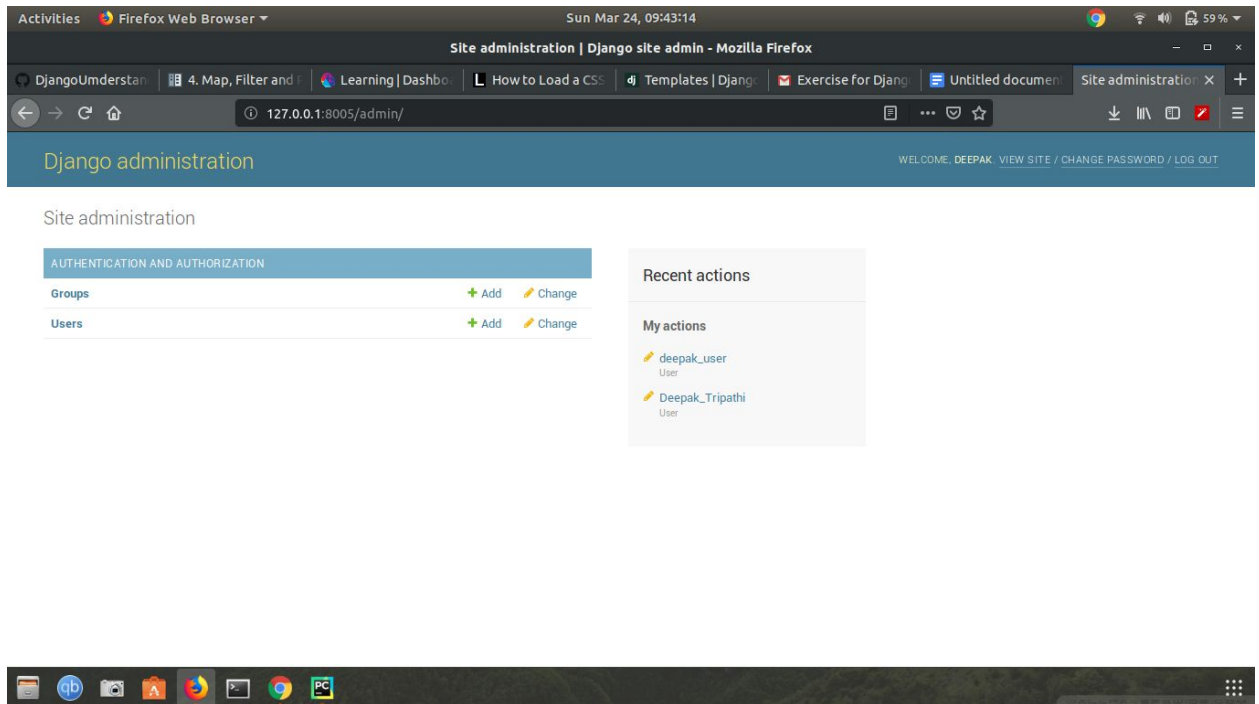
The screenshot shows the Django administration interface in a Mozilla Firefox browser. The page title is 'Change user | Django site admin - Mozilla Firefox'. The URL is '127.0.0.1:8005/admin/auth/user/1/change/?_changelist_filters=o%3D3.1'. The page content includes a welcome message for 'DEEPAK_TRIPATHI' and a breadcrumb trail: 'Home > Authentication and Authorization > Users > Deepak_Tripathi'. The main section is titled 'Change user' and contains a 'HISTORY' button. The user details are as follows:

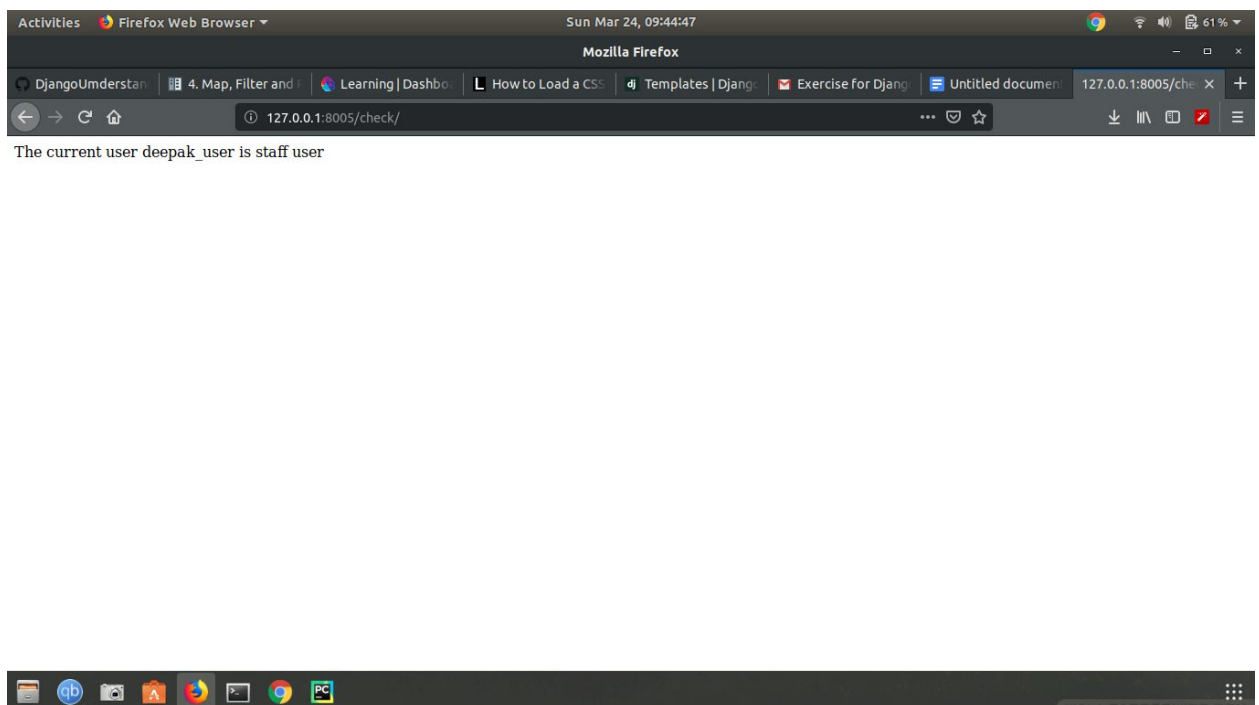
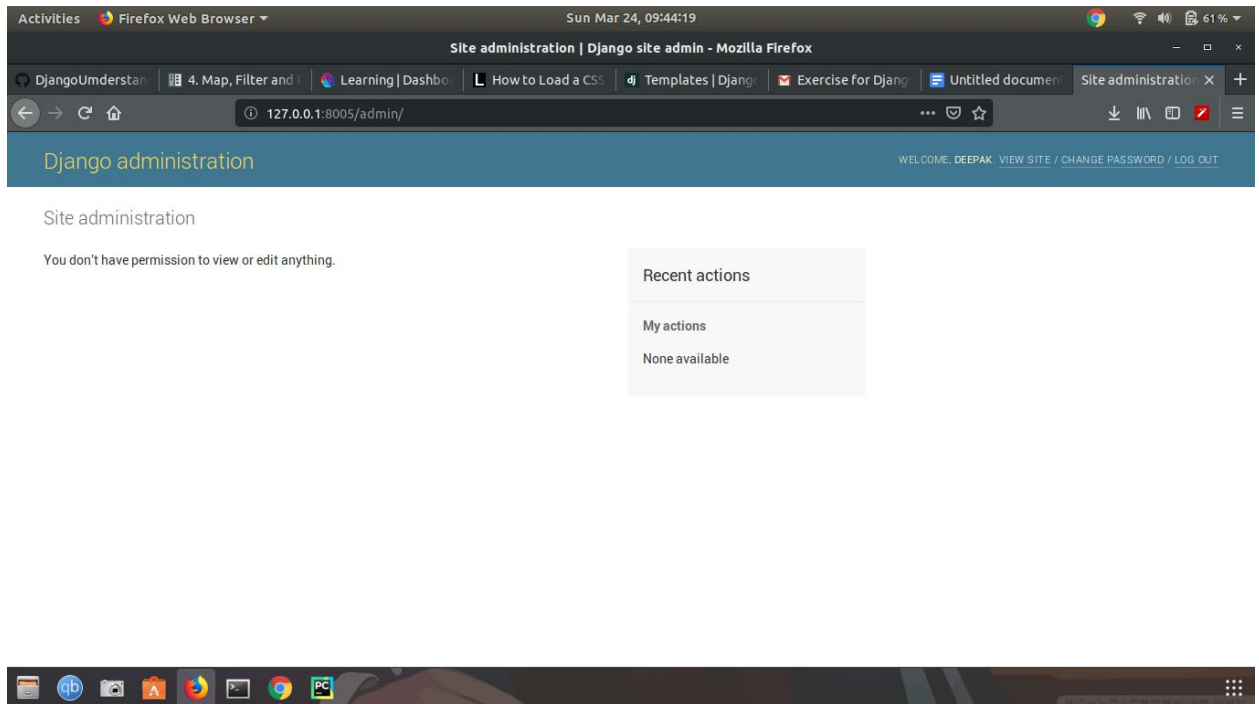
Field	Value
Username:	Deepak_Tripathi
Password:	algorithm: pbkdf2_sha256 iterations: 120000 salt: 8xKIQf***** hash: 19ZzUa***** Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.
Personal info	
First name:	Deepak
Last name:	Tripathi
Email address:	deepaktripathi1997@gmail.com
Permissions	

Create a view in your application which will authenticate the user and identify if is superuser or staff user and renders the string 'is super' on browser in case of super user and 'is staff' in case of staff user.

```
def do_authentication(request):  
  
    curr_user = request.user  
  
    if curr_user.is_superuser:  
  
        return HttpResponse("The current user "+curr_user.username+" is superuser")  
  
    elif curr_user.is_staff:  
  
        return HttpResponse("The current user "+curr_user.username+" is staff user")  
  
    else:  
  
        return HttpResponse("The current user is not a staff and not a super user")
```







Success full logout the user when someone make call on url: /logout/ of your app.

```
def do_logout(request):
```

```
    curr_user = request.user
```

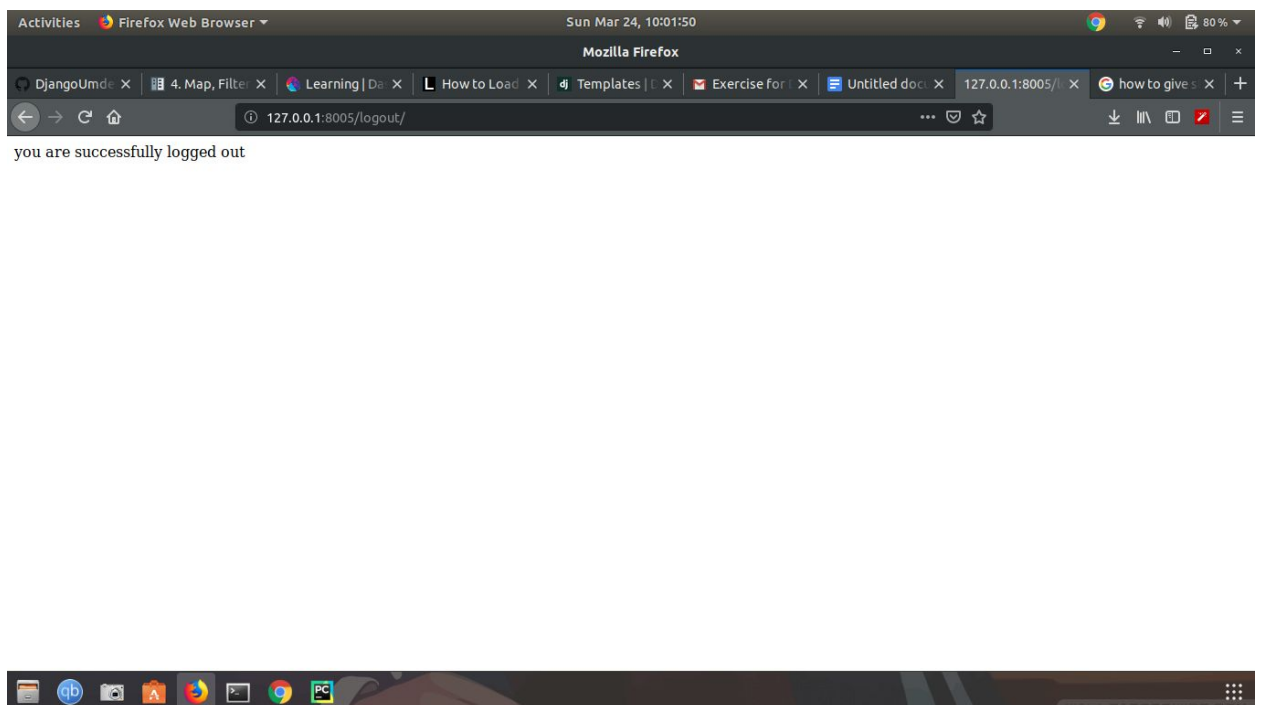
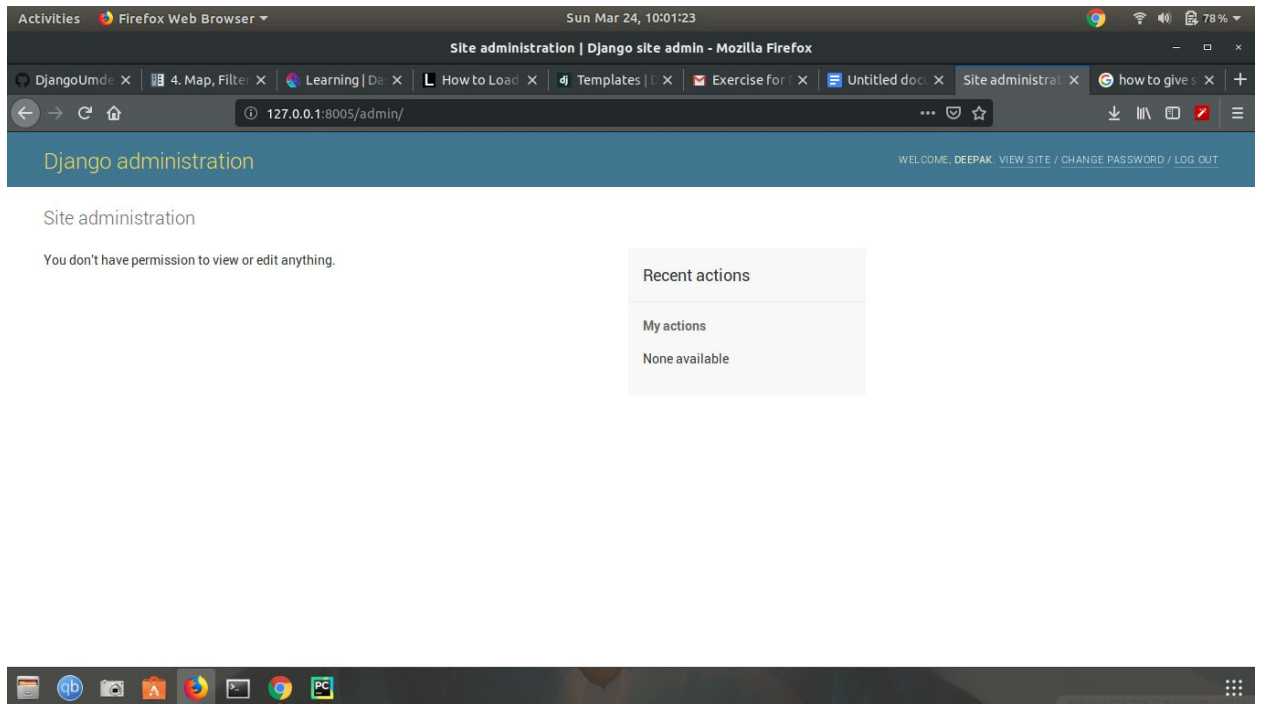
```
    if curr_user.is_active:
```

```
        logout(request)
```

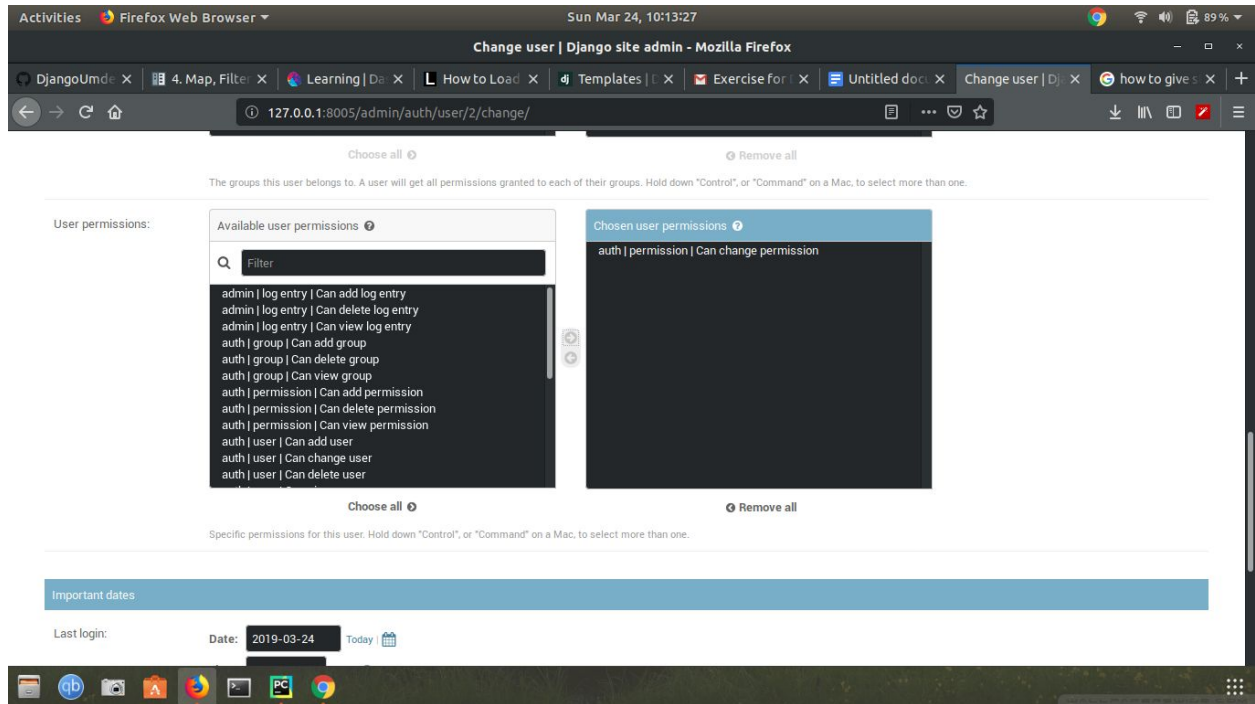
```
        return HttpResponseRedirect("you are successfully logged out")
```

```
    elif not curr_user.is_staff and not curr_user.is_superuser:
```

```
        return HttpResponseRedirect("you need to signup first")
```



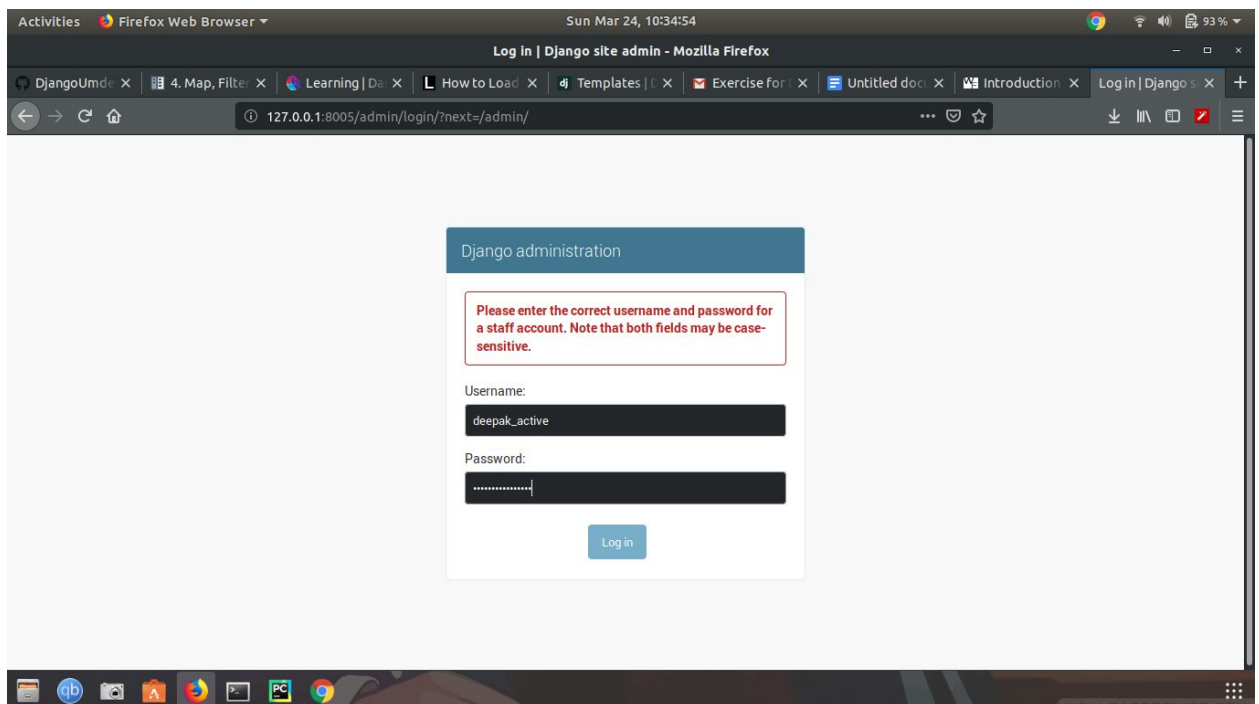
Give Staff user “can change” permission.

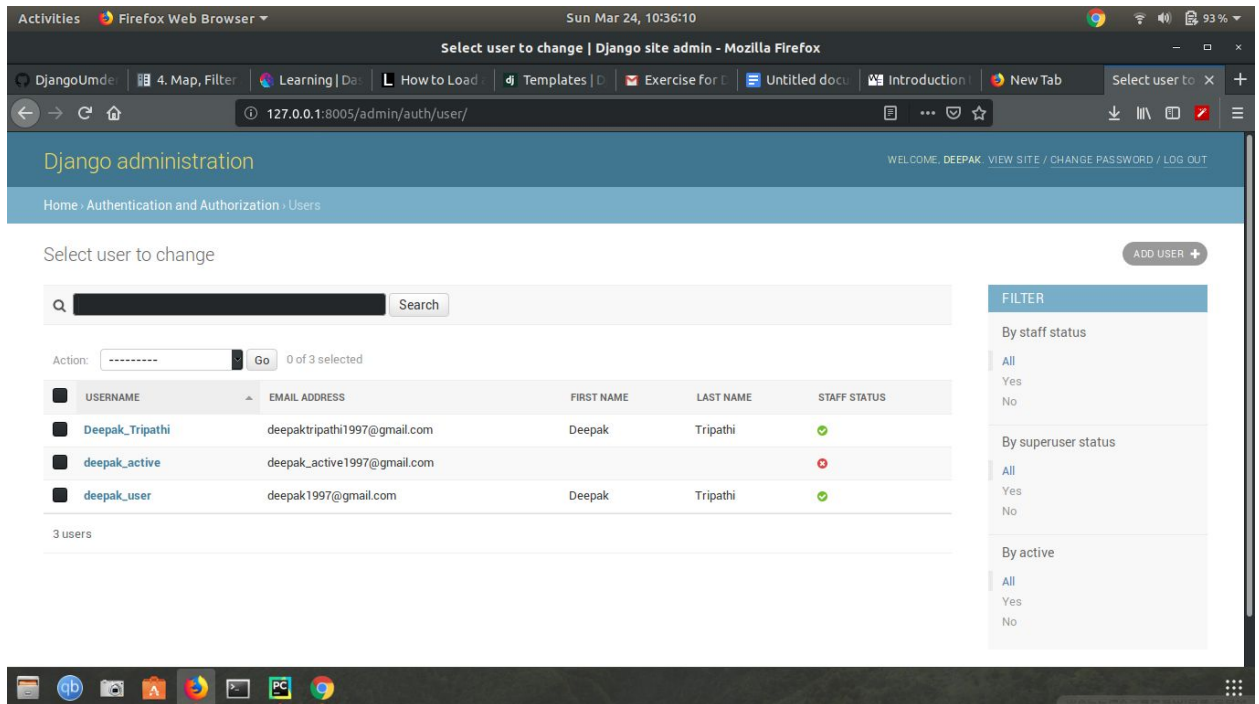


Create A third User which is an active user neither superuser or staff user. Try to login with that user in django Admin, and describe the difference between this non super non staff user and Staff user.


```
(venv) ttn@ttn:project1 (master)$ python manage.py shell
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more i
nformation.
(InteractiveConsole)
>>> from django.contrib.auth.models import User
```

```
>>> user1 = User.objects.create_user('deepak_active','deepa
k_active1997@gmail.com','9971332076deepak')
>>> user1.save()
>>> user1.is_staff
False
>>> user1.is_active
True
>>> 
```





If logged in as a superuser, you have access to create, edit, and delete any object (models).

You can create staff user using staff flag. The “staff” flag controls whether the user is allowed to log in to the admin interface (i.e., whether that user is considered a “staff member” in your organization). Since this same user system can be used to control access to public (i.e., non-admin) sites, this flag differentiates between public users and administrators.

“Normal” admin users – that is, active, non-superuser staff members – are granted admin access through assigned permissions. Each object editable through the admin interface has three permissions: a create permission, an edit permission and a delete permission for all the models you had created.

Django’s admin site uses a permissions system that you can use to give specific users access only to the portions of the interface that they need. When you create a user, that user has no permissions, and it’s up to you to give the user specific permission

