Day 25: Admin panel list(fetching user by type)
Date: 18 Baisakh, Monday

1)queryset = User.objects.filter(is_staff=True)

Adminuser list

```
 def list(self, request, *args, **kwargs):
 users = self.get_queryset()
       adminuser = self.get_serializer(users, many=True)
```

Explanation

self.get_queryset() retrieves the queryset of objects from the model based on the view's filtering or search criteria.

self.get_serializer() instantiates a serializer object that will convert the queryset into a format that can be returned in the HTTP response.

adminuser = self.get_serializer(users, many=True) creates a serialized representation of the queryset of objects, where many=True specifies that there can be multiple objects in the queryset. The serialized data is stored in the adminuser variable.

StudentUser list

```
student = UserSerializer(
          User.objects.filter(
             pk__in=list(
                map(
                   lambda x: x.get("user"),
                   Recruiter.objects.all().values("user"),
                )
             )
          ),
          many=True,
       )
```

Explanation:

```
  pk__in=list(
                map(
                   lambda x: x.get("user"),
                   Recruiter.objects.all().values("user"),
                )
             )
```

Output: [23]

Recruiter filter get a query set with user value. <QuerySet [{'user': 94}]>.Then lamda function get the value i.e. 23. List make it in list [23].

```
user = User.objects.filter(pk__in=[23])
Student = UserSerializer(user, many=True)
```

This code creates a serialized representation of a queryset .

The __in lookup is used to specify a list of values to match against a collection.