

## CUSTOMIZATION Of Django Admin Site:

Here you are going to learn How to override default logics of admin site and make admin site as more admin friendly ?

Open admin.py file and create the required overriding logics to interacting admins with admin site for easy access our application data.

Create user defined admin class and add all overding fields in this class.

### admin.py

```
from django.contrib import admin
from .models import Student
```

```
class StudentAdmin(admin.ModelAdmin):
    list_display = ['sno', 'sname', 'marks']
    list_editable = ['sname']
    search_fields = ['marks']
    list_display_links = ['marks']
    list_per_page = 2
```

```
admin.site.register(Student, StudentAdmin)
```

Here,

### list\_display

- represents the model fields which are displaying on admin site presentation.

### list\_editable

- represents the model fields which are displaying on admin site as an editable field option.

### search\_fields

- represents the model fields which are used for searching the table data on admin site presentation.
- It displays one "search field box and button" on top of our table data. By enter any value we can filter all related values of filter.

### list\_display\_links

- represents the model fields which are used as a links on tables and when we clicking the specific filed link then it opens a updating form for updating data.

### **list\_per\_page**

- represents how many records we want to display on each page. It display pagination effect for admin site presentation.
- If you want to see all records then click on "show all" link in pagination style links then you will get all records at a time.

### **list\_filter**

- represents the fields names what are used for filtering purpose on admin site presentation.
- It displays Filter Menu with all filter field values on right side edges bar. By clicking any values then we can filter the exact value.

### **fields = ('fieldname1','fieldname2',....)**

- using "fields" attribute we can only display required fields only on form for updating or deleting on admin site.

### **exclude = ('fieldname1','fieldname2',....)**

- using "exclude" attribute we can display all fields except given specified fields on form for updating or deleting on admin site.

### **readonly\_fields = ('fieldname1','fieldname2',....)**

- using "readonly\_fields" attributes we can specify the fields which are only readable and we can not update these fields using admin update form.

### **Q. How add a model table into admin site ?**

By using register() we can add our required models on admin site.

```
admin.site.register(ModelName)
```

### **Q. How to disable a model table from admin site ?**

By using unregister() we can not add (or) disable our required model tables on admin site.

```
admin.site.unregister(ModelName)
```

ModelAdmin.actions\_on\_top and ModelAdmin.actions\_on\_bottom

Controls where on the page the actions bar appears.

By default, the admin changelist displays actions at the top of the page like(actions\_on\_top = True; actions\_on\_bottom = False).

Means on admin site Action bar displays for deleting selected rows on top side.  
If you want bottom side the make True for bottom.

### **Q. How to change the Django Administration Header, Title and Index Title?**

In urls.py you can override the 3 most important variables to overcome the default admin site related header, title text changing.

Open project level urls.py and add bellow code on top of urlpatterns or over.

```
admin.site.site_header = 'Student project'    # default: "Django Administration"
```

```
admin.site.index_title = 'Student Information' # default: "Site administration"
```

```
admin.site.site_title = 'Student Admin'      # default: "Django site admin"
```