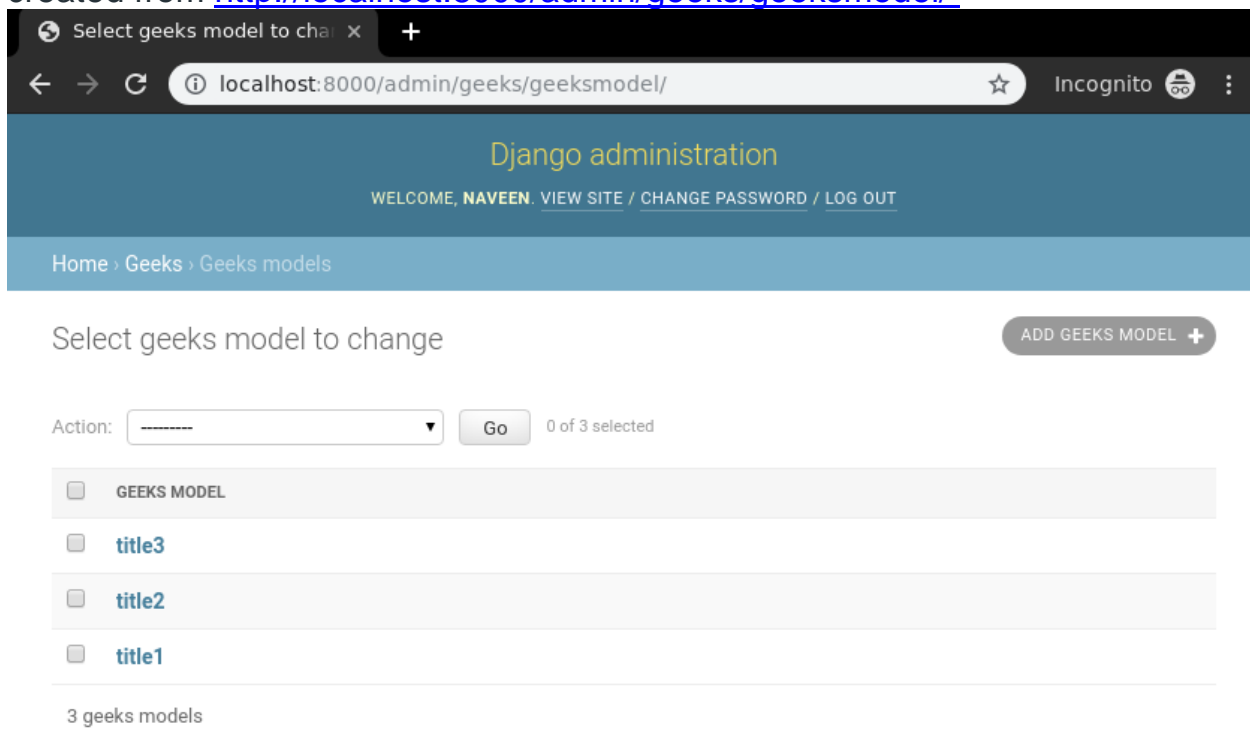




```
>>> GeeksModel.objects.create(
    title="title2",
    description="description2").save()
>>> GeeksModel.objects.create(
    title="title2",
    description="description2").save()
```

Now we have everything ready for back end. Verify that instances have been created from <http://localhost:8000/admin/geeks/geeksmodel/>



Class Based Views automatically setup everything from A to Z. One just needs to specify which model to create DetailView for, then Class based DetailView will automatically try to find a template in `app_name/modelname_detail.html`. In our case it is `geeks/templates/geeks/geeksmodel_detail.html`. Let's create our class based view. In `geeks/views.py`,

```
from django.views.generic.detail import DetailView
```

```
from .models import GeeksModel

class GeeksDetailView(DetailView):
    # specify the model to use
    model = GeeksModel
```

Now create a url path to map the view. In `geeks/urls.py`,

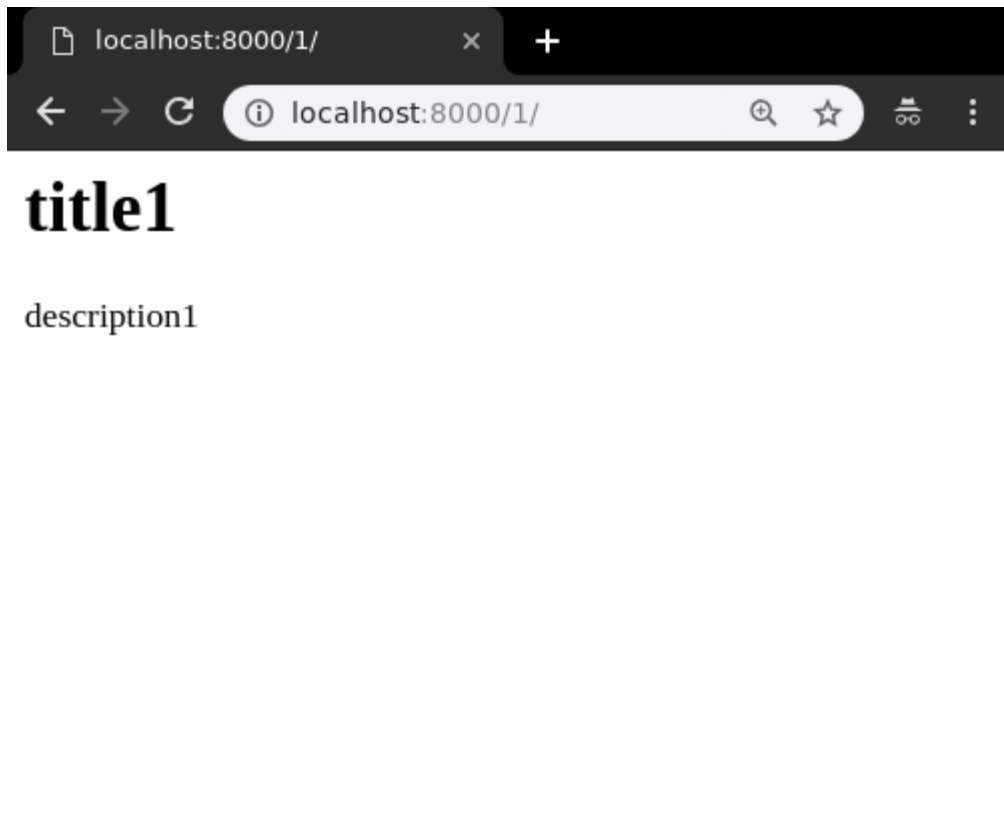
```
from django.urls import path

# importing views from views.py
from .views import GeeksDetailView
urlpatterns = [
    # <pk> is identification for id field,
    # slug can also be used
    path('<pk>/', GeeksDetailView.as_view()),
]
```

Create a template in `templates/geeks/geeksmodel_detail.html`,

```
<h1>{{ object.title }}</h1>
<p>{{ object.description }}</p>
```

Let's check what is there on <http://localhost:8000/1/>



## Manipulate Context Data in DetailView

By default DetailView will only display fields of a table. If one wants to modify this context data before sending it to template or add some extra field, context data can be overridden. In `geeks/views.py`,

```
from django.views.generic.detail import DetailView

from .models import GeeksModel

class GeeksDetailView(DetailView):
    # specify the model to use
    model = GeeksModel

    # override context data
    def get_context_data(self, *args, **kwargs):
        context = super(GeeksDetailView,
                        self).get_context_data(*args, **kwargs)
        # add extra field
        context["category"] = "MISC"
        return context
```

In `geeks/templates/geeksmodel_detail.html`,

```
<h1>{{ object.title }}</h1>  
<p>{{ object.description }}</p>  
<p>{{ category }}</p>
```

Now check, if the category is added.

