Django MySQL CRUD

**CRUD** stands for **Create**, **Retrieve**, **Update**, and **Delete**

* **Create:** In a database table, create or add new entries.
* **Retrieve:** Read, Retrieve or Fetch all or some entries from the table in a database.
* **Update:** In a database table, update or amend existing entries.
* **Delete:** In a database table, delete existing records.

C:\>python -m venv venv

C:\>venv\Scripts\activate

C:\> pip install django

(venv) C:\>django-admin startproject blog\_project

(venv) C:\>cd blog\_project

(venv) C:\blog\_project>python manage.py startapp blog\_app

(venv) C:\blog\_project>code .

Create the **templates** folder in the Django app **‘blog\_app’,** add the below files

**create.html:** To add a new blog.

**search.html:** To display the fetched data.

**update.html:** To update the existing blog.

**remove.html:** To delete a blog entity.

**In the**settings.py**file of your Django project**‘blog\_project’**include the**‘blog\_app’

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'blog\_app'

]

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'Blog',

'USER':'',

'PASSWORD':'',

'HOST':' ',

'PORT':' '

}

}

(venv) C:\blog\_project>pip install mysqlclient

Open the **urls.py** file of your Django project **‘blog\_project’**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('blog\_app.urls')),

]

create a new file called **urls.py** in your Django app **‘blog\_app’**

from django.urls import path

from . import views

urlpatterns = [

path('', views.create\_blog, name='create-blog'),

path('search/', views.retrieve\_blog, name='retrieve-blog'),

path('update/<int:pk>', views.update\_blog, name='update-blog'),

path('delete/<int:pk>', views.delete\_blog, name='delete-blog'),

]

open the **models.py** file of your Django app **‘blog\_app’** and add

from django.db import models

class Blog(models.Model):

blog\_id = models.CharField(max\_length=30)

title = models.CharField(max\_length=200)

author\_name = models.CharField(max\_length=300)

start\_date = models.DateField ()

end\_date = models.DateField()

class Meta:

db\_table = 'Blog'

In the **admin.py** file of your Django app **‘blog\_app’**

from django.contrib import admin

from .models import Blog

admin.site.register(Blog)

Create a new file called **forms.py** in your Django app **‘blog\_app’**

from django import forms

from .models import Blog

class BlogForm(forms.ModelForm):

class Meta:

model = Blog

fields = "\_\_all\_\_"

open the **views.py** file of your blog\_app

from django.shortcuts import render, redirect

from .forms import BlogForm

def create\_blog(request):

if request.method == "POST":

form = BlogForm(request.POST)

if form.is\_valid():

try:

form.save()

return redirect('search/')

except:

pass

else:

form = BlogForm()

return render(request, 'create.html', {'form':form})

In create.html file that you have created in the templates folder of your Django app **‘blog\_app’**.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Create Blog</title>

</head>

<body>

<h2 style="text-align:center"> Create New Blog </h2>

<form method="POST" >

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit" class="btn btn-success">

</form>

</body>

</html>

Add the following code in the views.py file.

from django.shortcuts import render, redirect

from .models import Blog

# Search Blog

def retrieve\_blog(request):

blogs = Blog.objects.all()

return render(request,'search.html',{'blogs':blogs} )

in the templates folder of your Django app **‘blog\_app’**,**search.html** file.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Search Blog</title>

</head>

<style>

table,

th,

td

{

border: 2px solid;

}

</style>

<body>

<h2 style="text-align:center"> Blog Information </h2>

<table align="center" style="margin: 0px auto;">

<thead>

<tr>

<th>Blog ID</th>

<th>Title</th>

<th>Author</th>

<th>Start Date</th>

<th>End Date</th>

<th>Edit Blog</th>

<th>Remove Blog</th>

</tr>

</thead>

<tbody>

{% for blog in blogs %}

<tr>

<td>{{blog.blog\_id}}</td>

<td>{{blog.title}}</td>

<td>{{blog.author\_name}}</td>

<td>{{blog.start\_date}}</td>

<td>{{blog.end\_date}}</td>

<td>

<a href="/update/{{blog.pk}}">Update</a>

</td>

<td>

<a href="/delete/{{blog.pk}}">Delete</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</body>

</html>

**UPDATE VIEW:**  Add the following code in the**views.py** file.

from django.shortcuts import render, redirect

from Blog.forms import BlogForm

from Blog.models import Blog

# Update Blog

def update\_blog(request,pk):

blogs = Blog.objects.get(id=pk)

form = BlogForm(instance=blogs)

if request.method == 'POST':

form = BlogForm(request.POST, instance=blogs)

if form.is\_valid():

form.save()

return redirect('/search')

context = {

'blogs': blogs,

'form': form,

}

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

**UPDATE HTML:** Add the following code in the**update.html** file.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Update Blog</title>

</head>

<body>

<h2 style="text-align: center; padding-top: 1em; padding-bottom: 1em;">Edit Blog</h2>

<form action="" method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit">

</form>

</body>

</html>

**DELETE VIEW:** Add the following code in the**views.py** file.

from django.shortcuts import render, redirect

from .forms import BlogForm

from .models import Blog

# Delete Blog

def delete\_blog(request, pk):

blogs = Blog.objects.get(id=pk)

if request.method == 'POST':

blogs.delete()

return redirect('/search')

context = {

'blogs': blogs,

}

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

**REMOVE HTML:**

in the**remove.html** file of the templates folder in your Django app **‘blog\_app’**.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Delete Blog</title>

</head>

<body>

<form action="" method="POST">

{% csrf\_token %}

<br><br>

<div class="alert alert-danger" role="alert">

Are you sure you want to delete "{{ blogs.emp\_name }}"?

</div>

<p>

<a href="{% url 'retrieve-blog' %}"><--Return</a>

</p>

<p>

<input type="submit" value="Confirm">

</p>

</form>

</body>

</html>

python manage.py migrate

python manage.py makemigrations

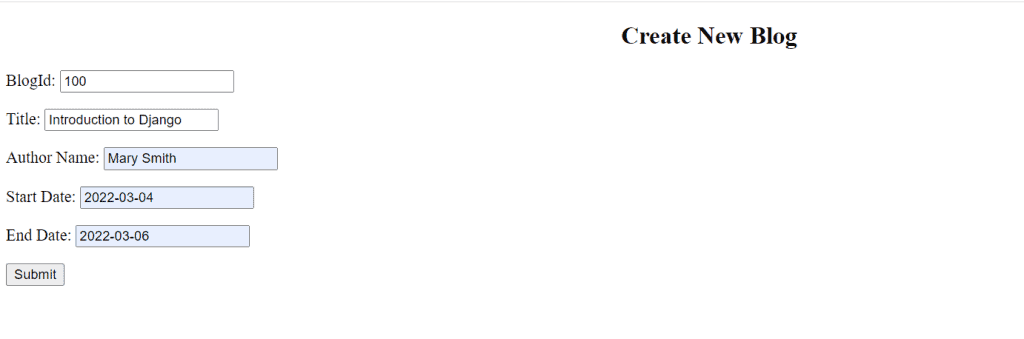
python manage.py migrate

Then, we have to run the server, for this type of the following command in the terminal.

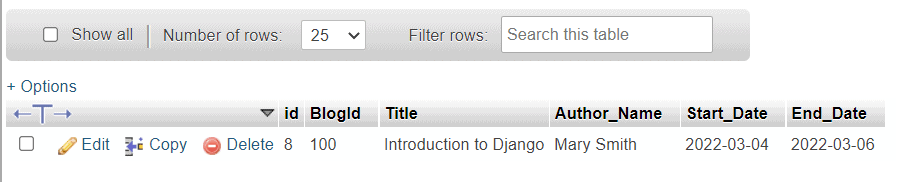
python manage.py runserver

Start the server and access the form by entering the URL as **http://127.0.0.1:8000**.

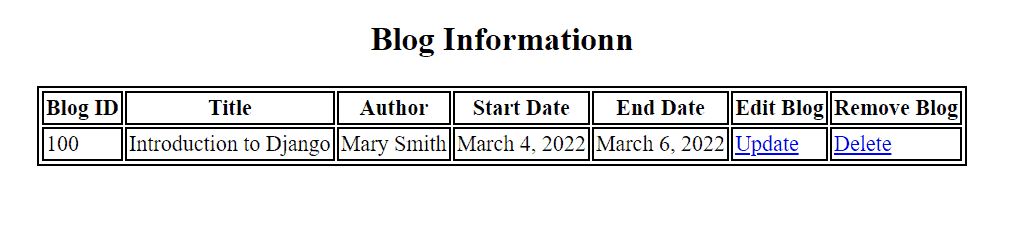
First, enter all the details in the form to create a new blog and click on **Submit** Button.



Let’s see how it looks in MySQL.



When we click on the Submit button, it renders to search HTML template.

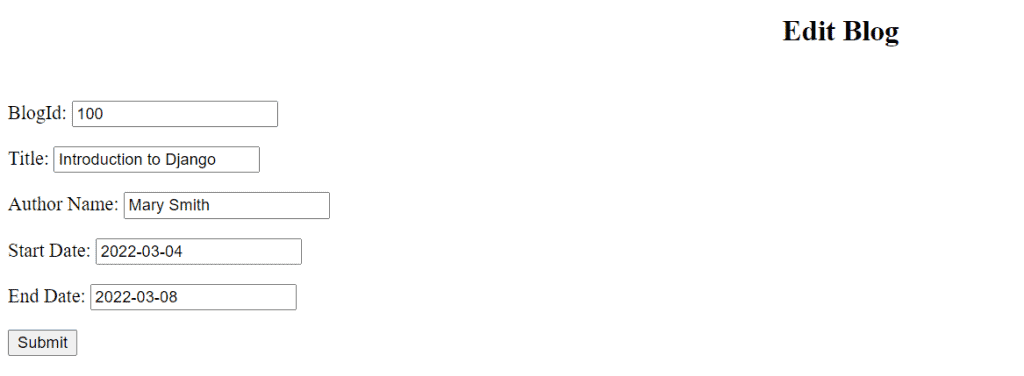


We can see that if you want to edit a blog, we can use the **Update Button**, and also for deleting a blog, you can use the **Delete Button**.

Now, see how it looks when we click on the **Update** button.

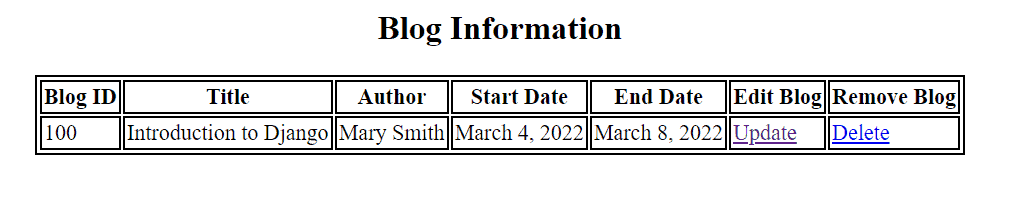
UPDATE HTML TEMPLATE

Now, we can edit any of the fields.

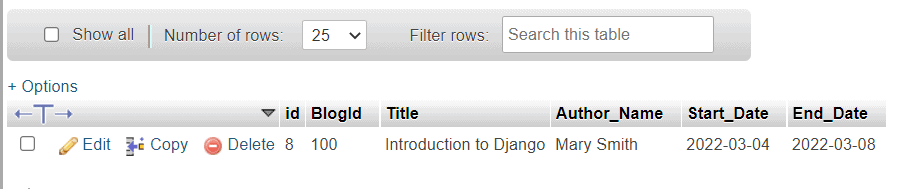


* Here I edit the End Date field of the form.
* I changed the End Date from 2022-03-06 to 2022-03-08.

Now, see what will happen when we click on the **Submit** button.

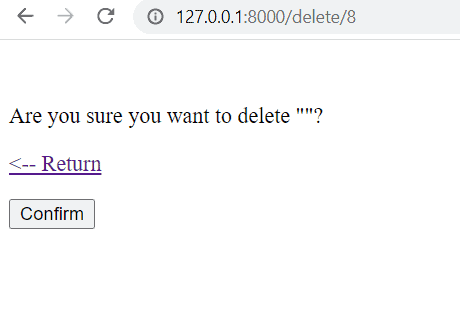


Here, we see that the End Date has been updated.



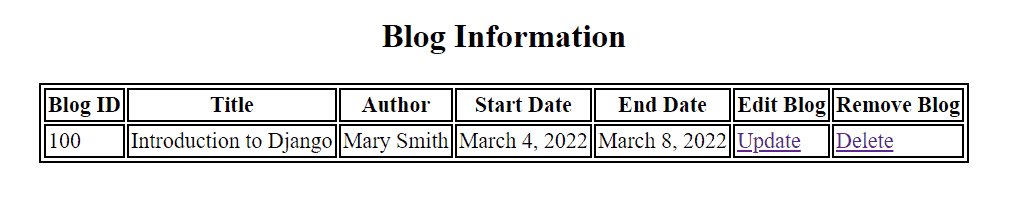
We can see it also updates blog information at MySQL.

Now, see how it looks when we click on the **Delete** button.



Now, you can see that before deletion it confirms once **“Are you sure you want** **to delete”**.

Now, see what will happen if we click on **Return**.



We can see that it takes us back to blog information.

Now, see what will happen if we click on **Confirm**.