

Here is a **step-by-step Django CRUD example** for a `Record` table with:

- `name`
- `email`
- `phone`
- `city`

We'll build:

- Model
- Forms
- Views
- URLs
- Templates
- Database setup

This is a **simple beginner-friendly CRUD (Create, Read, Update, Delete)** guide.

□ **STEP 1 — Install Django**

```
pip install django
```

Check version:

```
django-admin --version
```

□ **STEP 2 — Create Project**

```
django-admin startproject myproject  
cd myproject
```

Run server:

```
python manage.py runserver
```

Open browser:

```
http://127.0.0.1:8000/
```

□ **STEP 3 — Create App**

```
python manage.py startapp records
```

Now register the app.

Open:

```
myproject/settings.py
```

Add in `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    ...  
    'records',  
]
```

STEP 4 — Create Model (Database Table)

Open:

```
records/models.py
```

Add:

```
from django.db import models  
  
class Record(models.Model):  
    name = models.CharField(max_length=100)  
    email = models.EmailField()  
    phone = models.CharField(max_length=15)  
    city = models.CharField(max_length=100)  
  
    def __str__(self):  
        return self.name
```

STEP 5 — Run Migrations

```
python manage.py makemigrations  
python manage.py migrate
```

Now table is created in database

STEP 6 — Create Form

Create file:

```
records/forms.py
```

Add:

```
from django import forms
from .models import Record

class RecordForm(forms.ModelForm):
    class Meta:
        model = Record
        fields = ['name', 'email', 'phone', 'city']
```

□ STEP 7 — Create Views (CRUD Logic)

Open:

```
records/views.py
```

Add:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Record
from .forms import RecordForm

# CREATE + READ
def record_list(request):
    records = Record.objects.all()
    return render(request, 'records/record_list.html', {'records': records})

def record_create(request):
    if request.method == 'POST':
        form = RecordForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('record_list')
    else:
        form = RecordForm()
    return render(request, 'records/record_form.html', {'form': form})

# UPDATE
def record_update(request, pk):
    record = get_object_or_404(Record, pk=pk)
    if request.method == 'POST':
        form = RecordForm(request.POST, instance=record)
        if form.is_valid():
            form.save()
            return redirect('record_list')
```

```
else:
    form = RecordForm(instance=record)
    return render(request, 'records/record_form.html', {'form': form})

# DELETE
def record_delete(request, pk):
    record = get_object_or_404(Record, pk=pk)
    if request.method == 'POST':
        record.delete()
        return redirect('record_list')
    return render(request, 'records/record_confirm_delete.html', {'record':
record})
```

□ STEP 8 — Create URLs

Create:

`records/urls.py`

Add:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.record_list, name='record_list'),
    path('create/', views.record_create, name='record_create'),
    path('update/<int:pk>/', views.record_update, name='record_update'),
    path('delete/<int:pk>/', views.record_delete, name='record_delete'),
]
```

Now connect app URLs to main project.

Open:

`myproject/urls.py`

Modify:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('records.urls')),
]
```

□ STEP 9 — Create Templates

Create folder structure:

```
records/  
  templates/  
    records/
```

1 □ record_list.html

```
records/templates/records/record_list.html  
<!DOCTYPE html>  
<html>  
<head>  
  <title>Record List</title>  
</head>  
<body>  
  <h2>Record List</h2>  
  <a href="{% url 'record_create' %}">Add New Record</a>  
  <br><br>  
  
  <table border="1">  
    <tr>  
      <th>Name</th>  
      <th>Email</th>  
      <th>Phone</th>  
      <th>City</th>  
      <th>Actions</th>  
    </tr>  
  
    {% for record in records %}  
    <tr>  
      <td>{{ record.name }}</td>  
      <td>{{ record.email }}</td>  
      <td>{{ record.phone }}</td>  
      <td>{{ record.city }}</td>  
      <td>  
        <a href="{% url 'record_update' record.id %}">Edit</a>  
        <a href="{% url 'record_delete' record.id %}">Delete</a>  
      </td>  
    </tr>  
    {% endfor %}  
  </table>  
</body>  
</html>
```

2 record_form.html

```
records/templates/records/record_form.html
<!DOCTYPE html>
<html>
<head>
  <title>Record Form</title>
</head>
<body>
  <h2>Record Form</h2>

  <form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save</button>
  </form>

  <br>
  <a href="{% url 'record_list' %}">Back to List</a>
</body>
</html>
```

3 record_confirm_delete.html

```
records/templates/records/record_confirm_delete.html
<!DOCTYPE html>
<html>
<head>
  <title>Delete Record</title>
</head>
<body>
  <h2>Are you sure you want to delete "{{ record.name }}"?</h2>

  <form method="POST">
    {% csrf_token %}
    <button type="submit">Yes, Delete</button>
  </form>

  <br>
  <a href="{% url 'record_list' %}">Cancel</a>
</body>
</html>
```

□ STEP 10 — Run Server

```
python manage.py runserver
```

Visit:

```
http://127.0.0.1:8000/
```

□ You now have full CRUD working!

□ What You Built

| Operation | URL |
|-----------|------------|
| List | / |
| Create | /create/ |
| Update | /update/1/ |
| Delete | /delete/1/ |

□ Bonus: Add Admin Panel

Open `records/admin.py`:

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

admin.site.register(Record)
```

Create superuser:

```
python manage.py createsuperuser
```

Login at:

```
http://127.0.0.1:8000/admin/
```

□ Summary

You learned:

- Create project
- Create app
- Create model
- Migrate database
- Create form
- Create CRUD views
- Configure URLs
- Create templates
- Run project

Record List

[Add New Record](#)

| Name | Email | Phone | City | Actions |
|--------------|-----------------|------------|---------|---|
| om maurya | omair@gmail.com | 8149996597 | hwar | Edit Delete |
| KIHAN NARGIS | hanip@gmail.com | 8149996598 | andheri | Edit Delete |

Activate Windows
Go to Settings to activate Windows.

Type here to search

25°C Clear 6:56 PM 2/18/2026