

Here's a **clear step-by-step guide** to build a CRUD app with a **Django backend** and a **React frontend** for records with fields: `id`, `name`, `city`, `phone`.

□ PART 1: Django Backend (API)

1 □ Install & Setup

```
pip install django djangorestframework django-cors-headers
```

```
(env) D:\django-warmup>pip install django djangorestframework django-cors-headers
WARNING: Ignoring invalid distribution -ip (d:\env\lib\site-packages)
Requirement already satisfied: django in d:\env\lib\site-packages (5.2.12)
Requirement already satisfied: djangorestframework in d:\env\lib\site-packages (2.17.1)
```

```
django-admin startproject backend
```

```
cd backend
```

```
python manage.py startapp api
```

```
(env) D:\django-warmup>django-admin startproject backend
(env) D:\django-warmup>cd backend
(env) D:\django-warmup\backend>python manage.py startapp api
```

2 □ Update settings.py

Add apps:

```
INSTALLED_APPS = [
    ...
    'rest_framework',
    'corsheaders',
    'api',
]
```

Add middleware:

```
MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    ...
]
```

Allow frontend:

```
CORS_ALLOW_ALL_ORIGINS = True
```

3 Create Model (api/models.py)

```
from django.db import models

class Record(models.Model):
    name = models.CharField(max_length=100)
    city = models.CharField(max_length=100)
    phone = models.CharField(max_length=15)

    def __str__(self):
        return self.name
```

4 Migrate DB

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

```
(env) D:\django-warmup\backend>python manage.py makemigrations
Migrations for 'api':
  api\migrations\0001_initial.py
  + Create model Record

(env) D:\django-warmup\backend>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, api, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
```

5 Create Serializer (api/serializers.py)

```
from rest_framework import serializers
from .models import Record

class RecordSerializer(serializers.ModelSerializer):
    class Meta:
        model = Record
        fields = '__all__'
```

6 Create Views (api/views.py)

```
from rest_framework import viewsets
from .models import Record
from .serializers import RecordSerializer

class RecordViewSet(viewsets.ModelViewSet):
    queryset = Record.objects.all()
    serializer_class = RecordSerializer
```

7 Setup URLs

api/urls.py

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import RecordViewSet

router = DefaultRouter()
router.register(r'records', RecordViewSet)

urlpatterns = [
    path('', include(router.urls)),
]
```

backend/urls.py

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

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

8 Run Server

```
python manage.py runserver
```

□ API endpoints:

- GET → /api/records/
- POST → /api/records/
- PUT → /api/records/{id}/
- DELETE → /api/records/{id}/

step-by-step guide to build your React frontend using **Vite + React** for your Django CRUD API.

□ PART 1: Setup React with Vite

1 □ Create Project

```
npm create vite@latest frontend
cd frontend
npm install
```

Choose:

- Framework → **React**
- Variant → **JavaScript**

```
D:\django-warmup>npm create vite@latest frontend
Need to install the following packages:
create-vite@9.0.4
Ok to proceed? (y) y

> npx
> create-vite frontend

|
o Select a framework:
| React
|
o Select a variant:
| JavaScript
|
o Install with npm and start now?
| Yes
|
o Scaffolding project in D:\django-warmup\frontend...
|
o Installing dependencies with npm...

added 151 packages, and audited 152 packages in 31s
```

2 Install Axios

```
cd frontend  
npm install axios
```

```
D:\django-warmup\frontend>npm install axios  
  
added 23 packages, and audited 175 packages in 9s  
  
42 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
  
D:\django-warmup\frontend>npm run dev  
  
> frontend@0.0.0 dev
```

3 Start Development Server

```
npm run dev
```

Runs at:

```
http://localhost:5173/
```

PART 2: Clean Project

Inside `src/`, remove unnecessary files and keep:

```
src/  
├── App.jsx  
├── main.jsx  
└── index.css
```

□ PART 3: CRUD Frontend Code

Replace `App.jsx` with:

```
import { useEffect, useState } from "react";
import axios from "axios";

function App() {
  const [records, setRecords] = useState([]);
  const [form, setForm] = useState({ name: "", city: "", phone: "" });
  const [editId, setEditId] = useState(null);

  const API = "http://127.0.0.1:8000/api/records/";

  useEffect(() => {
    fetchRecords();
  }, []);

  const fetchRecords = async () => {
    const res = await axios.get(API);
    setRecords(res.data);
  };

  const handleChange = (e) => {
    setForm({ ...form, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();

    if (editId) {
      await axios.put(API + editId + "/", form);
      setEditId(null);
    } else {
      await axios.post(API, form);
    }

    setForm({ name: "", city: "", phone: "" });
    fetchRecords();
  };

  const handleEdit = (rec) => {
    setForm({
      name: rec.name,
      city: rec.city,
      phone: rec.phone,
    });
    setEditId(rec.id);
  };

  const handleDelete = async (id) => {
    await axios.delete(API + id + "/");
    fetchRecords();
  };
}
```

```

return (
  <div style={{ padding: "20px", maxWidth: "500px", margin: "auto" }}>
    <h2>CRUD App (Vite + React)</h2>

    <form onSubmit={handleSubmit}>
      <input
        name="name"
        value={form.name}
        onChange={handleChange}
        placeholder="Name"
      /><br /><br />

      <input
        name="city"
        value={form.city}
        onChange={handleChange}
        placeholder="City"
      /><br /><br />

      <input
        name="phone"
        value={form.phone}
        onChange={handleChange}
        placeholder="Phone"
      /><br /><br />

      <button type="submit">
        {editId ? "Update" : "Add"}
      </button>
    </form>

    <hr />

    <ul>
      {records.map((rec) => (
        <li key={rec.id}>
          {rec.name} | {rec.city} | {rec.phone}
          <br />
          <button onClick={() => handleEdit(rec)}>Edit</button>
          <button onClick={() => handleDelete(rec.id)}>Delete</button>
          <hr />
        </li>
      ))}
    </ul>
  </div>
);
}

export default App;

```

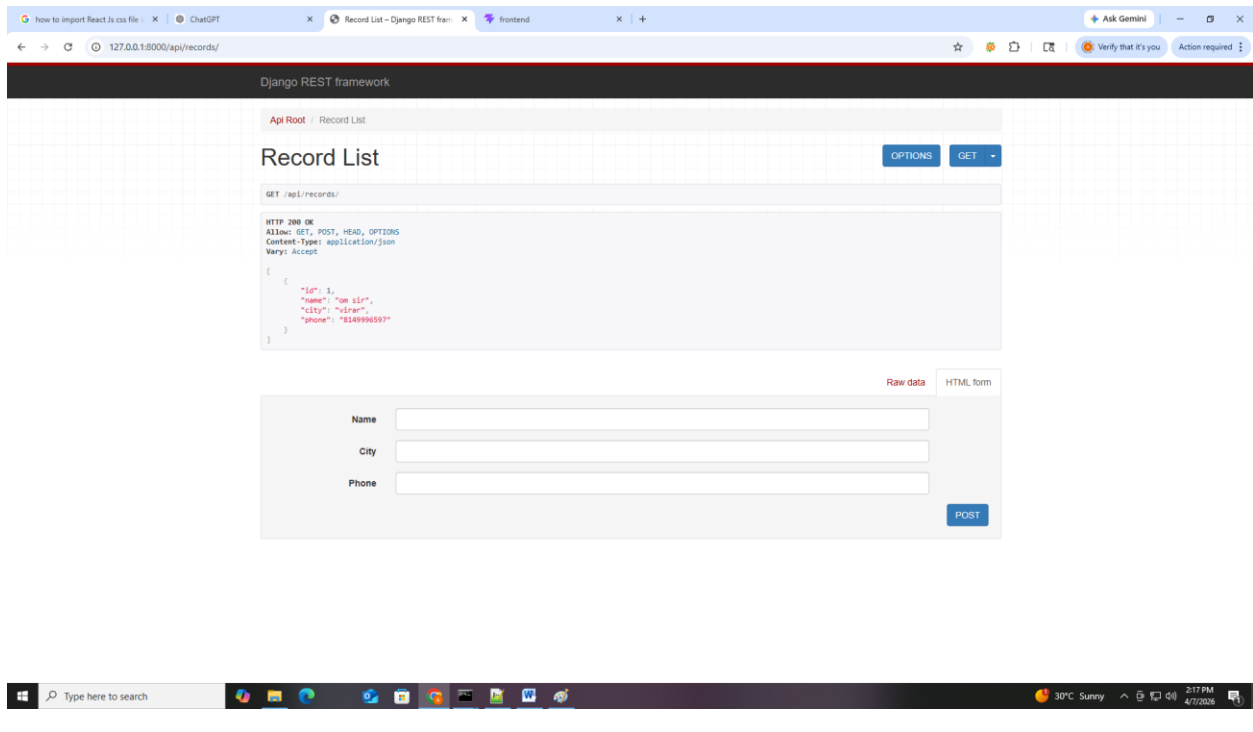
□ PART 4: Connect with Backend

Make sure your **Django backend is running**:

```
python manage.py runserver
```

API should be:

<http://127.0.0.1:8000/api/records/>



The screenshot shows a web browser window with the URL `127.0.0.1:8000/api/records/`. The page displays the Django REST framework interface for the 'Record List' endpoint. The 'GET /api/records/' method is selected, and the response is shown as JSON:

```
{
  "id": 1,
  "name": "on s1r",
  "city": "s1r",
  "phone": "s1s999599?"
}
```

Below the response, there are input fields for 'Name', 'City', and 'Phone', and a 'POST' button. The interface also includes 'Raw data' and 'HTML form' tabs.

□ PART 5: Fix CORS (Important)

In Django `settings.py`:

```
CORS_ALLOW_ALL_ORIGINS = True
```

Or

```
CORS_ALLOWED_ORIGINS = [
    "http://localhost:5173",
]
```

□ FINAL RESULT

You now have:

- Vite React frontend → `http://localhost:5173`
- Django backend → `http://127.0.0.1:8000`
- Full CRUD working:
 - Add record
 - View records
 - Edit record
 - Delete record

Output:-

