

Let's build a full CRUD system using Django (backend) and React + Axios (frontend).

□ What You'll Get

- **Backend:** Django REST API with full CRUD for Student
 - **Frontend:** React (Vite) + Axios calling the Django API
-

□ BACKEND – Django REST Framework Setup

1. □ Install Django + Django REST Framework

```
django-admin startproject backend
cd backend
pip install django djangorestframework corsheaders
python manage.py startapp students
```

2. □ Add to backend/settings.py

```
INSTALLED_APPS = [
    ...,
    'rest_framework',
    'your_app_name', # replace with actual app name
    'corsheaders',
]

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

CORS_ALLOW_ALL_ORIGINS = True # For development only
```

3. □ Your models.py

Backend Models in django

students/models.py :-

```
from django.db import models

# Create your models here.

class student (models.Model):

    name=models.CharField(max_length=100)

    age=models.IntegerField()

    grade=models.CharField(max_length=10)

    def __str__(self):

        return self.name
```

students/serializers.py file code:-

```
from rest_framework import serializers

from .models import student

class StudentSerializer(serializers.ModelSerializer):

    class Meta:

        model=student

        fields = '__all__'
```

students/views.py file code :-

```
from rest_framework import viewsets
from .models import student
from .serializers import StudentSerializer
```

```
class StudentViewSet(viewsets.ModelViewSet):
    queryset=student.objects.all()
    serializer_class=StudentSerializer
```

students/urls.py file code(inside app folder):-

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

```
router=DefaultRouter()
router.register(r'students',StudentViewSet)
```

```
urlpatterns=[
```

```
    path("",include(router.urls)),  
]
```

backend/urls.py file(project folder):-

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

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

FRONTEND – Vite + React + Axios CRUD

Set Up Vite + React Project

If you don't already have one:

```
npm create vite@latest my-crud-app --template react
cd my-crud-app
npm install
```

1. Install Axios

```
npm install axios
```

2. React Component: `src/components/Students.jsx`

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

const API_URL = 'http://127.0.0.1:8000/api/students/';

function Students() {
  const [students, setStudents] = useState([]);
  const [newStudent, setNewStudent] = useState({ name: '', age: '', grade: '' });
  const [editingStudent, setEditingStudent] = useState(null);

  // Fetch all students
  const fetchStudents = async () => {
    try {
      const res = await axios.get(API_URL);
      setStudents(res.data);
    } catch (err) {
      console.error('Fetch error:', err);
    }
  };

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

  // Add new student
  const handleAdd = async (e) => {
```

```

    e.preventDefault();
    try {
      const res = await axios.post(API_URL, newStudent);
      setStudents([...students, res.data]);
      setNewStudent({ name: '', age: '', grade: '' });
    } catch (err) {
      console.error('Add error:', err);
    }
  };

  // Update student
  const handleUpdate = async (e) => {
    e.preventDefault();
    try {
      const res = await axios.put(`${API_URL}${editingStudent.id}/`,
editingStudent);
      setStudents(students.map(s => (s.id === editingStudent.id ? res.data :
s)));
      setEditingStudent(null);
    } catch (err) {
      console.error('Update error:', err);
    }
  };

  // Delete student
  const handleDelete = async (id) => {
    try {
      await axios.delete(`${API_URL}${id}/`);
      setStudents(students.filter(s => s.id !== id));
    } catch (err) {
      console.error('Delete error:', err);
    }
  };

  return (
    <div style={{ padding: '2rem' }}>
      <h2>Django + React CRUD (Students)</h2>

      { /* Add New */ }
      <form onSubmit={handleAdd} style={{ marginBottom: '1rem' }}>
        <input
          type="text"
          placeholder="Name"
          value={newStudent.name}
          onChange={e => setNewStudent({ ...newStudent, name: e.target.value
}})
          required
        />
        <input
          type="number"
          placeholder="Age"
          value={newStudent.age}
          onChange={e => setNewStudent({ ...newStudent, age: e.target.value
}})
          required
        />
        <input

```

```

        type="text"
        placeholder="Grade"
        value={newStudent.grade}
        onChange={e => setNewStudent({ ...newStudent, grade: e.target.value
    ))}
        required
    />
    <button type="submit">Add</button>
</form>

{/* Edit Form */}
(editingStudent && (
    <form onSubmit={handleUpdate} style={{ marginBottom: '1rem' }}>
        <input
            type="text"
            value={editingStudent.name}
            onChange={e => setEditingStudent({ ...editingStudent, name:
e.target.value })}
            required
        />
        <input
            type="number"
            value={editingStudent.age}
            onChange={e => setEditingStudent({ ...editingStudent, age:
e.target.value })}
            required
        />
        <input
            type="text"
            value={editingStudent.grade}
            onChange={e => setEditingStudent({ ...editingStudent, grade:
e.target.value })}
            required
        />
        <button type="submit">Update</button>
        <button onClick={() => setEditingStudent(null)}
type="button">Cancel</button>
    </form>
    ))
)

{/* List Students */}
<ul>
    {students.map(student => (
        <li key={student.id}>
            <strong>{student.name}</strong> | Age: {student.age} | Grade:
{student.grade}
            <button onClick={() => setEditingStudent(student)} style={{
marginLeft: '1rem' }}>Edit</button>
            <button onClick={() => handleDelete(student.id)} style={{
marginLeft: '0.5rem', color: 'red' }}>Delete</button>
        </li>
    ))}
</ul>
</div>
);
}

```

```
export default Students;
```

3. Add to App

```
// src/App.jsx
import React from 'react';
import Students from '../components/Students';

function App() {
  return (
    <div>
      <Students />
    </div>
  );
}

export default App;
```

Run Everything

1. Start Django API

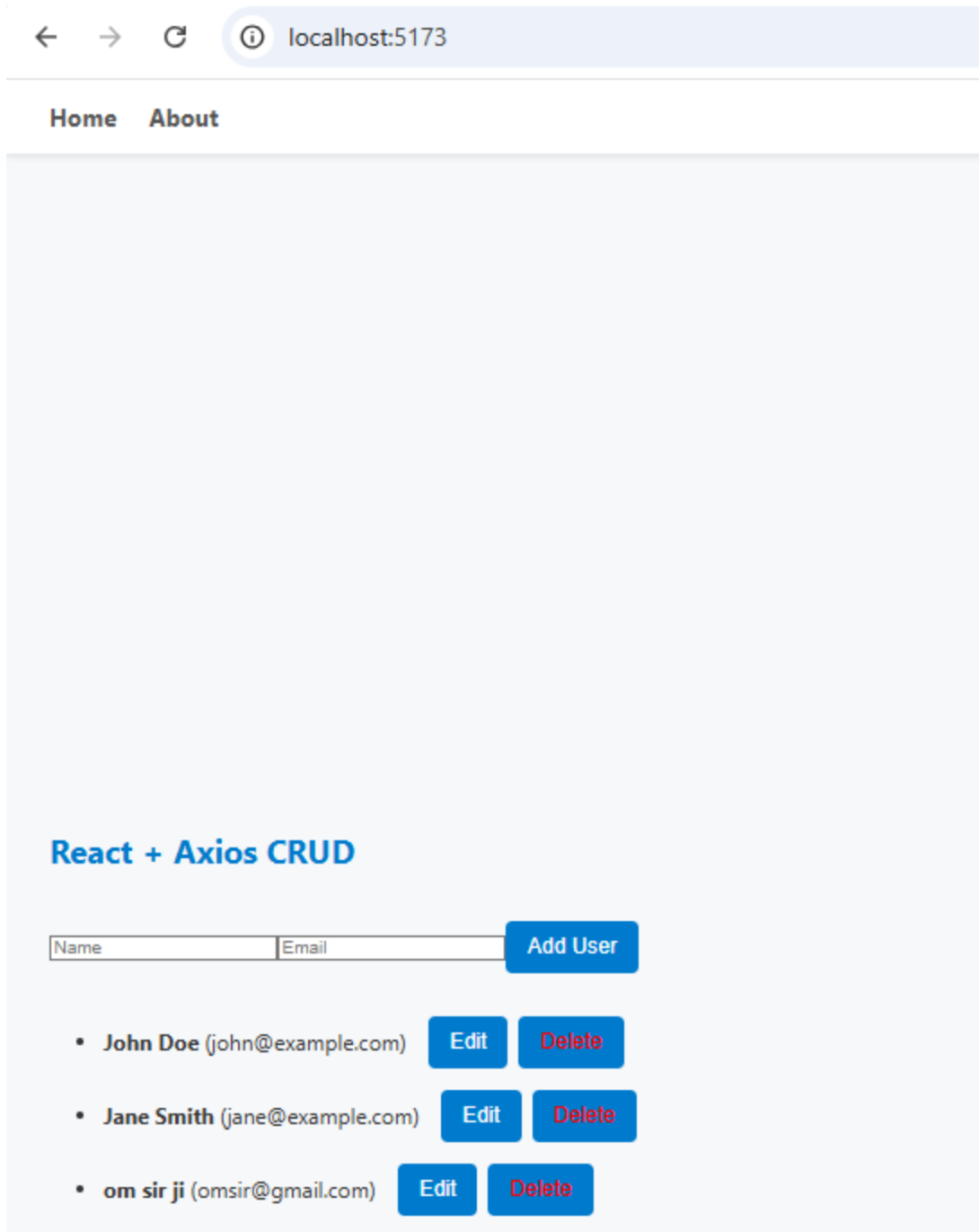
```
python manage.py runserver
```

2. Start Vite App

```
npm run dev
```

3. Open:

<http://localhost:5173>



☐ Test CRUD

| Action | URL | Method |
|--------|----------------|--------|
| List | /api/students/ | GET |

| Action | URL | Method |
|---------------|---------------------|---------------|
| Add | /api/students/ | POST |
| Update | /api/students/<id>/ | PUT |
| Delete | /api/students/<id>/ | DELETE |