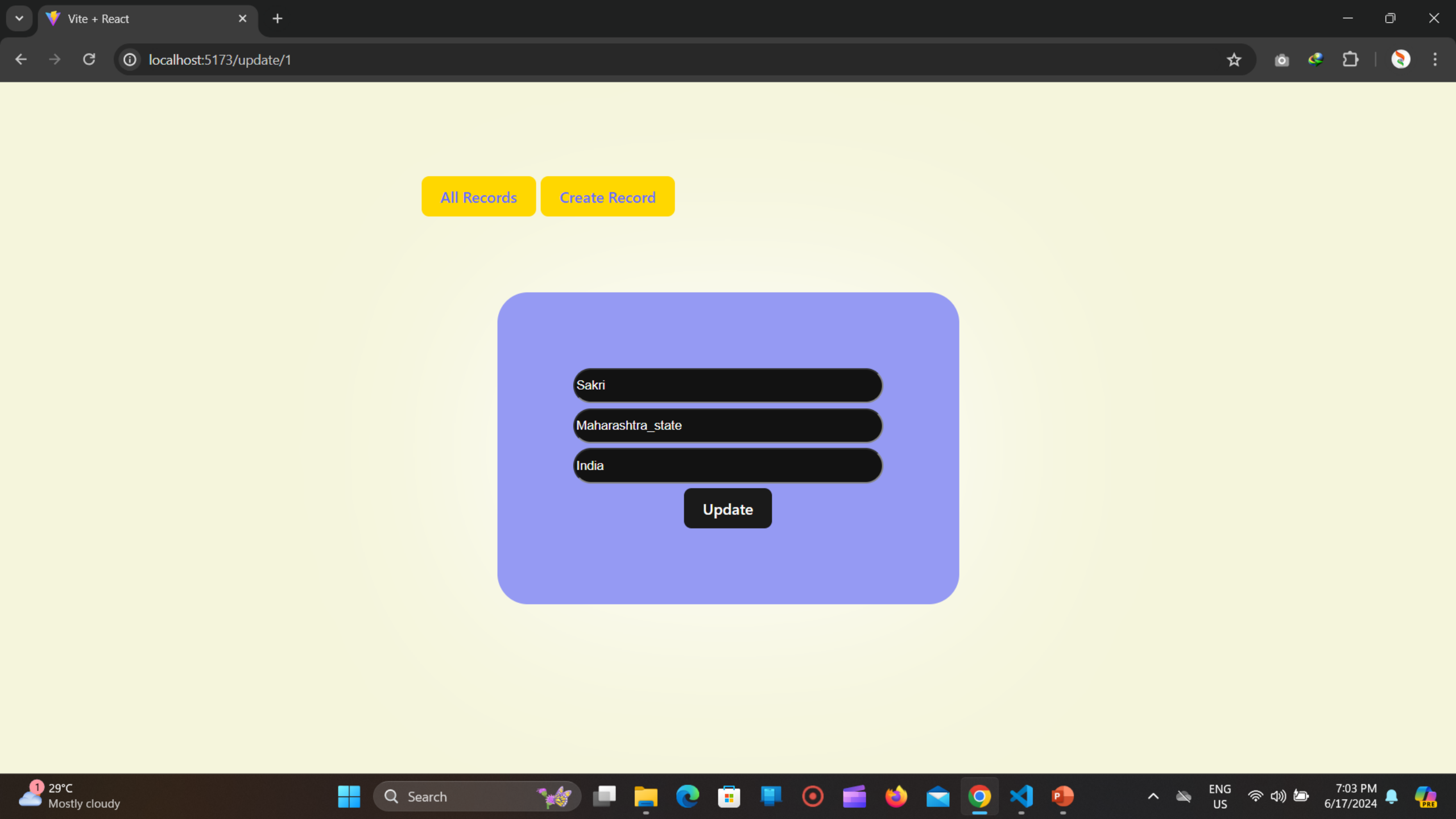


All Records Create Record

Django API With React JS

ID	City	State	Country	Actions  	
1	Sakri	Maharashtra_state	India	Update 	Delete 
2	Dhule	Maharashtra	India	Update 	Delete 
7	New York	US	America	Update 	Delete 
9	Amalner_now	Maharashtra	India	Update 	Delete 
20	Nagpur	Maharashtra	India	Update 	Delete 
21	Ahmadnagar	Maharashtra	India	Update 	Delete 
22	Sangamner	Maharashtra	India	Update 	Delete 



All Records

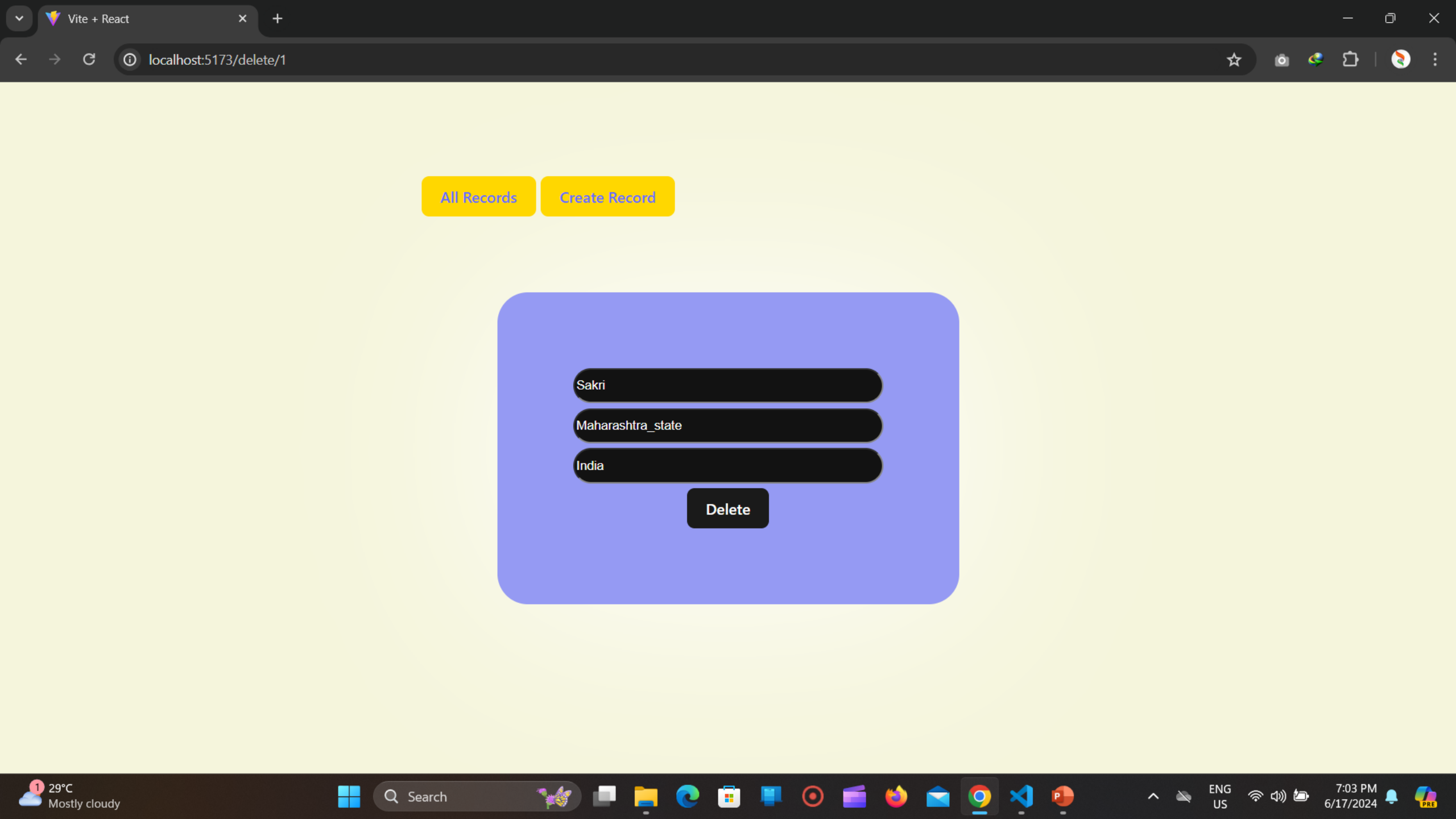
Create Record

Sakri

Maharashtra_state

India

Update



All Records

Create Record

Sakri

Maharashtra_state

India

Delete

Backend “Django REST API” Configuration

```
settings.py |
-----
INSTALLED_APPS = [
    'app',
    'rest_framework',
    'corsheaders',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = '/static/'
CORS_ORIGIN_ALLOW_ALL=True
# Default primary key field type
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

EXPLORER

...

App.jsxviews.pyForm.jsxApp.css

▼ SUNDAY

> my-react-app

> node_modules

▼ project

▼ app

> __pycache__

> migrations

🔗 __init__.py

🔗 admin.py

🔗 apps.py

🔗 models.py

🔗 serializers.py

🔗 tests.py

🔗 urls.py

🔗 views.py

▼ project

> __pycache__

🔗 __init__.py

🔗 asgi.py

🔗 settings.py

🔗 urls.py

🔗 wsgi.py

≡ db.sqlite3

🔗 manage.py

{ } package-lock.json

{ } package.json

project > project > 🔗 urls.py > ...

1

2 from django.contrib import admin

3 from django.urls import path,include

4

5 urlpatterns = [

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

7 | path("",include("app.urls"))

8]

9

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

VITE v5.2.13 ready in 258 ms

→ Local: http://localhost:5173/

→ Network: use --host to expose

→ press h + enter to show help

EXPLORER

...

+

+

↺

📄

▼ SUNDAY

> my-react-app

> node_modules

▼ project

▼ app

> __pycache__

> migrations

🔌 __init__.py

🔌 admin.py

🔌 apps.py

🔌 models.py

🔌 serializers.py

🔌 tests.py

🔌 urls.py

🔌 views.py

▼ project

> __pycache__

🔌 __init__.py

🔌 asgi.py

🔌 settings.py

🔌 urls.py

🔌 wsgi.py

≡ db.sqlite3

🔌 manage.py

{ } package-lock.json

{ } package.json

App.jsx

views.py

models.py X

Form.jsx

project > app > 🔌 models.py > 🗑 cityModel > 📦 __str__

1

2

3

4

5

6

7

8

9

10

11

12

13

from django.db import models

Create your models here.

class cityModel(models.Model):

id = models.AutoField(primary_key=True)

city = models.CharField(max_length=500)

state = models.CharField(max_length=500)

country = models.CharField(max_length=500)

def __str__(self) -> str:

return super().__str__()

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

EXPLORER

...

✓ SUNDAY

> my-react-app

> node_modules

✓ project

✓ app

> __pycache__

> migrations

🔌 __init__.py

🔌 admin.py

🔌 apps.py

🔌 models.py

🔌 serializers.py

🔌 tests.py

🔌 urls.py

🔌 views.py

✓ project

> __pycache__

🔌 __init__.py

🔌 asgi.py

🔌 settings.py

🔌 urls.py

🔌 wsgi.py

≡ db.sqlite3

🔌 manage.py

{} package-lock.json

{} package.json

App.jsx

views.py

serializers.py ✕

Form.jsx

project > app > 🔌 serializers.py > 🗑 citySerializer > 🗑 Meta

1

from rest_framework import serializers

2

from .models import cityModel

3

4

5

class citySerializer(serializers.ModelSerializer):

6

class Meta:

7

model = cityModel

8

fields = "__all__"

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

EXPLORER

SUNDAY

> my-react-app

> node_modules

▼ project

▼ app

> __pycache__

> migrations

🔗 __init__.py

🔗 admin.py

🔗 apps.py

🔗 models.py

🔗 serializers.py

🔗 tests.py

🔗 urls.py

🔗 views.py

▼ project

> __pycache__

🔗 __init__.py

🔗 asgi.py

🔗 settings.py

🔗 urls.py

🔗 wsgi.py

≡ db.sqlite3

🔗 manage.py

{ } package-lock.json

{ } package.json

App.jsx

views.py

urls.py

Form.jsx

project > app > urls.py > ...

1 from django.urls import path

2 from . import views

3

4 urlpatterns = [

5 path("", views.create),

6 path("home/", views.list),

7 path("home/<int:id>/", views.list_detail),

8 path("delete/<int:id>/", views.delete_detail),

9 path("update/<int:id>/", views.update),

10 path("list_react/", views.list_react)

11]

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

SUNDAY

> my-react-app

> node_modules

▼ project

▼ app

> __pycache__

> migrations

🔗 __init__.py

🔗 admin.py

🔗 apps.py

🔗 models.py

🔗 serializers.py

🔗 tests.py

🔗 urls.py

🔗 views.py

▼ project

> __pycache__

🔗 __init__.py

🔗 asgi.py

🔗 settings.py

🔗 urls.py

🔗 wsgi.py

🔗 db.sqlite3

🔗 manage.py

() package-lock.json

() package.json

project > app > views.py > ...

```
1 from django.shortcuts import get_object_or_404, redirect, render
2 from .models import cityModel
3 from .serializers import citySerializer
4 from rest_framework.response import Response
5 from rest_framework.decorators import api_view
6 # Create your views here.
7
8 # new record
9 @api_view(['POST'])
10 def create(request):
11     if request.method == "POST" or None:
12         serializer = citySerializer(data=request.data)
13         if serializer.is_valid():
14             serializer.save()
15             return Response(serializer.data)
16         return Response(serializer.errors)
17
18 @api_view(['GET'])
19 def list(request):
20     instance = cityModel.objects.all()
21     serializer = citySerializer(instance=instance, many=True)
22     return Response(serializer.data)
23
24 @api_view(['GET'])
25 def list_react(request):
26     instance = cityModel.objects.all()
27     serializer = citySerializer(instance=instance, many=True)
28     output = []
29     for i in serializer.data:
30         output.append(i)
31     return Response({"message": output})
32
33 @api_view(['GET'])
34 def list_detail(request, id):
35     instance = get_object_or_404(cityModel, id=id)
36     serializer = citySerializer(instance=instance)
37     if serializer is not None:
38         return Response(serializer.data)
39
40 @api_view(['DELETE'])
41 def delete_detail(request, id):
42     instance = get_object_or_404(cityModel, id=id)
43     if instance is not None:
44         instance.delete()
45     return Response({"data": "Record Deleted Successfully."})
46
47 @api_view(['PUT', 'GET'])
48 def update(request, id):
49     instance = get_object_or_404(cityModel, id=id)
50     serializer = citySerializer(instance=instance, data=request.data)
51     if serializer.is_valid():
52         serializer.save()
53         return Response(serializer.data)
54     else:
55         serializer = citySerializer(instance=instance)
56         return Response(serializer.data)
```

Frontend “React JS” Configuration

EXPLORER

...

my-react-app

node_modules

public

src

assets

App.css

App.jsx

DeleteForm.jsx

Form.jsx

GetDjango.jsx

index.css

LatestForm.jsx

main.jsx

UpdateForm.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

node_modules

project

package-lock.json

package.json

App.jsx

views.py

urls.py

Form.jsx

App.css

GetDjango.jsx

my-react-app > src > App.jsx > App

```
1  import React from 'react';
2  import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
3  import GetDjango from './GetDjango';
4  import UpdateForm from './UpdateForm';
5  import LatestForm from './LatestForm';
6  import DeleteForm from './DeleteForm';
7  import './App.css';
8
9  function App() {
10     return (
11         <>
12         <Router>
13             <nav>
14                 <button><Link to="/">All Records</Link></button>
15                 <button><Link to="/create">Create Record</Link></button>
16             </nav>
17             <Routes>
18                 <Route path="/" element={<GetDjango/>}/>
19                 <Route path="/update/:id" element={<UpdateForm/>}/>
20                 <Route path="/delete/:id" element={<DeleteForm/>}/>
21                 <Route path="/create" element={<LatestForm/>}/>
22             </Routes>
23         </Router>
24     );
25
26 }
27
28 export default App;
29
```

EXPLORER

...

App.jsxviews.pyurls.pyForm.jsxApp.cssGetDjango.jsx

SUNDAY

my-react-app

node_modules

public

src

assets

App.cssApp.jsxDeleteForm.jsxForm.jsxGetDjango.jsx



index.cssLatestForm.jsxmain.jsxUpdateForm.jsx.eslintrc.cjs.gitignoreindex.htmlpackage-lock.jsonpackage.jsonREADME.mdvite.config.js

node_modules

project

package-lock.jsonpackage.json

my-react-app > src > GetDjango.jsx > ...

```
1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import UpdateForm from "../UpdateForm";
4 import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
5
6 function GetDjango(){
7   const [data, setData] = useState([]);
8
9   // get all recrods using axios.get method from django endpoint
10  useEffect(()=>{
11    axios.get(`http://127.0.0.1:8000/home`)
12      .then(response=>{
13        setData(response.data);
14      })
15      .catch(error=>{
16        console.error(error);
17      });
18  },[]);
19
20  // return axios response in table format using below
21  return (
22    <>
23    <center><h1>Django API With React JS</h1></center>
24    <center><table border="1">
25      <thead>
26        <tr>
27          <th>ID</th>
28          <th>City</th>
29          <th>State</th>
30          <th>Country</th>
31          <th colSpan={2}>Actions  
```

EXPLORER

...

App.jsxviews.pyurls.pyForm.jsx# App.cssGetDjango.jsxUpdateForm.jsx

▼ SUNDAY

▼ my-react-app

> node_modules

> public

▼ src

> assets

App.css

App.jsx

DeleteForm.jsx

Form.jsx

GetDjango.jsx

index.css

LatestForm.jsx

main.jsx

UpdateForm.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

> node_modules

> project

package-lock.json

package.json

my-react-app > src > LatestForm.jsx > ...

```
1  import React, { useState } from "react";
2  import axios from 'axios';
3  import { useNavigate } from "react-router-dom";
4
5  function LatestForm(){
6      const [city,setCity] = useState("");
7      const [state,setState] = useState("");
8      const [country, setCountry] = useState("");
9      const navigate = useNavigate();
10
11      // function to perform new record creation
12      const handleSubmit=(event)=>{
13          event.preventDefault();
14          axios.post(`http://127.0.0.1:8000/`,{city,state,country})
15              .then(response=>{
16                  alert("data has been registered on database.")
17                  navigate("/");
18              })
19              .catch(error=>{
20                  console.error(error);
21              });
22      };
23
24      // form to create new record using post method
25      return (
26          <>
27          <form method="POST" onSubmit={handleSubmit}>
28              <input type="text" placeholder='city' value={city} onChange={e=>setCity(e.target.value)}/><br/>
29              <input type="text" placeholder='state' value={state} onChange={e=>setState(e.target.value)} /><br/>
30              <input type="text" placeholder='country' value={country} onChange={e=>setCountry(e.target.value)} />
31              <button type="submit">Submit</button>
32          </form>
33          </>
34      )
35  };
36  export default LatestForm;
37
```

EXPLORER

...

App.jsxviews.pyurl.pyForm.jsxApp.cssGetDjango.jsxUpdateForm.jsxDeleteForm.jsx

SUNDAY

my-react-app

node_modulespublicsrc

assetsApp.cssApp.jsxDeleteForm.jsxForm.jsxGetDjango.jsxindex.cssLatestForm.jsxmain.jsxUpdateForm.jsx

.eslintrc.cjs.gitignoreindex.htmlpackage-lock.jsonpackage.jsonREADME.mdvite.config.jsnode_modulesprojectpackage-lock.jsonpackage.json

OUTLINE

TIMELINE

my-react-app > src > UpdateForm.jsx

```
1 import React, { useEffect } from "react";
2 import axios from "axios";
3 import { useState } from 'react';
4 import { useNavigate, useParams } from "react-router-dom";
5 function UpdateForm(){
6   const {id} = useParams();
7
8   const [city,setCity] = useState('');
9   const [state,setState] = useState('');
10  const [country,setCountry] = useState('');
11  const navigate = useNavigate();
12  useEffect(()=>{
13    axios.get(`http://127.0.0.1:8000/update/${id}/`)
14      .then((response)=>{
15        setCity(response.data.city);
16        setState(response.data.state);
17        setCountry(response.data.country);
18      })
19  },[])
20
21  const handleChange = (e) => {
22    const { name, value } = e.target;
23    if (name === 'city') {
24      setCity(value);
25    } else if (name === 'state') {
26      setState(value);
27    } else if (name === 'country') {
28      setCountry(value);
29    }
30  };
31
32  let handleSubmit = (e)=>{
33    e.preventDefault();
34    console.log(city,state,country);
35    axios.put(`http://localhost:8000/update/${id}/`,{city,state,country})
36      .then((response)=>{
37        setCity(city);
38        setState(state);
39        setCountry(country);
40        alert("Record Updated Successfully...")
41        navigate("/");
42      })
43  }
44
45  return (
46    <form method="POST" onSubmit={handleSubmit}>
47      <input type="text"
48        value={city}
49        name="city"
50        onChange={handleChange}
51      />
52      <input type="text"
53        value={state}
54        name="state"
55        onChange={handleChange}
56      />
57      <input type="text"
58        value={country}
59        name="country"
60        onChange={handleChange}
61      />
62      <button type="submit">Update</button>
63    </form>
64  )
65 }
66
67 export default UpdateForm;
```

EXPLORER

...

App.jsxviews.pyurls.pyForm.jsxApp.cssGetDjango.jsxUpdateForm.jsx

▼ SUNDAY

▼ my-react-app

> node_modules

> public

▼ src

> assets

App.css

App.jsxDeleteForm.jsxForm.jsxGetDjango.jsx# index.cssLatestForm.jsxmain.jsxUpdateForm.jsx.eslintrc.cjs.gitignoreindex.htmlpackage-lock.jsonpackage.jsonpackage.jsonREADME.mdvite.config.js

> node_modules

> project

{ } package-lock.json

{ } package.json

my-react-app > src > DeleteForm.jsx > ...

```
1 import React, { useEffect } from "react";
2 import axios from "axios";
3 import {useState} from 'react';
4 import { useParams,useNavigate } from "react-router-dom";
5 function UpdateForm(){
6   const {id} = useParams();
7
8   const [city,setCity] = useState('');
9   const [state,setState] = useState('');
10  const [country,setCountry] = useState('');
11  const navigate = useNavigate();
12  // getting data from below link from django bakend with id on react js delete form
13  useEffect(()=>{
14    axios.get(`http://127.0.0.1:8000/home/${id}/`)
15    .then((response)=>{
16      setCity(response.data.city);
17      setState(response.data.state);
18      setCountry(response.data.country);
19    })
20  },[])
21
22  // on submitting post form it will redirect to django endpoint for delete and record will get deleted.
23  let handleSubmit = (e)=>{
24    e.preventDefault();
25    console.log(city,state,country);
26    axios.delete(`http://localhost:8000/delete/${id}/`,{city,state,country})
27    .then((response)=>{
28      alert("data has been deleted.")
29      navigate("/");
30    })
31  }
32
33  // form to retrieve and delete data
34  return (
35    <form method="POST" onSubmit={handleSubmit}>
36      <input type="text"
37        value={city}
38        name="city"
39      />
40      <input type="text"
41        value={state}
42        name="state"
43      />
44      <input type="text"
45        value={country}
46        name="country"
47      />
48      <button type="submit">Delete</button>
49    </form>
50  )
51 }
52
53 export default UpdateForm;
```


EXPLORER

SUNDAY

my-react-app

node_modules

public

src

assets

App.css

App.jsx

DeleteForm.jsx

Form.jsx

GetDjango.jsx

index.css

LatestForm.jsx

main.jsx

UpdateForm.jsx

.eslintrc.cjs

.gitignore

index.html

package-lock.json

package.json

README.md

vite.config.js

node_modules

project

package-lock.json

package.json

OUTLINE

TIMELINE

App.jsx

views.py

urls.py

Form.jsx

App.css

GetDjango.jsx

UpdateForm.jsx

my-react-app > src > # App.css > #root

```
1 #root {
2   max-width: 1280px;
3   margin: 0 auto;
4   padding: 2rem;
5   text-align: center;
6 }
7
8
9 @media (prefers-reduced-motion: no-preference) {
10   a:nth-of-type(2) .logo {
11     animation: logo-spin infinite 20s linear;
12   }
13 }
14
15 body{
16   background-color: #beige;
17   display: flex;
18   align-items: center;
19 }
20
21 h1{
22   color: #chartreuse;
23 }
24 form {
25   display: flex;
26   flex-direction: column;
27   align-items: center;
28   margin: 5rem;
29   padding: 5rem;
30   gap: 5px;
31   background-color: #646cffaa;
32   box-shadow: 0 0 30rem;
33   justify-content: stretch;
34   border-radius: 2rem;
35 }
36 input{
37   height: 2rem;
38   width: 20rem;
39   border-radius: 5rem;
40 }
41 input:active{
42   background-color: #goldenrod;
43 }
44 table {
45   width: 100%;
46   border-collapse: collapse;
47 }
48
49 table th, table td {
50   border: 1px solid #ddd;
51   padding: 8px;
52   text-align: left;
53   color: black;
54 }
55 }
```

my-react-app > src > # App.css > ...

```
21 h1{
22
23 }
24 form {
25   display: flex;
26   flex-direction: column;
27   align-items: center;
28   margin: 5rem;
29   padding: 5rem;
30   gap: 5px;
31   background-color: #646cffaa;
32   box-shadow: 0 0 30rem;
33   justify-content: stretch;
34   border-radius: 2rem;
35 }
36 input{
37   height: 2rem;
38   width: 20rem;
39   border-radius: 5rem;
40 }
41 input:active{
42   background-color: #goldenrod;
43 }
44 table {
45   width: 100%;
46   border-collapse: collapse;
47 }
48
49 table th, table td {
50   border: 1px solid #ddd;
51   padding: 8px;
52   text-align: left;
53   color: black;
54 }
55 }
56
57 table thead {
58   background-color: #f2f2f2;
59 }
60
61 table tr:nth-child(even) {
62   background-color: #f9f9f9;
63 }
64
65
66 nav{
67   display: flex;
68   gap: 5px;
69   align-items: center;
70 }
71
72 nav button {
73   background-color: #gold;
74 }
75 }
```