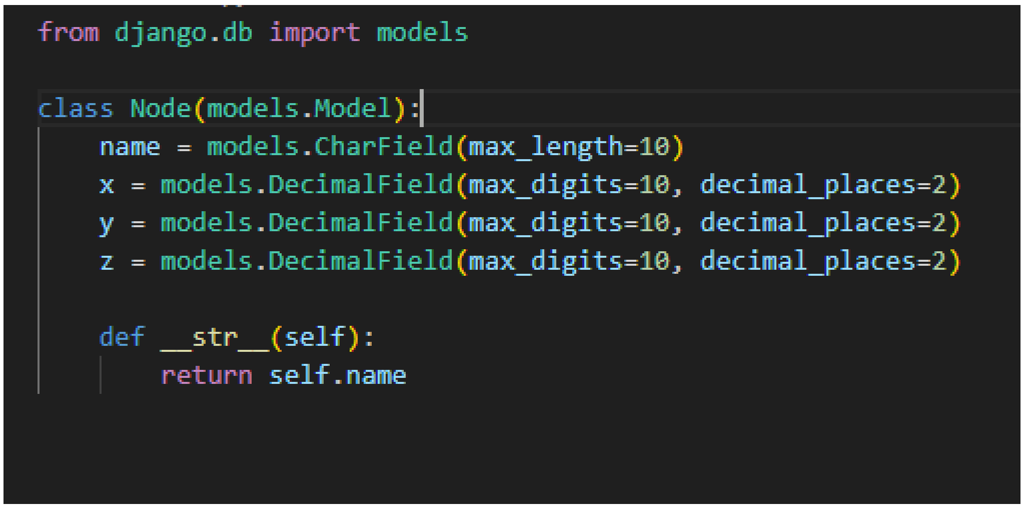
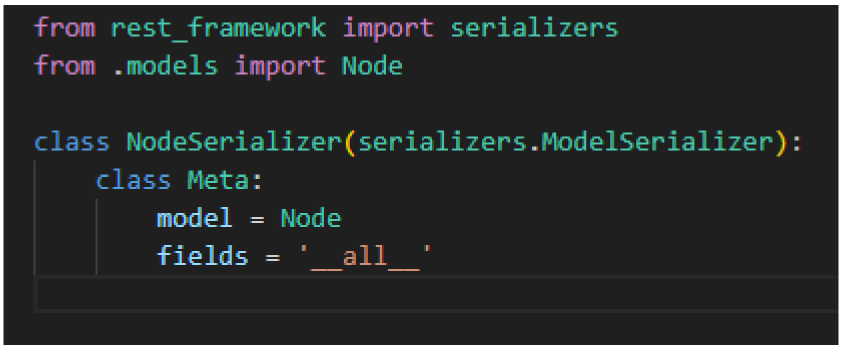
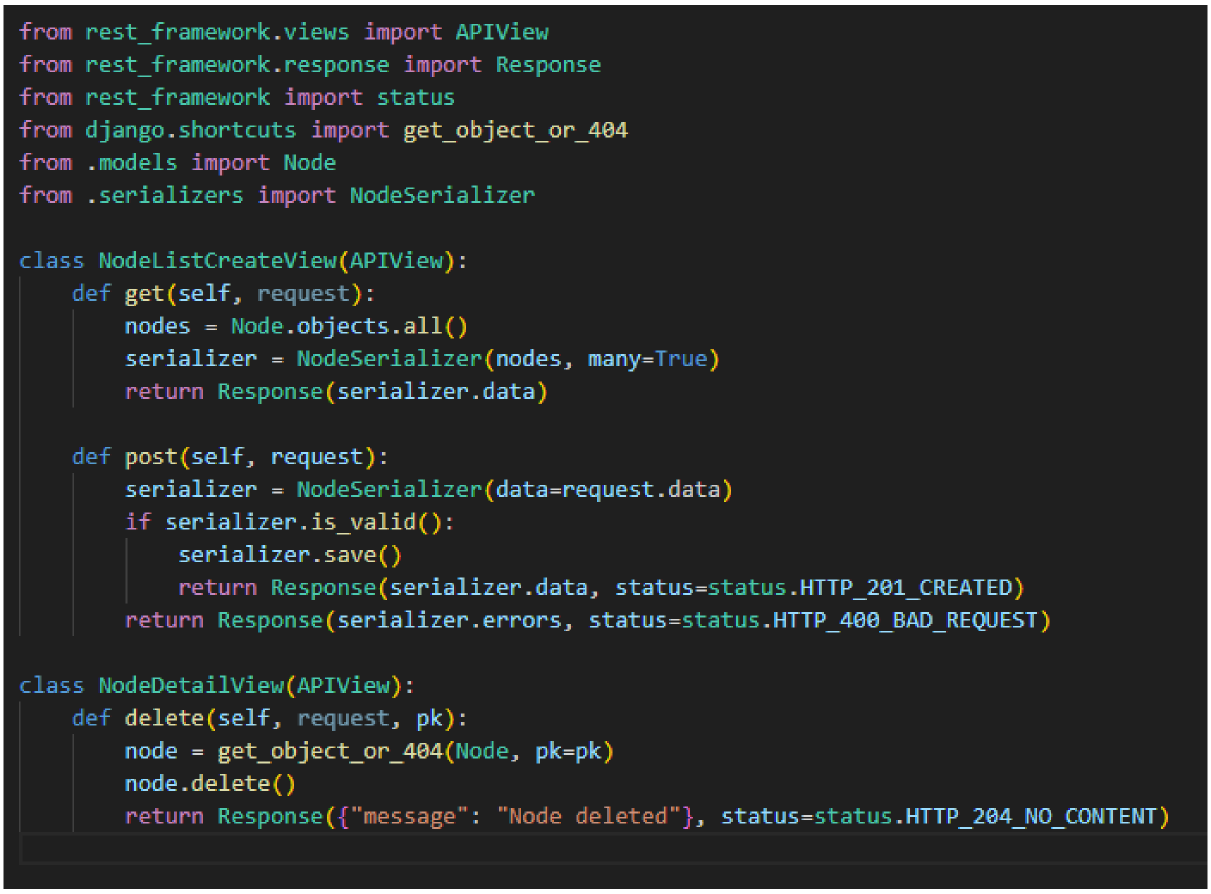


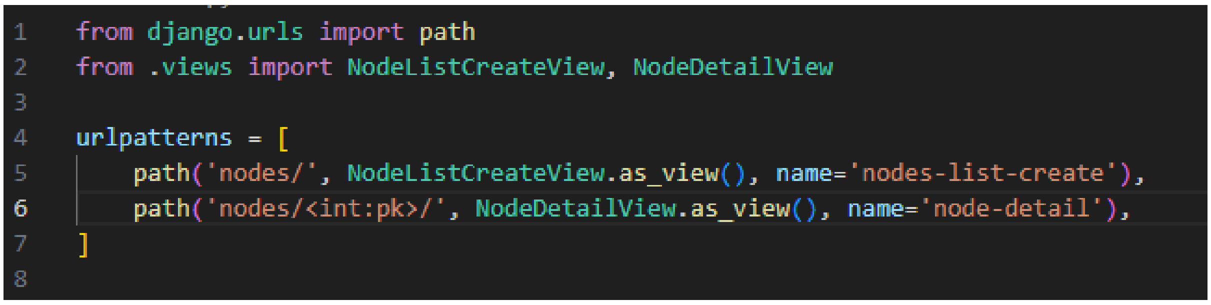
```python
from django.db import models

class Node(models.Model):
    name = models.CharField(max_length=10)
    x = models.DecimalField(max_digits=10, decimal_places=2)
    y = models.DecimalField(max_digits=10, decimal_places=2)
    z = models.DecimalField(max_digits=10, decimal_places=2)

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



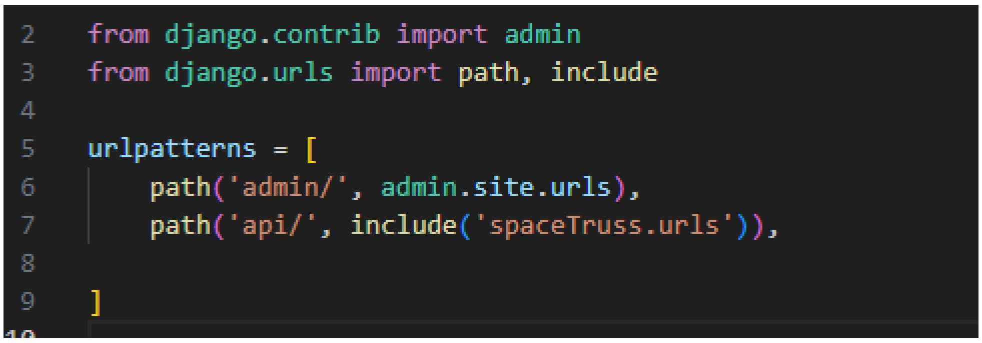
```python
from rest_framework import serializers
from .models import Node

class NodeSerializer(serializers.ModelSerializer):
    class Meta:
        model = Node
        fields = '__all__'
```



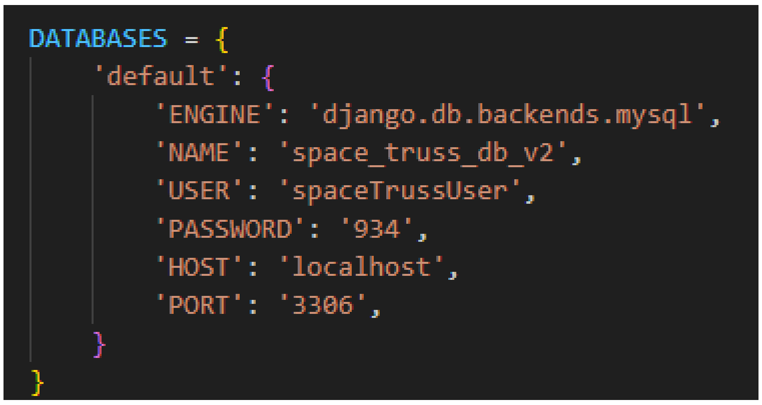
```python
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from django.shortcuts import get_object_or_404
from .models import Node
from .serializers import NodeSerializer

class NodeListCreateView(APIView):
    def get(self, request):
        nodes = Node.objects.all()
        serializer = NodeSerializer(nodes, many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = NodeSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

class NodeDetailView(APIView):
    def delete(self, request, pk):
        node = get_object_or_404(Node, pk=pk)
        node.delete()
        return Response({"message": "Node deleted"}, status=status.HTTP_204_NO_CONTENT)
```



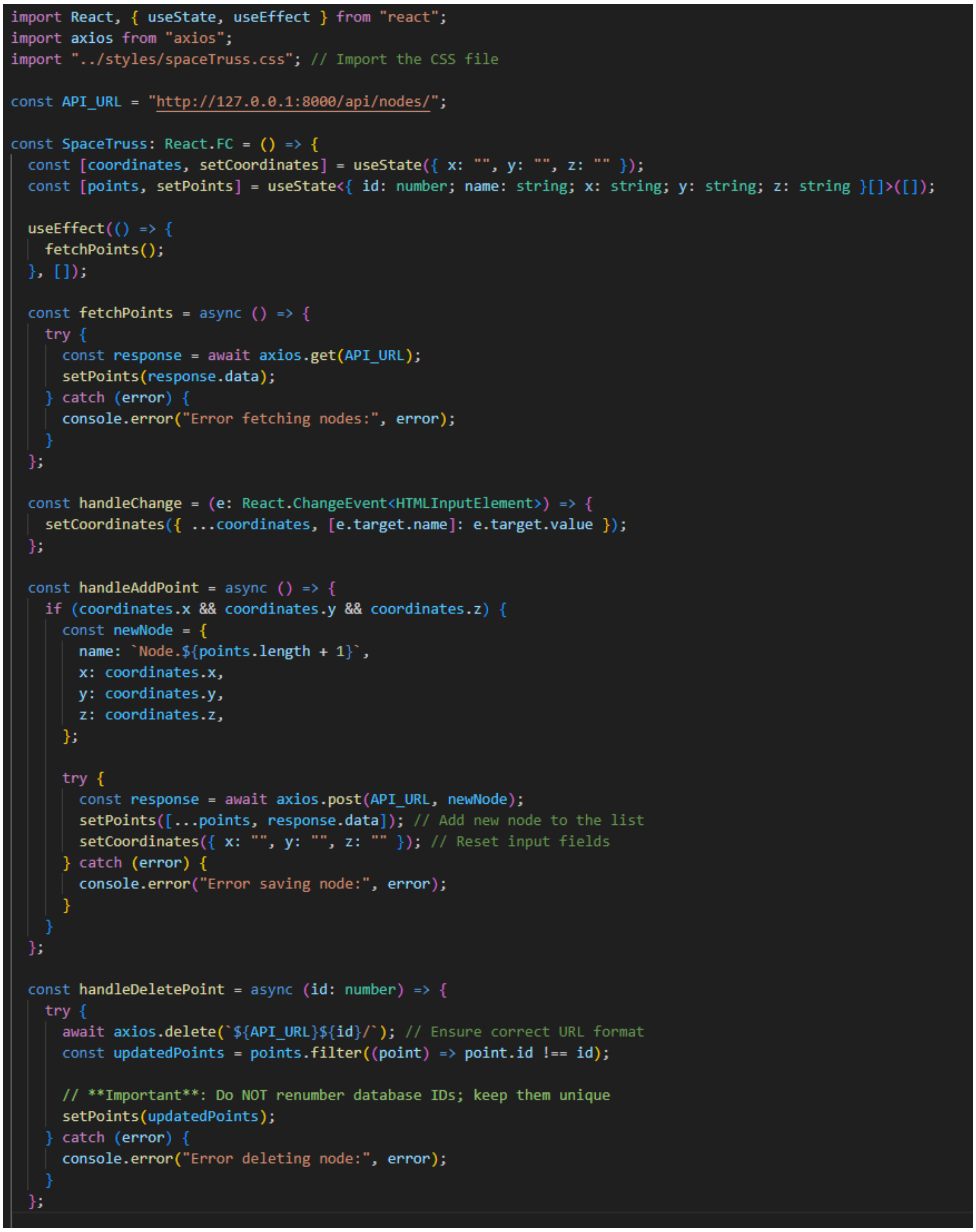
```python
from django.urls import path
from .views import NodeListCreateView, NodeDetailView

urlpatterns = [
    path('nodes/', NodeListCreateView.as_view(), name='nodes-list-create'),
    path('nodes/<int:pk>/', NodeDetailView.as_view(), name='node-detail'),
]

```



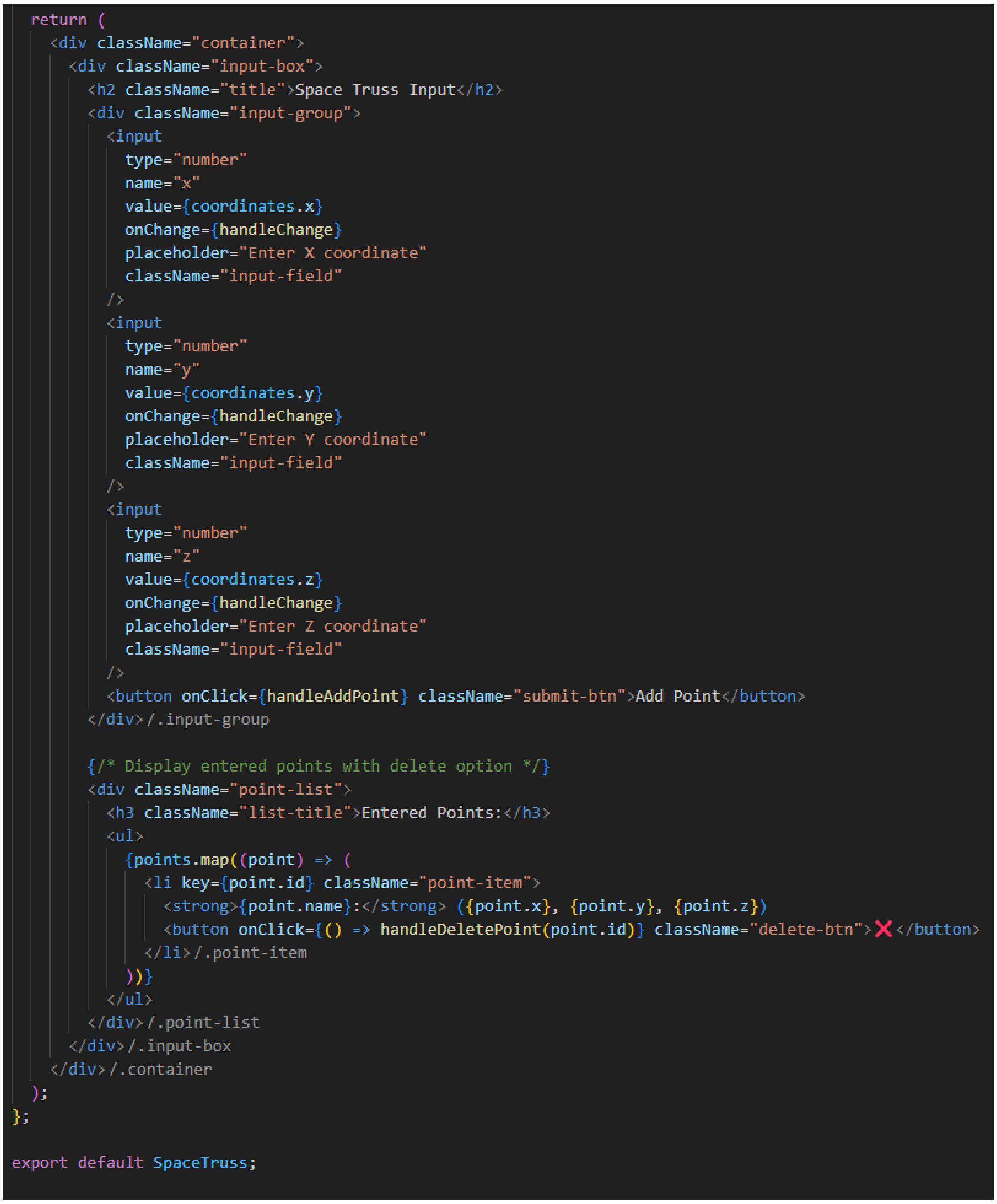
```python
from django.contrib import admin
from django.urls import path, include

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

]
```



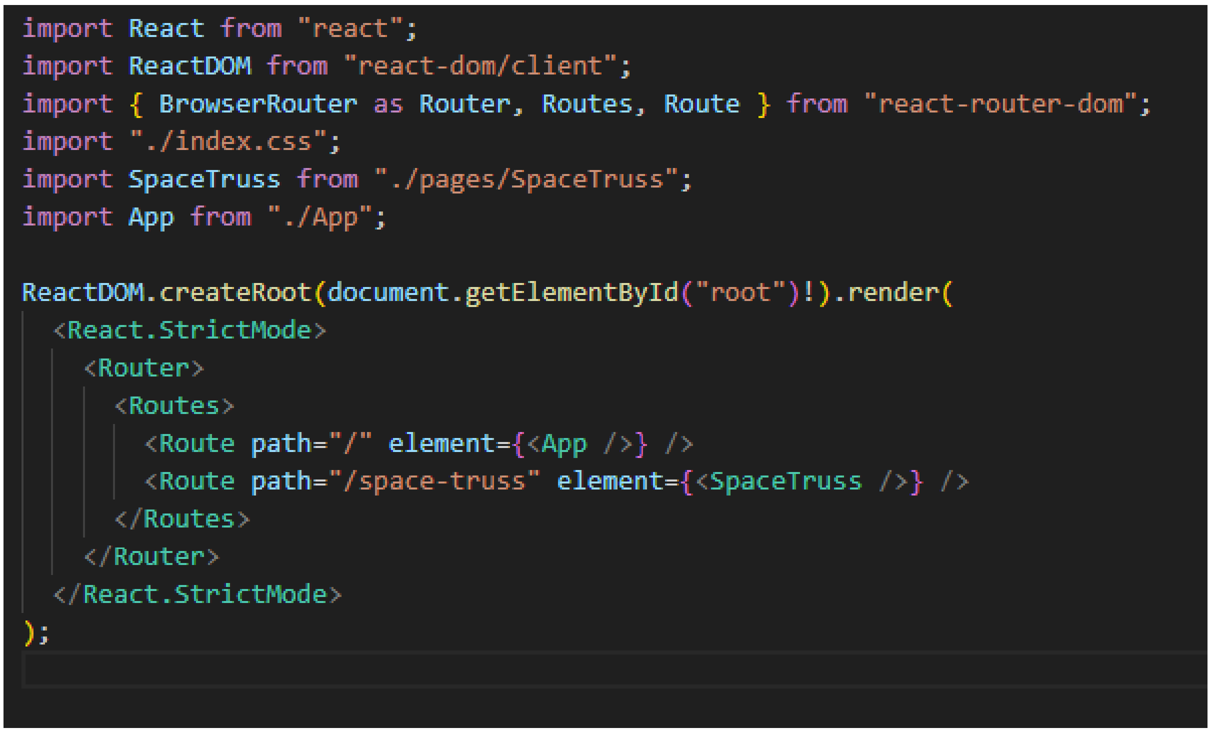
```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'space_truss_db_v2',
        'USER': 'spaceTrussUser',
        'PASSWORD': '934',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```



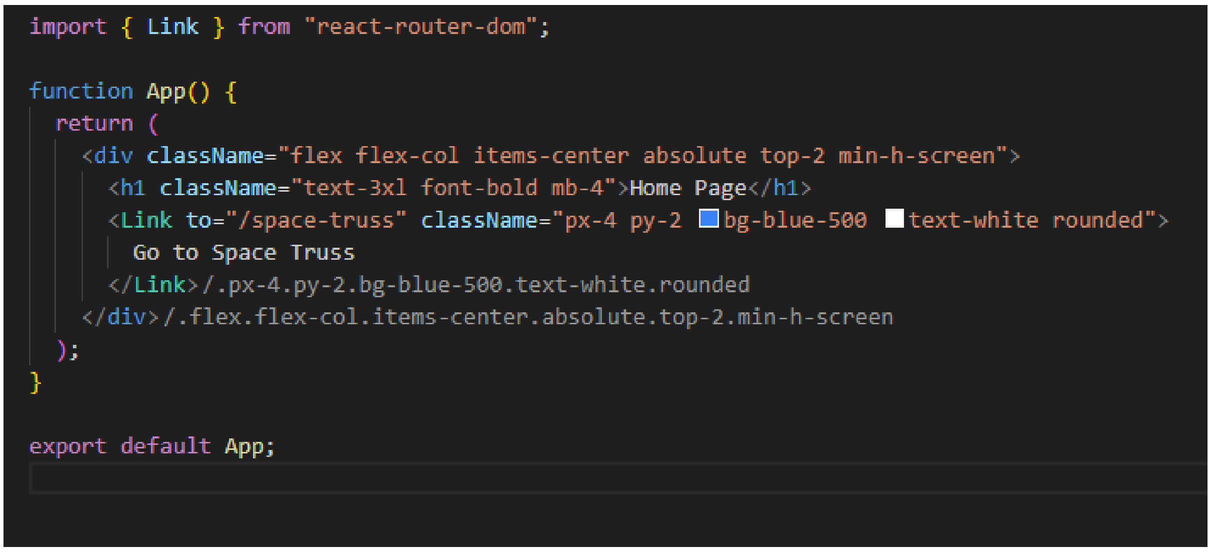
```tsx
import React, { useState, useEffect } from "react";
import axios from "axios";
import "../styles/spaceTruss.css"; // Import the CSS file

const API_URL = "http://127.0.0.1:8000/api/nodes/";

const SpaceTruss: React.FC = () => {
  const [coordinates, setCoordinates] = useState({ x: "", y: "", z: "" });
  const [points, setPoints] = useState<{ id: number; name: string; x: string; y: string; z: string }[]>([]);

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

  const fetchPoints = async () => {
    try {
      const response = await axios.get(API_URL);
      setPoints(response.data);
    } catch (error) {
      console.error("Error fetching nodes:", error);
    }
  };

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
    setCoordinates({ ...coordinates, [e.target.name]: e.target.value });
  };

  const handleAddPoint = async () => {
    if (coordinates.x && coordinates.y && coordinates.z) {
      const newNode = {
        name: `Node.${points.length + 1}`,
        x: coordinates.x,
        y: coordinates.y,
        z: coordinates.z,
      };

      try {
        const response = await axios.post(API_URL, newNode);
        setPoints([...points, response.data]); // Add new node to the list
        setCoordinates({ x: "", y: "", z: "" }); // Reset input fields
      } catch (error) {
        console.error("Error saving node:", error);
      }
    }
  };

  const handleDeletePoint = async (id: number) => {
    try {
      await axios.delete(`${API_URL}${id}/`); // Ensure correct URL format
      const updatedPoints = points.filter((point) => point.id !== id);

      // **Important**: Do NOT renumber database IDs; keep them unique
      setPoints(updatedPoints);
    } catch (error) {
      console.error("Error deleting node:", error);
    }
  };
```



```jsx
  return (
    <div className="container">
      <div className="input-box">
        <h2 className="title">Space Truss Input</h2>
        <div className="input-group">
          <input
            type="number"
            name="x"
            value={coordinates.x}
            onChange={handleChange}
            placeholder="Enter X coordinate"
            className="input-field"
          />
          <input
            type="number"
            name="y"
            value={coordinates.y}
            onChange={handleChange}
            placeholder="Enter Y coordinate"
            className="input-field"
          />
          <input
            type="number"
            name="z"
            value={coordinates.z}
            onChange={handleChange}
            placeholder="Enter Z coordinate"
            className="input-field"
          />
          <button onClick={handleAddPoint} className="submit-btn">Add Point</button>
        </div>/.input-group

        {/* Display entered points with delete option */}
        <div className="point-list">
          <h3 className="list-title">Entered Points:</h3>
          <ul>
            {points.map((point) => (
              <li key={point.id} className="point-item">
                <strong>{point.name}:</strong> ({point.x}, {point.y}, {point.z})
                <button onClick={() => handleDeletePoint(point.id)} className="delete-btn">❌</button>
              </li>/.point-item
            ))}
          </ul>
        </div>/.point-list
      </div>/.input-box
    </div>/.container
  );
};

export default SpaceTruss;
```



```
import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import "./index.css";
import SpaceTruss from "./pages/SpaceTruss";
import App from "./App";

ReactDOM.createRoot(document.getElementById("root")!).render(
  <React.StrictMode>
    <Router>
      <Routes>
        <Route path="/" element={<App />} />
        <Route path="/space-truss" element={<SpaceTruss />} />
      </Routes>
    </Router>
  </React.StrictMode>
);
```



```
import { Link } from "react-router-dom";

function App() {
  return (
    <div className="flex flex-col items-center absolute top-2 min-h-screen">
      <h1 className="text-3xl font-bold mb-4">Home Page</h1>
      <Link to="/space-truss" className="px-4 py-2 bg-blue-500 text-white rounded">
        Go to Space Truss
      </Link>/.px-4.py-2.bg-blue-500.text-white.rounded
    </div>/.flex.flex-col.items-center.absolute.top-2.min-h-screen
  );
}

export default App;
```