1. th
2. npm install –no-fund three
3. npm install –save-dev vite
4. npx vite

/\*\*import \* as THREE from 'three';

import { GLTFLoader } from 'three/examples/jsm/loaders/GLTFLoader.js'; // Updated import path

const scene = new THREE.Scene();

const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);

const renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );

document.body.appendChild( renderer.domElement );

// Load GLTF model

const loader = new GLTFLoader();

loader.load(

    'models/table\_and\_chairs.glb',

    function (gltf) {

        gltf.scene.scale.set(10, 10, 10);

        console.log('GLTF model loaded successfully:', gltf);

        scene.add(gltf.scene);

    },

    undefined,

    function (error) {

        console.error(error);

    }

);

function animate() {

    requestAnimationFrame(animate);

    renderer.render(scene, camera);

}

animate();\*\*/

import \* as THREE from 'three';

const scene = new THREE.Scene();

const camera = new THREE.PerspectiveCamera( 100, window.innerWidth / window.innerHeight, 0.1, 2000 );

const renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );

document.body.appendChild( renderer.domElement );

const geometry = new THREE.BoxGeometry( 1, 1, 1 );

const material = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );

const cube = new THREE.Mesh( geometry, material );

scene.add( cube );

camera.position.z = 5;

function animate() {

    requestAnimationFrame( animate );

    cube.rotation.x += 0.01;

    cube.rotation.y += 0.01;

    renderer.render( scene, camera );

}

animate();