In App.js

```
import React, { Component } from 'react'
import Project from './component/Project'
import './App.css';
export class App extends Component {
  constructor(props){
    super(props);
    this.canvasRef= React.createRef();
  }

  componentDid

  render() {
    return (
      <div>
        <canvas className="Webgl" ref={this.canvasRef} ></canvas>

        <nav>
        <a href='/'>Sphere</a>
          <ul>
            <li>good</li>
            <li>bad</li>
            <li>things</li>
          </ul>
        </nav>
        <h1 className="title">goood analysis</h1>

        <Project canvasRef={this.canvasRef} />
      </div>
    )
  }
}

export default App
```

in project.js

```
import React, { Component } from 'react'
import * as THREE from 'three';
import {OrbitControls} from 'three/examples/jsm/controls/OrbitControls'
export class Project extends Component {
    //constructor(props){
      //  super(props);
       // };
    //shape
    componentDidMount(){
        //canvas ref from App.js
        this.canvasRef=this.props.canvasRef;

        //create scene
        this.scene=new THREE.Scene();

        //create the shape by providing geometry aand material
        const geometry=new THREE.SphereGeometry(3,64,64);


        const material =new THREE.MeshStandardMaterial({
            color:"#00ff83",
        });
        const mesh =new THREE.Mesh(geometry,material);
        this.scene.add(mesh)


        //Sizes of the view port
        this.size={
            width:window.innerWidth,
            height:window.innerHeight
        }

        //Light
        this.light= new THREE.PointLight(0xffffff,1,100);
        this.light.position.set(0,10,10)//[1(-num)=left],[1(+num)=right],[2(-
num)=goes to down],[2(+num)=top],[3(-num)=back],[3(+num)=front]
        this.scene.add(this.light);


        //camera
        this.camera=new
THREE.PerspectiveCamera(45,this.size.width/this.size.height,0.1,100)
        //camera.position
        this.camera.position.z=20

        this.scene.add(this.camera)
```

```
        //Renderer
        this.renderer=new THREE.WebGLRenderer({
            canvas:this.canvasRef.current
        });
        this.renderer.setSize(this.size.width,this.size.height);
        this.renderer.setPixelRatio(2)
        this.renderer.render(this.scene,this.camera);

        //controles
        const controls =new OrbitControls(this.camera,this.canvasRef.current)
        controls.enableDamping=true
        controls.enablePan=false
        controls.enableZoom=false
        controls.autoRotate=true
        controls.autoRotateSpeed=5


        //Resize
        window.addEventListener('resize',()=>{
            //update Sizes
            console.log(window.innerWidth);
            this.size.width=window.innerWidth;
            this.size.height=window.innerHeight;

            //update the width of camera
            this.camera.aspect=this.size.width/this.size.height;
            this.camera.updateProjectionMatrix();
            this.renderer.setSize(this.size.width,this.size.height)
        })

        const loop=()=>{
            this.renderer.render(this.scene,this.camera)
            window.requestAnimationFrame(loop);
            controls.update()
        }
        loop()
    }
```

```
    render() {
        return (
      <div>

      </div>
    )
  }
}

export default Project
```

In App.css

```
*{
  margin:0;
  padding:0;
  box-sizing: border-box;
}

body,html{
  overflow-x: hidden;
}

.Webgl{
  position: absolute;
  top:0;
  left:0;
  z-index:1;

}

nav{
  color:white;
  z-index:2;
  position:relative;
  padding:4rem 8rem;
```

```
  display: flex;
  justify-content: space-between;

}

nav a{
  text-decoration: none;
  color:white;
  font-weight: bold ;
}

nav ul{
  list-style: none;
  display: flex;
  gap:4rem;

}

.title{
  color:white;
  z-index:2;
  position: absolute;
  font-size: 3rem;
  left:50%;
  top:75%;
  transform: translate(-50%,-75%);

}
```