// Import CesiumJS

const Cesium = require('cesium');

// Create a Cesium Viewer

const viewer = new Cesium.Viewer('cesiumContainer');

// Define the 5 sensors

const sensors = [

{

name: 'Sensor 1',

position: Cesium.Cartesian3.fromDegrees(longitude: -100.0, latitude: 40.0, height: 0.0),

submenuOptions: [

{ name: 'Option 1', value: 'option1' },

{ name: 'Option 2', value: 'option2' },

{ name: 'Option 3', value: 'option3' }

]

},

{

name: 'Sensor 2',

position: Cesium.Cartesian3.fromDegrees(longitude: -105.0, latitude: 40.0, height: 0.0),

submenuOptions: [

{ name: 'Option 1', value: 'option1' },

{ name: 'Option 2', value: 'option2' },

{ name: 'Option 3', value: 'option3' }

]

},

{

name: 'Sensor 3',

position: Cesium.Cartesian3.fromDegrees(longitude: -110.0, latitude: 40.0, height: 0.0),

submenuOptions: [

{ name: 'Option 1', value: 'option1' },

{ name: 'Option 2', value: 'option2' },

{ name: 'Option 3', value: 'option3' }

]

},

{

name: 'Sensor 4',

position: Cesium.Cartesian3.fromDegrees(longitude: -115.0, latitude: 40.0, height: 0.0),

submenuOptions: [

{ name: 'Option 1', value: 'option1' },

{ name: 'Option 2', value: 'option2' },

{ name: 'Option 3', value: 'option3' }

]

},

{

name: 'Sensor 5',

position: Cesium.Cartesian3.fromDegrees(longitude: -120.0, latitude: 40.0, height: 0.0),

submenuOptions: [

{ name: 'Option 1', value: 'option1' },

{ name: 'Option 2', value: 'option2' },

{ name: 'Option 3', value: 'option3' }

]

}

];

// Create a Windows layout with 5 tabs for each sensor

const tabs = document.getElementById('tabs');

for (let i = 0; i < sensors.length; i++) {

const tab = document.createElement('div');

tab.innerHTML = `

<h2>${sensors[i].name}</h2>

<select id="submenu-${i}">

${sensors[i].submenuOptions.map(option => `<option value="${option.value}">${option.name}</option>`).join('')}

</select>

<button id="register-${i}">Register Tracking</button>

`;

tabs.appendChild(tab);

// Add an event listener to the register button

document.getElementById(`register-${i}`).addEventListener('click', () => {

const selectedOption = document.getElementById(`submenu-${i}`).value;

console.log(`Registering tracking for ${sensors[i].name} with option ${selectedOption}`);

// Send a request to the server to register the tracking

fetch('http://localhost:8080/register-tracking', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

sensorName: sensors[i].name,

option: selectedOption

})

})

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error(error));

});

}

// Initialize the Cesium viewer

viewer.camera.setView({

destination: Cesium.Cartesian3.fromDegrees(longitude: -105.0, latitude: 40.0, height: 2000000.0),

orientation: {

heading: Cesium.Math.toRadians(0),

pitch: Cesium.Math.toRadians(-90),

roll: Cesium.Math.toRadians(0)

}

});

// Add a data source to the Cesium viewer

const dataSource = new Cesium.CustomDataSource();

viewer.dataSources.add(dataSource);

// Create a function to update the Cesium viewer with new data

function updateCesiumViewer(data) {

// Clear the existing entities

dataSource.entities.removeAll();

// Add