```

Backend Flask API

from flask import Flask, jsonify

import math

app = Flask(\_\_name\_\_)

Define the antenna's properties

antenna = {

'name': 'Antenna',

'position': [0, 0, 0],

'orientation': [0, 0, 0],

'frequency': 2.4e9, # Hz

'gain': 10, # dBi

'radiation\_pattern': {

'type': 'directional',

'beamwidth': 30, # degrees

'sides': 10

}

}

Define the antenna's elements

elements = [

{

'name': 'Element 1',

'position': [0, 0, 0],

'orientation': [0, 0, 0],

'type': 'dipole'

},

{

'name': 'Element 2',

'position': [0.5, 0, 0],

'orientation': [0, 0, 0],

'type': 'reflector'

},

{

'name': 'Element 3',

'position': [1, 0, 0],

'orientation': [0, 0, 0],

'type': 'director'

}

]

Define a function to calculate the radiation pattern

def calculate\_radiation\_pattern(antenna, elements):

# Calculate the radiation pattern using the antenna's properties and elements

radiation\_pattern = []

for i in range(antenna['radiation\_pattern']['sides']):

angle = i \* 360 / antenna['radiation\_pattern']['sides']

gain = calculate\_gain(antenna, elements, angle)

radiation\_pattern.append([angle, gain])

return radiation\_pattern

Define a function to calculate the gain

def calculate\_gain(antenna, elements, angle):

# Calculate the gain using the antenna's properties and elements

gain = antenna['gain']

for element in elements:

if element['type'] == 'reflector':

gain += calculate\_reflector\_gain(element, angle)

elif element['type'] == 'director':

gain += calculate\_director\_gain(element, angle)

return gain

Define a function to calculate the reflector gain

def calculate\_reflector\_gain(element, angle):

# Calculate the reflector gain using the element's properties and angle

reflector\_gain = 10 \* math.log10(math.cos(math.radians(angle)))

return reflector\_gain

Define a function to calculate the director gain

def calculate\_director\_gain(element, angle):

# Calculate the director gain using the element's properties and angle

director\_gain = 10 \* math.log10(math.cos(math.radians(angle)))

return director\_gain

Define a route to return the antenna's properties

@app.route('/antenna', methods=['GET'])

def get\_antenna():

return jsonify(antenna)

Define a route to return the antenna's elements

@app.route('/antenna/elements', methods=['GET'])

def get\_antenna\_elements():

return jsonify(elements)

Define a route to return the antenna's radiation pattern

@app.route('/antenna/radiation-pattern', methods=['GET'])

def get\_antenna\_radiation\_pattern():

radiation\_pattern = calculate\_radiation\_pattern(antenna, elements)

return jsonify(radiation\_pattern)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

```

```

<!-- Frontend CesiumJS -->

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Antenna Radiation Pattern</title>

<script src="https://unpkg.com/cesium@1.94/Build/Cesium/Cesium.js"></script>

<style>

#cesiumContainer {

width: 100%;

height: 600px;

margin: 0;

padding: 0;

overflow: hidden;

}

</style>

</head>

<body>

<div id="cesiumContainer"></div>

<script>

// Initialize the Cesium viewer

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

// Load the antenna's properties from the backend API

fetch('/antenna')

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

.then(antenna => {

// Load the antenna's elements from the backend API

fetch('/antenna/elements')

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

.then(elements => {

// Load the antenna's radiation pattern from the backend API

fetch('/antenna/radiation-pattern')

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

.then(radi

```