Given a JSON schema as follows,

{

"$schema": "http://json-schema.org/draft-07/schema#",

"type": "object",

"properties": {

"selectedAttribute": {

"type": "string",

"description": "The attribute selected by the user for exploration"

},

"data": {

"type": "array",

"items": {

"type": "object",

"properties": {

"attribute1": {

"type": "number",

"description": "Description of Attribute 1"

},

"attribute2": {

"type": "number",

"description": "Description of Attribute 2"

},

"attribute3": {

"type": "number",

"description": "Description of Attribute 3"

},

"targetAttribute": {

"type": "number",

"description": "The target attribute for potential linear regression"

}

},

"required": ["attribute1", "attribute2", "attribute3", "targetAttribute"]

}

}

},

"required": ["selectedAttribute", "data"]

}

Your tool should generate something similar to,

import React, { useState, useEffect } from 'react';

import AttributeSelector from './AttributeSelector';

import AttributeVisualization from './AttributeVisualization';

import DataService from './services/DataService'; // You might need to create a service to handle data fetching and processing

const InteractiveTool = () => {

const [selectedAttribute, setSelectedAttribute] = useState(null);

const [data, setData] = useState([]);

// useEffect to fetch initial data when the component mounts

useEffect(() => {

// Replace this with actual data fetching logic

const fetchData = async () => {

const result = await DataService.fetchData();

setData(result);

};

fetchData();

}, []);

// useEffect to update data when the selected attribute changes

useEffect(() => {

if (selectedAttribute) {

// Replace this with actual data processing logic based on the selected attribute

const updatedData = DataService.processData(selectedAttribute, data);

setData(updatedData);

}

}, [selectedAttribute, data]);

return (

<div>

<h1>Interactive Tool</h1>

<AttributeSelector

attributes={/\* List of attributes from your data \*/}

onSelect={(attribute) => setSelectedAttribute(attribute)}

/>

<AttributeVisualization data={data} selectedAttribute={selectedAttribute} />

</div>

);

};

export default InteractiveTool;