Dropsource Javascript Challenge

Background

Our application lets users visually design their mobile app. One of the use cases we support lets users create a page and add certain mobile elements to that page. Our application then renders the page and its elements.

Objectives

- To create a javascript function that renders a page and its elements.
- To allow the user to select a page or element

Requirements

The page and its elements are represented via an Element object. The Element object has the following properties:

Property	Type	Description
id	string	Uniquely identifies the Element
type	string	Determines the type of element. For the purpose of this exercise there will be three types: page, view, button and label.
frame	object	Object that represents the relative position of the element within its parent. The frame has left, top, width and height (all in pixels).
properties	object	Object that represents the visual properties of each Element. Different element types will have different properties. For example, most elements have a text property, but a view does not.
children	array	List of Element objects that are children of this Element. This Element is said to be their parent.

Sample Element object

```
{
	id: ‘page-1’,
	type: ‘page’
	frame: {
		width: 375,
		height: 592
	},
	properties: {
		backgroundColor: ‘white’
	},
	children: [
		{
			id: ‘label-1’,
			type: ‘label’
			frame: {
				left: 20,
				top: 20,
				width: 40,
				height: 40
			},
			properties: {
				text: ‘Foo’
			},
			children: []
		}
	]
}
```

Rendering each Element type
Each element should be positioned relative to its parent using the frame object. The left/top positions of the frame object are given relative to the top left corner of the parent element.

The page can be positioned anywhere on the screen.

Sibling elements should be rendered in the order in which they appear in the children array.

Every element type has its own rendering requirements:

Type	**Rendering Requirements**
page	Its background color comes from its properties. It should always have a dashed border of 2px of color #1a40a6.
view	Its background color comes from its properties. It should not render text.
label	Its background color will always be transparent. It should render text from its properties. Text will always be left-aligned. Text will always be black.
button	Its background color comes from its properties. It should render text from its properties. Text will always be centered. Text will always be black.

The Element object you should use is serialized into JSON and stored at this endpoint:
https://s3.amazonaws.com/dropsource/code-challenge/element-v1.json

When the application loads it should fetch the JSON object from the CDN above. It will then use the object to render the page and its elements.

While the data is being fetched the application should display a ‘Fetching...’ message. If fetching fails then it should display ‘Error fetching data’.

Selecting an element
When the user clicks an element, the element should become selected. A selected element has a solid border of 2px of color #00FFFF. The page should also be selectable. There can only be one selected element at a time.

Notes

- You may use any javascript framework or just keep it vanilla.
- You must use Javascript ES6.
- All of your code should go in index.html. Please only submit index.html to us.
- You must solve this challenge alone. You may not solicit direct help solving this problem.
- Using online resources, like Stack Overflow or Mozilla Developer Network (MDN) is acceptable.
- Use of HTML5 Canvas is not permitted.
- If possible, please don't post your code in github (or any other public source repository). We wouldn't want other candidates copying your solution.

End Result

When done, the page and elements should resemble the following diagram.