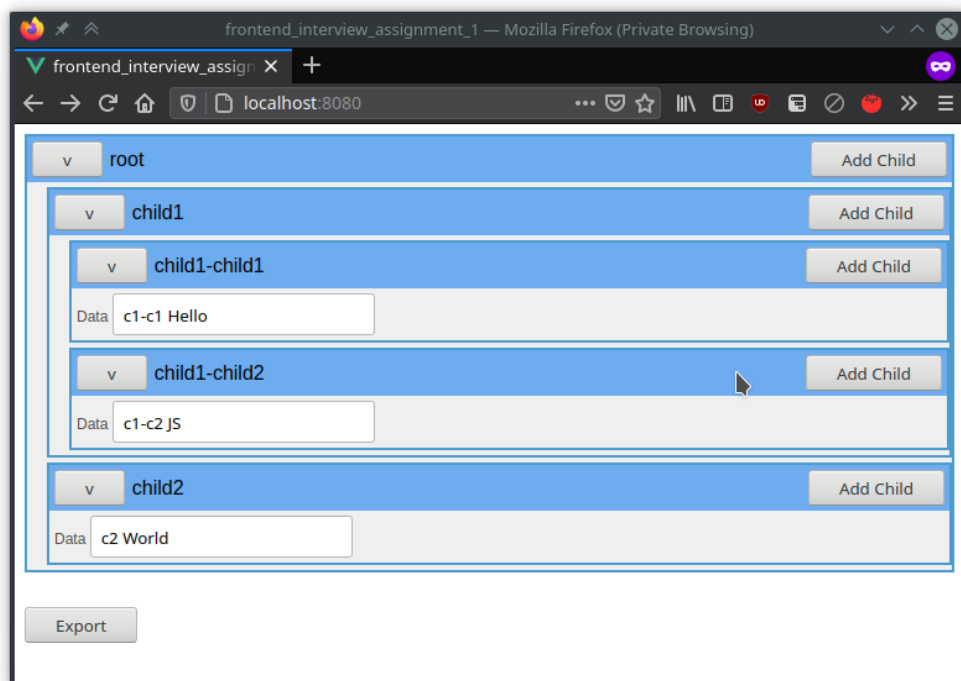


Frontend Coding Assignment

Description



Create a UI for Nested Tags Tree as shown in the screenshot above. The UI must contain a “TagView” component that supports rendering nested Tags children and text data. The tree hierarchy (data structure) for the Tags Tree is as given below:

```
tree: {
  name: 'root',
  children: [
    {
      name: 'child1',
      children: [
        {name: 'child1-child1', data: "c1-c1 Hello"},
        {name: 'child1-child2', data: "c1-c2 JS"}
      ]
    }
  ]
}
```

```

    },
    {name: 'child2', data: "c2 World"}
  ]
}

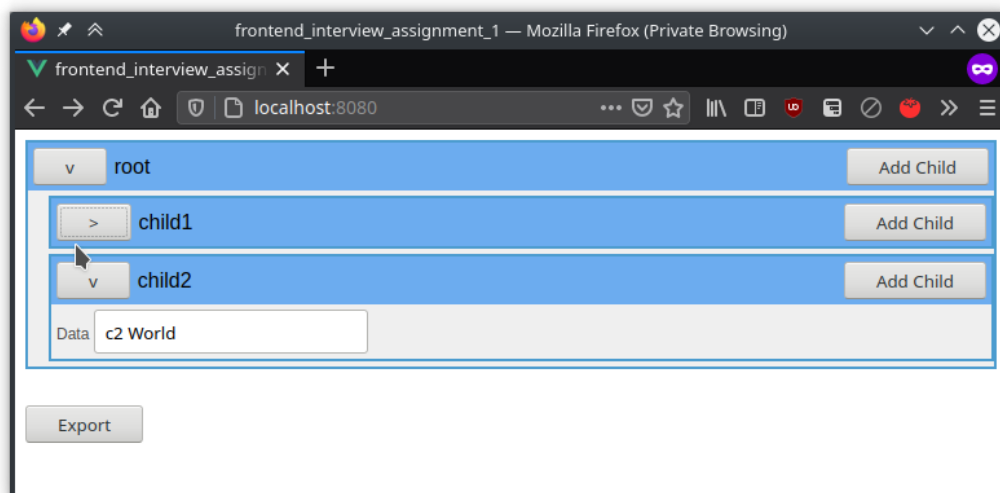
```

In the above nested tag tree data-structure, each tag element has a “name” property along with either a “children” Array or a “data” String but not both. The Tags elements can be nested recursively inside the children Array as shown in the example data structure. In addition, each Tag element must have an “Add Child” button as shown in the screenshot above. On clicking the “Add Child” button, a new TagView component must be instantiated inside its parent by replacing the “Data” textfield and its value..

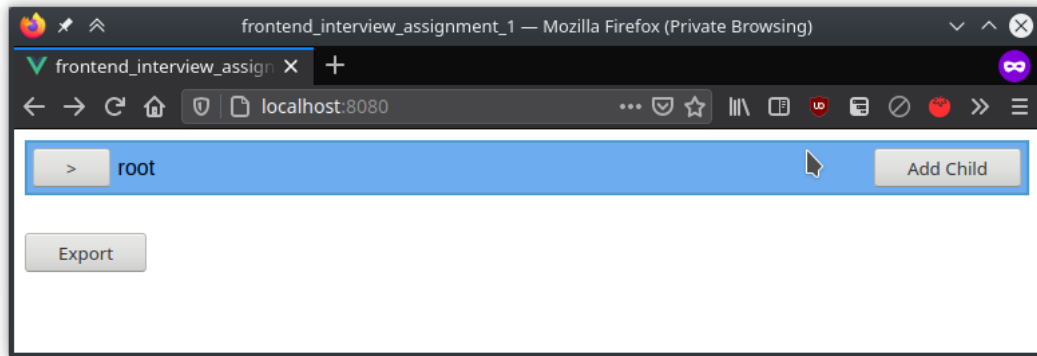
Very Important: The above data-structure is an example data structure. The tree hierarchy (data structure) can be altered by adding new children using the “Add Child” button or by modifying the tree object in the source code. The UI must update and render the new hierarchy as and when new children are added.

Requirements

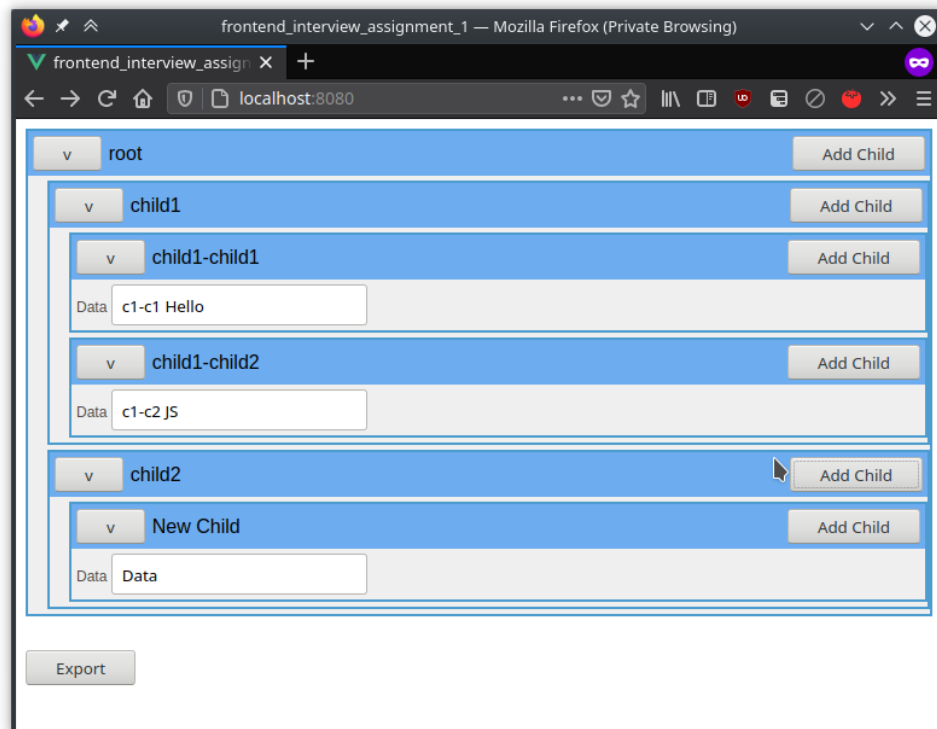
1. The “data” String inside each Tag should be rendered using a text input field. Editing the contents of the text input elements should modify the contents of the internal tree data structure object.
2. Each Tag must be collapsible on clicking the “v” or “>” button next to the tag name header. The collapsed Tag must display only the “name” header but hide its “children” and “data” string as shown in the screenshot below.



Note: Every Tag element must be collapsible recursively, including the root tag as shown in the below screenshot. If a tag is collapsed, then the leftmost button should show the string “>” else it must show “v”.

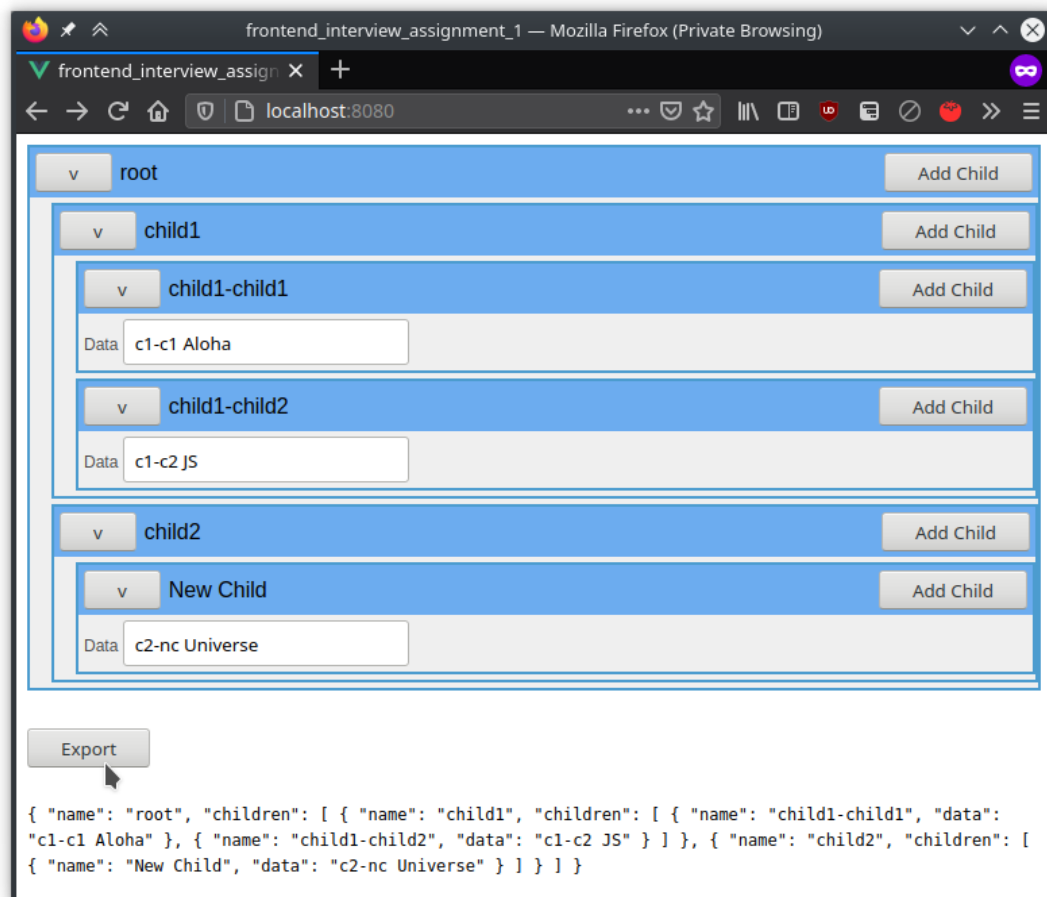


- When the “Add Child” button is clicked, then a new child must be created inside the parent. If the parent had a “data” property and textfield, then the “data” property and textfield must be replaced with a “children” property and single new child must be added inside it. The new child Tag element must have the name property set to “New Child” and “data” property must be “Data” as shown in the screenshot below.



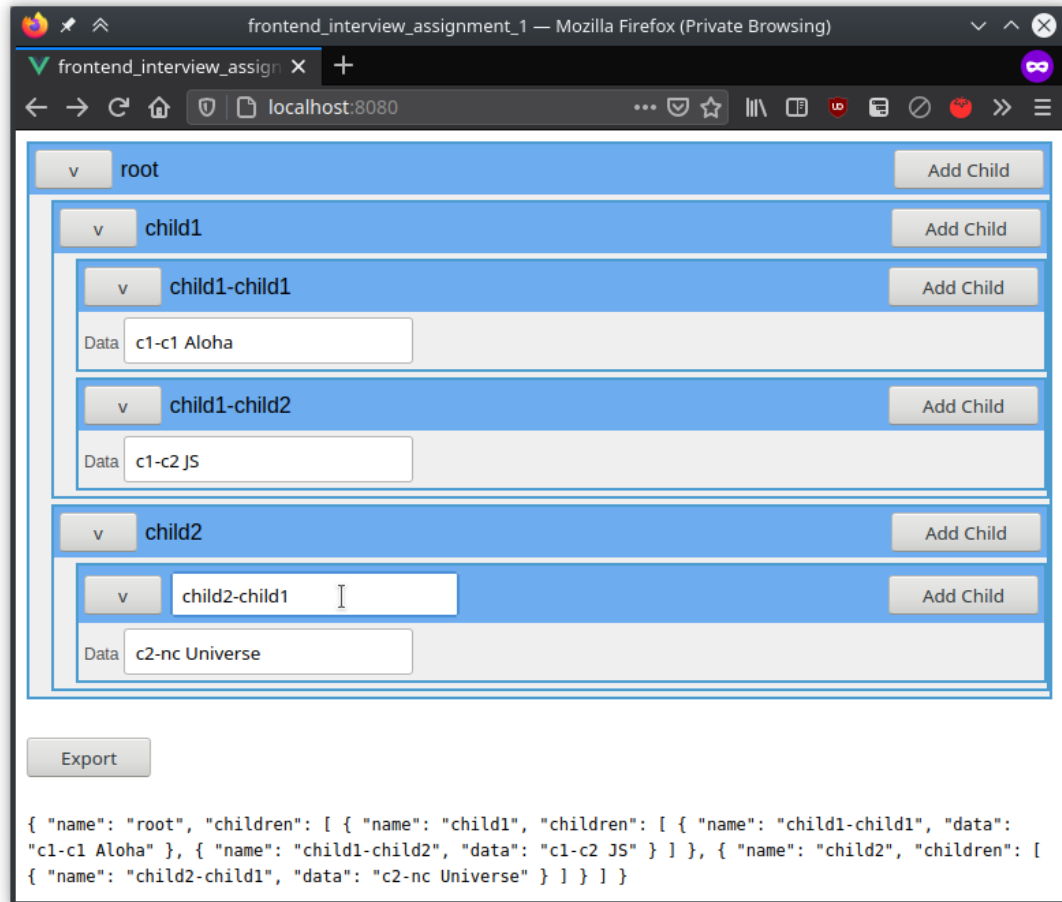
- There must be an “Export” button which must output the current data and JSON structure of the tree hierarchy on click as shown in the screenshot below.

(Hint: You can use `JSON.stringify` to convert the tree object to string, the necessary properties can be extracted recursively. Also, you must only output the “name”, “children”, and “data” properties of each Tag, all other internal values must not be present in the exported string.)



- All modified values of the “data” property for each Tag must be reflected in the exported string on clicking the export button. For example, the “data” string for the first textfield has been modified to “c1-c1 Aloha” which is reflected in the exported string as shown in the screenshot above.
- Bonus Question:** It should be possible to modify the name of any Tag element by clicking on the tag name displayed in the blue header as shown in the screenshot below. A new text input form element should be shown in place of the name. The user must be able to type a new name in the textfield and must press the Enter key on the keyboard to set the

new name and hide the input text field. The newly typed name must be shown and updated on the UI and JSON exported string.



Note

You can either style the Tag Elements as shown in the screenshots or you may use your own creativity to style them using any CSS framework of your choice to make it look nicer.

Please share your solution in a zip file containing the full project along with all the source files and package.json file or as a github repository link.

