As a part of the assignment, below is the Problem Statement from Locus. We request you to complete the assignment within 3-4 hours.

Problem Statment:

Keyboard Shortcuts Library

John works at Locus. Off-late John has been incorporating a lot of keyboard shortcuts into the projects that he has been working on. He is using a vanilla JS library for this (https://github.com/dmauro/Keypress), and one day he asked himself, shouldn't there be a more ReactJS/Angular/<<Insert the framework that you know>> friendly way of doing this. He didn't want to reinvent the wheel, so he decided to release a library which provides a better framework-friendly-interface on top of the plain JS library above.

The first part of this problem is the design, where you figure out

1. **An API for the library —**
    - **Think of this like making a README for the library you are going to publish, write a simple how to use code snippet, like a lot of libraries out there. Eg., react-big-calendar (https://github.com/intljusticemission/react-big-calendar) , recharts (https://github.com/recharts/recharts) etc.**
2. **A technical approach for the implementation --**
    - **in any component-based framework of your choice**
    - **Talk about how you are going to create classes, how and what are they going to communicate with each other**

This library needs to support the following functionalities

1. **Allow any component to bind/unbind shortcuts**
2. **Any component can provide key combination & a callback to be triggered when the key is pressed -- (for eg - Ctrl + A changes the background color of componentA).**
3. **The library needs to keep track of the active shortcuts & should provide a way to retrieve the list of shortcuts which are currently active — this could be used to show a help page of sorts (the library only needs to provide the data -- the user can decide where to show this -- for example in a help screen)**

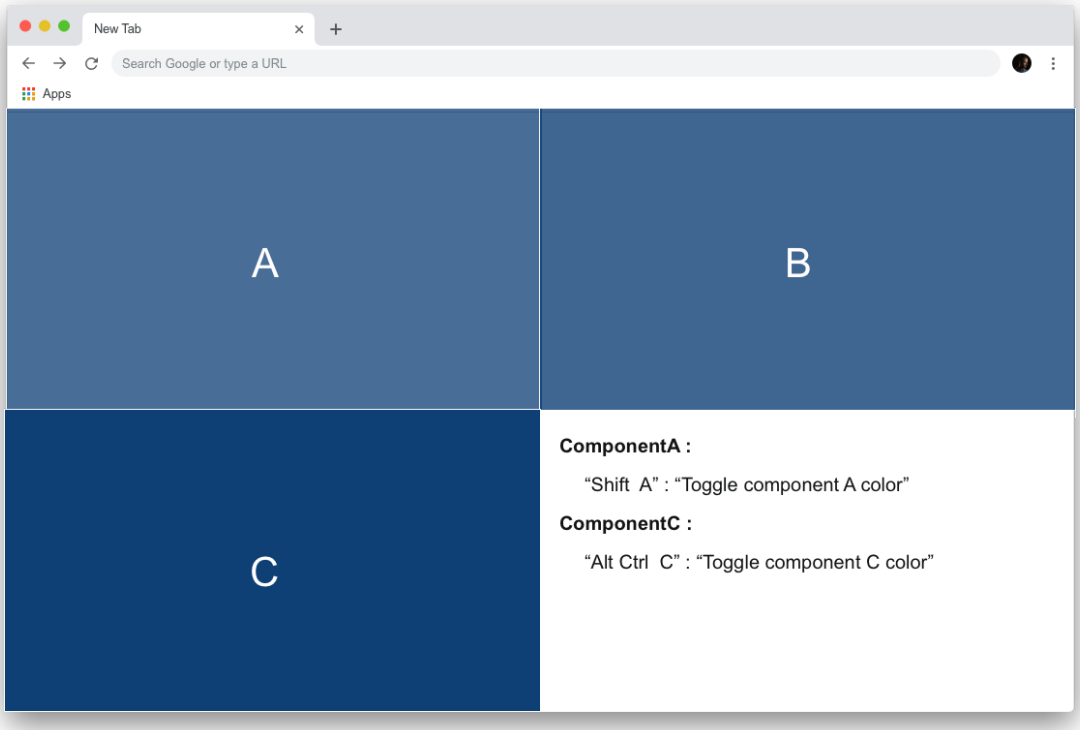
Note:

1. **To do the actual handling of keyboard, use functionality from Keypress.js (https://dmauro.github.io/Keypress/). Do not rewrite the logic to listen to keypress and emit the actions in your library. Just use Keypress for that. Please do not reinvent the wheel.**

2. **You are free to use any component-based framework like Angular / React etc. React would be preferable.**

Example Use case

Assume your library is now used to build a sample build a sample screen with multiple components (as shown below). Each component has its own keyboard shortcuts/callback



Eg: Shift A toggles component A color from blue to red . Assume A, B & C are components already present & using your library, a user is able to implement the above use case.

How to attempt:

1. **Research & design/documentation - 45-90 min**

    1. **Write down the API interface & usage doc for the library**
    2. **Discuss with the interviewer in the first 15 min showing him/her your API interface to make sure you are on the right path. Get a signoff before proceeding to the next step.**
    3. **Create a technical approach. Call out how various classes that you are going to create & how they are going to interact with each other .**
    4. **Again discuss with the interviewer & get a signoff on the approach.**
2. **Implementation - 1.5 hours**
    1. **Implement the given use case mentioned above using your library**

Approach Expected:

To build the wrapper around keypress we would need know figure out three things

1. **What parts of keypress do we need to utilize?**
2. **How to provide the interface for the user to give in the shortcut, callback and the description?**
3. **How to populate the list of active shortcuts?**

1. What parts of keypress do we need to utilize?

- **Create a single listener of keypress.js**
- **Use the "simple_combo" interface provided by keypress.**
- **Store the object returned during registration to deregister the shortcut at a later point in time along with the description.**
- **Use internal ids to de-register specific instances.**
- **NPM package link [https://www.npmjs.com/package/keypress.js]**

Note: Deregistering via key combination will create bugs when two different components are listening on the same shortcut as it will deregister all the callbacks on the provided key combination.

2. How to provide the interface for the user to give in the shortcut, callback and the description?

- **Create a new keyboard shortcut component that takes in a shortcut description, key combo, and callback function as props. Incase of multiple shortcuts per component repeat the use of keyboard shortcut components.**

```

**class ... extends React.PureComponent {**
```

```
turnGreen = () => {
 ...
 }

turnYellow = () => {
...
}

render () {
<KeyboardShortcut
   combo="shift g"
   callback={this.turnGreen}
   description="Turns the components background color to green"
/>
<KeyboardShortcut
   combo="shift y"
   callback={this.turnYellow}
   description="Turns the components background color to yellow"
  />
 }
}
```

Note: Creating the shortcut as a component will help the library handle bind and unbind operations on mount and unmount. Users of the library need not worry about this and makes it simpler for them to use.

- To get the list of active shortcuts create a higher order component similar to withRouter that gives the list of active shortcuts and their descriptions.

3. How to populate the list of active shortcuts?

- **To maintain a global state use React Context.**

Kindly work on this on your system/laptop and submit the link for the completed assignment back to us.

To submit the solution, please upload the entire folder on Google Drive / Dropbox share the folder as 'anybody with link can view' & then paste the link in the text box while submitting the assignment.

The assignment should be preferably done using ReactJS.