

Table of contents:

Prerequisite	2
Features of the Popup Component	2
How to use the Popup Component in your project	6
Attributes to be passed for Customisable feature	10
Code level detail- State management of the Popup Component	11
Reference	13
About the Author	13

Prerequisite

1.1 The Popup is only for projects built using React Library

Features of the Popup Component

2.1 Fully **customizable** Header text and message text

code detail in section 4.2 and 4.3

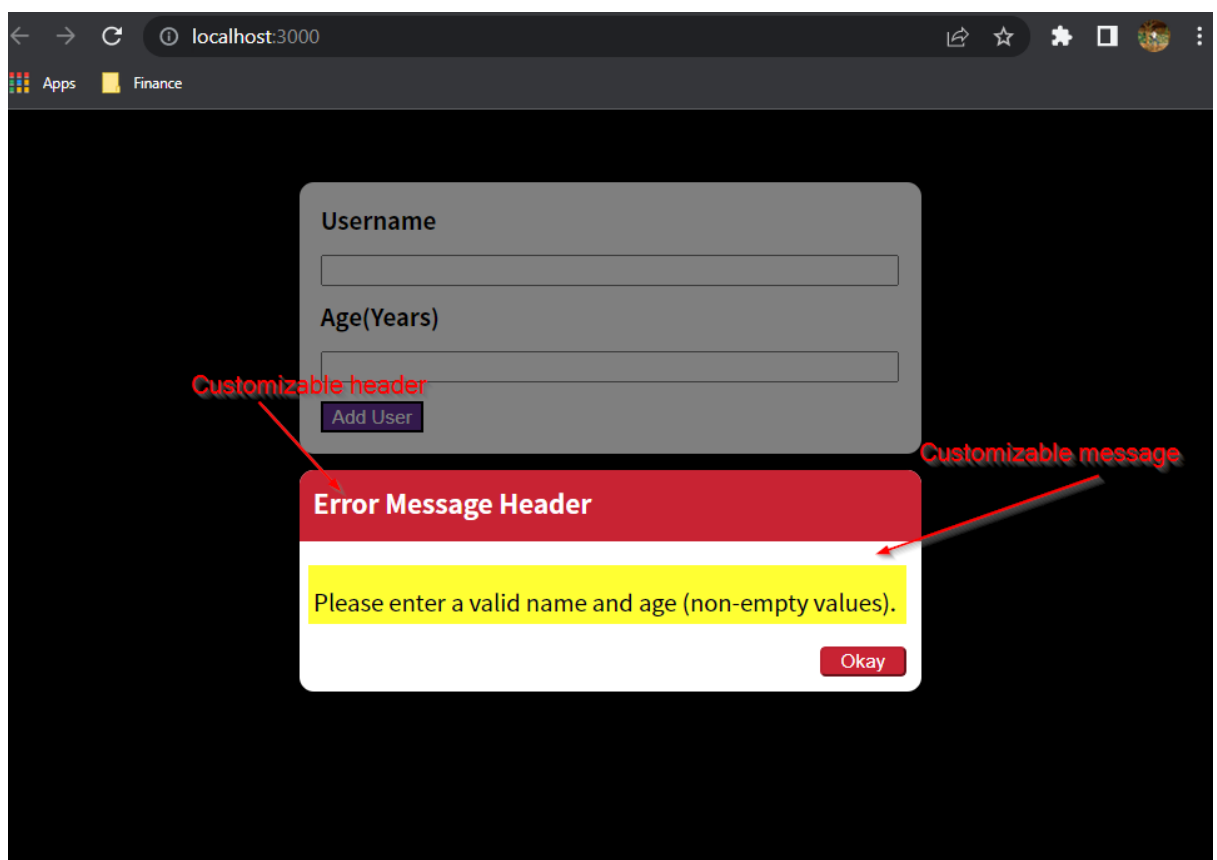


Image 1

2.2 The popup message can be **customised** for Success, Warning(/Info) and Error

code detail in section 4.1

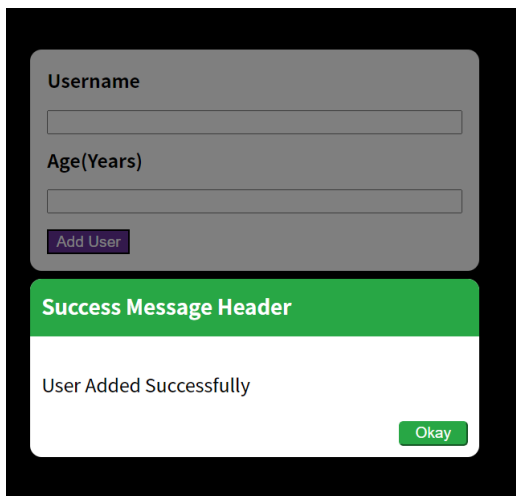


Image 2

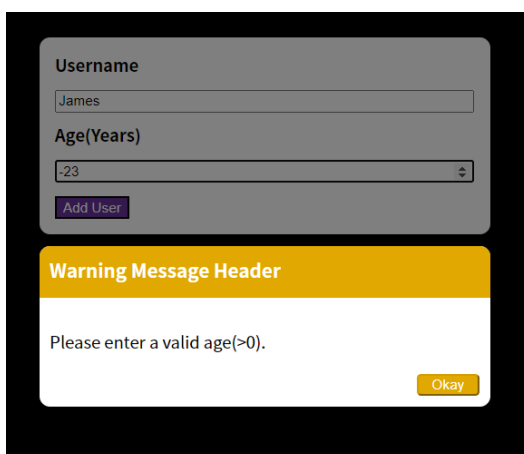


Image 3

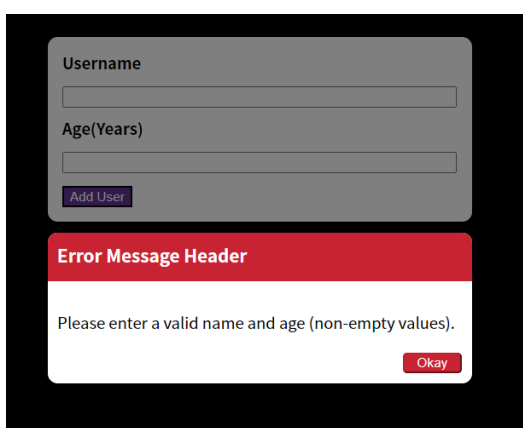
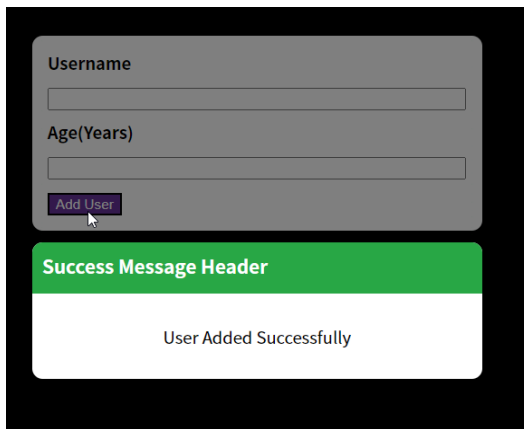


Image 4

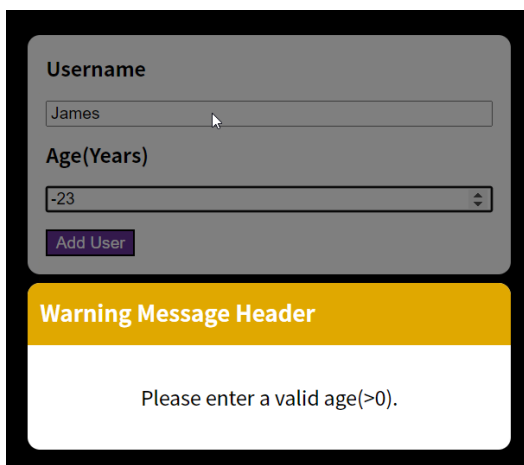
- 2.3** The popup messages can also be **customised** to auto-close. In that scenario the “Okay” Button will not be shown. *The “popup” will still close if clicked outside the popup window.* The auto-close time is **customisable**

code detail in section 4.6 and 4.7



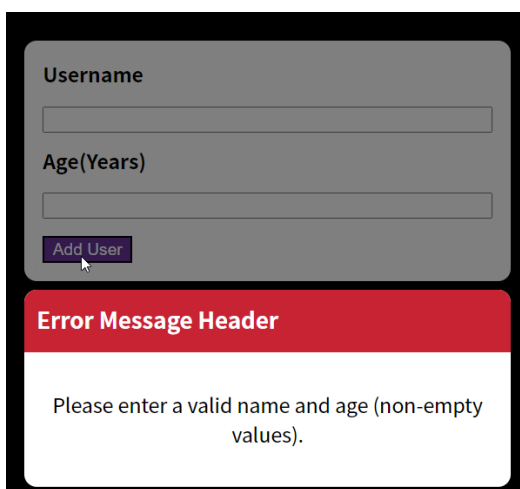
A user registration form with two input fields: "Username" and "Age(Years)". Below the "Age(Years)" field is a purple "Add User" button. Below the form is a green header for a "Success Message" with the text "User Added Successfully".

Image 5



A user registration form with two input fields: "Username" (containing "James") and "Age(Years)" (containing "-23"). Below the "Age(Years)" field is a purple "Add User" button. Below the form is a yellow header for a "Warning Message" with the text "Please enter a valid age(>0).".

Image 6



A user registration form with two empty input fields: "Username" and "Age(Years)". Below the "Age(Years)" field is a purple "Add User" button. Below the form is a red header for an "Error Message" with the text "Please enter a valid name and age (non-empty values).".

Image 7

2.4 The popup messages can be closed on clicking “Okay” or anywhere outside the popup window.

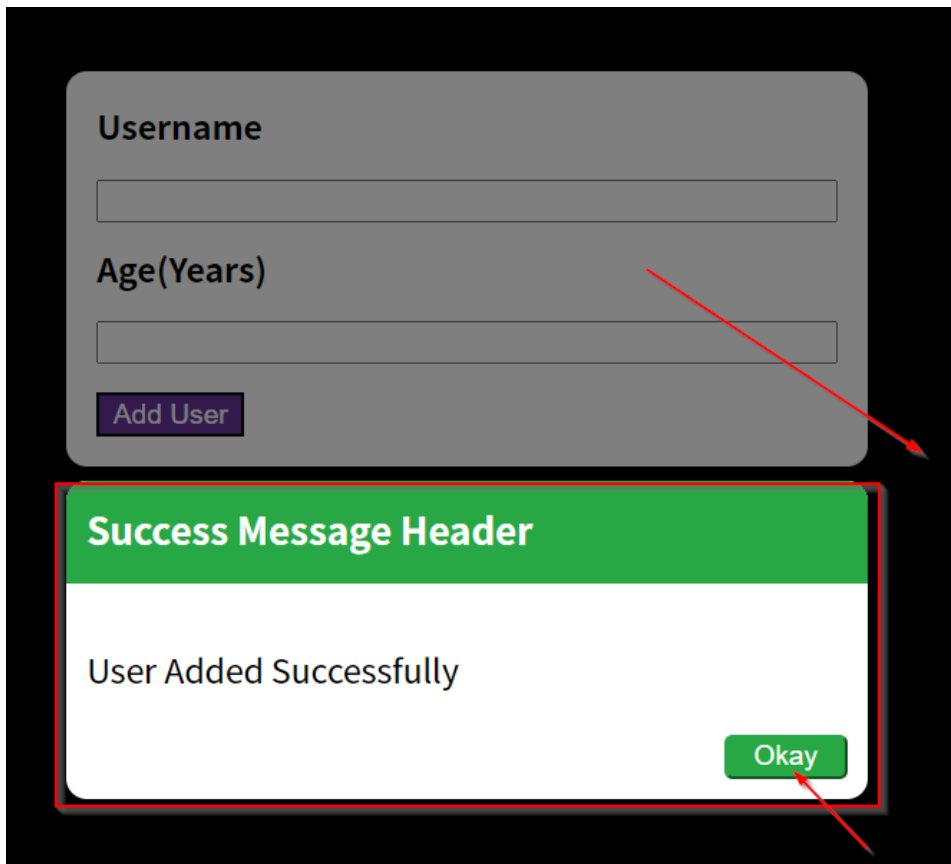


Image 8

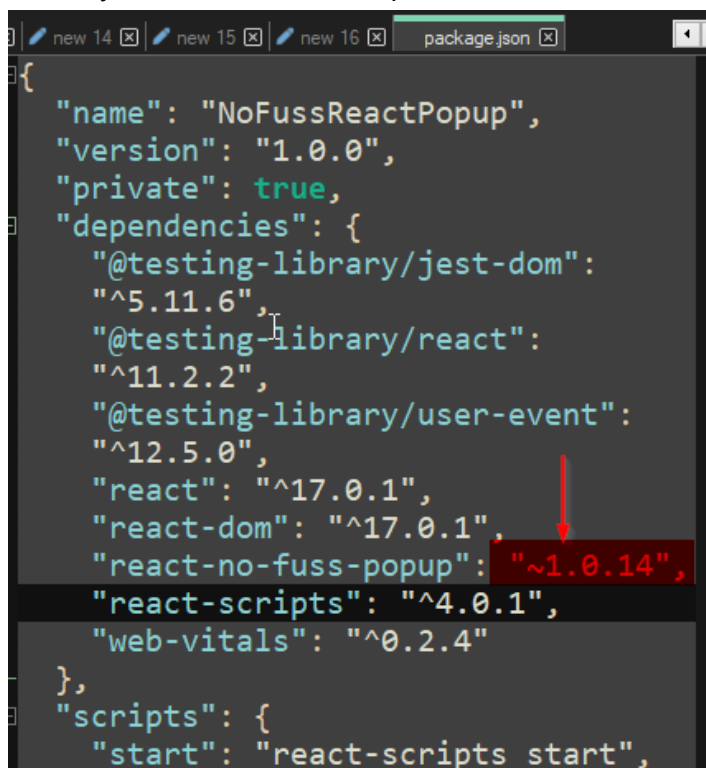
2.5 The popups are fully responsive for all screen sizes

2.6 The popup component does not destroy/create frequently, hence it's light on the DOM

How to use the Popup Component in your project

3.1 Added dependency in the Demo project:

Note: Always check for the latest patch version.



```
{
  "name": "NoFussReactPopup",
  "version": "1.0.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom":
      "^5.11.6",
    "@testing-library/react":
      "^11.2.2",
    "@testing-library/user-event":
      "^12.5.0",
    "react": "^17.0.1",
    "react-dom": "^17.0.1",
    "react-no-fuss-popup": "~1.0.14",
    "react-scripts": "^4.0.1",
    "web-vitals": "^0.2.4"
  },
  "scripts": {
    "start": "react-scripts start",
```

Image 9

3.2 Run npm install

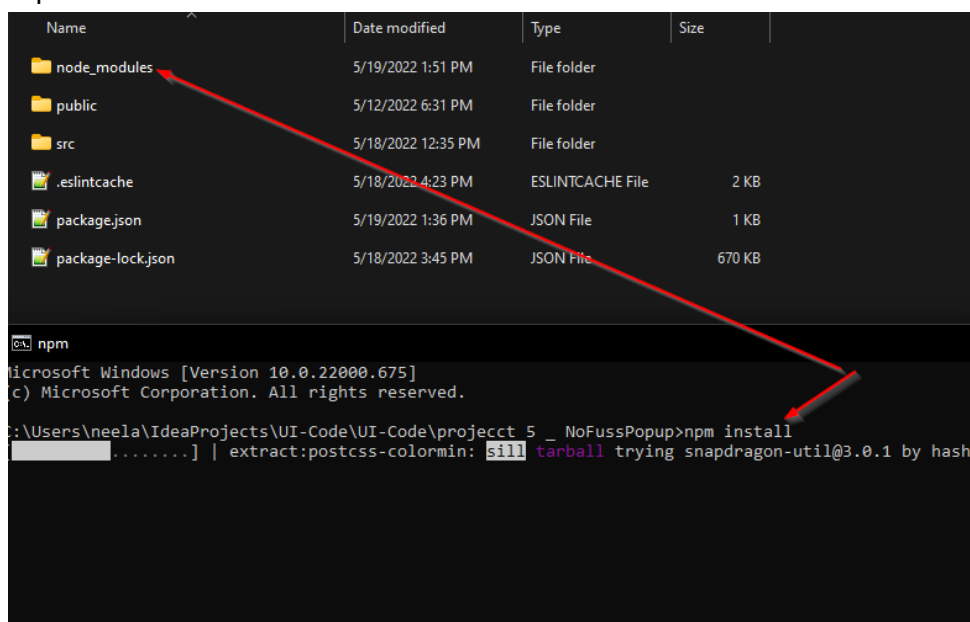


Image 10

3.3 In the project where we want to integrate the Pop-up, import the component

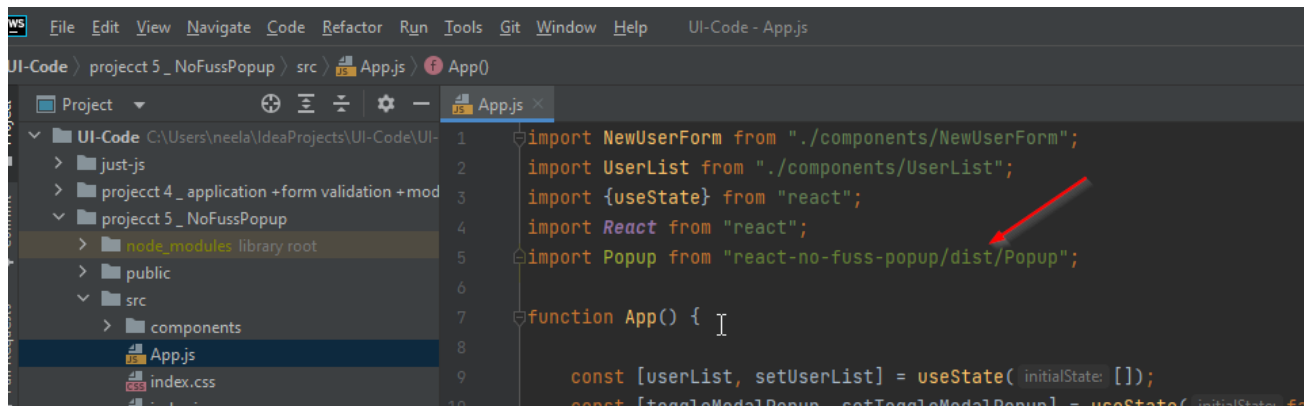


Image 11

3.4 Then add the JSX with all the required attributes

Detailed Code can be referred here:

https://github.com/neelabhraman/UI-Code/blob/main/project%205%20_%20NoFussPopup/src/App.js

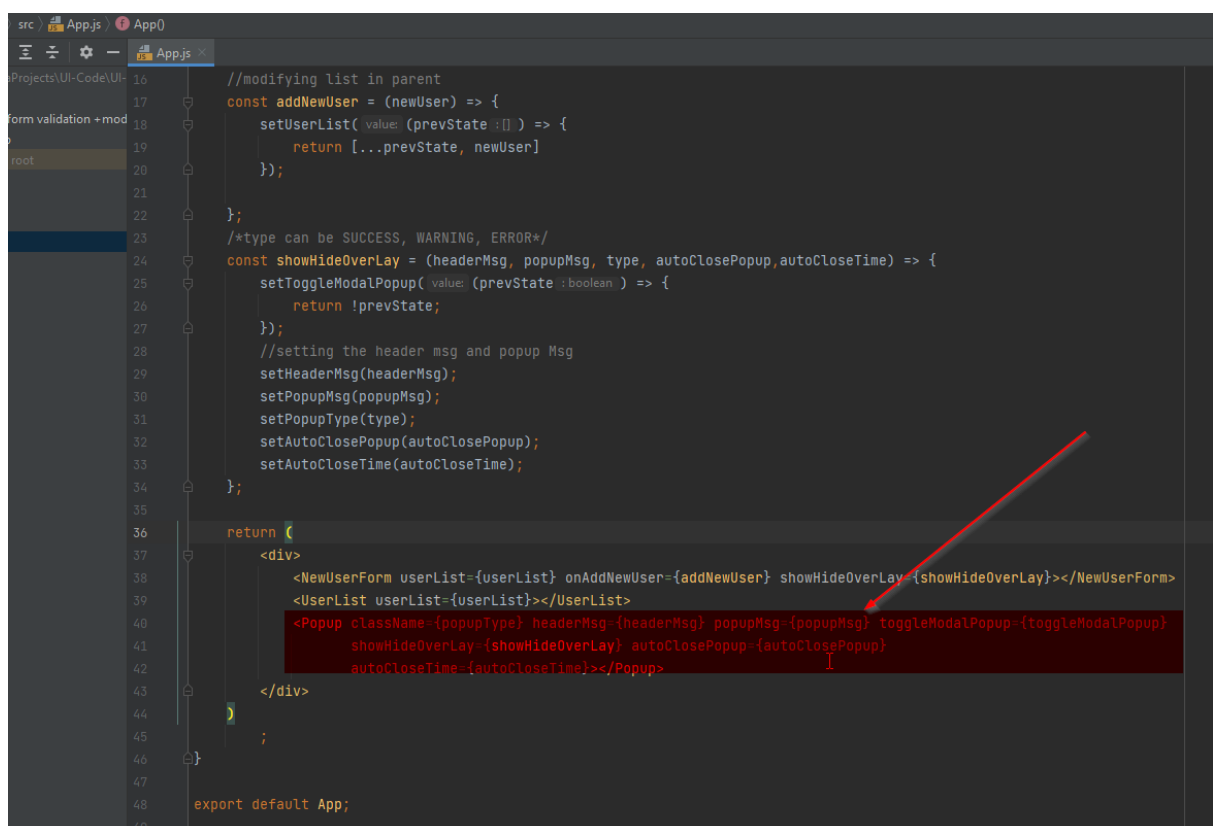


Image 12

3.5 On starting the local we should see the Popup Component loaded from console

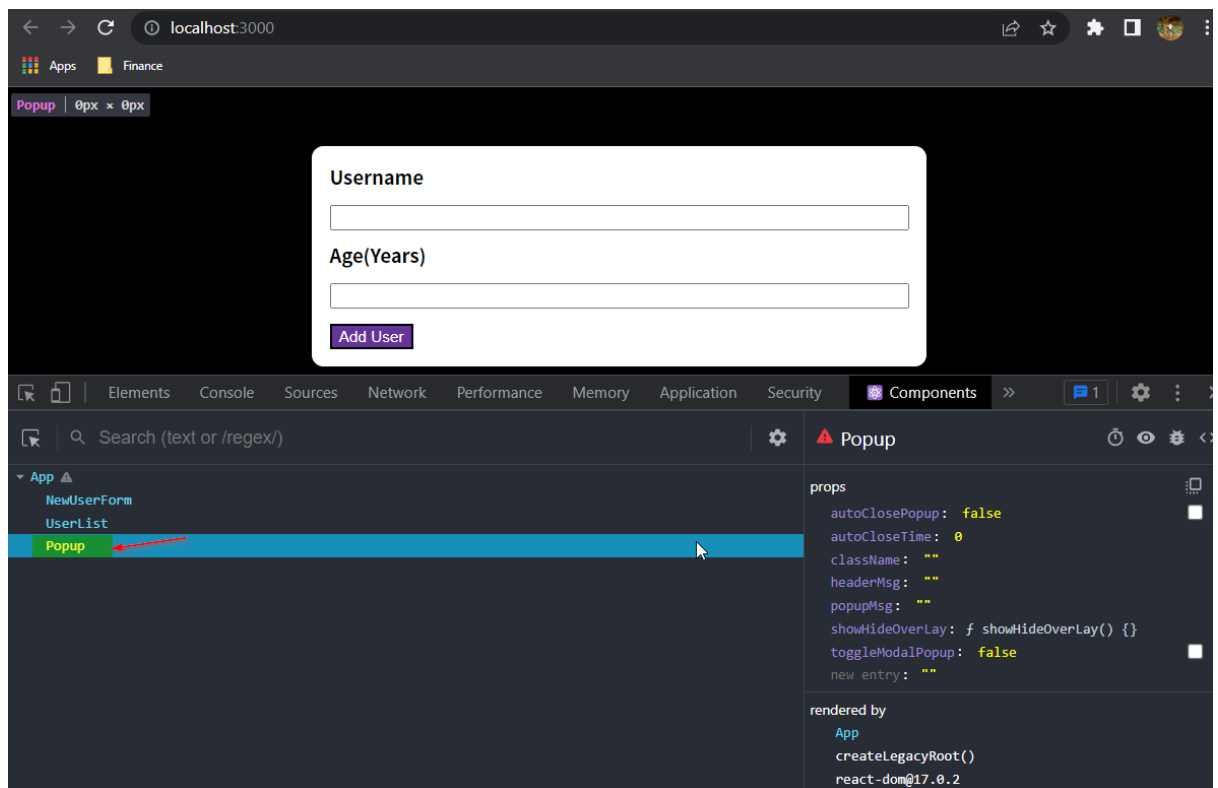


Image 13

Attributes to be passed for Customisable feature

The JSX :

```
<Popup
  className={popupType}
  headerMsg={headerMsg}
  popupMsg={popupMsg}
  toggleModalPopup={toggleModalPopup}
  showHideOverLay={showHideOverLay}
  autoClosePopup={autoClosePopup}
  autoCloseTime={autoCloseTime}>
</Popup>
```

4.1 popupType is a STRING, Valid Values are "success", "warning_info", "error"

Feature 2.2 can be achieved by above

4.2 headerMsg is a STRING. Valid Values are any string

Feature 2.1 can be achieved by above

4.3 popupMsg is a STRING. Valid Values are any string

Feature 2.1 can be achieved by above

4.4 toggleModalPopup is a BOOLEAN. Valid Values are true, false

4.5 showHideOverLay is a function. Valid method is:

```
//type can be SUCCESS, WARNING, ERROR/
const showHideOverLay = (headerMsg, popupMsg, type, autoClosePopup, autoCloseTime) => {
  setToggleModalPopup((prevState) => {
    return !prevState;
  });
  //setting the header msg and popup Msg
  setHeaderMsg(headerMsg);
  setPopupMsg(popupMsg);
  setPopupType(type);
  setAutoClosePopup(autoClosePopup);
  setAutoCloseTime(autoCloseTime);
};
```

Detailed Code can be referred here:

<https://github.com/neelabhraman/UI-Code/blob/main/project%20-%20NoFussPopup/src/App.js>

4.6 autoClosePopup is a BOOLEAN. Valid Values are true, false

Feature 2.3 can be achieved by above

4.7 autoCloseTime is a NUMBER. Valid Values positive integers

Feature 2.3 can be achieved by above

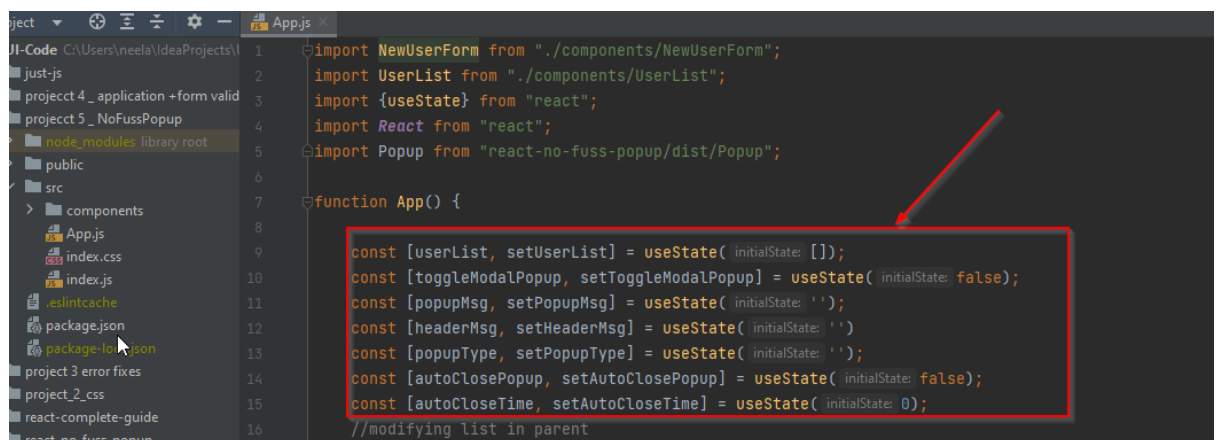
Code level detail- State management of the Popup Component

5.1 The Popup component is designed to be included in the top-most JSX (preferably the App.js)

5.2 The state management happens from the same top-most component(preferably the App.js)

Detailed Code can be referred here:

<https://github.com/neelabhraman/UI-Code/blob/main/project%20-%20NoFussPopup/src/App.js>



```

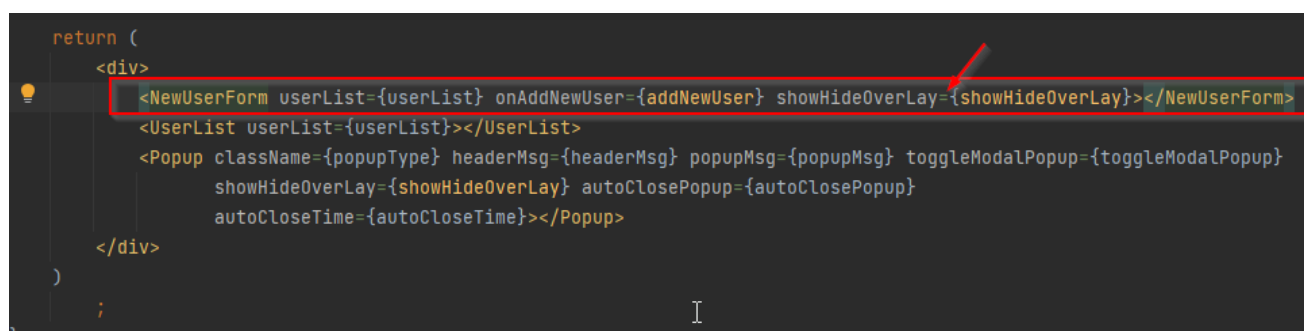
1  import NewUserForm from "../components/NewUserForm";
2  import UserList from "../components/UserList";
3  import {useState} from "react";
4  import React from "react";
5  import Popup from "react-no-fuss-popup/dist/Popup";
6
7  function App() {
8
9      const [userList, setUserList] = useState( initialState: []);
10     const [toggleModalPopup, setToggleModalPopup] = useState( initialState: false);
11     const [popupMsg, setPopupMsg] = useState( initialState: '');
12     const [headerMsg, setHeaderMsg] = useState( initialState: '');
13     const [popupType, setPopupType] = useState( initialState: '');
14     const [autoClosePopup, setAutoClosePopup] = useState( initialState: false);
15     const [autoCloseTime, setAutoCloseTime] = useState( initialState: 0);
16     //modifying list in parent
  
```

Image 14

5.3 The showHideOverLay function(of section 4.5) has to be passed in your desired component from where we want to invoke the popup.

Detailed Code can be referred here:

<https://github.com/neelabhraman/UI-Code/blob/main/project%20-%20NoFussPopup/src/App.js>



```

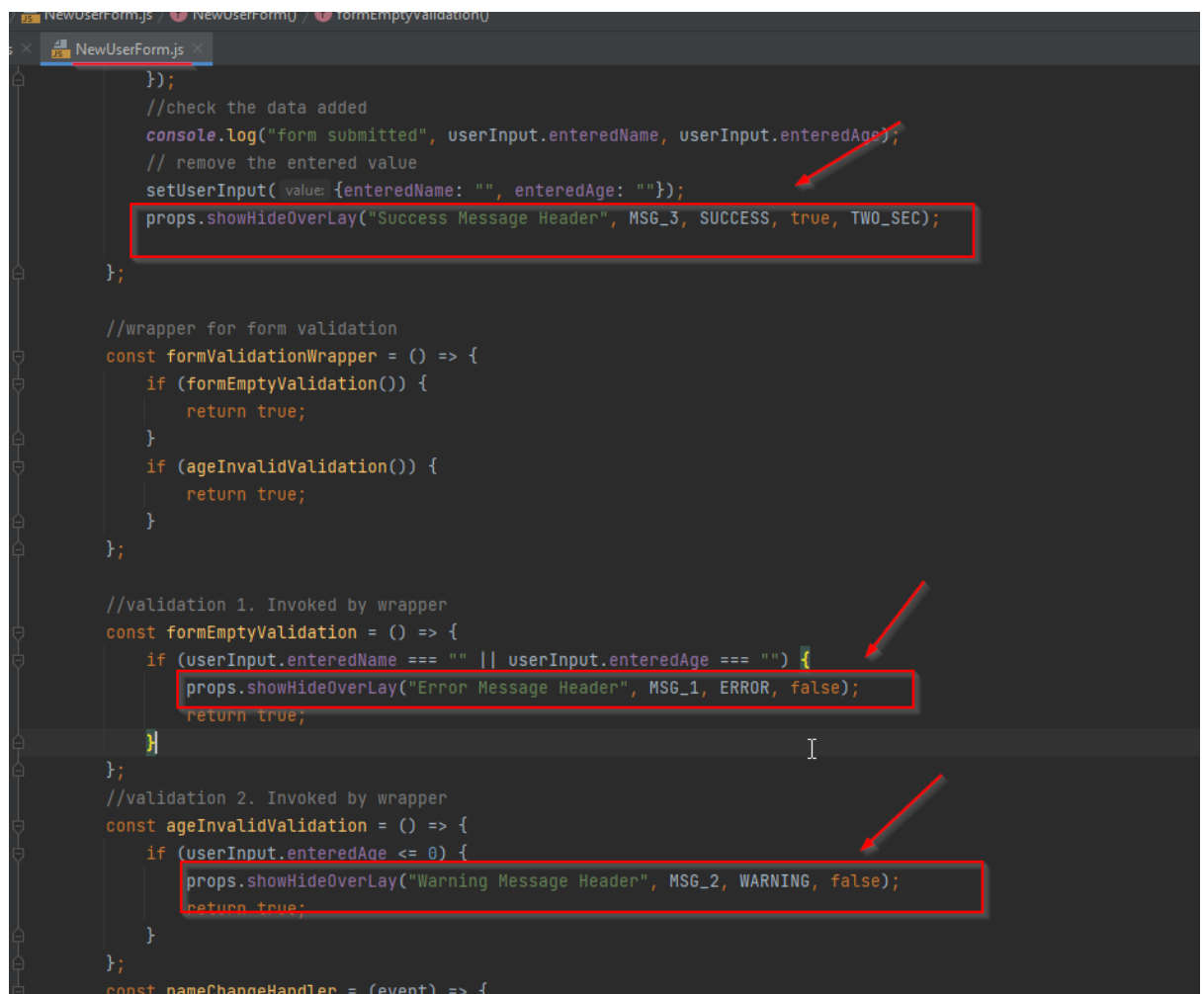
return (
  <div>
    <NewUserForm userList={userList} onAddNewUser={addNewUser} showHideOverLay={showHideOverLay}></NewUserForm>
    <UserList userList={userList}></UserList>
    <Popup className={popupType} headerMsg={headerMsg} popupMsg={popupMsg} toggleModalPopup={toggleModalPopup}
      showHideOverLay={showHideOverLay} autoClosePopup={autoClosePopup}
      autoCloseTime={autoCloseTime}></Popup>
  </div>
)
;
  
```

Image 15

5.4 From the desired component invoke the showHideOverLay function with desired parameters.

Detailed Code can be referred here:

<https://github.com/neelabhran/UI-Code/blob/main/project%20-%20NoFussPopup/src/components/NewUserForm.js>



```
};  
//check the data added  
console.log("form submitted", userInput.enteredName, userInput.enteredAge);  
// remove the entered value  
setUserInput( value: {enteredName: "", enteredAge: ""});  
props.showHideOverLay("Success Message Header", MSG_3, SUCCESS, true, TWO_SEC);  
};  
  
//wrapper for form validation  
const formValidationWrapper = () => {  
  if (formEmptyValidation()) {  
    return true;  
  }  
  if (ageInvalidValidation()) {  
    return true;  
  }  
};  
  
//validation 1. Invoked by wrapper  
const formEmptyValidation = () => {  
  if (userInput.enteredName === "" || userInput.enteredAge === "") {  
    props.showHideOverLay("Error Message Header", MSG_1, ERROR, false);  
    return true;  
  }  
};  
  
//validation 2. Invoked by wrapper  
const ageInvalidValidation = () => {  
  if (userInput.enteredAge <= 0) {  
    props.showHideOverLay("Warning Message Header", MSG_2, WARNING, false);  
    return true;  
  }  
};  
  
const nameChangeHandler = (event) => {
```

Image 16

Reference

<https://reactjs.org/docs/create-a-new-react-app.html>

<https://levelup.gitconnected.com/how-to-publish-your-react-component-package-to-npm-20ee4f81a506>

About the Author

Neelabh is a Problem Solver by Attitude and a Full Stack Developer by Profession

Check more about him here:

<https://in.linkedin.com/in/neelabh-raman-a8713284>

GitHub Link here:

<https://github.com/neelabhraman>

Leet code link here:

<https://leetcode.com/NeelabhRaman/>