

## CRUD React App

```
<App>
  <UserCrudApp>
    <UserForm
      userInfo={this.state.formData}
      onChange={this.handleFormChange}
      saveUser={this.saveUser}
      resetForm={this.resetFormData}
    />
    <UserList
      users={this.state.users}
      onEdit={this.editUser}
      deleteUser={this.deleteUser}
    />
  </UserCrudApp>
</App>
```

### UserCrudApp.js

- UserCrudApp component should render UserForm and UserList components as shown above and pass the props to both these components.
- UserCrudApp component should hold the state like below.
- Initial State object should look like:

```
this.state = {
  nextUserId: 1,
  users: [],
  formData: {
    id: null,
    firstName: '',
    lastName: '',
    email: '',
    phone: '',
    address: ''
  }
};
```

- UserCrudApp component should contain the following functions:
  - handleFormChange(updatedFormData)  
this function should be passed to UserForm and called when user updates any field like firstName, lastName etc. updatedFormData object should be updated and UserCrudApp state should update formData like below.  

```
handleFormChange(updatedFormData) {
  this.setState({ formData: updatedFormData });
}
```
  - saveUser(userInfo)

- `saveUser(userInfo)` should accept a `userInfo` object as argument with the following properties.
  - `id` (optional)
    - If `id` exist in `userInfo`, then update the existing user, else add a new user with `nextUserId` in `this.state.users` array.
  - `firstName`
  - `lastName`
  - `email`
  - `phone`
  - `address`

After saving user, update the `nextUserId` to + 1;

- `editUser(userId)` function should find the `userInfo` from `this.state.users` and set the `userInfo` to `this.state.formData`
- `deleteUser(userId)` Delete the user from `this.state.users` with particular `userId`
- `resetFormData()` should reset the `this.state.formData`

## UserForm.js

- Should be responsible for rendering add/edit user form
- Should accept the following props
  - `userInfo: PropTypes.shape({`  
     `firstName: PropTypes.string,`  
     `lastName: PropTypes.string`  
     `...`  
     `})`
  - `onChange: PropTypes.func.isRequired`
  - `saveUser: PropTypes.func.isRequired`
  - `resetForm: PropTypes.func.isRequired`
- UserForm should contain following functions:
  - `updateFormData(fieldName, fieldValue) {`  
     `const updatedData = Object.assign({},`  
     `this.props.userInfo, {[fieldName]: fieldValue});`  
     `this.props.onChange(updatedData);`  
   `}`
  - Respective fields should implement their `onChange` handlers and call the above `updateFormData()` function so that data is updated in parent component's (`UserCrudApp`) state.
    - Eg. `<input type="text" name="firstName" onChange={e => this.updateFormData("firstName", e.target.value)} />`
  - `handleCancelClick()` function should be registered on click of Cancel button and call the `this.props.resetForm()` which will clear the form data.

- `handleSaveClick()` function should be registered on click of Save/Update button. Required validations should be applied in this function or another function. If data is valid, `this.props.save`
- UserForm should display Save/Update and Cancel buttons
  - Button text should be displayed as Save/Update, if `userInfo.id` exist and `userInfo.id > 0`
  - On clicking Save, data should be validated (simple required validation) and `this.props.saveUser()` function should be called.
  - On clicking Cancel, `this.props.onCancel()` should be called.

## UserList.js

- UserList component should be responsible for rendering users listing and filtering
- UserList component should display a search textbox on top right to filter users based on `firstName` and `lastName`.
- As soon as, we enter text in filter input box, user listing should be filtered. It should work like search functionality.
- It should also display a clear button with input box, to clear filter text
- Initial state should look like:

```
this.state = {
  filterText: ""
}
```

- It should accept the following props:
  - `users: PropTypes.array`
  - `onEdit: PropTypes.func,`
  - `deleteUser: PropTypes.func`
- UserList component should contain this following functions:
  - `handleFilterInputChange()` should update the state value
  - `clearFilterInputText()` should clear/remove the `filterText` from state and show all users
  - `getUsers()` {
 

```
return this.props.users.filter((user)=> {
  //if user.firstName or lastName does not
  contains this.state.filterText
  // filter this item
});
```

It should implement the filtering logic on `this.props.users` using `this.state.filterText` and returns the filtered users. It will be called from `render()` to render filtered users.

```
deleteUser() {
  if(confirm("Do you really want to delete {firstName
+ lastName} from list ?")) {
    this.props.deleteUser(userId)
  }
}
```

It should confirm before deleting user from list. It should display a confirm message like “Do you really want to delete {firstName + lastName} from list ?” If user confirms, then it should call parent deleteUser() like this.props.deleteUser(userId), otherwise nothing to do.

- render() should call getUsers() to get the filtered users and then render those filtered users.

```
render() {  
  return (  
    this.getUsers().map((user) => {  
  
      })  
  )  
}
```

- UserList should also display Edit and Delete icons and work as explained above.