execution flow

add button -> setAddSection(true) after

<Formtable/> will execute

onchange -> if user type something that will stor in setFormData

const[ editsection,setEditSection] = useState(false) aa line edit karva mate nu box open karse

handleEditOnChange -> if user type something that will stor in setFormDataEdit

const handleOnSubmit = (e) => {

// Prevent the default form submission behavior

e.preventDefault();

// Access form data or perform any necessary actions

// For example, you can use e.target to access form elements

// Example: Log form data

console.log('Form submitted!');

// Additional logic or API calls can be added here

// Reset the form if needed

// e.target.reset();

};

e is the event object passed to the function.

e.preventDefault() prevents the default form submission behavior, allowing you to handle the submission manually.

You can access form data or form elements using e.target or other methods depending on your form structure.

The function can include any custom logic or API calls related to form submission.

addSection && (

// JSX content for rendering when addSection is truthy

)

This is a common pattern where the content inside the parentheses is only rendered if the condition (**addSection**) is true. The **&&** (logical AND) operator is used for conditional rendering in this case. If **addSection** is truthy, the content inside the parentheses will be rendered.

Here's a breakdown of how it works:

* If **addSection** is falsy (e.g., **false**, **null**, **undefined**, **0**, or an empty string), nothing will be rendered.
* If **addSection** is truthy, the content inside the parentheses will be rendered.

[name] : value (in handleOnChange)

In form name= “name”, “email”, “mobile” in value will update from the form according

**State:**

1. **addSection:**
   * A boolean state variable that controls the visibility of the form for adding a new record.
2. **editsection:**
   * A boolean state variable that controls the visibility of the form for editing an existing record.
3. **formData:**
   * State variable holding the data for adding a new record (name, email, mobile).
4. **formDataEdit:**
   * State variable holding the data for editing an existing record (name, email, mobile, \_id).
5. **dataList:**
   * State variable containing the list of records fetched from the backend.