

WorkShop 2

Project setup:

1. Create a new react app.
2. Open the app to your IDE.
3. Make sure you remove the default codes generated by React.
4. Create index.js file in the **src** folder.
5. Render a greeting message inside the index.js file.
6. Make sure the react application runs without errors.
7. Install the **bootstrap** using `npm install bootstrap`
8. Install the **hook form** using `npm install react-hook-form`
9. Install the **axios** using `npm install axios`
10. Install the **router** using `npm install react-router-dom@5.3.0`
11. Commit & Push.

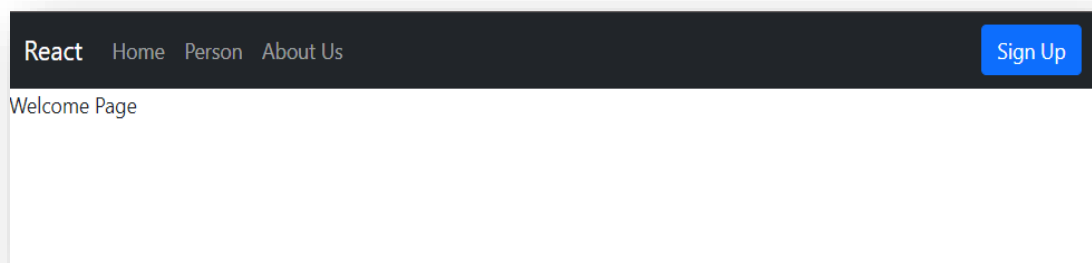
Defining components:

Part 1

1. Create a **DemoRouter.js** using arrow function component.
2. Define **Welcome**, **About**, **Home**, **Person** and **NotFound** components that return or render the simple information.
3. Define a **Header** component that uses `<nav>` HTML element and `<Link>` component for navigation between components.

path	component
/	Welcome
/home	Home
/person	Person
/about	About
	NotFound

4. Inside the **DemoRouter** component, define routing between components using the `react-router-dom` components like `<Router>`, `<Switch>` and `<Route>`.
5. Commit & Push.

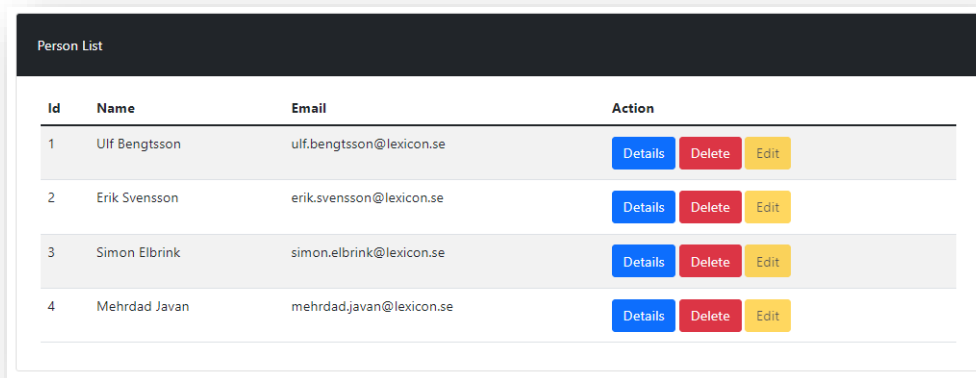


Part 2

1. Create a **CrudDemo.js** file using arrow function component.
2. Add a new **Route** into the **DemoRouter** component.

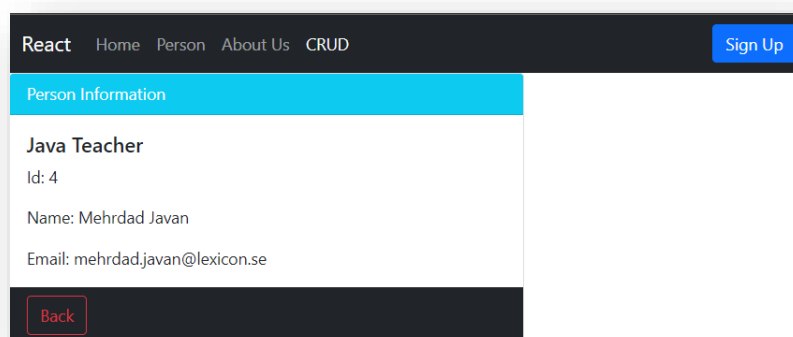
path	component
/crud	CrudDemo

3. Add a new **Link** to **"/crud"** inside the navigation.
4. Define a **const** variable **persons** (it is an array of persons) into the **CrudDemo** component using **useState** Hook function.
5. Implement an **useEffect** Hook function that invoke an **API** using **Get** method and set the fetched data to the **persons**.
6. Define a **Table** component that returns person list information.
 - ❖ Table component should breakdown to **TableHeader**, **TableAction**, and **TableRow** components.
7. Commit & Push when all components are done.



Id	Name	Email	Action
1	Ulf Bengtsson	ulf.bengtsson@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>
2	Erik Svensson	erik.svensson@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>
3	Simon Elbrink	simon.elbrink@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>
4	Mehrdad Javan	mehrdad.javan@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>

8. Define an **onClick** event for details button that takes **id** from **props** and then **route** it to the **PersonDetails** component with id. (Note: Add a new Route for PersonDetails with id param)
 - 1) Define a **PersonDetails** component.
 - 2) Define a **const** **person** variable into the **PersonDetails** component using **useState**.
 - 3) Define a **useEffect** Hook function.
 - 4) Get the **id** **parameter** using the **useParams** Hook function.
 - 5) Call the **API Get by Id** and then set the fetched data to the **person** variable.



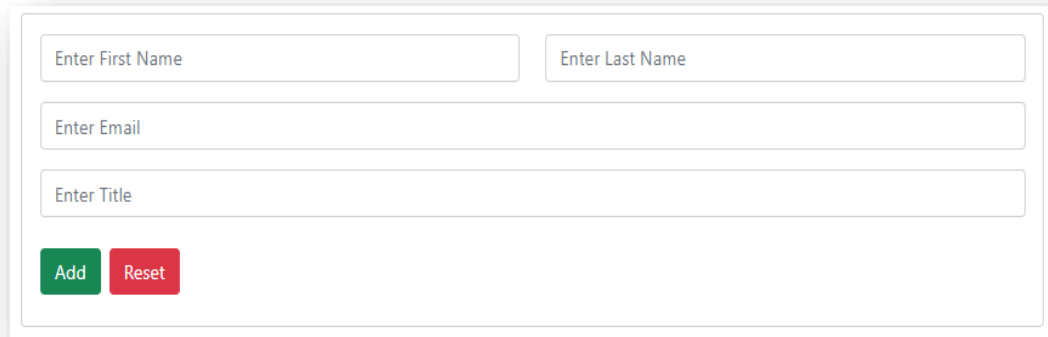
React	Home	Person	About Us	CRUD	Sign Up
Person Information					
Java Teacher					
Id: 4					
Name: Mehrdad Javan					
Email: mehrdad.javan@lexicon.se					
<button>Back</button>					

9. Define an **onClick** event for delete button that calls **Delete API By Id**.
 - ❖ When the selected person's is removed by API, then the person list should be updated if the person was removed.
10. Commit & Push.

Part 3

1. Define a **Form** component using **useForm** Hook function with validation that returns a HTML form like this:

❖ Person object should have properties like: id, firstName, lastName, email & title.



Enter First Name

Enter Last Name

Enter Email

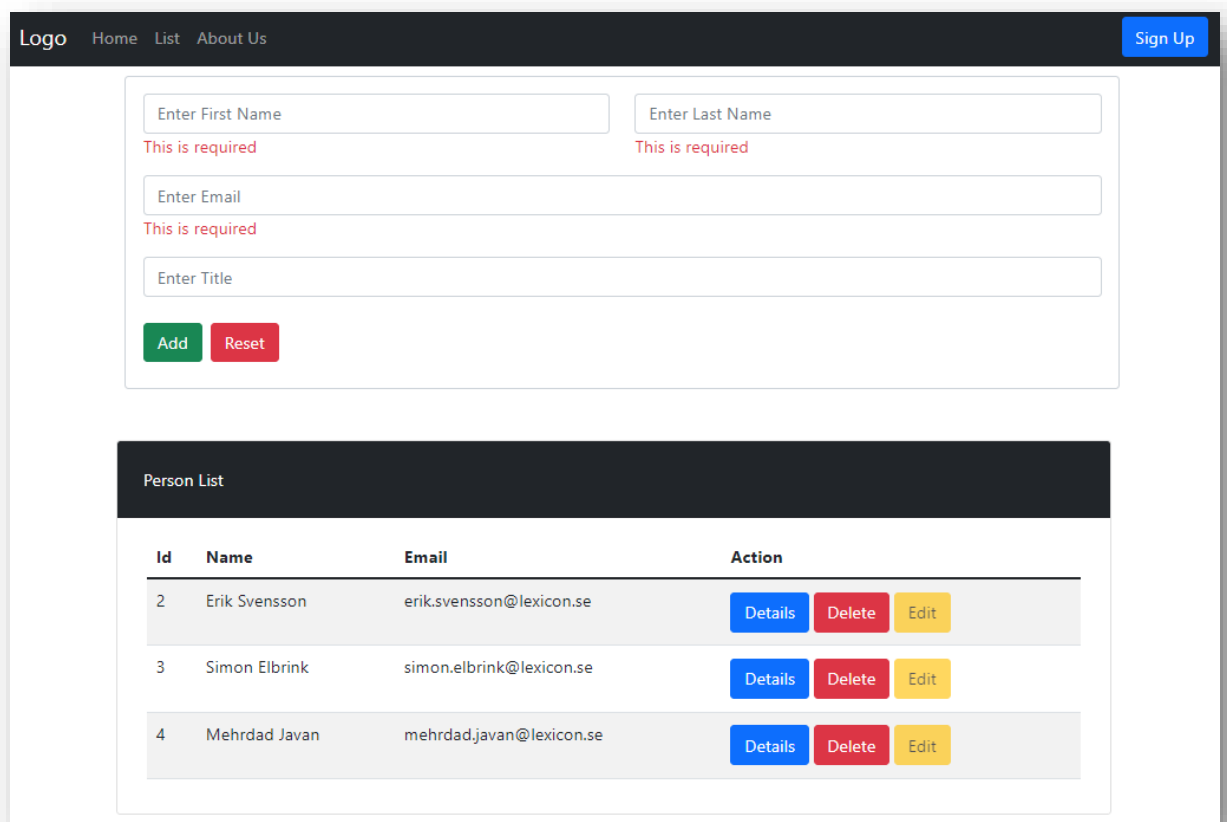
Enter Title

Add Reset

2. Create a function in the **Form** component like **savePerson** that is used to **post** the **data** object to the **API**.

❖ When a new person is added, the person list should be updated.

3. Commit & Push.



Logo Home List About Us Sign Up

Enter First Name Enter Last Name

This is required This is required

Enter Email

This is required

Enter Title

Add Reset

Person List

Id	Name	Email	Action
2	Erik Svensson	erik.svensson@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>
3	Simon Elbrink	simon.elbrink@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>
4	Mehrdad Javan	mehrdad.javan@lexicon.se	<button>Details</button> <button>Delete</button> <button>Edit</button>