Fascoon - Doubt Solving Session - 24th October 2021 - May Batch

Note: We will start the session @12

Attendance for the Session:

- Andy
- Prathamesh
- Suresh Yadav
- Vinay
- Amit
- Poonam
- Rushi
- Mohsin

Module Completion

---

1. HTML Basics - **DONE**
2. CSS Basics - **DONE**
3. Bootstrap CSS Framework - **IN PROGRESS**
4. Javascript Basics - **DONE**
5. Javascript Advanced - **DONE**
6. Github - **DONE**

Doubts OR Queries

Q.1.[Poonam] Constructor this keyword?

- If you want to implement this keyword, always do it using javascript classes.
- Implementing this keyword in normal objects or normal variables is not recommended.

```
// class Person
let country = 'India'
const name = 'Amit'
class Person{
```

```
    constructor(firstName,lastName){
        console.log('I am a constructor')
        this.firstName = 'Atul'
        this.lastName =  'Sharma'
    }
    displayInfo(){
        console.log(this.firstName)
        console.log(this.lastName)
    }

}
// object or instance of class person
p1 = new Person()
p1.displayInfo()
console.log(country)
```

Q.2.[Amit] Mini Revision of Topics Completed?

- Let's create a small web app, which will display news dynamically.

Q.3.[Naresh] Difference between let and const?

- The value of a const doesn't change.
- The value of a let variable can change.

Q.4.[Suresh] Difference between Re-declaration and Re-initialization??

- **var** keyword - can be re-initialized, or value can be updated from inside and outside the scope.
- **let** keyword - can be re-initialized, or value can be updated only from inside the block scope.
- **const** keyword - once initialized it cannot be re-initialized.

Q.5.[Poonam] Terminal Operations??

- **Save the list of commands in a notepad**

- **Creating a Folder**

```
$ mkdir <folder-name>
```

- **Go inside the Folder**

```
$ cd <folder-name>
```

- **Go Back**

```
$ cd ..
```

- **Clear Terminal**

```
$ clear
```

Q.6.[Vinay] What is the use of Return in a Function?

Ans:

1. return keyword is used to return back to the main program the calculated value or output.
2. return value is supposed to saved in a new variable when calling the function.

```
   calcNewSalary(){

        const newSalary = this.salary + this.increment
        return newSalary
    }
const newSalary = emp1.calcNewSalary()
```

```
    console.log('New Salary after Hike:', newSalary)
```

Q.7[Poonam] Two functions were used in to fetch news and display the newsdata?? How and why these functions were used??

Ans:

---

1. Use of nested function calling.
2. getNewsData() function is just fetching the news, and storing the data in a variable ➜ news_data.
3. fetchNews() function is calling the getnewsData() function, which is called a nested function call i.e ➜ calling a function from inside a function.
4. Once you call fetchNews() function, it will call getNewsData() function
5. getNewsData() will update the value of variable ➜ news_data.
6. This news_data variable will be then given to the HTML View.

Notes:

---

1. **var** keyword is accessible inside and outside the block scope.
2. **let** keyword is accessible only inside the block scope.
3. Anything **let** and **const** declared inside the block scope cannot be accessed from outside the block scope.

Practice with Dynamic Data using the following Free APIS

- https://randomuser.me/api/
- https://newsapi.org

- Javascript Advanced - **IN PROGRESS**

- Advanced Javascript Topics:

  - Classes
  - Objects ➜ JSON Object , Javascript Object , JSON Array, **Object Array**
  - Arrays
  - Functions

- Local Storage
- Session Storage

Questions:

Q.1[Poonam] What is fetch API , how to get the Data? The Process of API??

Ans:

- What is an API???
  - API stands for application programming interface.

- When computers were introduced as a phone, tablet, people started visiting softwares on their smartphones.

- These smartphones did not have access to the same features as the computer back then.

- API creates a common platform, for data to be accessed on all type of computing devices like:
    - Laptop
    - Smartphone
    - Smart TV
    - Radio
    - Washing Machine which runs on voice assistant
    - Smart Devices like Smart Speaker etc etc etc

- Two popular types of API's are:
    - XML
    - JSON

- What does an API do??

- An API creates and endpoint to access the data in your database.
- Previously, databases were different for applications and websites.
- Now we use the same database for your website and mobile application as well.
- However, the database if written for Web, cannot be accessed on the mobile.
- Which is why we use API endpoints.
- APIs need to be designed.

- Popular technologies to design APIs:
    - Node.js
    - Laravel
    - Express
- Hence, API is just a technology which gives an access point to access the data of a database.
- Popular free APIs for learning software development:
    - News API
    - Random User API
    - Stock Market API etc etc etc

- **Javascript Mini App:** Display Random user data using Random User API

Q.2[Vinay][General] Not Understanding Javascript works???

Ans:

---

1. Javascript is just a tool, a programming language.
2. There are two programming languages:
    - Application Programming languages.
        - Examples:
            - Flutter
            - Javascript
            - PHP
            - Python
    - System Programming languages.
        - Examples:
            - Python
            - C++
            - Rust

---

3. Programming languages are used to create your end product.
    - End Product examples:
        - E-commerce website
        - Blog
        - Social Media website

To create a simple recipe book using javascript.

- HTML ➜ Structure of the Application
- CSS ➜ Design of the Application
- Javascript ➜ Behavior of the Application
- Database ➜ NOSQL, MYSQL, Firebase etc etc
  - To store your recipes
  - Storage can be done using tables or collections

Module Completion

---

1. HTML Basics - **DONE**
2. CSS Basics - **DONE**
3. Bootstrap CSS Framework - **IN PROGRESS**
4. Javascript Basics - **DONE**
5. Javascript Advanced - **DONE**
6. Github - **DONE**
7. React.js - Basics - **DONE**

React Topics Done:

- React Structure
- React Installation
- React Components??
  - Stateful Components
  - Stateless Components

- **React Installation**

- To install a react app, use the following command:

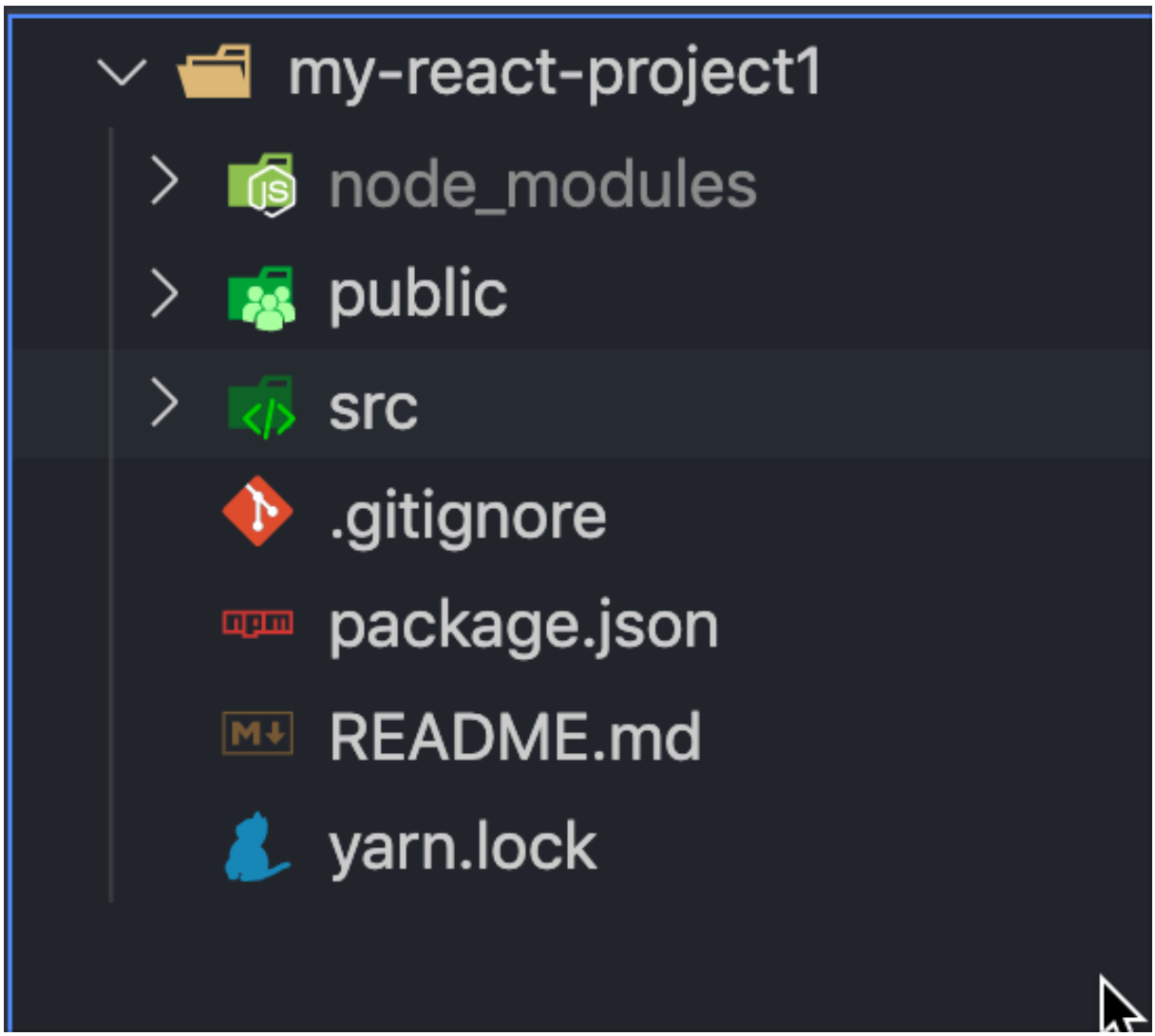
```
$ npx create-react-app <name-of-your-app>
$ npx create-react-app myReactProject1
```

- The above command will install the required files to run a react app.

- **What is Yarn?**

- Yarn is alternative to npm.
- It was developed by facebook.
- It is not recommended to use both package managers at the same time.
- Either you use yarn or npm.
- It is recommended to use npm over yarn.

- **React Application Structure:**



- The main parts of your React App Structure are:
  - **node_modules folder:**
    - contains the npm packages that is used to build your project.
  - **public folder:**
    - contains the index file of your project which is used by the server to process your app.
    - contains metadata of your project.

- **src:**
  - contains your components
  - contains your assets
  - contains your apis
  - contains your state management - redux, mobX.
- **app.js file:**:
  - This file will render your application components.
- **index.js file:**
  - This file will render your app.js file.
- **package.json file:**
  - This file contains the information of the packages you are using.
  - Also contains the information of the commands you can run.

- **What are Components???**

- In Modern programming, component is a small broken down module of a software program.
- Parts of a software program are broken down into components.
- Some types of these components are:
  - Stateful Components
  - Stateless Components
  - Container Components
  - Dumb Components
  - Smart Components
- Different types of React Components:
  - Class Component
  - Function Component
  - **New ➜** React Hooks Component

- **Stateful Components**

- A component which holds some data is called stateful component.

- **Example:**

```
import React from 'react'
class ProductInfo extends React.Component(){
```

```
        constructor(props){
                super(props);
                this.state = {
                    product_name:'Iphone6',
                    product_price:50000,
                    product_qty:100
                }
        }
        render(){

            return(
                <h1> Products Info </h1>
                )

            }
}
```

- **Stateless Components**

- A component which holds no data is called a stateless component.
- It is declared using a function.

- **Example:**

```
import React from 'react'
function display(){
            return(
                <h1> I am Display </h1>
                )
}
export default display;
```

- **Refer the Example**

Next Topic:

- Containers
- React Routing

- **Topic:** Containers

- **What is a Container Component?**

- A container component is a component doesn't have its own component.
- A container component has other components imported inside it.
- A Container component is used to create a singlepage of your web application.
- A single page can have multiple components.
  - **For Example: Divide tatamotors.com - landing page into components?**
  - **Solution:**
    - We need the following components:
      1. <NavigationMenu />
      2. <Slider1 />
      3. <Slider2 />
      4. <Latest />
      5. <About us />
      6. <Footer />
    - Hence, tatamotors.com - main page has total 6 components.
    - These components can be imported in a single page which is your container file in react.

```
import React from 'react'
export default function IndexPageContainer(){

  return(
      <React.Fragment>
          <NavigationMenu />
          <Slider1 />
           <Slider2 />
          <Latest />
          <About us />
          <Footer />
      </React.Fragment>
```

```
        )
}
```

- **Topic:** React Routing

- **What is Routing?**

- Routing in programming means switching from one page to another OR switching from one component to another component.
- Routes are created to switch between components or pages.
- The user will be given a link in the View of your application to navigate to that specific page.

- **Creating Routing Configuration in your React Application:**

- **Step1:** Install your react router

- For web applications you will use ➜ react-router-dom

```
$ npm i react-router-dom
```

- **Step2:** Import BrowserRouter from react-router-dom in your application root file.

```
import { BrowserRouter as Router } from 'react-router-dom'
```

- **Step3:** Wrap the Router around your main <App /> Component

```
import React from 'react';
import ReactDOM from 'react-dom';
import { BrowserRouter as Router } from 'react-router-dom'
import './index.css';
import App from './App';
```

```
import reportWebVitals from './reportWebVitals';
ReactDOM.render(
  <React.StrictMode>
          <Router>
                  <App />
          </Router>
  </React.StrictMode>,
  document.getElementById('root')
);
reportWebVitals();
```

- **Step4:** Create your Routes

- You can create routes in your App.js file or separately in a new file like Routes.js

- Import Switch and Route from react-router-dom

```
import { Switch,Route } from 'react-router-dom'
```

```
import React from 'react'
import './App.css';
import DashboardContainer from './Containers/
DashboardContainer';
import { Switch,Route } from 'react-router-dom'
function App() {
  return (
      <Switch>
          <Route
              exact
              path='/dashboard'
              component={DashboardContainer}
          />
      </Switch>
  );
```

```
}
export default App;
```

- **Topic: Adding Links using react-router-dom**

- **What is the difference between HTML Anchor Tag and React Link Tag??**

  - HTML Anchor Tag is used to do navigate between pages.
  - It has attribute or properties like href.
  - href is given the value of the page you want to navigate to.
  - React Link tag is similar to HTML Anchor Tag.
  - However, it does not refresh the entire page.
  - Instead, it just switches the component which needs to be switched.
  - React Link helps to reduce the refresh rate of the website.
  - Also the pages are switched very fast and enhances the User Experience.

- **Creating Links using your React-Router**

- **Step1:** Import Link from react-router-dom

```
import { Link } from 'react-router-dom'
```

- **Step2:** Create the Link Tag

```
<Link to='/dashboard' />
```

- **Difference between HTML <a> Tag and React <Link> Tag??**

```
<a href='/dashboard' />
```

```
<Link to='/dashboard' />
```

- **Topic:** 404 Route

  - If the route is not found in your application, then the 404 Route will be trigged by the React Router.
  - Usually the 404 Route is by default created by many web technologies such as PHP, Laravel etc etc.
  - However in react, you need to create your custom 404 Page.

- **Creating a 404 Route in React**

- **Step1:** Create a component for the 404 Route

```
import React, from 'react'
import './index.css'
const imgURL = 'https://images.unsplash.com/photo-1532382147828-96bdb28b7b04?
ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&ixlib=rb-1.2.1&auto=format&fi
t=crop&w=1100&q=80'
export default function Page404() {
    return (
        <div className="not-found-page">
            <img
                src={imgURL}
                alt='404.png'
            />
        </div>
    )
}
```

- **Step2:** Design the component with some CSS

```
.not-found-page{
```

```
        top:10%;
        left:10%;
        position:fixed;

}
```

- **Step3:** Create a Route in your router to activate the 404 Page

```
<Route path='*' component={Page404} />
```

Questions:

Q.1[Andy]
Why we do import:

```
import {Route} from ' react-route-dom'
```

Instead of:

```
import Route from ' react-route-dom'
```

Ans:

- Because it is a not a default exported file.

Q.2[Poonam] Types of file exports

1. **Default Exports:**

    - If you want to export a file, **after you create it** use default exports

2. **Normal Exports:**

- If you want to export a file, **when you are creating it**. then just use a normal export.

Topic: Passing Dynamic data using React Link Tags

- We can pass data in the url.

```
<Link to='/player/:id'/>
```

- We can pass data in the body.

```
<Link to={{
                pathname:'/mumbai-indians',
                props:{
                    owner:'Nita Ambani',
                    head_coach:'Mahela Jayawardene',
                    bowling_coach:'Shane Bond'
                }
            }}>
</Link>
```

Questions and Doubts:

Q.1[Poonam] Form Submission in React?

Ans:

- In React, form submission can be handled using various ways.
- Let us look at one approach.

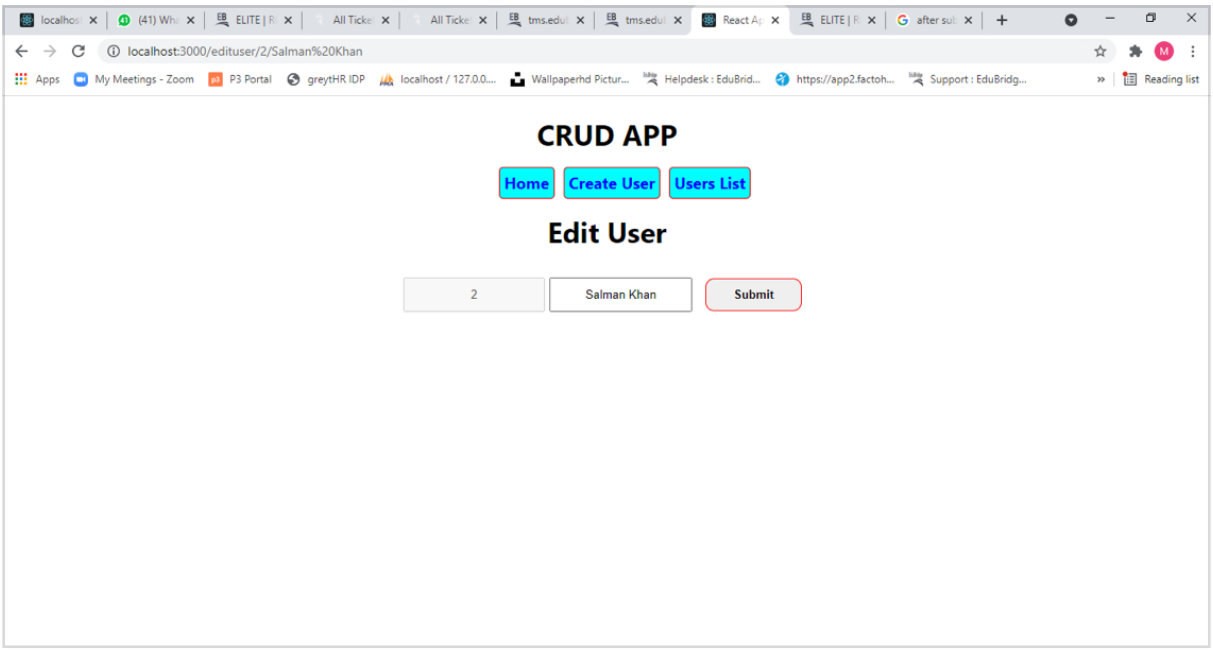
1. Create your HTML Form.

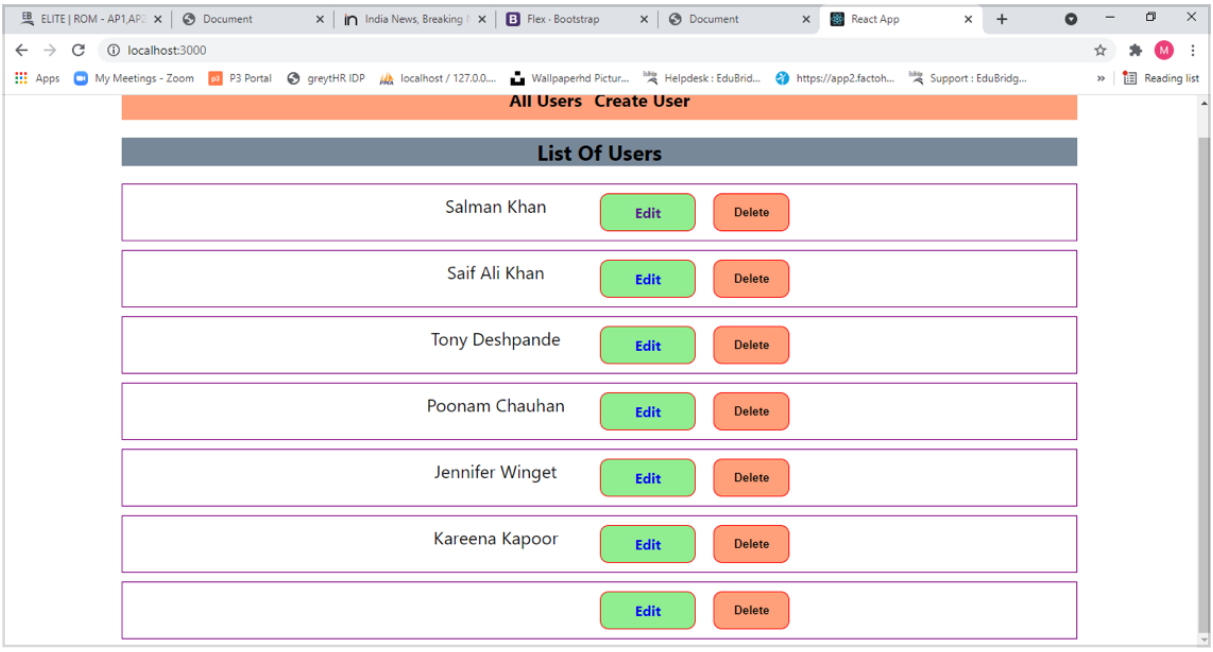
2. Create submitData() function for the form and add it as an attribute of the <form> tag.

3. Create handleInputChange() function and add it as an attribute to all the HTML Input tags in the form.

Questions:

- **Q.1:** Any more Questions or Doubts on React-Hooks??
- **Q.2[Poonam]:** After submitting need an message “Data inserted Successfully , Data deleted successfully and data updated successfully how to do that?





- **Q.3** What are the Drawbacks of using JSON-Server?

Ans:

- Drawbacks can be:

1. You cannot use JSON-server for highly complicated database applications.
2. You cannot define the data type of the data you are storing in the JSON-server.
3. It is good for learning purposes and to test your API functionalities.

- **Topic:** Submitting Forms in React using React-Hook-Form

- **Step1:** Install npm package

```
$ npm install react-hook-form
```

- **Step2:** Import useForm()

```
import { useForm } from 'react-hook-form'
```

- **Step3:** Import the properties of useForm()

```
const { register, handleSubmit, formState: { errors } } = useForm();
```

- **Step4:** Call the handleSubmit() method on the <form> tag.

```
<form onSubmit={handleSubmit()}>
```

```
</form>
```

- **Step5:** Create a function to submit the data of the form in your react component.

```
const submitData = ()=>{
}
```

- **Step6:** Add name property and register() to every HTML input tag of the form.

- register() method will take 2 arguments:
- name attribute - the name attribute of the input box
- boolean value - whether the input is required or not

```
<input
      name="name"
      type="text"
      placeholder="Enter Name"
      {...register('name',{required:false})}
/>
```

- **Step7:** Show errors

```
<input
      name="name"
      type="text"
      placeholder="Enter Name"
      {...register('name',{required:false})}
/>
{errors.name && <span>Name field is required!</span>}
```

Topics Covered in last session - 16th Oct 2021:

1. React Hooks
2. React Form Submission using package called as react-hook-form
3. React Toast Notifications

- **Doubts or Questions:**

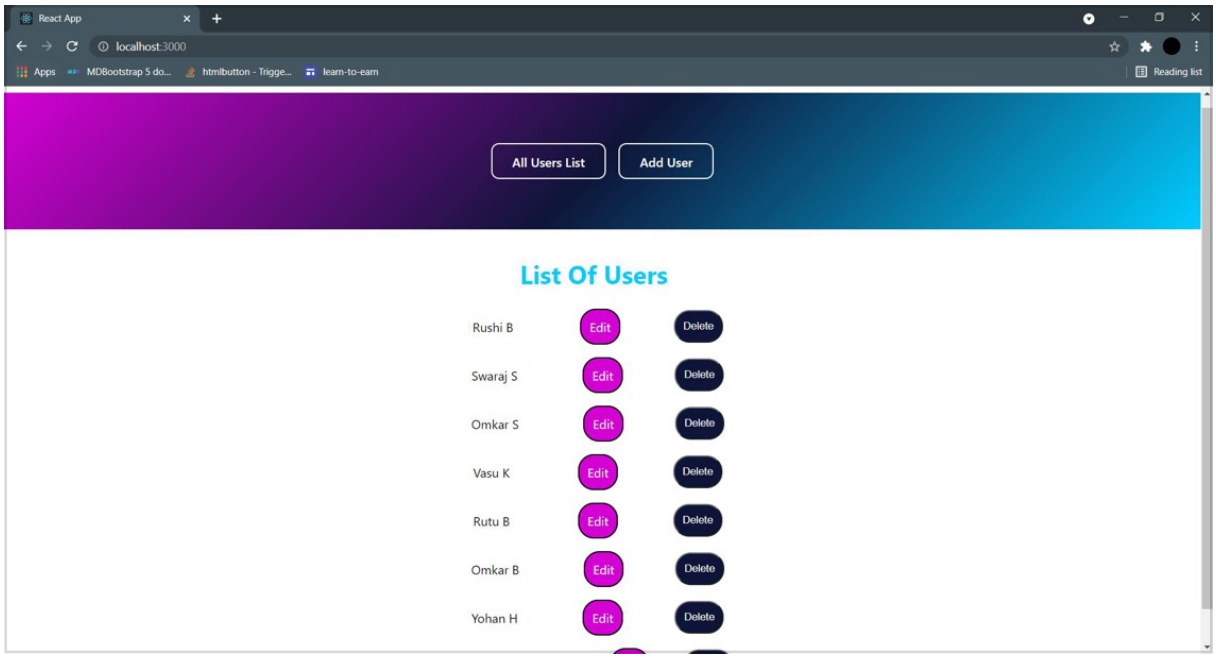
1. **[Poonam]** Why is React Hooks Introduced in React?

Ans:

- To improve App Performance.
- To make developers possible to have component state in functional components.

2. Redux completed????

Last Completed Project in React: Users CRUD Application



- **Technologies used**: Users CRUD Application

1. React.js - **Yes**
2. HTML - **Yes**

3. CSS(Normal) - **Yes**
4. react-toastify - **No**
5. axios - **Yes**
6. Form handling packages - **No**

Mini Project - 24th Oct 2021: Users Crud Application Version 2.0 using React and React Redux

---

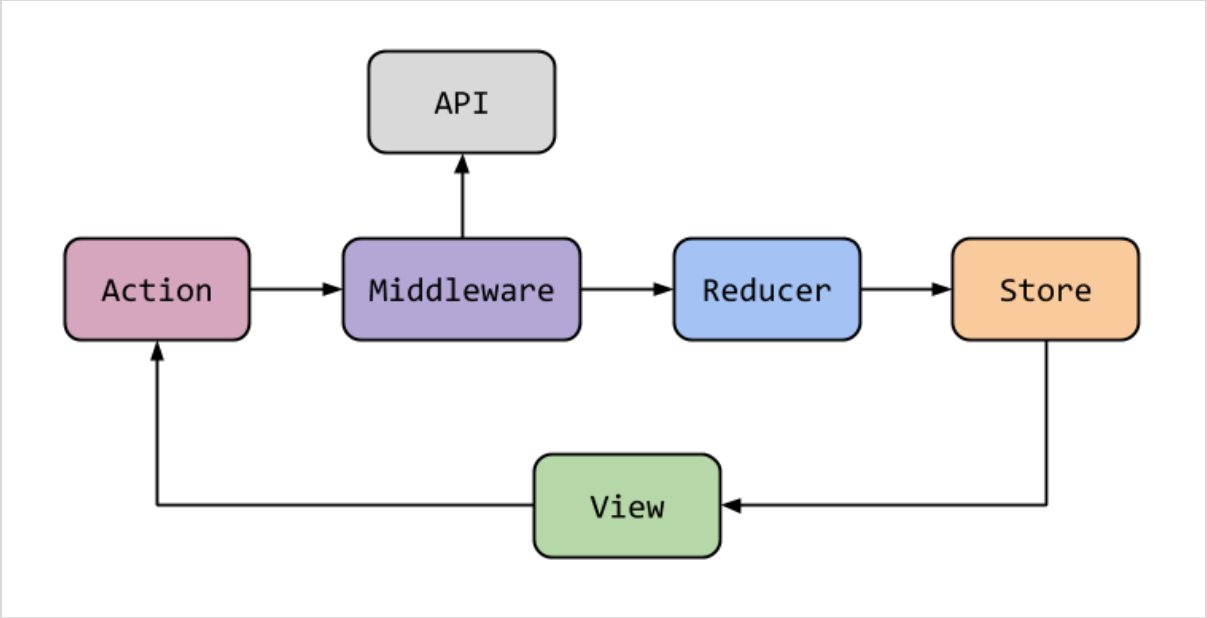
- **Technologies to use:**

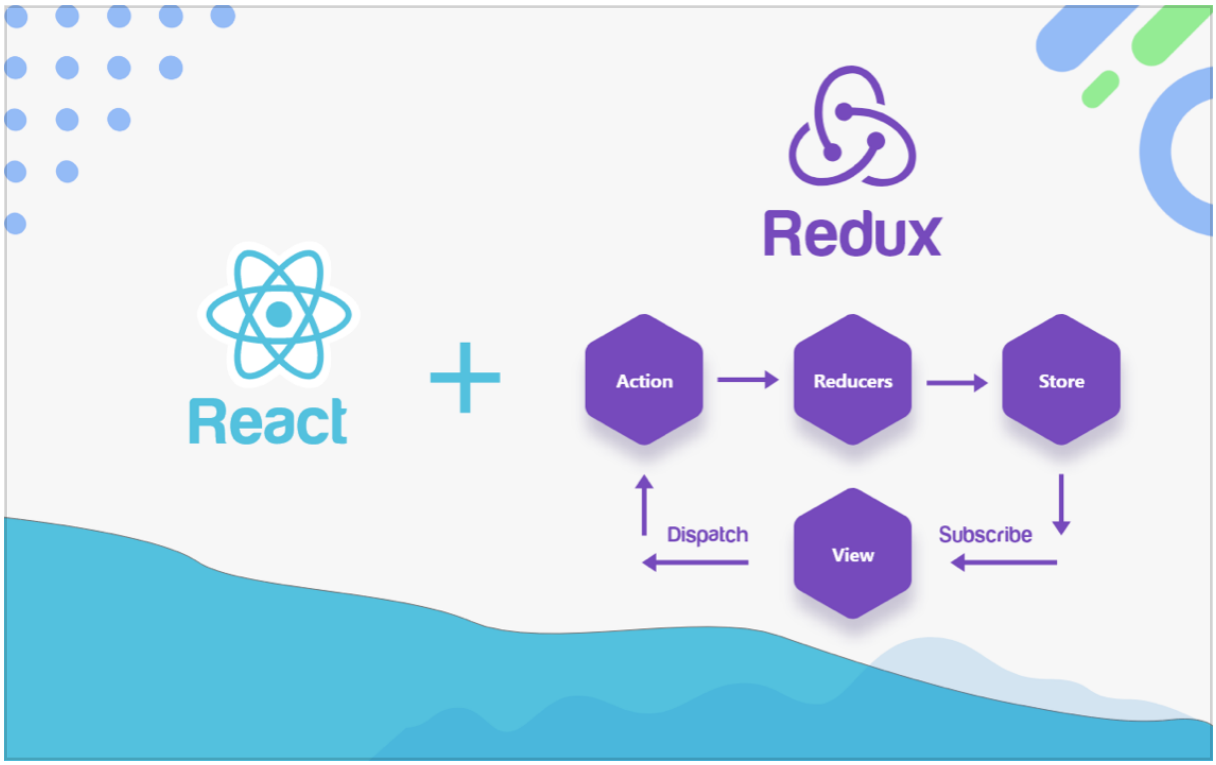
---

1. React.js
2. Redux
3. React-Redux
4. react-hook-form
5. Axios

- **What is Redux?**

- Redux is an application state manager.
- It stores data inside it.
- It provides that data to the entire application structure.





Why Redux is needed in your App

1. To improve the speed of your data transmission from one component to another component.
2. You **don't have to call your database** every time you need to update your application.
3. For Security Purposes - to store user login data etc.

- **Step1:** Install React App with Redux and React-Redux
- **Step2:** Create store.js inside src folder
- **Step3:** Create reducers folder inside src folder
- **Step4:** Create actions folder and create types.js inside it.