

CLOUD APPLICATION DEVELOPMENT GROUP ASSIGNMENT 2

Maxwel Ochieng- 20/02886

Sylvester Omondi -20/02869

Francis Gichuhi 20/02967

20/02824 Samuel Mwangi Nduati

Matta Kimani - 20/03569

Lucy Kuria 20/02796

Daniel Goye - 20/02965

Collins Kathuli - 20/02832

Monica Nabifwo -20/03218

Peter muhoro-20/03226

Collins Cheruiyot - 20/02912

Question

Develop a simple application using html css and javascript that can accept names, email ID, Date of birth and Course.

Using Javascript validate the data entered by the user

Answer

In our assignment, we used react js, Tailwind css and react-toastify library, react is javascript library used to enhance faster development of web pages while tailwind is a CSS utility class library that is used to fasten styling during development and on the other hand react-toastify is used to handle errors in logical manner to the user.

React has virtual dom already setup for the developer, so you don't have to start targeting individual DOM elements. React is setup in a way everything that get rendered on the browser resides inside **index.js** and the pages you build which are termed as components resides inside **app.js**

Index .js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5 import { ToastContainer } from "react-toastify";
6 import "react-toastify/dist/ReactToastify.css";
7
8
9 const root = ReactDOM.createRoot(document.getElementById('root'));
10 root.render(
11   <React.StrictMode>
12     <ToastContainer />
13     <App />
14   </React.StrictMode>
15 );
```

App.js



```
1 import Registers from "../components/Registers";
2
3 function App() {
4   return (
5     <div>
6       <Registers />
7     </div>
8   );
9 }
10
11 export default App;
12
```

Registers is our main page that has all the jsx(html code inside react) code that will enable the user to login.

Register.js will contain all our signup logic taking into account all the user inputs during signup. Down below we will have screen of the whole code and then we will explain bit by bit

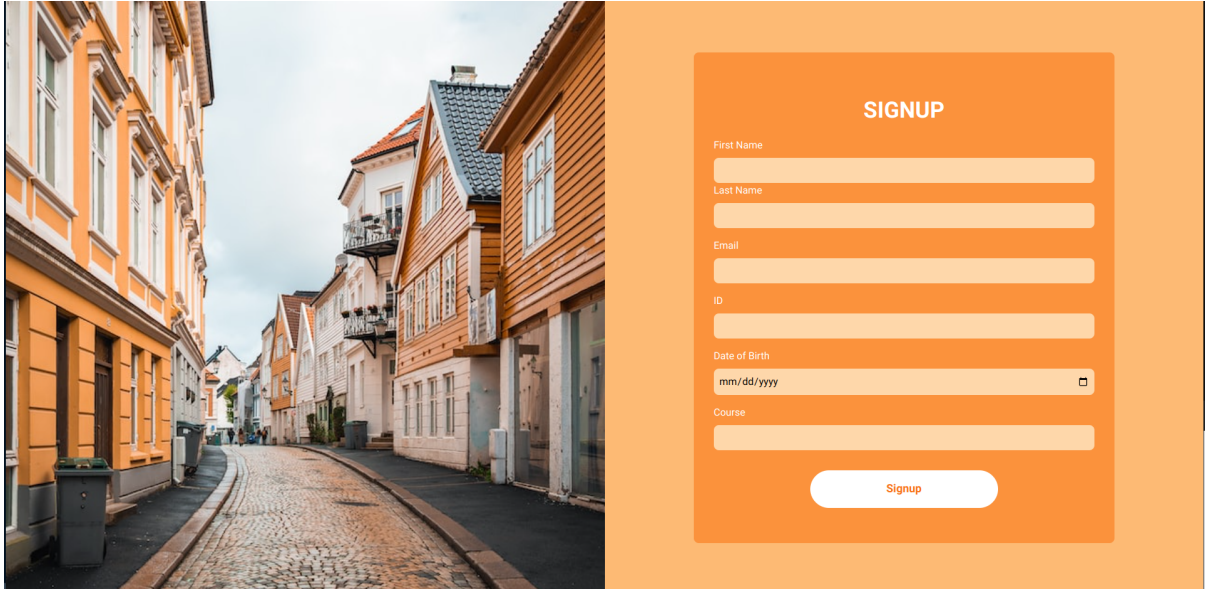
```
1 import React, { useState } from "react";
2 import login from "../assets/login1.jpg";
3 import { toast } from "react-toastify";
4
5 const Registers = () => {
6   const [state, setState] = useState({
7     firstName: "",
8     lastName: "",
9     email: "",
10    dateofBirth: "",
11    course: "",
12    userId: "",
13  });
14
15  function handleOnChange(e) {
16    setState({
17      ...state,
18      [e.target.name]: e.target.value,
19    });
20  }
21
22  function handleSubmit(e) {
23    e.preventDefault();
24    if (
25      !state.firstName ||
26      !state.lastName ||
27      !state.course ||
28      !state.dateofBirth ||
29      !state.userId ||
30      !state.email
31    ) {
32      return toast.error("all fields are required");
33    }
34    console.log("values:", state);
35    toast.success("details captured successfully");
36    return setState({
37      ...state,
38      firstName: "",
39      lastName: "",
40      email: "",
41      dateofBirth: "",
42      course: "",
43      userId: "",
44    });
45  }
```

```

1   return (
2     <div className="grid grid-cols-1 md:grid-cols-2 h-screen w-full">
3       <div className="hidden md:block">
4         <img
5           className="h-screen w-full object-cover"
6           src={login}
7           alt="login_image"
8         />
9       </div>
10      <div className="bg-orange-300 flex flex-col justify-center text-white">
11        <form
12          onSubmit={(e) => handleSubmit(e)}
13          className="max-w-2xl mx-auto w-full bg-orange-400 p-14 px-8 rounded-lg"
14        >
15          <h2 className="text-4xl font-bold text-center p-4">SIGNUP</h2>
16          <div className="flex flex-col py-2">
17            <label>First Name</label>
18            <input
19              className="rounded-lg text-black bg-orange-200 mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
20              type="text"
21              name="firstName"
22              value={state.firstName}
23              onChange={(e) => handleOnChange(e)}
24            />
25            <label>Last Name</label>
26            <input
27              className="rounded-lg text-black bg-orange-200 mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
28              type="text"
29              name="lastName"
30              value={state.lastName}
31              onChange={(e) => handleOnChange(e)}
32            />
33          </div>
34          <div className="flex flex-col py-2">
35            <label>Email</label>
36            <input
37              className="rounded-lg text-black bg-orange-200 mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500"
38              type="text"
39              name="email"
40              value={state.email}
41              onChange={(e) => handleOnChange(e)}
42            />
43          </div>
44          <div className="flex flex-col py-2">
45            <label>ID</label>
46            <input
47              className="rounded-lg text-black bg-orange-200 mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
48              type="number"
49              name="userId"
50              value={state.userId}
51              onChange={(e) => handleOnChange(e)}
52            />
53          </div>
54          <div className="flex flex-col py-2">
55            <label>Date of Birth</label>
56            <input
57              className="rounded-lg bg-orange-200 mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 text-black"
58              type="date"
59              name="dateOfBirth"
60              value={state.dateOfBirth}
61              onChange={(e) => handleOnChange(e)}
62            />
63          </div>
64          <div className="flex flex-col py-2">
65            <label>Course</label>
66            <input
67              className="rounded-lg bg-orange-200 text-black mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
68              type="text"
69              name="course"
70              value={state.course}
71              onChange={(e) => handleOnChange(e)}
72            />
73          </div>
74          <div className="flex justify-center">
75            <button
76              type="submit"
77              className="bg-white text-orange-500 text-lg font-semibold w-[300px] p-4 mt-6 rounded-full"
78            >
79              Signup
80            </button>
81          </div>
82        </form>
83      </div>
84    </div>
85  );
86 };
87
88 export default Registers;

```

So basically Register is a functional component in react that can both execute javascript code and return a JSX(html code) inside it. From the return pont we render our html code with styles that will produce this signup form below on a desktop and mobile view(it's a good practice to have both when designing web pages)

The image shows a desktop view of a web page. On the left side, there is a photograph of a narrow, cobblestone street in a European-style town, lined with colorful, multi-story buildings. On the right side, there is a large, orange-themed signup form. The form has a title 'SIGNUP' at the top. Below the title, there are several input fields: 'First Name', 'Last Name', 'Email', 'ID', 'Date of Birth' (with a date picker icon), and 'Course'. At the bottom of the form is a 'Signup' button.

Desktop View

The image shows a smartphone screen displaying a web browser with a signup form. The browser's address bar shows the URL 'http://localhost:3000'. The form has an orange background and contains the following fields: 'First Name', 'Last Name', 'Email', 'ID', 'Date of Birth' (with a date picker icon), and 'Course'. A white 'Signup' button is at the bottom of the form. The phone's status bar at the top shows '12:34 PM', '5G', and a battery icon. The bottom of the screen features a standard mobile browser navigation bar with back, forward, share, bookmark, and tabs icons.

12:34 PM 5G

AA http://localhost:3000

SIGNUP

First Name

Last Name

Email

ID

Date of Birth

Course

Signup

Mobile View

Forms Inputs

React handles form inputs in form of states. Checking at my screenshot above with the JSX code you notice the form attributes (onSubmit) and if you check the input tags I have attribute (onChange and value). The onChange attribute which handles the function handleOnChange() helps to capture any change in that field which maps it to the value that takes the state.name of the field. At first when user visits this page I have my states at empty strings

```
1  const Registers = () => {  
2    const [state, setState] = useState({  
3      firstName: "",  
4      lastName: "",  
5      email: "",  
6      dateOfBirth: "",  
7      course: "",  
8      userId: "",  
9    });
```

After the user starts to fill in the form I want to populate these states with event.target.value that gets keyed in in every field which the handleOnChange function suggests

```
1  function handleOnChange(e) {  
2    setState({  
3      ...state,  
4      [e.target.name]: e.target.value,  
5    });  
6  }
```


After the user is done, now he/she has the ability to click signup and this invokes the function `handleSubmit` which will validate certain criterias before submitting the data entered by the user at that particular instance.

1. Check if all the fields have value in them if so return an error using toast.

We do not need to validate ID or email because form inputs will do that for me with types eg `type="email"` or `type="number"`

```
1 function handleSubmit(e) {
2   e.preventDefault();
3   if (
4     !state.firstName ||
5     !state.lastName ||
6     !state.course ||
7     !state.dateOfBirth ||
8     !state.userId ||
9     !state.email
10  ) {
11    return toast.error("all fields are required");
12  }
13  console.log("values:", state);
14  toast.success("details captured successfully");
15  return setState({
16    ...state,
17    firstName: "",
18    lastName: "",
19    email: "",
20    dateOfBirth: "",
21    course: "",
22    userId: "",
23  });
24 }
```

If the user enters every field correctly I will definitely console the data and look at it in my console. At this point you can do whatever you want with the data eg send it to a database.

Below are scenarios of the form if correct data is keyed in or not

1. Form rules not Obeyed

all fields are required

First Name

Developer

Last Name

Maxwel

Email

developer.maxwel@gmail.com

ID

Date of Birth

11/17/2022

Course

Signup

I left two fields which are of course and ID, and toast returned me that error (“all fields are required”). This means that you can ensure that no null value is sent back to the database.

2. Ensuring all fields are corrects and submitting

✓ details captured successfully

First Name

Last Name

Email

ID

Date of Birth

mm/dd/yyyy

Course

Signup

The data get submitted and If I check my console I get this

id successfully

3 messages

2 user mess...

No errors

1 warning

2 info

No verbose

Download the React DevTools for a better development experience: <https://reactjs.org/link/react-devtools>

values: {firstName: 'Developer', lastName: 'Maxwel', email: 'developer.maxwel@gmail.com', dateofBirth: '2022-11-24', course: 'B5D', firstName: 'Developer', lastName: 'Maxwel', userId: '2002886'}

[[Prototype]]: Object