CLOUD APLICATION DEVELOPMENT
GROUP ASSIGNMENT 2

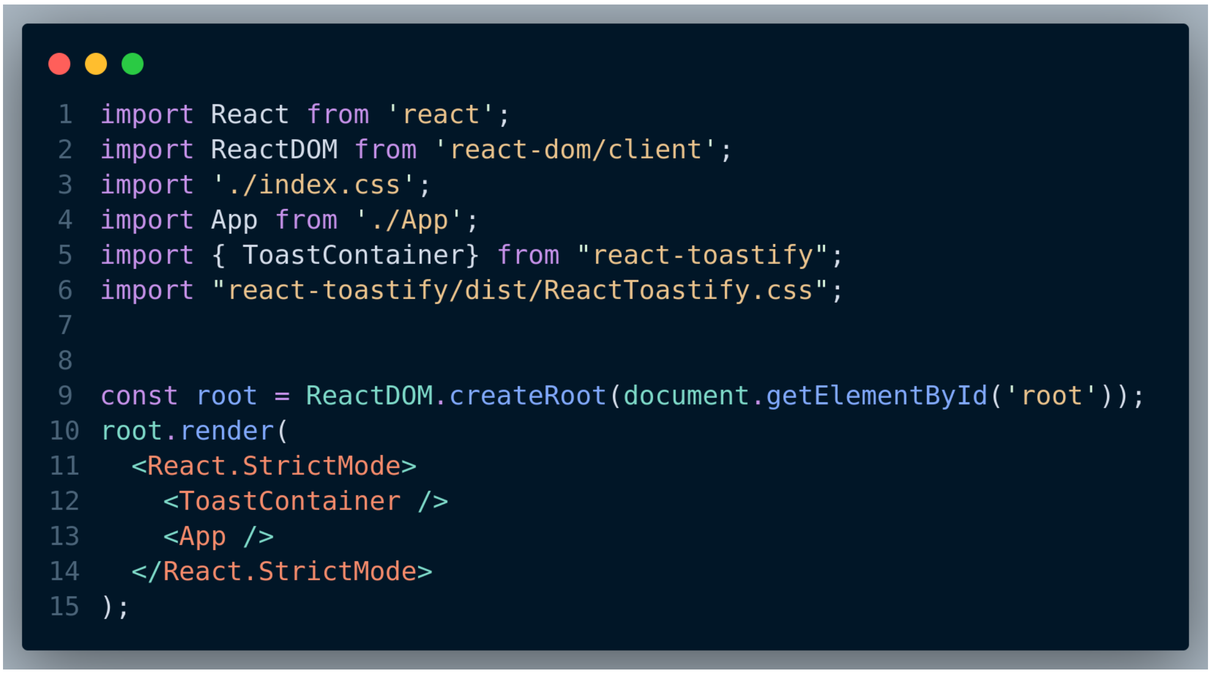
Maxwel Ochieng- 20/02886
Sylvester Omondi -20/02869
Francis Gichuhi 20/02967
20/02824 Samuel Mwangi Nduati
Matta Kimani - 20/03569
Lucy Kuria 20/02796
Daniel Goye - 20/02965
Collins Kathuli - 20/02832
Monica Nabifwo -20/03218
Peter muhoro-20/03226
Collins Cheruiyot - 20/02912

Question
Develope 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 a 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.

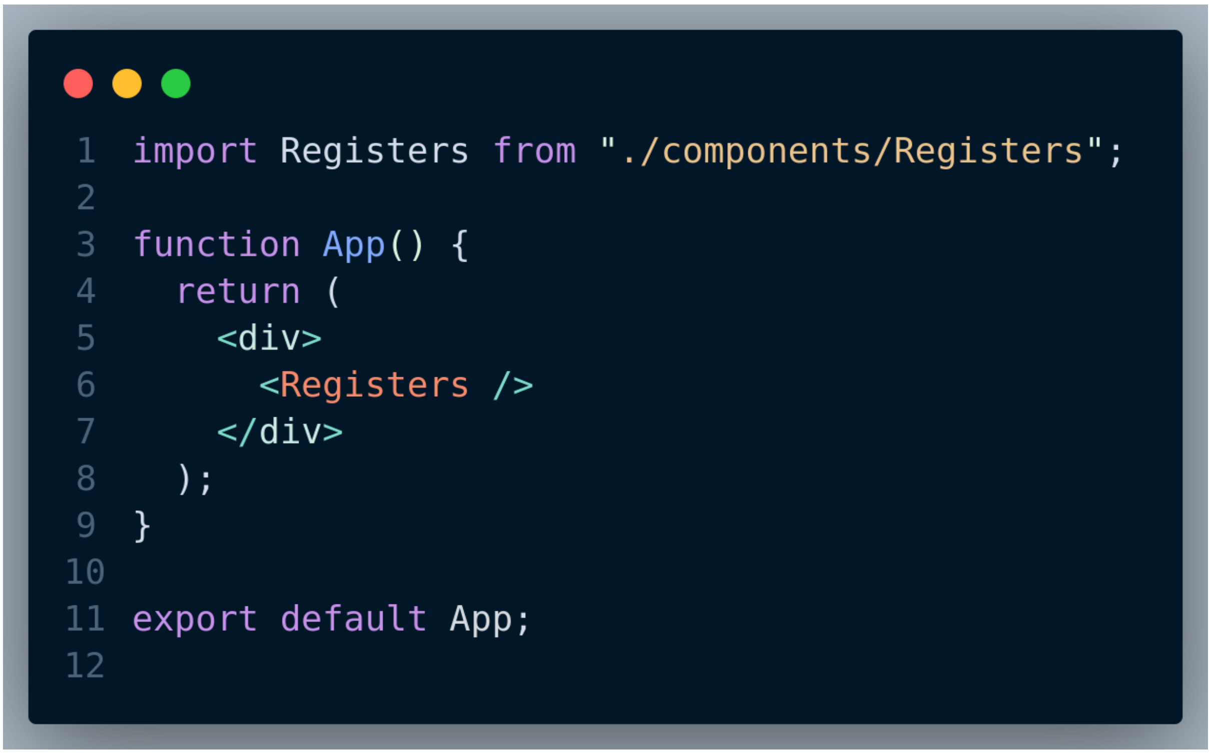
React has virtual dom already setup for the developer, so you don't have to start targeting individual DOM elements. React is setup in a way everything that get rendered on the browser resides inside **index.js** and the pages you build which are termed as components resides inside **app.js**

Index .js



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { ToastContainer} from "react-toastify";
import "react-toastify/dist/ReactToastify.css";


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <ToastContainer />
    <App />
  </React.StrictMode>
);
```

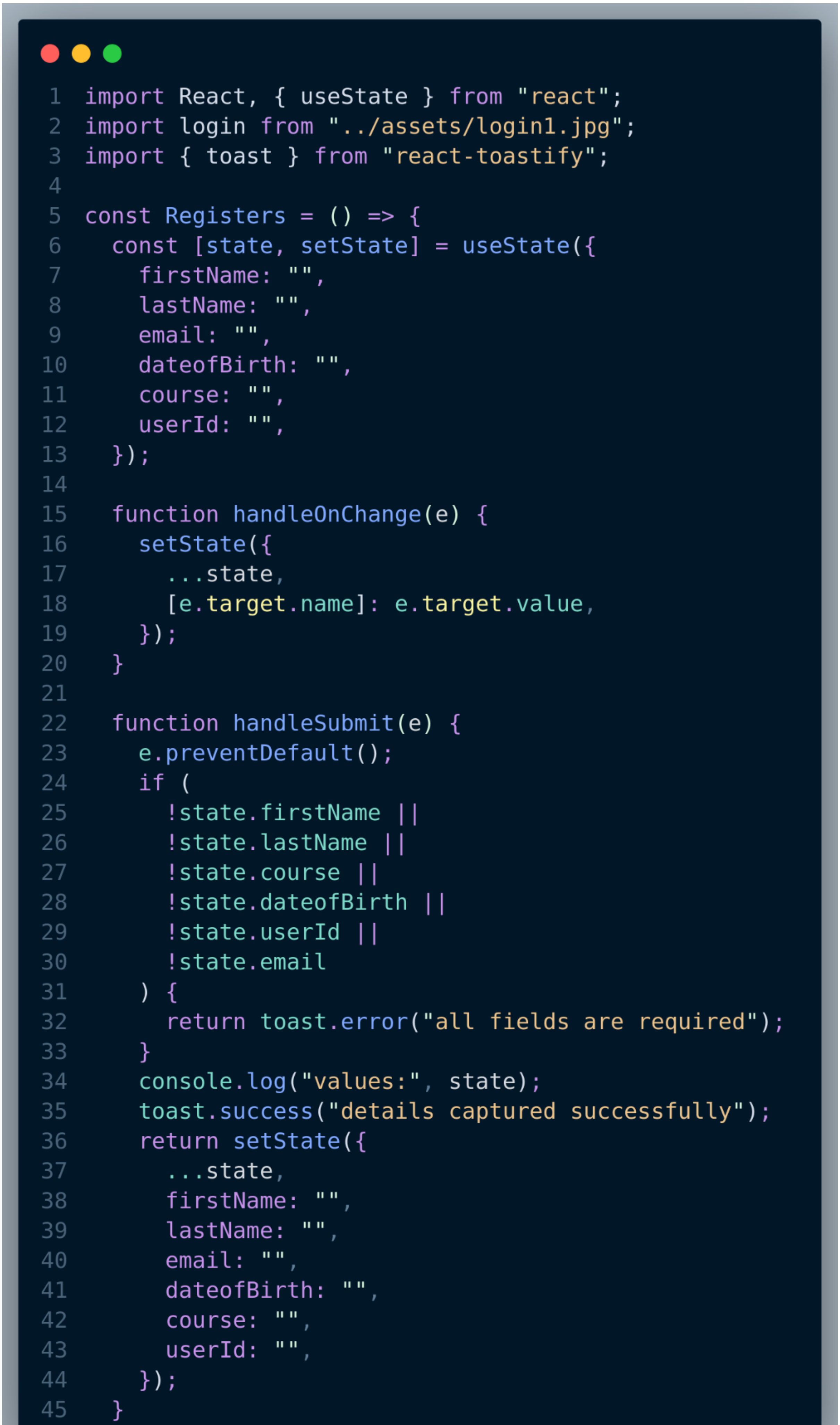
App.js



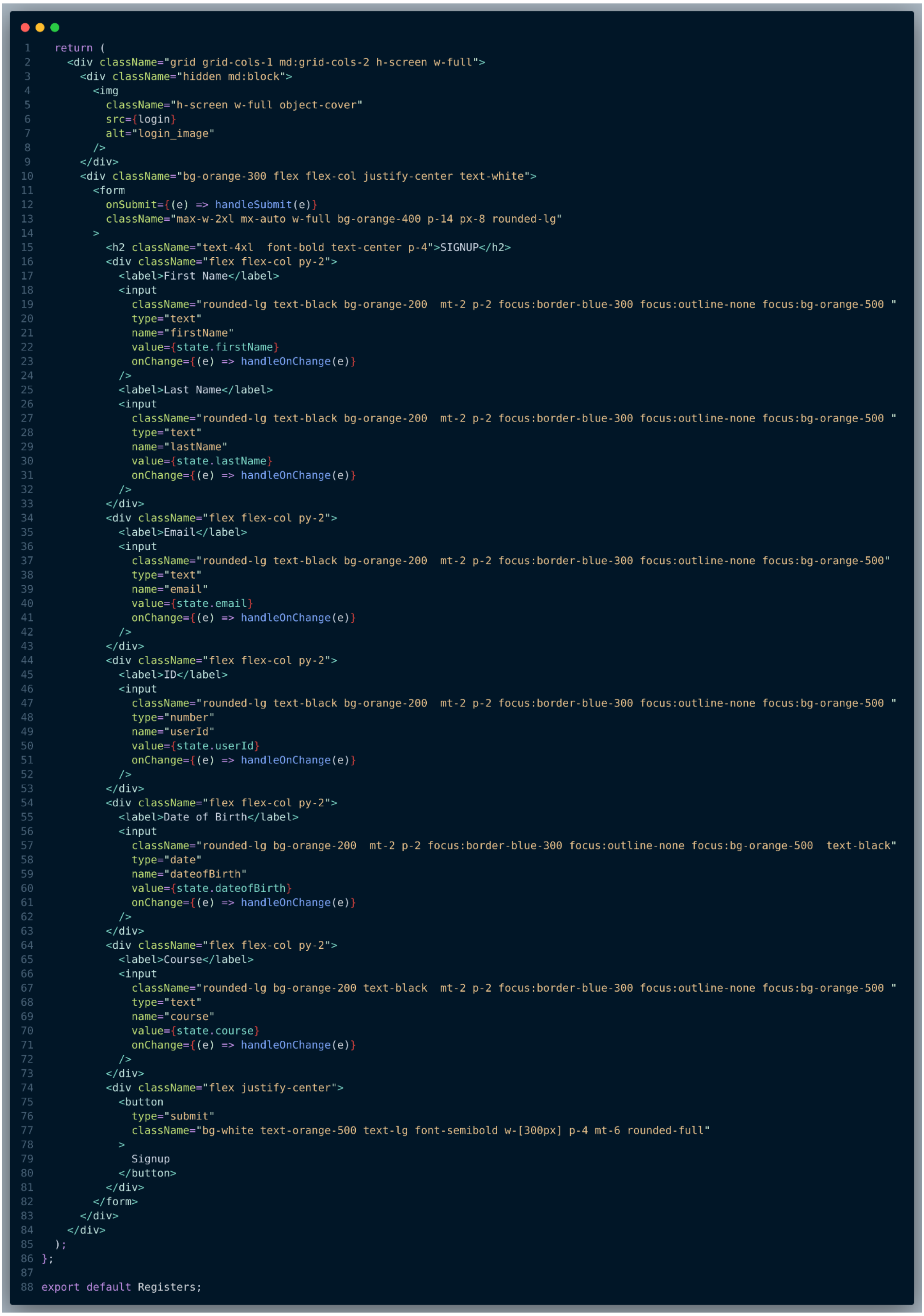
```
import Registers from "./components/Registers";

function App() {
  return (
    <div>
      <Registers />
    </div>
  );
}

export default App;

```

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

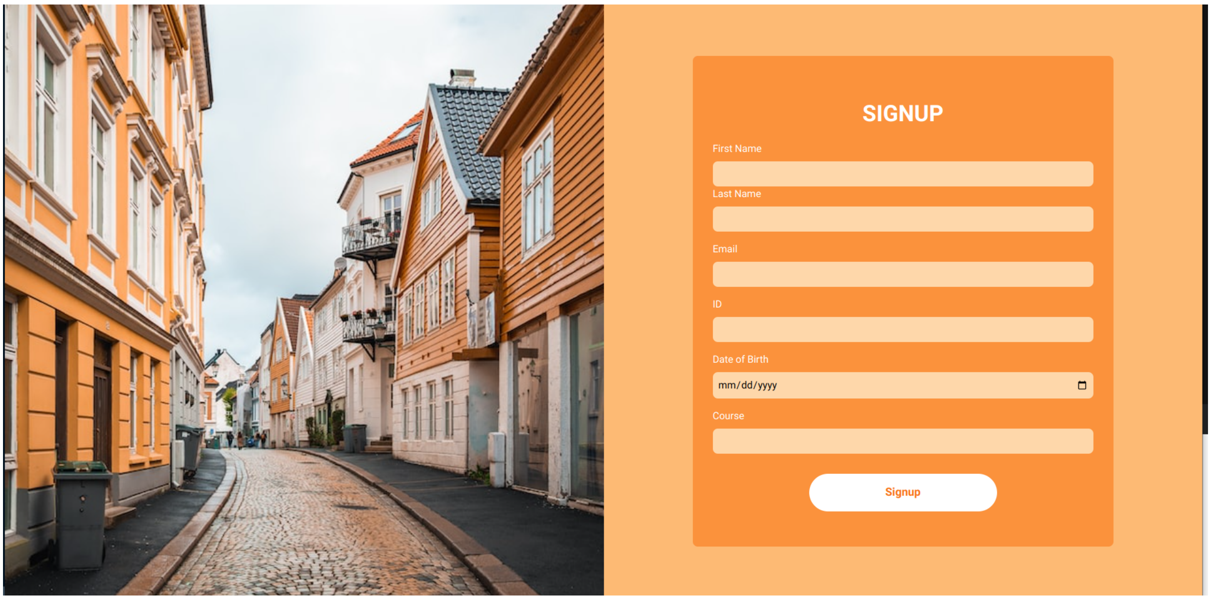


```
import React, { useState } from "react";
import login from "../assets/login1.jpg";
import { toast } from "react-toastify";

const Registers = () => {
  const [state, setState] = useState({
    firstName: "",
    lastName: "",
    email: "",
    dateofBirth: "",
    course: "",
    userId: "",
  });

  function handleOnChange(e) {
    setState({
      ...state,
      [e.target.name]: e.target.value,
    });
  }

  function handleSubmit(e) {
    e.preventDefault();
    if (
      !state.firstName ||
      !state.lastName ||
      !state.course ||
      !state.dateofBirth ||
      !state.userId ||
      !state.email
    ) {
      return toast.error("all fields are required");
    }
    console.log("values:", state);
    toast.success("details captured successfully");
    return setState({
      ...state,
      firstName: "",
      lastName: "",
      email: "",
      dateofBirth: "",
      course: "",
      userId: "",
    });
  }
```

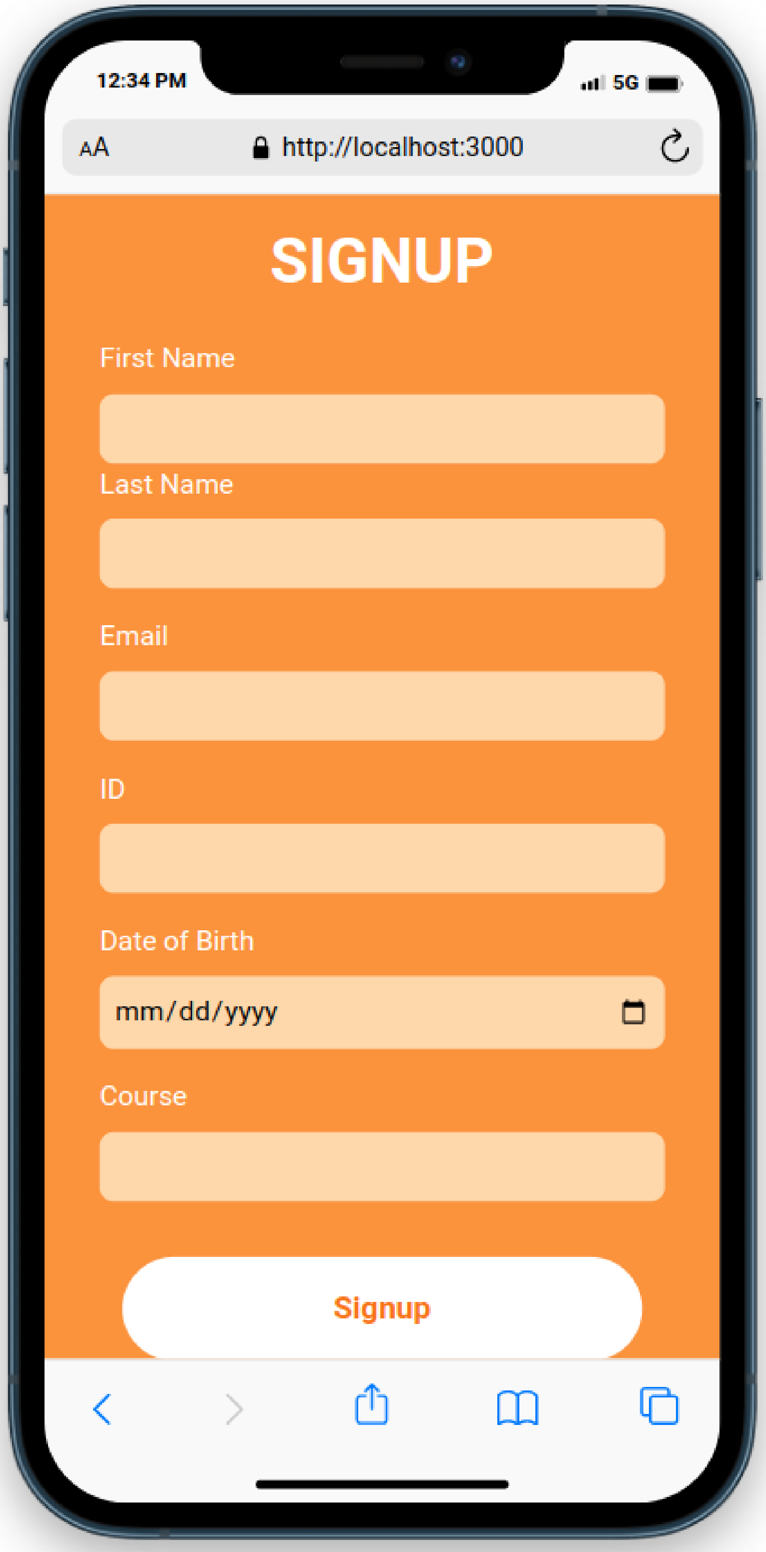


```
  return (
    <div className="grid grid-cols-1 md:grid-cols-2 h-screen w-full">
      <div className="hidden md:block">
        <img
          className="h-screen w-full object-cover"
          src={login}
          alt="login_image"
        />
      </div>
      <div className="bg-orange-300 flex flex-col justify-center text-white">
        <form
          onSubmit={(e) => handleSubmit(e)}
          className="max-w-2xl mx-auto w-full bg-orange-400 p-14 px-8 rounded-lg"
        >
          <h2 className="text-4xl  font-bold text-center p-4">SIGNUP</h2>
          <div className="flex flex-col py-2">
            <label>First Name</label>
            <input
              className="rounded-lg text-black bg-orange-200  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
              type="text"
              name="firstName"
              value={state.firstName}
              onChange={(e) => handleOnChange(e)}
            />
            <label>Last Name</label>
            <input
              className="rounded-lg text-black bg-orange-200  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
              type="text"
              name="lastName"
              value={state.lastName}
              onChange={(e) => handleOnChange(e)}
            />
          </div>
          <div className="flex flex-col py-2">
            <label>Email</label>
            <input
              className="rounded-lg text-black bg-orange-200  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500"
              type="text"
              name="email"
              value={state.email}
              onChange={(e) => handleOnChange(e)}
            />
          </div>
          <div className="flex flex-col py-2">
            <label>ID</label>
            <input
              className="rounded-lg text-black bg-orange-200  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
              type="number"
              name="userId"
              value={state.userId}
              onChange={(e) => handleOnChange(e)}
            />
          </div>
          <div className="flex flex-col py-2">
            <label>Date of Birth</label>
            <input
              className="rounded-lg bg-orange-200  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500  text-black"
              type="date"
              name="dateofBirth"
              value={state.dateofBirth}
              onChange={(e) => handleOnChange(e)}
            />
          </div>
          <div className="flex flex-col py-2">
            <label>Course</label>
            <input
              className="rounded-lg bg-orange-200 text-black  mt-2 p-2 focus:border-blue-300 focus:outline-none focus:bg-orange-500 "
              type="text"
              name="course"
              value={state.course}
              onChange={(e) => handleOnChange(e)}
            />
          </div>
          <div className="flex justify-center">
            <button
              type="submit"
              className="bg-white text-orange-500 text-lg font-semibold w-[300px] p-4 mt-6 rounded-full"
            >
              Signup
            </button>
          </div>
        </form>
      </div>
    </div>
  );
};

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)



Desktop View



12:34 PM

5G

AA http://localhost:3000

SIGNUP

First Name

Last Name

Email

ID

Date of Birth

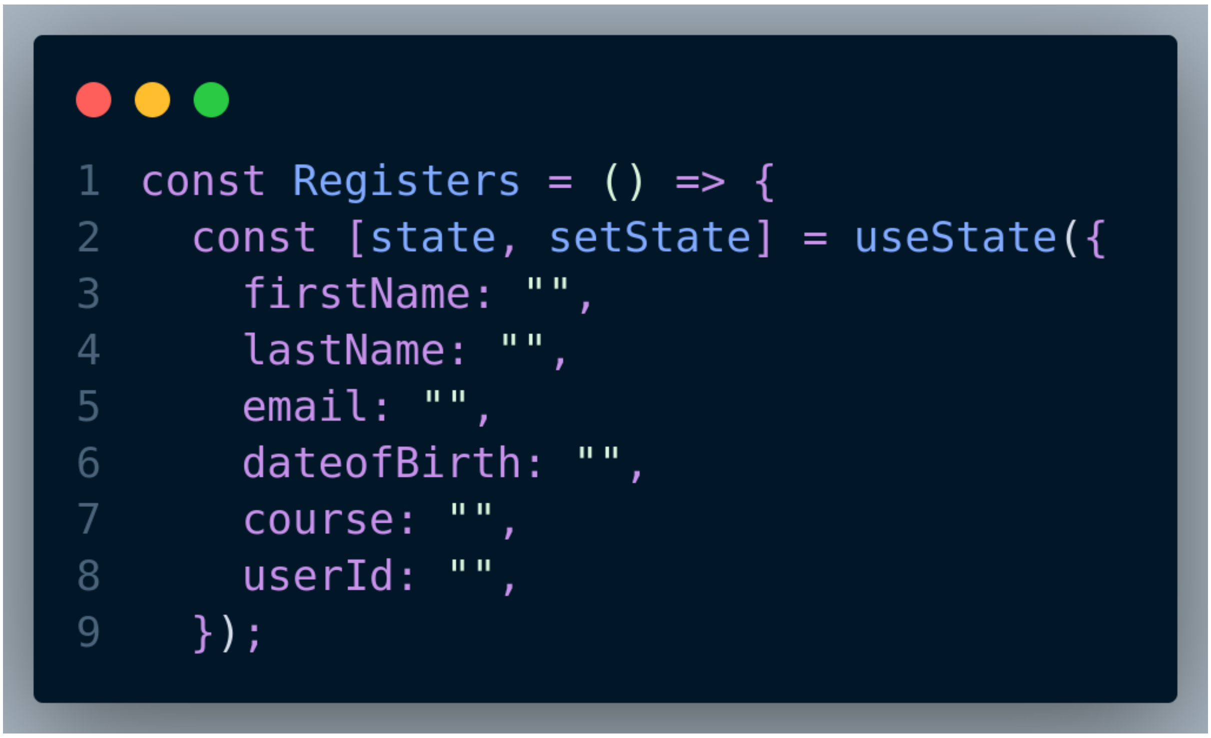
mm/dd/yyyy

Course

Mobile View

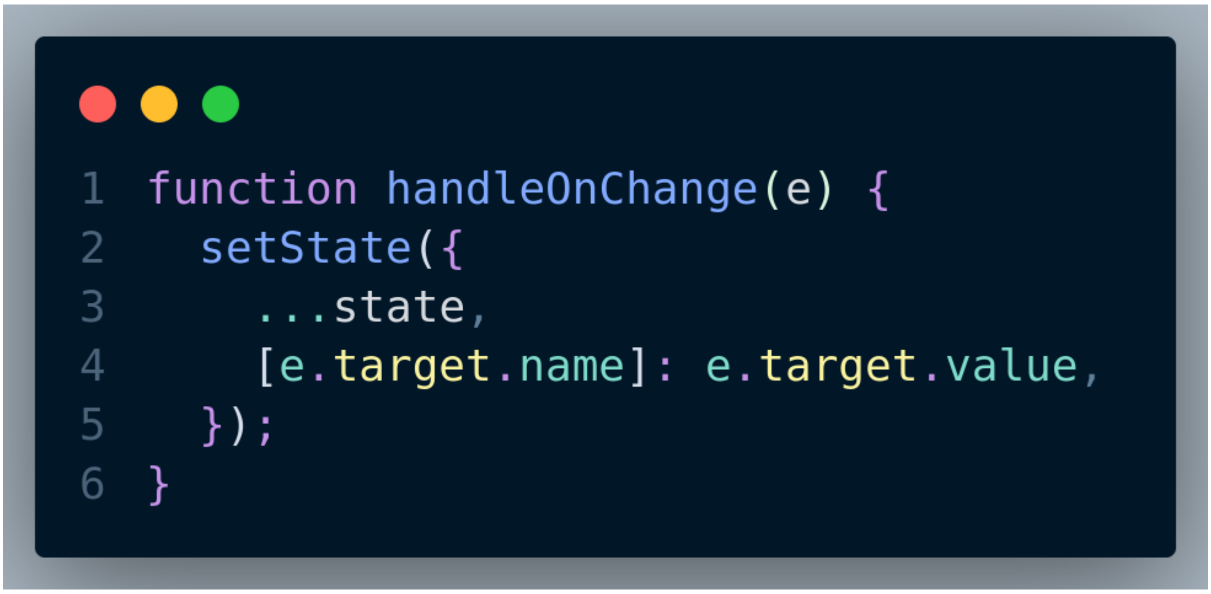
Forms Inputs

React handles form inputs in form of states. Checking at my screenshot above with thye 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 mappes it ot the value that takes the state.name of the field. At first when user visits this page I have my states at empty strings



```
const Registers = () => {
  const [state, setState] = useState({
    firstName: "",
    lastName: "",
    email: "",
    dateofBirth: "",
    course: "",
    userId: "",
  });
```

After th user starts to fill in the form I want to populate this states with event.target.value that get keyed in in every field which th handleOnchange function suggests

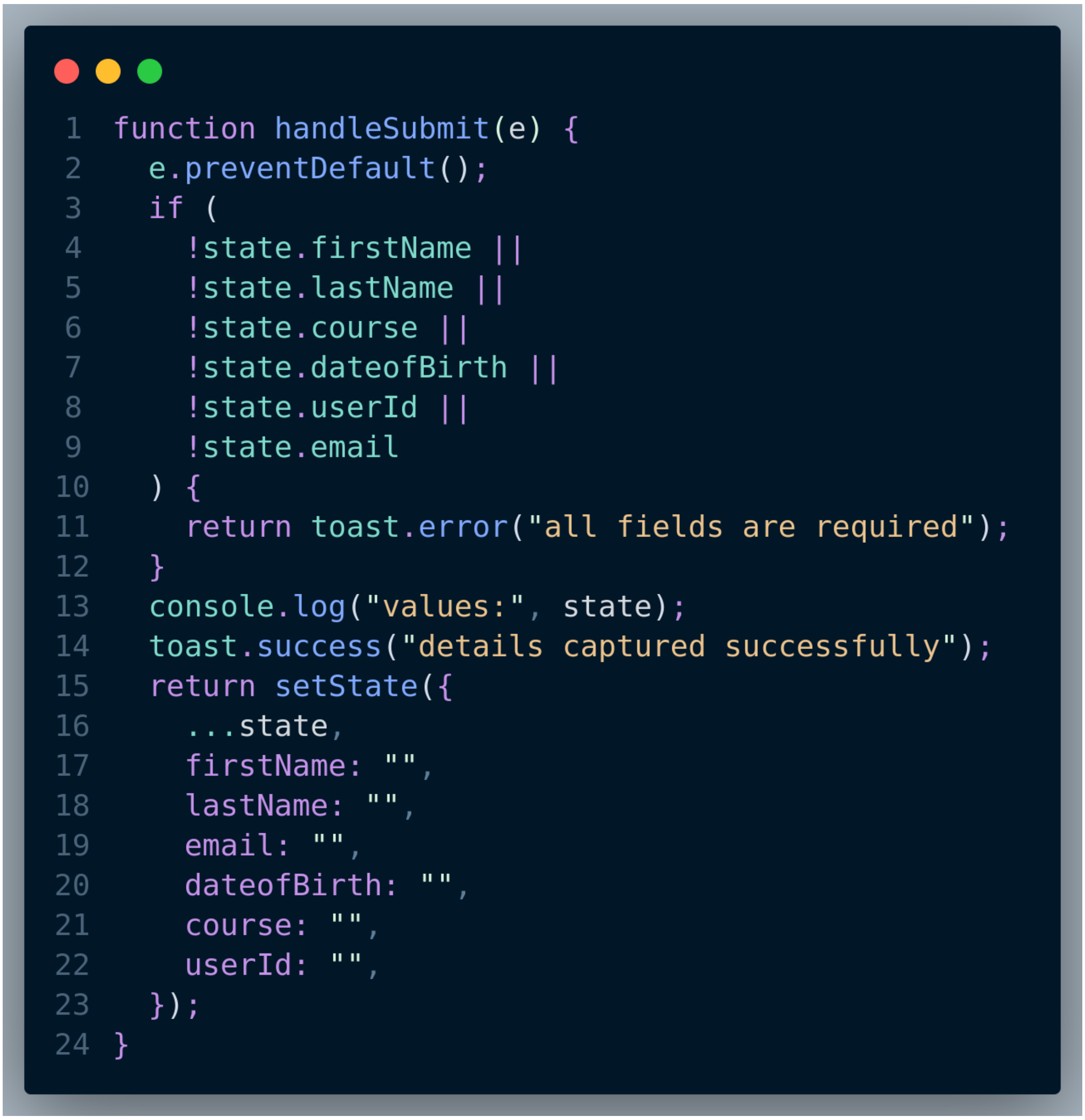


```
function handleOnChange(e) {
  setState({
    ...state,
    [e.target.name]: e.target.value,
  });
}
```

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”

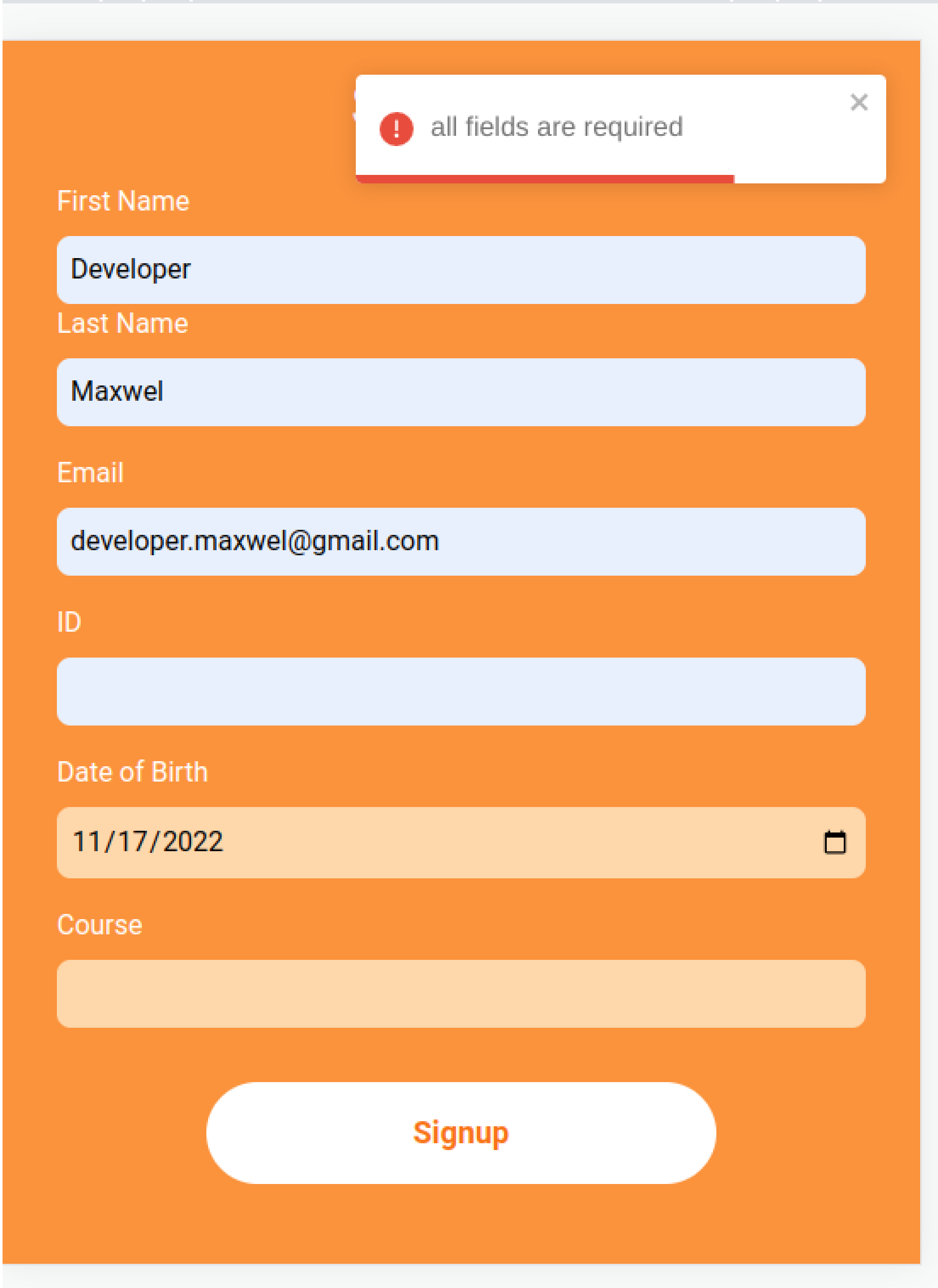


```
function handleSubmit(e) {
  e.preventDefault();
  if (
    !state.firstName ||
    !state.lastName ||
    !state.course ||
    !state.dateofBirth ||
    !state.userId ||
    !state.email
  ) {
    return toast.error("all fields are required");
  }
  console.log("values:", state);
  toast.success("details captured successfully");
  return setState({
    ...state,
    firstName: "",
    lastName: "",
    email: "",
    dateofBirth: "",
    course: "",
    userId: "",
  });
}
```

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

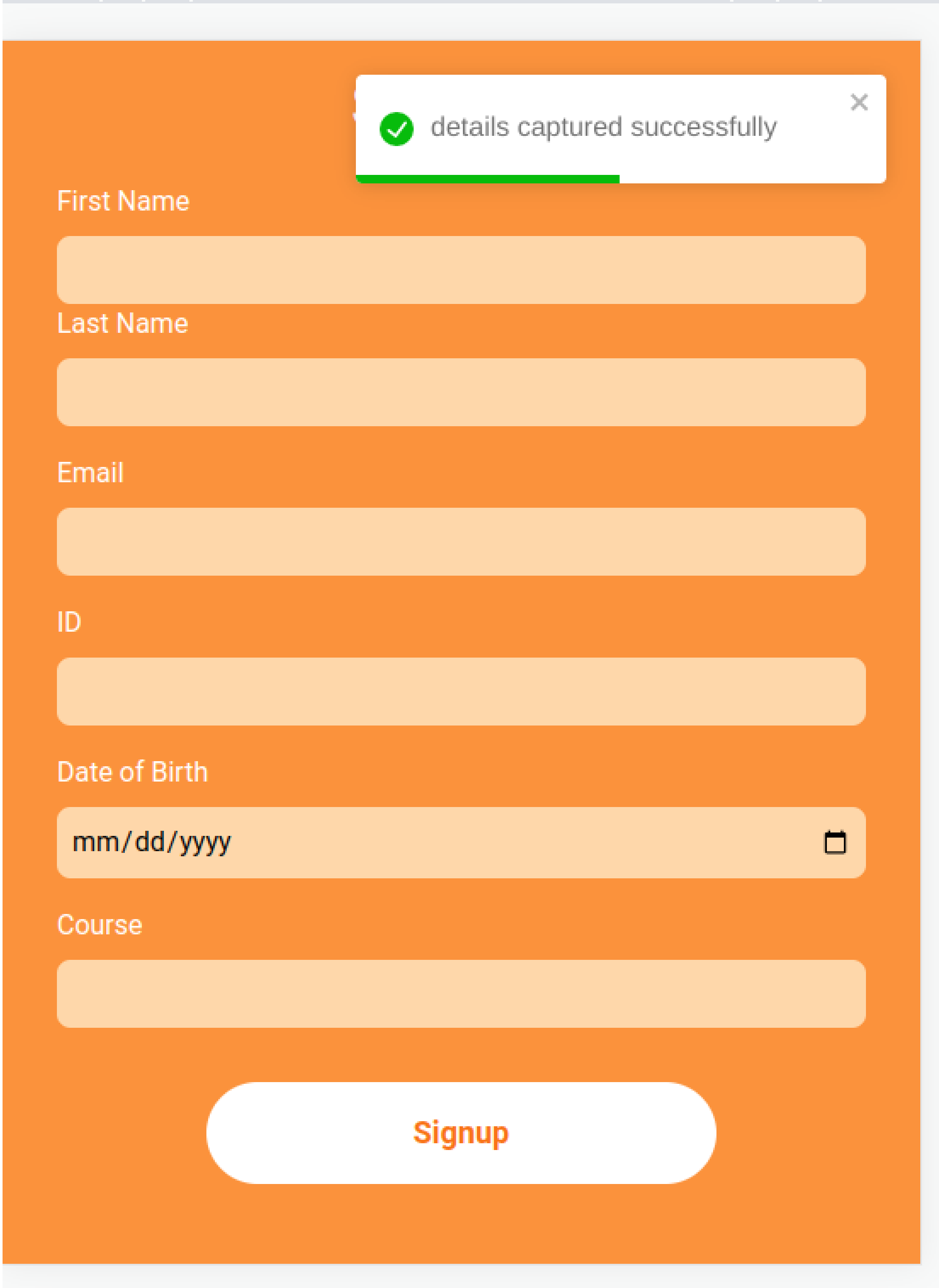
Date of Birth

11/17/2022

Course

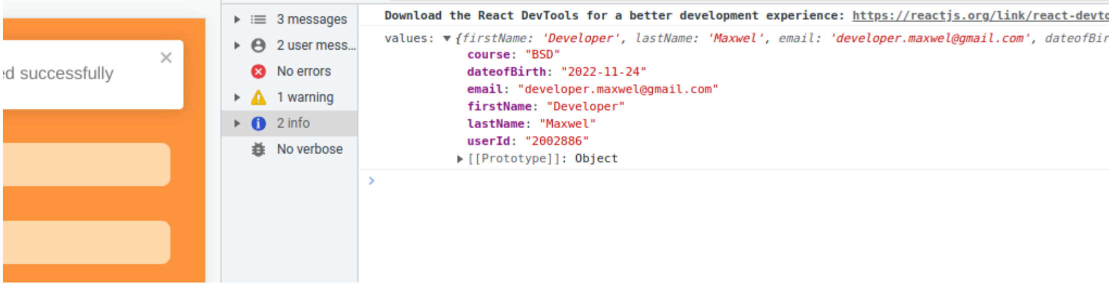
I left two fields which are of course and ID, and toast returned me that error (“all fields are required”). This means that you can ensure that no null value is sent back to the database.

2. **Ensuring all fields are corrects and submitting**



details captured successfully

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



```
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtc
values: ▼{firstName: 'Developer', lastName: 'Maxwel', email: 'developer.maxwel@gmail.com', dateofBir
            course: "BSD"
            dateofBirth: "2022-11-24"
            email: "developer.maxwel@gmail.com"
            firstName: "Developer"
            lastName: "Maxwel"
            userId: "2002886"
          ▶[[Prototype]]: Object
>
```