

```
import React, { useContext, useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import { AppContext } from "../context/AppContext";
import { registerUser } from "../proxies/backend_api";
import { emailRegex } from "../utils/helper";

const SignUp = () => {
  const { setShowAlert, setUser } = useContext(AppContext);

  const navigate = useNavigate();

  const [inputs, setInputs] = useState({
    name: "",
    email: "",
    phone_number: "",
    password: "",
    confirm_password: "",
  });

  const [error, setErrors] = useState({
    name: "",
    email: "",
    phone_number: "",
    password: "",
    confirm_password: "",
  });

  const handleChange = ({ target: { name, value } }) => {
    setErrors((prev) => {
      return { ...prev, [name]: "" };
    });
  };
}
```

```
setInputs((prev) => ({ ...prev, [name]: value }));  
};
```

```
const checkInputErrors = () => {  
  let status = true;  
  if (inputs.email.trim() === "" || !emailRegex.test(inputs.email.trim())) {  
    setErrors((prev) => {  
      return { ...prev, email: "Enter a valid email" };  
    });  
    status = false;  
  }  
}
```

```
if (inputs.name.trim() === "") {  
  setErrors((prev) => {  
    return { ...prev, name: "Enter a valid name" };  
  });  
  status = false;  
}
```

```
if (inputs.phone_number.trim() === "") {  
  setErrors((prev) => {  
    return { ...prev, phone_number: "Enter a valid phone number" };  
  });  
  status = false;  
}
```

```
if (inputs.confirm_password.trim() === "") {  
  setErrors((prev) => {  
    return { ...prev, confirm_password: "Enter a valid password" };  
  });  
  status = false;  
}
```

```
}
```

```
if (inputs.password.trim() === "") {  
  setErrors((prev) => {  
    return { ...prev, password: "Enter a valid password" };  
  });  
  status = false;  
}
```

```
if (inputs.password.trim().length < 6) {  
  setErrors((prev) => {  
    return { ...prev, password: "Minimum 6 characters" };  
  });  
  status = false;  
}
```

```
if (inputs.password.trim() !== inputs.confirm_password.trim()) {  
  setErrors((prev) => {  
    return { ...prev, confirmPassword: "Password don't match" };  
  });  
  status = false;  
}  
return status;  
};
```

```
const handleSignUp = async () => {  
  if (checkInputErrors()) {  
    const data = await registerUser(inputs);  
    if (data.error) {  
      setShowAlert({ type: "error", message: data.error, duration: 3000 });  
    }  
    return;  
  }  
}
```

```

    }
    setUser(data);
    setShowAlert({
      type: "success",
      message: `Your journey starts here ${data.name}`,
      duration: 3000,
    });
    localStorage.setItem("user", JSON.stringify(data));
    navigate("/profile");
  }
};

return (
  <div className="flex flex-col justify-center items-center gap-10 mt-5">
    <div>
      <button className="bg-base-300 rounded-box flex flex-row justify-evenly items-center gap-10 px-10 py-5 w-fit mx-auto">
        <span>Sign in with Github</span>
        <img src={`github-dark.png`} alt="github" width="14%" />
      </button>
      <div className="divider max-w-xs">or</div>
      <div className="card bg-base-300 rounded-box flex flex-col justify-center items-center gap-3 px-10 py-5 w-fit mx-auto">
        <div>
          <input
            value={inputs.name}
            type="text"
            name="name"
            placeholder="name"
            className="input input-bordered input-primary w-full"
            onChange={handleChange}
          />

```

```

{error.name !== "" && (
  <p className="text-sm text-red-500 font-medium">{error.name}</p>
)}
</div>
<div>
  <input
    value={inputs.email}
    type="text"
    name="email"
    placeholder="email"
    className="input input-bordered input-primary w-full"
    onChange={handleChange}
  />
  {error.email !== "" && (
    <p className="text-sm text-red-500 font-medium">{error.email}</p>
  )}
</div>
<div>
  <input
    value={inputs.phone_number}
    type="text"
    name="phone_number"
    placeholder="phone number"
    className="input input-bordered input-primary w-full"
    onChange={handleChange}
  />
  {error.phone_number !== "" && (
    <p className="text-sm text-red-500 font-medium">
      {error.phone_number}
    </p>
  )}

```

```
</div>
```

```
<div>
```

```
  <input
```

```
    value={inputs.password}
```

```
    type="password"
```

```
    name="password"
```

```
    placeholder="password"
```

```
    className="input input-bordered input-primary w-full"
```

```
    onChange={handleChange}
```

```
{error.password !== "" && (
```

```
  <p className="text-sm text-red-500 font-medium">
```

```
    {error.password}
```

```
  </p>
```

```
  )}
```

```
</div>
```

```
<div>
```

```
  <input
```

```
    value={inputs.confirm_password}
```

```
    type="password"
```

```
    name="confirm_password"
```

```
    placeholder="confirm password"
```

```
    className="input input-bordered input-primary w-full"
```

```
    onChange={handleChange}
```

```
{error.confirm_password !== "" && (
```

```
  <p className="text-sm text-red-500 font-medium">
```

```
    {error.confirm_password}
```

```
  </p>
```

```
  )}
```

```
</div>
```

```
<div className="text-center">

  <button
    onClick={handleSignUp}
    className="btn btn-sm btn-primary mb-4"
  >
    Sign Up
  </button>

  <p>
    Already have an account?{" "}
    <Link className="text-blue-400" to="/">
      Sign in
    </Link>
  </p>
</div>

</div>

</div>

</div>

);

};

export default SignUp;
```