Create UI to Interact with Application

Team ID	PNT2022TMID04959
Project	Skill/Job Recommender Application

AIM:

To create UI to interact with application

SOLUTION:

Login.jsx:

```
import { useToast } from "@chakra-ui/react";
import React, { useContext, useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";
import { AppContext } from "../context/AppContext";
import { loginUser } from "../proxies/backend_api";
import { emailRegex } from "../utils/helper";

const Login = () => {
  const toast = useToast();
  const { setUser } = useContext(AppContext);

  const navigate = useNavigate();

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

  const [error, setErrors] = useState({
    email: "",
  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.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;
  }
  return status;
};

const handleLogin = async () => {
  if (checkInputErrors()) {
    const data = await loginUser(inputs);
    if (data.error) {
      toast({
        title: data.error,
        status: "error",
        duration: 3000,
        isClosable: true,
        variant: "left-accent",
        position: "top",
      });
      return;
    }
    setUser(data);
    toast({
      title: `Welcome back ${data.name}`,
      status: "success",
      duration: 3000,
      isClosable: true,
      variant: "left-accent",
        position: "top",
      });
      localStorage.setItem("user", JSON.stringify(data));
      navigate("/dashboard");
    }
  };

  return (
    <>
      <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>
        <form
          onSubmit={(e) => e.preventDefault()}
          className="card bg-base-300 rounded-box flex flex-col justify-center items-center gap-5 px-10 py-5 w-fit mx-auto"
        >
          <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 mt-1 font-medium">
      {error.email}
    </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 mt-1 font-medium">
      {error.password}
    </p>
  )}
</div>
<div className="text-center">
  <button
    type="submit"
    onClick={handleLogin}
    className="btn btn-sm btn-primary mb-4"
  >
```

```
              Login
            </button>
          </div>
        </form>
      </div>
    </>
  );
};
export default Login;
```

Signup.jsx:

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

const SignUp = () => {
  const toast = useToast();

  const { 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) {
      toast({
        title: data.error,
        status: "error",
        duration: 3000,
        isClosable: true,
        variant: "left-accent",
        position: "top",
      });
      return;
    }
    setUser(data);
    toast({
      title: `Your journey starts here ${data.name}`,
      status: "success",
      duration: 3000,
      isClosable: true,
      variant: "left-accent",
      position: "top",
    });
    localStorage.setItem("user", JSON.stringify(data));
    navigate("/dashboard");
  }
};

return (
  <>
    <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>
      <form
        onSubmit={(e) => {
          e.preventDefault();
          handleSignUp();
        }}
        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="number"
    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
      unclicks={handleSignUp}
      className="btn btn-sm btn-primary mb-4"
    >
      Sign Up
    </button>
  </div>
</form>
```

```
      </div>
    </>
  );
};

export default SignUp;
```

RESULT:

Thus UI is created and integrated with the application.