Learner Details

- **Name:Nikhil k**
- **Enrollment Number: su625mr004**
- **Batch / Class: Mern Stack**
- **Assignment: role–based navigation system**
- **Date of Submission: 05-08-2025**

Problem Solving Activity 1.1

1. Program Statement

Create a **role-based navigation system** using React and React Router. Based on the username entered (Admin, Mentor, or Student), redirect the user to the appropriate component page. Each role page should include simple navigation and identify its role clearly..

---

2. Algorithm

- Create Login.jsx with input fields for username and password.
- On clicking the login button, check if the entered username matches "admin", "mentor", or "student" (case-insensitive).
- Use useNavigate() to redirect to the matching route.
- Create three components: Admin, Mentor, and Student.
- Add basic role-specific navigation links using NavLink.
- Handle unmatched input with an alert.

---

3. Pseudocode

```
START
  DISPLAY login form
  ON click of login button:
    GET entered username
    CONVERT to lowercase and trim
    IF username is "admin" THEN navigate to "/admin"
    ELSE IF username is "mentor" THEN navigate to "/mentor"
    ELSE IF username is "student" THEN navigate to "/student"
    ELSE SHOW alert "Invalid user"
  IN each role page:
    DISPLAY role-specific heading
    SHOW navigation links using NavLink
END
```

---

4. Program Code

```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";

const Login = () => {
  const [User, setUser] = useState("");
  const [Password, setPassword] = useState("");
  const navigate = useNavigate();

  const handleClick = () => {
    const user = User.trim().toLowerCase();

    if (user === "admin") {
      navigate("/admin");
    } else if (user === "mentor") {
      navigate("/mentor");
    } else if (user === "student") {
      navigate("/student");
    } else {
      alert("Invalid user. Please type Admin, Mentor, or Student.");
    }
  };

  return (
    <div className="flex items-center justify-center min-h-screen bg-gradient-to-br from-purple-
100 to-lime-100">
      <div className="w-full max-w-md p-8 bg-white shadow-xl rounded-xl border border-
purple-300">
        <h2 className="text-3xl font-bold text-center text-purple-700 mb-6">
          Login Page
        </h2>

        <div className="mb-4">
          <label className="block text-left font-medium text-gray-700 mb-1">Username</label>
          <input
            type="text"
            placeholder="Enter username"
            onChange={(e) => setUser(e.target.value)}
            className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none
focus:ring-2 focus:ring-purple-400 bg-pink-100"
          />
        </div>

        <div className="mb-6">
          <label className="block text-left font-medium text-gray-700 mb-1">Password</label>
          <input
            type="password"
            placeholder="Password not required"
            onChange={(e) => setPassword(e.target.value)}
            className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-purple-400 bg-pink-100"
          />
        </div>


        <button
          type="submit"
          onClick={handleClick}
          className="w-full py-2 bg-green-200 text-purple-700 font-semibold rounded-lg border-2 border-purple-400 hover:bg-green-300 transition"
        >
          Login
        </button>
      </div>
    </div>
  );
};

export default Login;
```

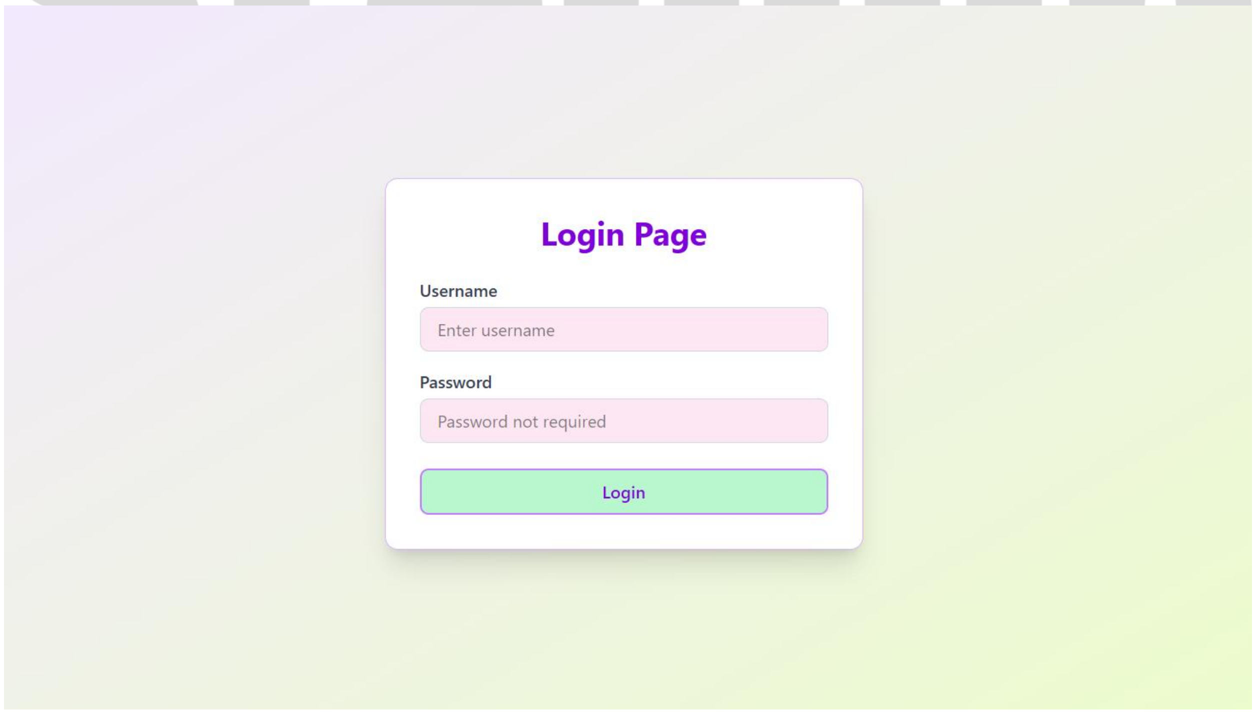
---

5. Test Cases

Input	Expected Behavior
Admin	Redirects to /admin and shows admin page
mentor	Redirects to /mentor and shows mentor page
Student	Redirects to /student and shows student page
developer	Shows alert: "Invalid user. Please type Admin..."
(empty input)	Shows alert: "Invalid user..."

---

6. Screenshots of Output



7. Observation / Reflection

- I learned how to use `useNavigate()` for routing and `NavLink` for styled navigation.
- Handling case-insensitive input using `.trim().toLowerCase()` was useful to avoid errors.
- This task helped me understand how routing can be dynamically controlled in React based on user input.