

frontend/src/components/pages/Register.jsx

```
1
2 import React, { useContext, useState } from "react";
3 import { Link, Navigate, useNavigate } from "react-router-dom";
4 import { Context } from "../../main";
5 import axios from "axios";
6 import toast from "react-hot-toast";
7 import { ClipLoader } from 'react-spinners'; // Assuming you're using react-spinners for a nice loader
8
9 import { FaEye, FaEyeSlash, FaCheck, FaTimes } from 'react-icons/fa';
10
11
12 const Register = () => {
13   const [name, setName] = useState("");
14   const [email, setEmail] = useState("");
15   const [phone, setPhone] = useState("");
16   const [password, setPassword] = useState("");
17   const [passwordErrors, setPasswordErrors] = useState({
18     hasUpper: false,
19     hasNumber: false,
20     hasSpecial: false
21   }); // Track password requirement errors
22   const [confirmPassword, setConfirmPassword] = useState(""); // New state for confirm password
23   const [showPassword, setShowPassword] = useState(false); // State to toggle password visibility
24   const [role, setRole] = useState("");
25   const [education, setEducation] = useState("");
26   const [avatar, setAvatar] = useState(null);
27   const [avatarPreview, setAvatarPreview] = useState("");
28   const [loading, setLoading] = useState(false);
29   const [passwordTouched, setPasswordTouched] = useState(false); // State to track if password has been touched
30
31
32   const { mode, isAuthenticated } = useContext(Context);
33   const navigateTo = useNavigate();
34
35   const changeAvatarHandler = (e) => {
36     const file = e.target.files[0];
37     if (file) {
38       const reader = new FileReader();
```

```
39     reader.readAsDataURL(file);
40     reader.onload = () => {
41         setAvatarPreview(reader.result);
42         setAvatar(file);
43     };
44 }
45 };
46
47 const validatePassword = (value) => {
48     const errors = {
49         hasUpper: /[A-Z]/.test(value),
50         hasNumber: /[0-9]/.test(value),
51         hasSpecial: /[!@#$$%^&*]/.test(value)
52     };
53     setPasswordErrors(errors);
54     setPassword(value);
55     if (!passwordTouched) setPasswordTouched(true); // Set password touched to true when typing starts
56 };
57
58
59
60 const handleRegister = async (e) => {
61     e.preventDefault();
62     if (password !== confirmPassword) {
63         toast.error("Passwords do not match! Please enter the same password.");
64         return;
65     }
66     if (!passwordErrors.hasUpper || !passwordErrors.hasNumber || !passwordErrors.hasSpecial) {
67         toast.error("Please make sure your password meets all the requirements.");
68         return;
69     }
70     setLoading(true);
71
72     const formData = new FormData();
73     formData.append("name", name);
74     formData.append("email", email);
75     formData.append("phone", phone);
76     formData.append("password", password);
77     formData.append("education", education);
78     formData.append("role", role);
79     formData.append("avatar", avatar);
```

```
80
81   try {
82     const { data } = await axios.post(
83       "http://localhost:3000/api/v1/user/register",
84       formData,
85       {
86         withCredentials: true,
87         headers: {
88           "Content-Type": "multipart/form-data",
89         },
90       }
91     );
92     toast.success(data.message);
93     navigateTo("/login");
94   } catch (error) {
95     toast.error(error.response ? error.response.data.message : "Error registering user");
96   } finally {
97     setLoading(false);
98   }
99 };
100
101 if (isAuthenticated) {
102   return <Navigate to="/" />;
103
104 }
105
106 const allConditionsMet = passwordErrors.hasUpper && passwordErrors.hasNumber && passwordErrors.hasSpecial;
107
108 const sectionStyle = {
109   backgroundImage: `url('p.jpg')`, // Adjust the path as necessary
110   backgroundSize: 'cover',
111   backgroundPosition: 'center',
112   backdropFilter: 'blur(0.8%)'
113 };
114
115 return (
116   <article className={mode === "dark" ? "dark-bg" : "light-bg"} style={sectionStyle}>
117     <section className="auth-form">
118       <form onSubmit={handleRegister}>
119         <h1>REGISTER</h1>
120         <select value={role} onChange={e => setRole(e.target.value)} disabled={loading}>
```

```
121     <option value="">SELECT ROLE</option>
122     <option value="Reader">READER</option>
123     <option value="Author">AUTHOR</option>
124 </select>
125 <input
126     type="text"
127     placeholder="Your Name"
128     value={name}
129     onChange={(e) => setName(e.target.value)}
130     disabled={loading}
131 />
132 <input
133     type="email"
134     placeholder="Your Email"
135     value={email}
136     onChange={(e) => setEmail(e.target.value)}
137     disabled={loading}
138 />
139 <input
140     type="number"
141     placeholder="Phone Number"
142     value={phone}
143     onChange={(e) => setPhone(e.target.value)}
144     disabled={loading}
145 />
146 <div style={{ position: 'relative' }}>
147     <input
148         type={showPassword ? "text" : "password"}
149         placeholder="Password"
150         value={password}
151         onChange={(e) => validatePassword(e.target.value)}
152         disabled={loading}
153         style={{ paddingRight: '30px' }} // Add padding to prevent text overlap
154     />
155
156     <span
157         style={{ position: 'absolute', right: '10px', top: '50%', transform: 'translateY(-50%)', cursor: 'pointer' }}
158         onClick={() => setShowPassword(!showPassword)}
159     >
160         {showPassword ? <FaEyeSlash /> : <FaEye />}
161     </span>
```

```
162
163     {passwordErrors.length > 0 && (
164       <div style={{ color: 'red', marginTop: '5px' }}>
165         Missing: {passwordErrors.join(', ')}
166       </div>
167     )}
168
169     </div>
170 {passwordTouched && !allConditionsMet && (
171   <ul style={{ listStyleType: 'none', padding: 0, margin: '10px 0' }}>
172     <li style={{ color: passwordErrors.hasUpper ? 'green' : 'red' }}>
173       {passwordErrors.hasUpper ? <FaCheck /> : <FaTimes />} At least one uppercase letter
174     </li>
175     <li style={{ color: passwordErrors.hasNumber ? 'green' : 'red' }}>
176       {passwordErrors.hasNumber ? <FaCheck /> : <FaTimes />} At least one number
177     </li>
178     <li style={{ color: passwordErrors.hasSpecial ? 'green' : 'red' }}>
179       {passwordErrors.hasSpecial ? <FaCheck /> : <FaTimes />} At least one special character
180     </li>
181   </ul>
182 )}
183
184 <input
185   type={showPassword ? "text" : "password"}
186   placeholder="Confirm Password"
187   value={confirmPassword}
188   onChange={(e) => setConfirmPassword(e.target.value)}
189   disabled={loading}
190 />
191
192
193 <select
194   value={education}
195   onChange={(e) => setEducation(e.target.value)}
196   disabled={loading}
197 >
198   <option value="">SELECT YOUR EDUCATION</option>
199   <option value="Matric">Matric</option>
200   <option value="Intermediate">Intermediate</option>
201   <option value="Graduation">Graduation</option>
202   <option value="Masters">Masters</option>
```

```
203     <option value="PhD">PhD</option>
204   </select>
205   <div style={{ display: "flex", flexDirection: "row", alignItems: "center" }}>
206     <div className="avatar">
207       <img
208         src={avatarPreview || "/pic.jpg"}
209         alt="avatar"
210       />
211     </div>
212     <input
213       type="file"
214       onChange={changeAvatarHandler}
215       className="avatar_input_tag"
216       style={{ border: "none" }}
217       disabled={loading}
218     />
219   </div>
220   <p>
221     Already Registered? <Link to="/login">Login Now</Link>
222   </p>
223   <button className="submit-btn" type="submit" disabled={loading}>
224     {loading ? <ClipLoader size={20} color="#fff" /> : 'REGISTER'}
225   </button>
226   <p>
227     Already Registered? <Link to="/login">Login Now</Link>
228   </p>
229 </form>
230 {/* <div className="banner">
231   
232 </div> */}
233 </section>
234 </article>
235 );
236 };
237
238 export default Register;
239
```