

Code

main

Go to file

- > .vscode
- > public
- > src
- > Context
- > Dashboard
- > Pages
- > AllAuthors
- > AllBooks
- > AllBooksDetails
- > Authentication
- Loginjs
- Registerjs
- > Author
- > CategoryDetails
- > Contact
- > ErrorPage
- > Home
- > PublicationDetails
- > Subscribe
- > assets
- > components
- > BookingForm
- > PricingPlan
- > SearchComponent
- Documentation • Share feedback

kitabkhazanah-client-v1 / src / Pages / Authentication / Registerjs

Murad501 user reads function implemented and many more

085ce72 · 2 weeks ago

History

Code Blame 314 lines (303 loc) · 12.3 KB

Raw

```
1 import React, { useState, useContext, useEffect } from "react";
2 import { useNavigate } from "react-router-dom";
3 import { useForm } from "react-hook-form";
4 import { fetchUrls } from "../../components/Shared/FetchUrls";
5 import axios from "axios";
6 import {
7   createUserWithEmailAndPassword,
8   signInWithEmail,
9   updateProfile,
10 } from "react-firebase-hooks/auth";
11 import auth from "../../firebaseInit";
12 import { toast } from "react-hot-toast";
13 import { dataContext } from "../../App";
14 import { AiOutlineEyeInvisible } from "react-icons/ai";
15 import { FcGoogle } from "react-icons/fc";
16 import LoadingButton from "../../components/Shared/LoadingButton";
17
18 const Register = () => {
19   const navigate = useNavigate();
20   const [showPassword, setShowPassword] = useState(false);
21   const [isLoading, setIsLoading] = useState(false);
22   const {
23     handleSubmit,
24     register,
25
26     formState: { errors },
27   } = useForm();
28   const [createUserWithEmailAndPassword, user, loading, error] =
29     createUserWithEmailAndPassword(auth, {
30       sendEmailVerification: true,
31     });
32   const [updateProfile] = updateProfile(auth);
33
34   // signup with google provider
35   const [signInWithEmail, googleError] = signInWithEmail(auth);
36   const value = useContext(dataContext);
37   const { setupdate } = value;
38
39   const onSubmit = async (data) => {
40     setIsLoading(true);
41     await createUserWithEmailAndPassword(data.email, data.password);
42     await updateProfile({
43       displayName: data.name,
44       phoneNumber: data.phone,
45     });
46     const { name, email, password } = data;
47     axios
48       .post(fetchUrls("auth/register"), {
49         username: name,
50         email,
51         password,
52         subscription: null,
53       })
54       .then((res) => {
55         if (res.status === 200) {
56           navigate("/");
57           setIsLoading(false);
58           toast.success("User register Successfully!");
59           // setupdate(Math.random());
60           localStorage.setItem("user", JSON.stringify(res.data?.user));
61           localStorage.setItem("token", JSON.stringify(res.data?.token));
62         } else {
63           toast.error("User Already Exist!");
64           setIsLoading(false);
65         }
66       })
67       .catch((err) => {});
68   };
69
70   // signup in google
71   const handleGoogleSignUp = async () => {
72     try {
73       const { user } = await signInWithEmail();
74       if (user) {
75         axios
76           .post(fetchUrls("auth/register"), {
77             username: user?.displayName,
78             email: user?.email,
79             password: user?.uid,
80             subscription: null,
81           })
82           .then((res) => {
83             if (res.status === 200) {
84               navigate("/");
85               toast.success("User register Successfully!");
86               localStorage.setItem(
87                 "user",
88                 JSON.stringify({
89                   ...res.data?.user,
90                   photo: user?.photoURL,
91                 })
89               );
90               localStorage.setItem("token", JSON.stringify(res.data?.token));
91               setupdate(Math.random());
92               // localStorage.setItem("token", JSON.stringify(user?.user?.accessToken));
93             } else {
94               navigate("/");
95               setupdate(Math.random());
96               localStorage.setItem(
97                 "user",
98                 JSON.stringify({
99                   ...res.data?.user,
100                   photo: user?.photoURL,
101                 })
102               );
103               localStorage.setItem("token", JSON.stringify(res.data?.token));
104               toast.success("Google Login Successfully!");
105             }
106           })
107           .catch((err) => {
108             // toast.error(err?.response?.data?.message)
109           });
110       }
111     } catch (err) {
112       // toast.error(err?.response?.data?.message)
113     }
114   };
115 }
```

Symbols

Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

Filter symbols

func Register

func onSubmit

func handleGoogleSignUp

```

113     }
114   } catch (error) {
115     if (googleError && error.message) {
116       toast.error(googleError?.message);
117     }
118   }
119 };
120
121 useEffect(() => {
122   if (loading) {
123     setIsLoading(true);
124   } else {
125     setIsLoading(false);
126   }
127 }, [loading]);
128
129 return (
130   <div>
131     <div className="bg-gradient-to-tl from-green-400 to-indigo-900 w-full py-10 px-4 min-h-screen">
132       <div className="flex flex-col items-center justify-center">
133         <div className="bg-white shadow rounded xl:w-1/3 lg:w-1/2 md:w-1/2 w-full p-10 mt-5">
134           <p>
135             <button
136               tabIndex={0}
137               aria-label="register now"
138               className="text-2xl font-extrabold leading-6 text-gray-800"
139             >
140               Register Now
141             </button>
142             <p className="text-sm mt-4 font-medium leading-none text-gray-500">
143               Already have an account?{" "}
144               <button
145                 onClick={() => navigate("/login")}
146                 tabIndex={0}
147                 role="link"
148                 aria-label="Sign up here"
149                 className="text-sm font-medium leading-none underline text-gray-800 cursor-pointer hover:text-primary"
150               >
151                 Login here
152               </button>
153             </p>
154             <form onSubmit={handleSubmit(onSubmit)}>
155               <div className="mt-4">
156                 <label>
157                   <input
158                     type="text"
159                     id="name"
160                     className="bg-gray-200 border rounded focus:outline-none text-xs font-medium leading-none text-gray-800 py-3 w-full pl-3 mt-2"
161                     {...register("name", {
162                       required: {
163                         value: true,
164                         message: "Name is Required",
165                       },
166                     })}
167                   />
168                 <label className="label">
169                   {errors.name?.type === "required" && (
170                     <span className="label-text-alt text-red-500">
171                       {errors.name.message}
172                     </span>
173                   )}
174                 </label>
175               </div>
176               <div className="mt-4">
177                 <label>
178                   <input
179                     type="text"
180                     id="email"
181                     className="bg-gray-200 border rounded focus:outline-none text-xs font-medium leading-none text-gray-800 py-3 w-full pl-3 mt-2"
182                     {...register("email", {
183                       required: {
184                         value: true,
185                         message: "Email is Required",
186                       },
187                       pattern: {
188                         value: /^[a-z0-9]+@[a-z]+\.[a-z]{2,3}/,
189                         message: "Provide a valid Email",
190                       },
191                     })}
192                   />
193                 <label className="label">
194                   {errors.email?.type === "required" && (
195                     <span className="label-text-alt text-red-500">
196                       {errors.email.message}
197                     </span>
198                   )}
199                 </label>
200               </div>
201               <div>
202                 <label>
203                   <input
204                     type="password"
205                     id="password"
206                     className="bg-gray-200 border rounded focus:outline-none text-xs font-medium leading-none text-gray-800 py-3 w-full pl-3 mt-2"
207                     {...register("password", {
208                       required: {
209                         value: true,
210                         message: "Password is Required",
211                       },
212                       minLength: {
213                         value: 6,
214                         message: "Must be 6 characters or longer",
215                       },
216                     })}
217                   />
218                 <label>
219                   {errors.password?.type === "required" && (
220                     <span className="label-text-alt text-red-500">
221                       {errors.password.message}
222                     </span>
223                   )}
224                 </label>
225               </div>
226             </form>
227           </div>
228         </div>
229       </div>
230     </div>
231   );
232 }

```

```

239     <div
240       onClick={() => setShowPassword(!showPassword)}
241       className="absolute right-0 mt-2 mr-3 cursor-pointer"
242     >
243       {showPassword === true ? (
244         <svg
245           width={16}
246           height={16}
247           viewBox="0 0 16 16"
248           fill="none"
249           xmlns="http://www.w3.org/2000/svg"
250         >
251           <path
252             d="M7.99978 2C11.5944 2 14.5851 4.58667 15.2124 8C14.5858 11.4133 11.5944 14 7.99978 14C4.40511 14 1.41444 11.4133 0.787109 8C1.41378 4.58667 4.40511 2 7.99978 2"
253             fill="#71717A"
254           />
255         </svg>
256       ) : (
257         <AiOutlineEyeInvisible />
258       )}
259     </div>
260   </div>
261   <label className="label">
262     {errors.password?.type === "required" && (
263       <span className="label-text-alt text-red-500">
264         {errors.password.message}
265       </span>
266     )}
267     {errors.password?.type === "minLength" && (
268       <span className="label-text-alt text-red-500">
269         {errors.password.message}
270       </span>
271     )}
272   </label>
273 </div>
274 <div className="mt-8">
275   {isLoading ? (
276     <button
277       type="submit"
278       aria-label="create my account"
279       className="focus:ring-indigo-700 text-sm font-semibold leading-none text-white focus:outline-none bg-indigo-700 border rounded hover:bg-indigo-600 py-4"
280     >
281       Register
282     </button>
283   ) : (
284     <LoadingButton customClass={"w-full py-3"}></LoadingButton>
285   )}
286 </div>
287 </form>
288
289 <div className="w-full flex items-center justify-between py-5">
290   <hr className="w-full bg-gray-400" />
291   <p className="text-base font-medium leading-4 px-2.5 text-gray-400">
292     OR
293   </p>
294   <hr className="w-full bg-gray-400" />
295 </div>
296
297 <button
298   aria-label="Continue with google"
299   onClick={handleGoogleSignUp}
300   className="focus:outline-none py-3.5 px-2 sm:px-0 border rounded-lg border-gray-700 flex justify-center items-center w-full hover:bg-primary hover:text-white"
301 >
302   <FcGoogle className="w-5 h-5 mr-2 sm:mr-4" />
303   Continue with Google
304   { /* <p className="text-base font-medium ml-4">
305     </p> */ }
306 </button>
307 </div>
308 </div>
309 </div>
310 </div>
311 };
312 };
313
314 export default Register;

```