

w12-p1 switch Register and Login, and log

This screenshot shows the initial state of the application. In VS Code, the `Register_76.js` file is open, displaying the initial state and the `Register` button. The browser at `localhost:3000/register` shows the 'Jobify' login/register page with the 'Register' button highlighted. The Redux DevTools sidebar shows the initial state of the application.

```
client_76 > src > pages > JS Register_76.js > Register_76
```

```
1 // import { is } from "express/lib/request";
2 import { useState, useEffect } from "react";
3 import Wrapper from "../assets/images/wrappers/
  Register_76";
4 import { Logo_76, FormRow_76 } from "../components";
5 import { useAppContext } from "../context/
  appContext_90";
6
7 const initialState = {
8   name: "",
9   email: "",
10  password: "",
11  isMember: true,
12  showAlert: false,
13 };
14
15 const Register_76 = () => {
16   const [values, setValues] = useState(initialState)
17   ;
18   const { showAlert, displayAlert } = useAppContext();
19   const toggleMember = () => {
20     setValues({ ...values, isMember: !values.
      isMember });
21   };
22   const handleChange = (e) => {
23     console.log("e.target", e.target);
24     setValues({ ...values, [e.target.name]: e.
      target.value });
25   };
26
27   return (
28     <div>
29       <FormRow_76
30         type="email"
31         value={values.email}
32         name="email"
33         className="form-input"
34         handleChange={handleChange}
35       />
36       <FormRow_76
37         type="password"
38         value={values.password}
39         name="password"
40         className="form-input"
41         handleChange={handleChange}
42       />
43       <button className="btn btn-block"
44         type="submit">
45         Submit
46       </button>
47       <p>
48         {values.isMember ? 'Not a member
49         yet?' : 'Already a member?'}
50         <button type="button"
51           className="member-btn" onClick={
52             toggleMember}>
53         {values.isMember ? 'Register' :
54         'Login'}
55       </p>
56     </div>
57   );
58 }
```

Browser: `localhost:3000/register`

Redux DevTools:

```
Register_76
  name: ""
  email: ""
  password: ""
  isMember: false
  showAlert: false
  new entry: ""
  AppContext:
    rendered by
      App_76
    Console
    What's New
    Issues
    Highlights from the Chrome 101 update
    Import and export user flows
    as a JSON file
```

This screenshot shows the application after switching to the Login form. In VS Code, the `Register_76.js` file is open, displaying the initial state and the `Login` button. The browser at `localhost:3001/register` shows the 'Jobify' login/register page with the 'Login' button highlighted. The Redux DevTools sidebar shows the initial state of the application.

```
client_76 > src > pages > JS Register_76.js > Register_76
```

```
1 // import { is } from "express/lib/request";
2 import { useState, useEffect } from "react";
3 import Wrapper from "../assets/images/wrappers/
  Register_76";
4 import { Logo_76, FormRow_76 } from "../components";
5 import { useAppContext } from "../context/
  appContext_90";
6
7 const initialState = {
8   name: "",
9   email: "",
10  password: "",
11  isMember: true,
12  showAlert: false,
13 };
14
15 const Register_76 = () => {
16   const [values, setValues] = useState(initialState)
17   ;
18   const { showAlert, displayAlert } = useAppContext();
19   const toggleMember = () => {
20     setValues({ ...values, isMember: !values.
      isMember });
21   };
22   const handleChange = (e) => {
23     console.log("e.target", e.target);
24     setValues({ ...values, [e.target.name]: e.
      target.value });
25   };
26
27   return (
28     <div>
29       <FormRow_76
30         type="email"
31         value={values.email}
32         name="email"
33         className="form-input"
34         handleChange={handleChange}
35       />
36       <FormRow_76
37         type="password"
38         value={values.password}
39         name="password"
40         className="form-input"
41         handleChange={handleChange}
42       />
43       <button className="btn btn-block"
44         type="submit">
45         Submit
46       </button>
47       <p>
48         {values.isMember ? 'Not a member
49         yet?' : 'Already a member?'}
50         <button type="button"
51           className="member-btn" onClick={
52             toggleMember}>
53         {values.isMember ? 'Register' :
54         'Login'}
55       </p>
56     </div>
57   );
58 }
```

Browser: `localhost:3001/register`

Redux DevTools:

```
Register_76
  name: ""
  email: ""
  password: ""
  isMember: true
  showAlert: false
  new entry: ""
  AppContext:
    rendered by
      App_76
    Console
    What's New
    Issues
    Highlights from the Chrome 101 update
    Import and export user flows
    as a JSON file
```

w12-p2-1 test for notFoundMiddleware_xx.js

Visual Studio Code interface showing the development of a Node.js application. The Explorer pane on the left shows the project structure for '1102-WP2-MERN-ID', including files like .env, server_xx.js, error-handler_xx.js, not-found_xx.js, package.json, Register_xx.js, client_xx, db, middleware, node_modules, .gitignore, package-lock.json, and package.json. The main editor displays the code for server_xx.js, which includes database connection, middleware setup, and a GET route. The code is as follows:

```
10 // db and authenticateUser
11 import connectDB_xx from './db/connect_xx.js';
12
13 // middleware
14 import notFoundMiddleware_xx from './middleware/not-found_xx.js';
15 import errorHandlerMiddleware_xx from './middleware/error-handler_xx.js';
16
17 app.get('/', (req, res) => {
18   throw new Error('testing for error');
19   res.send('Welcome htchung 123456789');
20 });
21
22 app.use(notFoundMiddleware_xx);
23 app.use(errorHandlerMiddleware_xx);
24
25 const port = process.env.PORT || 5000;
26
27 const start = async () => {
28   try {
29     await connectDB_xx(process.env.MONGO_LOCAL_URL).then(() => {
30       console.log('Connecting to MongoDB');
31     });
32     app.listen(port, () => console.log('Server is running on port ${port}'));
33   } catch (err) {
34     console.log(err);
35   }
36 }
37
```

The terminal output shows the server starting successfully on port 5000, with an error message: "error Error: testing for error".

The browser window on the right shows the response at localhost:5000/:

```
1 // 20220512194857
2 // http://localhost:5000/
3
4 {
5   "msg": "there wss an error"
6 }
```

Visual Studio Code interface showing the development of a Node.js application. The Explorer pane on the left shows the project structure for '1102-WP2-MERN-ID', including files like .env, server_xx.js, error-handler_xx.js, not-found_xx.js, package.json, Register_xx.js, client_xx, db, middleware, node_modules, .gitignore, package-lock.json, and package.json. The main editor displays the code for not-found_xx.js, which includes a 404 handler:

```
1 const notFoundMiddleware_xx = (req, res) => {
2   res.status(404).send('Route does not exist');
3 };
4
5 export default notFoundMiddleware_xx;
6
```

The browser window on the right shows the response at localhost:5000/abc:

```
Route does not exist
```