

Project Structure

```
mern-hooks-demo
|
|   backend
|   |
|   |   server.js
|   |   models/User.js
|   |   routes/userRoutes.js
|   |
|   frontend
|   |
|   |   src
|   |   |
|   |   |   context/AuthContext.jsx
|   |   |   reducer/authReducer.js
|   |   |   App.jsx
|   |   |   main.jsx
```

BACKEND (Express + MongoDB)

Install backend dependencies

```
cd backend
npm init -y
npm install express mongoose cors
```

server.js

```
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");

const app = express();
app.use(cors());
app.use(express.json());

mongoose.connect("mongodb://127.0.0.1:27017/mern_hooks")
  .then(() => console.log("MongoDB Connected"))
  .catch(err => console.log(err));

app.use("/api/users", require("./routes/userRoutes"));

app.listen(5000, () => {
  console.log("Server running on port 5000");
});
```

models/User.js

```
const mongoose = require("mongoose");
```

```
const UserSchema = new mongoose.Schema({
  name: String,
  email: String
});

module.exports = mongoose.model("User", UserSchema);
```

routes/userRoutes.js

```
const express = require("express");
const router = express.Router();
const User = require("../models/User");

router.post("/", async (req, res) => {
  const user = await User.create(req.body);
  res.json(user);
});

router.get("/", async (req, res) => {
  const users = await User.find();
  res.json(users);
});

module.exports = router;
```

FRONTEND (React + Hooks)

Create React app (Vite)

```
cd frontend
npm create vite@latest
npm install axios
```

useContext + useReducer (Global Auth State)

context/AuthContext.jsx

```
import { createContext, useReducer } from "react";
import { authReducer, initialState } from "../reducer/authReducer";

export const AuthContext = createContext();

export const AuthProvider = ({ children }) => {
  const [state, dispatch] = useReducer(authReducer, initialState);

  return (
    <AuthContext.Provider value={{ state, dispatch }}>
      {children}
    </AuthContext.Provider>
  );
}
```

```
</AuthContext.Provider>
);
};
```

reducer/authReducer.js

```
export const initialState = {
  user: null
};

export const authReducer = (state, action) => {
  switch (action.type) {
    case "LOGIN":
      return { user: action.payload };
    case "LOGOUT":
      return { user: null };
    default:
      return state;
  }
};
```

App.jsx (ALL hooks used here)

App.jsx

```
import { useContext, useRef, useMemo, useCallback, useLayoutEffect } from "react";
import axios from "axios";
import { AuthContext } from "./context/AuthContext";

function App() {
  const { state, dispatch } = useContext(AuthContext);

  const nameRef = useRef();
  const emailRef = useRef();

  // useLayoutEffect
  useLayoutEffect(() => {
    console.log("DOM updated before paint");
  }, []);

  // useCallback
  const addUser = useCallback(async () => {
    const user = {
      name: nameRef.current.value,
      email: emailRef.current.value
    };

    const res = await axios.post("http://localhost:5000/api/users", user);
    dispatch({ type: "LOGIN", payload: res.data });
  }, [nameRef, emailRef]);
}
```

```
}, [dispatch]);

// useMemo
const userStatus = useMemo(() => {
  return state.user ? "Logged In" : "Logged Out";
}, [state.user]);

return (
  <div style={{ padding: "20px" }}>
    <h2>MERN Hooks Demo</h2>

    <input ref={nameRef} placeholder="Name" />
    <br /><br />
    <input ref={emailRef} placeholder="Email" />
    <br /><br />

    <button onClick={addUser}>Save User</button>

    <h3>Status: {userStatus}</h3>

    {state.user && (
      <div>
        <p>Name: {state.user.name}</p>
        <p>Email: {state.user.email}</p>
      </div>
    )}
  </div>
);
}

export default App;
```

main.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import { AuthProvider } from "./context/AuthContext";

ReactDOM.createRoot(document.getElementById("root")).render(
  <AuthProvider>
    <App />
  </AuthProvider>
);
```
