Database.json

[

    {"name": "Jenifer", "work": "Assistant Professor", "password": "12345"},

    {"name": "radha", "work": "Assistant Professor", "password": "radha12"},

    {"name": "jaya", "work": "Assistant Professor", "password": "jaya12"}

  ]

Server.js

const express = require("express");

const fs = require("fs");

const jwt = require("jsonwebtoken");

const app = express();

const port = 3001;

// Middleware to parse JSON

app.use(express.json());

// Read user data from database.json

let users = [];

fs.readFile("database.json", (err, data) => {

    if (err) throw err;

    users = JSON.parse(data);

});

// Middleware to validate JWT

function authenticateToken(req, res, next) {

    const authHeader = req.headers["authorization"];

    if (!authHeader) return res.status(401).json({ message: "Access denied" });

    const token = authHeader.split(" ")[1];

    if (!token) return res.status(401).json({ message: "Access denied" });

    jwt.verify(token, "secret", (err, user) => {

        if (err) {

            console.log("JWT Error:", err); // Print error in console

            return res.status(403).json({ message: "Invalid or Expired Token" });

        }

        req.user = user;

        next();

    });

}

// Login route (POST request)

app.post("/auth", (req, res) => {

    const { name, password } = req.body;

    // Find user in the database

    const user = users.find((u) => u.name === name && u.password === password);

    if (!user) {

        return res.status(401).json({ message: "Invalid credentials" });

    }

    // Generate JWT token

    const token = jwt.sign({ name: user.name, work: user.work }, "secret", { expiresIn: "1h" });

    res.json({ login: true, token });

});

// Protected route (only accessible with valid JWT)

app.get("/protected", authenticateToken, (req, res) => {

    res.json({ message: "You are authorized!", user: req.user });

});

// Start server

app.listen(port, () => {

    console.log(`Server is running on http://localhost:${port}/`);

});