Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Home</title>

</head>

<body>

    <center>

        <h2>Welcome!</h2>

        <a href="/signup.html">Sign Up</a> |

        <a href="/login.html">Login</a>

    </center>

</body>

</html>

Login.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Login</title>

    <script>

        async function loginUser(event) {

            event.preventDefault();

            const name = document.getElementById("username").value;

            const password = document.getElementById("password").value;

            const response = await fetch("/login", {

                method: "POST",

                headers: { "Content-Type": "application/json" },

                body: JSON.stringify({ name, password })

            });

            const data = await response.json();

            if (data.token) {

                localStorage.setItem("token", data.token);

                window.location.href = "/profile.html";

            } else {

                alert("Invalid Credentials");

            }

        }

    </script>

</head>

<body>

    <center>

        <h2>Login</h2>

        <form onsubmit="loginUser(event)">

            <input type="text" id="username" placeholder="Username" required><br>

            <input type="password" id="password" placeholder="Password" required><br>

            <button type="submit">Login</button>

        </form>

    </center>

</body>

</html>

Profile.html

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Welcome</title>

</head>

<body>

    <center><h2>Welcome to your profile!</h2></center>

</body>

</html>

Signup.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Sign Up</title>

    <script>

        async function registerUser(event) {

            event.preventDefault();

            const name = document.getElementById("username").value;

            const password = document.getElementById("password").value;

            const work = document.getElementById("work").value;

            const response = await fetch("/signup", {

                method: "POST",

                headers: { "Content-Type": "application/json" },

                body: JSON.stringify({ name, password, work })

            });

            const data = await response.json();

            alert(data.message);

            window.location.href = "/login.html";

        }

    </script>

</head>

<body>

    <center>

        <h2>Sign Up</h2>

        <form onsubmit="registerUser(event)">

            <input type="text" id="username" placeholder="Name" required><br>

            <input type="text" id="work" placeholder="Profession" required><br>

            <input type="password" id="password" placeholder="Password" required><br>

            <button type="submit">Register</button>

        </form>

    </center>

</body>

</html>

Database.json

[

  {

    "name": "fatima",

    "work": "Assistant Professor",

    "password": "abc"

  },

  {

    "name": "Sarvesh",

    "work": "Assistant Professor",

    "password": "ambala"

  },

  {

    "name": "ram",

    "work": "Assistant Professor",

    "password": "priya"

  },

  {

    "name": "Lakshmi",

    "work": "Asst Prof",

    "password": "Cmrit@123"

  },

  {

    "name": "afshan",

    "password": "abc123",

    "work": "Assistant Professor"

  },

  {

    "name": "sara",

    "password": "abc123",

    "work": "Assistant Professor"

  },

  {

    "name": "don",

    "password": "123456",

    "work": "professor"

  },

  {

    "name": "simran",

    "password": "abc",

    "work": "Assistant Professor"

  },

  {

    "name": "priya",

    "password": "123456789",

    "work": "professor"

  },

  {

    "name": "Jenifer",

    "password": "12345",

    "work": "Asst prof"

  },

  {

    "name": "Jeni",

    "password": "12345",

    "work": "Asst prof"

  }

]

Server.js

// Import required modules

const express = require("express");

const fs = require("fs/promises"); // Async file handling

const jwt = require("jsonwebtoken");

const path = require("path");

const app = express();

const PORT = 3000;

const SECRET\_KEY = "secret"; // JWT signing key

app.use(express.json()); // Middleware to parse JSON

// Serve static files from 'public' folder

app.use(express.static(path.join(\_\_dirname, "public")));

// Read & Write database (database.json)

const loadUsers = async () => {

    try {

        const data = await fs.readFile("database.json", "utf-8");

        return JSON.parse(data);

    } catch (err) {

        console.error("Error reading database.json", err);

        return [];

    }

};

const saveUsers = async (users) => {

    try {

        await fs.writeFile("database.json", JSON.stringify(users, null, 2));

    } catch (err) {

        console.error("Error writing to database.json", err);

    }

};

// Register Route

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

    const { name, password, work } = req.body;

    const users = await loadUsers();

    if (users.some(user => user.name === name)) {

        return res.status(400).json({ error: "User already registered" });

    }

    users.push({ name, password, work });

    await saveUsers(users);

    res.json({ message: "Successfully Registered" });

});

// Login Route

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

    const { name, password } = req.body;

    const users = await loadUsers();

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

    if (!user) {

        return res.status(401).json({ error: "Invalid name or password" });

    }

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

    res.json({ token });

});

// JWT Middleware

const verifyToken = (req, res, next) => {

    const token = req.headers["authorization"]?.split(" ")[1];

    if (!token) return res.status(403).json({ error: "Access denied, token missing" });

    jwt.verify(token, SECRET\_KEY, (err, decoded) => {

        if (err) return res.status(403).json({ error: "Invalid token" });

        req.user = decoded;

        next();

    });

};

// Protected Profile Route

app.get("/profile", verifyToken, (req, res) => {

    res.json({ message: "Welcome to your profile", user: req.user });

});

// Start Server

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));