**🛡️ Authentication-Dashboard**

*A responsive full-stack authentication system with user signup/login, validation, hashed passwords (a full stack vault.,), and a dynamic dashboard. Built with* ***Node.js****,* ***Express****,* ***MongoDB****, and modern* ***HTML5/CSS/JS****.*

**🚀 Features**

* ✅ User Signup & Login Forms
* ✅ MongoDB Database with Mongoose
* ✅ Cloud Hosting via MongoDB Atlas
* ✅ Duplicate Username/Email Check
* ✅ Password Hashing using bcryptjs
* ✅ Dashboard with Categorized quiz & learning resources
* ✅ Responsive design with animations

This project is a full-featured Authentication Dashboard built using HTML, CSS, JavaScript, Node.js, Express.js, and MongoDB (via Mongoose). It features a clean, responsive frontend for user signup and login, with backend validation and secure password hashing using bcrypt. The system checks for duplicate usernames and emails, offers smart suggestions, and securely stores user credentials in a MongoDB Atlas database. After logging in, users are redirected to a dashboard that includes a categorized list of learning and quiz resources. This project demonstrates practical full-stack integration using the MEAN stack technologies.

**Note:** *Angular is not used in this project …,*

🔧 Technologies & Basic Syntax

✅ HTML5

***<!-- Basic form -->***

<form onsubmit="handleSignup(event)">

<input type="text" placeholder="Name" />

<input type="email" placeholder="Email" />

<input type="password" placeholder="Password" />

<button type="submit">Sign Up</button>

</form>

🎨 CSS

***/\* Button style \*/***

.logout-btn {

background: #e74c3c;

color: white;

padding: 0.5rem 1rem;

border-radius: 8px;

cursor: pointer;

transition: background 0.3s;

}

.logout-btn:hover {

background: #c0392b;

}

🧠 JavaScript

***// Handle form submission***

async function handleSignup(event) {

event.preventDefault();

const name = document.querySelector('input[placeholder="Name"]').value;

const email = document.querySelector('input[placeholder="Email"]').value;

const password = document.querySelector('input[placeholder="Password"]').value

const res = await fetch('https://api/users/signup', {

method: 'POST',

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

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

});

const data = await res.json();

alert(data.message);

}

🚀 Node.js + Express.js

const express = require('express');

const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

const cors = require('cors');

const app = express();

app.use(express.json());

app.use(cors());

***// MongoDB Connection***

mongoose.connect(process.env.MONGO\_URI, {

useNewUrlParser: true,

useUnifiedTopology: true

}).then(() => console.log('Connected to MongoDB'));

🍃 MongoDB (with Mongoose)

const mongoose = require('mongoose');

***// Connect to MongoDB Atlas***

mongoose.connect(process.env.MONGO\_URI, {

useNewUrlParser: true,

useUnifiedTopology: true

})

.then(() => console.log('✅ Connected to MongoDB Atlas'))

.catch(err => console.error('❌ MongoDB connection error:', err));

***// Define a User schema***

const userSchema = new mongoose.Schema({

name: String,

email: String,

password: String

});

***// Create a model from the schema***

const User = mongoose.model('User', userSchema);

***// Query example: find user by email***

User.findOne({ email: 'example@email.com' }, (err, user) => {

if (err) throw err;

console.log('User found:', user);

});

🔐 bcryptjs

const bcrypt = require('bcryptjs');

const hashed = bcrypt.hashSync(password, 10);

const isMatch = bcrypt.compareSync(inputPassword, hashed);