Project Title	**Full stack Authentication System**
Technologies	MERN
Domain	Security
Project Level	Medium

Table

Table of Contents

1. Problem Statement:

Design a web application “Full stack Authentication System” to allow users to browse movies and keep track of the movies they have watched and add a rating and review to the movie.

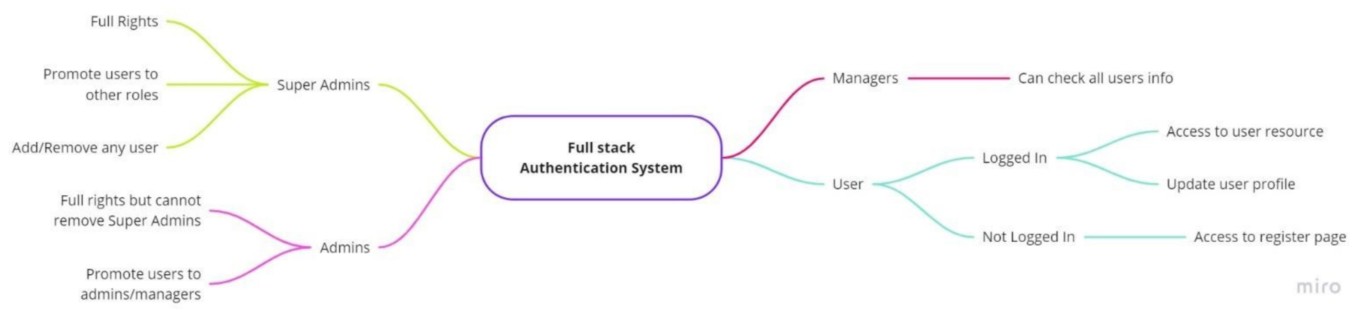
1.1. What is Authentication?

The process of confirming that someone or something is, in fact, who or what it claims to be is known as authentication. By comparing a user's credentials to those stored in a database of authorized users or on a data authentication server, authentication technology controls access to systems. Authentication ensures secure systems, secure business processes, and secure corporate data.

Now that you understand what an authentication System is, let’s discuss some of the functionality of the “**Full stack Authentication System**” that you will design.

1. To create a web-based system that will have a register / login form where users can come and register themselves.
2. After the user registers, the user password must be properly encrypted and a default role of user must be added to the user account.
3. Users should have the ability to add/update their account details during and after registration.
4. There has to be a user dashboard where the users can see their profile.
5. The admins can add/edit/delete any of the users and also change the user role.

6. There has to be different roles with different permissions/level of access (Ex: Super Admin, Admin, manager, Team Leader, User).
7. Each page accessible to only certain role based authenticated users.



Low-Level system flow

1.2. Project Objective

• **Data Safety:** To create a secure authentication system with encryption so that if in case there is a breach of data the user passwords are safe because of encryption.

• **Authorization:** To make sure that the user with proper roles can access certain parts of the application.

1.3. Scope of The Project

1. To build a system which lets users sign up and encrypts their password by default with some encryption algorithm automatically.

2. This authentication system for now is a standalone project but if required with minimal modification one should be able to integrate it in their own project.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

Super Admin:

- Has access to everything.
- Can assign different roles to different users.
- Add/Remove any user **Admin:**

- Has access to everything.
- Cannot remove/change super admin.
- Add/Remove other admins and users.

Manager:

- Has restricted access.
- Cannot remove super admins or admins.

User:

- Has restricted access.
- Cannot remove anyone.

UI:

- There should be a dashboard.
- Frontend should be clean and minimal which is accessible based on the role of the user.

1.4.2. Non-Functional Requirements

1. **Security:** The system should have a certain level of security such that not anybody can have access to sensitive information and the passwords should be properly encrypted.
2. **Robustness:** If the user's system crashes, a backup of the user data must be stored on remote database servers to enable recovery.
3. **Performance:** The application must be lightweight and the UI should be fast and responsive.
4. **Authentication Requirements:** There should be an authentication system in place which checks the credentials and user role each time a user logs in.

1.5. Modules List

Super Admin:

- Login
- Assign Roles o Admin o Manager
- Update/Delete roles and users **Admin:**
- Login
- Assign Roles o Admin o Manager
- Update/Delete other admins and managers
- My Profile
- Logout **Manager:**
- Login
- My Profile
- Logout **User:**
- Login/Register
- Default role: user
- My Profile
- Logout

2. Project Evaluation metrics:

2.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Deploy the application on your preferred service.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You must submit a system architecture design in your wireframe and architecture documents.

2.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.1. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.2. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.3. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link.

The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.