



Project Title	HR Management System
Technologies	MERN
Domain	Industry
Project Level	Medium

Table

Table of Contents

1. Problem Statement:	2
1.1. What is a HR Management System?	2
1.2. Project Objective	3
1.3. Scope of The Project	3
1.4. Functional and Non-Functional Requirements: -	3
1.4.1. Functional Requirements	3
1.4.2. Non-Functional Requirements	3
2. Project Evaluation metrics:	4
2.2. Database:	4
2.3. API Details or User Interface:	4
2.4. Deployment:	4
Deploy the application on your preferred service.....	4
2.5. Solutions Design:.....	4



2.6. System Architecture:	4
2.7. Optimization of solutions:	4
3. Submission requirements:	
<hr/>	
3.1. High-level Document:	5
3.2. Low-level document:	5
3.3. Architecture:	5
3.4. Wireframe:	5
3.5. Project code:	5
3.6. Detail project report:	5
3.7. Project demo video:	5
3.8. The project LinkedIn a post:	5

1. Problem Statement:

Design a web application “HR Management System” to allow any organization to manage employees and everything related to employees with ease.

1.1. What is a HR Management System?

If we do it manually, it will take a lot of time to keep the data of all active and inactive personnel. To carry out both internal and external human resource functions, every organization has its own human resources. The most important responsibility for any HR is managing the human capital, or the employees, out of all the operations related to human resources. The primary responsibility of HR is to hire, manage, and store employee data, which includes personal information about them, such as their job positions, job streams, projects assigned to them, salary, and more, enabling them to handle a heavy workload.

Now that you understand what HR Management System is, let's discuss some of the functionality of the “**HR Management System**” that you will design.

1. To create a web-based system that will help in managing the recruitment of workforce.
2. To help keep track of development related to employees.
3. To provide detailed reports of employment for any specific period.



4. There has to be a dashboard where all the details is visible (Ex: Number of Employees, Employee details, etc.,).
5. A calendar where all the important events can be marked.
6. To help onboard new employees and give them the training materials and training programs.
7. Track employee's leaves, improve the work quality by utilizing the available man force properly and effectively.
8. Be able to manage audits and store audit logs



Low-Level system flow

1.2. Project Objective

- To improve the proficiency and efficiency of the business by automating the manual and repetitive tasks.
- To manage employees, teams effectively to ultimately achieve the organizational goals.

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

- Employee onboarding and administration
- Employee Leave management
- Employee details management



- Employee payroll management
- Events management
- Performance management
- Applicant tracking and recruitment

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 each time a user logs in.

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](#)

3.5. 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](#)



3.6. 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](#)

3.7. Project demo video:

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

3.8. 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.