**Employee Management System (EMS) Documentation**

**1. Introduction**

The **Employee Management System (EMS)** is a comprehensive platform designed to simplify and automate HR operations, including employee management, task assignment, performance tracking, and access control. With a user-friendly interface and a robust backend, the system enhances efficiency in managing employee-related processes within an organization.

**2. Features**

The Employee Management System (EMS) includes the following key modules:

* **Employee Management**: Manage employee details, roles, departments, and their reporting hierarchy.
* **Task Management**: Enable managers to create, assign, and track employee tasks efficiently.
* **Role-Based Access Control**: Restrict access to specific functionalities based on user roles (Admin, Manager, Employee).
* **Employee Authentication**: Secure login system with unique username and password credentials for employees.
* **Data Management**: Add, update, and delete employee records while ensuring data consistency.

**3. Technology Stack**

**Frontend**

* HTML, CSS, JS Bootstrap (for UI design)

**Backend**

* Django (Python)

**Database**

* SQLlite

**Deployment**

* Hosted on a cloud platform

### ****4. System Modules****

#### ****4.1 Employee Management****

**Functionalities:**

* Add, update, and delete employee records.
* Assign roles such as Admin, Manager, Team Leader, and Employee.
* Link employees to their respective departments.
* Define and manage the reporting hierarchy.

**Database Structure:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| employee\_id | int (PK) | Unique Employee ID |
| first\_name | varchar(100) | Employee's First Name |
| last\_name | varchar(100) | Employee's Last Name |
| role\_id | int (FK) | Reference to roles table |
| dept\_id | int (FK) | Reference to department table |
| manager\_id | int (FK) | Reporting Manager ID |
| created\_at | datetime | Timestamp of record creation |

#### ****4.2 Task Management****

**Functionalities:**

* Create and assign tasks to employees.
* Filter tasks by employee, status, or priority.
* Monitor task progress and completion.

**Database Structure:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| task\_id | int (PK) | Unique Task ID |
| title | varchar(100) | Task Title |
| description | varchar(300) | Task Details |
| priority | varchar(50) | Task Priority (High, Medium, Low) |
| assigned\_to | int (FK) | Reference to assigned employee |
| status | varchar(50) | Task Status (Pending, In Progress, Completed) |
| created\_at | datetime | Timestamp of task creation |

#### ****4.3 Performance Management****

**Functionalities:**

* Conduct periodic employee performance reviews.
* Assign ratings and provide feedback.
* Generate performance reports for evaluation.

**Database Structure:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| review\_id | int (PK) | Unique Review ID |
| employee\_id | int (FK) | Employee being reviewed |
| reviewed\_by | int (FK) | Reviewer ID (Manager/Admin) |
| rating | int | Performance Rating (1-10) |
| comments | varchar(300) | Additional feedback |
| created\_at | datetime | Timestamp of review creation |

#### ****4.4 Leave Management****

**Functionalities:**

* Employees can apply for leave and track approval status.
* Manage employee leave quotas.
* Managers/Admins can approve or reject leave requests.

**Database Structure:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| leave\_id | int (PK) | Unique Leave ID |
| employee\_id | int (FK) | Employee applying for leave |
| leave\_type | varchar(50) | Type of Leave (Sick, Casual, etc.) |
| start\_date | date | Leave Start Date |
| end\_date | date | Leave End Date |
| status | varchar(50) | Leave Status (Pending, Approved, Rejected) |
| approved\_by | int (FK) | Manager/Admin ID who approved/rejected |
| created\_at | datetime | Timestamp of leave request |

### ****4.5 User Authentication****

**Functionalities:**

* Secure login system with username and password.
* Role-based access control for different user permissions.
* Password reset functionality for account recovery.

**Database Structure:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| user\_id | int (PK) | Unique User ID |
| username | varchar(100) | Unique Username |
| password | varchar(100) | Encrypted Password |
| role\_id | int (FK) | Assigned User Role |
| created\_at | datetime | Timestamp of user creation |

### ****5. Deployment****

* The application is hosted on a cloud-based platform for high availability and scalability.
* Version control is maintained through a GitHub repository for seamless updates and collaboration.
* The system operates 24/7 on a live server, ensuring uninterrupted access.

### ****6. Conclusion****

The **Employee Management System (EMS)** is a comprehensive solution designed to optimize HR operations by automating employee management, task tracking, and performance evaluation. With its modular architecture and scalability, the system ensures an efficient and user-friendly experience for managing workforce-related activities.

For further details, refer to the **GitHub repository** and **deployment link**.

Github :

<https://github.com/Aartibhosalepatil/Internship_proj>