# Office Employee Management System - Documentation

## 1. Project Overview

The Office Employee Management System is a web-based application designed to efficiently manage employee records. It allows users to add, view, remove, and filter employees based on various criteria. This system simplifies employee data management for HR and managers.

## 2. Technologies Used

Frontend: HTML5, CSS3, Bootstrap 4.3, JavaScript  
Backend: Django (Python Framework)  
Database: SQLite   
Django ORM: Used for managing database operations

## 3. Features

* Home Page: Navigation to all features
* Add Employee: Allows users to enter employee details and save them in the database
* View All Employees: Displays a table of all employee records
* Remove Employee: Allows users to select and delete an employee
* Filter Employees: Enables filtering based on name, department, and role
* User-Friendly Interface: Clean and responsive design using Bootstrap

## 4. Installation Guide

### Step 1: Clone the Repository

git clone https://github.com/your-repo-url.git  
cd Office\_Employee\_Management

### Step 2: Create and Activate Virtual Environment

python -m venv env  
source env/bin/activate # On Windows use: env\Scripts\activate

### Step 3: Install Dependencies

pip install -r requirements.txt

### Step 4: Migrate Database

python manage.py makemigrations  
python manage.py migrate

### Step 5: Run the Server

python manage.py runserver  
Now, open http://127.0.0.1:8000/ in your browser.

## 5. Database Schema

Employee Model (models.py)

class Employee(models.Model):  
 first\_name = models.CharField(max\_length=50)  
 last\_name = models.CharField(max\_length=50)  
 salary = models.IntegerField()  
 bonus = models.IntegerField()  
 phone = models.CharField(max\_length=15)  
 dept = models.ForeignKey(Department, on\_delete=models.CASCADE)  
 role = models.ForeignKey(Role, on\_delete=models.CASCADE)  
 hire\_date = models.DateTimeField(auto\_now\_add=True)

## 6. Application Structure

Office\_Employee\_Management/  
│── employee\_app/ # Django App  
│ ├── templates/ # HTML Templates  
│ ├── static/css/ # Stylesheets (Optional)  
│ ├── models.py # Database Models  
│ ├── views.py # Business Logic  
│ ├── urls.py # URL Routing  
│── manage.py # Django Management Script  
│── db.sqlite3 # Database File (SQLite)

## 7. Usage Guide

- Add Employee: Fill out the form on /add\_emp/ and submit.

- View Employees: Navigate to /all\_emp/ to see all employees.

- Remove Employee: Select an employee from the dropdown on /remove\_emp/ and confirm deletion.

- Filter Employees: Use /filter\_emp/ to search by name, department, or role.

## 8. Future Enhancements

* Add pagination for employee records
* Implement search and sorting functionality
* Improve UI with better dashboards and analytics
* Add user authentication & access control

## 9. Conclusion

The Office Employee Management System simplifies employee record handling and is designed for easy scalability. This documentation provides a complete overview of the project setup, usage, and future improvements.