

DOCTOR MANAGEMENT SYSTEM



**Submitted by
E.MUKESH SUDHAN**

INTRODUCTION

- The Doctor Management System is a web application developed using Python and Django framework. This system is designed to streamline the management of doctors, patients, appointments, and medical records within a healthcare facility.



OBJECTIVES

- The primary objectives of this project were:
- To create a user-friendly interface for managing doctors, patients, and appointments.
- To implement secure authentication and authorization mechanisms.
- To facilitate efficient scheduling and tracking of appointments.
- To enable the storage and retrieval of patient medical records.



TECHNOLOGIES USED

- **Backend Framework:** Django
- **Database:** SQL (SQLite for development)
- **Frontend:** HTML, CSS, JavaScript
- **Authentication:** Django's built-in authentication system
- **Deployment:** Django's development server



SYSTEM FEATURES

1. User Authentication

- Implemented user authentication and authorization using Django's authentication system.
- Users can register, log in, and log out securely.

2. Doctor Management

- Admin can add, view, edit, and delete doctor profiles.
- Each doctor profile includes details like name, specialization, contact information, and schedule.



3. Patient Management

- Admin can manage patient records by adding, viewing, updating, and deleting patient profiles.
- Patient profiles store information such as name, age, gender, contact details, and medical history.

4. Appointment Scheduling

- Users can schedule appointments by selecting a doctor and filling out a form with patient details and preferred time.
- Appointments are stored in the database and can be viewed in a calendar format.



DATABASE DESIGN

- Used SQL database (SQLite for development) to store application data.
- Designed tables to represent doctors, patients, appointments, and medical records with appropriate relationships.



CHALLENGES FACED

- **Database Management:** Designing efficient database models to handle complex relationships.
- **User Interface:** Ensuring a responsive and intuitive frontend design for better user experience.
- **Security:** Implementing secure authentication and access controls to protect sensitive data.



CONCLUSION

- In conclusion, the Doctor Management System developed using Python Django and SQL provides an efficient solution for managing doctors, patients, appointments, and medical records within a healthcare setting. The system's modular design allows for scalability and potential future enhancements to meet evolving requirements.

