**SYNOPSIS**

**Report on**

**“PATIENT MANAGEMENT SYSTEM”**

**By**

**PRAJJWAL YADAV**

**Session:2023-2024 (III Semester)**

**Under the supervision of**

**Dr. SHASHANK BHARDWAJ(Asst. Professor)**

### KIET Group of Institutions, Delhi-NCR, Ghaziabad



### Department Of Computer Applications

**KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206**

( - 2023)

**ABSTRACT**

The Patient Management System is a Python-based software solution designed to streamline the operations of healthcare facilities. This system offers a comprehensive set of features for efficient patient data management, **appointment scheduling, medical records maintenance, billing, and more.** With a user-friendly interface and a focus on data security and compliance, it empowers healthcare professionals to deliver superior patient care. Key components include patient registration, appointment scheduling, electronic medical records, billing and invoicing, reporting, user access control, integration capabilities, data backup, and adherence to healthcare data privacy regulations. **This system aims to enhance efficiency of Patient , reduce administrative overhead, and improve the overall quality of healthcare services** **related to Patient.**

**TABLE OF CONTENTS**

Page Number

1. Introduction --
2. Literature Review --
3. Project / Research Objective --
4. Research Methodology --
5. Project / Research Outcome --
6. Proposed Time Duration --

References --

**INTRODUCTION**

Hospital are the essential part of our lives which provides us with the best medical facilities for various sickness, it may be due to the change in climatic conditions, stress (emotional trauma) etc. It is necessary for the hospital to keep track of all activities and records day in and day out of its patient, doctors, nurses and other staffs that keeps the hospital in its operation

Keeping track of all activities and reports on paper is very inefficient and time consuming and also error prone. Keeping records on paper is a traditional base system that sometimes do not make it robust, in any case of damage all files will be lost that will cost a lot to the organization Day in and day out many people visit the hospital and when using the traditional base system it make it unreliable in the sense that it will take longer time to enter or access data and also maintaining. It is not economically and technically feasible to maintain these records on paper.

Patient Management System (PMS) is a comprehensive web-based application designed to streamline healthcare facilities and enhance patient management. This project aims to provide a centralized platform that allows doctors, patients, and clinic administrators to efficiently manage medical records, appointments, and communication.

The primary objective of the Patient Management System is to improve the patient experience, optimize healthcare provider efficiency, and maintain accurate and secure medical records. By leveraging technology, the system simplifies the management of healthcare facilities and enhances the overall quality of care provided.

**BACKGROUND/PROBLEM STATEMENT**

A patient must wait in the hospital for a physician's appointment. The patient also queues for an appointment. The patient will not learn whether the doctor cancelled the appointment due to an emergency until they arrive at the hospital. This is tedious for all involved.

Most people do not know what to eat when they have a fever or other illness, so eating the wrong foods will make them sicker.

Our Python-Based Patient Management Scheduling System manages appointments. It is simple to schedule a doctor's appointment online using the system. It also provides an efficient solution in which users can view various booking slots and select their preferred date and time, as well as an AI-powered chatbot that advises patients to avoid foods that will make them sicker, recommends common medications, etc.

This project's front end consists **of HTML, CSS, and JavaScript**, while the back end is written in **Python**. MySQL Database as the database, Flask as the framework, and **AI chatboat uses ML**.

**PROJECT OBJECTIVE**

1. The system offers a dedicated dashboard for doctors to manage their appointments, view patient records, and update their profiles. Doctors can efficiently organize their schedules and access patient information in a user-friendly interface.
2. Patients have access to a personalized dashboard that enables them to schedule appointments, access their medical history, and communicate with their doctors. They can easily book appointments, track their healthcare journey, and receive important notifications.
3. Clinic administrators can efficiently manage doctors, appointments, and clinic information through a comprehensive admin dashboard.
4. AI Chatbot will be introduced which will empower patients with personalised recommendation through AI – powered chatbot, offering guidance on yoga , dietary choices and medications.

**RESEARCH METHODOLOGY**

The research methodology for developing a Patient Management System involves a systematic approach to ensure the successful design and implementation of the software. The methodology typically includes:

1. **Problem Identification**: Define the specific challenges and requirements within the healthcare facility, such as data management, appointment scheduling, and billing.
2. **Literature Review**: Study existing patient management systems and related healthcare software to understand best practices and potential innovations.
3. **Requirement Analysis**: Gather detailed requirements through interviews, surveys, and consultations with healthcare professionals and administrators.
4. **System Design**: Develop a comprehensive system architecture, database structure, and user interface design based on the gathered requirements.
5. **Prototyping**: Create a prototype of the system to validate design choices and gather feedback from stakeholders.
6. **Development**: Implement the system using Python and appropriate frameworks, ensuring scalability, security, and compliance with healthcare regulations.
7. **Testing and Validation**: Conduct rigorous testing to identify and rectify bugs and ensure the system functions as expected.
8. **Deployment**: Install and configure the system within the healthcare facility, providing necessary training to users.
9. **Data Migration**: Safely migrate existing patient data into the new system.
10. **Monitoring and Maintenance**: Continuously monitor system performance and provide ongoing maintenance and support.
11. **Evaluation**: Assess the system's impact on healthcare operations, efficiency, and patient care through feedback and data analysis.
12. **Documentation**: Maintain comprehensive documentation for system architecture, design, and user manuals.

This research methodology ensures a structured and data-driven approach to develop a Patient Management System that meets the specific needs of healthcare providers while adhering to industry standards and regulations.

**PROJECT OUTCOME**

The project or research outcome of a Patient Management System (PMS) can have several significant benefits for healthcare facilities and professionals. Here are some key outcomes:

1. **Improved Patient Care:** A well-implemented PMS can lead to better patient care by providing healthcare providers with easy access to comprehensive patient records. This, in turn, enables more accurate diagnoses and treatment decisions.
2. **Enhanced Efficiency:** PMS automates administrative tasks like appointment scheduling and billing, reducing paperwork and administrative overhead. This improves the efficiency of healthcare facilities, allowing them to focus more on patient care.
3. **Accurate Data Management:** The PMS ensures accurate and up-to-date patient data, reducing the risk of errors in treatment and billing. This also facilitates compliance with data privacy regulations.
4. **Streamlined Workflow:** Healthcare professionals can quickly retrieve patient information and communicate more effectively with other team members, leading to a smoother workflow within the facility.
5. **Patient Engagement:** Many PMS solutions offer patient portals, allowing patients to access their medical records, schedule appointments online, and receive reminders. This increases patient engagement and satisfaction.
6. **Data Analysis:** The system generates reports and analytics, which can be used to identify trends, improve healthcare practices, and make data-driven decisions.
7. **Cost Savings:** By automating tasks and reducing errors, a PMS can lead to cost savings for healthcare facilities over time. It helps in more accurate billing and claims processing.
8. **Integration Capabilities:** PMS can integrate with other healthcare systems, such as laboratory information systems and electronic health records (EHRs), creating a more connected healthcare ecosystem.
9. **Compliance:** The PMS ensures that healthcare facilities adhere to data privacy and security regulations, such as HIPAA (Health Insurance Portability and Accountability Act) in the United States.
10. **Scalability:** It can grow with the healthcare facility's needs, accommodating an increasing volume of patients and data.

In summary, the outcome of implementing a Patient Management System is the transformation of healthcare operations, leading to improved patient care, increased efficiency, enhanced data accuracy, and better overall healthcare management. It's a valuable tool for modern healthcare facilities seeking to provide high-quality care while managing administrative tasks effectively.

**ADVANTAGES**

* 1. The system is straightforward to maintain.
  2. It is easy to use.
  3. It makes scheduling doctor's appointments simple and efficient.
  4. Patients can schedule appointments without leaving their homes.
  5. It can aid in reducing patient absences

**LIMITATIONS/DISADVANTAGES**

a. The patient cannot change the slot or date once booked.

b. It requires a large database.

**SYSTEM DESCRIPTION**

The system consists of the following three major modules and their sub-modules:

* **Admin:**

1. **Login:**

* The admin can log in using their credentials.

1. **Manage Doctor:**

* The admin can add, update, delete and view doctors’ details.

1. **View Patients:**

* The admin can search patients by their name and patient Id.
* They can view the patients’ details and their past treatments.

1. **View Appointments:**

* The admin can view the appointment details by filtering the dates.

1. **View Feedback:**

* They can also view the feedback given by patients.
* **Doctor:**

1. **Login**:

* The doctor can log in using their credentials.

1. **Profile**:

* The doctor can manage their profile.

1. **Change Password:**

* They can change their passwords if they want.

1. **View Appointments:**

* The doctor can view the appointment details by filtering the dates.
* They can also view patient details and their past treatments.
* They can add treatment for their patients.

1. **View Patients**

* The doctor can search for patients by their names or patient Ids.
* They can view patient details and their past treatments.
* **Patient:**

1. **Register:**

* The patient would need to register first to log in.

1. **Login:**

* The patient can log in after registering.

1. **Profile**:

* They can manage their profile.

1. **Change Password:**

* They can change their password if they want.

1. **New Booking:**

* The patient can choose the doctor, date and slot.
* After making all the selections, they can book an appointment.

1. **Booking History:**

* The patient can view all their appointments here.
* They can cancel bookings anytime they want.

1. **Search Doctor:**

* The patient can search doctors by their name, type and locality.
* They can view the doctors’ details.

1. **Feedback**

* They can give feedback to Admin.

1. **Treatments:** The patient can view treatments and the details added by doctors.

**System Requirements**

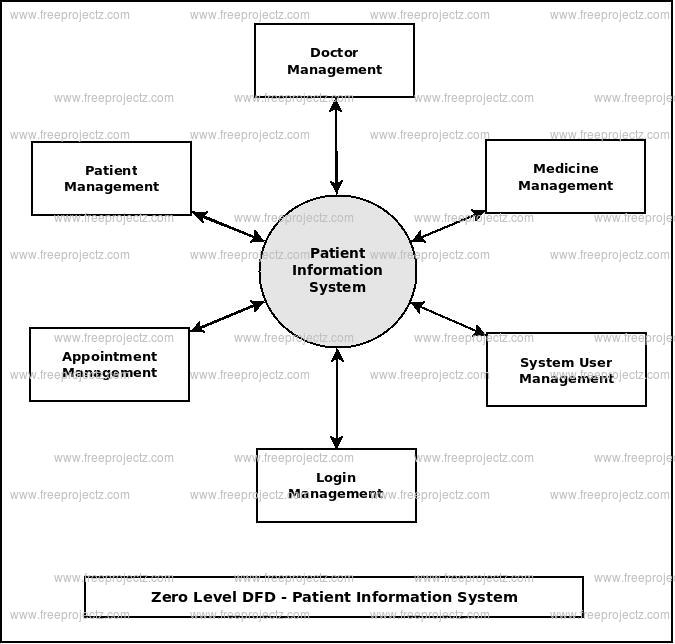
1. **Hardware Requirement**
2. **Laptop or PC**

* Windows 7 or higher
* I3 processor system or higher
* 4 GB RAM or higher
* 100 GB ROM or higher

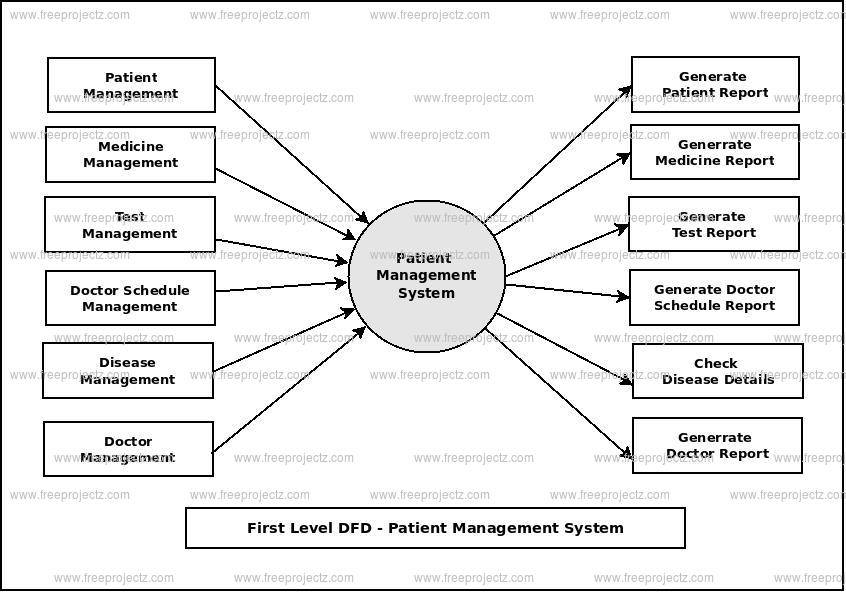
1. **Software Requirement**
2. **Laptop or PC**

* Python
* Sublime Text Editor
* XAMP Server

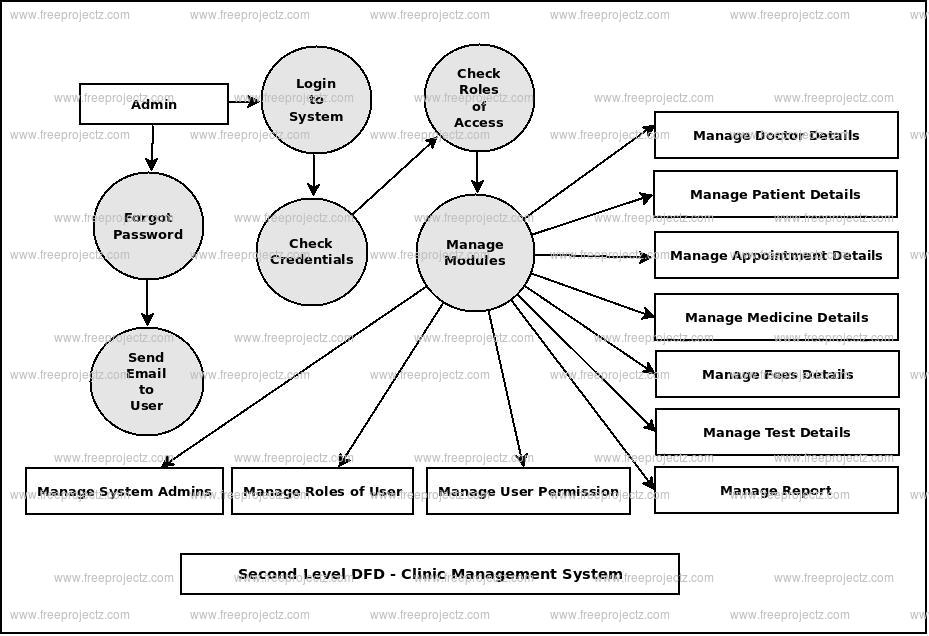
**DATA FLOW DIAGRAM(DFD)**



**FIRST LEVEL OF DFD OF PATIENT MANAGEMENT STSTEM-:**



**SECOND LEVEL OF DFD OF PATIENT MANAGEMENT STSTEM-:**



**Application**

* The system allows patients or any user to schedule a doctor's appointment online with ease.

**Reference**

* <https://www.jetir.org/papers/JETIR2005455.pdf>
* <https://www.academia.edu/26066176/Design_and_Development_of_Online_Doctor_Appointment_System>
* <https://ijcrt.org/papers/IJCRT1812133.pdf>

<https://arxiv.org/ftp/arxiv/papers/1701/1701.08786.pdf>

**REFERENCES**

**<Sample Format>**

1. Seonggeun Ryu, Kyung-Joon Park, and Ji-Woong Choi, “Enhanced fast handover for network mobility in intelligent transportation systems”, IEEE Transactions on Vehicular Technology, Vol. 63, No. 1, pp. 357-371, January 2014, DoI: 10.1109/TVT.2013.2272059.
2. S. Kong, W. Lee, Y. H. Han, M. K. Shin, H. You, “Mobility management for all-IP mobile networks: Mobile IPv6 vs. Proxy Mobile IPv6, IEEE Wireless Communications Vol. 15, Issue 2, pp. 36–45, April 2008, DoI: 10.1109/MWC.2008.4492976