Hospital Management System Report

1. Introduction *This document presents the design and implementation of a database-driven Hospital Management System aimed at efficiently managing patient information, doctor profiles, and appointment scheduling.*

2. Database Design *The system is structured into three primary tables:*

- **Patients**: *Stores information about patients including name, age, contact, and disease.*
- **Doctors:** *Contains details about doctors such as name, specialization, and contact.*
- **Appointments**: *Links patients to doctors with appointment date and status.*

3. SQL Scripts Overview

- **Table Creation**: *The* `Patients`*,* `Doctors`*, and* `Appointments` *tables were created with appropriate primary and foreign key constraints.*
- **Sample Data Insertion**: *Test data representing real-world scenarios was inserted into each table.*
- **Data Cleanup**:
  - *Cancelled appointments were deleted.*
  - *Patients with no active appointments were removed.*

4. Stored Procedures *To facilitate various operations, the following stored procedures were developed*:

i) **GetAppointmentsByDoctor** – *Retrieves all appointments for a specified doctor by name.*

ii) **GetFrequentPatients** – *Lists patients who have had more than one appointment.*

iii) **UpdateAppointmentStatus** – *Updates the status of a specific appointment.*

iv)**GetDoctorsWithMaxAppointments** – *Finds the doctor(s) with the highest number of appointments.*

v) **GetCriticalPatients** – ***Identifies patients marked as 'Critical' in their appointments.***

vi)**GetNextWeekAppointments** – ***Shows all scheduled appointments for the upcoming week.***

vii) **DeleteOldPatients** – ***Removes patient records with no activity for the past five years.***

viii) **GetMonthlyAppointments** – ***Calculates how many patients were treated in a given month.***

*ix)***GetEmergencyDoctors** – ***Displays doctors who have handled emergency cases.***

x) **GetPatientsGroupedByDisease** – ***Groups patients based on their disease.***

5. Observations and Notes

- **All stored procedures are created using MySQL syntax with appropriate use of control structures.**
- **Foreign key integrity is preserved to avoid orphan records.**
- **Edge cases, such as null values or missing references, are handled using `IFNULL` and `LEFT JOIN`.**

6. Conclusion ***The system provides a foundational framework for managing core hospital operations and can be scaled further with features like billing, inventory, and staff management.***

Project by: ATMAJO BURMAN,

B.tech in Computer Science and Engineering,
National Institute of Technology Durgapur.