

**A
Research Project Report
on
“HOSPITAL MANAGEMENT SYSTEM”**



**CHAUDHARI CHARAN SINGH UNIVERSITY MEERUT
IN THE PARTIAL FULFILMENT OF THE REQUIREMENT FOR
THE DEGREE OF
BACHELOR OF COMPUTER APPLICATION
BATCH 2022-2025**

**SUBMITTED TO:
MISS. PAYAL SHARMA**

**SUBMITTED BY:
MR. ASRARUL HAQUE**

**B.C.A (VI SEM)
ROLL NO 220339106022**

**Project guide
MISS. PAYAL SHARMA**



**SHANTI
INSTITUTE
OF TECHNOLOGY**

SHANTI INSTITUTE OF TECHNOLOGY MEERUT

Certificate

1. This is to certify that **ASRARUL HAQUE**, Roll No. **220339106022**, a student of **BCA (VI Semester)** at **Shanti Institute of Technology, Meerut**, has successfully completed the **SUBMITTED BY**
2. Asrarul Haque project titled "**Hospital Management System**" under the guidance of **Miss Payal Sharma** during the academic session **2022–2025**.
3. The project work has been carried out in partial fulfillment of the requirements for the degree of **Bachelor of Computer Applications** under **Chaudhary Charan Singh University, Meerut**, and is a bonafide record of the original work done by the student.

Acknowledgment

1. I sincerely express my gratitude to **Miss Payal Sharma**, my project guide, for her valuable guidance, encouragement, and continuous support throughout the completion of this project.
 2. I would also like to thank the faculty members and staff of the **Department of Computer Applications, Shanti Institute of Technology, Meerut**, for their assistance and helpful suggestions.
 3. Last but not least, I express my heartfelt thanks to my **family and friends** for their motivation, moral support, and constant encouragement during the entire course of this project.
-

Contant Table

1. Introduction
2. Objectives
3. Existing System
4. Proposed System
5. System Design
6. Database Design
7. Implementation
8. Testing
9. Conclusion
10. References

1. Introduction

- The **Hospital Management System** is a software application designed to efficiently manage all aspects of a hospital's operations such as patient registration, doctor appointments, billing, medical records, and report generation.
- In traditional hospital systems, most tasks are performed manually, leading to delays, human errors, difficulty in accessing records, and inefficient management.
- With the increasing number of patients and medical data, there is a strong **need for a computerized system** that can streamline hospital activities, reduce paperwork, and improve accuracy.
- A hospital management system improves **workflow automation**, ensures better **data security**, and helps in **quick decision-making** by making patient information easily available to doctors and staff.
- This system is essential for **modern healthcare facilities** to deliver quality services, maintain efficiency, and ensure timely patient care.

2. Objectives

- To **digitally maintain patient records** including personal details, medical history, and treatment reports.
 - To develop a user-friendly **appointment scheduling system** for both patients and doctors.
 - To create a secure **login system** for different user roles such as admin, doctor, and receptionist.
 - To implement an accurate and efficient **billing system** that calculates consultation, treatment, and medicine charges.
 - To generate **automated reports** like patient history, billing summary, and doctor schedules for better management.
 - To **reduce manual errors** and paperwork by automating daily hospital operations.
 - To ensure **data security and privacy** of sensitive medical records.
-

3. Existing System

- In most traditional hospitals, **records are maintained manually** using paper files and registers.
- **Patient information, appointments, billing, and prescriptions** are written by hand, which is time-consuming and prone to human error.
- Retrieving past records becomes difficult and slow, especially when the data is stored in **physical files**.
- Manual systems **lack real-time updates**, making it harder to coordinate between departments like reception, doctors, and pharmacy.
- There is a **higher risk of data loss**, damage, or displacement due to the lack of digital backup.
- Calculations for billing and inventory management are done manually, which may lead to **inaccurate results and delays**.
- Overall, the manual system is **inefficient, error-prone, and unable to handle a large volume of data**, especially during emergencies.

4. Proposed System

- The proposed **Hospital Management System** is a fully automated software designed to overcome the limitations of the manual system.
 - It allows for **faster processing** of patient registration, appointment scheduling, billing, and report generation.
 - All **patient records are stored digitally**, making it easy to access, update, and manage information securely.
 - The system supports **multi-user login**, allowing doctors, admins, and receptionists to perform specific roles efficiently.
 - It reduces paperwork and **minimizes human error** by automating daily hospital tasks.
 - The system provides **real-time data access**, which helps doctors view patient history instantly during consultations.
 - **Billing and payment processes** are automated, ensuring quick and accurate calculations.
 - It enhances **data security** through password-protected access and controlled user roles.
 - Overall, the system improves **hospital efficiency, service quality, and patient satisfaction**.
-

5. System Design

a) Data Flow Diagram (DFD)

- **Level 0 (Context Level):**

- Represents the overall hospital management system as a single process.
- Shows external entities like Patient, Doctor, Admin interacting with the system.

- **Level 1 DFD:**

- Breaks the system into sub-processes such as:
 - Patient Registration
 - Appointment Scheduling
 - Billing
 - Report Generation

- **Level 2 DFD:**

- Provides detailed flow for each sub-process.
- For example, in "Patient Registration," data flows from patient input to validation and storage in the database.

b) Entity-Relationship (ER) Diagram:

- Shows the relationship between database tables/entities like:
 - Patient
 - Doctor
 - Appointment
 - Bill
 - Admin
- Displays attributes (fields) and primary/foreign keys.
- Helps in designing the database schema logically.

c) Use Case Diagram:

- Identifies the system's functionality from the user's perspective.
- Actors: Patient, Doctor, Admin
- Use Cases: Login, Register, Book Appointment, View Report, Generate Bill

d) User Interface Design:

- Describes how users interact with the system.
 - Should include screenshots or mock-ups of:
 - Login Page
 - Patient Registration Form
 - Appointment Page
 - Billing Interface
 - Doctor Dashboard
-

6. Database Design

a) Tables with Attributes:

1. Patient Table

- Patient_ID (Primary Key)
- Name
- Age
- Gender
- Contact_No
- Address
- Medical_History

2. Doctor Table

- Doctor_ID (Primary Key)
- Name
- Specialization
- Contact_No
- Email

3. Appointment Table

- Appointment_ID (Primary Key)
- Patient_ID (Foreign Key)
- Doctor_ID (Foreign Key)
- Date
- Time
- Status

4. Billing Table

- Bill_ID (Primary Key)
- Patient_ID (Foreign Key)
- Doctor_ID (Foreign Key)
- Amount
- Date
- Payment_Status
-

5. User Login Table

- User_ID (Primary Key)
- Username
- Password
- Role (Admin/Doctor/Receptionist)

b) Sample SQL Queries

Create Table Example:

```
CREATE TABLE Patient (  
    Patient_ID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Age INT,  
    Gender VARCHAR(10),  
    Contact_No VARCHAR(15),  
    Address VARCHAR(255),  
    Medical_History TEXT  
);
```

Insert Data Example:

```
INSERT INTO Patient (Patient_ID, Name, Age, Gender,  
Contact_No, Address, Medical_History)  
VALUES (1, 'Rahul Sharma', 30, 'Male', '9876543210', 'Meerut',  
'No allergies');
```

Join Example to view appointments with doctor and patient:

```
SELECT  
    Appointment.Appointment_ID,  
    Patient.Name AS Patient_Name,  
    Doctor.Name AS Doctor_Name,  
    Appointment.Date,  
    Appointment.Time  
FROM  
    Appointment  
JOIN Patient ON Appointment.Patient_ID = Patient.Patient_ID  
JOIN Doctor ON Appointment.Doctor_ID = Doctor.Doctor_ID;
```

7. Implementation

a) Backend Technologies Used

- The backend of the Hospital Management System is developed using **PHP**.
- PHP handles:
 - Server-side logic
 - Database connectivity (with MySQL)
 - Form processing (e.g., patient registration, login)
 - Session management and user authentication

b) Frontend Technologies Used

- The frontend is developed using:
 - **HTML** for structure and layout of the webpages.
 - **CSS** for styling, responsive design, and formatting.
 - **JavaScript** for client-side interactivity and form validations.

c) Modules of the System

1. User Login Module

- Allows different users (Admin, Doctor, Receptionist) to log in securely.
- Access control is based on user roles.

2. Patient Management Module

- Registers new patients.
- Stores and displays patient details and medical history.
- Allows updating patient records.

3. Appointment Scheduling Module

- Lets patients book appointments with available doctors.
- Displays doctor availability.
- Admin or receptionist can manage appointments.

4. Doctor Management Module

- Maintains doctor profiles including name, specialization, and availability.
- Allows doctors to view patient details and update diagnoses.

5. Billing Module

- Calculates charges for consultations, tests, and treatments.
- Generates and displays bills.
- Tracks payment status.

6. Report Generation Module

- Generates reports such as:
 - Daily appointments
 - Patient history
 - Revenue reports

8. Testing

a) Unit Testing

- Each module of the Hospital Management System was tested individually to ensure that it works as expected.
- Examples:
 - **Patient Registration Form** was tested for correct data entry and validation.
 - **Login Module** was tested with valid and invalid credentials.
- Unit tests helped identify small bugs in form handling and field validation.

b) System Testing

- After unit testing, the complete system was tested as a whole.
- Focus was on checking the **end-to-end flow** from registration to billing.
- Test scenarios included:
 - Booking an appointment after patient registration.
 - Generating bills and reports.
 - Logging in with different user roles (Admin, Doctor, Receptionist).
- All functionalities were verified to work correctly without errors or crashes.

c) User Acceptance Testing (UAT)

- The system was presented to **actual users (students/teachers/peers)** to get feedback.
 - Users were asked to perform tasks like booking appointments, updating records, and viewing bills.
 - Feedback was positive; users found the interface **easy to use**, and the system met **user expectations and requirements**.
-

9. Conclusion

- The **Hospital Management System** project successfully achieved its main objective of digitizing hospital operations like patient registration, appointment scheduling, billing, and report generation.
- It helped in **automating manual processes**, reducing paperwork, and improving the accuracy and speed of hospital services.
- The system provides an easy-to-use interface, secure data handling, and reliable performance for different users like doctors, admins, and receptionists.
- The project improved our understanding of **real-world application development**, including database design, frontend-backend integration, and system testing.

Future Scope

- **Online appointment booking portal** can be added for patients to schedule appointments remotely.
 - Integration of **SMS or email notifications** for appointment reminders and reports.
 - Implementation of **advanced security features** like OTP verification and data encryption.
 - Addition of **inventory management** for pharmacy and lab modules.
 - Making the system **mobile-responsive** or developing a **mobile app** for wider accessibility.
-

10. References

- **Books:**

- *Database System Concepts* by Abraham Silberschatz
- *Web Technologies* by Uttam K. Roy
- *Software Engineering* by Roger S. Pressman

- **Websites:**

- www.w3schools.com – for HTML, CSS, JavaScript, and PHP tutorials
- www.tutorialspoint.com – for database and system design concepts
- www.stackoverflow.com – for debugging and coding solutions

- **Tools Used:**

- **XAMPP** – for running Apache server and MySQL database
- **Visual Studio Code** – for writing and editing code
- **MySQL** – for database design and queries
- **Draw.io / Lucidchart** – for designing ER and DFD diagrams