



جامعة ٦ أكتوبر  
October 6 University



علوم الحاسوب ونظم المعلومات  
Info Sys & Comp Science

# Practical Project

## Hospital Management System

### Prepared By :

Mohab Nasser Abdelkader	NT
Mahmoud Mohamed Ahmed	CS
Mahmoud Hany Mahmoud	CS
Menna Khaled Abdelrehim	CS
Ali Mohamed Mahmoud	NT

### Supervised By :

**Dr. Ayman Hassanein**

### Co-Supervisor :

**Eng. Mohamed Abdelmoneim**

**Egypt 2025**

# **Table of Contents**

# **Table of Contents**

# **Table of Figures**

# **Table of Figures**

# **Abstract**

## **Abstract**

# **Chapter One**

## **"Introduction"**

# **Chapter One**

## **Introduction**

---

# **Chapter Two**

## **"Design and Methodology"**

## Chapter Two

### Design and Methodology

---

#### **2.1 User Requirements**

##### **1. For Patients**

- Patients want to book appointments easily.
- Patients want access to medical history, prescriptions, and test results.
- Patients want to view their bills and make online payments.

##### **2. For Doctors**

- Doctors want quick access to patient records.
- Doctors want to view lab results instantly.
- Doctors want an easy way to write prescriptions and notes.

##### **3. For Nurses & Staff**

- Nurses want to track patient vitals and treatment schedules.
- Staff want to manage rooms, beds, and internal tasks.

##### **4. For Administrators**

- Admins want full system control.
- Admins want to modify departments, staff accounts, and schedules.
- Admins want access to analytics and financial reports.

#### **2.2 System Requirements**

##### **1. Hardware Requirements**

- Central server with at least: 16GB RAM, Multi-core CPU, SSD storage.
- Backup server or cloud backup storage.

- Workstations or tablets for staff.
- Printer and barcode scanner (optional for pharmacy and lab).

## **2. Software Requirements**

- Database: MySQL / PostgreSQL / SQL Server.
- Backend: Java / Python / .NET / Node.js.
- Frontend: React / Angular / HTML5.
- OS: Linux/Windows Server.
- Web server: Apache / Nginx.
- API support: REST/GraphQL.

### **2.3 Functional Requirements**

#### **1. User Management**

- The system shall allow patients, doctors, nurses, and admins to create accounts.
- The system shall authenticate users through a secure login.
- The system shall allow admins to assign user roles.

#### **2. Patient Management**

- The system shall maintain electronic patient records (EHR).
- The system shall store patient demographics, medical history, allergies, lab results, and prescriptions.
- The system shall allow staff to update or review patient records.

#### **3. Appointment Management**

- The system shall allow patients to book, reschedule, or cancel appointments.
- The system shall display available time slots for each doctor.

- The system shall notify doctors and patients of appointment changes.

## **4. Doctor & Staff Management**

- The system shall allow doctors to view patient files prior to appointments.
- The system shall allow doctors to create prescriptions and medical notes.
- The system shall allow staff to manage schedules and tasks.

## **5. Pharmacy Management**

- The system shall manage drug inventory.
- The system shall track medication stock levels.
- The system shall notify when medicine stock is low.
- The system shall allow pharmacists to dispense medicine.

## **2.4 Non-Functional Requirements**

### **1. Performance**

- Appointment booking must load within 2 seconds.
- The system shall support at least 500 concurrent users.
- Transactions must be processed in under 3 seconds.

### **2. Security**

- All patient data must be encrypted.
- Role-based access must restrict unauthorized access.
- All login attempts must be logged.
- The system shall implement two-factor authentication for admins.

### **3. Usability**

- User interface must be simple and easy to navigate.
- The system must support both desktop and mobile interfaces.

- Medical record screens should be optimized for fast data entry.

## **4. Reliability & Availability**

- System uptime must be at least 99.5%.
- Automated backups must run daily.
- The system must automatically recover from failures without data loss.

## **5. Scalability**

- The system must support expansion to more departments or branches.
- Database must handle increasing patient records smoothly.

## **6. Maintainability**

- The system must follow modular design principles.
- Code must be documented for future updates.

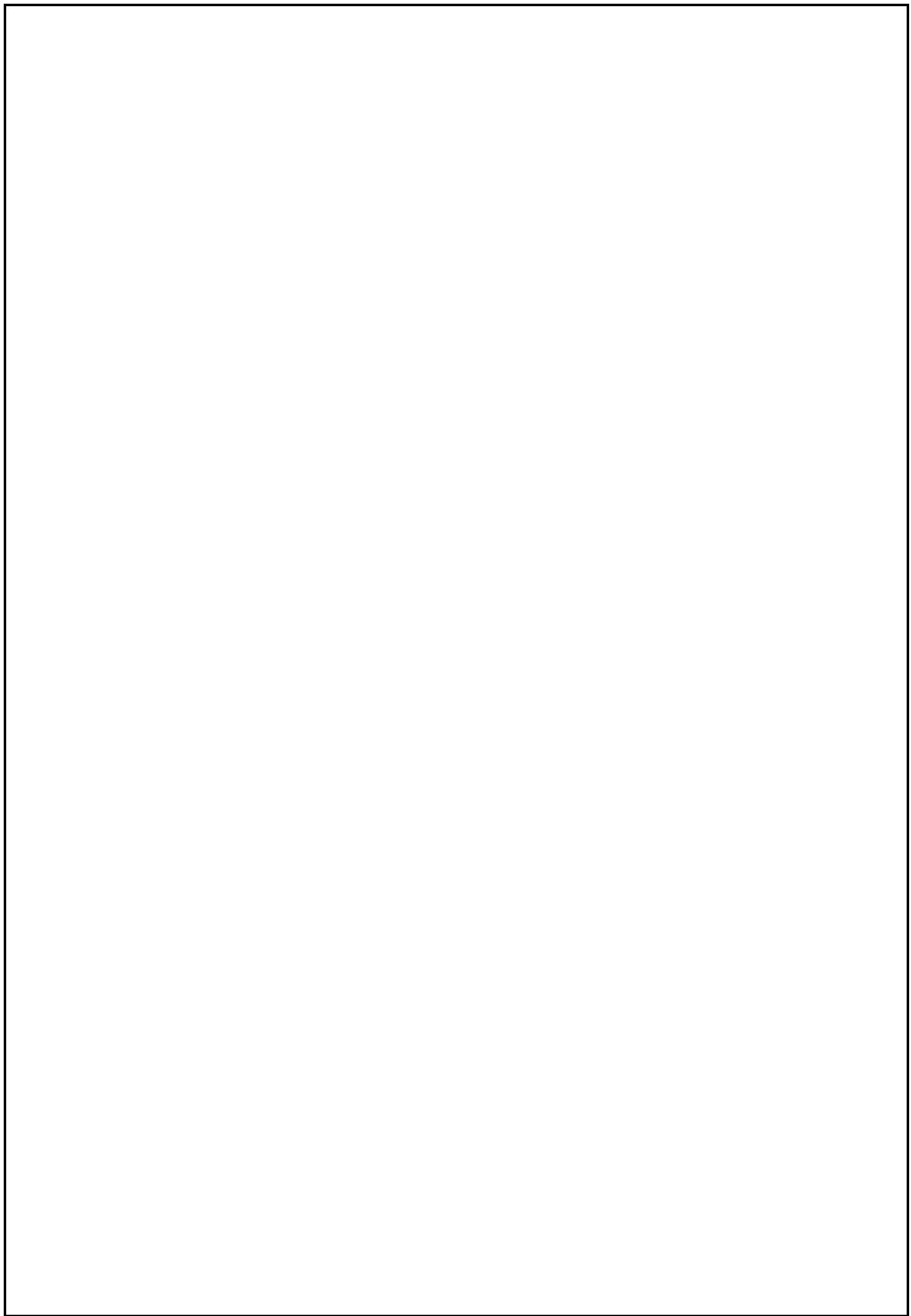
# **Chapter Three**

# **"Conclusion & Recommendations"**

## **Chapter Three**

### **Conclusion & Recommendations**

---



# **References**

**References**

---