VISVESVARAYA TECHNOLOGICALUNIVERSITY
BELAGAVI-590018, KARNATAKA

MINI PROJECT REPORT

ON

"HEALTHCARE DATABASE MANAGEMENT"

Submitted in the partial fulfilment of requirements

FOR

DATABASE MANAGEMENT SYSTEM (BCS403)

Submitted by

SHRADDHA A RATHOD - 4BD23CG035

PROJECT GUIDE

Prof. Kaveri C **B.E, M.Tech**
Assistant Professor

2024-2025

BAPUJI EDUCATIONAL ASSOCIATION®
BAPUJIINSTITUTE OF ENGINEERINGAND TECHNOLOGY
DAVANGERE-577004

BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY
DAVANGERE-577004

DEPARTMENTOF COMPUTER SCIENCEANDDESIGN ENGINEERING

CERTIFICATE

This is to certify that **SHARADDHA A RATHOD** bearing **USN:4BD23CG035** respectively, of **Computer Science and Design** department has satisfactorily submitted the mini project report entitled **“HEALTHCARE DATABASE MANAGEMENT SYSTEM”.** The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the year 2024-2025.

Project Guide	**Head of Department**
Prof. Kaveri C B.E, M.Tech.,	**Dr. Chetana Prakash** B.E, MS, Ph.D.,
Asst. Professor	Prof & Head
Department of CS&D	Department of CS&D
B.I.E.T. Davanagere.	B.I.E.T. Davanagere.

Date:

Place:

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute. "**BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY**" for having well qualified staff and lab furnished with necessary equipment's.

We express our sincere thanks to our resourceful guide **Prof. Kaveri C,** Assistant Professor, Department of Computer Science and Design, B.I.E.T**,** Davanagere who helped us in every aspect of our project. We are indebted to her discussions about the technical aspects and suggestions pertaining to our project.

We express whole hearted gratitude to **Dr. Chetana Prakash**, H.O.D of Computer Science and Design Department. We wish to thank her for making our task easy by providing her valuable help and encouragement.

We also express our whole hearted gratitude to our principal, **Dr. H B Aravind** for his moral support and encouragement.

We would like to extend our gratitude to all the staff of **Computer Science and Design** department for their help and support. We have benefited a lot from the feedback, suggestions given by them.

We would like to extend our gratitude to all our family members and friends for their advice and moral support.

SHRADDHA A RATHOD
(4BD23CG035)

Vision and Mission of the Institute

Vision

“To be a center of excellence recognized nationally and internationally in distinctive areas of engineering education and research, based on a culture of innovation and invention.”

Mission

“BIET contributes to the growth and development of its students by imparting a broad-based engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values.”

Vision and Mission of Computer Science and Department

Vision

"To be a global leader in Computer Science and Design Engineering education and research, fostering innovation and societal impact through interdisciplinary collaboration and technological excellence".

Mission

M1: To educate and inspire the next generation of computer engineers through rigorous in academic programs.

M2: To conduct cutting-edge research that drives technological innovation and addresses global challenges.

M3: To engage in interdisciplinary collaboration, bridging technology and other fields to create holistic solutions.

M4: To contribute to society by translating research into practical, ethical, and sustainable technological advancements**.**

Program Specific Outcomes (PSO):

PSO1: Design and develop software and hardware solutions, meeting industry standards and adapting to evolving technological landscapes.

PSO2: Analyze and address complex engineering problems, and to design and develop interdisciplinary and innovative systems.

PSO3: Inculcate effective communication skills, team work, ethics, leadership in preparation for a Successful career in industry and R&D organizations.

Program Educational Objectives (PEO):

PEO1: Capable of adapting to dynamic technology, fostering innovation, and excelling in the computer science and design engineering industry.

PEO2: Prepare for advanced studies and research, enabling them to contribute to cutting-edge knowledge and emerging technologies.

PEO3: Cultivate ethical professionals who communicate effectively, collaborate in diverse teams, an address complex problem with societal impact.

PEO4: Empower with leadership skills, social responsibility, and a global perspective, enabling them to make positive contributions to society

ABSTRACT

The Healthcare Database Management System (HDBMS) is a software application designed to efficiently store, manage, and retrieve healthcare-related data for patients, doctors, and hospital administrators. The system aims to streamline medical recordkeeping, enhance patient care, and improve administrative workflows in hospitals and clinics. This project focuses on developing a secure, scalable, and user-friendly database management system that can handle a variety of data types, including patient demographics, medical histories, treatment plans, prescriptions, diagnostic reports, and billing information. The system allows healthcare providers to quickly access and update patient records, track appointments, and monitor treatment progress. Key features include role-based access control to ensure data privacy, automated backups, real-time data analytics, and interoperability with existing hospital management systems. The database is designed using relational database management principles (e.g., MySQL/PostgreSQL), and the user interface is implemented through a web-based platform or application interface. Overall, the HDBMS aims to improve healthcare delivery by minimizing errors, reducing paperwork, and ensuring accurate, timely access to critical health information.

CONTENTS

CHAPTER-1 INTRODUCTION

1.1 INTRODUCTION

In today's digitally driven world, the healthcare industry is undergoing rapid transformation. At the heart of this change lies the Healthcare Database Management System, or HDBMS—a critical technology that stores, manages, and secures vast amounts of patient data. A healthcare database management system is a specialized software solution designed to handle the unique and complex needs of healthcare providers. From patient records, diagnostic reports, and treatment histories to billing, prescriptions, and scheduling, an HDBMS brings all critical data under one integrated platform.

These systems enable healthcare professionals to access accurate information instantly, leading to faster diagnoses, safer treatments, and improved patient outcomes. Moreover, they help maintain data privacy and compliance with regulatory standards like HIPAA and GDPR, ensuring that sensitive health information is always protected. Modern HDBMS platforms support advanced features such as electronic health records (EHR), cloud-based access, real-time updates, and data analytics, allowing for smarter decision-making and efficient hospital operations.

A Healthcare Database Management System is not just a tool—it is the digital backbone of modern healthcare, transforming how care is delivered, managed, and evolved in the 21st century. Database management is an integral aspect of record keeping, clinical data extraction, and data compliance across the healthcare sector, ensuring a provider or service has the right processes to accurately store, coordinate, and access information. Healthcare organizations rely on data for a wide variety of tasks and functions, from raising invoices and insurance claim documentation to updating patient medical records, logging test and lab results, and communicating data to devolved practices and other providers.

Innovations in records automation are only made possible with accurate, up-to-date, and responsive databases, a necessary asset to help professionals carry out their work, collate vast amounts of data, and refer back to it whenever interacting with a patient or payer.

1.2 OVERVIEW

The Healthcare Database Management System is a digital solution designed to efficiently manage and streamline the storage, retrieval, and use of healthcare data within hospitals, clinics, or other medical institutions. It addresses the critical need for accurate, accessible, and secure health information to support better patient care, reduce administrative burdens, and comply with regulatory requirements.

1.3 PROBLEM DEFINITION

Modern healthcare facilities generate and rely on vast amounts of data, including patient records, medical histories, diagnostics; treatment plans, billing information, and more. However, many healthcare institutions still face significant challenges in managing this data effectively due to:

Fragmented Data Storage: Patient information is often stored across multiple systems, leading to redundancy, inconsistency, and difficulty in accessing complete records.

Data Security & Privacy Concerns: With sensitive patient data involved, ensuring compliance with privacy laws (e.g., HIPAA) and protecting against unauthorized access is critical.

Lack of Real-Time Accessibility: Healthcare professionals need instant access to up-to-date information, but traditional systems may suffer from delays, limiting effective decision-making.

Inefficient Data Retrieval: Non-standardized and poorly indexed data makes searching for specific patient information time-consuming and error-prone.

Integration Issues: Difficulty in integrating with other health information systems (labs, pharmacies, insurance providers) affects the continuity and quality of care.

Scalability: As the volume of healthcare data grows exponentially, systems must scale without compromising performance or reliability.

1.4 OBJECTIVES

The main objectives of this project is –

1. To *design* store all patient-related information—such as personal details, medical history, test results, prescriptions, and treatment plans—in a unified and accessible digital format.
2. To *develop* robust security measures that ensure patient data is protected from unauthorized access and to provide authorized users with real-time access to current patient information, which is critical for timely medical decision-making.
3. To *implement* enable healthcare providers to search, retrieve, and update patient records efficiently, thereby improving diagnosis, treatment, and administrative processes.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY REVIEW

[1]. Dave Garets and Mike Davis, "Electronic Patient Records, EMRs and EHRs" October 2020-Healthcare lriformatics.

E-Healthcare data management is a core issue in a modern hospital. The data generated is enormous due to newer diagnostic techniques. The medical doctors and patients demand a greater life time for medical data records, they require fast data retrieval, and presentation services for Clinical Decision Support Systems (CDSS). The cost on maintaining a data service Center in an average hospital may be substantial both on infrastructure and qualified ICT staff. The availability of large storage capacities and computing resources from Cloud Service Providers (CSP) is seen as an opportunity to save investment by outsourcing information services. Electronic Patient Record (EPR) management is sensitive issue which requires a foolproof security. It is extremely important to develop methods to relieve the clients from problems of security, integrity, availability and cross CSP mobility. We present E-Healthcare models for ubiquitous services for data acquisition archiving and presentation in Cloud. The management issues and security concerns in cloud domains are addressed by a services architecture proposed. The model includes Wireless Sensor Networks besides communication and storage systems for a typical hospital taking advantage of the Cloud Services Architecture (CSA).

[2]. Jovanov E, Raskovic D, Price J, Chapman J, Moore A, Krishnamurthy A, "Patient monitoring using personal area networks of wireless intelligent sensors", Biomed Sci Instrum. 2021;37:373-8.

The value of clinical data bases has been touted for some time. Yet the current use of clinical data banks has been somewhat limited. Two forces are acting to reverse this situation and to stimulate the growth of data base systems, the costs of computer storage have been dropping exponentially. The price per million characters of magnetic disk storage, the most common form of storage for current computer data bases, has fallen from thousands of dollars a decade ago to well under a hundred dollars today. The use of video disc devices promises even greater economies of scale for some applications as well as exciting possibilities for integrating text and pictures. The problems of the human interface with the data base have been better addressed. In particular the papers in this session demonstrate well the tremendous progress that has been made in aiding the clinician to define and enter his data easily, to retrieve information conveniently, and to analyze it appropriately.

[3]. Berg M. Health information management: integrating information and communication technology in health care work. London: Routledge; 2022.

The technique is designed an application that helps to maximize productivity and efficiency of the hospitals by managing huge chunks of unstructured data in the cloud. Design an elegant and simple user interface by using HTML and CSS in bootstrap framework. Add functionality using JavaScript. The problem that is persisting in health care in India is our lack of insight to patient's data and inefficient work flow in the hospitals. These problems have resulted in improper diagnosis of the patients resulting in death for some extreme cases. Many health care professionals find EMR to be slow and inefficient due to its complexity in using these systems. This makes it hard for the physicians in proper maintenance of case history. Though entering the patient's data seems to be a tedious job it can also provide long term results that are helpful in treating patients. Also, the interaction with the patients with critical health problems by questioning them repeatedly about their case history can be avoided by maintaining this EHRs'.

CHAPTER-3

SYSTEM REQUIREMENT SPECIFICATION

3.1 Hardware Requirement

The hardware required for the development of this project is:

• Processor	:	Intel Core i5
• Processor speed	:	1.19 GHz
• RAM	:	8GB RAM
• System Type	:	64-Bit Operating System

3.2 Software Requirement

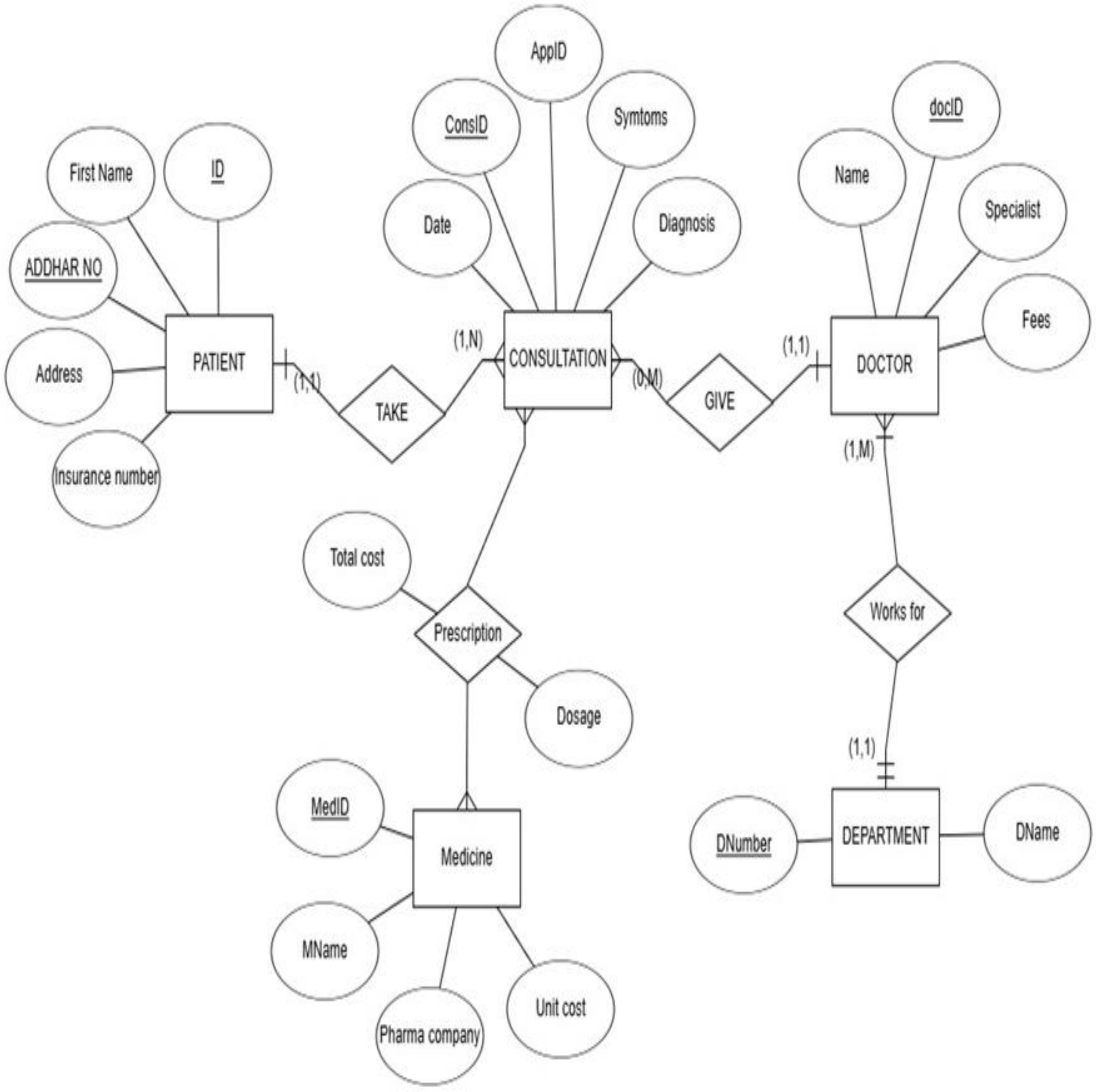
The software required for the development of this project is:

• Software	:	MySQL and Wamp
• OS	:	Windows 11 and higher
• Front End	:	HTML, CSS, Bootstrap
• Programming Language	:	Php and SQL
• Database	:	MySQL

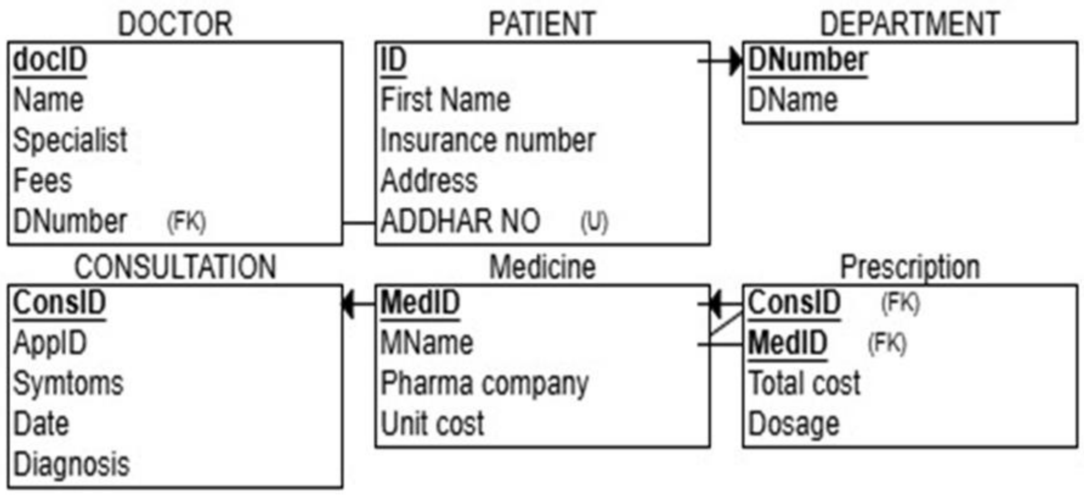
CHAPTER-4

SYSTEM DESIGN

4.1 ER DIAGRAM



4.2 SCHEMA DIAGRAM



4.3 CONSTRAINTS AND RELATIONAL SCHEMA

4.3.1 Entities and Attributes

4.3.1.1 Patients

- patient_id (Primary Key)
- name
- dob (date of birth)
- gender
- contact_number
- email
- address
- medical_history

4.3.1.2 Doctors

- doctor_id (Primary Key)
- name
- specialization

- phone
- email
- department_id (Foreign Key)

4.3.1.3. Departments

- department_id (Primary Key)
- name
- location

4.3.1.4. Appointments

- appointment_id (Primary Key)
- patient_id (Foreign Key)
- doctor_id (Foreign Key)
- appointment_date
- status (scheduled, completed, cancelled)

4.3.1.5. Medical Records

- record_id (Primary Key)
- patient_id (Foreign Key)
- doctor_id (Foreign Key)
- visit_date
- diagnosis
- treatment
- notes

4.3.1.6. Prescriptions

- prescription_id (Primary Key)
- record_id (Foreign Key)
- medicine_name
- dosage

- duration

4.3.1.7. Billing

- bill_id (Primary Key)
- patient_id (Foreign Key)
- appointment_id (Foreign Key)
- amount
- payment_status
- payment_date

2. Relationships between Entities

- **One-to-Many** between Patients and Appointments: A patient can have multiple appointments.
- **One-to-Many** between Doctors and Appointments: A doctor can have many appointments.
- **One-to-Many** between Patients and Medical Records: Each patient can have multiple medical records.
- **One-to-Many** between Doctors and Medical Records: A doctor may create many medical records.
- **One-to-One** between Medical Records and Prescriptions: Each medical record can generate one prescription list (though this could also be One-to-Many depending on the design).
- **One-to-Many** between Patients and Billing: A patient may have multiple billing entries.

4.4 MAPPING CONCEPTUAL DESIGN INTO A LOGICAL DIAGRAM

Step 1: Mapping of Regular Entity Types

- Include only the simple component attributes of a composite attribute .
- Choose one of the key attributes of E as the primary key for R.

- If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R.
- If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R
- In our example-COMPANY database, we create the relations EMPLOYEE, MANAGER , and CUSTOMER.
- We choose Employee_id, Manager_id, and Customer_id as primary keys for the relations EMPLOYEE, MANAGER, and CUSTOMERS, respectively
- The relations that are created from the mapping of entity types are called entity relations because each tuple represents an entity instance.

Step 2: Mapping of Weak Entity Types

- Each weak entity type, create a relation R and include all simple attributes of the entity type as attributes of R
- Include primary key attribute of owner as foreign key attributes of R
- In our example, we create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT
- We include the primary key M_id of the MANAGER_ID relation which corresponds to the entity type as a foreign key attribute of SUPPLIER; we rename it as Sup_id.
- The primary key of the EMPLOYEE relation is the **combination {**EMPLOYEE_SSN/EMP_ID}
- It is common to choose the propagate (CASCADE) option for the referential triggered action on the foreign key in the relation corresponding to the weak entity type, since a weak entity has an existence dependency on its owner entity.
- This can be used for both ON UPDATE and ON DELETE.

Step 3: Mapping of Binary 1:1 Relationship Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R

- There are three possible approaches:
  - foreign key approach
  - merged relationship approach
  - cross reference or relationship relation approach

1. **The foreign key approach**

- Choose one of the relations S, say and include as a foreign key in S the primary key of T.
- It is better to choose an entity type with total participation in R in the role of S
- Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.
- In our example, we map the 1:1 relationship type by choosing the participating entity type EMPLOYEE who works on in any of the project in any of the department.

2. **Merged relation approach:**

- Merge the two entity types and the relationship into a single relation
- This is possible when both participations are total, as this would indicate that the two tables will have the exact same number of tuples at all times

3. **Cross-reference or relationship relation approach:**

- Set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.
- required for binary M:N relationships
- The relation R is called a relationship relation (or sometimes a lookup table), because each tuple in R represents a relationship instance that relates one tuple from S with one tuple from T
- The relation R will include the primary key attributes of S and T as foreign keys to S and T.
- The primary key of R will be one of the two foreign keys, and the other foreign key will

be a unique key of R.

- The drawback is having an extra relation, and requiring an extra join operation when combining related tuples from the tables.

Step 4: Mapping of Binary 1:N Relationship Types

- For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type.
- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R
- Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S
- In our example, we now map the 1:N relationship types DEPENDENT, PROJECT
- For EMPLOYEE we include the primary key SSN/ID of EMPLOYEE relation .
- For MANAGER we include the primary key of the MGR_SSN relation as foreign key in the E_SSN(Employee SSN) relation itself because the relationship is recursive.
- The PROJECT relationship is mapped to the foreign key attribute NAME.

Step 5: Mapping of Binary M:N Relationship Types

- Create a new relation S
- Include primary key of participating entity types as foreign key attributes in S
- In our example, we map the M:N relationship of EMPLOYEE by creating relation between EMPLOYEE and COMPANY. We include the primary keys as SSN and MGR_SSN.
- The propagate (CASCADE) option for the referential triggered action should be specified on the foreign keys in the relation corresponding to the relationship R, since each relationship instance has an existence dependency on each of the entities it relates. This can be used for both ON UPDATE and ON DELETE.

Step 6: Mapping of Multivalued Attributes

- For each multivalued attribute
- Create a new relation
- Primary key of R is the combination of A and K
- If the multivalued attribute is composite, include its simple components
- In our example, we create a relation Address
- The attribute Fname,Lname represents the multivalued attribute of NAME of the EMPLOYEE database.
- A separate tuple will exist in Addrees for each location that a database has
- The propagate (CASCADE) option for the referential triggered action should be specified on the foreign key in the relation R corresponding to the multivalued attribute for both ON UPDATE and ON DELETE.

4.5 DATABASE DESCRIPTION

Database contains five modules i.e. paints, doctors, departments, appointments and Medical records.

1. ***Patients***
   - Stores demographic and contact details of patients.
   - Attributes: patient_id, name, dob, gender, contact_number, email, address
2. ***Doctors***
   - Stores personal and professional details of doctors.
   - Attributes: doctor_id, name, specialization, phone, email, department_id
3. ***Departments***
   - Contains information about hospital departments.
   - Attributes: department_id, name, location

4. Appointments

- Tracks scheduling of patient visits with doctors.
- Attributes: appointment_id, patient_id, doctor_id, appointment_date, status

5. Medical Records

- Records clinical data from consultations or check-ups.
- Attributes: record_id, patient_id, doctor_id, visit_date, diagnosis, treatment, notes

6. Prescriptions

- Contains medication instructions based on medical records.
- Attributes: prescription_id, record_id, medicine_name, dosage, duration

7. Billing

- Manages financial transactions and patient payments.
- Attributes: bill_id, patient_id, appointment_id, amount, payment_status, payment_date

CHAPTER-5

IMPLEMENTATION OF CODE

We have organized Healthcare Database Management System (HDBMS) design. It can be accessed directly or sequentially by registered. This database and its table component are described by using following code

1. **Database Schema (My SQL):**

```
CREATE DATABASE healthcare_db;
USE healthcare_db;

-- Patients Table
CREATE TABLE Patients (
    patient_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    dob DATE,
    gender VARCHAR(10),
    contact_number VARCHAR(15),
    email VARCHAR(100),
    address TEXT
);

-- Doctors Table
CREATE TABLE Doctors (
    doctor_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    specialization VARCHAR(100),
    phone VARCHAR(15),
    email VARCHAR(100)
);
```

```
-- Appointments Table
CREATE TABLE Appointments (
    appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    patient_id INT,
    doctor_id INT,
    appointment_date DATETIME,
    status VARCHAR(20),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);

-- Medical Records Table
CREATE TABLE MedicalRecords (
   record_id INT AUTO_INCREMENT PRIMARY KEY,
   patient_id INT,
   doctor_id INT,
   visit_date DATE,
   diagnosis TEXT,
   treatment TEXT,
   FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
   FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

2. Backend Code (Python+)

```
from flask import Flask, request, jsonify
import mysql.connector

app = Flask(__name__)

# Database connection
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="your_password",
    database="healthcare_db"
)
cursor = db.cursor(dictionary=True)

# Add new patient
@app.route('/patients', methods=['POST'])
def add_patient():
    data = request.json
    query = "INSERT INTO Patients (name, dob, gender, contact_number, email, address) VALUES
(%s, %s, %s, %s, %s, %s)"
    values = (data['name'], data['dob'], data['gender'], data['contact_number'], data['email'],
data['address'])
    cursor.execute(query, values)
    db.commit()
    return jsonify({'message': 'Patient added successfully'}), 201

# Get all patients
@app.route('/patients', methods=['GET'])
def get_patients():
    cursor.execute("SELECT * FROM Patients")
    result = cursor.fetchall()
    return jsonify(result)

# Schedule appointment
@app.route('/appointments', methods=['POST'])
def schedule_appointment():
    data = request.json
    query = "INSERT INTO Appointments (patient_id, doctor_id, appointment_date, status)
VALUES (%s, %s, %s, %s)"
```

```
    values = (data['patient_id'], data['doctor_id'], data['appointment_date'], data['status'])
    cursor.execute(query, values)
    db.commit()
    return jsonify({'message': 'Appointment scheduled'}), 201

# Add medical record
@app.route('/records', methods=['POST'])
def add_record():
    data = request.json
    query = "INSERT INTO MedicalRecords (patient_id, doctor_id, visit_date, diagnosis, treatment)
VALUES (%s, %s, %s, %s, %s)"
    values = (data['patient_id'], data['doctor_id'], data['visit_date'], data['diagnosis'], data['treatment'])
    cursor.execute(query, values)
    db.commit()
    return jsonify({'message': 'Medical record added'}), 201

if __name__ == '__main__':
    app.run(debug=True)
```

CHAPTER-6

SNAPSHOTS

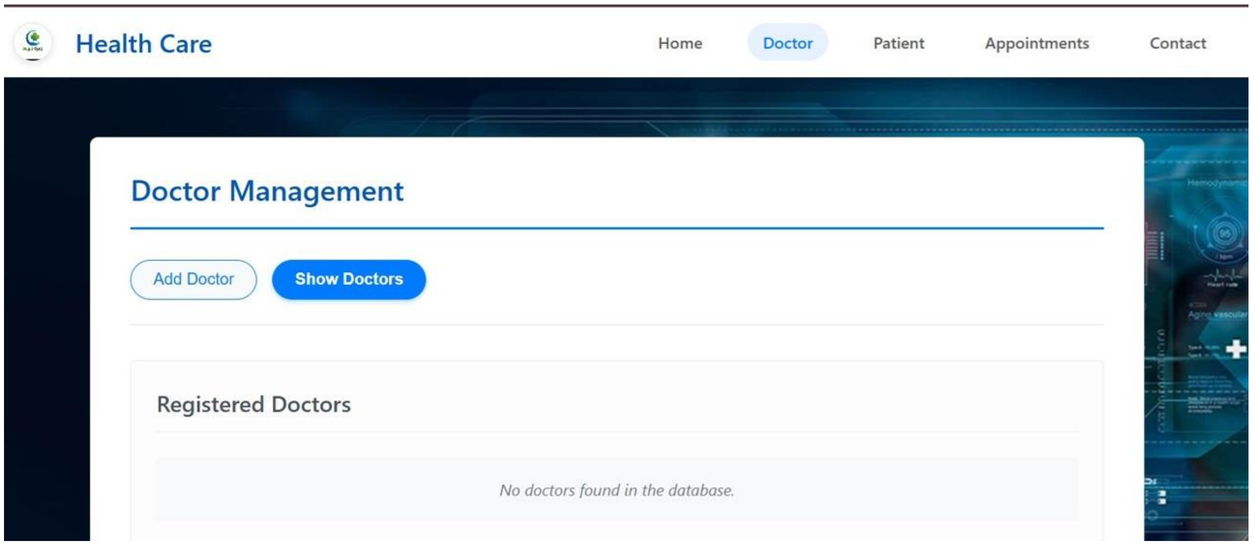
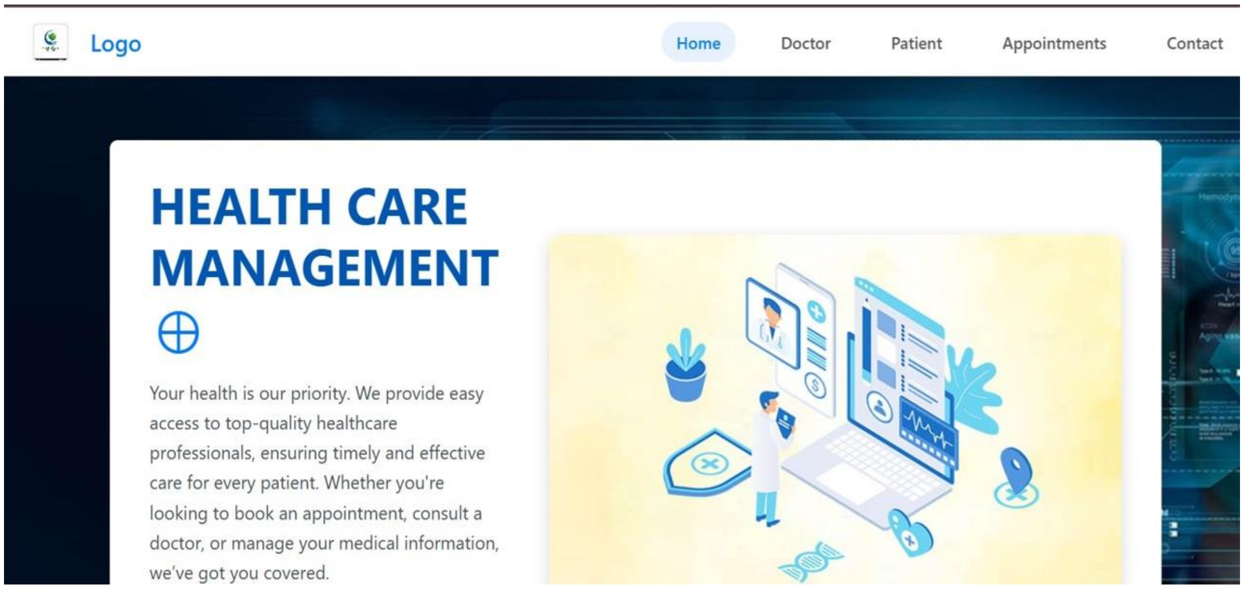
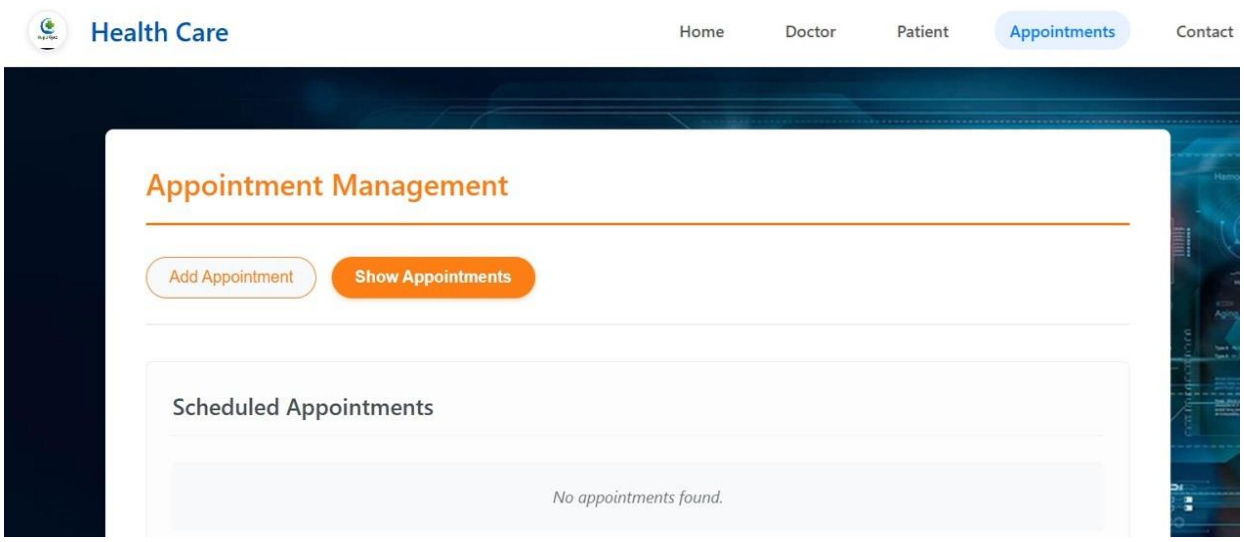


Fig 6.1 LOGIN PAGE



Your health is our priority. We provide easy access to top-quality healthcare professionals, ensuring timely and effective care for every patient. Whether you're looking to book an appointment, consult a doctor, or manage your medical information, we've got you covered.

Fig 6.2: Doctor management module

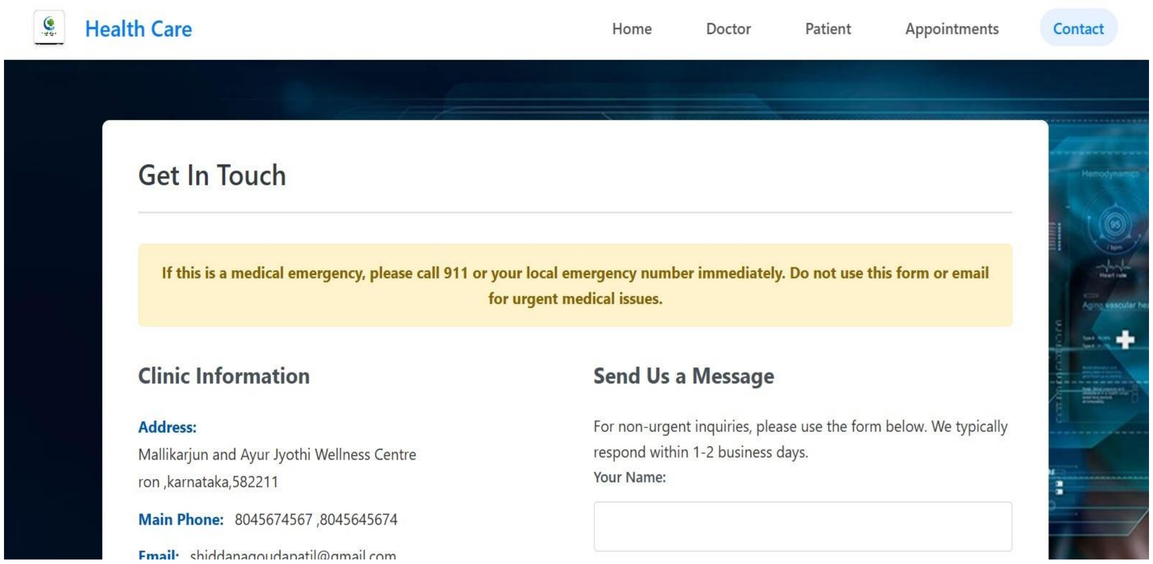


Appointment Management

Scheduled Appointments

No appointments found.

Fig 6.3: Appoint management interface



Get In Touch

If this is a medical emergency, please call 911 or your local emergency number immediately. Do not use this form or email for urgent medical issues.

Clinic Information

Address:
Mallikarjun and Ayur Jyothi Wellness Centre
ron ,karnataka,582211

Main Phone: 8045674567 ,8045645674

Send Us a Message

For non-urgent inquiries, please use the form below. We typically respond within 1-2 business days.

Your Name:

Fig 6.4: Contact us page

CHAPTER-7

CONCLUSION

The Healthcare Database Management System (HDBMS) project successfully addresses the growing need for efficient, accurate, and secure management of patient and clinical data within healthcare institutions. By implementing a centralized and relational database system, the project enhances data accessibility, reduces redundancy, and streamlines core hospital operations such as patient registration, appointment scheduling, medical record keeping, and billing. This system not only improves the overall operational workflow but also supports better decision-making by healthcare professionals through quick and reliable access to critical patient information. The use of structured tables, well-defined relationships, and secure access controls ensures that sensitive medical data is protected and compliant with data privacy standards. Moreover, the modular and scalable design allows for future expansion—such as integrating lab results, pharmacy management, and insurance processing—making it adaptable to both small clinics and large hospital environments. Finally the HDBMS project demonstrates how information technology can transform traditional healthcare practices into more modern, efficient, and patient-focused services. It serves as a foundational step toward building smart, connected, and data-driven healthcare systems.

REFERENCES

[1]. "Development of User-Friendly Web-Based Lost and Found System"

Khairunnahar Suchana, Syed Md. Eftekhar Alam, Anika Tahsin Meem, Manoshi Das Turjo, Mohammad Monirujjaman Khan, October 2021

[2]. "Lost and Found Web Application"

Prof. Pramod Kanjalkar, Jayesh Pingle, Anurag Sagar, Rounak Lohe, Saurabh Patil, Prof. Jyoti Kanjalkar, November 2024

[3] "Digital Lost and Found Item Portal"

Published in: International Research Journal of Engineering and Technology (IRJET), 2021