**+**

**HOSPITAL INFORMATION**

**MANAGEMENT SYSTEM**

*by*

**Abhinav Bansal(22BAI1074)**

*Under the Guidance of*

**Prof. Abishi Chowdhury**



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY CHENNAI - 600127

*February 2024*

**INDEX**

**S.No Name Page No.**

1 Introduction 3

2 Problem statement and our solution 4

3 Assumptions 5

4 ER-Diagram 6

5 RelationalModel 7

6 Normalised Form 8

7 SQL Queries 9

8 Screenshots 11

9 Future Plans 13

**INTRODUCTION**

In the contemporary landscape of healthcare, the efficient management of information is paramount to ensuring the delivery of quality patient care. As the healthcare industry continues to evolve, the need for robust and comprehensive information systems within hospitals becomes increasingly evident. The Hospital Information Management System (HIMS) plays a pivotal role in streamlining operations, enhancing clinical decision-making, and ensuring seamless communication across various departments.

Our Database Management System (DBMS) project aims to address the intricate challenges faced by healthcare institutions by developing an advanced and tailored HIMS. This project recognizes the vital role of information technology in not only digitizing patient records but also in optimizing the overall workflow of hospitals. By leveraging the power of a well-designed database, we aim to create a centralized, secure, and efficient system that meets the unique needs of healthcare providers, administrators, and patients alike.

The HIMS DBMS project will focus on the integration of diverse functionalities, including patient information management, appointment scheduling, inventory control, billing, and prescription management. Moreover, the project will emphasize data accuracy, security, and accessibility, ensuring that authorized personnel can retrieve and update information seamlessly while maintaining the highest standards of confidentiality and integrity.

In this project, we will delve into the intricacies of designing a database schema that reflects the complex relationships and dependencies inherent in a hospital environment. From patient admissions to treatment plans, from doctor schedules to medication records, the database will serve as the backbone, fostering efficient data retrieval and analysis. Additionally, we will explore the implementation of user-friendly interfaces, making the system intuitive for healthcare professionals and staff.

As we embark on this DBMS project, our objective is to contribute to the enhancement of healthcare delivery by providing a scalable, secure, and integrated Hospital Information Management System. Through the judicious use of database technology, we aim to empower healthcare institutions to make informed decisions, reduce administrative overhead, and, ultimately, improve the overall quality of patient care.

**1. PROBLEM STATEMENT:**

The Hospital Management System aims to streamline and enhance the management and operations within a hospital environment. It seeks to provide an efficient platform for managing patient information, doctor schedules, medical records, test results, room assignments, and guardian details. The system should facilitate seamless communication and coordination among doctors, patients, and administrative staff, ultimately improving the quality of healthcare services delivered to patients while optimizing resource utilization and minimizing errors.

**2.PROBLEM DESCRIPTION:**

The Hospital Management System is designed to address the following key challenges and requirements:

- Patient Management: The system should enable the efficient management of patient demographics, medical history, and current health status. It should allow for easy registration of new patients, updating existing patient information, and maintaining accurate records of patient visits, treatments, and diagnoses.

- Doctor Management: The system should facilitate the scheduling and coordination of doctor appointments, surgeries, and consultations. It should provide comprehensive profiles of doctors, including their specialties, contact information, availability, and patient workload.

- Medical Record Management: The system should support the creation, storage, retrieval, and updating of electronic medical records (EMRs). It should allow doctors to record patient diagnoses, treatments, prescriptions, and test results in a standardized format, ensuring data accuracy, confidentiality, and accessibility.

- Test Management: The system should facilitate the ordering, tracking, and reporting of medical tests and procedures. It should allow doctors to request specific tests for patients, monitor test statuses, and access test results promptly. Integration with laboratory systems may be necessary to streamline the test process.

- Room Management: The system should manage the allocation and utilization of hospital rooms and facilities. It should enable administrators to assign patients to appropriate rooms based on their medical needs, room availability, and other relevant criteria. Additionally, the system should support the tracking of room occupancy and cleanliness status.

- Guardian Management: The system should allow for the recording and management of patient guardians or next of kin. It should maintain accurate contact information for guardians and facilitate communication between healthcare providers and guardians regarding patient care and treatment plans.

- Reporting and Analytics: The system should generate comprehensive reports and analytics to support decision-making, resource planning, and performance evaluation. Reports may include patient demographics, doctor productivity, room utilization, test volumes, and financial metrics.

Overall, the Hospital Management System aims to optimize hospital operations, enhance patient care delivery, and improve overall efficiency and effectiveness in healthcare management. It should adhere to industry standards and regulations, ensuring data security, privacy, and compliance with healthcare guidelines and protocols.

**3.ASSUMPTIONS**

1. Doctor Entity Assumptions:

- Each doctor is uniquely identified by a DoctorID.

- Attributes include DoctorID, Name, Specialization, Contact Information, etc.

- Assumption that each doctor may treat multiple patients.

2. Patient Entity Assumptions:

- Each patient is uniquely identified by a PatientID.

- Attributes include PatientID, Name, Age, Gender, Contact Information, etc.

- Each patient may have multiple medical records and tests associated with them.

- A patient may be associated with exactly one guardian (weak entity).

3. Test Entity Assumptions:

- Each test is uniquely identified by a TestID.

- Attributes include TestID, Name, Description, rate, etc.

- Each test is performed on one or more patients.

4. Room Entity Assumptions:

- Each room is uniquely identified by a RoomNumber.

- Attributes include RoomNumber, Type (e.g., ICU, General Ward), Rate, etc.

- Each room may accommodate multiple patients.

5. Guardian (Weak Entity) Assumptions:

- A guardian is not uniquely identifiable without the patient they are associated with.

- Attributes include GuardianID, Name, Relationship to Patient, Contact Information, etc.

- A patient may have one or more guardians.

6. Relationship Assumptions:

- Treat Relationship: A doctor treats one or more patients. A patient is treated by one or more doctors.

- Record Relationship: A medical record is associated with one patient and one doctor and may involve one or more tests.

- Assign Relationship: A patient is assigned to one room at a time. A room may accommodate multiple patients.

- Has Relationship: A patient has one or more guardians. A guardian may be associated with one or more patients.

7. General Assumptions:

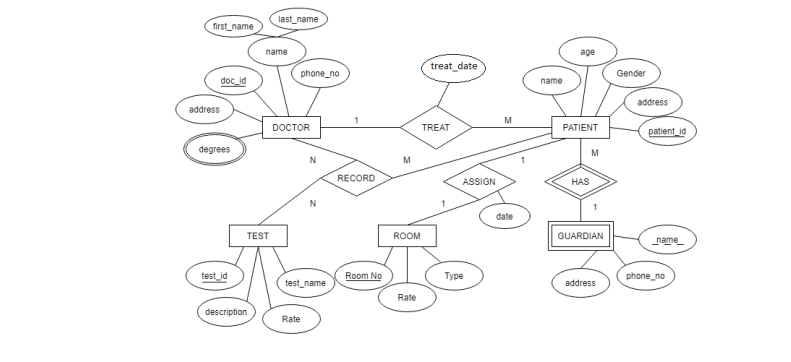
- A patient must exist in the system to be assigned to a room, treated by a doctor, or have medical records.

- Every test or medical record must be associated with at least one patient and one doctor.

- The system maintains historical data, so a patient's medical history includes past treatments, tests, and assigned rooms.

- Integrity constraints such as foreign key constraints, primary key constraints, and cardinality constraints are enforced to ensure data consistency and accuracy.

**4.ER-DIAGRAM**



**5.RELATION MODEL**

Doctor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| address | Doc\_id | First\_name | Last\_name | Phone\_no. |

Doctor\_degree

|  |  |
| --- | --- |
| Doc\_id | degrees |

Treat\_Patient

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Doc\_id | Patient\_id | address | gender | age | name | Treat\_date |

Assign\_Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| date | Patient\_id | Room\_no | rate | type |

Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test\_id | description | rate | Test\_name |

Record

|  |  |  |
| --- | --- | --- |
| Doc\_id | Patient\_id | Test\_id |

Has\_Guardian

|  |  |  |  |
| --- | --- | --- | --- |
| Patient\_id | Name | Phone\_no. | address |

**6.NORMALISED FORM**

**Doctor:**

Primary Key: doc\_id

All other columns (address, first\_name, last\_name, phone\_no) depend on the entire primary key (doc\_id), satisfying 3NF.

**Doctor\_Degree:**

Composite Primary Key: doc\_id and degrees

No other columns present, so there's no dependency issue. This table satisfies 3NF.

**Treat\_Patient:**

Primary Key: patient\_id

doc\_id is a foreign key referencing Doctor(doc\_id).

All other columns (address, gender, age, name, treat\_date) depend on the entire primary key (patient\_id), satisfying 3NF.

**Assign\_Room:**

Primary Key: room\_no

patient\_id is a foreign key referencing Treat\_Patient(patient\_id).

All other columns (date, rate, type) depend on the entire primary key (room\_no), satisfying 3NF.

**Test:**

Primary Key: test\_id

All other columns (description, rate, test\_name) depend on the entire primary key (test\_id), satisfying 3NF.

**Record:**

Composite Primary Key: doc\_id and test\_id

No other columns present, so there's no dependency issue. This table inherently satisfies 3NF.

**Has\_Guardian:**

Composite Primary Key: patient\_id and name

phone\_no and address depend on the entire primary key (patient\_id and name), satisfying 3NF.

**7.SQL Queries for Creating Tabels**

DOCTOR:-

CREATE TABLE `hims`.`doctor` (

`Doc\_id` INT NOT NULL,

`First\_name` VARCHAR(45) NOT NULL,

`Last\_name` VARCHAR(45) NOT NULL,

`Address` VARCHAR(45) NOT NULL,

`Phone No.` INT NOT NULL,

PRIMARY KEY (`Doc\_id`));

Doctor\_Degree:-

CREATE TABLE `hims`.`doctor\_degree` (

`Doc\_id` INT NOT NULL,

`degrees` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Doc\_id`, `degrees`));

ALTER TABLE `hims`.`doctor\_degree`

ADD CONSTRAINT `Doc\_deg`

FOREIGN KEY (`Doc\_id`)

REFERENCES `hims`.`doctor` (`Doc\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT;

Treat\_patient:-

CREATE TABLE `hims`.`treat\_patient` (

`Doc\_id` INT NOT NULL,

`Patient\_id` INT NOT NULL,

`Address` VARCHAR(45) NOT NULL,

`Gender` VARCHAR(45) NOT NULL,

`Age` INT NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Treat\_date` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Patient\_id`),

INDEX `Patient-Doctor\_idx` (`Doc\_id` ASC) VISIBLE,

CONSTRAINT `Patient-Doctor`

FOREIGN KEY (`Doc\_id`)

REFERENCES `hims`.`doctor` (`Doc\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

Assign Room:-

CREATE TABLE `hims`.`assign\_room` (

`Patient\_id` INT NOT NULL,

`Date` DATE NOT NULL,

`Room\_no` INT NOT NULL,

`Rate` INT NOT NULL,

`Type` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Room\_no`),

INDEX `Patient-room\_idx` (`Patient\_id` ASC) VISIBLE,

CONSTRAINT `Patient-room`

FOREIGN KEY (`Patient\_id`)

REFERENCES `hims`.`treat\_patient` (`Patient\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

Test:-

CREATE TABLE `hims`.`test` (

`Test\_id` INT NOT NULL,

`Description` VARCHAR(45) NOT NULL,

`Rate` INT NOT NULL,

`Test\_name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Test\_id`));

Record:-

CREATE TABLE `hims`.`record` (

`Doc\_id` INT NOT NULL,

`Patient\_id` INT NOT NULL,

`Test\_id` INT NOT NULL,

PRIMARY KEY (`Doc\_id`, `Patient\_id`, `Test\_id`),

INDEX `Test\_Connect\_idx` (`Test\_id` ASC) VISIBLE,

CONSTRAINT `Patient\_connect`

FOREIGN KEY (`Doc\_id` , `Patient\_id`)

REFERENCES `hims`.`treat\_patient` (`Doc\_id` , `Patient\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `Test\_Connect`

FOREIGN KEY (`Test\_id`)

REFERENCES `hims`.`test` (`Test\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

Has\_Guardian:-

CREATE TABLE `hims`.`has\_guardian` (

`Patient\_id` INT NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Phone\_No.` INT NOT NULL,

`Address` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Patient\_id`, `Name`),

CONSTRAINT `Patient\_connection`

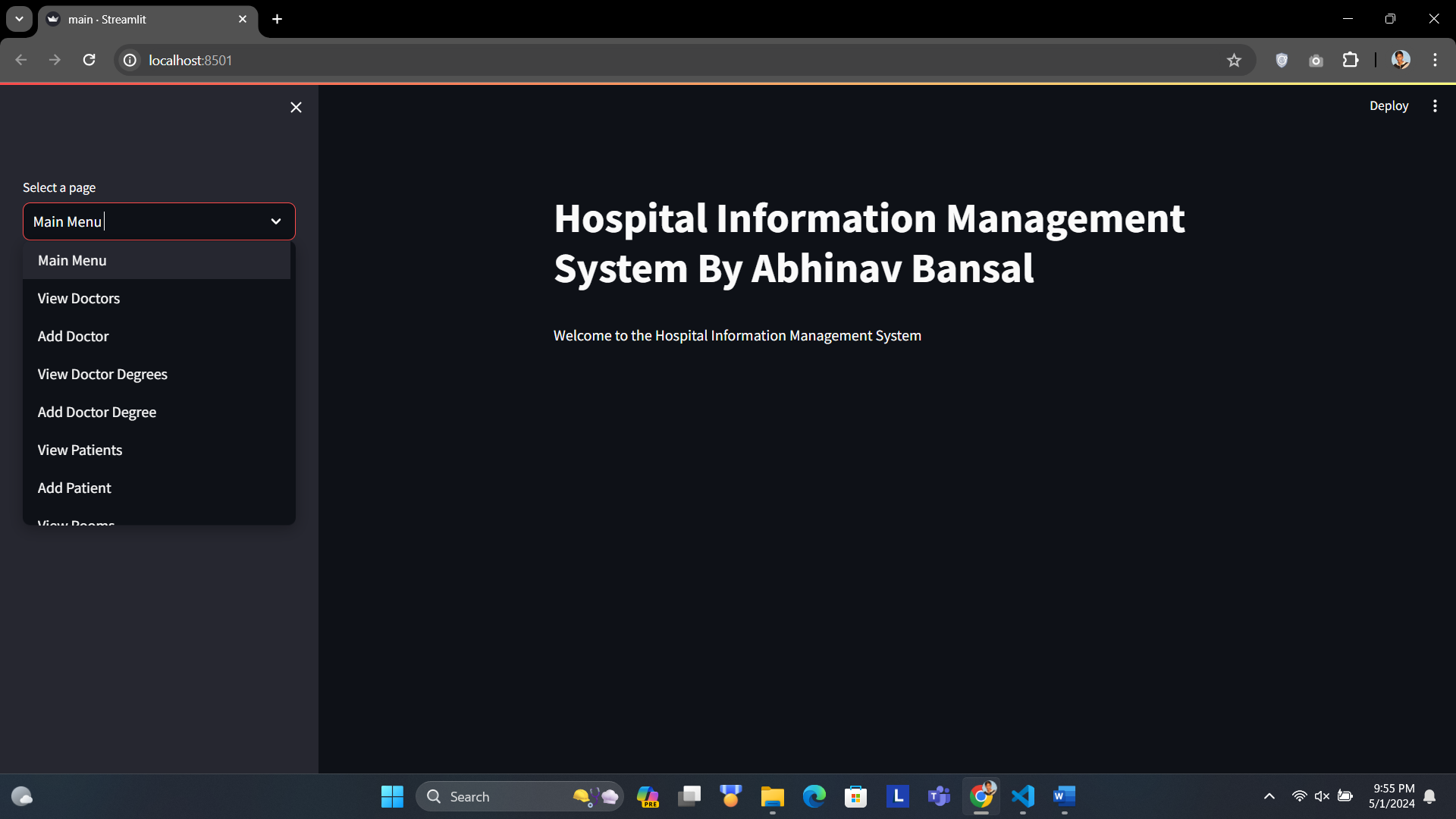
FOREIGN KEY (`Patient\_id`)

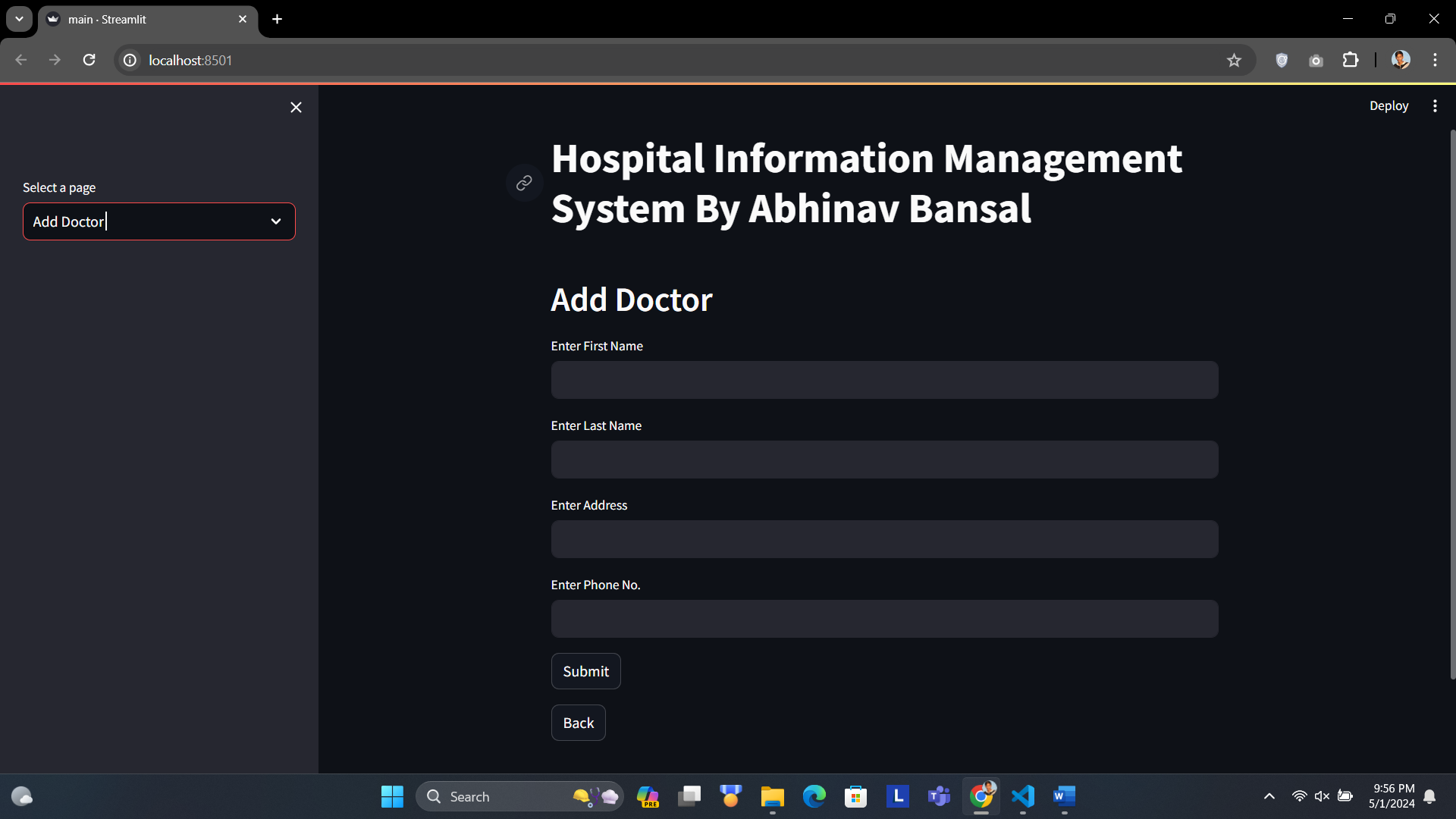
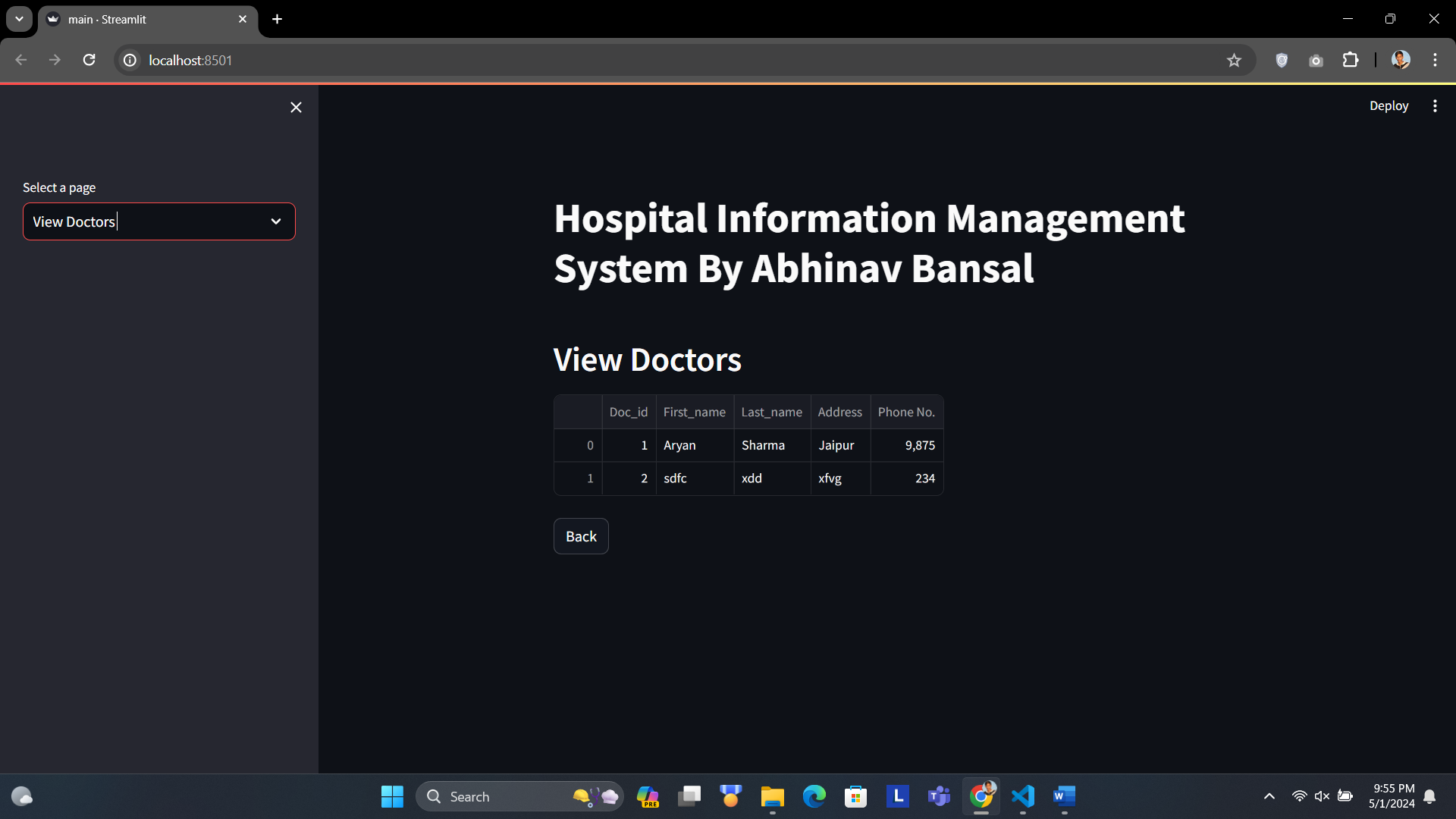
REFERENCES `hims`.`treat\_patient` (`Patient\_id`)

ON DELETE RESTRICT

ON UPDATE RESTRICT);

**8.SCREENSHOTS**

****

****

**9. FUTURE PLANS**

1. Improved User Experience: Enhance interface design for better usability and responsiveness across devices, incorporating user feedback.

2. Advanced Decision Support: Integrate AI-driven decision support tools to aid doctors in diagnosis and treatment planning, leveraging predictive analytics for better patient outcomes.

3. Telemedicine Integration: Expand system capabilities to support telemedicine consultations and remote patient monitoring via mobile apps.

4. Interoperability: Strengthen integration with external systems and standards to ensure seamless data exchange and continuity of care.

5. Patient Engagement: Develop patient portals and educational resources to empower patients and gather feedback for continuous improvement.

6. Security Measures: Enhance cybersecurity protocols and compliance measures to protect patient data and ensure regulatory adherence.

7. Continuous Innovation: Foster a culture of innovation, research, and development to explore emerging technologies and market opportunities for sustained growth and relevance in the healthcare sector.