CSE 111 Project

PATIENT DATA ANALYTICS SYSTEM FOR DOCTORS

Prerana Somarapu
Varsha Vinod

SYSTEM DESCRIPTION

- The **Patient Data Analytics System** is a secure, multi-tier data management platform designed to help doctors efficiently manage and analyze patient medical data across multiple hospitals.

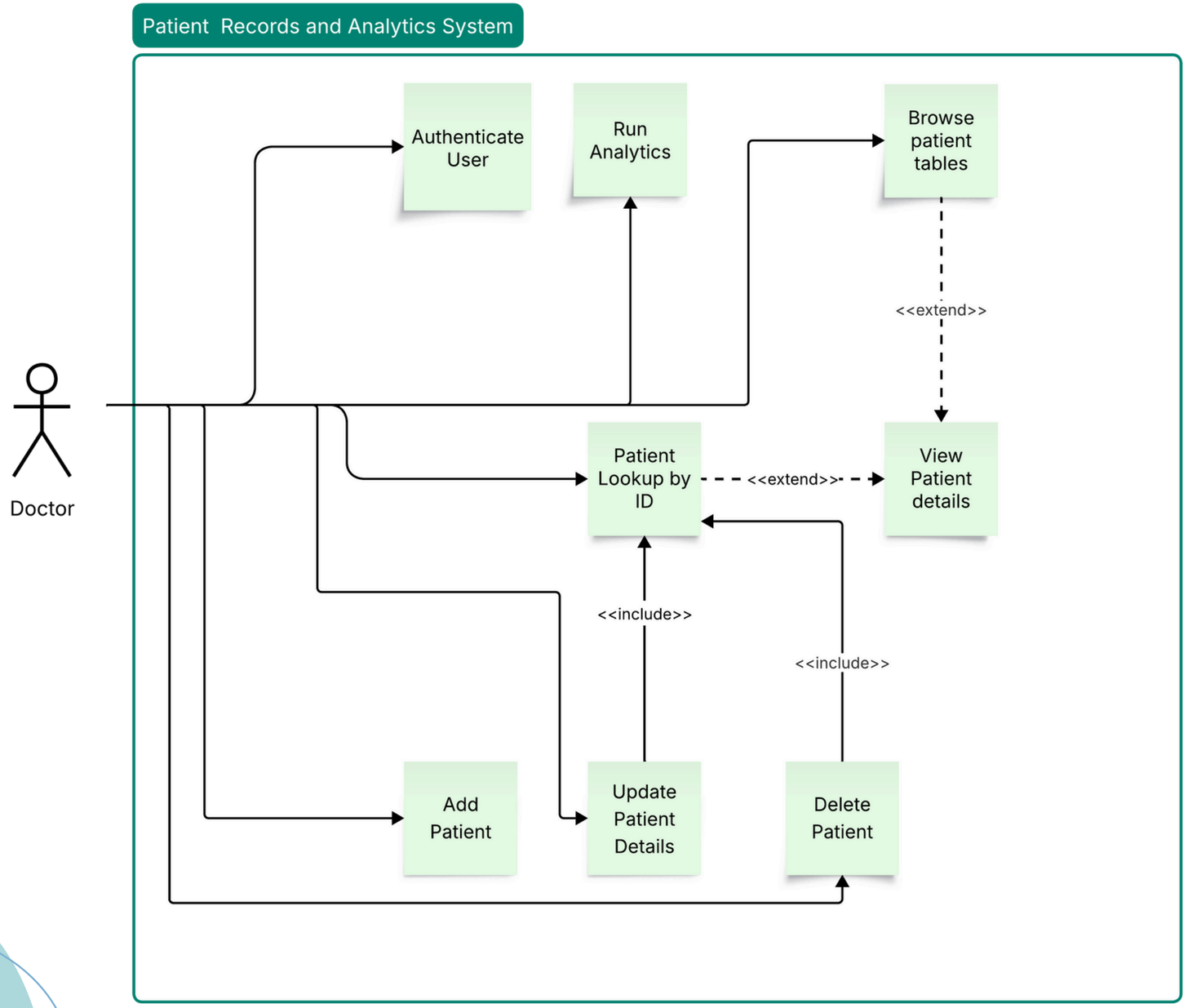
- The system includes data management tools to add, view, update, and delete patient records as well as analytical tools to identify common conditions and analyze demographic and disease trends through interactive charts.

- To protect patient privacy, identifiable information (such as patient name, birth date, and contact information) are excluded from the data used, ensuring compliance with data protection requirements while still allowing doctors to extract helpful statistical insights.

USE CASES

- **Authenticate User**
  - Secure login to the system with ID # and password
- **Patient Lookup/View patient details**
  - Retrieve a patient's information using MRN
- **Add Patient**
  - Create a new patient record
- **Update Patient**
  - Update patient information
- **Delete Patient**
  - Delete outdated/invalid patient records
- **Browse Patient Tables**
  - View patient data by table (ex: condition, hospital, etc.)
- **Analyze Disease/Demographic Prevalence**
  - Extracts statistical insights from patient data
  - Example: Most prevalent gender for a condition, most prevalent blood type per condition, number of patients that took vaccine, etc.

UML USE CASE DIAGRAM



ENTITIES AND RELATIONSHIPS

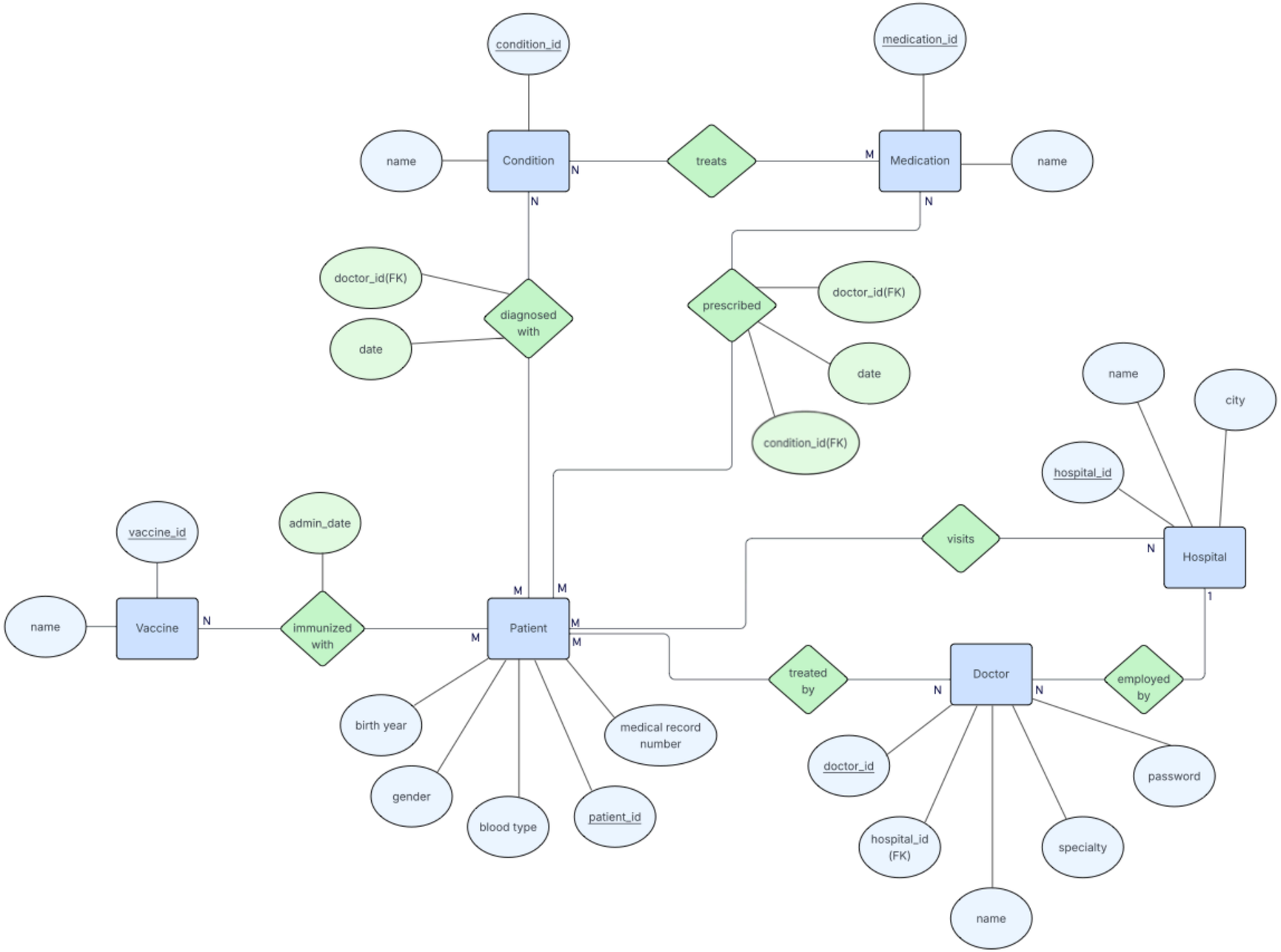
Entities (6):

1. **Patient**
   a. *(patient_id, medical_record_number, birth_year, gender, blood_type)*
2. **Doctor**
   a. *(doctor_id, name, specialty, hospital_id, password)*
3. **Hospital**
   a. *(hospital_id, name, city)*
4. **Condition**
   a. *(condition_id, name)*
5. **Medication**
   a. *(medication_id, name)*
6. **Vaccine**
   a. *(vaccine_id, name)*

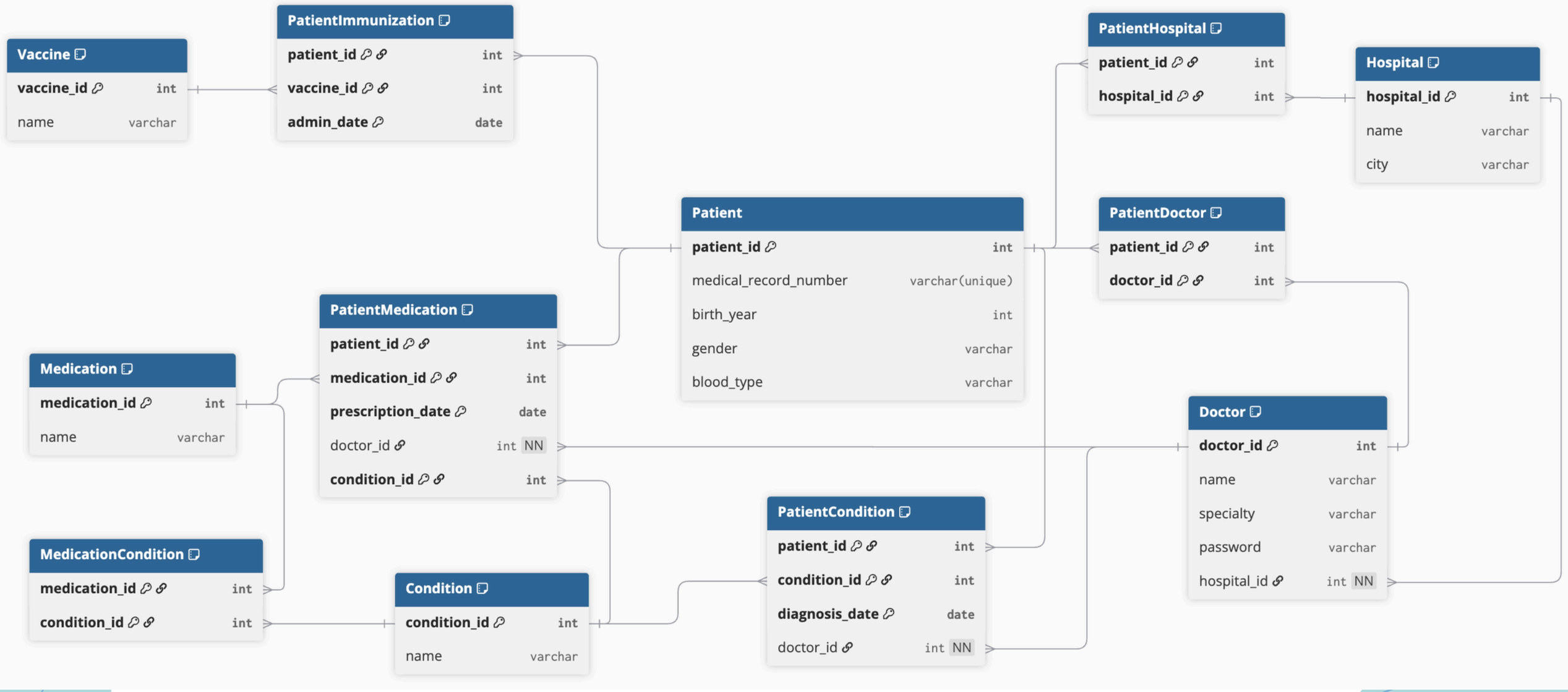
Relationships (7 Total, 6 Many to Many):

1. **Patient — Doctor (M:N)**
   a. A patient can be treated by many doctors; a doctor treats many patients.
2. **Patient — Hospital (M:N)**
   a. A patient can visit many hospitals; a hospital serves many patients.
3. **Doctor — Hospital (N:1)**
   a. Each doctor works at exactly one hospital; a hospital employs many doctors.
4. **Patient — Condition (M:N)**
   a. Patients can have multiple conditions; conditions can apply to many patients.
5. **Patient — Medication (M:N)**
   a. Patients can take multiple medications; medications can be prescribed to many patients.
6. **Patient — Vaccine (M:N)**
   a. Patients can receive multiple vaccines; each vaccine can be given to many patients.
7. **Medication — Condition (M:N)**
   a. A medication can treat many conditions,condition can be treated with many medications.

E/R DIAGRAM


condition_id
medication_id
name
Condition
N
treats
M
Medication
name
N
N
doctor_id(FK)
diagnosed with
date
prescribed
doctor_id(FK)
date
condition_id(FK)
name
city
hospital_id
admin_date
vaccine_id
visits
N
Hospital
1
name
Vaccine
N
immunized with
M
M
Patient
M
M
M
treated by
N
Doctor
N
employed by
birth year
medical record number
gender
blood type
patient_id
doctor_id
password
hospital_id (FK)
specialty
name

RELATIONAL SCHEMA



IMPLEMENTATION DETAILS

Backend (Application Layer)

- The backend was implemented in Python using the Flask framework, which handles routing, processes user input, and connects the user interface to the database.

Database

- The system uses a SQLite relational database we created (healthcare_analytics.sqlite) based on the E/R diagram and relational schema, with tables and relationships representing patients, doctors, conditions, medications, and vaccines.

User Interface

- A web-based graphical interface built with HTML and CSS allows users to interact with the system through forms and menu-style navigation.

THANK YOU !