DBMS (18CSC303J)
Mini Project

Hospital Management System

By -

Vishrut Pundir (RA1811027010029)

Rohan Mathur(RA1811027010048)

Tejas Chintala (RA1811027010014)

Abstract

Our project is aimed to act as a management system for both patients and doctors for managing their interactions. Patients & doctors both can see all the details that are required & add , delete & update it.

Hospital Management System project is a smart and effective way to store records of the patients tagged with their diseases & appointments with the doctors. The main target of this project is to target the patient information according to their details.

System Requirements

- Computer System with Windows / Linux/ Mac OS Operating System
- Python (3.0 or above)
- Locally installed MongoDB (For the database)
- A virtual environment with the following libraries installed -
    1) Flask
    2) Pymongo
- Online Search Engine with the ability to render HTML web pages & run our web app.
- Command line to run our web app.

Tools Used

Front End

Our front end is a simplistic UI which shows all the details of the Hospital management system. We created this simple system from scratch by designing the UI using HTML for structure and CSS for styling.

Back End

We are using MongoDB to connect to our backend database using python as our programming language.

MongoDB is being hosted locally for the data.
The following commands have to be run in the command line to activate & make our empty database.

How To Run

Firstly , ensure that MongoDB is added to your environment variable. After that , type the following commands in your command line -

- use medicords
- db.createCollection("appointment")
- db.createCollection("disease_data")
- db.createCollection("doctors_details")
- db.createCollection("medical_history")

- db.createCollection("patient_details")
- db.createCollection("posts")
- db.createCollection("treats")

Next, install flask & pymongo . Use -

- pip install flask
- pip install pymongo

Change your directory to the folder

- Type - “python app.py”

Copy the given web link into your browser.

Connectivity Steps & Procedure

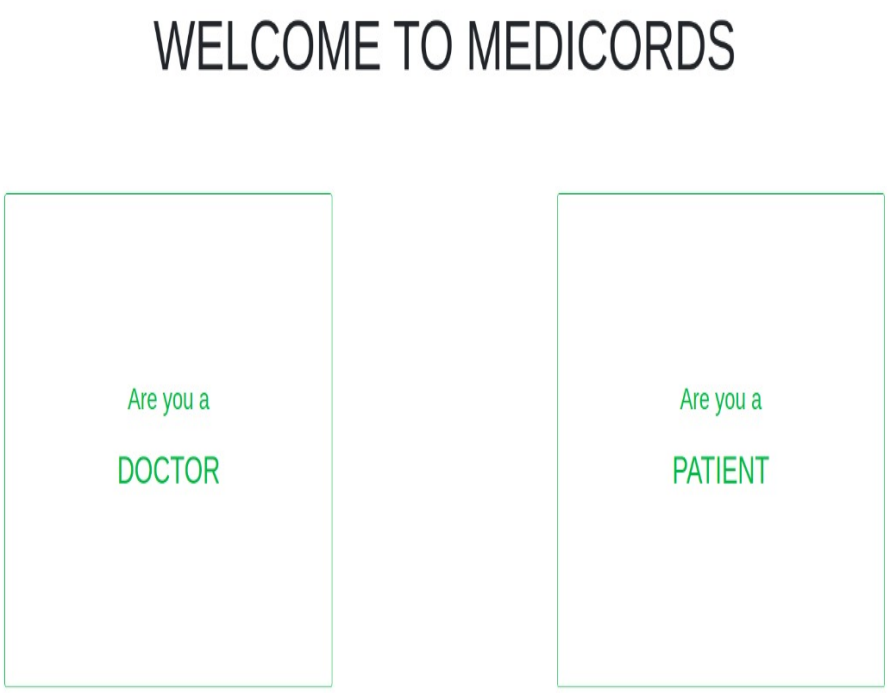
The data input is being handled from the frontend & is being stored in the database. This database can be added, stored. updated, deleted & can also be searched based on keywords The keywords are -

1) Patients - Username , Password , Email ID
2) Doctors - Name , Email ID , Specialization , Phone number , Password

The web app also has a login system which can be used with the above keywords.

Demo Screenshots

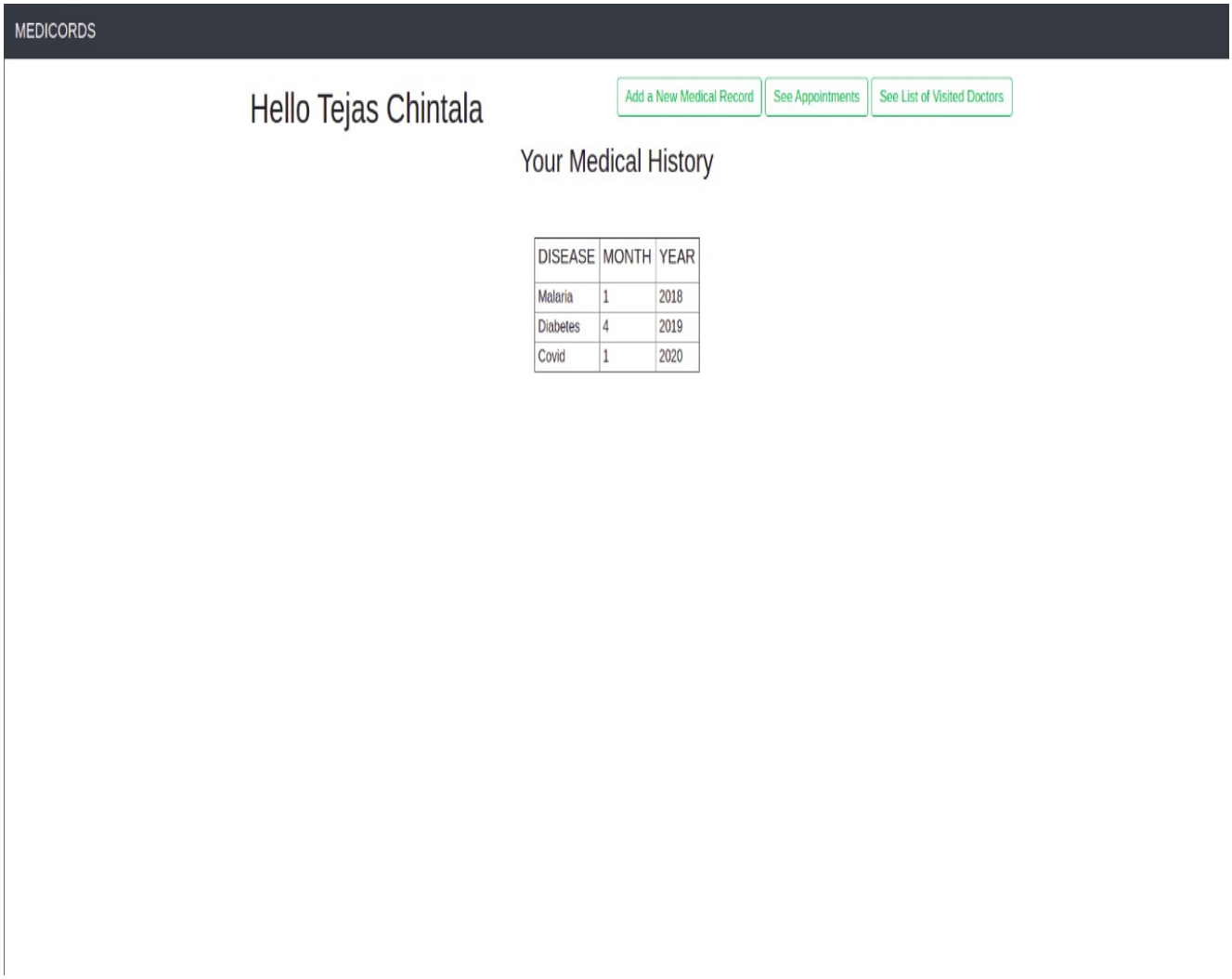
First Page - User can select if they are Doctor or Patient



The Patient's Page

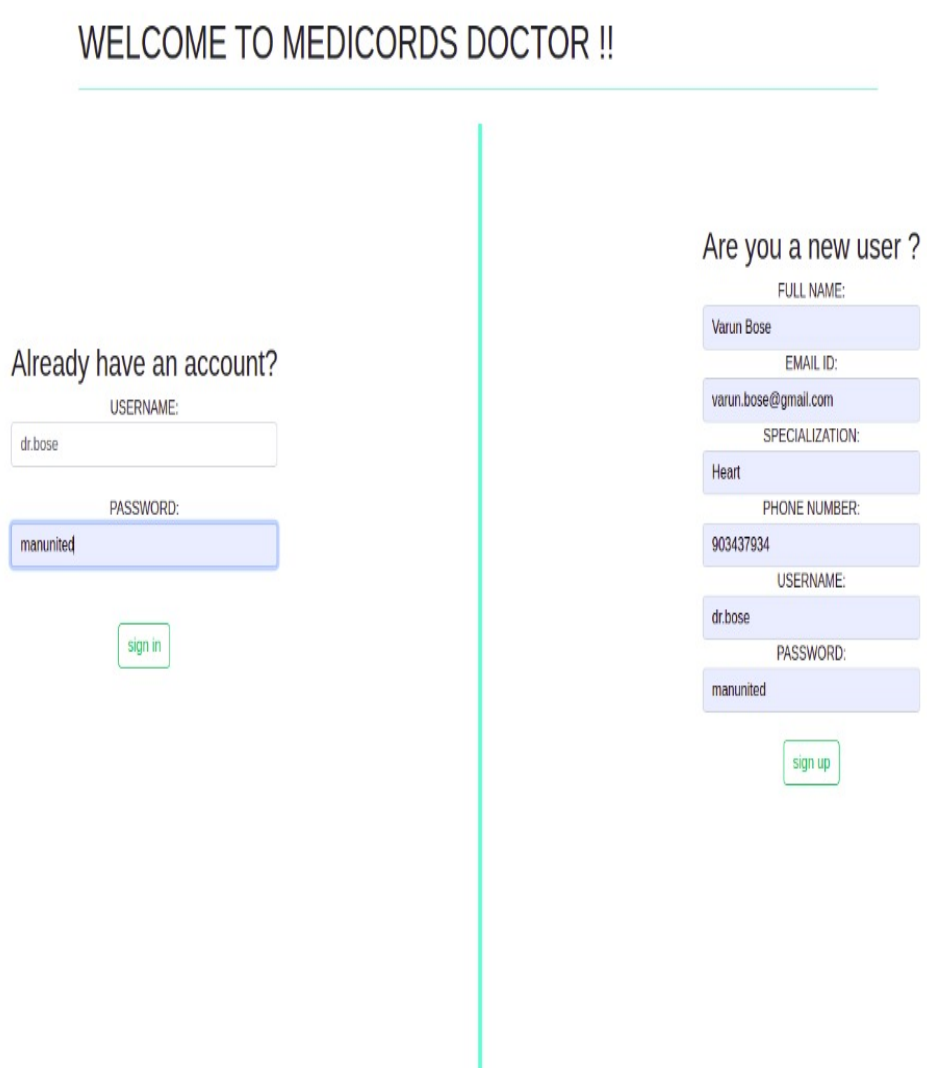
The Patient can edit the following -

1) Medical Record
2) See List of Appointments with the Doctor
3) See List of Available Doctors



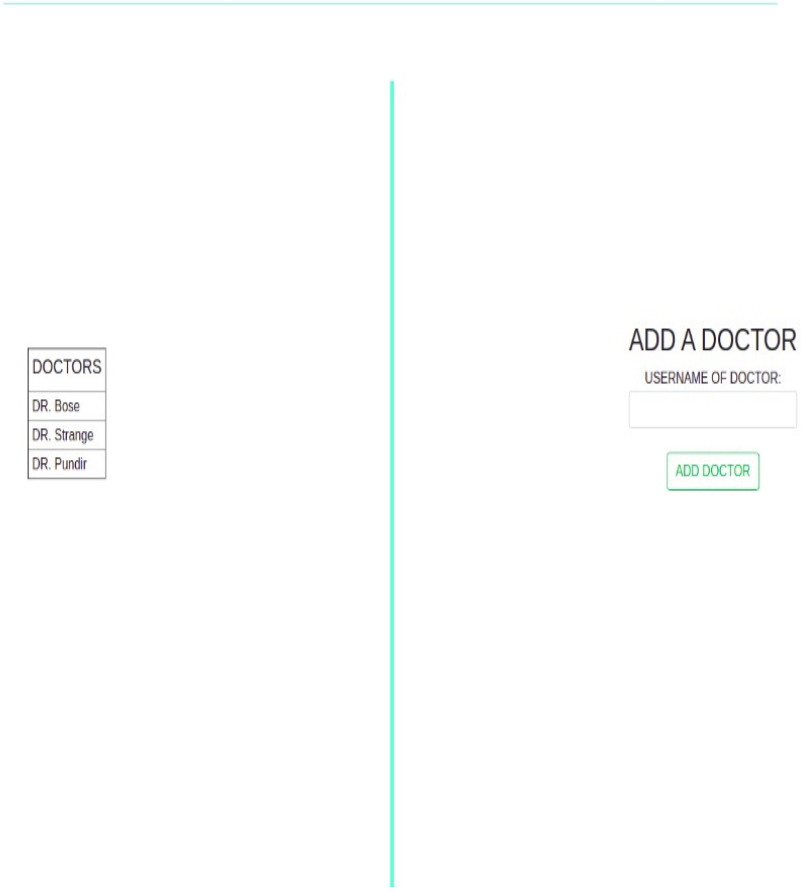
The Doctor's Login Page

The Doctor can enter their necessary details to login - such as Specialization , Email ID etc. Multiple Doctors can be added & selected for Appointments



Add Doctor Page

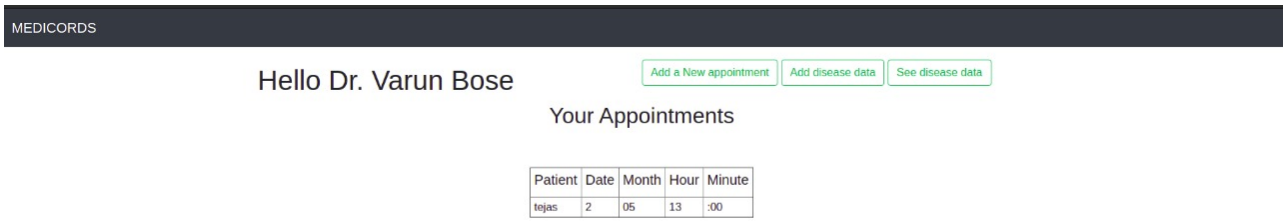
The add doctor page can be used by patients to add a new doctor in the patient database



Appointment Page

We have appointment pages both for doctor and the patient.So both patient and doctor can view their respective appointments.

1. Doctor Appointment Page - Here the doctor can view his appointment with his patients.



2. Patient Appointment Page - Similarly here the patient can view his appointments with doctors.

Conclusion

The hospital management system has been implemented using the necessary tools.

Future Add ons

- More details can be added such as Payment Status , Allotment of Room Numbers, Possible Home Remedies in Covid Times etc.
- Connecting our frontend to a better, online hosted database using APIs to expand & increase our database size

References

Github Link
https://github.com/RohanMathur17/Hospital-Management-System

Flask Documentation
https://flask.palletsprojects.com/en/1.1.x/

Pymongo Documentation
https://pymongo.readthedocs.io/en/stable/

MongoDB Documentation
https://docs.mongodb.com/