Python Doctor Appointment Booking System

1) ***Background/ Problem Statement***

A patient must wait in the hospital for a physician's appointment. The patient also queues for an appointment. The patient will not learn whether the doctor cancelled the appointment due to an emergency until they arrive at the hospital. This is tedious for all involved.

Most people do not know what to eat when they have a fever or other illness, so eating the wrong foods will make them sicker.

Our Python-Based Doctor Appointment Scheduling System manages appointments. It is simple to schedule a doctor's appointment online using the system. It also provides an efficient solution in which users can view various booking slots and select their preferred date and time, as well as an AI-powered chatbot that advises patients to avoid foods that will make them sicker, recommends common medications, etc.

This project's front end consists of HTML, CSS, and JavaScript, while the back end is written in Python. MySQL Database as the database, Flask as the framework, and AI chatboat uses ML.

2) ***Working of the Project***

The Admin, Doctor, and Patient modules comprise this Python-based project.

The administrator's credentials may be used to log in. They can add, modify, remove, and view physicians. They can access patient information and previous treatments using patient IDs and names. Applying a date filter allows the administrator to view appointment details. Additionally, they may examine the feedback provided by patients.

The doctor can log in using their credentials. They can manage their profile and change their password if they so choose. By filtering the dates, the details of any appointment can be viewed. If they wish to find a particular patient, they can conduct a name or ID search. They have access to all patient-related information, including prior treatments. In addition, they can provide additional treatments for their patients.

Prior to logging in, the patient must first register. After logging in, the patient can manage their profile and change their password. They can schedule a doctor's appointment by selecting the doctor, date, and available appointment times. They have complete access to the appointment history. Even cancellations are permitted at any time. They can search for doctors by name, specialty, or location. They have access to the doctor's information. They can also provide feedback to administration. They have access to the treatment details physicians have added.

3) ***Advantages***

a. The system is straightforward to maintain.
b. It is easy to use.
c. It makes scheduling doctor's appointments simple and efficient.
d. Patients can schedule appointments without leaving their homes.
e. It can aid in reducing patient absences.

4) ***System Description***

The system consists of the following two major modules and their sub-modules:

- **Admin:**

1. **Login:**
   - The admin can log in using their credentials.
2. **Manage Doctor:**
   - The admin can add, update, delete and view doctors' details.
3. **View Patients:**
   - The admin can search patients by their name and patient Id.
   - They can view the patients' details and their past treatments.
4. **View Appointments:**
   - The admin can view the appointment details by filtering the dates.
5. **View Feedback:**
   - They can also view the feedback given by patients.

- **Doctor:**

1. **Login**:
   - The doctor can log in using their credentials.
2. **Profile**:
   - The doctor can manage their profile.
3. **Change Password:**
   - They can change their passwords if they want.
4. **View Appointments:**
   - The doctor can view the appointment details by filtering the dates.
   - They can also view patient details and their past treatments.

- They can add treatment for their patients.

5. **View Patients**

- The doctor can search for patients by their names or patient Ids.
- They can view patient details and their past treatments.

- **Patient:**

1. **Register:**

- The patient would need to register first to log in.

2. **Login:**

- The patient can log in after registering.

3. **Profile**:

- They can manage their profile.

4. **Change Password:**

- They can change their password if they want.

5. **New Booking:**

- The patient can choose the doctor, date and slot.
- After making all the selections, they can book an appointment.

6. **Booking History:**

- The patient can view all their appointments here.
- They can cancel bookings anytime they want.

7. **Search Doctor:**

- The patient can search doctors by their name, type and locality.
- They can view the doctors' details.

8. **Feedback**

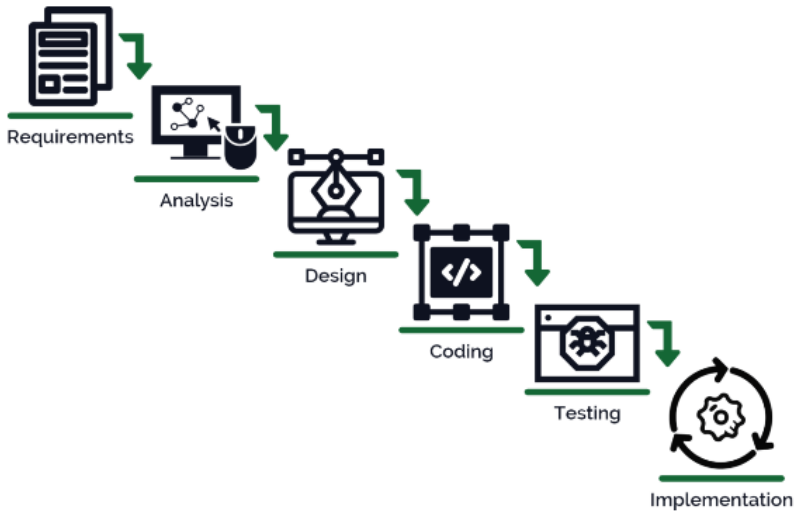
- They can give feedback to Admin.

9. **Treatments:**

- The patient can view treatments and the details added by doctors.

5) ***Project Life Cycle***

The waterfall model is a traditional system development life cycle model used to create a system using a sequential and linear approach. It is known as a waterfall because the model progresses systematically from one phase to the next in a descending order. The waterfall methodology does not specify a procedure for reverting to a previous phase to accommodate requirements changes. The waterfall method was the first method utilised for software development.



6) ***System Requirements***

I. Hardware Requirement

i. Laptop or PC

- Windows 7 or higher
- I3 processor system or higher
- 4 GB RAM or higher
- 100 GB ROM or higher

II. Software Requirement

ii. Laptop or PC

- Python
- Sublime Text Editor
- XAMP Server

7) Limitations/Disadvantages

- The patient cannot change the slot or date once booked.
- It requires a large database.

8) Application –

- The system allows patients or any user to schedule a doctor's appointment online with ease.

9) Reference

- ✓ *https://www.jetir.org/papers/JETIR2005455.pdf*
- ✓ *https://www.academia.edu/26066176/Design_and_Development_of_Online_Doctor_Appointment_System*
- ✓ *https://ijcrt.org/papers/IJCRT1812133.pdf*
- ✓ *https://arxiv.org/ftp/arxiv/papers/1701/1701.08786.pdf*