DocLink Portal : Streamlining Patient-Doctor Interactions

Swati Gupta[a], Tushar Verma[b], Pashupati Nath Singh[c], Bharti Chugh[d],Pankaj Bhatt[e]

[a]Department of Computer Science and Engineering KIET Group of Institutions, Delhi-NCR Ghaziabad-Meerut Road, Ghaziabad-201206, India
[b]Department of Pharmacy, Lloyd Institute of Management and Technology, Plot No. 11 Knowledge Park-II, Greater Noida, 201308, Uttar Pradesh, India

Abstract:
Proper healthcare is essential to leading a healthy lifestyle. However, making a doctor's appointment in today's hectic life may be quite difficult. Many people don't have time to wait in long queues and apart from that the majority of individuals are not aware of which doctors they should consult. This may take a significant amount of time and money to locate and schedule appointments with a variety of medical professionals. To simplify these issues, a system is developed that will connect the patients to the concerned doctors effortlessly. Using the suggested paradigm, patients can gain a basic understanding of the illness and a list of specialized doctors for treating the sickness. So this interface allows smooth and effortless communication between patients and the concerned doctors.
Keywords—Appointment Booking, Django, Website, Disease Prediction, Machine Learning

1. Introduction

1.1. The World Health Organization (WHO), states that hospitals are necessary for offering proper healthcare services. However, visiting doctors physically can pose various challenges for individuals. Some of these challenges include: Difficulty in finding time for appointments, particularly for individuals with demanding work schedules or family obligations. Waiting in long queues at the doctor's office can disrupt the daily routine. Discomfort or anxiety associated with visiting medical facilities, particularly for individuals with specific health conditions or phobias. Transportation challenges, especially for those without personal vehicles or in areas with inadequate public transportation. Limited time for effective communication with the doctor during appointments. There is a need for a system that can exponentially reduce people's troubles in a world where the population is growing quickly and the ratio of doctors to patients is very low. Using Django, a web development platform, we propose creating a doctor-patient interaction system[1].

1.2. Motivation

In today's bustling world, many people don't have time to wait in infamously long queues for medical appointments. The problem is that the hospital's administrative staff usually runs the line by hand. After receiving a token, we must wait for our turn before requesting the doctor. The annoying thing is that if we find out after long waiting hours that the doctor is either not in the office or is unable to make an appointment, it will be very annoying apart from that the majority of individuals are aware of which doctors they should consult according to their symptoms. This may take a significant amount of time and money to locate and schedule appointments with various medical professionals. Moreover, every doctor has their own specialized area. This study appears to be a solution to these problems. Patients may now book appointments from the convenience of their homes and verify whether the doctor they want to see is available, which will help us solve all of these issues. A doctor can easily approve or decline the appointment according to his convenience. The proposed approach essentially saves time and money because there is no need for the patient to physically visit the hospital.

1.3. Key Contribution

The objective of this model is to make healthcare more accessible as we can access any service 24/7, reduce waiting times as patients don't need to stand in long queues to book appointments and streamline administrative tasks for doctors and hospitals. Through the innovative use of technology, we aim to create a seamless and effective doctor-patient interaction experience that can connect patients to the concerned doctors effortlessly.

The patients can provide the symptoms as input, and this model will predict a potential disease or illness based on those symptoms. So this model bridges the gap between patients and doctors.

1.4. The proposed system's Perks

a) 24/7 Accessibility: Having a hospital website ensures constant accessibility for patients, regardless of the time or location. This means that patients can access comprehensive information even outside regular business hours. It provides convenience by allowing individuals to access necessary information from the comfort of their homes. This proves particularly beneficial during emergencies, ensuring patients can swiftly obtain the information they need. Healthcare professionals learn how to observe, diagnose, and provide care outside of normal work hours. This helps them be more flexible and sensitive to the needs of their patients. Thanks to advancements in technology, 24/7 access works as a link between patients and healthcare providers, ensuring a smooth flow of care that is not time-dependent.

b) Enhanced Communication: Improved doctor-patient communication through encrypted texting and video consultations promotes better understanding and collaboration, resulting in more effective and efficient management of medical conditions. This improved communication helps advance a patient-centric healthcare strategy by leveraging technology to provide more intensive and personalized treatment. By fostering a collaborative environment through the integration of secure communication systems, the overall effectiveness of managing health conditions is optimized.

c) Immediate Consultations: Encouraging quick access to medical assistance, particularly in an emergency, guarantees that people

obtain immediate advice regarding their health issues. The ability to obtain healthcare expertise more quickly through Internet platforms encourages health management and timely decision-making. Accepting such services improves the effectiveness of healthcare delivery by offering prompt and easily available assistance for people's health-related needs.

d) Appointment Scheduling: Enabling people to schedule appointments online gives them the freedom to make and manage their appointments conveniently and effectively. This creative solution greatly improves the overall effectiveness of the healthcare system in addition to helping to shorten wait times in long queues. Giving people the freedom to select appointment times that work with their schedules makes the procedure more efficient and patient-centered.

e) Security and Privacy: Hospital management systems have robust security features built into them to protect patient data and comply with privacy regulations. Role-based access controls, which limit entry to authorized persons only, are used by these systems to enforce strong procedures to manage access to sensitive information. By limiting access to and management of sensitive data to those who possess the necessary y authorizations, this framework strengthens the healthcare organization's overall security posture.

2. Preliminaries

To ensure the suggested system functions as intended, a thorough analysis was conducted both before and after the website was hosted. CSRF tokens were utilized to guard against cyberattacks on the internet page. We have ensured that vital information cannot be accessed without appropriate authorization. This solution can reach a larger user base than any mobile application because it only requires a browser to function. Additionally, due to its straightforward user interface, it is expected to have a lower churn rate. Because the proposed system is based on open-source tools and software that are built on reliable platforms, it may be easily improved and adjusted. To maximize flexibility and assure code reuse, it is divided into modules.

A patient is never aware of his exact illness, all that he knows is the symptoms, and visiting multiple doctors is necessary to diagnose the exact issue. Other websites prevent patients from being able to learn about potential illnesses they may have. Other Online platforms are only for searching for hospitals, medical stores, and clinics[2]. Certain programs restrict doctors from creating slots with their available times; instead, they only let users make online reservations or request information about doctors. Additionally, the hospital administration is unable to take control of operational matters such as appointment costs. So, in this system, we are addressing these issues.

3. Proposed Architecture

a) Admin: Admin is the superuser who manages everything.
b) Patient: The Patient is the one who is seeking medical attention or treatment for an illness or condition.
c) Doctor: Doctors play a crucial role in providing medical care and promoting patients' overall health and well-being.
User-friendliness is added to the front-end design. The patient will visit the website, register, and then sign in using their username and password. A doctor is also registered by the admin. The administrator is authorized to view patient and doctor records. An SQLite database, which is Django's default database is used here. Relational database interactions are made easier with the open-source SQLite database technology.
The server in this system maintains extensive data of patients, doctors, and appointments that are registered.

Fig shows the proposed architecture that involves 3 entities:

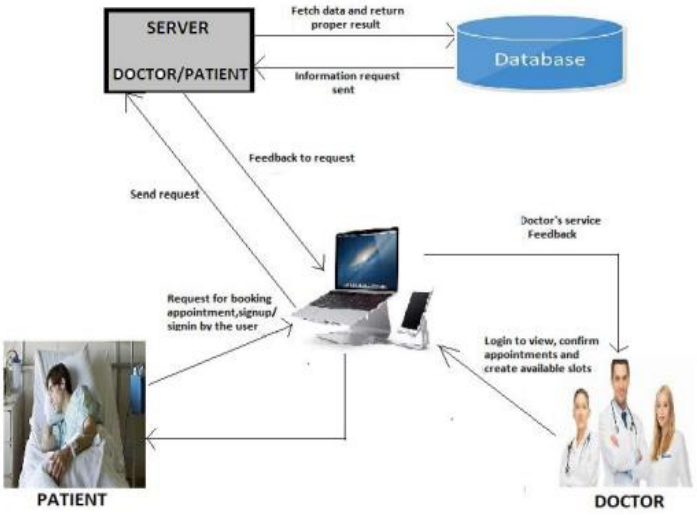

Figure 1: System Design for proposed system

4. Proposed Model

The three primary entities in the proposed system are Patient, Doctor, and Super Admin. They can access the website using any internet-connected device and web browser, which makes it more accessible and user-friendly. The patient begins by registering and providing basic details such as their name, phone number, email address, age, and gender. They also create a password for their account. Afterward, the patient uses their username and password to check in. They are then presented with a module of questions about their symptoms, answering with yes or no. Our machine learning-trained model determines which disease(s) the user is most likely to have based on the responses to questions concerning symptoms. It then provides the user with the name of the condition, its symptoms, and a confidence score. Furthermore, the patient will be able to see which doctor is assigned to him by the admin, and when the patient is discharged, he will be able to download the discharge slip also.
The doctors must register themselves with their basic information and specialty areas like cardiology, dermatology, etc. Then, the doctor needs to log in using the username and password. Then, he will be able to see the appointments he has however, he has the power to decline any appointment. He can see the details of the patients under him as well as the discharged patients.
The foremost entity in this system is the admin who manages everything. The admin has all the powers. He can register any patient or doctor. His approval is needed when the patient or doctor registers on a website only then they will be able to log in. He can see the history of every person whether it can be a doctor or patient. He will look for all the registered doctors on the website then as per the symptoms of the patient he will assign and book an appointment for the patient with the respective doctor.

So basically, he is managing all the backend and he is the backbone of DocLink

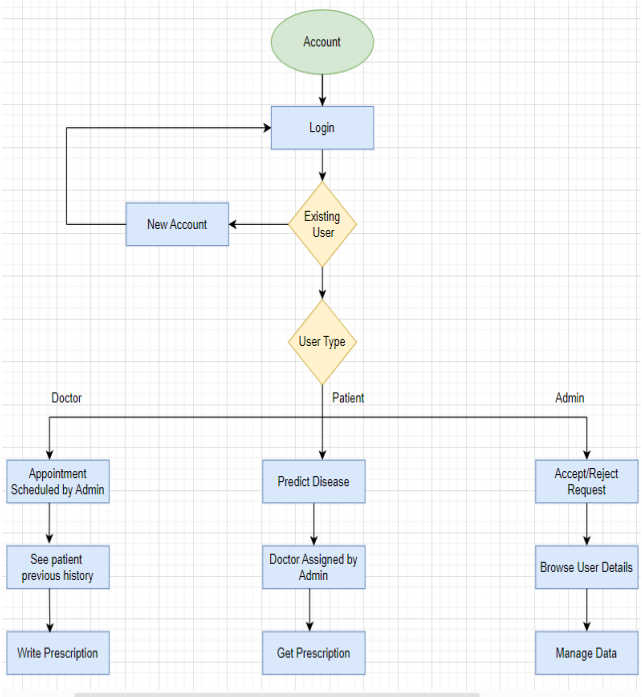

Figure 2: Flow Diagram of Proposed Model

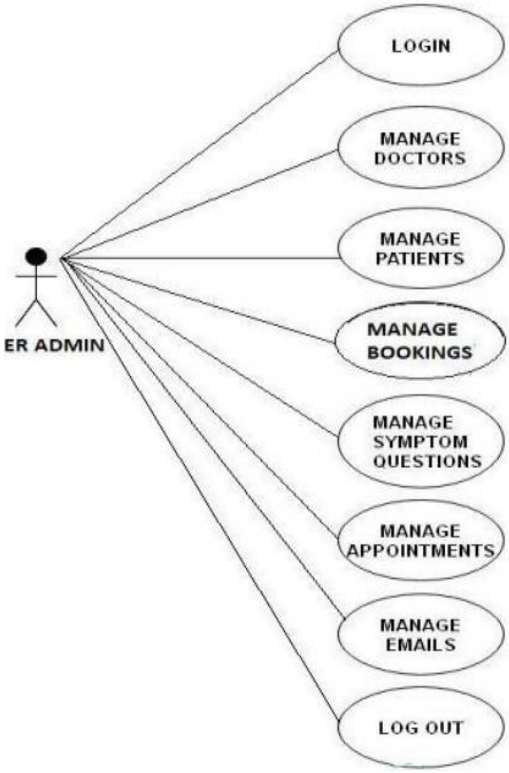

Figure 3: Use case diagram for Admin

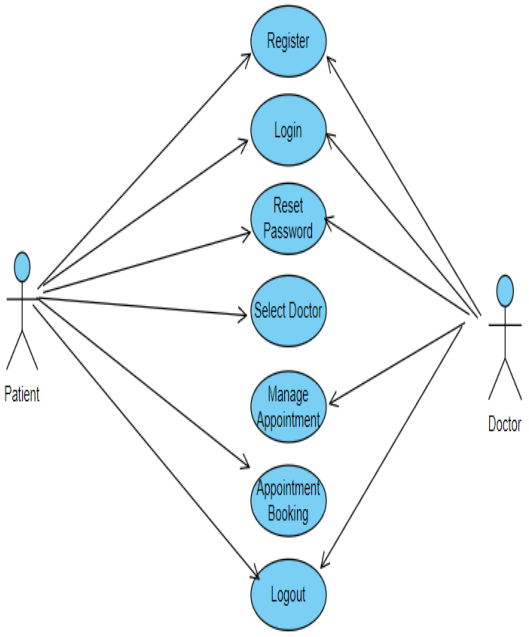

Figure 4: Use case diagram of Patient and Doctor

5. Experimental Studies

5.1. Technology Stack 1: Django

The open-source Python framework Django is used for developing websites and, in some cases, for building progressive web apps. Django is easy to learn and also allows us to write clean code, thus it improves code readability. It is maintained by the Django Software Foundation. Django was developed by Adrian Holovaty and Simon Willison in 2003 while working at the Lawrence Journal-World newspaper in Lawrence, Kansas, USA. Originally, Django was conceived to address the specific requirements and tight deadlines of web development within a news organization, specifically catering to the needs of the newsroom environment.

In July 2005, the Django framework was unveiled as an open-source project, taking inspiration from the jazz guitarist Django Reinhardt for its name. Its rapid rise in popularity can be attributed to the adoption of the "don't repeat yourself" (DRY) philosophy, emphasizing the significance of minimizing code redundancy, along with a dedicated focus on simplifying the process of constructing web applications. SQLite is the default database of Django. It is designed according to the MVT design pattern, a hierarchical technique in which the letter 'M' stands for model and represents the data it intends to display; This data is usually obtained from a database. A view is a request handler that responds to user requests by dynamically generating and providing the required templates and content. It is represented by the letter "V", which means "view". Another meaning of the letter 'T' is Template. The logic for displaying data on a web page is included in a text file called a template. Moreover, this framework is scalable, which enables the ease of managing user loads, this is a substantial advantage. Strong security features protect sensitive privacy-related data and user communications. Furthermore, it facilitates localization and internationalization, making it simple to adapt to different languages and locales. Additional benefits of using Django's integrated administrative

interface include ease of website content management and optimization of content management functions. It forms the foundation for the creation of content management systems, which optimize the process of producing digital material and tracking its production. With its essential components, this framework is the best choice for building a reliable e-commerce website. Whether utilized in social media applications, data analysis platforms, or government websites, Django's well-designed and practical approach consistently enhances the effectiveness of web development in various industries.

5.2. Technology Stack 2: React JS

The open-source web framework React was first created in JavaScript and is very good at creating single-page applications (SPAs). It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. It greatly facilitates the building of user interfaces, and it is overseen and maintained by Meta. React is a JavaScript framework, yet it works well with Typescript. Praised as the 'Future of Frameworks,' React is very much in use in the web development community, with a large user base and enormous growth potential. Its flexibility and compatibility are noteworthy since they make integration with Typescript easier and encourage further developments. React is a highly configurable front-end framework that is primarily used to enable rapid deployment on a range of cloud services and platforms, including AWS, Azure, Cloudflare, and Heroku. React, renowned for its component-based architecture, encourages the creation of modular and reusable user interface components, leading to a quicker interface-building process. Because of its usage of JSX, a JavaScript syntax that combines HTML elements, code is easier to read and element composition is simplified. React can be used with frameworks like Next.js to develop single-page, mobile, or server-rendered applications. Because React is only concerned with rendering components to the user interface and DOM, React applications often depend on libraries for routing and other client-side functionality.

React, a JavaScript library, is extensively applied in contemporary web development:

a) Creating User Interfaces: React is frequently used to create dynamic, interactive user interfaces for online applications, making it possible to render UI elements quickly.
b) Single-Page Applications (SPAs): By efficiently handling component states, it is the recommended option for developing SPAs and guarantees a smooth user experience.
c) Mobile App Development: Through frameworks like React Native, it extends its functionality to mobile app development, facilitating cross-platform app creation.
d) E-commerce Platforms: React's adeptness in managing complex and dynamic data makes it suitable for developing high-performance e-commerce platforms.
e) Interactive Web Applications: Its component reusability and performance optimizations make it suitable for creating highly interactive web applications with seamless user interactions.
f) Educational Platforms: React is widely used to create educational platforms for students and educators.

6. Limitations of proposed approach

a) Expense and Implementation at first time: Setting up an online hospital management system can be an expensive and time-consuming process. Initial expenses include data migration, staff training, hardware infrastructure, and software development.
b) Data security and privacy issues: Patient information is sensitive, and there are data security and privacy issues. To protect the privacy and accuracy of patient data, HMS systems need to comply with health care-related laws such as the Health Insurance Portability and Accountability Act (HIPAA).
c) Technical Problems and Downtime: Online HMS is just as vulnerable to technical problems, server outages or connectivity issues as any other technology. These can disrupt daily hospital operations and impact patient care.
d) Dependence on Internet Access: Online HMS requires Internet access. in locations with poor or unreliable Internet service, users may have trouble accessing and updating patient data quickly.

7. Conclusion

The hospital administration, doctors, and patients can all benefit greatly from this website. The program is easily accessible, and user-friendly. Both the doctor's and the patient's workload may be made easier by this system. Patients can wait for an appointment virtually instead of physically going. Patients will have a general notion of the type of sickness they may have, enabling them to look for specialists who specialize in that field. Physicians don't have to stress about running their clinics. Even though you are not visiting the clinic for an appointment, you can schedule one from anywhere at any time. The patient's time is saved as a result. The doctor can make better use of his time and increase his efficiency because he can access his schedule. Through this, the Hospital's productivity will increase. It will give them a good business analysis, and more control over the business aspects, such as determining the cost per visit and tracking the number of appointments scheduled, enabling them to make more informed business decisions. To develop a secure application and to protect the application from attacks, we implemented input validation, anti-forgery tokens, and other strong coding practices. To maintain ATO compliance continuous monitoring and scanning of the application are used to address vulnerabilities.

REFERENCES

Shafaq Malik, Nargis Bibi, Sehrish Khan, Razia Sultana, & Sadaf Abdul Rauf. (2017). Mr. Doc: A doctor appointment application system. Available at: https://arxiv.org/ftp/arxiv/papers/1701/1701.08786.pdf.
https://appdocpat.herokuapp.com/
https://docs.djangoproject.com/en/2.2/
https://devcenter.heroku.com/start
John Duckett, HTML and CSS: Design and Build Websites, 2015
Mastering HTML, CSS & JavaScript Web Publishing Paperback, 2016.
C Severance-IEEE Project Management journal, 2018.
Oana Frunza & Diana Inkpen. (2010). A machine learning approach for identifying disease-Treatment relations in short texts. Available at: https://ieeexplore.ieee.org/document/5560656.

Diwyanjalee, K. V. S., Widanapathirana, W. P. V. K., Perera, J. S., Warnakulasuriya, I. S. G., De Silva, D. I., & Samarasekara, H. M. P. P. K. H. (2022). Hospital management system (HMS). International Journal Of Engineering And Management Research, 12(5), 250-258.

Jan, Saeed Ullah, Muhammad Ishaq, and Abdul Aziz. "Hospital Management System." Available at SSRN 4350730 (2022).

Sathyabama Balasubramanian & Balaji Subramani. (2014). Symptoms based diseases prediction in medical system by using k-means algorithm. International Journal of Advances in Computer Science and Technology, 3(2), 123-128

Suryavanshi, Pulkit, Sagar Kaushik, and Deepali Dev. "An Automated Model for Booking Appointment in Health Care Sector." 2021 International Conference on Technological Advancements and Innovations (ICTAI). IEEE, 2021.

Dhiman, Ajay Kr, et al. "Telemedicine Solution using Django." International Journal of Engineering and Management Research (IJEMR) 10.2 (2020): 116-120.

Chauhan, Usha, et al. "Doctor Finder and Appointment Booking Website using DJANGO." 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM). Vol. 2. IEEE, 2022.

Rahman, Asifur. Web Based Hospital Management System. Independent University, Bangladesh, 2022.