

AI-Assisted Health Guidance and Medicine Support Portal

P. Narasimhaiah¹, Y. Sravani², S. Ashrafee³, S. Kousar Moin⁴, S. Shabnam⁵

¹Assoc.Prof, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P

²UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P

³UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P

⁴UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P

⁵UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P

Corresponding Author E-mail: yammanurusravani@gmail.com

Abstract

The availability of reliable medical information in time is still a burning issue in all medical systems on the globe, especially in the populations living in rural and semi-urban areas where there are limited specialists consulting patients. This paper describes the design and development of HealthCare AI, which is a web-based portal that will bring symptom-based health information, medicine information retrieval, prescription digitization, and doctor consultation together in one platform. The portal uses external Artificial Intelligence services that are accessed via application programming interfaces to analyse patient-reported symptoms and create structured medical recommendations. The tesseract engine is used to extract textual content by use of Optical Character Recognition technology which then has the ability to store and reference prescription images and medicine strip photographs digitally. The front-end will be developed using Next.js and React and the back-end will be using Python Flask with MongoDB as a data storage platform. Role-based access control provides a segregation between patient, doctor and administrator interaction to ensure that each type of user acts within well-defined limits. Functional testing and user feedback also allowed to evaluate the portal whereby the integrated approach minimizes the fragmentation that is often witnessed with the existing healthcare applications and enhance the user confidence in the preliminary self-assessment. This paper covers the system architecture, the module level design, implementation workflow and the results of the first deployment test.

Keywords: Healthcare portal; Symptom analysis; Optical character recognition; AI-based guidance; Medicine information system; Doctor Consultation.

1. Introduction

The provision of healthcare has already gone through a major change as digital technologies continue to rise, but a large part of the world population still experiences obstacles on their way to reliable medical advice. Estimates released by the World Health Organization indicate that about 50 percent of the world population is not receiving the full coverage of the basic health cares, and the situation is drastically experienced in the third world countries where doctor-to-patient ratio is uncommonly low. Under these conditions, people often turn to unsupported searches on the internet or unverified advice of people close to them and can lead to the misinterpretation of the symptoms and incorrect self-medication.

The availability of the digital health tools is further fragmented, which adds to the problem. The average user might require an app that will allow them to check their symptoms, a different application to get information on medications, another platform to make doctor appointments, and another application to monitor prescriptions and medication reminders. Not only does this disjointed ecosystem serve to add cognitive load to the user, but also introduces disparities in the healthcare journey, as the data produced in one platform often does not transfer to another.

There are a number of business applications like WebMD, Ada Health, and Practo that provide partial solutions in their fields. These platforms however work independently and do not give a single workflow that links symptom evaluation and medicine search, prescriptions management and doctor verification. Lack of this kind of continuity will hinder continuity of care, and the burden of continuity of care is left to the patient alone.

Based on this observation, this paper proposes HealthCare AI, a web-based healthcare support platform that consolidates various healthcare tasks in one platform. The portal uses external AI services to analyse the symptoms, uses Optical Character Recognition to digitize the prescriptions, has an organized dataset of medicine to find the information about a specific drug, and provides communication between doctors and patients by providing a consultation management module. The system architecture adheres to the philosophy of modular design which enables the development of every component, its testing and scaling.

The rest of this paper is divided into the following way. The Section 2 examines the related work and determines limitations of current approaches. Section 3 presents the methodology which includes the system architecture, module level design. Section 4 gives the results of implementation and user testing. Section 5 will end with a conclusion statement on the contributions and future improvement directions.

2. Literature Review

2.1 Existing Systems and Related Work

The problem of artificial intelligence and healthcare is a topic that has drawn attention to research over the last ten years. Topol (2019) has given a thorough overview of the way in which machine learning and deep learning methods are transforming clinical practice and highlighted that AI devices should act as decision-support tools and not as a substitute to trained clinicians. The findings in Esteva et al. (2017) indicated that convolutional neural networks could be used to form dermatologist-level in the classification of skin lesions when using photographic images, which sets a pace of AI-based medical image analysis.

In the realm of symptom assessment, Semigran et al. (2015) have carried out the analysis of 23 technology-based symptom-checking systems and discovered that though the identified systems were able to indicate the accurate diagnosis in their list of top options in approximately 58 percent of cases, their triage recommendations were overprotective and would have referred users to emergency care when self-care would have been appropriate.

More recent studies were done by Wolfram et al. (2016) who investigated rule-based and machine-learning-based methods of mapping patient-described symptoms to likely conditions and found that they were more accurate with contextual follow-up question included in the processing pipeline.

Digitization of handwritten prescriptions, laboratory reports and patient intake forms have been among the many applications of Optical Character Recognition in healthcare informatics. Smith (2007) reported the development of Tesseract OCR engine and its modification to extract text in multiple languages. Later work, such as that of Patel and Shah (2020), also revealed that Tesseract with pre-processing pipelines, which include noise removal and binarization, could achieve a character-level accuracy of over 90 percent on printed prescription documents. In spite of these personal gains, current practical healthcare solutions are in a very disjointed form. There are apps like Ada Health that specialize in symptom triage and apps like 1mg and PharmEasy specialize in ordering medicine and information. Practo is also a doctor-appointment booking application. All these platforms do not have a full workflow, which would enable the analysis of symptoms, search of medicine based on images and image recognition, prescriptions and doctor consultation on a single platform. This absence of integration is the heart of motivation of the current work.

2.2 Proposed System

The proposed HealthCare AI portal will resolve the limitations that were found in the review mentioned earlier, as it will integrate five fundamental healthcare functions into one online platform. The system has role-based access to three categories of users, which include, patients, doctors, and administrators. The portal allows patients to communicate with each other to describe the symptoms, find information about medicine, share the prescription images and consultations. Physicians examine information provided by a patient and compare it with the data generated by AI and give approved prescriptions and clinical records. Doctor onboarding is managed by the administrators, to avoid unauthorized access.

Analysis of symptoms is done by forwarding text inputted by the patient to the Gemini API, which passes the input to its underlying language model and provides a structured response of the input text, comprising severity classification and a follow-up suggestion. This design decision is not new to the industry practice, with pre-trained large language models accessed via APIs being presented as a viable alternative to creating and maintaining a custom machine-learning pipeline of medical text analysis.

Medicine information retrieval is performed in two channels. Users can search using a name of a medicine in which it will query a curated list with drug information (use instructions, dosage, contraindications and side effects). Alternatively, users can post pictures of medicine strips which are inputted through the Tesseract OCR engine to be converted to extract the name of the medicine and then compared to the data bank.

Digitization of prescription takes the same OCR-based route. Prescription images uploaded are identified to get names of medicines and instructions on dosage, which are stored in digital form and made available during doctor

appointments. One of the subsystems is a notification system that sends in-application notifications and optional SMS notifications to update users on the status of their consultations, prescription updates, and medications.

Methodology

3.1 System Architecture

The portal adheres to three tier architecture that includes front end presentation layer, backend application layer and external service integrations. Fig. 1 represents the general system architecture.

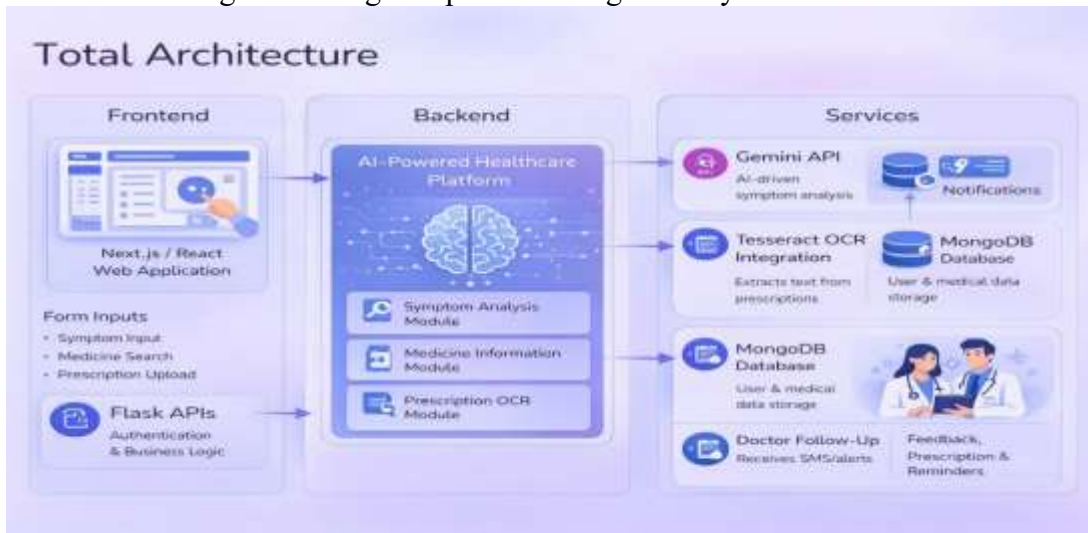


Fig. 1. System architecture of the HealthCare AI portal showing frontend, backend, and service integration layers.

The frontend layer is developed on Next.js and React, which offers server-side rendering to enhance the performance of first-time loads and components-based organization to ensure the modularity of the code. Users can use the system by interfaces which are based on the form where the user can enter his or her symptoms, search medicines, upload prescription and monitor consultation.

The backend layer is implemented using Python Flask, which is used to route the HTTP request, execute business logic, and also to communicate with the database and external services. RESTful API endpoints interface the front and back-end layers and each endpoint has a particular functional area e.g. authentication, symptom processing or consultation management.

MongoDB will be used as the main data storage, the choice of which is motivated by the flexibility of the schema-free document model, which can accommodate the diverse data structure types that arise in healthcare applications, such as user profiles, consultations records, prescription data, and notification logs.

Some of the external services offered are the Gemini API to analyse symptoms using AI, the Tesseract OCR to convert an image to text, and optionally Twilio to send notifications via SMS. These integrations are modelled behind service adapter modules in the background, and so separately, a service can be changed or upgraded without impacting the other parts of the system.

3.2 Module Design

Module 1: AI Health Interaction. The text is sent to the Gemini API with a structured prompt that requests the model to provide verbal advice on the severity of the symptoms on a three-point scale (mild, moderate, severe), make a preliminarily directive advice, and formulate clarifying follow-up questions with the ambiguous text. The backend reads the API response and displays it in the frontend formatted as a guidance card. In case of moderate or high severity of assessment, the system will show a noticeable warning to the user in order to prompt a doctor visit. Preoperative and operative nursing.

Module 2: The medicine information module is based on text as well as image queries. In the case of text querying, a search is conducted against a controlled medicine data base in MongoDB on a case-insensitive basis. Medicine records have their generic name, brand name, therapeutic category, dosage form, recommended dosage, contraindications, side effects, precautions in their fields. In case of querying an image, the image that has been uploaded is first processed with the OpenCV library on noise removal and contrast enhancement and then fed to the Tesseract OCR engine. Fuzzy string matching is then used to match the extracted text to the medicine dataset to allow OCR errors.

Module 3: Upload of Prescription, OCR Processing. Images uploaded as prescriptions are processed through a multi-stage processing pipeline in the form of JPEG, PNG, or PDF images. It is first turned into grayscale and adaptive thresholding to enhance readability of the text. Tesseract is then used to extract text and the scanned text is processed to give raw text which is then processed using pattern-matching rules to detect medicine names, dose amount, and frequency. The data about the structured prescription lies in MongoDB and is associated with the profile of the patient so that it can be accessibly retrieved during the consultations.

Module 4: Prescription and consultation development. The patients can then pick medicines and order them via the platform after being guided or informed about medicine with the help of AI or having reviewed information on medicine. The records of order entail the chosen medicines, quantities, and the respective pharmacy, and order position. At the same time, the system generates a consultation request with a connection to the patient profile, the history of symptoms, and the order information to provide the assigned doctor with the complete picture before they can offer any clinical contribution.

Module 5: Consultation with a doctor and prescription error check. Doctors will receive a dashboard with waiting consultation requests and the symptoms summary of the patient, AI-based assessment, and uploaded prescriptions. The doctor takes a diagnosis, prescribes drugs with a dose, and writes clinical notes after examining the information at hand. The chart record on consultation is indicated as a completed one, and the patient is notified about the final prescription.

Module 6: Notifications and following. The notification subsystem tracks the state changes in the platform, such as new consultation assignments, prescription updates, and changes in order status. The in-application

notifications are displayed in a special notification panel with the number of unread notifications. Taking important updates, the system will send SMS notifications via the Twilio API at will.

3.3 Technology Stack

Component	Technology
Frontend	Next.js, React
Backend	Python Flask, RESTful APIs
Database	MongoDB
AI Service	Google Gemini API
OCR Engine	Tesseract OCR with OpenCV
Notifications	In-app alerts, Twilio SMS (optional)
Development Tools	VS Code, Git, Chrome DevTools

Table 1. Technology stack employed in the HealthCare AI portal.

3.4 Data Flow and Processing Pipeline

The portal can have a summary of the end-to-end data flow in six stages. At the first stage, the user is authenticated and chooses a service module. The second stage is the process where the system authenticates and pre-processes the input, be it textual description of the symptoms or a picture uploaded. The third phase is to send the processed input to a specific service either the Gemini API to analyse the symptoms or the Tesseract engine to extract OCR. The fourth stage involves business logic implementation with the backend generating database entries of consultations and orders and the response payload. The fifth step is where the output is sent to the user via the frontend interface. The sixth stage generates the appropriate notifications depending on the actions done. This processing pipeline is shown in Fig. 2.



Fig. 2. Data flow pipeline showing the six processing stages from user input to notification delivery.

4. Results and Discussion

4.1 Implementation Overview

The portal was installed and tested on a development server with an Intel Core i5 with 8 GB RAM and Ubuntu 22.04. The MongoDB database was hosted locally in the course of the development, and the Gemini API was reached with a standard API key issued on the Google Cloud. The Next.js development server was used to serve the frontend application on port 3000, and the Flask backend was used to serve the port 5000.

4.2 Functional Testing Results

The individual modules were put through functional test through a combination of manual test cases and automated API test through Postman. Table 2 provides the results of the test in the five major modules.

Module	Test Cases	Passed	Pass Rate
AI Health Interaction	15	14	93.3%
Medicine Information	12	12	100%
Prescription OCR	10	8	80%
Doctor Consultation	8	8	100%
Notifications	6	6	100%

Table 2. Functional testing results across portal modules.

The AI Health Interaction module achieved a 93.3 percent pass rate, with one test case failing due to the Gemini API returning an overly generic response for a complex multi-symptom input. The Medicine Information module passed all test cases, confirming accurate retrieval from the curated dataset for both exact and partial name matches. The Prescription OCR module recorded an 80 percent pass rate, with the two failing cases attributed to poor image quality where handwritten text was heavily smudged, reducing Tesseract extraction accuracy. Both the Doctor Consultation and Notification modules achieved full pass rates.

4.3 User Interface Screenshots

Fig. 3 through Fig. 6 present representative screenshots of the portal interface captured during testing. These screenshots demonstrate the primary user interactions across symptom entry, medicine search, prescription processing, and doctor consultation workflows.

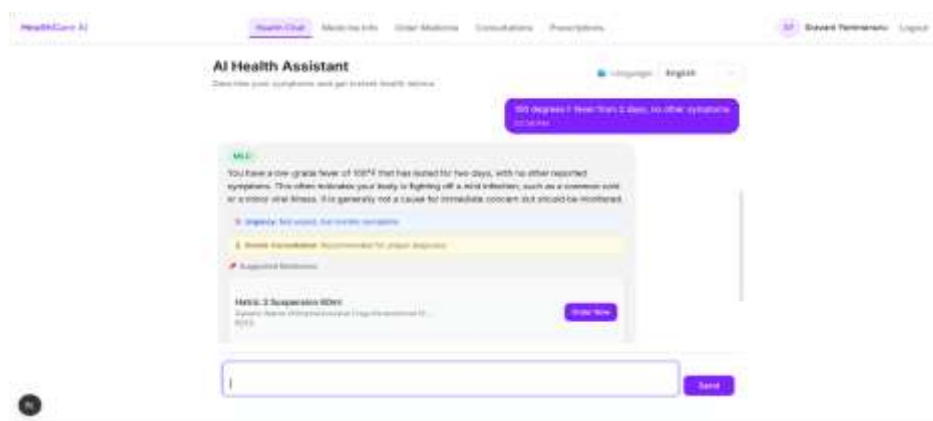


Fig. 3. Patient dashboard showing the symptom entry interface with AI-generated guidance panel.

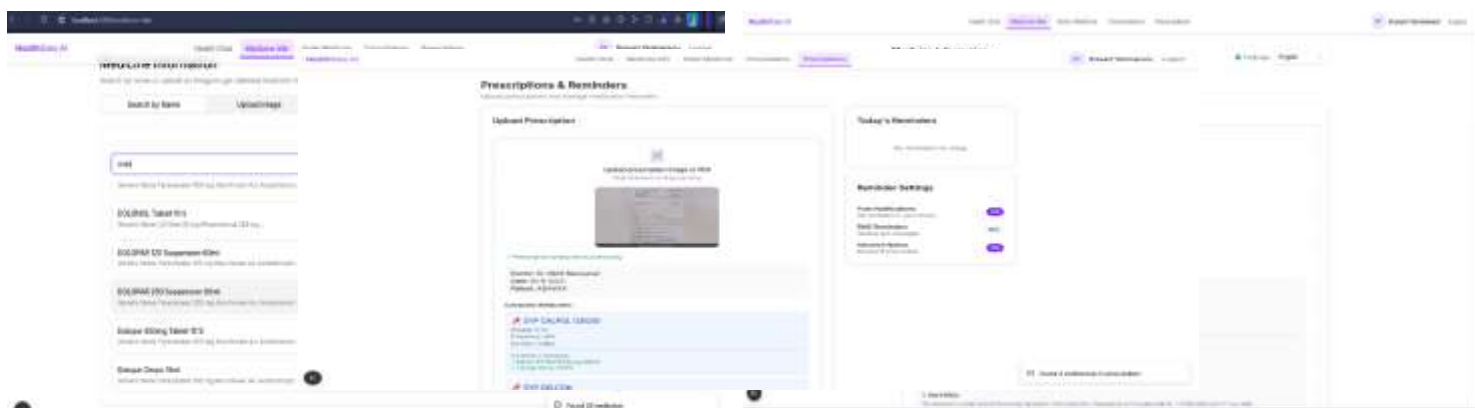
Fig. 4. Medicine search results displaying drug information including usage, dosage, and side effects.

Fig. 5. Prescription upload interface showing OCR-extracted text and parsed medicine details.

Fig. 6. Doctor consultation dashboard with patient symptom summary and prescription entry form.

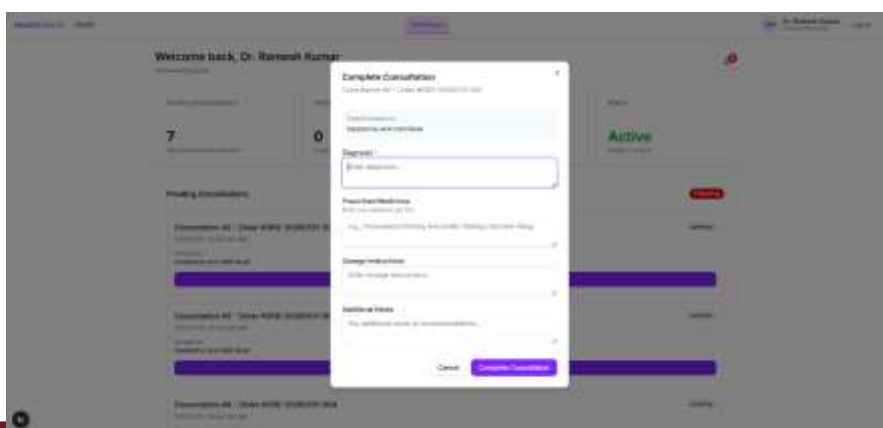
4.4 Performance Observations

The mean time of 2.3 seconds per query to respond to an AI symptom analysis module is acceptable when it comes to a web-based interaction. The OCR processing pipeline took 1.8 seconds to extract normal printed prescriptions and around 3.5 seconds to extract images that take a lot of pre-processing. The query of the



MongoDB database to do a search of the medicine area gave the results within less than 200 milliseconds when applied to the test dataset, which had 2,500 records of medicines. The initial user test was done on a sample of 20 participants (15 students and 5 faculty members of the Department of Computer Science and Engineering). The participants were requested to accomplish a set of tasks (symptom entry, medicine search, and prescription upload) and then rate their experience using the five-point Likert scale. The overall average level of satisfaction in relation to all the tasks was 4.2 on a 5-point scale, and the respondents have mentioned that the seamless nature of the platform was a major plus as compared to using multiple distinct applications.

4.5 Comparison with Existing Approaches



Feature	WebMD	Ada Health	Practo	HealthCare AI
Symptom Analysis	Yes	Yes	No	Yes
Medicine Info	Partial	No	Yes	Yes
OCR Prescription	No	No	No	Yes
Doctor Consult	No	No	Yes	Yes
Integrated Platform	No	No	No	Yes

Table 3. Feature comparison between existing platforms and the proposed HealthCare AI portal.

As indicated in Table 3, the proposed portal is the only platform whose features compare with those found in other platforms in that it incorporates all five healthcare functions in one application. Although a single platform might have a better point of view within its particular area, the integrated strategy embraced by HealthCare AI makes users not have to use several applications, and thus, the healthcare management process becomes less frictional.

5. Conclusion

In the present paper, the design, implementation, and evaluation of HealthCare AI, a unified web-based portal, including the synthesis of AI to analyse the symptoms, search the medicine information, and digitize the prescription process by using OCR technology, manage the doctor consultation process, and send the automatic notifications, were offered. The portal overcomes the lack of integration in current digital health care solutions by creating a unified workflow to lead the user through the steps of primary symptom evaluation to confirmed medical appointment.

The testing on the five main modules revealed a pass rate of between 80 and 100 percent, with prescription OCR module recording the lowest results because of the variation in image quality. Early user testing was able to produce an average satisfaction rating of 4.2 out of five and participants rated the integrated approach and the lack of platform switching to be beneficial.

The system is designed as modular, which allows making improvements in the future. Future plans also encompass support of wearable health monitoring gadgets to track vitals in real time, multilingual interaction via translation APIs and adoption of drug-drug interaction checking algorithms and placing the system on cloud infrastructure with scaling facilities. Also, with the help of a domain-specific language model trained on anonymized medical consultation data, the accuracy and contextual relevance of symptom-assessment AI-generated may be improved.

It shows that using pre-trained AI services by integrating APIs, existing technologies like Tesseract OCR, and current web frameworks can offer an opportunity to develop healthcare support systems that are both functional in scope and practical in implementation.

Author Contributions

Y. Sravani: System design, AI module integration, and manuscript preparation. S. Ashrafee: Frontend development and user interface design. S. Kousar Moin: Backend development and database design. S. Shabnam: OCR module implementation and testing. All authors reviewed and approved the final manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Semigran, H.L., Linder, J.A., Gidengil, C., & Mehrotra, A.(2015). Evaluation of symptom checkers for tone- opinion and triage examination study. *BMJ*, 31, h3480. [https// doi.org/10.1136/bmj.h3480](https://doi.org/10.1136/bmj.h3480)
2. Wolfram, D., Dierks, C., & Schulz, S.(2016). AI- rested symptom analysis systems A relative evaluation of rule- rested and machine knowledge approaches. *Journal of Medical Internet Research*, 18(6), e145. [https// doi.org/10.2196/jmir.5480](https://doi.org/10.2196/jmir.5480)
3. Patel, R., & Shah, V.(2020). OCR- rested tradition digitization Challenges and results in healthcare informatics. *International Journal of Health Informatics*, 8(2), 45- 58.
4. Smith, R.(2007). An overview of the Tesseract OCR machine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition(ICDAR)*, 629 – 633. IEEE [https// doi.org/10.1109/ICDAR.2007.4376991](https://doi.org/10.1109/ICDAR.2007.4376991)
5. Zhang, Y., Chen, M., Huang, D., Wu, D., & Li, Y.(2018). iDoctor Personalized and professionalized medical recommendations rested on crossbred matrix factorization. *unborn Generation Computer Systems*, 66, 30 – 35. [https// doi.org/10.1016/j.future.2016.07.002](https://doi.org/10.1016/j.future.2016.07.002)
6. Topol, E. J.(2019). *Deep drug How artificial intelligence can make healthcare mortal again.*
7. introductory Books.
8. World Health Organization.(2023). Universal health content – Fact distance. WHO [https//www.who.int/news-room/fact-sheets/detail/universal-health-coverage](https://www.who.int/news-room/fact-sheets/detail/universal-health-coverage)
9. Google.(2024). Gemini API attestation. Google Cloud. [https// ai.google.dev/ croakers](https://ai.google.dev/croakers)
10. MongoDB Inc.(2024). MongoDB attestation. [https// www.mongodb.com/docs/](https://www.mongodb.com/docs/)
11. Shortliffe, E. H., & Sepúlveda, M. J.(2018). Clinical decision support in the period of artificial intelligence. *JAMA*, 320(21), 2199 – 2200. [https// doi.org/10.1001/jama.2018.17163](https://doi.org/10.1001/jama.2018.17163)