iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85587

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

DiagnostiQ AI: Medical Report Intelligence and Simplification Platform

Mohamed Adhnaan J M[1], Mohan Kumar A[2], Mohammed Farhan Khan M N[3]
Department of Computer Science and Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India
Corresponding author: Mohamed Adhnaan J M | Email: [insert corresponding author email]

Abstract: ***Laboratory reports contain clinically useful measurements but are often difficult for patients to interpret because they combine abbreviations, units and laboratory-specific reference intervals. This paper presents DiagnostiQ AI, a web-based health-informatics platform that converts digital or scanned laboratory reports into structured biomarker results and patient-oriented explanations. The system uses PyMuPDF for digital PDF text extraction and Tesseract OCR as a fallback for scanned pages and images.***

A deterministic regular-expression parser, alias dictionary and reference-range knowledge base identify biomarkers, values and units, prioritising ranges printed on the source report. Abnormal markers and extracted alerts are combined using a transparent risk score, while the FLAN-T5 language model rewrites verified facts into plain-language explanations. The platform also supports English, Hindi and Tamil output, speech narration, report comparison, biomarker inspection, medication-interference screening, consultation preparation and guided breathing. A React 19 frontend communicates with a Flask REST backend. Functional testing covers document ingestion, parsing, classification, translation caching, comparison trends, fallback behaviour and endpoint responses.

The paper describes the architecture, decision logic, implementation and observed functional behaviour, while identifying limitations in document layout handling, reference-range personalisation and online translation. DiagnostiQ AI is designed as an educational aid and does not diagnose disease or prescribe treatment.

Keywords: Medical report analysis, biomarker extraction, optical character recognition, clinical text simplification, FLAN-T5, multilingual healthcare, risk assessment.

I. INTRODUCTION

Laboratory reports are primarily designed for clinical interpretation. Patients may receive complete blood count, glucose, thyroid or lipid-profile results containing unfamiliar abbreviations, units and reference intervals. A value outside a printed interval can cause anxiety, while searching the internet may lead to misleading or unnecessarily alarming conclusions. The problem is not simply access to a report; it is the gap between technical reporting and understandable health information.

Health-literacy research highlights the importance of enabling people to find, understand and use health information [1], [2]. Patient access to clinical notes can improve engagement, but technical language remains a barrier [2]. Recent work on simplifying radiology reports and the use of large language models in medicine shows promise while also identifying risks of omissions and confident inaccuracies [3]–[5].

DiagnostiQ AI addresses this gap through a two-layer design. A deterministic clinical layer extracts and classifies numerical findings, while a language layer explains only those verified findings. The platform is intended to support understanding and consultation preparation, not to replace professional diagnosis or treatment.

A. Objectives

- Accept digital PDFs, scanned or photographed reports, and pasted report text.
- Extract biomarker names, values and units, and compare them with available reference intervals.
- Classify reports and calculate an explainable Low, Medium or High risk level.
- Generate plain-language explanations and beginner-friendly summaries.
- Provide English, Hindi and Tamil output with optional speech narration.
- Compare two reports, screen selected medicine and supplement interactions, and prepare consultation notes.

©IJRASET: All Rights are Reserved

II. RELATED WORK

Patient-facing access to medical information is valuable, but the information must be understandable and actionable [3]. Nutbeam describes health literacy as a public-health goal and explains why technical information can limit effective participation [2]. Text simplification research provides methods for reducing linguistic complexity while preserving meaning [7].

Large language models have demonstrated the ability to produce accessible medical explanations, but evaluations also report limitations, including inaccurate or omitted information [4]–[6]. Research on instruction-finetuned models and text-to-text transfer learning informs the language component used in this work [8], [9]. The FLAN-T5 model is used as a controlled rewriting component rather than as the source of numerical clinical judgements.

The ingestion layer uses Tesseract OCR, whose capabilities and limitations have been documented for document analysis [10]. The platform also includes optional rules related to biotin interference in laboratory assays [11] and stress-associated blood-pressure variation [12]. Slow-breathing evidence motivates the guided breathing feature [13].

The research gap identified from these works is the need to combine reliable numeric extraction, controlled language simplification and practical patient-support tools in one workflow. DiagnostiQ AI separates deterministic interpretation from generated explanation and extends beyond a single explanation with comparison, screening and consultation-preparation features.

III. PROPOSED SYSTEM AND METHODOLOGY

DiagnostiQ AI is implemented as a browser-based single-page application with a React 19 frontend and Flask REST backend. The frontend provides report upload, analysis, comparison, consultation preparation, medication screening and breathing-coach views. The backend handles ingestion, parsing, range matching, alert extraction, risk scoring, narrative generation and translation.

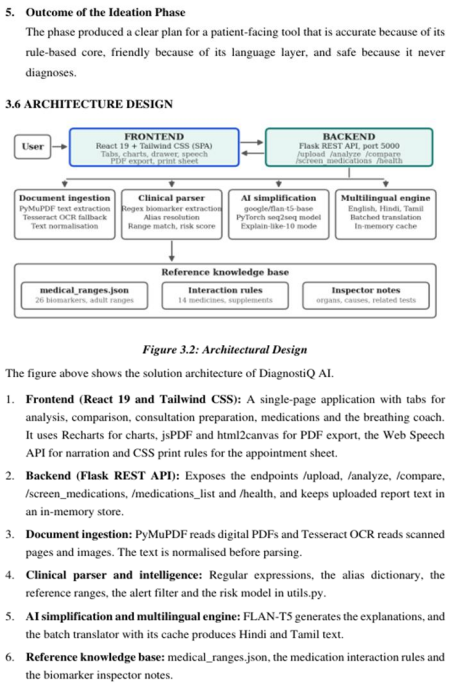


Fig. 1. DiagnostiQ AI system architecture (adapted from the project implementation).

A. Document Ingestion and OCR

The upload endpoint accepts supported PDF and image files, as well as pasted text. PyMuPDF extracts text from digital PDFs page by page. When a page contains insufficient text, it is rendered at higher resolution and passed to Tesseract OCR. Images are processed through OCR directly. Extracted text is normalised before parsing. If no readable text is found, the service returns a clear error rather than attempting to infer missing values.

B. Biomarker Parsing and Range Matching

A regular-expression parser identifies test names, numeric values, units and optional reference intervals. A dedicated pattern handles blood-pressure readings by separating systolic and diastolic values. An alias dictionary maps common forms such as FBS, TG and PCV to canonical biomarker keys. If a report provides its own reference interval, that interval takes priority over the built-in medical_ranges.json knowledge base. Unknown markers are not assigned guessed ranges.

C. Classification and Risk Assessment

The report classifier uses terms associated with common report groups, including lipid profile, complete blood count and diabetes-related testing. The alert extractor separates critical impressions from general alerts and ignores administrative lines. Let A denote the number of abnormal biomarkers, C the number of critical impressions and G the number of general alerts. The score is:

$$\textbf{Risk Score} = 2A + 3C + G$$

High: score ≥ 6; Medium: score ≥ 2; Low: otherwise.

For example, the project report describes a sample with nine abnormal markers, one critical impression and one general alert. Its score is $2(9) + 3(1) + 1 = 22$, classified as High. This example illustrates the rule; it is not a clinical outcome or a validated prediction.

D. AI-Based Explanation and Translation

The system constructs prompts from verified biomarker facts, including the measured value, unit, reference interval and deterministic status. The google/flan-t5-base model generates a plain-language explanation and can produce a simpler beginner-mode narrative. If the model cannot be loaded or fails, deterministic template sentences are returned. Hindi and Tamil translations are performed in batches and cached in memory to reduce repeated translation requests. Speech narration uses the browser Web Speech API.

E. Comparison and Patient-Support Modules

For two reports, the comparison module matches markers by canonical key and computes each value's distance from its reference interval. A marker is labelled Improved when its distance decreases, Worsened when it increases and No Change when it remains equal. Additional modules provide biomarker details, selected medicine/supplement interference rules, a printable consultation-preparation sheet and guided breathing protocols (box breathing, 4-7-8 breathing and paced breathing). These features are informational and are not treatment recommendations.

IV. IMPLEMENTATION

The backend is written in Python using Flask and Flask-CORS. PyMuPDF, pytesseract and Pillow support document reading and OCR. The language component uses Hugging Face Transformers with google/flan-t5-base, while deep-translator provides optional Hindi and Tamil translation. The frontend is built with React 19 and Tailwind CSS; Recharts displays marker visualisations, and jsPDF with html2canvas supports PDF export.

The main backend responsibilities are divided between the Flask application and utility functions for parsing, reference-range lookup, risk calculation, comparison and medication rules. The frontend manages the active report, selected language, beginner mode and navigation state. Components include UploadPanel, ResultView, BiomarkerDrawer, ConsultSheet, ComparePanel, MedicationPanel and BreathingCoach.

The application can be installed locally. The backend dependencies are installed in a Python virtual environment and launched through app.py. The frontend dependencies are installed with npm and the development server is started separately. Tesseract must be available for scanned reports; the FLAN-T5 weights are downloaded on first use. The project report specifies a server recommendation of at least 8 GB RAM, with 16 GB recommended, and notes that a GPU is not required.

V. TESTING AND RESULTS

Functional testing is organised around the core user workflow. The backend self-test is designed to run without downloading the language model, without network access and without an OCR installation. Browser-level checks use sample reports to verify the integrated workflow.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Test case	Expected behaviour
Digital PDF	Text layer extracted and analysed
Scanned PDF/image	OCR extracts readable text
Pasted text	Text analysed without file upload
Unsupported file	Rejected with HTTP 415
Unreadable/no lab values	Clear error with HTTP 422
Printed reference range	Source range takes priority
Aliases and BP	Aliases mapped; BP split into two values
Critical impression	Counted and included in risk score
Beginner mode	Simpler narrative produced
Hindi/Tamil	Batch translation and cache verified
Two-report comparison	Improved/Worsened/No Change trends
Model unavailable	Deterministic fallback returned

The documented test plan checks file handling, extraction, alias resolution, range priority, alert classification, language-mode behaviour, translation caching, comparison trends, fallback behaviour and endpoint status codes. The report does not provide a measured clinical accuracy rate, sensitivity, specificity or prospective patient study; therefore, no such performance claim is made here.

In the sample analysis described in the project report, the system identifies a diabetes report, extracts eleven markers, flags nine as outside their ranges, identifies one critical impression and one general alert, and calculates a High risk score of 22. The interface presents the report type, risk badge, summary, explanation, marker table and chart. Users can open a marker inspector, change language, compare reports and export a consultation-oriented result.

VI. DISCUSSION AND LIMITATIONS

The central design advantage is separation of responsibilities: deterministic code controls values, units, reference intervals and statuses, while the language model is restricted to explaining those facts. Prioritising the interval printed by the laboratory helps account for differences between laboratories. A transparent score makes the classification traceable, and fallback templates allow the application to remain usable when the language model is unavailable.

The current parser remains sensitive to unusual document layouts and poor-quality scans. Built-in reference intervals represent typical adult values and do not yet personalise for age, sex or pregnancy. FLAN-T5-base may produce awkward wording, so template-based statements remain important. Hindi and Tamil translation uses an optional online service and therefore requires connectivity; translated text may include laboratory values. The medication rule set is limited, and reports are held in memory rather than in a persistent patient account.

Most importantly, the risk score is a software prioritisation heuristic, not a clinically validated risk prediction model. A High label must not be interpreted as a diagnosis or emergency assessment. Users should confirm extracted values against the original report and consult a qualified healthcare professional.

VII. CONCLUSION AND FUTURE WORK

DiagnostiQ AI demonstrates a practical architecture for converting laboratory reports into structured findings and patient-oriented explanations. The system combines PDF extraction and OCR, deterministic biomarker parsing, reference-range matching, transparent rule-based scoring and controlled language simplification. Multilingual output, speech, comparison, biomarker inspection, medicine screening, consultation preparation and breathing guidance extend the workflow beyond a one-time explanation. Future work includes offline translation, additional Indian languages, age- and sex-specific reference intervals, clinician-reviewed expansion of medicine rules, improved layout-aware document understanding, stronger factual consistency checks for generated text, encrypted longitudinal storage, a clinician-facing view and a mobile application. Further evaluation should measure extraction accuracy across diverse laboratory formats and assess usability with appropriate clinical oversight.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

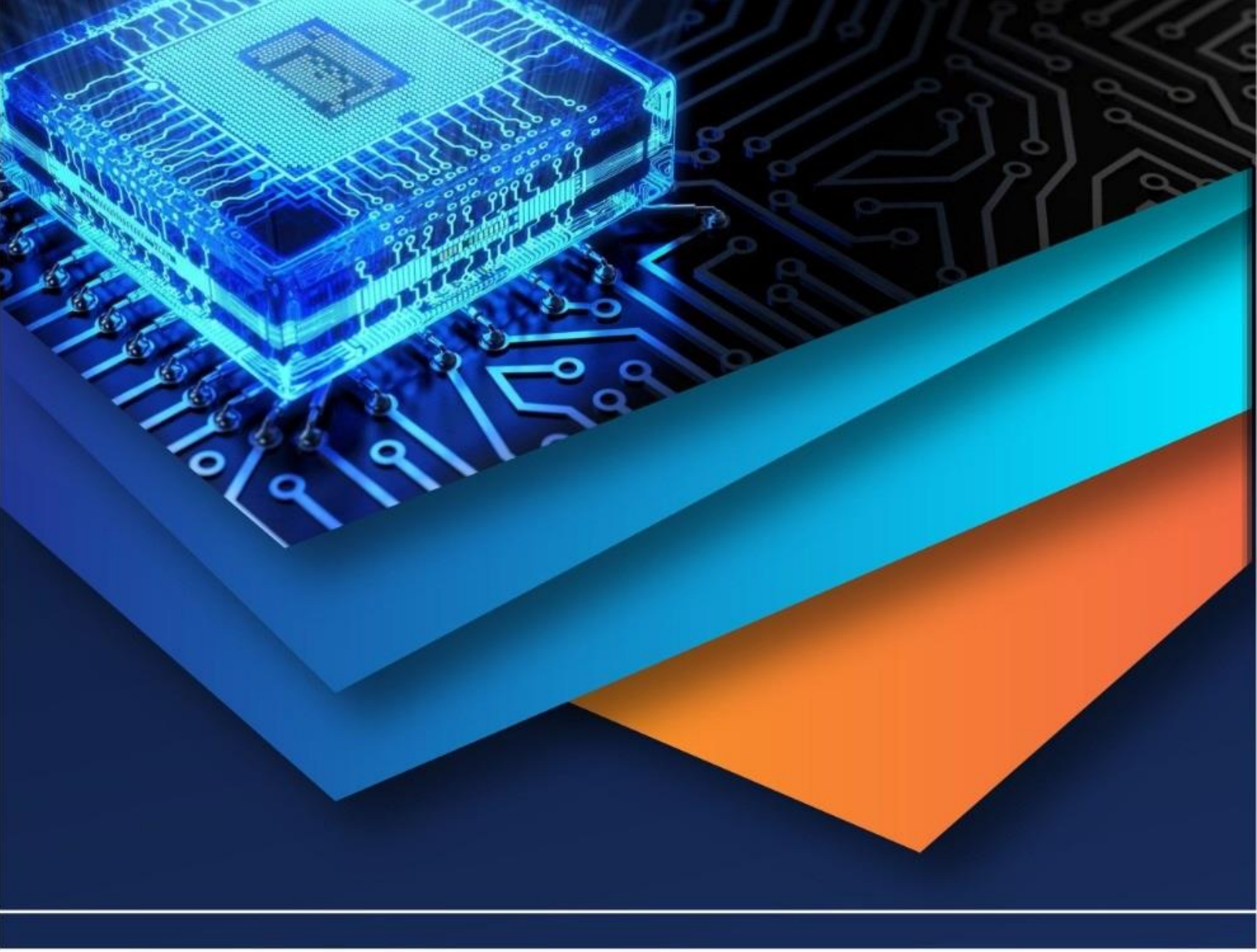
VIII. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, and acknowledge the guidance of the project supervisor and Head of the Department.

REFERENCES

[1] D. Nutbeam, “Health Literacy as a Public Health Goal: A Challenge for Contemporary Health Education and Communication Strategies into the 21st Century,” Health Promotion International, 2000.
[2] T. Delbanco et al., “Inviting Patients to Read Their Doctors’ Notes: A Quasi-experimental Study and a Look Ahead,” Annals of Internal Medicine, 2012.
[3] K. Jeblick et al., “ChatGPT Makes Medicine Easy to Swallow: An Exploratory Case Study on Simplified Radiology Reports,” European Radiology, 2024.
[4] P. Lee, S. Bubeck, and J. Petro, “Benefits, Limits, and Risks of GPT-4 as an AI Chatbot for Medicine,” New England Journal of Medicine, 2023.
[5] K. Singhal et al., “Large Language Models Encode Clinical Knowledge,” Nature, 2023.
[6] M. Balban et al., “Brief Structured Respiration Practices Enhance Mood and Reduce Physiological Arousal,” Cell Reports Medicine, 2023.
[7] A. Siddharthan, “A Survey of Research on Text Simplification,” ITL - International Journal of Applied Linguistics, 2014.
[8] H. W. Chung et al., “Scaling Instruction-Finetuned Language Models,” arXiv:2210.11416, 2022.
[9] J. Wei et al., “Finetuned Language Models Are Zero-Shot Learners,” International Conference on Learning Representations (ICLR), 2022.
[10] R. Smith, “An Overview of the Tesseract OCR Engine,” International Conference on Document Analysis and Recognition (ICDAR), 2007.
[11] D. Li et al., “Association of Biotin Ingestion With Performance of Hormone and Nonhormone Assays in Healthy Adults,” JAMA, 2017.
[12] T. G. Pickering et al., “How Common Is White Coat Hypertension?” JAMA, 1988.
[13] M. A. Russo, D. M. Santarelli, and D. O’Rourke, “The Physiological Effects of Slow Breathing in the Healthy Human,” Breathe, 2017.
[14] A. Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems (NeurIPS), 2017.
[15] C. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” Journal of Machine Learning Research, 2020.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)