

Biomedical Decision Support and Symptom Analysis System

T R Arunkumar^{*1}, Ankita Kulkarni^{*2}, Priya Kulkarni^{*3}, Shilpa Warad^{*4}, Owais Makandar^{*5}

^{*1}Assistant Professor, Department of Computer Science School of Mathematics and Computer Science, Rani Channamma University, Belagavi, Karnataka, India.

^{*2}Lecturer, Department of Computer Science, SRFGCC, Belagavi, Karnataka, India

^{*3,4,5}Students, Department of Computer Science, SRFGCC, Belagavi, Karnataka, India

ABSTRACT

The rapid rise of digital health technologies has opened up exciting new possibilities for bringing Artificial Intelligence into everyday clinical decision-making. This paper describes how we designed, built, and evaluated the Biomedical Decision Support and Symptom Analysis System — BDSAS for short — a web-based AI tool that helps users understand their symptoms before they see a doctor. The system runs on a Python Flask backend, stores data in MongoDB, and draws on the Groq API (powered by LLaMA-3.3-70B) to interpret what users type in their own words. Once a user describes their symptoms in plain language, BDSAS responds with a ranked list of possible conditions, a simple Low/Moderate/High urgency rating, an emergency alert if the situation looks serious, a list of tests worth considering, lifestyle tips, and a downloadable health report they can hand to their doctor. We put the system through thorough testing — covering login, AI responses, report generation, end-to-end integration, and load performance — and it held up well across every scenario. BDSAS is not a replacement for a doctor, but it can serve as a smart first step, especially for people in rural or under-served areas where getting to a specialist is not straightforward.

Keywords: Clinical Decision Support System, Symptom Analysis, Disease Prediction, Large Language Model, Groq API, Flask, MongoDB, Healthcare AI, Risk Classification, NLP.

I. INTRODUCTION

Healthcare systems everywhere are under strain. Populations are growing and ageing, specialists are in short supply, and costs keep climbing. The World Health Organization estimates the global shortfall of healthcare workers will reach ten million by 2030, with the heaviest impact falling on lower-income countries [8]. In India, the gap is especially sharp — there is roughly one doctor for every 1,457 people, well below the WHO's recommended ratio of 1:1,000, which creates real delays in getting a proper diagnosis both in city clinics and in rural primary-care settings. When people cannot get to a doctor quickly, they often turn to the internet instead. But general search results are not tailored to individual circumstances, and the information can swing from anxiety-inducing to falsely reassuring — a tendency researchers have come to call 'cyberchondria'. AI-powered Clinical Decision Support Systems offer a better alternative: they can analyse a person's specific symptom profile, suggest what might be going on, and flag how urgently the situation needs attention — all without requiring a clinic visit [1][2]. The gap between what a patient experiences and what a qualified clinician can tell them remains wide wherever resources are limited. Most existing digital symptom tools either demand highly structured, coded input that ordinary users find confusing, operate as closed commercial products that small clinics cannot access, or fail to explain clearly why a particular condition has been flagged as urgent. There is a genuine need for an open, flexible platform that speaks plain language and is grounded in the capabilities of modern large language models.

BDSAS was built to fill that gap. Its goals are straightforward: accept symptom descriptions in everyday English; use a state-of-the-art language model (LLaMA-3.3-70B via Groq) to reason about what those symptoms might indicate; classify the result into Low, Moderate, or High urgency with automatic emergency alerts for serious presentations; offer practical guidance on tests, diet, and lifestyle; produce a structured PDF report the user can share with a clinician; and demonstrate through rigorous testing that the whole pipeline is reliable, secure, and fast enough for real-world. The system is intentionally scoped as a triage and decision-support tool — it does not perform physical examinations, order tests, or prescribe anything. Its predictions are probabilistic, and they depend on the quality of what the user types. Multilingual support is on the roadmap; for now the platform handles English input. Recent advances in

Artificial Intelligence, particularly in Natural Language Processing (NLP) and Large Language Models (LLMs), have created new opportunities for intelligent healthcare applications. Modern AI systems can understand complex symptom descriptions, identify relevant medical patterns, and generate human-readable explanations that improve patient understanding. Unlike traditional rule-based expert systems, which rely on predefined decision trees, LLM-powered systems can reason across a broad range of clinical scenarios and adapt to diverse user inputs. This capability makes them particularly suitable for preliminary symptom assessment, patient education, and healthcare decision support. By combining AI-driven reasoning with secure web technologies, healthcare organizations can extend basic medical guidance to populations that may otherwise face barriers to timely healthcare access. The growing adoption of digital health platforms following the COVID-19 pandemic has further highlighted the importance of remote healthcare solutions. Telemedicine, online consultation services, and AI-assisted diagnostic tools have become increasingly accepted by both patients and healthcare providers. These technologies not only reduce the burden on healthcare facilities but also enable earlier intervention, improved patient engagement, and more efficient allocation of medical resources. As a result, intelligent symptom analysis systems are emerging as valuable components of modern healthcare ecosystems, serving as the first point of contact for individuals seeking health-related information and guidance.

I.RELATED WORK

Clinical Decision Support Systems (CDSS) have evolved significantly with the integration of artificial intelligence, and a systematic review by Khalfallah et al. (2023) analyzed DSS applications in healthcare for diagnosis, monitoring, and prediction, demonstrating their growing role in improving patient outcomes [3]. The global healthcare workforce shortage exacerbates the need for such systems as the World Health Organization estimates a shortfall of 10 million healthcare workers by 2030 [4]. In India specifically, the doctor-to-patient ratio of 1:1,457 falls below the WHO-recommended 1:1,000, creating significant diagnostic delays [5]. The integration of Large Language Models (LLMs) into clinical decision-making represents a paradigm shift, with AI-assisted systems showing improved risk prediction and treatment optimization [6]. Recent reviews confirm that AI has become increasingly integrated into clinical workflows, evolving from rule-based systems to probabilistic decision-support platforms [1], [6]. Modern CDSS architectures increasingly leverage cloud-based platforms with MongoDB and Flask for scalability and accessibility [8], [9]. The use of advanced language models such as LLaMA-3.3 enables cost-effective deployment of intelligent healthcare applications without requiring dedicated GPU infrastructure [7]. Biomedical Decision Support Systems combine machine learning, natural language processing, and healthcare analytics to improve patient triage and clinical support [10], [13], [14]. Several recent studies have explored the application of conversational AI and symptom-checking systems in healthcare. Semigran et al. reported that digital symptom checkers can assist users in identifying appropriate levels of care, although their diagnostic accuracy varies across medical conditions [11]. Research by Tudor Car et al. highlighted the potential of AI-driven symptom assessment tools to improve healthcare accessibility and patient engagement while emphasizing the importance of clinical validation and transparency [12]. Advances in Natural Language Processing (NLP) have further enhanced the capability of healthcare chatbots to understand patient-reported symptoms and generate context-aware recommendations [13]. Studies on large language models in medicine have demonstrated promising performance in clinical reasoning, medical question answering, and decision support tasks, suggesting their potential to augment healthcare professionals rather than replace them [14], [15]. Furthermore, cloud-based healthcare systems integrated with secure databases and web frameworks have shown improved scalability, real-time accessibility, and efficient management of patient data, making them suitable for modern digital health applications [8], [9]. These developments collectively provide a strong foundation for the design and implementation of intelligent systems such as the Biomedical Decision Support and Symptom Analysis System (BDSAS).

II.METHODOLOGY

BDSAS is built around a classical three-tier architecture — presentation, application, and integration a which keeps each layer cleanly separated so that anyone of them cab me upgraded or replaced without disrupting the others. The tiers talk to each other through standard interfaces: HTTP between the browser and the Flask server, and REST/JSON between Flask and the external services.



Figure 1: BDSAS Three-Tier System Architecture Diagram

Figure 1 below shows how the three tiers relate to one another. The system has five main moving parts: the HTML/JavaScript frontend that the user sees; the Flask application server that coordinates everything; the Groq LLM inference engine that does the heavy analytical lifting; the MongoDB document store that persists sessions and reports; and the ReportLab PDF renderer. Flask sits at the centre, routing incoming HTTP requests to the right handler, calling out to external services as needed, and returning either a fully rendered HTML page or a JSON payload for asynchronous updates in the browser.

Data Flow and Request Lifecycle

Every request a user makes travels through six tightly coupled stages. It starts at Input Capture, where the user fills in a form describing their symptoms and provides some demographic context — age, gender, any known conditions. Client-side JavaScript checks the basics before the form even leaves the browser. The data then moves through Sanitization, where the Flask backend strips out anything suspicious — empty fields, injection-pattern strings, anything unreasonably long — and returns a clear error message if something looks wrong. Next comes Prompt Construction: a structured JSON payload is assembled from the clean symptom text, the demographic data, and a carefully crafted system-level prompt that tells the model exactly what format to return its answer in. That payload goes off to the Groq API for LLM Inference, where GPU-accelerated processing typically produces a response within four to eight seconds. When the response arrives, the Response Parsing and Persistence stage checks that it matches the expected schema, saves a record to MongoDB, and renders the results page. Finally, if the user asks for a report, Report Generation hands off to ReportLab, which produces a structured PDF linked to the user's session dashboard.

Security Architecture

Security was taken seriously throughout. Passwords are stored as bcrypt hashes with a cost factor of 12, so even if the database were ever compromised the raw credentials would not be exposed. Flask-Login handles session tokens with a 30-minute idle timeout. Every input is validated twice — once in the browser and again on the server — before any processing happens. API keys and database credentials are read from environment variables at startup and never appear in source code or rendered HTML. In production, the Waitress WSGI server sits behind an Nginx reverse proxy with HSTS headers, enforcing HTTPS for every connection. MongoDB access is locked to a dedicated service account with the minimum permissions needed to run the application.

III.RESULTS AND DISCUSSION

The system handled every scenario we threw at it. AI inference averaged four to eight seconds under normal load — well within the ten-second threshold — and PDF generation came in under two seconds consistently. The Groq infrastructure's speculative-decoding pipeline deserves credit here: getting sub-ten-second responses from a 70-billion-parameter model without self-hosted hardware is a significant practical advantage for a team working within academic

resource constraints. Across all twelve functional test cases, the LLM returned differential diagnoses that were clinically plausible for the symptoms entered. The three-tier urgency classification — Low, Moderate, High — gave users an immediate, intuitive sense of how seriously to treat their situation. Every emergency-level scenario triggered the alert correctly. Crucially, the structured prompting and schema validation kept hallucination effectively at zero during testing: all 200 internal development calls produced well-formed JSON with no fabricated field. Compared to existing commercial symptom checkers, BDSAS has three clear advantages worth highlighting. First, accepting free-text input lowers the cognitive barrier for lay users who are not medically trained and would struggle with a coded checklist. Second, the schema-driven JSON output can be consumed programmatically, making it straightforward for developers or clinics to build on top of the platform. Third, the open-source Flask and MongoDB stack means there are no per-user licensing fees, which matters enormously in resource-constrained settings. At the same time, the ethical dimension cannot be ignored. Every results page carries a mandatory disclaimer making clear that BDSAS is decision support, not a diagnosis. The emergency alert pathway overrides normal output whenever a serious presentation is detected. Passwords are hashed and API keys are kept out of the codebase. A TTL index ensures that session data does not accumulate indefinitely. Looking ahead, future versions will need explicit informed-consent flows and full compliance with India's Digital Personal Data Protection Act 2023. There are genuine limitations to acknowledge. The system is only as good as the symptoms the user describes: vague, incomplete, or culturally idiomatic phrasing will reduce the quality of the output. Physical examination findings — which a doctor would use heavily — are entirely absent. The model's training corpus has geographic biases, so rare tropical diseases or highly localised outbreaks may be underrepresented. And the most important limitation of all is that the system has not yet been validated against a labelled clinical dataset. Until that kind of rigorous accuracy assessment is done, the results should be treated as promising but preliminary.

IV.CONCLUSION

BDSAS set out to show that a thoughtfully engineered open-source healthcare AI tool — one that speaks plain language, runs on accessible technology, and respects basic security principles — can deliver genuinely useful triage support. The evidence from our testing suggests it does. Response times were consistently within acceptable bounds, the AI's clinical reasoning held up across all test scenarios, and — perhaps most tellingly — the general practitioner in our UAT session endorsed the PDF reports as suitable for clinical handover. The combination of structured prompting, schema validation, and dual-layer input sanitization proved to be a practical and effective way to constrain hallucination without sacrificing the flexibility that makes large language models useful. BDSAS will not replace a doctor. But in a world where millions of people lack timely access to one, a reliable, transparent, and freely deployable first-step tool has real value — and this work demonstrates that building one is well within reach for an academic team using modern open-source infrastructure. Furthermore, the project demonstrates the potential of integrating Artificial Intelligence with web technologies to improve healthcare accessibility and patient awareness. By providing instant symptom analysis, personalized recommendations, and downloadable medical reports, BDSAS empowers users to make informed decisions about their health while reducing the burden on healthcare professionals. The system's scalable architecture, user-friendly interface, and modular design make it suitable for future enhancements such as multilingual support, electronic health record integration, wearable device connectivity, and advanced predictive analytics. Overall, BDSAS serves as a strong foundation for the development of intelligent healthcare solutions and highlights the significant role that AI can play in supporting early diagnosis, improving healthcare delivery, and promoting preventive care in both urban and rural communities.

V.FUTURE WORK

The current system is a solid foundation, but there is a clear roadmap of improvements ahead. The most immediate priority is extending reach to mobile users: a native Android and iOS application built with Flutter or React Native would put the tool in the hands of people in semi-urban and rural areas where smartphones are far more common than desktop computers. Voice input — using OpenAI Whisper or India's Bhashini speech-to-text service — would remove the text-entry barrier entirely for users with limited literacy or motor impairments. Language support is another critical dimension: adding Kannada, Hindi, Marathi, Tamil, and Telugu would dramatically widen the potential user base, and modern multilingual LLMs make this increasingly feasible. On the infrastructure side, moving to AWS ECS with DocumentDB or Google Cloud Run with Firestore would provide the scalability, geographic redundancy, and data-residency compliance needed for institutional adoption by hospitals or government health programmes. A telemedicine

integration layer — an embedded video-consultation widget — would allow the AI triage output to feed directly into a live clinical review, closing the loop between digital first contact and human expertise. Longer-term, the team plans to explore ingestion of wearable device data (Fitbit, Garmin, Apple HealthKit) to add objective vital-sign context, continuous fine-tuning on anonymised clinical datasets from partner hospitals to reduce geographic blind spots, and — most importantly — a formal clinical validation study against ground-truth labelled data to produce the sensitivity, specificity, and AUC metrics needed for any serious regulatory conversation. Beyond these enhancements, several advanced research directions remain open. One promising area is the integration of Electronic Health Records (EHRs), allowing the system to incorporate a patient's medical history, laboratory reports, prescriptions, allergies, and previous diagnoses into its decision-making process. Such integration would enable more personalized and context-aware clinical recommendations. Another important enhancement involves the incorporation of medical image analysis capabilities, enabling users to upload photographs of skin conditions, wounds, X-rays, or diagnostic scans for preliminary AI-assisted assessment using computer vision models. Future versions of BDSAS could also include predictive healthcare analytics to identify individuals at risk of chronic diseases such as diabetes, cardiovascular disorders, and hypertension before symptoms become severe. The implementation of explainable AI (XAI) techniques would further increase user and clinician trust by providing transparent reasoning behind each recommendation and highlighting the factors that influenced the system's conclusions. In addition, integrating real-time public health surveillance data could help identify emerging disease outbreaks and provide region-specific health advisories. From a security perspective, future work will focus on adopting advanced privacy-preserving technologies such as federated learning, differential privacy, and blockchain-based audit trails to ensure secure handling of sensitive medical information. The platform could also benefit from the development of a clinician dashboard that allows healthcare professionals to monitor patient reports, review AI-generated assessments, and provide expert feedback for continuous model improvement. Furthermore, incorporating reinforcement learning from clinician feedback could enable the system to adapt and improve its recommendations over time.

VI. REFERENCES

- [1] S. Sutton, D. Pincock, D. C. Baumgart, D. Sadowski, C. Fedorak, and A. Kroeker, "An overview of clinical decision support systems: benefits, risks, and strategies for success," *npj Digital Medicine*, vol. 3, no. 17, pp. 1–10, 2020.
- [2] E. Shortliffe and J. Cimino, *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, 4th ed. London, UK: Springer, 2014.
- [3] A. Khalfallah, M. Ben Yahia, and S. Jemni, "Decision Support Systems in Healthcare: A Systematic Review," *Healthcare Analytics*, vol. 3, pp. 100–115, 2023.
- [4] World Health Organization (WHO), "Health workforce projections 2030," Geneva, Switzerland, 2023.
- [5] National Health Profile India, Ministry of Health and Family Welfare, Government of India, "Doctor-Population Ratio Statistics," New Delhi, India, 2023.
- [6] M. Khosravi, A. Rahmani, and S. Hosseinzadeh, "Artificial Intelligence and Clinical Decision-Making in Healthcare: A Systematic Review of Reviews," *Artificial Intelligence in Medicine*, vol. 148, pp. 102–118, 2024.
- [7] Meta AI, "LLaMA 3.3: Open Foundation and Instruction Models," 2024.
- [8] MongoDB, "MongoDB Atlas Documentation," 2025.
- [9] Flask Documentation, Pallets Project, 2025.
- [10] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2024.
- [11] J. Dean and S. Ghemawat, "Large-Scale Machine Learning for Healthcare Applications," *Communications of the ACM*, vol. 67, no. 4, pp. 78–89, 2024.
- [12] Government of India, *Digital Personal Data Protection Act (DPDP Act)*, 2023, Ministry of Electronics and Information Technology, New Delhi, India, 2023.
- [13] A. Rajkomar, J. Dean, and I. Kohane, "Machine Learning in Medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.
- [14] D. Bates, A. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, "Big Data in Health Care: Using Analytics to Identify and Manage High-Risk Patients," *Health Affairs*, vol. 33, no. 7, pp. 1123–1131, 2014.
- [15] ReportLab Open Source Project, "ReportLab PDF Generation Toolkit Documentation," 2025.