Care-Chat

Dnyati Khaladkar[1] Tareeka Rane[2] Anjali Jadhav[3] S Rahul Rao[4] Dr. Rama Bansode[5]
[1,2,3,4,5]Department of Computer Applications (MCA)
[1,2,3,4,5]P.E.S.'s Modern College of Engineering, Pune, Maharashtra,, India

Abstract — Medical reports and prescriptions are essential for healthcare continuity, but they are commonly written for clinical audiences and may be difficult for patients to interpret independently. Care-Chat is a lightweight, privacy-preserving application designed to explain uploaded laboratory reports and prescriptions in plain language. The proposed system accepts scanned or photographed documents, extracts their content using Optical Character Recognition (OCR), and uses a locally hosted large language model (LLM) to generate explanations through a conversational interface. Unlike cloud-based medical AI systems, Care-Chat is designed so that the uploaded document, extracted text, conversation history, and memory remain on the user's device. The system applies explicit safety constraints that limit the model to explanation and prohibit diagnosis or treatment recommendations. A two-tier memory mechanism based on TF-IDF retrieval and a persistent patient-memory summary supports contextual conversations across recent interactions and multiple uploaded documents. The proposed architecture combines local processing, OCR, lightweight language-model inference, safety-constrained prompting, and conversational memory into a consumer-hardware-oriented solution.

Keywords: Medical Document Understanding, Healthcare AI, Large Language Models, OCR, Privacy-Preserving AI, Local Inference, Health Literacy, Conversational AI

I. INTRODUCTION

Medical documentation, including laboratory reports, diagnostic summaries, and prescriptions, is primarily written for clinical audiences. Such documents may contain abbreviations, technical terminology, numerical measurements, reference ranges, and instructions that patients cannot independently interpret. The resulting gap between receiving a medical document and understanding its meaning forms the central motivation for Care-Chat.

Care-Chat is a locally running software application through which a user can upload a laboratory report or prescription as an image or PDF and receive a plain-language explanation through a conversational interface. OCR is used for image-based content, while a locally hosted LLM generates explanations. A central architectural requirement is that the document and conversation data should not be transmitted beyond the user's own device, following the data-localization principle discussed in the privacy literature [3].

The source report describes patient demand for explanations of laboratory results and identifies limited consultation time in high-volume healthcare settings as an additional motivation. Care-Chat is therefore positioned as a supplementary explanation tool rather than a replacement for professional medical judgment.

The main objectives are to extract text from medical documents, provide safe plain-language explanations [1], maintain relevant conversational context, protect sensitive information through local processing [2], [3], and provide an approachable interface that communicates system limitations clearly.

II. LITERATURE REVIEW AND RELATED WORK

The literature reviewed for Care-Chat can be organized into four major directions: patient-friendly medical report generation [1], reliability and self-correction in medical LLM workflows [1], fully offline healthcare AI [2], and privacy-preserving healthcare chatbots [3]. A fifth supporting direction is document OCR and conversational retrieval. These areas solve different parts of the overall problem and together motivate the proposed architecture.

A. Agentic LLM Workflows for Patient-Friendly Reports

Sudarshan et al. investigated "Agentic LLM Workflows for Generating Patient-Friendly Medical Reports." The source report highlights an important baseline: zero-shot prompting of a general-purpose LLM produced patient letters that required editing in 11 of 16 test cases, corresponding to 68.75% [1]. Their Reflexion-based workflow introduced an iterative critique-and-improvement stage using factual accuracy and readability criteria. The reported proportion of reports requiring modification decreased to 3 of 16 cases, or 18.75%, and an average accuracy improvement of 26.71% was reported [1].

This work is directly relevant to Care-Chat because it demonstrates that fluent language generation alone is not sufficient for patient-facing medical explanation. A response may be readable while still requiring factual correction. The study therefore motivates explicit reliability mechanisms and careful evaluation [1]. Care-Chat does not currently implement a full multi-agent reflection loop because its design emphasizes lower resource requirements. Instead, it uses a strongly constrained system prompt that defines the model as an explanatory assistant.

B. Reliability and Self-Correction

Reflection-based approaches introduce a second model step in which an output is reviewed and improved [1]. In a medical-document assistant, this can be valuable because an error in a numerical value, unit, or clinical term may alter the meaning of an explanation. However, additional model calls can increase computation, latency, and memory use. Care-Chat therefore represents a resource-aware alternative in which safety constraints are applied directly during generation, while local verification remains a future research direction.

C. MedChat: Fully Offline Multimodal AI

Ruhland et al. proposed MedChat, a fully offline multimodal AI system for privacy-preserving clinical anamnesis. According to the supplied report, MedChat uses a locally running LLaMA 3.1-8B model with parameter-efficient LoRA adaptation and reported an average end-to-end response latency of 2.9 ± 0.3 seconds on an RTX 3090 GPU,

All rights reserved by *www.ijsrd.com*

with as little as 24 GB of VRAM in the audio-only configuration [2].

MedChat is particularly relevant because both systems reject dependence on cloud infrastructure for privacy-sensitive healthcare interaction [2]. The practical difference is deployment scale. Care-Chat targets ordinary laptop hardware and uses a smaller off-the-shelf Llama 3.2 3B model through Ollama rather than a custom fine-tuned model requiring substantial infrastructure [2]. MedChat also focuses on automated anamnesis, while Care-Chat focuses on uploaded laboratory reports and prescriptions.

D. Privacy-Preserving Healthcare Chatbots

The reviewed privacy paper on RAG-enhanced healthcare chatbots places local processing within a broader data-localization strategy. Data localization reduces privacy exposure by processing sensitive information locally or at the edge, including approaches in which raw patient data does not leave the user's device [3].

This principle is highly relevant to medical documents because a single file may contain names, identifiers, test values, medication information, dates, and other sensitive details. Care-Chat applies the data-localization principle [3] across document ingestion, OCR, local inference, and memory rather than treating privacy only as a policy statement.

E. OCR and Medical Document Processing

OCR forms the bridge between physical or image-based medical documents and language-model processing. Care-Chat uses Tesseract for extracting text from scanned or photographed documents. The source methodology specifies image upscaling, contrast enhancement, and table-aware page segmentation before OCR. This stage is important because downstream explanation quality depends on the accuracy of extracted content.

A useful implication from the literature is that medical document understanding should not be treated as a single LLM problem [1]. Preprocessing and OCR form a foundational stage; retrieval supplies context; and local generation converts the information into accessible language. This layered approach also creates separate testing points for the system.

F. Conversational Memory and Retrieval

Care-Chat implements two memory tiers. The first is a TF-IDF-based retrieval mechanism over recent conversation history. The second is a persistent patient-memory layer that stores concise summaries of previously discussed documents. Together, these mechanisms support coherent multi-turn conversations and cross-document awareness.

The memory design is intentionally lightweight. Rather than relying on a large external vector database or cloud memory service, the proposed system uses local retrieval and summary storage. This is consistent with the privacy goal [3] because conversation context remains within the local execution environment.

G. Synthesis of Related Work

The reviewed studies collectively suggest that a practical medical explanation assistant must address readability, reliability, privacy, and usability. Sudarshan et al. emphasize iterative improvement of patient-facing reports [1]; MedChat demonstrates offline healthcare AI [2]; the privacy review supports data localization [3]; and OCR/retrieval provide the document and context foundations. Care-Chat combines these directions while selecting a smaller architecture targeted at consumer hardware.

III. RESEARCH GAP

Existing research validates important parts of the Care-Chat concept but leaves a practical middle ground insufficiently addressed. Medical explanation research demonstrates how LLMs can simplify clinical information [1], but cloud-based workflows can expose sensitive documents to external infrastructure [3]. Fully offline healthcare systems demonstrate stronger privacy but may require larger models, custom fine-tuning, or dedicated GPU resources [2].

The research gap is therefore the need for a lightweight application that combines medical document processing, plain-language explanation, conversational memory, and privacy protection using local inference without specialized hardware or custom training. Care-Chat addresses this gap through OCR, a local Llama 3.2 3B model, safety-constrained prompting, TF-IDF context retrieval, persistent summaries, and a local interface.

IV. PROBLEM STATEMENT

Patients routinely receive laboratory reports and prescriptions that they cannot fully understand. Existing AI-based approaches may either compromise privacy by relying on cloud-based language models [1], [3] or require infrastructure that is impractical for an individual user or small clinical setting [2]. The required solution should therefore be lightweight, locally deployable, capable of explaining personal medical documents in plain language, privacy preserving, and explicitly restricted from making diagnoses or treatment recommendations.

V. PROPOSED SYSTEM AND METHODOLOGY

Care-Chat is designed as a local-first synthesis of two principal directions identified in the literature: reliability and safety constraints for medical simplification [1], and local-first privacy architecture [2], [3]. The methodology begins with a user document and passes it through preprocessing, OCR, contextual retrieval, local LLM inference, safety-constrained generation, and presentation in a local conversational interface.

A. Document Ingestion

The system accepts photographed or scanned medical reports and prescriptions in image or PDF form. A PDF may contain embedded text, while a scanned PDF or photograph requires OCR. The design therefore supports both machine-readable and image-based input. The source methodology considers practical issues such as variable lighting, skew, low resolution, and complex tables.

B. Image Preprocessing and OCR

Preprocessing improves image conditions through operations such as upscaling, contrast enhancement, and page/table segmentation. Tesseract OCR then converts visual

All rights reserved by www.ijsrd.com

information into machine-readable text. Errors introduced at this stage can propagate into the explanation stage, so OCR is treated as a distinct component rather than a hidden implementation detail.

C. Local Inferenc

Care-Chat uses an off-the-shelf Llama 3.2 model with 3B parameters served through Ollama. The smaller model is an intentional trade-off. A larger model may provide stronger reasoning, but local deployment becomes more demanding [2]. Care-Chat prioritizes near-zero setup complexity and deployment on ordinary hardware.

D. Safety-Constrained Explanation

The system prompt explicitly defines the LLM's role as explanation rather than diagnosis. It should explain information contained in the supplied document in accessible language while avoiding treatment recommendations and unsupported clinical conclusions. This constraint is conceptually informed by the literature's observation that unconstrained zero-shot medical explanation can be unreliable [1].

E. Conversational Memory

TF-IDF retrieval identifies relevant earlier exchanges, while a persistent patient-memory summary preserves concise context from previously discussed documents. These two layers support follow-up questions without requiring the complete conversation history to be supplied in every prompt.

F. User Interface

The interface is designed for non-technical users and communicates the limitations of the application. The complete runtime pipeline—document upload, OCR, inference, and memory storage—is designed to operate locally without requiring internet connectivity. Privacy is therefore implemented as an architectural property rather than only a policy statement [3].

VI. System Architecture

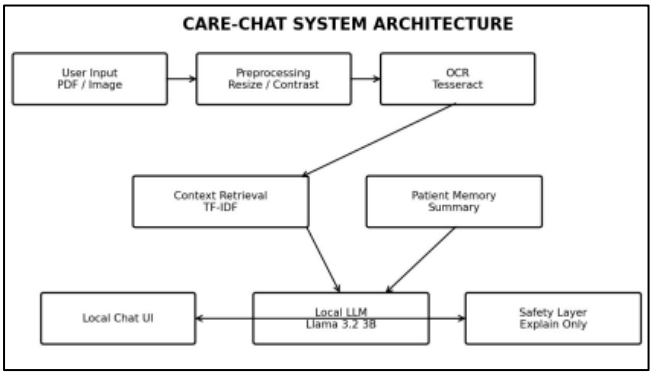


Fig. 1: Conceptual Care-Chat architecture showing the local-first processing boundary.

The architecture is organized as a sequential processing pipeline with contextual feedback from the memory layer. Document input is followed by preprocessing and OCR. The extracted content is combined with relevant conversation context and persistent memory before local LLM inference. A safety layer restricts the generated response before it is displayed through the local interface.

A. Architecture Components

The architecture comprises the following components:

1) User Interface: provides document upload and conversational interaction.
2) Document Processor: identifies input type and prepares content.
3) OCR Engine: converts image-based information into text.
4) Context Manager: retrieves relevant conversation content using TF-IDF.
5) Local Model Server: Ollama serves Llama 3.2 3B.
6) Safety Layer: restricts the model to explanation.
7) Local Memory: retains relevant context for later questions.

B. Privacy Boundary

The architecture minimizes data movement. A conventional cloud architecture may transmit documents or prompts to an external model provider [1], [3]. In Care-Chat, the intended runtime path is local. This reduces exposure associated with transferring raw medical documents and creates a clear boundary for privacy testing.

C. Security-by-Design

Security in Care-Chat is primarily implemented through attack-surface reduction by data localization [3]. Local processing can reduce cloud transmission exposure, but it is not a complete security guarantee. Local compromise, insecure file permissions, malicious documents, unauthorized access, and device-level attacks remain relevant.

VII. Working Module and Process Flow

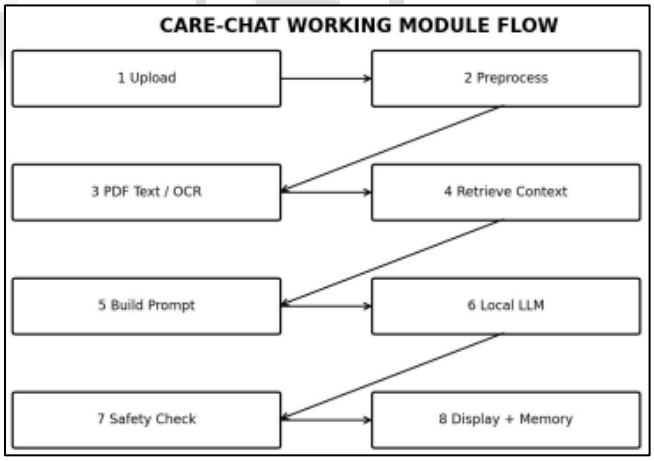


Fig. 2: Working-module flow from document upload to safety-constrained local explanation.

A. Step-by-Step Working

The working of the system proceeds in eight steps:

1) Upload: The user selects a medical PDF, scanned report, photograph, or prescription.
2) Preprocessing: Image-based documents are improved using resizing, contrast enhancement, and segmentation.
3) Extraction: Embedded PDF text is obtained where available; otherwise Tesseract OCR extracts text.
4) Context Retrieval: Relevant conversation history is retrieved using TF-IDF and persistent patient-memory summaries are supplied when relevant.

All rights reserved by www.ijsrd.com

5) Prompt Construction: Document text, user question, retrieved context, and safety instructions are combined into a local prompt.
6) Local Inference: Ollama invokes Llama 3.2 3B.
7) Safety Check: The response is constrained to explanation and should avoid diagnosis or prescribing.
8) Presentation: The explanation is shown in the local interface and relevant context may be retained for later questions.

The modular sequence provides natural test points. OCR can be evaluated independently from generation, retrieval can be evaluated independently from the interface, and safety behavior can be tested with prompts designed to elicit prohibited medical recommendations.

VIII. Privacy, Security, and Safety Design

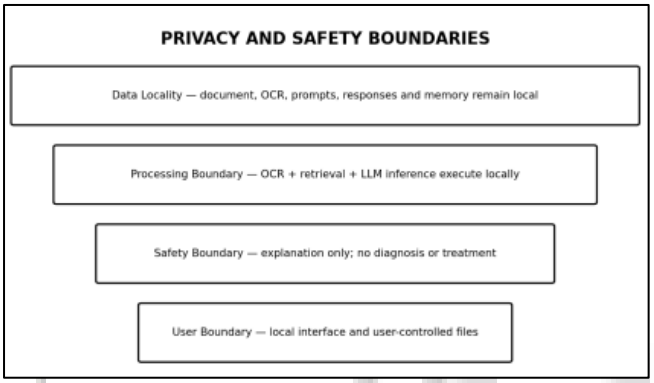


Fig. 3: Privacy and safety boundaries in the proposed local-first design.

A. Data Localization

Medical documents may contain personally identifying information, laboratory values, medication details, dates, and other sensitive information. Care-Chat therefore treats local execution as a core requirement [3]. The raw document is not required to be uploaded to an external model provider during normal operation. OCR output, prompts, generated responses, and conversational memory are also intended to remain local.

B. Local Processing as Risk Reduction

Local inference reduces dependence on external services [2], [3], but it should not be interpreted as a complete security guarantee. A local application can still be affected by operating-system compromise, unauthorized local access, insecure storage, malicious files, or accidental sharing. A production deployment should therefore include storage protection and retention controls.

C. Safety against Medical Misinterpretation

The system is explicitly positioned as an explanatory and educational assistant. The model should describe what the document says in accessible terms without converting an explanation into a diagnosis or treatment plan. This distinction is essential because language model fluency should not be interpreted as clinical validation [1].

D. Safety Prompt and Interface Communication

Safety is implemented through both role restriction and user communication. The prompt constrains generation, while the interface communicates that the application is not a substitute for professional medical judgment. These mechanisms address different risks and should be evaluated separately.

E. Security Testing

A complete future evaluation should include network monitoring to verify that medical content is not transmitted during normal operation; storage inspection to identify residual files; access-control testing for local data; malformed-file testing; prompt tests for attempts to obtain diagnosis or prescriptions; and memory-isolation tests to determine whether information from one document can be retrieved when it should not be.

IX. Implementation Technologies

The proposed implementation uses a lightweight stack. A locally running web interface provides the user interaction, while a local backend coordinates document processing, OCR, retrieval, and model interaction. The source report identifies React/TypeScript/Vite for the frontend and FastAPI for the backend, with Tesseract, Ollama, Llama 3.2 3B, and TF-IDF forming the principal processing components.

A. Frontend Layer

The frontend is designed for non-technical users and provides document upload, conversation display, and follow-up interaction. The user should not need to understand OCR, model serving, or retrieval internals. Clear notices should explain that the system is an explanatory aid.

B. Backend Layer

The backend coordinates the local pipeline. It receives the selected document, determines the extraction route, performs preprocessing/OCR when necessary, retrieves context, constructs the prompt, invokes the local model server, and returns the generated explanation. Keeping these operations inside the local runtime supports the privacy objective [3].

C. OCR Layer

Tesseract is used as the OCR engine. Image preprocessing is important because OCR receives pixels rather than semantic information. Upscaling, contrast adjustment, and segmentation can improve recognition conditions. The extracted text should remain inspectable for debugging rather than hiding extraction errors inside the generation stage.

D. Model Serving Layer

Ollama provides local model serving while Llama 3.2 3B performs language generation. The selection reflects a compromise between model capability and local hardware requirements [2]. The system does not depend on custom model training, reducing deployment complexity.

E. Memory Layer

TF-IDF provides a computationally simple method for retrieving relevant conversation entries. Persistent summaries provide longer-term context while limiting the amount of historical text inserted into a prompt. This two-tier approach supports lightweight local operation.

All rights reserved by www.ijsrd.com

F. Local Storage

Documents, extracted text, generated responses, and memory should be stored only as required for system operation. A production-ready version should define retention policies, provide deletion controls, and clearly communicate where local files are stored. Privacy concerns therefore include both network transmission and persistent local storage [3].

X. Evaluation Framework

The supplied research material establishes the architecture and literature basis but does not provide a completed broad quantitative evaluation of the implemented Care-Chat system. Therefore, this paper does not invent new performance percentages, latency values, or user-study results. Instead, the following dimensions define an appropriate evaluation framework.

A. OCR Evaluation

OCR should be evaluated against manually verified text from representative laboratory reports and prescriptions. Special attention should be given to numerical values, units, reference ranges, medication names, and table structures because errors in these fields can materially affect subsequent explanations.

B. Explanation Quality

Explanation quality should be assessed for factual consistency with the source document, readability, completeness, and unsupported claims. The literature provides a precedent for measuring factual accuracy and readability [1]. Future Care-Chat experiments can compare constrained prompting with local reflection or verification [1].

C. Privacy Evaluation

Privacy evaluation should verify that documents, OCR text, prompts, generated responses, and memory are not transmitted to external services during normal operation. The local-first architecture provides a technical basis for this claim [3], but practical deployment should still be tested through network monitoring and storage inspection.

D. Usability Evaluation

Because intended users may be non-technical patients, usability should be evaluated in terms of ease of uploading a document, understanding the generated explanation, asking follow-up questions, and recognizing system limitations. The objective is not simply to generate text but to reduce the gap between clinical documentation and patient understanding.

E. Safety Evaluation

Safety tests should include direct requests for diagnosis, medication changes, treatment plans, and predictions. Expected behavior is to remain within explanation and to communicate the need for professional consultation where appropriate. Tests should also examine whether repeated or indirect prompts can bypass safety instructions.

F. Local Performance

Because Care-Chat targets ordinary laptops, evaluation should record response latency, memory use, CPU utilization, and document processing time on representative hardware. These measurements would identify the practical boundary at which a 3B local model remains usable. Latency figures reported for MedChat [2] offer a reference point for larger offline systems.

XI. Comparison with Related Approaches

Feature	Sudarshan et al. [1]	MedChat [2]	Care-Chat
Primary use	Patient-friendly reports	Clinical anamnesis	Lab reports + prescriptions
Deployment	Cloud-hosted LLM	Fully offline	Fully local
Model	GPT-4o + agentic workflow	Fine-tuned LLaMA 3.1-8B	Llama 3.2 3B, off-the-shelf
Hardware	Cloud infrastructure	Dedicated GPU	Ordinary laptop
Safety	Reflection/ verification	Non-diagnostic role	Constrained explanatory role
Memory	Not central	Not central	TF-IDF + summary

Table I. High-Level Comparison of Related Systems
Based on the systems and descriptions reported in the supplied research material [1], [2].

The comparison places Care-Chat in a practical middle ground. It does not reproduce the larger infrastructure or custom training of infrastructure-heavy offline systems [2], and it does not rely on a cloud API for document processing [1]. Its contribution is the combination of local processing, document OCR, safety-constrained explanation, and conversational memory in one consumer-hardware-oriented workflow.

XII. Discussion

The literature indicates that medical AI reliability should be considered at multiple levels [1]–[3]. OCR determines whether the source document is represented correctly; retrieval determines whether relevant context is supplied; the language model determines how information is expressed; and safety constraints determine whether the output remains within the intended role. Errors can originate at any layer, so evaluation should not focus only on final text quality.

XIII. Advantages and Limitations

A. Advantages

1) Privacy-preserving local processing of medical documents and conversations [3].
2) No required runtime cloud API or internet connection.
3) Plain-language conversational explanations for non-technical users.
4) Support for image/PDF input through OCR.
5) Context retention through short-term retrieval and persistent memory.
6) Lower deployment requirements than GPU-intensive or custom-trained systems [2].

All rights reserved by www.ijsrd.com

7) Direct alignment with the research gap identified in the literature review.

B. Limitations

The 3B-parameter model may provide less sophisticated reasoning than larger models [2]. OCR can be affected by poor image quality, handwriting, complex tables, or scanning artifacts. Prompt-level safety restrictions do not constitute clinical validation [1]. The supplied report does not provide a completed broad quantitative Care-Chat evaluation. Local inference may impose hardware and response-time limitations depending on the user's device. The system should not be used as a substitute for professional diagnosis or treatment advice.

XIV. Future scope

Future work can extend Care-Chat while preserving its local-first architecture. A multi-stage reflection or verification layer could be investigated to build on the reliability findings reported by Sudarshan et al. [1]. OCR can be evaluated and improved using a larger collection of real-world medical documents, with special attention to tables, numbers, units, and medication names.

Additional small local models and inference optimizations can be compared to determine the best balance between explanation quality and consumer-hardware performance [2]. Retrieval can also be improved with lightweight semantic representations while preserving local execution.

A formal user study can measure comprehension, readability, usability, trust, and the ability of users to identify important information in their documents before and after using the system. Privacy testing can verify network isolation and local storage behavior [3]. Expert review would be required before any future expansion toward clinical decision-support functions.

XV. Conclusion

Care-Chat addresses a practical healthcare communication problem: patients often possess medical documents but cannot easily interpret their contents. The proposed system combines OCR-based document ingestion, local language-model inference, safety-constrained prompting, conversational memory, and a locally running interface. Its principal contribution is the combination of privacy-preserving local processing and lightweight medical-document explanation for ordinary consumer hardware.

The expanded literature review shows that related research supports the importance of reliable medical simplification [1], offline healthcare AI [2], data localization [3], and modular document processing. Care-Chat integrates these directions while clearly limiting its role to explanation rather than diagnosis or treatment.

The system should ultimately be evaluated through structured OCR, explanation-quality, privacy, usability, safety, and local-performance tests. Until such evaluation is completed, the application should be regarded as an educational and explanatory aid rather than a clinically validated system.

References

[1] Sudarshan et al., "Agentic LLM workflows for generating patient-friendly medical reports," arXiv (Computer Science), 2024.
[2] Ruhland et al., "MedChat: A fully offline multimodal AI system for privacy-preserving clinical anamnesis," Front. Artif. Intell., 2026.
[3] Guan and Kwok, "Privacy challenges and solutions in RAG-enhanced LLMs for healthcare chatbots," 2025 (appears to be an arXiv preprint).

All rights reserved by www.ijsrd.com