iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI:

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 X Oct 2026- Available at www.ijraset.com

NexResearch AI: An Intelligent Research Paper Analysis and Generation System

Mrs. E. Goma[1], Bhavana G.[2], Abinaya R.[3], Aishwarya R.[4]
[1]*Assistant Professor, Dept. of CSE, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India*
[2, 3, 4]*Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India*

Abstract: ***Research work increasingly requires researchers and students to examine lengthy papers, understand technical concepts, compare existing approaches, identify research opportunities, and prepare new academic content. Performing these activities manually can be time-consuming and may make it difficult to maintain a connected view of the research process. This paper presents NexResearch AI, a web-based research assistance platform developed using the MERN stack and generative AI. The proposed system organizes research support into four modules: Understand, Discover, Generate, and Review. Understand processes an uploaded PDF and supports structured summaries, section-level explanations, concepts, formulas, and question answering. Discover helps users examine limitations, possible enhancements, research gaps, related work, and future directions. Generate assists with research ideas, problem statements, methodologies, proposals, and draft paper sections. Review supports citation-related checks, similarity-oriented review, and assessment of research quality. The application uses a React-based interface, Node.js and Express.js backend services, MongoDB for paper-related data, PDF text extraction, and server-side integration with the generative AI API. A paper-specific workflow is intended to keep analysis associated with the selected paper and to encourage evidence-grounded responses. This paper describes the system design, processing workflow, implementation approach, expected benefits, and current limitations. Quantitative accuracy and performance claims are not made because a controlled evaluation dataset and measured results have not yet been established. The platform is intended to assist, not replace, researcher judgment, source verification, and academic integrity practices.***

Keywords: ***Artificial Intelligence, Research Paper Analysis, Generative AI, MERN Stack, Research Gap Identification, Academic Writing Assistance, Natural Language Processing.***

I. INTRODUCTION

Scientific and technical research depends on the ability to understand prior work, recognize unresolved problems, select suitable methods, and communicate findings clearly. A research paper may contain specialized terminology, mathematical expressions, tables, figures, and multiple sections that require careful reading. Students and early-stage researchers may need to repeat these tasks across several papers before they can formulate a focused research question.

Recent advances in large language models (LLMs) have created opportunities for natural-language interaction with technical documents and for assisting with summarization and drafting. However, generated responses can contain unsupported statements, incorrect references, or details that do not appear in the source material. Research-support systems therefore need to make the user's source document central to the workflow and preserve the distinction between extracted evidence and AI-generated suggestions [1], [2]. NexResearch AI is proposed as an integrated research decision-support platform that connects paper understanding with research opportunity discovery, content generation, and review. Instead of treating these tasks as unrelated utilities, the platform organizes them into a single workflow: upload a paper or enter a topic, understand the available material, discover possible directions, generate a draft, and review the output.

The main contributions of this work are: (1) a four-module architecture for a connected research-assistance workflow; (2) integration of PDF text extraction, paper-specific analysis, and generative AI-powered natural-language support; (3) a structured approach for exploring limitations, research gaps, and future directions; and (4) a review stage that reminds users to verify citations, similarity, and research quality before submission.

II. RELATED WORK

Automatic scientific-document summarization has been studied as a way to reduce the effort required to understand long research articles. Majadly and Last investigated extracting peer-review-related aspects, including motivation, contribution, claims, and supporting evidence, and using these aspects to improve scientific-article summaries [3]. Their work supports the importance of structuring summaries around meaningful research components rather than producing only a general abstract.

©IJRASET: All Rights are Reserved

Lewis et al. introduced retrieval-augmented generation (RAG), which combines a language model with retrieved information to improve responses to knowledge-intensive tasks [1]. The approach highlights the value of linking generated output to external evidence. NexResearch AI follows the broader principle of associating analysis with the selected paper; the current implementation should not be described as a complete RAG system unless a dedicated retrieval and grounding pipeline is implemented and evaluated. Han, Susnjak, and Mathrani reviewed the use of RAG for automating systematic literature reviews and discussed its potential for extraction, summarization, and trend identification, alongside risks such as hallucinations [2]. Their discussion reinforces the need for transparent workflows and human verification. Research on AI-enhanced literature review tools has also identified usability, integration, and standardized evaluation as important directions for future systems [4].

Existing tools and research demonstrate progress in individual activities such as summarization, literature retrieval, and writing assistance. NexResearch AI focuses on connecting several research activities within one web application. Its intended distinction is workflow integration across the Understand, Discover, Generate, and Review modules, rather than a claim that any one AI task is entirely new.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Researchers often use separate tools to read papers, extract important points, explore research gaps, prepare drafts, and check citations. Moving between these tools can fragment the workflow, require repeated manual copying, and make it harder to track which source paper informed a particular analysis. AI-generated content introduces additional concerns when claims or references are not checked against reliable sources. The problem addressed by NexResearch AI is the lack of a single, structured workflow that supports these activities while keeping uploaded-paper context associated with analysis.

B. Objectives

- To accept research papers in PDF format and extract text for subsequent analysis.
- To provide structured summaries, section explanations, concept and formula explanations, and paper-related question answering.
- To help users explore limitations, possible improvements, research gaps, related work, and future research directions.
- To assist in preparing research ideas, problem statements, methodologies, proposals, and draft paper content.
- To support review activities and encourage verification of references, similarity reports, and research quality.
- To provide a maintainable web application using React, Node.js, Express.js, MongoDB, and a server-side generative AI API integration.

IV. PROPOSED SYSTEM

NexResearch AI is a browser-based application that connects a user interface, backend API, document-processing utilities, database, and generative-AI service. The React interface provides access to the research modules. The Node.js and Express.js backend receives requests, processes application logic, manages paper-specific operations, and communicates with MongoDB and the generative AI API. PDF text extraction makes the document content available to the analysis workflow. MongoDB stores paper metadata and associated application data. The design separates the source document from the AI-generated output. A paper identifier is used to associate analysis requests with the selected paper, reducing the risk of accidentally using context from a different upload. The application should return source-grounded answers only when the required source content is available; otherwise, the interface should communicate that the answer could not be established from the uploaded material. AI-generated interpretations and suggestions still require user verification. The four modules are summarized in Table I.

TABLE I. SUMMARY OF SYSTEM MODULES

Module	Main functions
Understand	Structured summary, section-wise explanation, objectives, methodology, concepts, formulas, paper-specific question answering
Discover	Limitations, possible enhancements, research-gap candidates, related work, future directions
Generate	Research ideas, problem statements, objectives, methodology outlines, proposals, draft paper sections
Review	Citation-related checks, similarity-oriented review, research-quality feedback

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

A. *Understand*

The Understand module processes an uploaded paper and presents its main information in an accessible form. Functions include a paper summary, section-wise explanation, research objectives, methodology, concepts, formulas, document structure, and question answering. The aim is to help users comprehend the paper before attempting to identify new research directions.

B. *Discover*

The Discover module helps users inspect limitations and future-scope statements, consider possible improvements, and formulate potential research gaps or follow-up questions. Similar-paper and trend exploration may use external scholarly metadata services where configured. Any proposed gap is a candidate for further investigation, not proof that no prior work exists; users must confirm it through a systematic search of the literature.

C. *Generate*

The Generate module assists users in turning a topic or selected research opportunity into structured content, including research ideas, problem statements, objectives, methodology outlines, proposals, draft paper sections, and presentation material where implemented. Generated text is a starting point and must be checked for originality, factual support, correct citations, and compliance with the target journal's requirements.

D. *Review*

The Review module is intended to help assess generated or uploaded content through citation-related checks, similarity-oriented review, and research-quality feedback. These functions should be interpreted according to their actual implementation. An AI-based similarity estimate is not equivalent to a validated plagiarism-detection service, and citation formatting does not guarantee that a reference is real or that it supports the associated claim.

V. SYSTEM ARCHITECTURE AND METHODOLOGY

The system follows a layered client-server architecture, shown in Fig. 1. At the client layer, React provides the user interface and module navigation. The application server, built with Node.js and Express.js, exposes API endpoints for paper uploads and analysis or generation requests. PDF-processing utilities extract text from uploaded documents. MongoDB stores metadata and paper-related records. The generative AI API is called from the server so that the API key is not exposed in browser code.

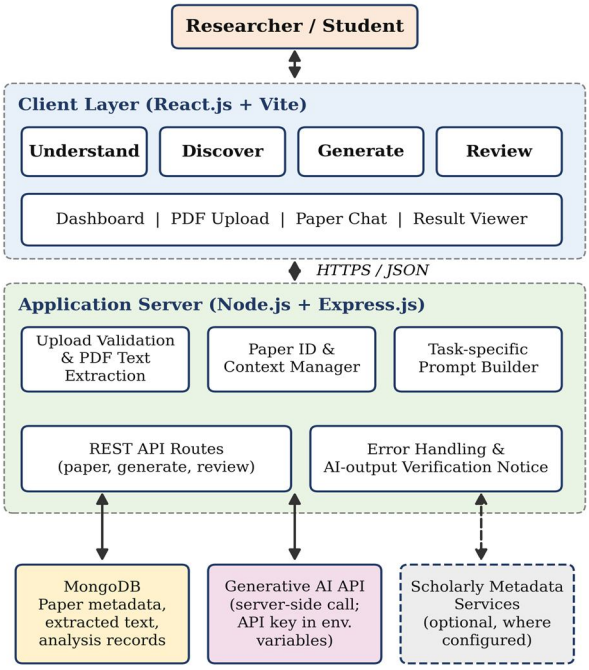


Fig. 1. Architecture of the NexResearch AI system.

The workflow begins when the user uploads a PDF or supplies a research topic. For a PDF, the backend validates the upload, extracts readable text, associates the result with a unique paper identifier, and stores relevant metadata and extracted content. When the user selects a module, the backend retrieves the appropriate paper context and constructs a task-specific prompt. The generative AI response is returned to the interface and presented as a summary, explanation, discovery suggestion, generated draft, or review output. The user then checks the result and may revise the request or move to another module.

A. *Processing Algorithm*

1) Start the application and initialize the frontend, backend services, database connection, PDF utilities, and AI service configuration.
2) Accept a PDF upload or a research-topic input from the user.
3) If a PDF is supplied, validate the file and extract its readable text; report an error if extraction fails or no usable text is found.
4) Create or retrieve a unique paper record and associate the extracted content and metadata with its paper identifier.
5) Receive the user's selected task: Understand, Discover, Generate, or Review.
6) Retrieve the selected paper's content and prepare a prompt appropriate to the requested task.
7) Send the prompt to the generative AI API through the backend and handle API errors or empty responses.
8) Display the generated result with a clear indication that AI output requires verification; preserve association with the selected paper.
9) Allow the user to review, refine, or continue to another module.
10) End the request and retain application data according to the configured storage and privacy practices.

B. *Evidence-Grounded Analysis*

To reduce cross-paper context errors, analysis requests should include the selected paper identifier and the relevant extracted text or sections. Prompts should explicitly instruct the model not to invent missing author details, results, formulas, or citations. Where feasible, answers should include the paper title and a page or section reference. If page-level mapping is unavailable in the extracted text, the system should not fabricate page numbers. These controls improve traceability but do not eliminate model errors; source checking remains necessary.

C. *Data and API Handling*

The browser communicates with the backend through application API routes. The backend handles file validation, text extraction, database operations, task selection, prompt construction, and calls to generative AI. API credentials should be held in server-side environment variables and must not be committed to source control or sent to the browser. Upload limits, request validation, error handling, and suitable access controls should be configured for deployment. Stored papers may contain unpublished or sensitive research, so data retention and access policies should be clearly communicated to users.

VI. IMPLEMENTATION

The system is implemented as a MERN-stack web application. React.js [8] and Vite are used for the frontend development workflow, while Node.js [7] and Express.js provide backend services. MongoDB [6] is used for paper metadata and associated analysis records. PDF text extraction provides the input text for downstream tasks, and the generative AI API [5] supports natural-language analysis and content generation. Keeping AI calls on the server allows the application to protect API credentials and centralize request handling. The interface groups the core research tasks into four modules so that users can move from comprehension to discovery, generation, and review. The upload and analysis workflow is designed to associate a selected paper with its own analysis context. The exact set of available outputs depends on which routes, UI pages, and integrations are enabled in the deployed version. External paper search, automated similarity scoring, and citation validation should only be claimed as fully operational when the relevant integration has been implemented and tested.

VII. RESULTS AND DISCUSSION

The present work describes the system design and implementation approach for an integrated research-assistance application. A qualitative functional walkthrough can examine whether a user can upload a paper, obtain extracted text, request structured explanations, explore possible research directions, generate draft content, and access review-oriented functions. These checks establish workflow coverage, but they do not by themselves establish the factual accuracy of AI responses or the effectiveness of research-gap detection.

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

The intended benefit is reduced switching between separate tools and a more organized transition from reading a paper to developing a research idea. The modular design also makes it possible to extend individual functions without changing the overall workflow. However, output quality depends on PDF extraction quality, the completeness of the source text, prompt design, model behavior, and the user's ability to verify generated statements. Similarity and citation-related functions require particular care because a generated score or formatted reference may be incomplete or incorrect.

No numerical performance results are reported in this paper because no controlled benchmark, independent expert evaluation, or measured accuracy and latency figures were available for this draft. Before submission of a full evaluation, the system should be tested using a documented set of papers and tasks, as outlined in Section IX.

VIII. ADVANTAGES AND LIMITATIONS

A. *Advantages*

1) Provides a connected workflow for understanding papers, discovering possible research directions, generating content, and reviewing output.
2) Uses a familiar web-based interface and a MERN-stack architecture.
3) Supports natural-language interaction with paper content through a generative-AI service.
4) Associates analysis with a selected paper record to help reduce context mixing between uploads.
5) Can be extended with scholarly metadata retrieval, stronger source-grounding, and structured evaluation.

B. *Limitations*

1) PDFs with scanned pages, complex layouts, multi-column text, or mathematical notation may not extract cleanly without OCR or specialized parsing.
2) Generative AI may produce inaccurate explanations, unsupported claims, or fabricated references.
3) A proposed research gap cannot be considered novel until it is checked against a sufficiently broad and current body of literature.
4) Similarity-oriented output is not a substitute for a validated plagiarism checker or institutional academic-integrity process.
5) Performance, accuracy, usability, and scalability remain to be quantified through systematic testing.
6) The availability and quality of external paper search depend on the services and APIs actually configured in the deployed application.

IX. EVALUATION PLAN AND REPRODUCIBILITY

A reproducible evaluation is required before making quantitative claims about the system. The evaluation should use a documented collection of research papers covering different subject areas, PDF layouts, and document lengths. The dataset should record each paper's source, publication year, file type, and any restrictions on reuse. Where possible, papers should be openly licensed or used with permission. The test procedure should evaluate each module separately and then assess the end-to-end workflow. For PDF processing, record whether text extraction succeeds and whether key sections are retained. For Understand, compare generated summaries and answers against source passages reviewed by a human evaluator. For Discover, ask domain reviewers to judge whether proposed opportunities are relevant, traceable to evidence, and appropriately labelled as hypotheses. For Generate, assess factual support, structure, and reference validity. For Review, test citation formatting and any similarity-related feature against a clearly identified baseline. Table II lists the proposed measures.

TABLE II. PROPOSED EVALUATION CHECKLIST

Component	Measure	Evidence to record
PDF extraction	Extraction success rate	Files tested, files with usable text, extraction errors
Understand	Source-supported answer rate	Human-reviewed answers and supporting source passages
Discover	Relevance and traceability	Expert ratings and evidence linked to each proposed gap
Generate	Factual and citation validity	Claim checks, reference checks, reviewer rubric
Review	Agreement with baseline	Baseline tool, test cases, false positives and false negatives
Usability	User-rated usefulness	Participant count, questionnaire, summary statistics

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

Each measure should include its denominator and scoring rules. At least two evaluators should independently score a sample of outputs where feasible, and disagreements should be resolved using a documented procedure. Results should report the number of papers and tasks tested, the model identifier and access date, prompt versions, relevant configuration settings, and known exclusions.

X. RESPONSIBLE USE AND DATA PRIVACY

The platform should be presented as a research-assistance tool rather than an autonomous authority on scientific validity. Users remain responsible for checking factual claims, validating citations, obtaining permission to use source material, and complying with their institution's policies on AI-assisted writing. Generated content should not be submitted as verified research without appropriate review and attribution.

Uploaded manuscripts may contain unpublished findings or confidential information. A production deployment should document what data are stored, how long they are retained, who can access them, and whether content is sent to an external AI provider. API credentials must remain on the server. Access controls, upload validation, error handling, secure configuration, and a deletion process should be considered before public deployment. These are design recommendations; their implementation status should be confirmed against the running application.

XI. CONCLUSION AND FUTURE SCOPE

This paper presented NexResearch AI, a web-based research assistance platform designed to connect research-paper understanding, opportunity discovery, content generation, and review in one workflow. Built around React.js, Node.js, Express.js, MongoDB, PDF text extraction, and server-side generative AI API integration, the system organizes its functionality into four modules: Understand, Discover, Generate, and Review. The approach aims to make research tasks more accessible and structured while keeping human verification central to academic work.

Future work includes improving extraction of tables, figures, equations, and scanned PDFs; implementing stronger retrieval and source-citation mechanisms; integrating reliable scholarly metadata services; validating references against authoritative records; and evaluating similarity-related functions against appropriate baselines. A formal evaluation should measure factual consistency, extraction quality, response time, citation validity, and usability using a documented dataset and expert review. Further development should also address security, privacy, reproducibility, and responsible use of AI in academic writing.

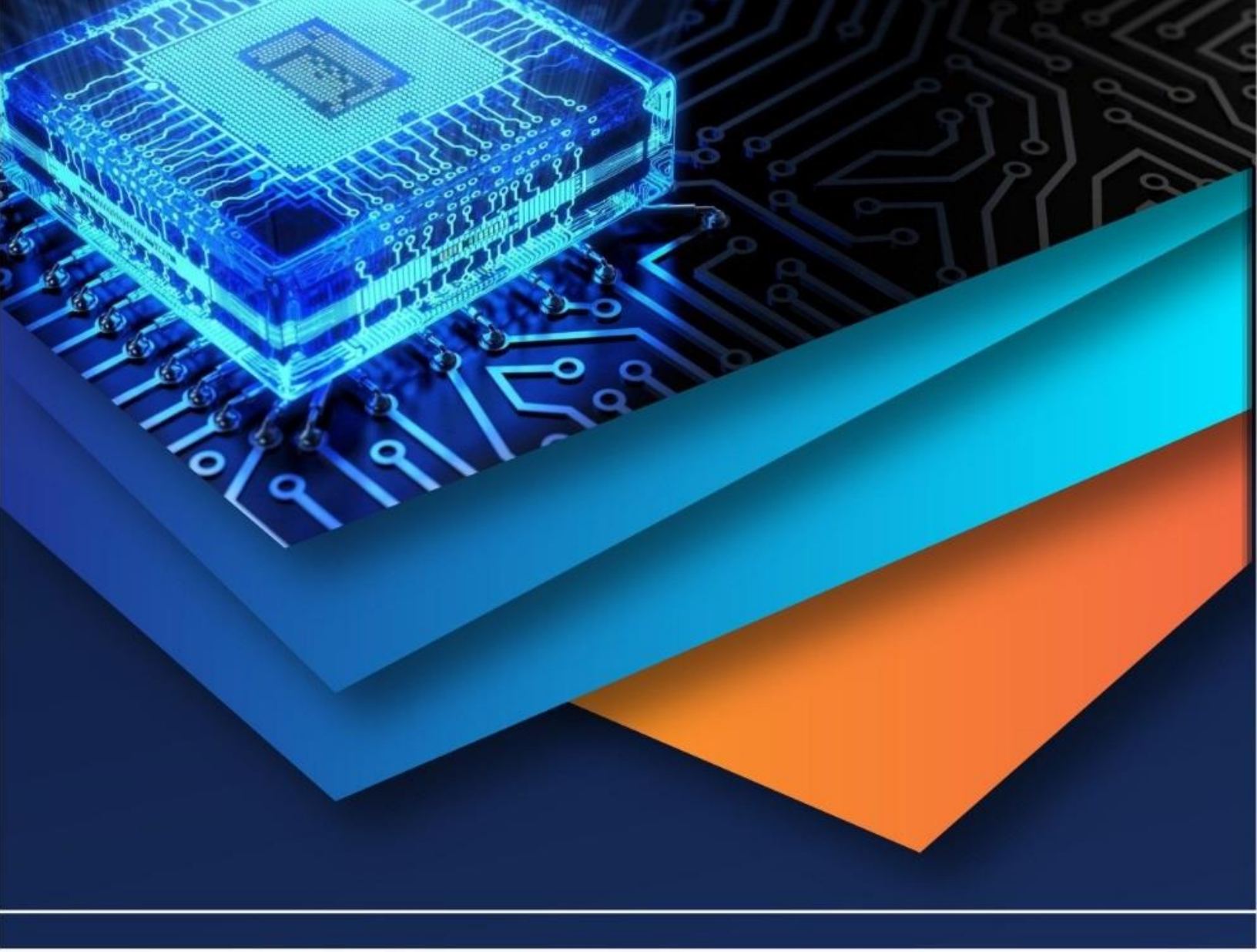
XII. ACKNOWLEDGMENT

The authors thank the project guide, Mrs. E. Goma, and the Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, for their guidance and academic support.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.
[2] B. Han, T. Susnjak, and A. Mathrani, "Automating Systematic Literature Reviews with Retrieval-Augmented Generation: A Comprehensive Overview," Applied Sciences, vol. 14, no. 19, Art. no. 9103, 2024, doi: 10.3390/app14199103.
[3] M. Majadly and M. Last, "Leveraging peer-review aspects for extractive and abstractive summarization of scientific articles," International Journal of Data Science and Analytics, vol. 19, pp. 537–550, 2025, doi: 10.1007/s41060-024-00665-z.
[4] F. Bolaños, A. Salatino, F. Osborne, and E. Motta, "Artificial intelligence for literature reviews: opportunities and challenges," Artificial Intelligence Review, vol. 57, Art. no. 259, 2024, doi: 10.1007/s10462-024-10902-3.
[5] Google, "Gemini API Documentation," Google AI for Developers. [Online]. Available: https://ai.google.dev/gemini-api/docs. Accessed: Oct. 29, 2026.
[6] MongoDB, "MongoDB Manual." [Online]. Available: https://www.mongodb.com/docs/. Accessed: Oct. 29, 2026.
[7] OpenJS Foundation, "Node.js Documentation." [Online]. Available: https://nodejs.org/docs/. Accessed: Oct. 29, 2026.
[8] Meta Platforms, Inc., "React Documentation." [Online]. Available: https://react.dev/. Accessed: Oct. 29, 2026.

©IJRASET: All Rights are Reserved



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)