RAPTOR AI: AI-Powered Intelligent Research Proposal Evaluation & Decision Support Platform

Kailash R[1], Deepan X[1], Sangeetha K[2], Gugan KM[1], Dev Nishanth K[1], Kishor D[1]
[1]Department of Computer Science and Engineering, SNS College of Engineering, Coimbatore, India
[2]Assistant Professor, CSE Department, SNS College of Engineering, Coimbatore, India

---

Abstract - Evaluating research proposals involves submissions, reviewer assignments, scoring sheets, overlap checks, and final decisions that are often handled through separate documents and message threads. This makes the process slow, difficult to audit, and sensitive to inconsistent reviewer judgement. RAPTOR AI is an AI-powered research proposal evaluation and decision support platform designed to organize these activities within a single web application. The proposed system combines proposal submission and document management, text extraction, semantic similarity analysis using sentence embeddings and a vector index, rubric-based reviewer evaluation, role-based access for administrators, researchers, and reviewers, and a decision dashboard with consolidated reports. This paper describes the system scope, workflow, architecture, prototype screens, and responsible-use considerations. The platform is intended to assist human reviewers, not to replace them: AI-generated outputs are advisory and final decisions remain with authorized people. The accuracy of similarity detection, scoring support, security-test results, system performance, and end-to-end behaviour require further validation. Accordingly, the reported outcomes should be considered design- and prototype-level observations rather than claims of measured production performance. A validation plan is outlined for assessing accuracy, reliability, security, usability, and decision-support quality before deployment.

Key Words: AI-powered proposal evaluation, decision support, semantic similarity, research proposal review, role-based access, reviewer dashboard

I. INTRODUCTION

Universities, funding agencies, and research institutions receive a steady flow of research proposals that must be screened, compared, reviewed, and decided upon. When proposal documents, reviewer comments, score sheets, and decision records are kept in separate files, e-mail threads, or spreadsheets, it becomes difficult to track the status of a proposal, check whether similar work has already been submitted, apply the same criteria to every submission, and explain why a decision was made. As a result, administrators may spend more time coordinating records than supporting informed decisions.

RAPTOR AI is proposed as a centralized research proposal evaluation platform that connects proposal submissions, uploaded documents, similarity analysis, reviewer evaluations, decision records, and reports. Its purpose is not to make funding or approval decisions without human involvement; rather, it is to organize proposal information and provide assistance through data analysis. Similarity findings and AI-generated insights are intended to help reviewers focus their attention. Administrators and reviewers retain responsibility for the final decision.

The workflow contains four major stages: proposal submission and ingestion, preprocessing and embedding, AI-assisted analysis, and review with decision support. The system accepts a proposal document, extracts and cleans its text, converts it into semantic embeddings, compares it with stored proposals, and presents the results alongside a structured evaluation form. These outputs represent the project's proposed functionality.

This paper makes three design-level contributions. First, it organizes the proposal lifecycle into a single platform with clearly separated roles. Second, it describes how document ingestion, semantic similarity analysis, rubric-based review, and decision reporting can be integrated into a modular architecture. Third, it identifies the implementation and evaluation requirements needed to avoid unsupported claims about AI accuracy, security, or production readiness. The work is presented as a system-design case study supported by project documentation and illustrative interface wireframes. Because reproducible runtime measurements were not supplied, the paper distinguishes specified features and prototype outputs from verified operational results.

II. LITERATURE REVIEW

Research on scientific peer review describes it as the main mechanism for judging the quality of proposals and manuscripts, while also documenting its limits, including subjectivity, variable reviewer standards, and the effort required to coordinate reviewers [1]. Empirical work on grant applications has reported low agreement among reviewers who evaluate the same submissions [2]. These findings suggest that structured criteria, transparent records, and consistent workflows can support review, but they do not remove the need for expert judgement.

Advances in natural language processing make it possible to compare documents by meaning rather than by keywords alone. Pre-trained transformer language models learn contextual representations of text [3], and Sentence-BERT adapts such models to produce

fixed-size sentence embeddings that can be compared efficiently using cosine similarity [4]. For large collections, vector-search libraries such as FAISS support fast nearest-neighbour retrieval over dense embeddings [5]. Together, these techniques are suitable for flagging proposals that overlap in topic or wording, provided that the output is checked by a person.

Studies of bias and fairness in machine learning show that automated systems can reproduce or amplify problems in their data and design [6]. This is especially relevant when an analytical result may influence whether a proposal is supported. A decision support platform should therefore explain its outputs, preserve the underlying records, allow correction of errors, and be evaluated across different proposal types and disciplines rather than summarized by one overall accuracy score.

Security and governance frameworks give additional guidance. The NIST AI Risk Management Framework emphasizes managing AI-related risks across design, development, deployment, and use [7]. The OWASP Application Security Verification Standard provides a reference for examining application security controls [8]. For token-based authentication, JSON Web Tokens are defined in RFC 7519 [9]. For RAPTOR AI, these perspectives support a design that separates AI assistance from authoritative business rules, protects proposal documents, records important state changes, and requires testing before operational claims are made.

The design gap addressed here is integration. Proposal submission, document storage, similarity checking, reviewer scoring, and reporting can exist as separate tools. RAPTOR AI proposes to connect these functions through a shared proposal record and a structured evaluation workflow. Its intended contribution is improved organization and traceability, not a claim that the proposed system or prototype outperforms existing proposal management or plagiarism-checking tools. Such a comparative claim would require a controlled evaluation with equivalent proposals, reviewers, and performance criteria.

III. METHODOLOGY

A. Study Scope and Design Approach

The work is organized as a design-oriented case study of a research proposal evaluation system. The source materials used for this manuscript describe the problem context, functional workflow, architecture, interface wireframes, limitations, and future validation needs. The analysis maps those artifacts into a common system model: users and modules are defined first, the evaluation workflow is then specified, and the architecture identifies where application rules, AI-assisted analysis services, persistent records, and file storage belong. This is not a controlled experiment. No new proposal dataset, reviewer study, or measured performance benchmark is claimed.

The design objective is to give administrators, researchers, and reviewers one organized view of proposal activity while keeping decisions under human control. The system defines three roles. Administrators manage users, assign reviewers, and record decisions. Researchers submit proposals and track their status. Reviewers evaluate assigned proposals against a rubric. Access should be controlled according to role and permission: a user should be able to access authorized information but not other users' private records, and recommendations should be based on verified records rather than manually changed dashboard values.

B. Functional Model and Proposal Data

The functional model contains several connected modules. User and role management stores accounts and permissions and issues authenticated sessions. Proposal management stores the title, abstract, discipline, and status, and links each proposal to its researcher. Document ingestion validates a supported file, stores it in a protected location, and invokes text extraction. Extracted text is presented for review because document structure, scanned pages, tables, and incomplete information can produce errors. The original document should remain available to authorized reviewers so that extracted values can be checked against the source.

Similarity analysis relates a new proposal to stored proposals and returns the closest matches with scores, not only an unexplained flag. Review management records rubric criteria, scores, and comments for each assigned reviewer. Decision management records the outcome, its justification, and the person responsible. Reporting consolidates proposal status, review results, and decision records, while an audit log records important changes. The platform is planned to be built incrementally across seven development phases, each ending with a demonstrable increment.

C. Workflow and Evaluation Stages

The workflow begins when a researcher registers and submits a proposal with its supporting document. The analysis supports the initial understanding of a proposal but does not itself make decisions. Users manage proposals through the following stages in the specified order:

- **Stage 1 - Proposal Submission and Ingestion:** The researcher enters proposal details and uploads a document. The system validates file type and size, stores the file, and extracts its text for review.
- **Stage 2 - Preprocessing and Embedding:** The system cleans and segments the text and generates sentence embeddings for each segment using a sentence-transformer model.
- **Stage 3 - AI-Assisted Analysis:** The platform searches a vector index of stored proposals and reports the nearest matches and their similarity scores. Generated insights are supporting information and should be reviewed before any decision is made.
- **Stage 4 - Review and Decision Support:** Assigned reviewers complete the rubric, and administrators view the dashboard, similarity reports, and review summaries before recording a decision. People make the final decision.

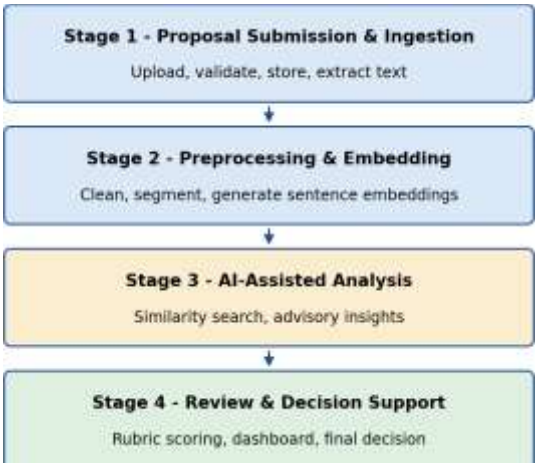


Fig. 1. Four-stage proposal evaluation workflow

The evaluation process is based on system rules, not just dashboard labels. The backend should calculate or verify results using stored records, apply the configured rubric consistently, and store the resulting status. Incomplete records, invalid uploads, or unavailable analysis services should not silently generate outputs. The application should use explicit states for submitted, processing, analysis completed, review in progress, and decision recorded, with retry and recovery procedures defined according to the system requirements.

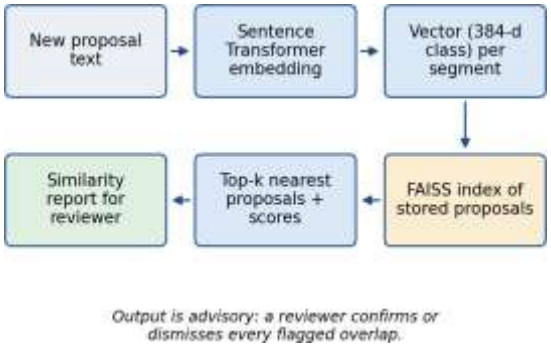


Fig. 2. Semantic similarity analysis model

D. System Architecture and Data Flow

The proposed architecture separates the presentation layer, application workflow, AI-assisted processing, persistent data, and file storage. The presentation layer is a React and Vite web application that provides role-specific interfaces. The application layer, implemented with FastAPI, validates requests, authenticates users with JSON Web Tokens, checks role permissions, manages proposal and review records, applies rubric rules, and generates reports. These responsibilities should remain authoritative on the server so that users cannot modify results simply by changing client-side values.

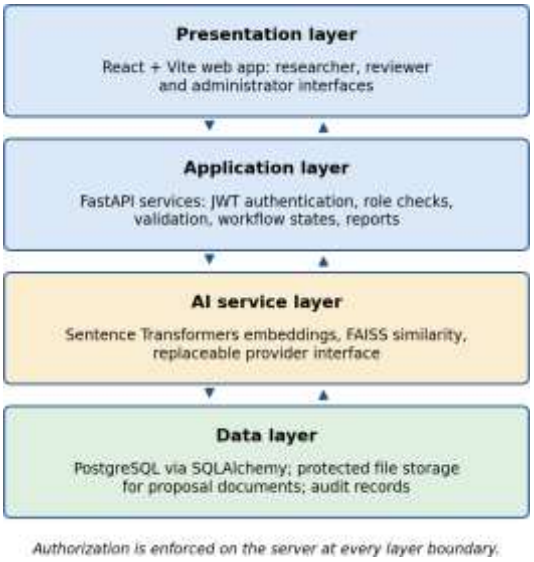


Fig. 3. Proposed system architecture

The AI service layer supports text embedding and similarity search. A replaceable provider interface can isolate the analysis workflow from a specific embedding model or index. Before being stored or displayed, AI-generated output should be checked for the expected format. Where practical, the system should retain enough context, such as the model identifier, the compared proposals, and the score, to help an authorized user understand the result. A prototype response or sample value must not be represented as a live AI-generated result.

The data layer uses PostgreSQL accessed through SQLAlchemy and holds structured records such as users, proposals, documents, embeddings metadata, reviews, rubric scores, decisions, and audit events. Uploaded files should be stored in protected storage with metadata linked to database records, and the server should check authorization before returning any document or report. Long-running tasks such as embedding generation and report creation may use background jobs with progress reporting, bounded retries, and clear failure states.

E. Security, Privacy, and Responsible AI

Proposal records contain unpublished research ideas, personal details, reviewer identities, scores, and decisions. Access should be restricted by role, connections should use secure transport, and stored files should not be publicly accessible by default. Passwords should be stored only as salted hashes, and tokens should have limited lifetimes. Input validation is needed for forms, document uploads, and API calls. Audit events should record important changes with timestamps and user information. Retention and deletion rules should be defined for documents, reviews, and reports. These requirements should be validated against the selected application-security controls [8], [9].

Users should receive clear information about what data is collected, why it is used, how long it is retained, and who can access it. Reviewer identities and comments need particular care so that they are visible only to

authorized roles. AI-generated similarity findings should be advisory, based on available evidence, and correctable by reviewers. A high similarity score does not by itself prove duplication or misconduct, and automated outputs should not be treated as final outcomes. The platform should be evaluated for errors and unreliable results, and its AI-related risks should be managed throughout the system lifecycle rather than only after deployment [6], [7].

IV. RESULTS

A. Prototype and Design Outcomes

The available project artifacts define the main workflow across proposal evaluation functions. The user-facing design includes a decision dashboard, proposal submission and tracking, reviewer evaluation, similarity reports, user and role management, and reports. User activities are represented as a sequence of proposal submission, embedding and similarity analysis, rubric-based review, and decision recording. The wireframes communicate the intended module boundaries and the information users are expected to review; they are illustrative interface designs rather than evidence that all backend integrations are operational.

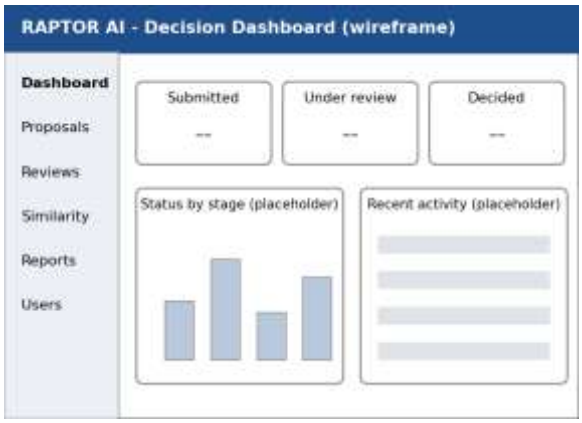


Fig. 4. Decision dashboard wireframe.

The dashboard illustrates how proposal information can be presented through counts by status, recent activity, and review progress. In a live system, each value should be traceable to its original proposal or review record, and users should be able to correct data-entry errors. Counts, charts, and scores shown in a wireframe must not be treated as experimentally validated results; the placeholders shown as dashes are intentionally empty.

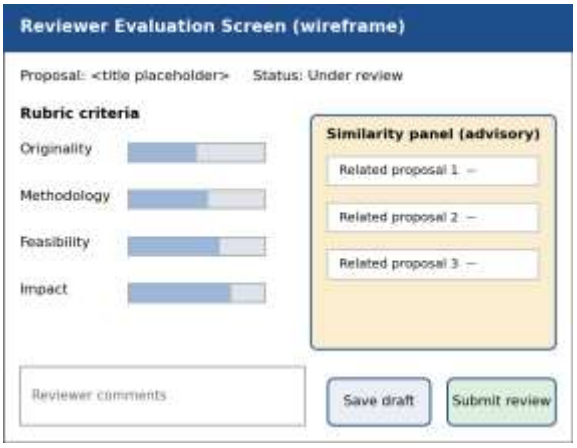


Fig. 5. Reviewer evaluation screen wireframe

The reviewer evaluation screen brings the rubric, the reviewer's comments, and the advisory similarity panel into one view. This design is intended to help reviewers examine a proposal alongside related submissions before scoring. Displayed fields and layout are illustrative placeholders; they are not results from a verified reviewer study or evidence of similarity-detection accuracy. The final decision remains with authorized people.

Capability	Admin	Researcher	Reviewer
Manage users and roles	Yes	No	No
Submit own proposals	No	Yes	No
View assigned proposals	Yes	Own only	Assigned
Enter rubric scores	No	No	Yes
View similarity report	Yes	No	Assigned
Record final decision	Yes	No	No

Illustrative access policy; to be enforced and tested on the server.

Fig. 6. Role-based access policy (illustrative)

At the workflow level, the reported design outcomes are the integrated proposal evaluation process, the separation of application and AI-assisted analysis responsibilities, the role-based access model, and the intention to consolidate proposal, review, and decision information in a centralized dashboard and report. The documents do not provide verified automated-test reports, analysis logs, load-test outputs, or validated similarity metrics. Consequently, no test pass rate, similarity accuracy, precision or recall, error rate, latency, throughput, uptime, or security assurance is reported as an observed result.

V. DISCUSSION

A. Design Implications

The principal design benefit is continuity across the proposal lifecycle. A centralized record can reduce repeated entry of information and make the current status of each proposal easier to identify, provided that documents, reviews, and decisions remain linked to persistent records. Consistent rubric criteria can improve the comparability of reviews, while a consolidated dashboard and report can make it easier for administrators to understand the information behind a decision. These benefits are plausible design goals; their magnitude has not yet been measured in the available materials.

The distinction between AI assistance and human decision-making is equally important. Text extraction can fail when documents are scanned or poorly structured, while embedding-based similarity may flag proposals that share a topic but not ideas, or miss paraphrased overlap. Generated scores may also be misleading when the stored collection is small or unrepresentative. RAPTOR AI should therefore display the compared proposals, allow reviewers to dismiss or confirm findings, and preserve the criteria used to produce each result. No single automated output should replace reviewer judgement or the final decision made by authorized people.

The security model is not an optional addition to the workflow. A platform that holds unpublished proposals and reviewer opinions needs authorization checks at the data-access layer, not only protected navigation in the interface. Role isolation is essential when administrators, researchers, and reviewers use the same system. Uploaded documents must be treated as sensitive data and processed only through suitably protected services. The project should document which of these security and privacy controls are implemented and which remain planned.

B. Limitations and Validation Plan

The main limitation is the absence of verified empirical evidence. Wireframes do not demonstrate that authentication, document ingestion, embedding, similarity search, reviewer scoring, reporting, or decision recording work end to end. Likewise, architecture diagrams do not establish that services are configured, secured, or available. The current paper therefore reports system design and prototype coverage, not production readiness or comparative superiority.

A reproducible evaluation should begin with unit tests for rubric calculations, status transitions, and permitted data updates. API integration tests should verify that proposals, reviews, and decisions are persisted correctly and that invalid data cannot produce incorrect reports. Role-based access tests should cover proposals, documents, dashboards, and reports for each role. File-handling tests should examine supported formats, size limits, malformed uploads, and unauthorized downloads. The analysis service should be tested for empty documents, extraction failures, and recovery behaviour.

Similarity analysis should be evaluated against a manually labelled set of proposal pairs, with appropriate consent and permission, using measures such as precision, recall, and agreement with human reviewers; different similarity thresholds should be compared. Reliability tests should specify workload and measure latency, throughput, failure rate, and recovery behaviour. Usability evaluation should involve administrators, researchers, and reviewers performing realistic tasks. Privacy and security reviews should examine data access, storage, document handling, and protection of reviewer identities. Reports should disclose the test setup, sample size, metrics, failures, and unresolved limitations rather than presenting only favourable outcomes.

VI. CONCLUSION AND FUTURE WORK

RAPTOR AI proposes an integrated research proposal evaluation and decision support platform that connects proposal submission, document management, semantic similarity analysis, rubric-based review, and decision reporting. Its core design contribution is a centralized and organized workflow that brings proposal records and analytical insights together. The accompanying wireframes illustrate how users could review proposal status, similarity findings, and review results within a single platform.

The available artifacts establish the intended workflow and interface direction but do not establish verified runtime accuracy, security, scalability, or end-to-end reliability. Future work includes completing the phased implementation, running the validation plan described above, tuning similarity thresholds on labelled data, adding explanation features for AI-assisted outputs, and studying reviewer workload and consistency in a controlled pilot.

ACKNOWLEDGMENT

The authors acknowledge the Department of Computer Science and Engineering, SNS College of Engineering, Coimbatore, for supporting the academic project. The authors also thank the project guide for guidance during the design and documentation process.

REFERENCES

[1] L. Bornmann, “Scientific peer review,” *Annual Review of Information Science and Technology*, vol. 45, no. 1, pp. 197–245, 2011, doi: 10.1002/aris.2011.1440450112.

[2] E. L. Pier *et al.*, “Low agreement among reviewers evaluating the same NIH grant applications,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 115, no. 12, pp. 2952–2957, 2018, doi: 10.1073/pnas.1714379115.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[4] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[5] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” *IEEE Trans. Big Data*, vol. 7, no. 3, pp.

535–547, 2021, doi: 10.1109/TBDATA.2019.2921572.

[6] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, art. 115, 2021, doi: 10.1145/3457607.

[7] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.

[8] OWASP Foundation, *OWASP Application Security Verification Standard (ASVS)*, Version 5.0.0, 2025. [Online]. Available: https://asvs.owasp.org.

[9] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, 2015, doi: 10.17487/RFC7519.