

what is currently available application provides and what extra we can provide

Existing market solutions focus heavily on **isolated tasks** in the research lifecycle. By analyzing tools like **Elicit**, **Scite.ai**, **Consensus**, **Semantic Scholar**, and **NotebookLM**, we can see where current products excel and where a unified **AI Research Paper Review Application** can create massive value.

1. What Currently Available Applications Provide

Current Tool	Primary Function	Core Strengths	Key Limitations
Semantic Scholar	Literature Discovery	Massive 230M+ paper graph, TL;DR summaries, basic citation counts.	Static indexing; no live PDF analysis, code verification, or automated review scoring.
Elicit	Data Extraction & Synthesis	Extracts methods, populations, and outcomes across papers into comparison tables.	Focuses heavily on literature reviews; does not evaluate single-paper rigor or code.
Scite.ai	Smart Citation Analysis	Classifies citations into Supporting , Mentioning , or Contradicting .	Limited full-paper PDF parsing; no automated reviewer checklist or anonymization.
Consensus	Consensus Meter & Q&A	Synthesizes yes/no answers and evidence strength across 200M+ papers.	High-level synthesis; misses deep technical sanity checks (math, code, datasets).
NotebookLM	Source-Grounded RAG	Highly accurate Q&A grounded strictly in user-uploaded PDFs.	Generic reader/notebook; lacks academic review workflows, DOI validation, or plagiarism detection.

2. What "Extra" (Value-Add Features) You Can Provide

To make your B.Tech project stand out from commercial tools, your application should bridge the gap between **isolated document analysis** and **full-stack academic peer review**.

1. Automated "Pre-Flight" Peer Review Scorecard

What's Missing Today: Current platforms don't evaluate a paper against formal review rubrics (like NeurIPS, IEEE, or Springer standards).

Your Feature: Generate a multi-dimensional metric report evaluating:

Novelty Score: Embedding distance relative to state-of-the-art literature in the same sub-field.

Reproducibility Index: Checks if datasets, hyperparameter tables, repository links, and code snippets are explicitly provided.

Clarity & Readability: Readability scores, jargon density, and section flow analysis.

2. Code & Dataset Sanity Verification Pipeline

What's Missing Today: PDF tools only look at text—they ignore code links and benchmark tables.

Your Feature:

GitHub/Zenodo Link Checker: Scrape GitHub links in the paper to verify repo status, commit recency, and `requirements.txt` existence.

Metric Audit: Automatically parse performance tables (e.g., Accuracy, F1-Score) and compare them against established benchmarks (like PapersWithCode) to flag unrealistic claims.

3. Citation Authenticity & Hallucination Guard

What's Missing Today: Authors (or AI writing tools) sometimes cite non-existent papers, preprints with dead links, or retracted studies.

Your Feature: Cross-check every cited DOI in the PDF against **CrossRef / OpenAlex / Retraction Watch APIs** to automatically tag:

- **Verified & Indexed**
- **Preprint / Unreviewed**
- **Retracted Paper or Unverifiable DOI**

4. Interactive Knowledge & Claim Mapping

What's Missing Today: Citation maps like Connected Papers show paper-to-paper links, but not claim-to-evidence links.

Your Feature: Generate an interactive claim map that connects the paper's primary claims (e.g., "Model X achieves 95% accuracy using ResNet-50") directly to the corresponding figures, tables, and references inside the document.

5. Automated Double-Blind Anonymizer

What's Missing Today: Preparing papers for double-blind submission requires manual redaction of author details, grants, and explicit self-citations.

Your Feature: Provide an automated NLP redaction module that detects and sanitizes author names, institutional affiliations, GitHub usernames, and self-referential phrases (e.g., "In our previous work [3]...").

3. Summary Comparison: Current Stack vs. Your Application

Existing Tools Spectrum

Semantic Scholar (Discovery)
Elicit (Extraction)
Scite.ai (Citation Context)
NotebookLM (PDF Q&A)

Your Unified Application

RESEARCH PAPER REVIEW APPLICATION
• All-in-one PDF Ingestion & Chat
• Smart Peer-Review Scorecard
• Code / Dataset / DOI Integrity Audit
• Citation Graph & Claim Verification

Is there any such application available in the market which matched these requirement

Yes, there are applications in the market that address parts of these requirements, but **none of them combine all these capabilities into a single, open full-stack application.**

Existing tools are fragmented: some act only as PDF chat readers, others focus purely on literature discovery, and a few provide commercial manuscript pre-checks.

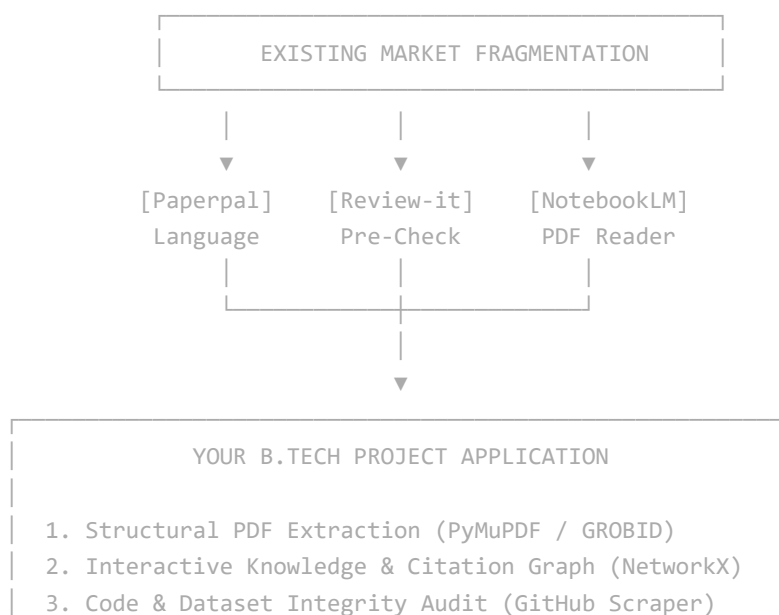
1. Existing Market Applications Addressing Similar Requirements

Tool	What It Matches	Where It Falls Short Compared to Your Plan
Review-it.ai / ScholarsReview	Commercial pre-submission tools that check manuscripts for desk rejection risks, basic citation validity, and language quality.	Gaps: Proprietary/black-box algorithms; no interactive citation network graph, no repository code inspection, and no local vector storage/privacy controls.

Tool	What It Matches	Where It Falls Short Compared to Your Plan
Paperguide.ai / Elicit	Systematic review, automated data extraction tables from multiple papers, and evidence mapping.	Gaps: Designed primarily for multi-paper literature reviews rather than evaluating a single paper's code, mathematical soundness, or peer-review score.
Paperpal	Language, grammar, and IEEE/APA compliance checking for manuscript preparation.	Gaps: Strictly an editing assistant; lacks RAG Q&A, graph visualization, and deep methodology auditing.
NotebookLM / ChatPDF	PDF ingestion, semantic search, and grounded retrieval-augmented Q&A.	Gaps: Generic document readers that do not understand academic peer-review rubrics, DOIs, or citation integrity checks.

2. Why Your B.Tech Project Has a Clear Unique Value Proposition (UVP)

While commercial platforms exist, your project stands out by **unifying four distinct systems** into one open-architecture pipeline:



Key Differentiators to Emphasize in Your Project Presentation:

End-to-End Transparency: Unlike commercial paywalled platforms (like Review-ly), your application provides transparent scoring logic, visual graph embeddings, and open-source models (e.g., SciBERT, LangChain, ChromaDB).

Code & Repository Validation: Most tools evaluate only the text; your platform inspects linked GitHub repositories and dataset references to calculate a **Reproducibility Index**.

Citation Hallucination & Retraction Guard: By linking directly to live APIs like CrossRef and Retraction Watch, your system actively flags dead DOIs and retracted papers.