

Project Proposal Report

Collaborative Research Paper Management System

Batch ID: 22CSTA015

Submitted by

Bethina Sowmya Sri Durga (23K65A0601)

Deva Bhavya Sri Lakshmi (22K61A0618)

Bobbili Pushpa Nagavalli (22K61A0608)

Mallipudi Vani Satya Sri (22K61A0636)

Under the Guidance of:

G.Venkata Subbarao

Collaborative Research Paper Management System

1. Project Title

Collaborative Research Paper Management System (Web Application)

2. Introduction & Background

2.1 General Context:

Academic research is often done in teams where people work together to plan, write, and review research papers. These teams can include students, teachers, and researchers from different places. To work efficiently, they need tools to help them write, manage, and edit documents together in real time. Traditional tools like emails and separate file-sharing apps are not enough for smooth collaboration.

2.2 Specific Problem/Gap:

Currently, teams use a mix of email, cloud storage, and local writing software. This causes problems like:

- Confusing file versions
- Difficulty giving feedback
- Hard to manage citations and references
- No proper tracking of changes

There is a need for one system that includes all these features in a single place to make research writing easier and faster.

2.3 Literature Review Summary:

Some tools like Google Docs and Overleaf allow collaborative writing. Tools like Zotero and Mendeley help with managing citations. But no single tool combines all the needs of research paper writing — from planning and writing to citations and feedback — in one easy-to-use system. This project will combine these features to create a better solution for academic teams.

2.4 Significance:

This project will build a web-based platform where research teams can work together easily. It will reduce confusion, improve feedback and organization, and help manage references and writing tasks. It can help researchers focus more on the content and less on managing tools.

3. Project Goals and Objectives

3.1 Overall Goal:

To build an easy-to-use web platform where academic teams can write, manage, and share research papers together.

3.2 Specific Objectives:

- Create a secure login system and project workspace (Weeks 1–3)
- Build a writing tool that allows multiple users to write and edit together (Weeks 3–7)
- Add version tracking and change history (Weeks 5–8)
- Add a tool to manage references and automatically generate citations (Weeks 7–10)
- Add task management and chat features for better teamwork (Weeks 9–11)
- Test the system with users and improve based on feedback (Weeks 11–12)
- Add AI tools to generate summaries, suggest keywords, and check grammar to improve writing productivity

4. Methodology

4.1 Research Design:

We will follow an agile software development approach, which means building and improving the system in steps with user feedback. The design will focus on being easy to use, secure, and able to grow in the future.

4.2 Data Collection:

- User Data: Login details and profiles
- Project Data: Document content, comments, and versions
- Reference Data: Details of citations and sources
- Task Data: Task info like deadlines and assignments
- Usage Data: Logs and performance records

Data will come from users using the system. During testing, we'll use fake data to simulate usage.

4.3 Data Analysis:

System Architecture:

- **Frontend:** Built with React or similar
- **Backend:** Built with Node.js, Django, or other server tools
- **Database:** Stores users, projects, and tasks
- **Real-time Sync:** Uses WebSockets for live editing and chat
- **AI Features:** Uses GPT-based APIs to help users auto-generate abstracts, summarize paragraphs, check grammar, and recommend keywords based on content. These tools support summarization, grammar checking, abstract generation, and keyword suggestions to improve writing efficiency.

Features to Build:

- User registration, login, and access roles
- A shared writing space with live collaboration and comments
- Automatic saving of document versions

Tools to add and manage references in different citation styles

- Task tracker and team chat system
- Optional AI assistance for improving writing productivity

Testing:

- Make sure the system works with multiple users at once
- Test system speed and response time
- Check for security issues
- Get feedback from researchers on usability

4.4 Ethical Considerations:

- Keep research papers and user data safe with encryption
- Show who made what changes to give proper credit
- Follow privacy laws like GDPR
- Make sure the system works for users with disabilities
- Keep backups in case of system issues
- Ensure that AI features are used to assist writing, not for generating unethical or plagiarized content
- Clearly inform users that all submitted content must follow academic integrity policies

Plagiarism detection is integrated using third-party APIs. The system warns users about unethical writing practices, highlights duplicate content, and logs all checks for academic integrity.

5. Timeline and Work Plan

5.1 Major Milestones:

- Weeks 1–3: Login system and project creation
- Weeks 3–7: Writing tool and real-time editing
- Weeks 5–8: Version control system
- Weeks 7–10: Citation and reference manager
- Weeks 9–11: Task and chat module
- Weeks 11–12: Final testing and improvements
- Weeks 13–14: Report writing and presentation

5.2 Detailed Tasks:

- Design database and create login system
- Build and connect the text editor
- Set up version tracking
- Build citation import and generation
- Add task and notification system
- Combine everything and test it all together
- Write project report and prepare for final submission

5.3 Contingency Plan:

- If real-time editing is hard, use existing libraries like Yjs
- If speed issues come up, optimize database and APIs
- If security issues are found, fix them quickly and test again
- If users don't like the UI, update it based on feedback
- If citation formats are too complex, support just the common ones first

6. Expected Outcomes and Dissemination

6.1 Expected Deliverables:

- A working collaborative web app for research writing
- Real-time editor with version tracking
- Tools for citation and literature management
- A task and chat module
- Optional AI tools for summarization, grammar checks, and keyword suggestions
- Complete documentation and report
- Codebase that can be shared or published online

6.2 Contribution to Knowledge:

This project shows how to apply modern web tools and real-time collaboration to help academic writing. It will help researchers work more effectively and may lead to future systems that improve how academic papers are written and managed.

6.3 Dissemination Plan:

- Submit the report to the university
- Present the system at college exhibitions or demo days
- Submit the project as a research paper to academic conferences
- Optionally, publish the code as open source to help others

7. Use Case Scenarios

• Collaborative Paper Writing

Multiple users edit the same research document in real-time. Each user's edits are tracked, and comments can be added inline for feedback. This simulates a real-world research team working together remotely.

• AI-Powered Writing Assistance

Users use integrated AI tools to summarize paragraphs, generate abstracts, check grammar, and suggest keywords. This saves time and improves academic writing quality.

• Plagiarism Detection Before Submission

Before finalizing the document, a user runs a plagiarism check. The system highlights duplicate content and provides originality score, allowing users to revise accordingly.

- **Citation Management with IEEE Style**

Users add references manually or by importing BibTeX files. The citation manager supports IEEE, APA, MLA, and formats references automatically in the chosen style.

- **Task Management and Communication**

Team members use the chat system to coordinate tasks, comment on document sections, and track assignments through the built-in task manager.

- Integrated plagiarism checker with originality report
- AI-powered writing assistant tools with GPT-based APIs
- Inline comments and project-wide chat functionality