Graphic Era Hill University
DEHRADUN • BHIMTAL • HALDWANI

PROJECT AND TEAM INFORMATION

Project Title

Plagiarism Detection using String Matching Algorithms

Student/Team Information

Student / Team Information

Team Name:	DynamicPathFinders
Team member 1 (Team Lead)	Negi, Aditya - 230114984 ADITYA.230114984@gehu.ac.in 
Team member 2	Rawat, Sarthak – 230112045 SARTHAKRAWAT.230112045@gehu.ac.in 

Team Name:	DynamicPathFinders
Team member 3	Kumar, Aman - 230112444 AMANKUMAR.230112444@gehu.ac.in 
Team member 4	Singh, Anshika – 230122594 ANSHIKASINGH.230122594@gehu.ac.in 

Project Abstract

The project aims to develop an efficient plagiarism detection system leveraging classical string matching algorithms such as Rabin-Karp, Knuth-Morris-Pratt (KMP), and Longest Common Subsequence (LCS). The system compares documents for exact and near-exact text matches, generates similarity scores, and highlights

copied sections. With a focus on transparency and scalability, the solution features a user-friendly web interface for document upload, real-time analysis, and structured reporting in CSV and JSON formats, supporting academic integrity and fair evaluation processes.

Updated Project Approach and Architecture

- **Data Preprocessing:** Documents are loaded (plain text, CSV), cleaned, tokenized, and normalized.
- **String Matching Algorithms:** Implements Rabin-Karp, KMP, LCS, and Jaccard Similarity for comprehensive comparison.
- **Plagiarism Scoring:** Calculates plagiarism percentages and highlights matching text.
- **Reporting:** Results are structured for export in CSV and JSON formats.
- **Web Interface:** Enables document upload and interactive result visualization.
- **Architecture:** Backend handles processing; frontend manages user interaction and display[1][2].

Roles and Responsibilities

Team Member	Role(s) and Responsibilities
Aditya Negi	Backend development, Data preprocessing, Text highlighting, Plagiarism percentage score calculation, Tester
Sarthak Rawat	Implemented Rabin-Karp string matching algorithm, Compared accuracy with KMP and LCS algorithms
Anshika Singh	Research on integration of machine learning/cloud technology into the project
Aman Singh	Created interactive frontend, Login page, and Register page

- **Backend Development, Data Preprocessing, Text Highlighting, Plagiarism Percentage Score, Testing:**
  Aditya Negi is responsible for implementing and maintaining the backend logic, handling data preprocessing, highlighting plagiarized text, calculating plagiarism scores, and conducting system testing.
- **String Matching Algorithm Implementation and Accuracy Comparison:**
  Sarthak Rawat implemented the Rabin-Karp algorithm and performed comparative analysis with KMP and LCS algorithms for accuracy evaluation.

- **Research on Advanced Technologies:**
  Anshika Singh conducted research on how machine learning and cloud technologies can be integrated to enhance the plagiarism detection system.
- **Frontend and User Authentication:**
  Aman Singh developed the interactive frontend as well as the login and registration pages to ensure a seamless user experience.

Tasks Completed

- Finalized project scope, objectives, and selected algorithms.
- Set up the development environment and installed required libraries.
- Implemented document loading and preprocessing.
- Developed and tested core string matching algorithms.
- Built the basic web interface for document upload and result display.
- Integrated backend and frontend for real-time plagiarism detection.
- Conducted initial testing with sample datasets[1][2].

Challenges/Roadblocks

- Designing accurate plagiarism percentage calculation logic.
- Automated report generation in CSV and JSON formats.
- Performance optimization for large documents.
- Precise highlighting of plagiarized sections.
- Comprehensive testing coverage.

Planned Solutions:

- Research established methods for plagiarism scoring.
- Use Python libraries for report generation.
- Optimize code for performance.
- Enhance match-highlighting logic.

- Expand test cases for robustness.

Tasks Pending

- Implement and validate plagiarism percentage calculation.
- Develop and test CSV/JSON report generation.
- Finalize and polish the web interface.
- Conduct extensive testing.
- Prepare final documentation and demo materials.

Project Outcome/Deliverables

- Fully functional plagiarism detection tool.
- User-friendly web interface.
- Structured reports in CSV and JSON formats.
- Highlighted matching text sections.
- Comparative analysis of algorithm accuracy and performance.
- Comprehensive documentation and demo video.

Progress Overview

All major phases except plagiarism percentage calculation and report generation are complete. Core algorithms, preprocessing, and the web interface are functional and tested. Remaining tasks are in progress, with no major delays anticipated[2].

Codebase Information

https://github.com/adinegi8273/Plagiarism-detection-using-string-matching-algorithms

Testing and Validation Status

- Preliminary tests on sample datasets verify algorithm correctness and detection accuracy.
- Pending: Full-scale testing for percentage calculation, report generation, and edge cases.

Deliverables Progress

Deliverable	Status
Source Code (Algorithms, Backend)	Completed
Web Interface	Completed
Plagiarism Percentage Calculation	In Progress
CSV/JSON Report Generation	In Progress
Final Documentation	Pending
Demo Video	Pending
Testing and Performance Evaluation	In Progress

⁂

1. Aditya-pbl-proposal.docx
2. Project-Progress-Temp-GEHU.docx