

TechLib: Plagiarism Free Project Management

Susan Fernandes¹, Iqra Khan², Shaun Mendes³, Monali Shetty⁴ and Prachi Patil⁵

¹ Fr Conceicao Rodrigues College of Engineering, Bandra/Computer Engineering, Mumbai, India
Email: susanfernandes1305@gmail.com

²⁻³ Fr Conceicao Rodrigues College of Engineering, Bandra/Computer Engineering, Mumbai, India
Email: {iqra.khann610@gmail.com, shaunkmmendes@gmail.com}

⁴⁻⁵ Fr Conceicao Rodrigues College of Engineering, Bandra/Computer Engineering, Mumbai, India
Email: {monali_shetty@fragnel.edu.in, prachi@fragnel.edu.in}

Abstract— TechLib is an innovative platform that revolutionizes the sharing and showcasing of original college projects, addressing common challenges in existing project management systems. At its core is a rigorous plagiarism detection system employing a cosine similarity algorithm to identify original work. The platform facilitates seamless communication between administrators and students, streamlining project approval, rating, and feedback processes. TechLib stands out with its innovative plagiarism checker, ensuring that no two similar projects are uploaded to the platform. This unique feature not only promotes academic integrity but also encourages the flourishing of original ideas, effectively addressing the issue of concept duplication. The platform facilitates seamless collaboration among students while enabling faculty members to easily monitor, verify, and provide feedback on their students' projects. This paper delves into the platform's architecture and methodology, exploring how TechLib transforms the academic landscape by fostering originality and improving project quality.

Index Terms — Project Management System, Plagiarism Detection, Cosine Similarity Algorithm, Project Quality Management, Collaborative Learning Platform.

I. INTRODUCTION

The old ways of teaching and learning have evolved significantly in an educational environment where technology is having a greater and bigger impact. The relevance of developing creative platforms that support collaborative learning and knowledge sharing is highlighted by the growing prominence of student-led projects and the requirement for the practical, real-world application of theoretical knowledge.

A. An Innovative Method for Education

The term "Technology Library," or "TechLib," refers to a novel online resource developed to fill the gap between academic study and real-world use. It acts as an innovative platform for academic institutions, business leaders, and students to exchange, present, and celebrate creative ideas and projects. TechLib enhances the educational experience by extending the bounds of traditional learning environments, enabling students to take an active role in building innovative projects.

B. Technology to Enhance Academic Performance

With the use of contemporary technologies consisting of Tailwind CSS, React.js, and Next.js, TechLib is dedicated to offering a simple, user-friendly platform that improves the quality of learning. TechLib utilizes technology to make learning more interesting, interactive, and available to a larger audience. It also makes the process of broadcasting and gaining information simple.

C. Empowering the Next Generation of Innovators

Fundamentally, TechLib is dedicated to empowering students by providing them with a platform on which to present their original ideas, projects, and inventions to a worldwide audience. TechLib provides students with a sense of pride, inspiration, and self-confidence by acknowledging and encouraging their efforts. This concept drives students to pursue their passions, explore novel opportunities, and make significant contributions to society.

D. Promoting Collaboration and Community Engagement

Beyond individual achievements, TechLib fosters the collaborative spirit by encouraging students, educational organizations, and executives to converge on impacts share innovations, collaborate on projects, and promote innovation in education. TechLib attempts to build a community of lifelong learners who are dedicated to pushing the envelope of what is possible and having a good impact on the world by providing a collaborative and inclusive learning environment.

E. A Focus on Continuous Progress

TechLib aims to contribute to the ongoing evolution of education by promoting a culture of continuous improvement. By providing a platform for students to showcase their innovative projects and receive constructive feedback from administrators and peers, TechLib encourages iterative learning and refinement of ideas. This emphasis on continuous improvement empowers students to strive for excellence and adapt to the changing demands of the modern world, ultimately enhancing the quality and relevance of education.

II. LITERATURE SURVEY

A comprehensive analysis of plagiarism detection in college programming courses from 2015 to 2020 highlights a significant increase in global research on this topic. The study reveals that Java, C++, C, and Python are the most frequently analyzed languages in plagiarism detection research. Tools such as MOSS (Measure of Software Similarity) and JPlag have gained prominence for their effectiveness in identifying code similarities. Notably, static analysis has emerged as the dominant methodology for detecting plagiarized source code, offering a balance of efficiency and accuracy. This research underscores the growing emphasis on maintaining academic integrity in computer science education and provides a foundation for understanding current trends in plagiarism detection techniques. [1]

An innovative approach to addressing collaboration challenges in multi-developer projects, particularly in distributed teams, is proposed in a study focusing on change modeling. The research introduces a methodology that treats changes as primary entities, facilitating more effective collaboration. By prioritizing synchronous changes, this approach aims to mitigate coordination issues among multiple developers, potentially enhancing both productivity and team cohesion. This work is particularly relevant to large-scale software development projects and offers insights into improving collaborative coding practices. [2]

The proposed approach aims to combat plagiarism within proprietary software by utilizing Mining Software Repositories (MSR) techniques to create programming style profiles for maintenance engineers. By analyzing version histories, the method seeks to identify similarities in code structure and style, offering a proactive solution to maintain the integrity of closed-source software systems. [3]

This study introduces SEREBRO, a lightweight courseware designed to enhance student collaboration in project teams. By integrating an idea forum with project management tools, SEREBRO provides a dynamic platform for brainstorming, coordinating tasks, and tracking progress. The system aims to revolutionize collaborative learning by empowering teams to work efficiently and cohesively towards their goals. [4]

This paper provides a comprehensive examination of the vital role international collaborative projects play in shaping modern engineering education. Drawing from real-world examples and insightful recommendations, the paper delves into the complexities of managing such endeavors effectively. It highlights the transformative impact of cross-cultural exchanges on student learning outcomes and underscores the importance of fostering

sustainable partnerships across borders. This seminal work serves as a guiding resource for educators and institutions seeking to navigate the intricacies of global collaboration in engineering education. [5]

The paper presents a comprehensive methodology for identifying code clones, focusing specifically on statement-level features. The research highlights the significant impact of duplicate code on software quality and maintenance, emphasizing the need for efficient detection techniques. By concentrating on statement-level characteristics, the proposed approach aims to enhance both the precision and effectiveness of clone detection. [6]

This paper is a significant contribution to the field of plagiarism detection in programming assignments. The paper underscores the critical need for accurate detection methods in educational settings, where the integrity of student work is paramount. By leveraging features extracted from ultra-fine-grained repositories, the authors propose a novel approach to enhance detection accuracy, potentially mitigating the widespread issue of academic dishonesty in programming coursework. This innovative method holds promise for educators and institutions seeking robust tools to uphold academic integrity standards in computer science education. [7]

The paper presents a comprehensive examination of source code plagiarism within academic settings. The paper introduces a valuable public dataset aimed at facilitating the evaluation and comparison of detection methodologies. Through meticulous evaluation, the authors shed light on the efficacy of different information retrieval (IR) techniques in identifying instances of plagiarism in source code. This research significantly contributes to the ongoing discourse surrounding academic integrity and provides practical tools for educators and researchers to combat plagiarism effectively. [8]

This paper offers an empirical study comparing different code representation techniques for clone detection, addressing a crucial aspect of software maintenance. By evaluating 12 clone detection algorithms across text-based, token-based, tree-based, and graph-based approaches, the authors provide valuable insights into their relative effectiveness. The study's findings suggest that while token and tree-based representations perform well for simpler clones, graph-based approaches excel with complex code. The proposed TACC tool, which combines token-based and tree-based methods, demonstrates superior performance, making a significant contribution to the field of code clone detection. This work is particularly relevant for software engineers and researchers focused on optimizing clone detection techniques in large-scale projects. [9]

This paper explores the enhancement of source code representations for deep learning models through the integration of static analysis and additional contextual information like bug reports and design patterns. By augmenting traditional ASTNN methods, the authors significantly improve the performance of deep learning tasks such as code classification and clone detection. This study is a notable contribution to the field of program comprehension, offering a novel approach that considers both the code's structure and its associated context. The findings provide a foundation for future research on the intersection of static analysis and deep learning, with potential applications in software engineering and automated code analysis. [10]

This paper investigates the application of machine learning and deep learning techniques to detect plagiarism in C programming assignments. The authors compare various methods, including KNN, SVM, D-Trees, RNNs, and attention-based transformer networks, highlighting their strengths and weaknesses in the context of source code plagiarism detection. The research presents a comprehensive dataset of code pairs and introduces a tool that leverages these methods to improve detection accuracy. This work contributes to the ongoing efforts to combat academic dishonesty in programming courses by providing educators with advanced tools to identify and address plagiarism more effectively. [11]

This comparative study offers an in-depth analysis of computer-based plagiarism detection techniques, emphasizing the challenges posed by modern plagiarism practices, such as text modification and idea theft. The authors explore various detection methods using the PAN Dataset 2009-2011, providing insights into the strengths and limitations of each approach. By categorizing plagiarism into verbatim and smart plagiarism, the paper highlights the evolving nature of academic dishonesty and the need for more sophisticated detection tools. This research is particularly relevant for academic institutions seeking to enhance their plagiarism detection capabilities and maintain the integrity of scholarly work. [12]

III. RESEARCH METHODOLOGY

A. Current Challenges in Project Approval

Student projects are not promptly recognised and evaluated in Indian universities due to a number of inefficiencies in the manual project approval process. It is typical to experience delays of up to four weeks, which severely hinders the timely evaluation and recognition of student work.

Furthermore, there is no official industry support for the current approval system, which limits the projects' real-world relevance and practical applicability. Projects run the risk of not meeting professional standards or successfully addressing real-world challenges in the absence of industry validation, which would reduce their practical value.

Another critical issue is the inadequate handling of similar or plagiarized content. The current methods for detecting plagiarism are often insufficient, allowing projects with copied material or significant similarities to other works to pass through without rigorous scrutiny. This oversight compromises academic integrity and undermines the recognition of genuinely original work, impacting the credibility of the approval process and the quality of student contributions.

B. TechLib: A Streamlined and Effective Solution

TechLib tackles these issues by offering a more efficient platform for project submission, evaluation, and approval. The platform introduces a multi-stage approach to enhance efficiency, fairness, and academic integrity. Initially, students submit their projects through the TechLib platform for an internal review conducted by their respective colleges. This preliminary review ensures that projects meet institutional standards before proceeding to further stages. Once a project is approved at the college level, it undergoes a rigorous plagiarism check using the cosine similarity algorithm. This algorithm compares the submitted project against a vast database of existing sources to detect potential similarities and ensure originality. Projects flagged for potential plagiarism are subject to additional review to confirm their uniqueness. A team of industry professionals then assesses the initiatives for their real-world applicability and practical worth, providing significant professional insights. Finally, universities provide final permission before projects are uploaded on the portal.

This multi-stage procedure provides quick approvals, promotes fairness and academic integrity through plagiarism detection, and includes industry certification for practical application.

Based on the expert's evaluation, the project is either approved or rejected. If the expert confirms that the project is indeed original and the student's work, it gets the green light for uploading to TechLib. Approved projects are then made available on the platform for others to view and learn from.

However, if the expert finds that the project is mostly copied from other sources, it gets rejected. This ensures that only original and high-quality projects are shared on TechLib, maintaining academic integrity and giving credit where it's due.

This thorough process not only ensures the quality of projects on TechLib but also promotes fairness and honesty in academic work.

3.1. TechLib: A Streamlined and Effective Solution

TechLib values academic integrity and employs a strong plagiarism detection system. This technology automatically analyses submitted projects to a large database of existing sources to find any major similarities. Projects that include potentially copied content are tagged for additional evaluation to ensure originality and correct attribution. This critical step protects the platform's legitimacy while reinforcing ethical academic practices.

The plagiarism detection model used in the project uses the following algorithms to analyze and compare textual content for similarity:

A. Cosine Similarity Calculation

The model utilizes the cosine similarity algorithm, a widely used technique for measuring similarity between two vectors. Initially, the text documents undergo text vectorization, where each word in the document is transformed into a numerical representation. Using the Counter class from the collections module, the frequency of each word in the document is computed, creating a dictionary-like object. The model then calculates the dot product of the two vectors, representing the documents. By computing the magnitude of each vector and its dot product, the model determines the cosine similarity, ranging between -1 and 1, where higher values indicate greater similarity. If the computed cosine similarity exceeds a predefined threshold (0.8 in this case), the model detects plagiarism between the documents.

B. Text Vectorization

Before cosine similarity calculation, the text documents are converted into numerical vectors through text vectorization. This process involves

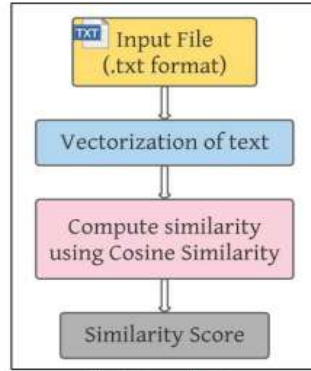


Fig. 1 Plagiarism Module Working

representing each document as a set of features or dimensions, with each unique word being a key and its frequency in the document being the corresponding value. Text vectorization is accomplished using the Counter class, which facilitates the creation of a dictionary-like object containing word frequencies. By transforming text documents into vectors of word frequencies, the model can effectively measure the similarity between documents based on their textual content.

These algorithms collectively enable the plagiarism detection model to analyze text documents and identify instances of plagiarism by comparing their textual similarity.

C. Implementation

Sci-kit-learn is a built-in library for machine learning applications. It includes tools for machine learning and statistical modelling. This library was utilised in the proposed system to extract text features. The Tf-idf vectorizer converts textual data into a numerical array for word embedding purposes. Then this textual data is transformed into vectors and used to find the similarities between two files which helps in calculating the cosine similarity algorithm. The cosine similarity algorithm calculates the angle between the two text file vectors. The computation yields a score ranging from 0 to 1, indicating the level of similarity between two input files.

The implementation approach involves key steps as shown in Fig. 1, including:

1. Input File: The file is intended to be the input for the plagiarism detection system. It should be in text format or readme file of the project.
2. Vectorization of text: Sci-kit's built-in features convert textual input into vector format.
3. Calculate similarity: Two text files are compared using Cosine Similarity. To calculate the similarity between two vectorized text files, use the dot product (cos), where it represents the angle between the vectors.

Step 1. The TF-IDF approach was employed in this system. TF-IDF refers to Term Frequency - Inverse Term Frequency. This algorithm prioritises recurrent words based on their frequency and relevance in the input context.

TF-IDF Calculation:

- Term Frequency (TF):

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

(1)

- Inverse Document Frequency (IDF):

$$IDF(t, D) = \log \left(\frac{\text{Total number of documents } |D|}{\text{Number of documents containing term } t} \right)$$

(2)

- TF-IDF:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

(3)

Step 2. Cosine similarity uses vectors as input and returns their cosine. This programme determines the cosine of two vectors based on their angle. The output in 0-1 format indicates the similarity score.

Cosine Similarity Calculation:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{|A||B|}$$

(4)

Where:

- $A \cdot B$ is the dot product of vectors A and B
- $|A|$ and $|B|$ are the magnitudes (Euclidean norms) of vectors A and B, respectively.

The dot product $A \cdot B$ is calculated as:

$$A \cdot B = \sum_{i=1}^n A_i \times B_i$$

(5)

Where A_i and B_i are the components of vectors A and B

The magnitude of a vector A is calculated as:

$$|A| = \sqrt{\sum_{i=1}^n A_i^2}$$

(6)

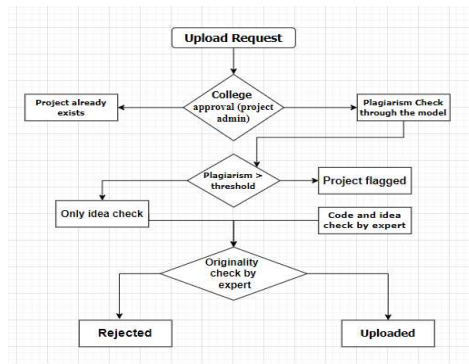


Fig.2. Proposed architecture workflow

TechLib Home Dashboard Projects Profile

Dashboard

Approved Projects
2

Pending Projects
1

Rejected Projects
1

Search View

☒ Approved ☐ Rejected

0 of 4 rows selected

☐	Name	Category	Approved	Rejected	
☐	susan	Blockchain	X	X	...
☐	susan	Machine Le...	X	✓	...
☐	ahmed	Machine Le...	✓	X	...
☐	hassan	IoT	✓	X	...

Rows per page: 10 Page 1 of 1

Fig. 3. Dashboard Module

Fig 2. shows the proposed architecture diagram. The User Authentication Module serves as the system's cornerstone, managing user accounts and ensuring secure access for students, faculty, and administrators. It offers user registration, login, and authentication processes, empowering users to securely create and manage their profiles. This module also includes essential features such as password resets, account recovery, and user role assignment to maintain appropriate permissions within the system.

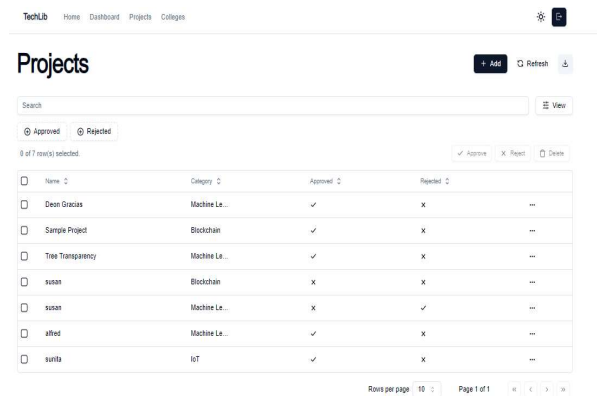
The Student Project Submission Module provides an efficient platform for students to submit their projects. It streamlines project uploading, descriptions, and tagging, making the process user-friendly and accessible. Students can attach relevant documents, describe their work in detail, and even save project drafts for gradual completion and submission.

The Student Approval Model Module utilizes Natural Language Processing (NLP) and predefined criteria for the initial screening of student-uploaded projects. It ensures projects meet basic standards, providing automated feedback to help students improve their work.

The Industry Expert Validation Module engages industry experts to evaluate projects for real-world relevance and practical value, adding professional insights. The College Review and Approval Module empowers colleges to ensure projects align with their educational objectives. The Mumbai University Final Approval Module ensures high standards before publishing approved projects.

IV. RESULTS & ANALYSIS

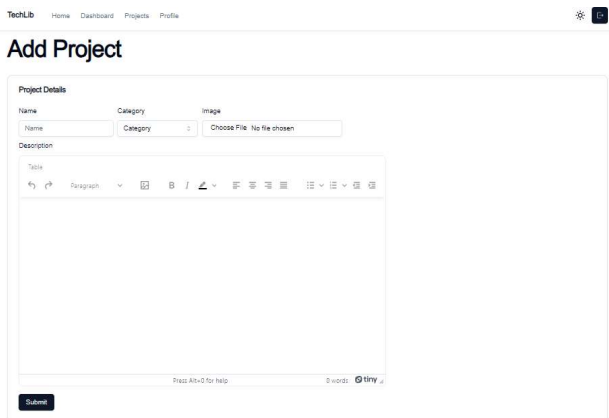
The project's execution has transformed the project approval process and made sure that every project presented is essentially unique and original. By utilising a plagiarism checker integrated into the project, the project ensures that every student's project is authentic and unique. This creative method preserves the integrity of the project approval procedure while also giving the academic community greater trust in the validity of student work.



The screenshot shows the 'Projects' page of the TechLib application. It features a search bar, a 'View' button, and a table of project submissions. The table has columns for Name, Category, Approved, and Rejected. There are 7 rows of data, with the first row being a header. The table shows various projects like 'Dean Gracia', 'Sample Project', 'Tree Transparency', 'sutan', 'sutan', 'afed', and 'kurita' with their respective categories and approval/rejection status.

Name	Category	Approved	Rejected
Dean Gracia	Machine Le...	✓	X
Sample Project	Blockchain	✓	X
Tree Transparency	Machine Le...	✓	X
sutan	Blockchain	X	X
sutan	Machine Le...	X	✓
afed	Machine Le...	✓	X
kurita	IoT	✓	X

Fig. 4. Project Approval Module



The screenshot shows the 'Add Project' page of the TechLib application. It features a form for adding a new project with fields for Name, Category, Image, and Description. The form includes a 'Submit' button and a 'Cancel' button. The description field is a rich text editor with various formatting options.

Fig. 5 Add New Project Module

The approval process functions as an effective guarantee for the legitimacy and quality of student work as shown in Fig.4 is assisted by both administrative evaluation and the plagiarism checker. The system carefully reviews project submissions for any indications of plagiarism or resemblance to already published works by utilising advanced techniques like cosine similarity in the plagiarism checker. In addition to preserving academic standards, this stringent evaluation procedure encourages students to be innovative and honest in their work.

Additionally, the project's implementation has made it much easier to store and manage a database of student projects for colleges. The technology streamlines the process of project documentation and archiving by offering a centralized platform for project submission and approval. This facilitates universities in keeping thorough records of student accomplishments. In addition to making student projects easy to access, this centralized database helps universities monitor how student work grows and improves over time.

In conclusion, the project's execution goes beyond just making it easier for academic projects to be approved; it also allows students to display their originality and creativity through mini-projects guaranteeing their originality and authenticity and at the same time facilitating administrative approval through a thorough evaluation process, and giving students a platform to present their projects and ideas.

V. CONCLUSION

In conclusion, the project has accomplished an important objective with the effective development and functioning of crucial features. Next.js is used to provide a user-friendly front end that allows students to quickly submit their work. One of the project's main features is the approval system, which simplifies the screening procedure and gives students important input. Additionally, the addition of a college approval module gives educational establishments the authority to support initiatives that meet their goals and curriculum requirements. The project fosters academic quality and creativity while facilitating collaborative learning by utilizing technology to bridge the gap between academics and real-world applications. As the initiative develops further, it has the potential to completely transform the nature of education by encouraging collaboration among students and institutions and encouraging them to welcome innovation.

REFERENCES

- [1] Rodrigo C Aniceto Maristel, Carla Castanho Dilma Da Silva, "Source Code Plagiarism Detection in an Educational Context: A Literature Mapping", December 2021.
- [2] Lile Hattor, "Enhancing Collaboration of Multi-developer Projects with Synchronous Changes", October 2011.
- [3] Girish Maskeri, Deepthi Karnam, Sree Aurovindh Viswanathan, Srinivas Padmanabhuni, "Version History Based Source Code Plagiarism Detection in Proprietary Systems", January 2013.
- [4] R.F. Gamble, M.L. Hale, "SEREBRO: Facilitating Student Project Team Collaboration", October 2011.
- [5] Åsa Cajander, Mats Daniels, Can Kultur, Lori Russell Dag, Cary Laxer, "Managing International Student Collaborations", February 2013.
- [6] Lalitha Rangarajan, Sudhamani Mallaiah, "Code clone detection: An approach based on statement-level features".
- [7] Vedran Ljubovic, Enil Pajic, "Plagiarism Detection in Computer Programming Using Feature Extraction from Ultra-Fine-Grained Repositories"
- [8] Oscar Karnalim, Setia Budi, Hapnes Toba, Mike Joy, "Source Code Plagiarism Detection in Academia with Information Retrieval: Dataset and the Observation"
- [9] Y. Wang, Y. Ye, Y. Wu, W. Zhang, Y. Xue, and Y. Liu, "Comparison and Evaluation of Clone Detection Techniques with Different Code Representations," 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023, pp. 332-344, doi: 10.1109/ICSE48619.2023.00039
- [10] X. Guan and C. Treude, "Enhancing Source Code Representations for Deep Learning with Static Analysis," 2024 IEEE/ACM 32nd International Conference on Program Comprehension (ICPC), Lisbon, Portugal, 2024, pp. 64-68. doi: 10.1109/ICPC56422.2024.10556422.
- [11] A. Eppa and A. Murali, "Source Code Plagiarism Detection: A Machine Intelligence Approach," 2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAEECC), Bengaluru, India, 2022, pp. 1-7, doi: 10.1109/ICAEECC54045.2022.9716671.
- [12] Mansoor, Marwah & Al-Tamimi, Mohammed. (2022). "Computer-Based Plagiarism Detection Techniques: A Comparative Study." The International Journal of Nonlinear Analysis and Applications (IJNAA). 13. 3599-3611. doi: 10.22075/ijnaa.2022