RepoRadar – Agentic AI Framework for Fake Project Detection

RepoRadar leverages advanced AI and a multi-agent framework to ensure the uniqueness of software projects. We'll explore how this system analyzes code, ideas, and contributions to provide comprehensive plagiarism reports, helping maintain integrity in software development.

Project Overview and Objectives

Detect Code Plagiarism

Identify similarities in code across GitHub repositories

Evaluate Idea Uniqueness

Assess the novelty of project concepts based on repository descriptions and documentation

Analyze Contributions

Examine contributor activity to detect potential copied or shared work

Generate Detailed Reports

Provide clear, actionable insights into plagiarism risks with similarity scores

System Architecture: Multi-Agent Framework

Repo Fetcher

Retrieves comprehensive repository details from GitHub API

Scorer & Report Generator

Computes plagiarism scores and compiles detailed reports

Code Parser

Extracts and normalizes code components across languages

Similarity Agents

Analyzes code and idea similarity using FAISS vector database

Contribution Analyzer

Examines commit patterns and contributor data

Technical Execution

Frontend

- React for dynamic UI
- Axios for API calls
- Bootstrap and Tailwind CSS for styling

Backend

- LLaMA-3.3-70B-Versatile model
- FAISS vector database
- Python/Flask or FastAPI

Data Processing

- GitHub API integration
- Creating vector embeddings
- Semantic similarity metrics

Tool / Project	Drawbacks
MOSS	- Closed source and usage restricted by Stanford license - Requires special registration key - No support for direct GitHub link input
JPlag	- Requires manual code upload - No direct GitHub integration - Limited to supported languages - May not catch advanced semantic similarities
Sherlock	- Not a plagiarism checker (focused on username discovery) - Mentioned here only for GitHub scraping inspiration
PlagScan / Turnitin	- Paid/commercial tools - Closed source - Not designed for GitHub or bulk repository checking - Focuses more on academic documents
Checkmarx / SonarQube / DeepSource	- Focused on code quality/security, not plagiarism - May flag only syntactic duplication - Requires repo integration, not link-based on-demand checking
CCFinderX / Deckard / PMD CPD	- Mostly offline tools - Require manual setup and input - Language support limited - Often focus on exact or structural duplication, not semantic
CodeBERT / GraphCodeBERT	- Requires ML expertise to implement - Not plug-and-play - Slow on large codebases - No direct GitHub repo handling built-in

Comparison with Existing Plagiarism Tools

Existing Plagiarism Tools

Widely used but with limitations:

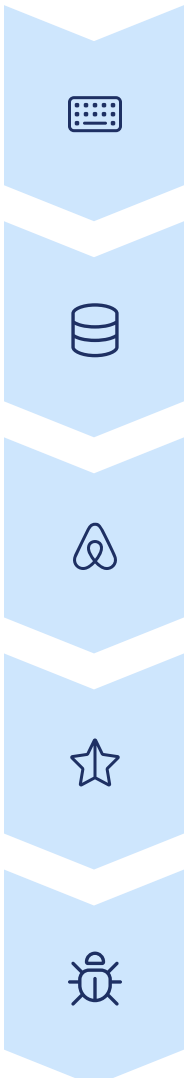
- Limited to exact match detection
- Struggle with paraphrased code
- Do not analyze idea uniqueness
- Accepts only limited files of the project

RepoRadar

Advanced features for comprehensive analysis:

- Detects similar code and ideas
- Analyzes contributions for shared work
- Generates detailed reports with similarity scores

Workflow: From User Input to Report Generation



User Input

Enter repository URL

Data Fetching

Retrieve data from GitHub API

Analysis

Process code, ideas, contributions

Scoring

Compute plagiarism score

Report Generation

Compile detailed report

Key Features and Challenges Addressed

Key Features

- Comprehensive analysis of code, ideas, and contributions
- Efficient similarity detection using FAISS
- User-friendly interface with interactive charts
- Modular design for easy updates

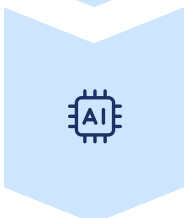
Challenges Addressed

- Scalability for large codebases
- Support for diverse programming languages
- Handling of GitHub API
- Reduction of false positives in plagiarism detection

Future Enhancements

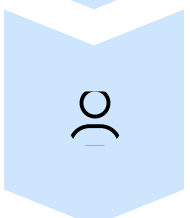
Cross-Repository Comparison

Extend analysis to compare against a broader set of GitHub repositories



Advanced NLP

Incorporate more sophisticated NLP models for deeper idea similarity analysis



Authentication Improvements

Replace hardcoded GitHub API token with OAuth for secure access

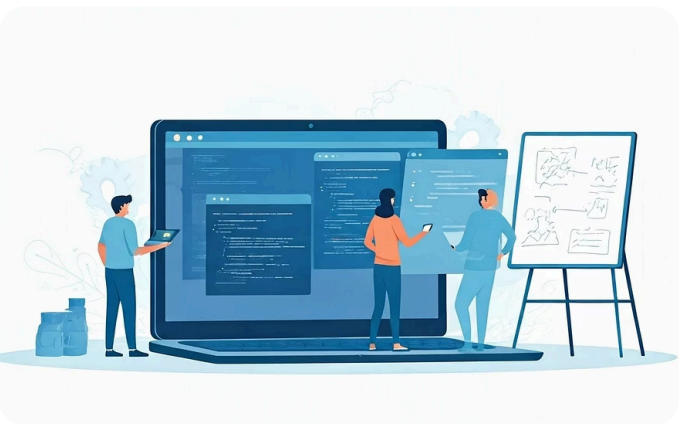
Real-Time Monitoring

Add support for continuous monitoring of repositories for new commits

Impact and Applications

Education

Helps educators ensure originality in student projects, promoting academic integrity.



Open Source Community

Enables maintainers to verify the uniqueness of contributions, fostering innovation.

Corporate Development

Assists companies in protecting their IP and ensuring compliance in software development.

Conclusion: Ensuring Code Integrity

The GitHub Repository Plagiarism Detection System significantly improves software development integrity. It combines AI analysis with a user-friendly interface to effectively detect code and idea similarities.

This tool helps identify plagiarism, promotes originality, and ensures proper attribution. Future enhancements will further streamline code integrity maintenance in the development process.