

Classgenuity

Imagine a tool that feels like a trust assistant for teachers, helping them ensure students turn in original work without any stress. It is designed to be straightforward, practical, and focused on catching similarities in student assignments, all while being easy to use.

I will explain how it all comes together from the moment a teacher sets up an assignment to the detailed report they get at the end. Let's walk through how it all comes together, from the moment a teacher sets up an assignment to the detailed report they get at the end.

At its core, ClassGenuity is a web app built with Django, a Python framework that keeps things organized and secure. Teachers start by logging in and creating an assignment, what we called a "node". They give it a name, like "History Essay," or "UDUS competition IT essay" set a deadline (say, August 10th), and specify that students or participants should submit their PDFs. This gets stored in a PostgreSQL database, where we keep track of assignments, users, and submissions. Students, also logged in, see the assignment on their dashboard and upload their PDF files. These files are saved securely on the server, and now the magic begins.

Once a student uploads a PDF, ClassGenuity uses a Python library called `pdfminer.six` to extract the text from the file. PDFs can be tricky some have weird formatting or even images but this library does a solid job of pulling out the words. The extracted text is saved in the database alongside the file, so we don't have to reprocess it later. If a PDF is unreadable (like if it's all images), the system flags it for the teacher, so they know to check it manually.

When the deadline hits, the real action starts. ClassGenuity kicks off a process to compare all the submissions for that assignment. Without the use of AI we are using (for resource management) old-fashioned text similarity techniques, specifically TF-IDF and cosine similarity.

Here's how it works: we clean up the extracted text first, stripping out common words like "the" or "and" (using NLTK's stopword list), removing punctuation, and converting everything to lowercase. This ensures we're comparing meaningful content, not formatting words.

Next, we turn each submission's text into a numerical vector using TF-IDF (Term Frequency-Inverse Document Frequency). Think of it like giving each word a score based on how unique it is across all submissions. Common words get low scores, while distinctive ones get higher. The scikit-learn library handles this, creating a matrix where each row is a submission. Then, we use cosine similarity to compare these vectors. It's like measuring the angle between two lines—closer to 1 means the texts are nearly identical, closer to 0 means they're different. This gives us a similarity matrix, showing how every submission compares to every other.

From there, ClassGenuity groups similar submissions. If two or more files have a similarity score above a threshold (say, 80%), they're put in the same group, suggesting possible copying. We also calculate a %genuity score for each submission: it's $(1 - \text{highest similarity score with another file}) * 100$. So, if a student's essay is 90% similar to another, their %genuity is 10%, signaling it's not very original. Teachers can be able tweak the threshold to avoid flagging minor overlaps, like common phrases.

The final report is where it all comes together. Teachers get a clean dashboard showing groups of similar submissions like "Group 1: Sarah and Tom's essays" along with each student's %genuity score. They can click into a detailed view to see side-by-side comparisons with highlighted matching text, thanks to Python's `diff` library. Additional metrics, like the number of similar passages or

total overlapping words, give teachers more context. We use Celery to process this in the background, so the app stays snappy even with lots of submissions.

The system is built to be intuitive. Teachers don't need to be tech wizards to use it, and students just upload and go. It's secure, private, and focused only on comparing files within the assignment, making it fast and relevant. That's ClassGenuity helping teachers uphold honesty while keeping things simple and fair.

This project to an extent has been described as for school environment only, but its functionality can be seen in many other fields.

Thank you Sir/Ma