# Intelligent test interface with plagiarism detection

Team Infernos

Jay Bansal  Harmeen Kaur  Sriniwas Paliwal

Indian Institute of Technology, Bombay

October 15, 2019

# Overview

# About project

- An intelligent test interface which can automatically check submitted responses for plagiarism
- Detects percentage plagiarism for fair relative judgement of copying
- 24x7 Server Management, always round-the-clock
- A nice subjective answer matching algorithm will be used to accomplish the task
- Choice based questions will also be given weightage accordingly to get even more accurate plagiarism results
- promotes healthy, honest and plagiarism-free environment

# Target Audience & Roles

We would be creating an test interface in which following roles can do the following activities:

## Instructor

- Register/Login.
- Create Courses.
- Interface to create a test with multiple type of questions.
- Add/Remove students to courses.
- Start or Stop the test.
- Use Auto-analyzer for plagiarism checker and score calculator.
- See the results of plagiarism detection algorithm, set threshold acceptable plagiarism index for the test.
- Notify the suspects to justify themselves if needed :p

## Student

- Request to join a course / Withdraw from a course.
- Attempt a test and submit it.
- See results of tests and plagiarism results if found suspect.

## Admin

Manages the database, users, and back-end completely.

We would be creating separate interfaces with these features to provide desired capabilities to corresponding roles.
Our main motive is to create an accurate plagiarism detecting algorithm for various types of tests.

# Plagiarism Detection

**Create an intelligent 'index'**

1. Goal is to come up with a smart plagiarism 'index' between any two sets of solutions.
2. Take into account various types of questions with weightage according to their types.
3. Instructor sets a threshold acceptable index value for a particular test according to analysis results.

**How does it work?**

For subjective questions:- Use LCS[1], Shingle (n-gram)[2] based algorithms. Also study algorithms which currently existing plagiarism detection softwares such as MOSS[3] use to improve measure.
For MC-based questions:- Set the index for these questions proportional to no. of distinct possible solutions to that problem.
Finally, assign a weight to each question, and find overall 'index' for the test

# Technologies

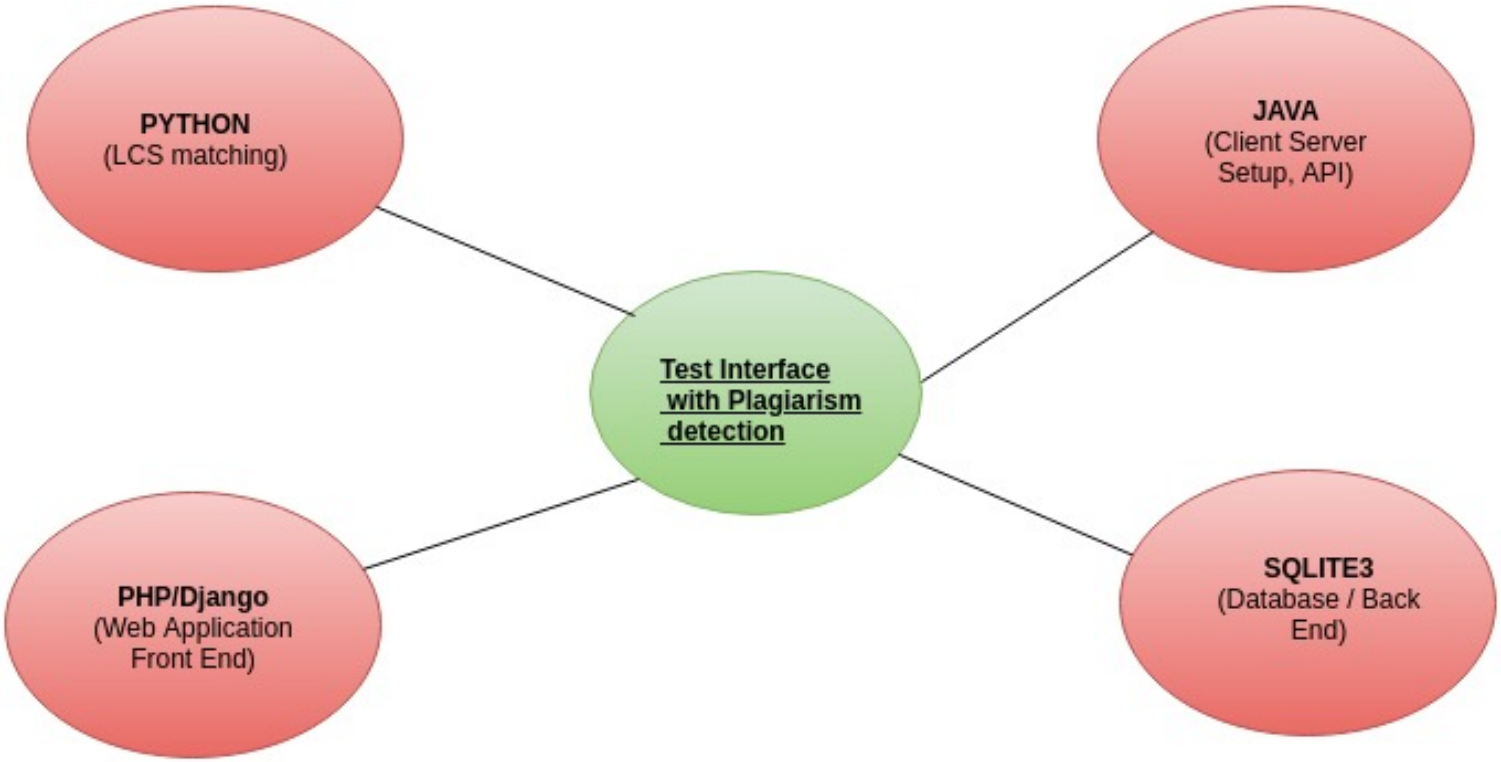


Figure: Tools and Technologies to be used

# References


Mahalakshmi S. et al.(2016)

Plagiarism Detection Using Longest Common Subsequence Method

*International Journal of Computer Techniques* 3(4).

Efstathios Stamatatos(2011)

Plagiarism Detection Using Stopword n-grams

*Journal of the American Society for Information Science and Technology* 62(12), pp. 2512-2527, Wiley, 2011.

Saul Schleimer et al.(2003)

Winnowing: Local Algorithms for Document Fingerprinting

https://en.wikipedia.org/wiki/Plagiarism_detection

# The End