CS5500 TEAM-6 : PLAGIARISM DETECTOR

INTRODUCTION

The aim of this project is to design and implement a plagiarism detection software as a web application so as to identify plagiarized solutions submitted by students. A solution is said to be plagiarized when it is derivable from another solution, merely by renaming variables, changing the order of functions in the code, converting a sequence of statements into methods and so on. The application aims to assist the teaching staff to identify such plagiarised solutions and inhibit plagiarism amongst students.

PROJECT DESCRIPTION

The application will be designed to run the plagiarism check against source files written in the **Python** (version 2) programming language. The source files for student submissions will be uploaded to the system by the graders or professors. When run, our application will start a plagiarism check for every submission against every other submission. Since plagiarism is a relative concept, we enforce configurable ‘match thresholds’. If the similarity of two submissions crosses this threshold, we mark them as plagiarised and store their details in a report. Each run will generate a report that will aggregate these similarities in the form of statistics for all submissions. The teaching staff can view this report.

PROJECT DESIGN

The project comprises of three main components or layers: the front-end, the services/back-end layer and the database. The front-end layer is what is visible to the user and what the user interacts with. The goal of this layer is to facilitate easy, meaningful and smooth user interaction with the system. The front-end would be implemented in React. The crux of the program lies in its back-end, implemented solely in Java as a Java web service. The front-end interacts with the back-end using REST services (APIs). All the logic for comparing submissions and detecting plagiarism resides in the back-end. The database layer provides a mechanism to store and retrieve persistent data. This would be necessary for storing user objects, source files, generated reports, etc. The database is planned to be implemented using PostgreSQL. This would be our relational database on which all queries would work on. We plan to store the submission source files inside a directory in the server.

DEVELOPMENT MODEL

The entire development process will implement principles of the **Agile** model as it has an effective response to change and thus reduces the cost of operation. We would be developing the software in four sprints, each sprint being two weeks long:
Sprint 1: *Project design, system architecture, database schema and project setup.*
Sprint 2: *User registration, authentication and navigation; course and assignment creation.*
Sprint 3: *Uploading submission source files; Develop and implement plagiarism algorithm.*
Sprint 4: *Generate meaningful reports off algorithm results; finishing touches; buffer week.*
Our team will have two scrum meets every week to discuss what we did, what obstacles we faced and what we plan to do in the coming days. After every sprint, there will be a sprint review to discuss the ups and downs of the sprint and also, plan for the sprint ahead. Every alternate week (on weeks when a sprint review is not there), a team meeting will be held to discuss how happy a team member is with his/her role and to peer evaluate each others’ performances. The development team would employ the “test as you build” philosophy throughout the project to maintain continuous code quality.