PLAGIARISM DETECTOR

Introduction:

The Plagiarism Detection System is being built for Instructors and Teaching Assistants to check for similar submission of assignments by students and also to find students who try of take someone else's work or ideas and submit it as their own. The system aims to detect plagiarism on submissions made in **Python** Language.

Project/System Description:

The Plagiarism Detection System will allow Instructors to clone the git repository into the system memory and then select the assignments of students on which he wishes to run the plagiarism detector on. The plagiarism detection system aims at detecting the plagiarism on Python source code and test files, for transformations like renaming of variables, extracting code into functions, moving code around and handling the same code being distributed across various files. There will be a comparison of submissions of all the students in the github repository and the system will find the matching pairs. Instructors can view the complete reports of matching pairs where he can see the both the codes side-by-side with matching parts being highlighted. Instructors can further generate a pdf file of reports and either download a copy or share via email. There are two types of users for the system, Instructors/TAs who will run the plagiarism detection system and admin who will be responsible for granting access to the Instructors and handle logistical issues.

System Design Plan:

The system will contain a backend which will hold data that contains all the past reports generated using the plagiarism detector and also the user profile details. Implementation of backend will be done using "PostgreSQL" which is an object-relational database management system. The data will be exposed to our front-end using REST API services in JSON format. The REST services will be written in Java. The front-end will be a Web app built using React

Development Model/Roadmap:

The **Agile** software development approach will be followed to build the system. The software is planned to be developed at the end of 8 weeks(approximately) and sprint meetings are planned to be conducted every 2 weeks to share the progress on the work done in the past week, the work intended to be done during the next week and also if there is any need to incorporate any changes in the requirements.
Roadmap for the sprint meetings are:
SPRINT 1: Base architecture and database setup. User Registration and authentication module
SPRINT 2: Cloning student repositories on entering Github link. Project report overview page
SPRINT 3: Implementing the plagiarism detection algorithm.
SPRINT 4: Generate,store and delete reports. Share reports as pdf on mail or download a copy.
Different branches for each functionality will be forked and each week there will be a merge to a development branch where in all the merge conflicts can be resolved and integration testing can be performed. Once a major functionality or set of functionalities are complete, this branch can be pushed to the master branch upon completion of regression testing. Scrum meetings will be planned every alternate day to discuss the roadblocks that anyone is facing and collaborate on to find a solution. The **TDD**(Test Driven Development) approach will be followed to build the system.