Basic project
Plagiarism checker python

```
# Function to check for plagiarism
def find_plagiarism():
    # Initialize an empty set to store plagiarism results
    plagiarism_results = set()

    # Access the global variable doc_filename_pairs
    global doc_filename_pairs

    # Iterate through each student's file and vector
    for student_a_file, student_a_vec in doc_filename_pairs:
        # Create a copy of the document-filename pairs for iteration
        remaining_pairs = doc_filename_pairs.copy()

        # Find the index of the current document-filename pair
        current_index = remaining_pairs.index((student_a_file, student_a_vec))

        # Remove the current pair from the remaining pairs
        del remaining_pairs[current_index]

        # Iterate through the remaining pairs to compare with other students
        for student_b_file, student_b_vec in remaining_pairs:
            # Calculate the cosine similarity between student_a_vec and student_b_vec
```

```
            similarity_score = calc_cosine_similarity(
                student_a_vec, student_b_vec)[0][1]

            # Sort the filenames to maintain consistency in results
            sorted_filenames = sorted((student_a_file, student_b_file))

            # Create a plagiarism result tuple with sorted filenames and similarity score
            plagiarism_result = (
                sorted_filenames[0], sorted_filenames[1], similarity_score)

            # Add the result to the plagiarism_results set
            plagiarism_results.add(plagiarism_result)

    # Return the set of plagiarism results
    return plagiarism_results


# Print plagiarism results
plagiarism_results = find_plagiarism()
for result in plagiarism_results:
    print(result)
```

Result
'fatma.txt', 'juma.txt', 0.22010931810615814)
('john.txt', 'juma.txt', 0.9999999999999998)
('fatma.txt', 'john.txt', 0.22010931810615814)

Presented by surekha