**Semantic Similarity Analysis of Textual Data** aims to measure the semantic similarity between textual data using advanced Natural Language Processing techniques. By leveraging pre-trained language models like BERT, the system computes similarity scores between sentences or documents, enabling applications in text clustering, plagiarism detection, and more."

. Introduction

* Background: Why is semantic similarity important? Mention real-world applications like chatbots, search engines, and plagiarism checkers.
* Problem Statement: Describe the issue your project addresses (e.g., "How to efficiently compute and interpret semantic similarity between text?").
* Objective: Define the project’s goals.
* Scope: Explain what is covered and what is out of scope.

System Design

* Architecture Diagram: Include a flowchart or block diagram of your system.
* Modules:
* Preprocessing (e.g., tokenization, stopword removal).
* Embedding generation using models like BERT.
* Similarity computation using metrics like cosine similarity.
* Output and visualization.

Methodology

Explain the step-by-step process:

* Data Collection
* Preprocessing
* Feature Extraction (e.g., word embeddings)
* Model/Algorithm Selection
* Similarity Computation
* Evaluation Metrics (e.g., accuracy, Pearson correlation)

Implementation Include a detailed explanation of the tools and technologies used (e.g., Python, Hugging Face Transformers, Sklearn). Provide code snippets for major functions like text preprocessing, embedding generation, and similarity computation. Mention hardware/software requirements

**Compare User Inputs and Get Semantic Similarity Score**

We will create a **function that takes multiple inputs from two users**, compares their inputs, and calculates the **semantic similarity score** using OpenAI embeddings

 Add **CSV file writing function**

 Save **source input, reference input, and similarity score**