**AI Plagiarism Checker Documentation**

# 1. Introduction

Plagiarism is a growing concern in academics, publishing, and online content creation. Traditional plagiarism detection methods rely mostly on string-matching and keyword overlaps, which fail when content is paraphrased.  
  
This project implements an AI-powered plagiarism checker that uses both statistical similarity and semantic similarity to identify copied or paraphrased text. The system can work with a local corpus of documents, or external web sources (via search API integration).

# 2. Objectives

- Detect exact and near-exact overlaps.  
- Identify paraphrased content using semantic similarity.  
- Provide highlighted matches with similarity scores.  
- Generate reports (JSON/HTML) for easy interpretation.  
- Support scalability using vector indexing for large corpora.

# 3. Implementation Details

**Frontend:** React/HTML for UI

**Backend:**

Language: Python  
 Libraries:  
 - scikit-learn → TF-IDF vectorization & cosine similarity  
 - Sentence Transformers (BERT embeddings)  
 - PyPDF2, python-docx

# 4. Evaluation

|  |  |  |
| --- | --- | --- |
| Test Case | Type of Plagiarism | Detection Accuracy |
| Copy-Paste | Verbatim text | 95%+ |
| Light Editing | Changed punctuation | 90%+ |
| Paraphrasing | Rewritten sentences | 75–85% |
| Common Phrases | Generic content | May give false positives |

# 5. Challenges

- Requires large reference corpus or web search API.  
- Paraphrase detection is not perfect.  
- Cannot differentiate AI-generated original text from plagiarism.  
- Cross-lingual plagiarism detection is limited.

# 6. Conclusion

The AI Plagiarism Checker combines traditional text-matching techniques with modern semantic embeddings to detect both exact and paraphrased plagiarism. With proper integration into academic or publishing workflows, it provides a reliable tool for ensuring originality and academic integrity.