PLAGIARISM CHECKER PROPOSAL

(Final Submission)

Prepared for:
Department of Computer Science
[Your University Name]

Prepared by:
Mohsin Ali
Student – BS Computer Science
Email: your.email@example.com
Date: [Insert Date]

SUMMARY

This proposal presents a comprehensive software solution — a Plagiarism Checker capable of detecting similarities in both text documents and programming code files. The system leverages preprocessing, tokenization, and similarity algorithms to ensure fair academic evaluation and discourage dishonest practices. It offers a user-friendly interface and precise results for students, instructors, and institutions.

PROBLEM STATEMENT

Plagiarism in academic settings undermines originality and intellectual honesty. Manual checking is inefficient, time-consuming, and error-prone. Moreover, existing plagiarism detection tools often focus on either text or code, lacking a unified solution for both types. This project addresses these challenges by providing an integrated plagiarism detection system.

PROPOSED SOLUTION

The proposed solution is a dual-mode plagiarism checker that allows users to choose between Text File Plagiarism and Code Plagiarism detection.

1. Text File Plagiarism Workflow
• Menu: Choose Text File mode
• Input: Number and names of text files
• Preprocessing: Convert all text to lowercase, remove punctuation and extra spaces
• Tokenization & N-gram: Split text into words and create 3-word sequences (3-grams)
• Comparison: Similarity % = (Matched Tokens / Total Tokens) × 100
• Output: Display similarity percentage for all file pairs

2. Code Plagiarism Workflow
• Menu: Choose Code mode
• Input: Number and names of code files
• Preprocessing / Normalization: Remove comments, replace variable names and numbers with placeholders, remove extra spaces
• Tokenization: Break code into keywords, operators, and placeholders
• Comparison: Use cosine similarity based on token frequency vectors
• Output: Display similarity percentage for code file pairs

MARKET OPPORTUNITY

Educational institutions and software companies require reliable plagiarism detection systems. An in-house, open-source, or lightweight alternative to paid tools can benefit students and teachers by providing accurate, cost-effective, and transparent results. The system can also be integrated into online learning platforms for automated checks.

EXPECTED OUTCOME

- Detect plagiarism across both text and code efficiently.
- Provide clear similarity metrics and reports.
- Improve academic integrity through easy verification tools.
- Reduce manual checking workload for teachers.

TECHNOLOGIES USED

- Programming Language: Python
- Libraries: re, sklearn, collections, nltk
- Algorithms: Cosine Similarity, N-gram Tokenization
- Interface: CLI / Simple GUI (optional enhancement)

FINANCIAL SUMMARY / RESOURCES

Resource	Description	Estimated Cost (PKR)
Laptop / PC	Development and testing	Existing
Python Libraries	Open-source	Free
Data Samples	Text & code files for testing	Free
Internet	For research and testing	~1,000/month
Total Estimated Cost		~1,000 PKR/month

CONCLUSION

This plagiarism checker will serve as a robust academic tool capable of analyzing both text files and source codes with high accuracy. It will promote originality, integrity, and automation in academic environments while being easy to deploy and cost-efficient.