



Department of Computer Science & Engineering and Information Technology

Major Project Proposal (2024-25)

Group No.

86

1. **Project Title:** (mention your project title which clearly defines the purpose and scope of your work. Capitalize first and major words of your project title.)

Identification of redundant code using AI

2. **Team Members**

S. No.	Roll No.	Name	Mobile No.	Proficiency
1.	211397	Piyush Joshi	9015368062	Data Science
2.	211150	Samriti Thakur	7876179881	Information security
3.	211327	Prakhar Varshney	7017986599	Information security

3. **Name of Supervisor (s)** (mention the name and designation of your supervisor. If there is any co-supervisor, please mention the department as well.)

Mr. Ramesh Narwal

Assistant Professor (Contractual)

Ms. Seema Rani

Assistant Professor (Contractual)

4. **Work Distribution** (clearly state the distribution of work among team members.)

S. No.	Roll No.	Work Distribution
1.	211397	<ul style="list-style-type: none">Analyzing project statement and research on dataset development.Study various types of code clones and build dataset.Testing and validation of model.Documentation and maintaining code.



2.	211150	<ul style="list-style-type: none">• Collect open-source code repositories.• Working on preprocessing of the dataset.• Research and select appropriate AI models.• Documentation and maintaining code.
3.	211327	<ul style="list-style-type: none">• Literature review and analysis.• Train the model on a labeled dataset to recognize redundant code patterns.• Testing and validation of model.• Documentation and maintaining code.

5. Problem Statement (*serves as a basis of your project and should comprise of max. 500 words spread over at least two paragraphs*)

In modern software development, code redundancy is a significant issue that affects code maintainability, readability and efficiency. Redundant code patterns, often manifested as code clones, are repeated segments of code that perform similar or identical functions across different parts of a codebase. These patterns can arise due to various reasons, such as copy-pasting code, lack of awareness of existing functions, or divergent development practices across teams. While small codebases might manage redundancy without significant issues, large-scale software systems suffer from increased maintenance costs and higher bug proliferation risks. Traditional methods for detecting and managing redundant code, such as manual code reviews or basic static analysis tools, are often insufficient, particularly in large and complex projects where the sheer volume of code makes it nearly impossible to detect and eliminate redundancy effectively. Redundant code not only bloats the codebase but also introduces potential security vulnerabilities and inconsistencies, making the system more prone to errors and harder to debug. Furthermore, redundant code can lead to performance degradation as the system may execute unnecessary instructions, consuming more memory and processing power.

The proposed project seeks to address this challenge by leveraging artificial intelligence to develop a sophisticated Code Clone Detection System that can automatically identify and classify redundant code patterns within a codebase. By utilizing AI techniques such as machine learning and natural language processing (NLP), the system aims to go beyond traditional syntax-based methods, offering a more nuanced understanding of code similarity. This approach allows the system to detect not only exact matches but also near-miss clones, where code segments perform similar functions with slight variations.

The project will involve the collection and preprocessing of a substantial dataset of source code, including labeled examples of code clones. The AI model will be trained on this dataset to recognize different types



of redundancy, including syntactic and semantic clones. The system will be integrated into a user-friendly tool that developers can use to scan their codebases, highlight redundant code patterns, and suggest potential refactoring opportunities. This project will significantly contribute to the field of software engineering by providing a scalable, automated solution for code redundancy detection. It will enhance code quality, improve maintainability, and reduce the overall cost of software development. The successful implementation of this system will empower developers to maintain cleaner, more efficient codebases, ultimately leading to more robust and reliable software products. By integrating this tool into the development lifecycle, organizations can ensure that their software remains maintainable and scalable, even as it grows in complexity.

6. Main Objectives *(mention at least three objectives)*

- 1) **Develop an AI-Driven Model for Code Clone Detection:** To create and train an artificial intelligence model capable of accurately identifying and classifying redundant code clones within a source codebase.
- 2) **Create a User-Friendly Tool for Code Clone Detection:** To develop an intuitive and effective tool that integrates the AI model and allows users to easily scan their codebases for redundant code patterns.
- 3) **Enhance Code Quality and Maintainability:** To improve the overall quality and maintainability of software by effectively identifying and managing code redundancy.

7. Resources Required *(mention software, hardware, and other resources)*

Category	Description	
Software Resources	<ul style="list-style-type: none">• Python• TensorFlow, PyTorch, Scikit-learn• NLTK, SpaCy• Jupyter Notebook, Google Collab	<p>Version: 3.8 or higher</p> <p>Version: 0.24, 3.5, 2.4 or higher</p> <p>Version: 3.0, 4.0 or higher</p> <p>Version: Python 3.8</p>
Hardware Resources	<ul style="list-style-type: none">• Processor: intel i5 or higher• RAM: 16GB or Higher• 256 GB SSD or Higher	
Others	<ul style="list-style-type: none">• Datasets• Development Environment (Visual Studio Code)• GPU (RTX 2070 or higher)	



8. **Project Plan** (please update the provided Gantt Chart according to your project work plan, breaking down the proposed work into phases and tasks along with their timelines for the entire academic year 2024-25.)

Activity	Year 2024						Year 2025					
	Aug.	Sept.	Oct.	Nov.	Dec.		Jan.	Feb.	Mar.	Apr.	May	
Literature Review												
Analysis and Requirements												
Project Design and Architecture												
Implementation												
Testing and Validation												
Documentation and Write-up												

Signatures (please also mention the name of team members and supervisor (s) with date)

Piyush Joshi
(Name of Member 1)

Samriti Thakur
(Name of Member 2)

Prakhar Varshney
(Name of Member 3)

Mr. Ramesh Narwal
(Name of Supervisor)

Ms. Seema Rani

(Name of Co-Supervisor, if any)

Date of Submission: 21 August 2024