*ASSIGNMENT NO:-9*

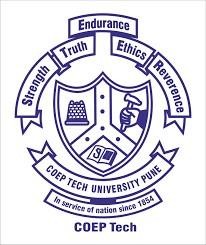
**CodeScout - Semantic Code Search & Retrieval**

*submitted in fulfillment of the*

*requirements for completion of SE LAB of*

TY COMP *in*

Computer Engineering *by*Aditya Deshmukh MIS:612203036

*Department of Computer Engineering*

Department of Computer Engineering, COEP Technological University (COEP Tech)

(A Unitary Public University of Govt. of Maharashtra) Shivajinagar, Pune-411005,Maharashtra, INDIA

October, 2024

# CodeScout - Semantic Code Search & Retrieval

# Abstract

# CodeScout is a cutting-edge solution that empowers development teams to efficiently search and retrieve relevant code snippets from large codebases using natural language queries. Leveraging advanced NLP techniques, including transformer-based models and embeddings, CodeScout enhances developer productivity by simplifying code discovery while ensuring adherence to the company’s custom technical guidelines. It includes a CLI tool for command-line searches and a VSCode extension that seamlessly integrates with the developer’s workflow, offering an intuitive, accessible UI and promoting faster onboarding, code reuse, and compliance with internal standards.

# Table of Contents

# [Introduction](file:///D:\Semester%205\SE\Practical%208\readme.docx#introduction)

# [System Architecture](file:///D:\Semester%205\SE\Practical%208\readme.docx#system-architecture)

# [Features](file:///D:\Semester%205\SE\Practical%208\readme.docx#features)

# [Technology Stack](file:///D:\Semester%205\SE\Practical%208\readme.docx#technology-stack)

# [Installation & Setup](file:///D:\Semester%205\SE\Practical%208\readme.docx#installation--setup)

# [How to Use](file:///D:\Semester%205\SE\Practical%208\readme.docx#how-to-use)

# [Business Value](file:///D:\Semester%205\SE\Practical%208\readme.docx#business-value)

# [Future Enhancements](file:///D:\Semester%205\SE\Practical%208\readme.docx#future-enhancements)

# [Contributing](file:///D:\Semester%205\SE\Practical%208\readme.docx#contributing)

# [License](file:///D:\Semester%205\SE\Practical%208\readme.docx#license)x

# Introduction

# CodeScout addresses a major challenge faced by large organizations: finding and reusing existing code that aligns with custom technical practices. As repositories grow, developers often struggle to locate the right pieces of code efficiently. CodeScout solves this by enabling semantic search of function definitions and code snippets across multiple repositories, using advanced NLP models to return the most relevant results based on natural language queries.

# With its robust CLI tool and a VSCode extension, CodeScout integrates directly into the development environment, promoting a faster, more intuitive workflow. It supports multiple programming languages and leverages embeddings to ensure accurate, contextually appropriate search results.

# System Architecture

# Repository Analysis: CodeScout processes multiple repositories, extracting relevant source files to focus on key areas that need to be indexed for search.

# Code Splitting: The code is divided into atomic units, primarily functions, to ensure that search queries return meaningful, concise results.

# Embedding Generation: These atomic parts are passed through Google’s Gemma2b LLM model, which generates detailed descriptions of each code segment. The descriptions are converted into embeddings—dense numerical vectors that capture the semantics of the code.

# Vector Database (ChromaDB): The embeddings are stored in ChromaDB, a vector database optimized for fast, scalable retrieval. It allows CodeScout to quickly compare the query embeddings with the stored code embeddings and return the most relevant code snippets.

# Search & Retrieval: When a developer makes a natural language query, it is also converted into embeddings, and the system uses cosine similarity to find the closest match in the database.

# Features

# Semantic Code Search: Retrieve relevant code snippets from large repositories using natural language queries.

# Multi-Language Support: Supports multiple programming languages, including C++, Python, Java, and more.

# CLI Tool: A command-line interface for quick code searches and embeddings management.

# VSCode Extension: Seamlessly integrated into VSCode for in-editor searching and retrieval.

# Chroma xDB Integration: Leverages a vector database to ensure fast, accurate searches.

# Customizable Models: Ability to experiment with different sentence transformer models.

# Well-documented and Extensible: Built with maintainability in mind, making future enhancements and adjustments easy to implement.

# Technology Stack

# Frontend: React, TypeScript

# Backend: Flask, Python

# Machine Learning: Google Gemma2b LLM, Sentence Transformers (krlvi/sentence-t5-base-nlpl-code\_search\_net)

# Database: ChromaDB (Vector Database)

# CLI: Custom CLI tool using Python

# VSCode Extension: JavaScript, Node.js

# Installation & Setup

# Prerequisites

# Python 3.8+

# Node.js

# Git

# VSCode

# Setup Instructions

# Clone the repository:

# git clone https://github.com/username/CodeScout.git cd CodeScout

# Backend Setup:

# cd backend pip install -r requirements.txt python run.py

# Frontend Setup:

# cd frontend npm install npm start

# CLI Setup:

# cd cli pip install -r requirements.txt ./codescout --help

# VSCode Extension: Install the extension from the VSCode marketplace or manually load it by navigating to the extension directory.

# How to Use

# CLI Tool

# Embedding Generation:

# ./codescout --embed <repository-path>

# This will generate embeddings for all functions in the repository.

# Search Code Snippets:

# ./codescout --search "<natural language query>"

# This will return the most relevant code snippets from the repository.

# VSCode Extension

# Once installed, you can use the CodeScout extension directly within your VSCode editor. Open the extension, enter your natural language query, and it will search the connected repositories for relevant code snippets.

# Business Value

# CodeScout provides measurable improvements in developer productivity and onboarding efficiency. On average, developers spend 40-60% of their time searching for and understanding code. With CodeScout, this time is reduced by up to 35%, saving valuable time, effort, and resources. The tool also ensures enhanced adherence to organizational coding standards, resulting in fewer code review cycles and better code quality.

# Increased Efficiency: Reduced search time allows developers to focus more on building new features.

# Cost Savings: By reducing onboarding time and improving code reuse, organizations can save costs on training and development.

# Faster Development Cycles: CodeScout speeds up development cycles by ensuring easy retrieval of reusable code.

# Future Enhancements

# Support for More Programming Languages: Expand language support beyond the existing ones.

# Advanced Query Processing: Use more advanced NLP models for deeper context understanding.

# Code Refactoring Suggestions: Provide suggestions for code refactoring based on organization’s guidelines.

# Cloud Integration: Offer hosted solutions for large-scale, distributed repositories.

# Contributing

# We welcome contributions to improve CodeScout. Please follow our [Contribution Guide](file:///D:\Semester%205\SE\Practical%208\CONTRIBUTING.md) for details on how to contribute to this project.

# License

# This project is licensed under the MIT License - see the [LICENSE](file:///D:\Semester%205\SE\Practical%208\LICENSE.md) file for details.

# This README provides an overview of CodeScout and all necessary details for usage, setup, and contributions. Feel free to customize or add more details specific to your project.