National University of Computer and Emerging Sciences

Software Re-Engineering

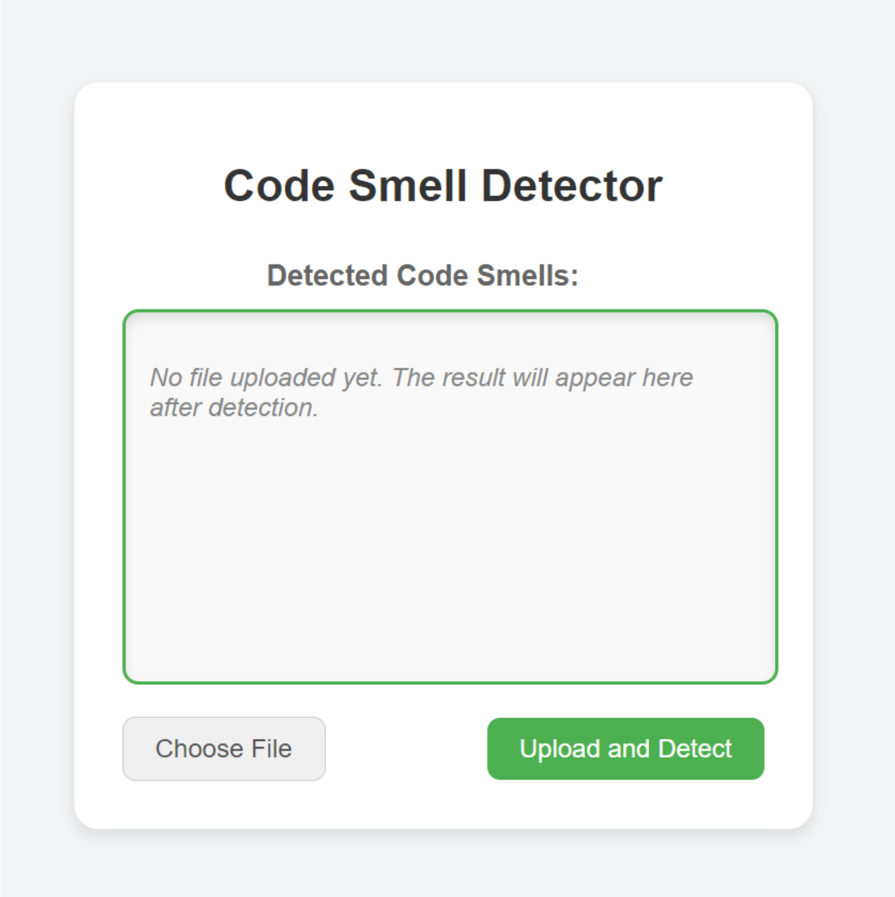
Assignment 1

Group Members
Muhammad Mujtaba 21L-6058
Junaid Khalid 21i-1142
Muhammad Sameer 21i-1165

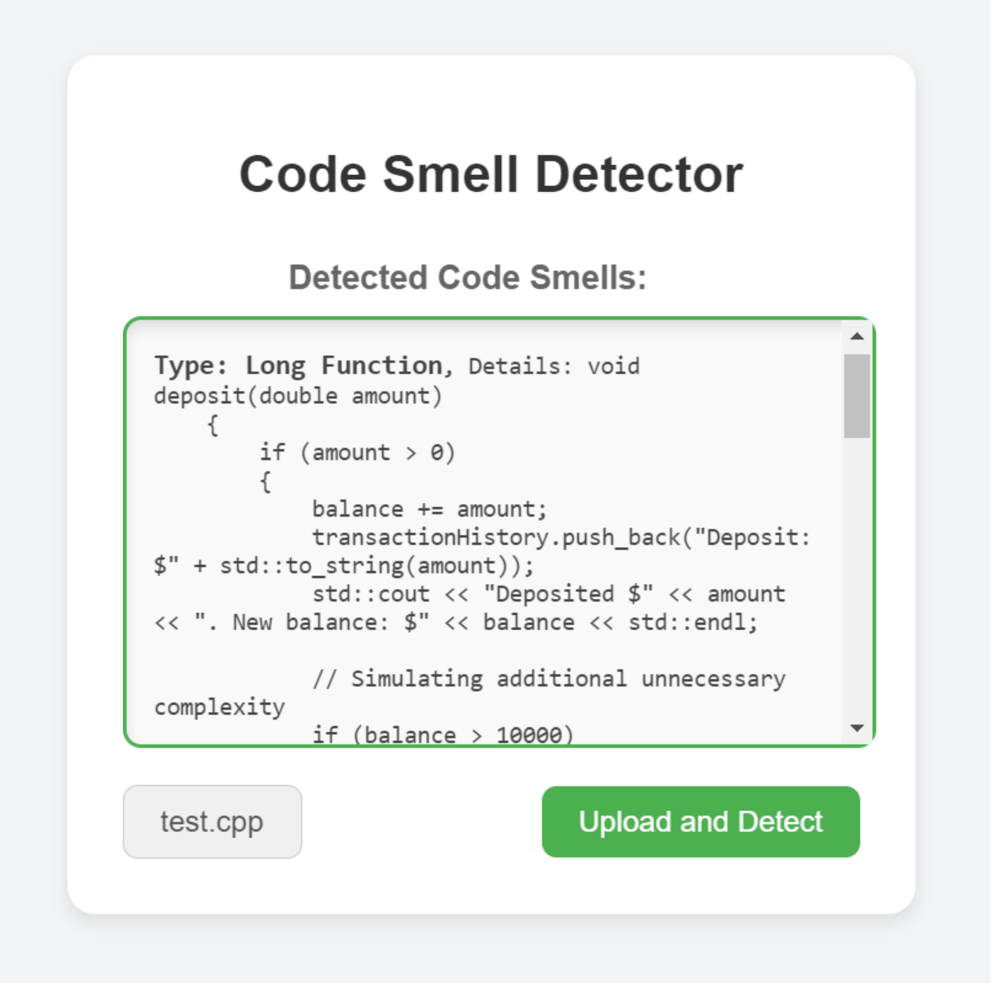
Introduction

In software engineering, writing a clean and efficient code is vital to ensure a maintainable and readable software system for future upgrades or bug fixes. Code smells are potential issues in the code which hinders the future development and maintainability of the system due to decreased readability. In this assignment, we were tasked to develop a smart code smell detector to detect code smells like Long Functions, God Classes, Duplicated Code and Large Parameters Lists. We developed this tool in MERN Stack to provide with an interface for ease of detection and usage. Our tool works for code written in C++, Java and Python.

User Interface



No file uploaded yet. The result will appear here after detection.



```
Type: Long Function, Details: void
deposit(double amount)
    {
        if (amount > 0)
        {
            balance += amount;
            transactionHistory.push_back("Deposit:
$" + std::to_string(amount));
            std::cout << "Deposited $" << amount
<< ". New balance: $" << balance << std::endl;

            // Simulating additional unnecessary
complexity
            if (balance > 10000)
```

- **Code Smell Types Detection**

  Types of code smells detected: -

  - Long Functions
  - God Classes
  - Duplicated Code
  - Large Parameters List
  - Magic Numbers
  - Complex Nested Code
  - Commented Out Code

- ## Code Smell Detection Logic

  - **Long Functions:**

    In this function, firstly all function definitions in the code are found. Then split each function into lines. If the count of line exceeds the threshold, it is marked as long function.

  - **God Classes:**

    In this function, firstly all classes in the code are found. Then, the count of methods, attributes and lines of code is calculated. If this count exceeds the threshold, class is marked as large class.

  - **Duplicated Code:**

    In this function, it splits the code into lines and occurrences of each line is recorded. If this count exceeds the threshold, it is marked as duplicated code.

  - **Large Parameters List:**

    It is like long function. All methods are recorded, and number of parameters is counted. Then comparing with threshold, it determines the code smell.

  - **Magic Numbers:**

    Finds all numeric values and is marked with magic number code smell.

  - **Complex Nested Code:**

    Iterates through the code and finds the depth of nested blocks. It increases with the nesting level by tracking the opened and closed braces. Then compared with threshold to determine code smell.

- **Commented Out Code:**
  Identifies the commented code lines and are declared with code smell.