Automating VBA Macro Documentation and Transformation

A PROJECT DOCUMENTATION

submitted by

Karthikeyan A S (715521104022)

Department Computer Science and Engineering

PSG Institute of Technology and Applied Research
Coimbatore -641 062

ABSTRACT

This work presents a tool for automating VBA Macro Documentation and Transformation built using React JS. The application enables a user to upload a macro enabled excel file. The file is then analysed and the VBA code is extracted from the file. The extracted code is analysed using the Gemini Pro Large Language Model that generates the code with comprehensive documentation and extracts the function logic that is implemented by the macro. Using the generated documentation, the process flow diagram is created. A PDF file with the documented code and the process flow diagram is generated.

Keywords: VBA Macro, React JS, Gemini Pro, Large Language Model

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

VBA – Visual Basic
LLM – Large Language Model
JS – Java Script
DB – Data Base
PDF – Portable Document Format

1. PROJECT MOTIVATION

The present approach to perform documentation is manual and time-consuming, often leading to errors and inconsistencies.

The project seeks to simplify VBA macro documentation by introducing a React Application integrated with the Gemini-Pro LLM to generate documentation for VBA macros. This approach offers a multitude of benifis. Developers can upload their macro enabled excel file to the application. The VBA macro code is extracted and is analysed by the Gemini-Pro LLM model. The model then generates a comprehensive documentation, extracts the functional logic and generates a process flow diagram. Ultimately, this innovative approach fosters significant efficiency gains by automating manual documentation tasks, minimizing errors, and streamlining the process of working with VBA code.

2. PROBLEM STATEMENT AND OBJECTIVES

The problem statement is to develop a solution that automates the documentation and Understanding of legacy VBA macros used in critical processes across DFIN, facilitating their transformation into modern technologies and IT platforms.

The problem statement involves developing a Macro Analysis Tool that

1. Utilizes Gen AI or ML techniques to automate documentation process.

2. Provides a interactive user interface that is easy to use.

3. Generates reliable and accurate documentation.

3. SYSTEM ARCHITECTURE

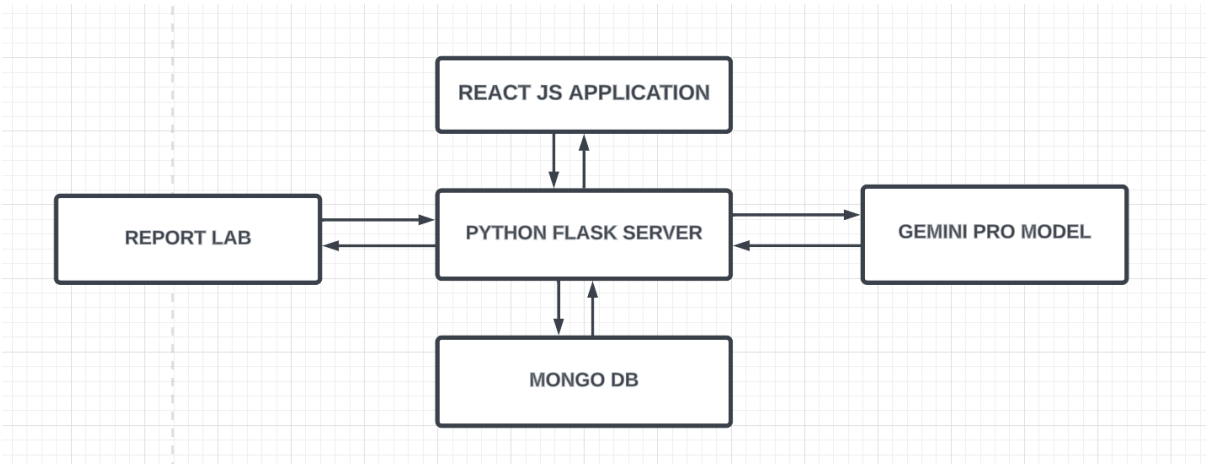


Figure 1. System Architecture

The architecture of the VBA documentation and transformation tool is shown in Figure 1. It has five main components, they are : React JS Application, Mongo DB, Gemini Pro Model and the Report Lab.

The React JS serves as the front end of the application. It is the medium through which the user uploads the excel file on which the analysis has to be performed . The analysis result is also viewed in React JS application.

The Python Flask Server is the heart of the system. It acts as a intermediate between the other components. The Flask Server receives the files uploaded by the user and extracts the VBA macro from it. It then sends the extracted code to the Gemini Pro Model to generate the documentation. The Flasks server is also responsible for generating PDF using the report lab too and store the response in the MongoDB database.

The Gemini Pro model is a pre trained LLM that was developed by google. In this project, the Gemini Pro model is used to generate the documentation for the extracted VBA code . The documentation by the Gemini AI is used to generate the process flow diagram.

The Mongo DB database is a No SQL database that is used to store the analysis results so that it can be used at a later point of time. Mongo DB can be accessed by using the PyMongo interface for python.

The Report Lab tool is used to generate the PDF report for for the documented code . The PDF document contains the documented code, extracted functional logic and the process flow diagram.

Thus the integration of all the components serves as a excellent platform to generate documentation for the legacy VBA code that simplifies understanding and reduces time taken for manual documentation

4. WORK FLOW OF THE PROJECT

- The user logs in to the application using his/her credentials.
- The user is prompted to select a Macro Enabled Excel file (.xlsm extension) to perform the analysis.
- The VBA Macro Code is extracted from the .xlsm file for analysis.
- The extracted code is given as input to the Gemini Pro LLM to generate a comprehensive documentation.
- The LLM generates a comprehensive documentation and extracts the functional logic from the VBA code.
- The documentation generated is used to construct the process flow diagram by using the schemedraw tool.
- Once the diagram is generated, a overall PDF report for the analysis is generated by using the reportlab tool and can be downloaded by the user.
- The analysis results are stored in the Mongo DB database . This is done to ensure that the results can be retrieved in the future without the need to analyse the same code again.
- The UI of this project is intuitive and can be easily understood even by new users .
- The tool generates documentation at real time speeds which makes it suitable in scenarios where a huge number of lines of code has to be documented in a short time frame.

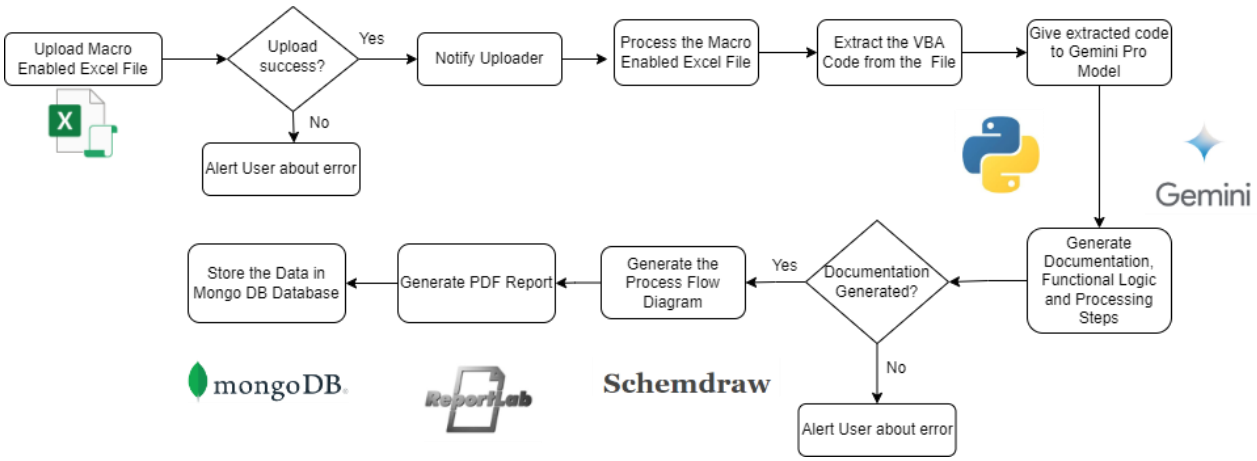


Figure 2. Workflow

5. RESULTS

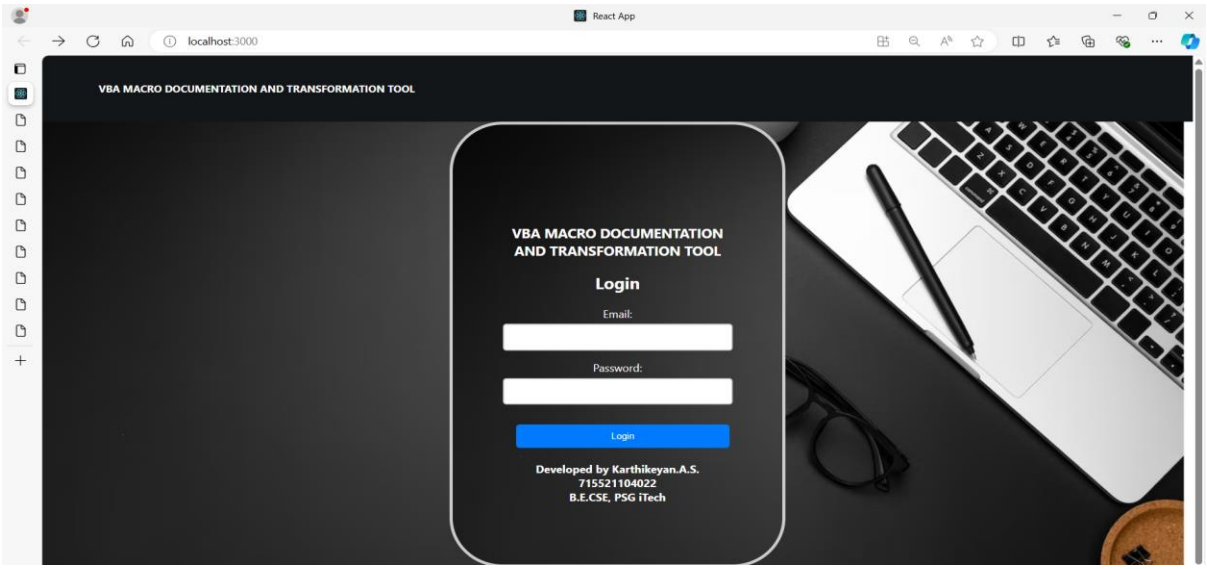


Figure 3. User Login Page

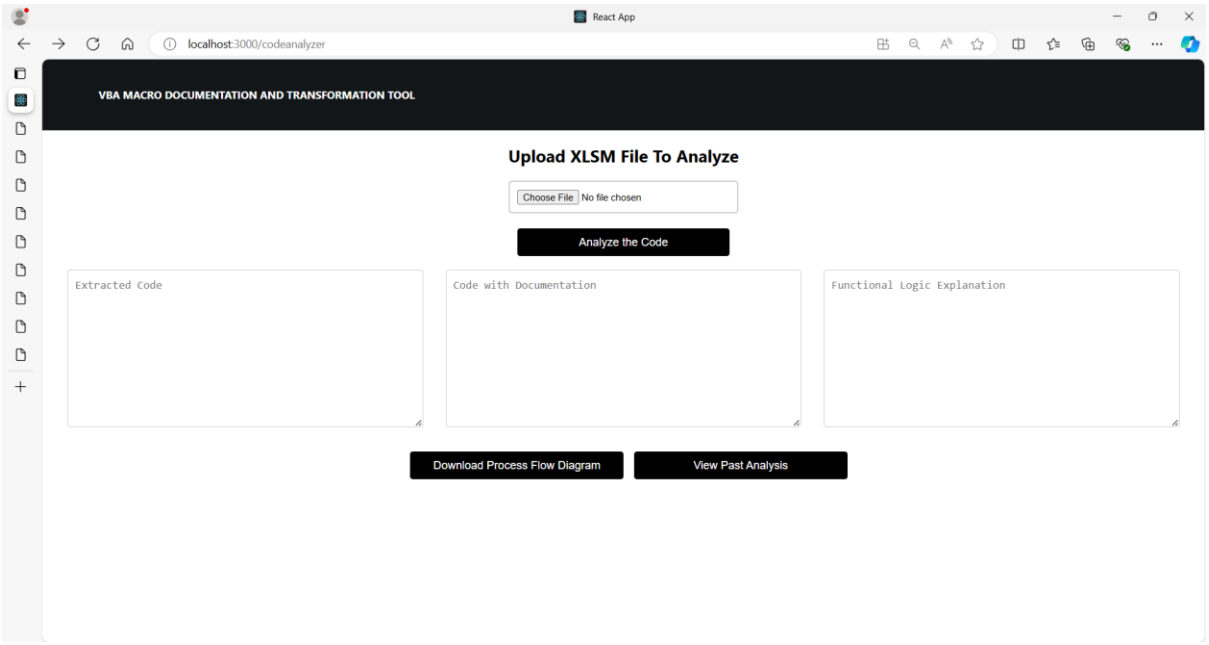


Figure 4. Interface to upload file and perform the analysis

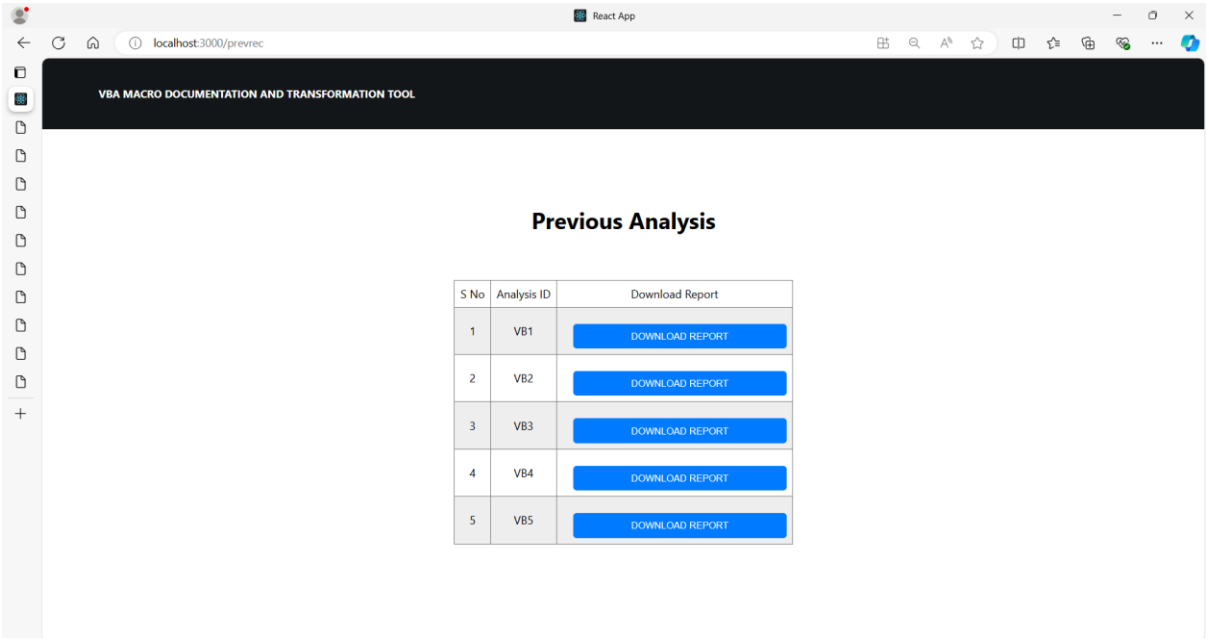
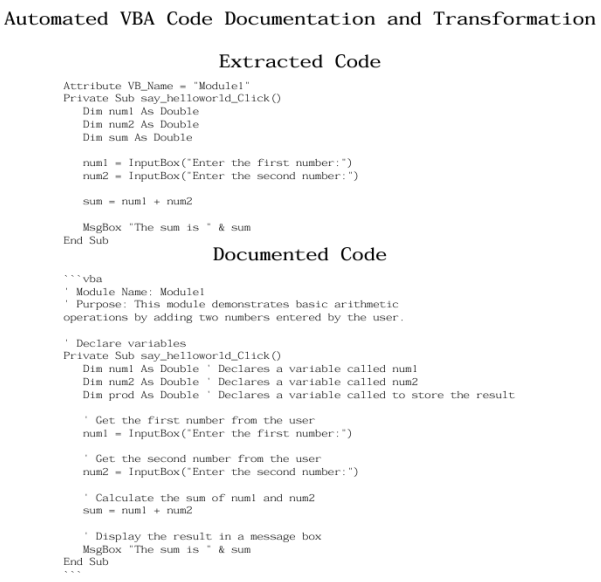


Figure 5. Interface to download the reports of previous analysis



Automated VBA Code Documentation and Transformation

Extracted Code

```
Attribute VB_Name = "Module1"
Private Sub say_helloworld_Click()
    Dim num1 As Double
    Dim num2 As Double
    Dim sum As Double

    num1 = InputBox("Enter the first number:")
    num2 = InputBox("Enter the second number:")

    sum = num1 + num2

    MsgBox "The sum is " & sum
End Sub
```

Documented Code

```vba
' Module Name: Module1
' Purpose: This module demonstrates basic arithmetic
operations by adding two numbers entered by the user.

' Declare variables
Private Sub say_helloworld_Click()
    Dim num1 As Double ' Declares a variable called num1
    Dim num2 As Double ' Declares a variable called num2
    Dim prod As Double ' Declares a variable called to store the result

    ' Get the first number from the user
    num1 = InputBox("Enter the first number:")

    ' Get the second number from the user
    num2 = InputBox("Enter the second number:")

    ' Calculate the sum of num1 and num2
    sum = num1 + num2

    ' Display the result in a message box
    MsgBox "The sum is " & sum
End Sub
```

Figure 6. PDF Report

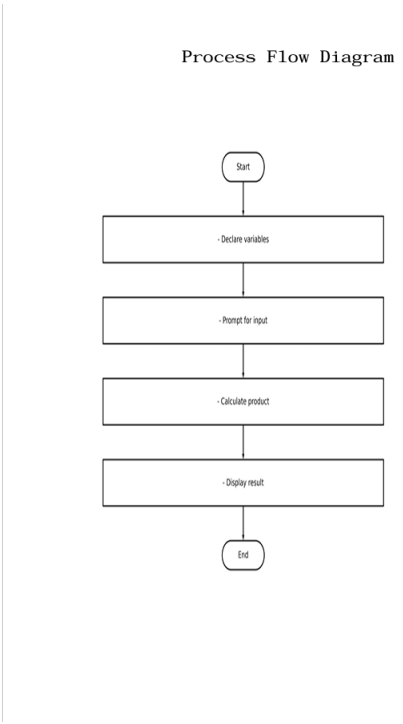


Figure 7. Process Flow Diagram

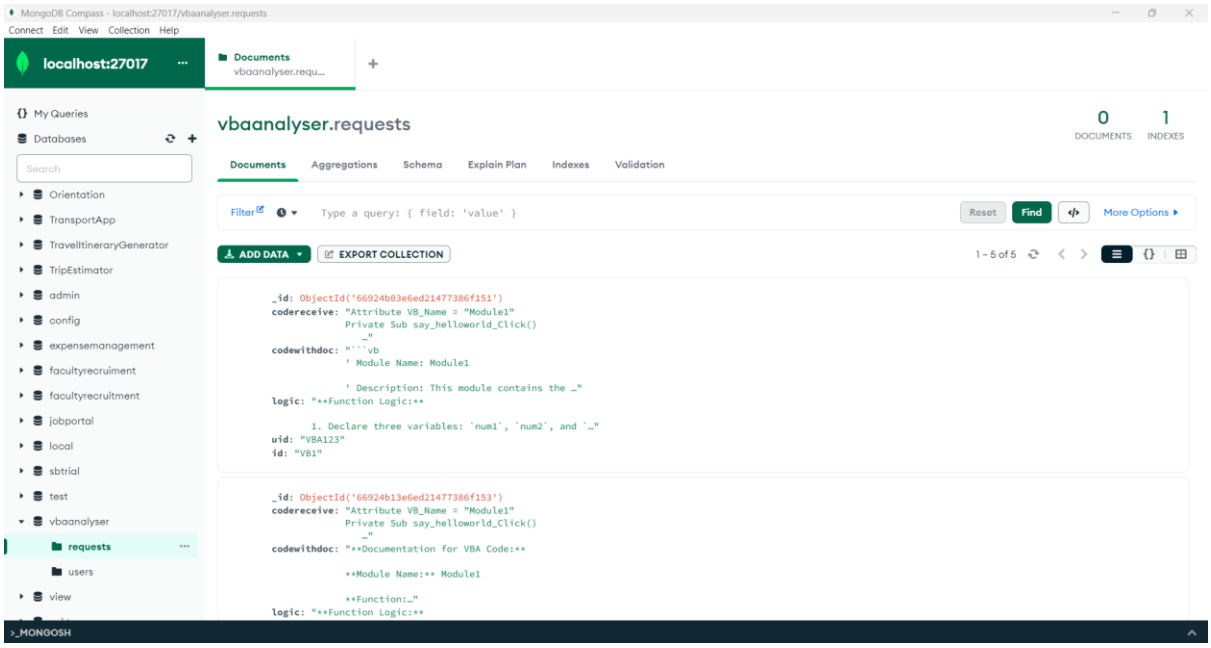


Figure 8. Mongo DB collection to store analysis results

6. REFERENCES

https://www.tutorialspoint.com/vba/index.htm

https://ai.google.dev/aistudio

https://react.dev

https://docs.reportlab.com/

https://schemdraw.readthedocs.io/en/stable/

https://flask.palletsprojects.com/en/3.0.x/