“DocuCore: Text & Code Summariser in Cloud”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

SKIT

असतो मा सद्गमय

Project Mentor:
Mr. Praveen Kumar Yadav
Assistant Professor

Submitted By :
Garvita Sakhrani(21ESKIT049)
Hiteshi Agrawal(21ESKIT056)
Abhishek Gupta(21ESKIT003)

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2024-2025

1. Introduction
2. Problem Statement
3. Literature Review
4. Proposed Solution
5. Architecture Diagram
6. Implementation Details
7. Features and Functionality
8. Testing and Evaluation
9. Result and Analysis
10. Future Work
11. Conclusion
12. References

Introduction

- This project is designed to automate the analysis of documents and source code using Databricks models.
- **Goals:**
  - Simplify code quality checks.
  - Summarize lengthy PDFs using DBRX Instruct.
  - Automate deployment using CI/CD pipelines.
- **Relevance:**
  - Reduces manual analysis time.
  - Encourages best practices in modern software development.

Problem Statement

- Manual analysis is time-consuming and prone to errors.
- Lack of scalable solutions leads to delays in development cycles.
- Need for an integrated, scalable, and automated system for code and document analysis.

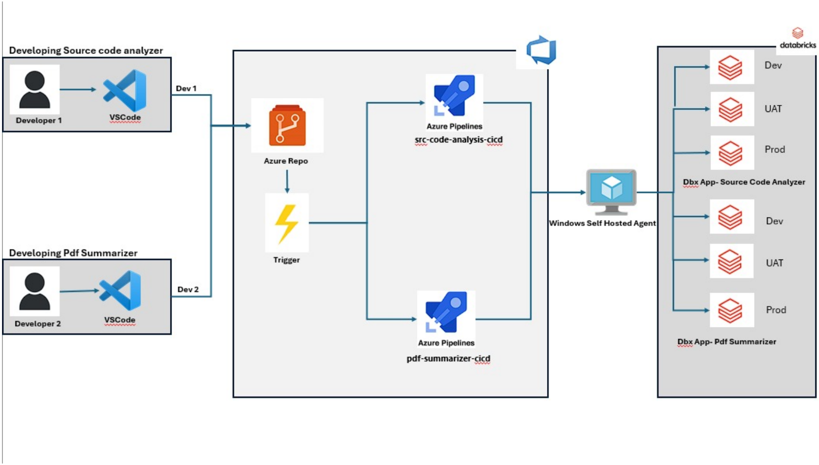
Literature Review

- **Existing Approaches:** Tools like SonarQube and ESLint help in code quality checks. LLMs are used for text summarization.
- **Gaps Identified:** Lack of integration between document summarization, code analysis, and cloud automation pipelines.
- Our project bridges this gap by automating both document and code analysis using Azure's scalable infrastructure.

Proposed Solution

- A CI/CD-based automated workflow that:
  - Analyzes source code provided.
  - Summarizes documents (PDFs) using LLM models in Databricks.
- Key Technologies:
  - Azure DevOps Pipelines
  - Databricks Apps
  - Microsoft Self-Hosted Agent
  - LLM-powered Summarizer
  - Source Code Analyzer Integration

Architecture Diagram



Implementation Details

- **Azure DevOps:** Used for setting up CI/CD pipelines and managing code repositories.
- **Databricks & Databricks Apps:** Deployed LLM-based summarization and code analysis tasks in a scalable notebook environment.
- **Microsoft Self-Hosted Agent:** Custom agent runs locally to build and deploy artifacts on specific environments securely.
- **Source Code Analyser:** Used for static code analysis integrated directly into the CI/CD pipeline.
- **PDF Summarizer:** Accepts long-form technical documents and generates intelligent summaries using GPT models.
- **Integration:** All components are deployed and orchestrated via Azure ensuring seamless execution across modules.

Features and Functionality

- Cloud-based deployment ensures high availability and scalability.
- Integrated CI/CD pipelines enable multi-environment deployment.
- Secure access with role-based control and data encryption.

Testing and Evaluation

- **Testing:**
  - **Unit Testing:** Individual module testing (code parser, summarizer).
  - **Integration Testing:** Checked Azure pipeline, Databricks execution, and output delivery.
- **Evaluation:**
  - Output summaries were concise and accurate.
  - Code analysis matched expected static metrics.
  - Pipelines performed consistently under load.

Result and Analysis

- Reduced Deployment Time.
- Automated Multi-Environment Deployment.
- Scalable and Efficient Architecture.
- Team Integration & Collaboration

Future Work

- Implementation of Rollback Deployment.
- Support additional code languages and document formats.
- Implement best practices for Network Security.

Conclusion

- Built a scalable system for automatic code and document analysis.
- Integrated AI and DevOps tools for practical, real-world application.
- Delivered a proof-of-concept that can evolve into a fully adaptive developer assistant.

References

◀ 1.Microsoft Azure DevOps Documentation

◀ 2.Databricks Documentation