Faculty of Informatics
Engineering Department of
Software Engineering &
Information System

كلية هندسة المعلوماتية
قسم هندسة البرمجيات و نظم
المعلومات

AutoTest & DocGen Manager – Intelligent System for Automated Testing and Documentation

Prepared by

Areej nooraldeen **Kamar aldiab** **Ola najibeh** **Wiaam alouni**

Supervised by

Dr. Riad Sonbol **Eng.Raghad Alhossny**

2025-2026

Project Overview: AutoTest & DocGen Manager

- **Project description:**

  The AutoTest & DocGen Manager is an intelligent web-based system designed to help software developers manage projects and analyze source code in an automated and structured way.

The goal of Project:

The system allows users to sign up and sign in, create and manage projects, and analyze source code by pasting code or uploading code files with the ability to select the programming language. Through static code analysis, the system extracts the code structure, generates UML class diagrams, and provides high-level and low-level logical explanations of the code. All generated documentation, including diagrams and explanations, can be exported in PDF or Markdown formats. Additionally, the system includes a notification mechanism to inform users about important project and analysis events. The main objective of the project is to simplify code understanding, automate technical documentation, and support developers in organizing and documenting their software projects efficiently.

System Architecture:

The AutoTest & DocGen Manager adopts a microservices architecture to ensure scalability, maintainability, and independent deployment of core functionalities. The system is decomposed into three distinct, loosely-coupled services:

1. User & Project Management Service: Handles all user authentication, authorization, profile management, and CRUD operations for software projects.

2. Core AI Analysis Service: Dedicated to processing source code. It performs static analysis, generates UML class diagrams, and produces high-level and low-level logical explanations using advanced AI models.

3. Notification Service: Manages the asynchronous generation and delivery of real-time alerts and updates regarding project status and analysis completion.

This architectural separation allows each service to be developed, scaled, and updated independently, optimizing resource utilization and system resilience.

Technology Stack

A modern and robust technology stack has been selected to implement the system's requirements effectively:

1)Frontend (Presentation Layer):

· **HTML5 & CSS3:** For creating the foundational structure and styling of the user interface, ensuring semantic markup and responsive design.

· **JavaScript (React):** The primary frontend library is used to build a dynamic, component-based single-page application (SPA). React facilitates the creation of a seamless and interactive user experience for managing projects and viewing analysis results

2)Backend (Application & Business Logic):

· **Django (Python):** Serves as the core backend framework for developing the microservices. Django REST Framework is utilized to build robust, secure, and well-structured RESTful APIs that facilitate communication between the frontend and the individual microservices.

3)Artificial Intelligence Integration:

· **OpenRouter API:** Provides standardized access to a variety of state-of-the-art Large Language Models (LLMs). This service is leveraged by the Core AI Analysis Service to interpret code semantics, generate comprehensive documentation, and explain code logic at different abstraction levels, forming the intelligent core of the application.