Nodify - The Graph-based Python Visualizer

Description

Throughout our work experience, our group has found that diagrams, flow charts, or other visual aids can be an efficient method of transferring high-level knowledge and ideas regarding a large-scale project. These aids are especially useful at reducing the mental strain and onboarding time for any newcomers to large projects, accelerating new-developer proficiency and increasing productivity. Furthermore, in our internships, we have used similar forms of documentation to understand our own work and the impact of changes within a system.

However, all too often, these diagrams quickly become dated and difficult to maintain, or sometimes don't exist in the first place, creating a significant amount of tedious and repetitive documentation overhead. Therefore, we want to create an automation tool to help developers debug, document, and understand large-scale Python projects. In its most basic form, Nodify will serve as live-update graphical documentation, with interactions to assist and simplify the process of navigating complex codebases. We are also planning to incorporate the graph as a VSCode extension so developers can utilize the benefits as they develop programs. Additionally, an LLM will be used to summarize and compartmentalize large blocks of code for even more efficient processing and understanding of the project. Finally, we plan to provide telemetry data of production services and functionality within the graph.

Our project will be broken up into several features with assigned priority levels, and each feature will be broken down into multiple tasks. We will use 2-week sprints and have weekly standups to stay on track, with Github as the main tool for product management. Our tech stack includes: Typescript, React, React Flow, OpenAI API (GPT 4o - mini), Node.js. We will be generating synthetic data using a bigger LLM model like Claude 3.5 Sonnet and train a smaller LLM by using that data.

Stakeholders:

- **Software Developers:** Contributors of medium to large scale python projects, especially those mentoring or onboarding other developers
- **Educators and Learners**: Assist those teaching and learning programming through visualizing code structure and performance
- **Devops**: Used to monitor real-time performance of a project in production with telemetry data
- **Architects**: To ensure efficient function interactions and spot bottlenecks or issues across codebase
- **Quality Engineers**: Performance and function flows for debugging and optimizing code and performing comprehensive unit testing