[CNCF - KCL]

LFX Mentorship 2024

KCL IDE Quick Fix

Name: Shashank Mittal
GitHub: shashank-iitbhu
LinkedIn: shashankmittal27
Email: shashank.mittal.mec22@itbhu.ac.in
Alternate Email: latashashank.3@gmail.com
Slack Username: Shashank Mittal
Country: India
Time Zone: IST (UTC+05:30)
Institution: Indian Institute of Technology (BHU) Varanasi
Course: Bachelor of Technology in Engineering (B.tech)

How did you find out about our mentorship program?

I found out about the LFX mentorship program through one of my friends. On exploring the site, I saw the program listed under open applications.

Why are you interested in this program?

I'm drawn to this mentorship opportunity because of my curiosity about the mechanisms behind IDE "Quick Fix" suggestions that significantly ease developers' tasks. Understanding the project's impact through community discussions has highlighted its potential to enhance automation and integration, fueling my motivation. This ambitious project promises a valuable learning experience in KCL and IDE functionalities, allowing me to contribute meaningfully to tools that benefit others. In addition, I'll get to test my Rust and Go capabilities.

What experience and knowledge/skills do you have that are applicable to this program?

I have been working on open source for more than a year now. I have been actively involved with another CNCF Project - Jaeger tracing demonstrating my Go, bash scripting and Rust skills. Enhanced the reliability and performance of the Jaeger project by fixing breaking changes introduced by a Dependant which required meticulous code adjustments. Spearheaded the

development of an integration test to ensure Kafka broker readiness. Identifying and resolving a critical race condition in the Go codebase, by designing a targeted unit test to detect and eliminate the concurrency issue.
Resume: Link
This, along with my work showcased on my main GitHub and alternate GitHub Accounts.
Main GitHub Account: shashank-iitbhu
Alternate GitHub Account: ishashankmittal

My involvement in KCL:

- Successfully completed the LFX pre test for LSP Quick Fix by implementing a Quick Fix feature for CompileError in the KCL IDE. This feature dynamically generates and presents code corrections within the IDE, streamlining the debugging process and improving coding efficiency. Link
- The pre test task showcases my proficiency in Rust as this task involved code improvements in multiple files in multiple directories.
- I was able to implement this feature quickly and continuously test it in a separate project because of my experience of Git and contributing to open source projects.
- My initial idea involved extending the KCL diagnostics struct but it made the code redundant and after going through the reviews, I was quickly able to fix this redundancy.
- I have been actively involved in Slack Channel and try to find bugs and resolve them.

What do you hope to get out of this mentorship experience?

By the end of the mentorship, I aim to get more proficient with Rust and Go and have a more profound knowledge of KCL and its components. However, the most valuable thing I hope to get out of this mentorship is guidance and experience, which will help me throughout my life.

Project Proposal

Title: KCL IDE Quick Fix

Description: This project aims to improve the KCL IDE by extending the already present “Quick Fix” feature that offers real-time, actionable solutions to common coding errors. The primary goal is to streamline the debugging process, enhance code quality, and boost developer productivity by seamlessly integrating Quick Fix prompts for selected errors directly within the IDE environment.

Implementation and Plan:

- **Week 1-2:**
    - Get familiar with the codebase and the mentors.
    - Discuss my approaches, logic, and milestones with the mentor.

    - Analyze **ErrorKind Enum** in **error.rs** and review each error type to understand its context, causes, and potential fixes.
    - Prioritizing a few errors where quick fixes are more straightforward and start with them.
- **Week 3-4:**
    - Understand **Error Contexts** and dive deeper into each selected error to understand when and why it occurs.
    - For each targeted error, design one or more quick fix suggestions. This might include template-based fixes, common mistake corrections, or guidance for more complex issues.
- **Week 5-6:**
    - Different errors and warnings might require different approaches for a Quick Fix feature which might not just fall under the scope of suggested replacement. Investigate those errors and warnings.
    - Midterm mentee evaluations.
- **Week 7-8:**
    - Effectively utilize the provide_suggestions function of suggestions crate.
    - Currently only the first suggestion out of the vector of suggestions returned by provide_suggestions is used as suggested_replacement. Look into this.
    - Write unit tests (if any) for newly implemented quick fix suggestions to ensure each fix is generated correctly based on the error context.
- **Week 9-10:**
    - Completion and testing of Code Actions associated with errors and warnings.
    - Brainstorming more possibilities and code enhancements.
- **Week 11-12:**
    - Documenting how the Quick Fix feature works, including how developers can extend or customize quick fix suggestions.
    - Code enhancements and Final mentee evaluations.

Future Deliverables:
I intend to stay with the community after the LFX mentorship and will try to optimize the project. I'll keep on contributing to improving the features of the KCL. It would give me immense pleasure to be a part of the KCL community and improve the lives of fellow developers.