**Real-Time Monitoring Dashboard and Terminal User Interface (TUI) for Crankshaft**

(St. Jude GSoC 2025 Proposal)

By: Natan Kassa

**Abstract:**

This project aims to build a real-time monitoring dashboard and terminal UI (TUI) for Crankshaft, a headless workflow execution engine designed to submit jobs across backend environments. While Crankshaft crates provide the language-agnostic infrastructure for submitting jobs, it lacks an intuitive way to track and visualize workflow status and system-wide activity at scale. This project will fill the gap by delivering real-time monitoring tools to ensure that end users know what’s going on with thousands of their workflows. Through the use of responsive dashboards, that provide high-level visibility into the performance metrics, status, and error states of workflows, and a terminal UI for lightweight interaction with jobs through the command line, engineers and researchers can monitor workflows with confidence, ensuring faster debugging and improved system reliability.

**Project Description:**

d

**Major Contributions:**

**Timeline:**

**Time Period/Working Hours:**

**Personal Information:**

Name: Natan M. Kassa

Email: [nkassa@usc.edu](mailto:nkassa@usc.edu)

GitHub: <https://github.com/nkassa>

Website: <https://natankassa.site/>

LinkedIn: <https://www.linkedin.com/in/natankassa/>

**Resume:**

<https://drive.google.com/file/d/1dnBw24Tej-9AEs1l33B3_noqP9hCDCh2/view?usp=sharing>



**Code Review:**

<https://github.com/nkassa/bike_safety_project>

One Project I am particularly proud of is my work with Signal Cycle Solutions, a startup created by myself and fellow college students at USC. Signal Cycle solutions is a bicycle safety enhancing system we developed to improve the safety and overall riding experience on college campuses or areas with high traffic. This project involved integrating hardware and software components to create a “smart bike” system that enhances rider awareness and allows for communication with others on the road. I was involved in the hardware setup but mostly focused on leading the integration of software with the setup of the hardware. I wrote the embedded code for the ATMEGA328P microcontroller to manage inputs and outputs of sensors and modules in real-time. Some of the key features of our safety system include:

* Live speed monitoring via a rotary encoder and LCD
* Turn signal control using push buttons to activate LEDs, allowing riders to indicate turns without using outdated hand gestures
* Obstacle-avoiding lane switching using sonar sensors to detect nearby obstacles; the system intelligently prohibits unsafe lane switches by disabling turn signal activation when object is detected

The project required interrupt-driven C/C++ code to manage sensor inputs and actuator output, initiated by the user, in a safe and responsive manner. I implemented all logic for interpreting sensor readings, managing obstacle safe rules, and providing real-time safety and speed feedback to the rider. You can find the code sample for this project in the link above. This sample reflects my ability to write clean code while navigating through critical safety and real-wide constraints.