Google Developer Groups

On Campus • Indian Institute of Information Technology Kalyani

Winter of Code 5.0

Proposal

for the Project

“LIBCUPSFILTERS”

under

Open Printing

making printing just work

About Me

Name - Anusha Muhuri **(Student, 1st year, B.Tech CSE, IIIT Kalyani)**

Email-Id - cse25018@iiitkalyani.ac.in

GitHub Username - anushamuhuri7

Country - India

Timezone - GMT+5:30

Primary Language - English

LinkedIn Profile Link - www.linkedin.com/in/anusha-muhuri-052545394

Other Links (If Applicable) - https://github.com/anushamuhuri7 (GitHub), https://devfolio.co/@anushamuhuri (Devfolio).

Synopsis

LIBCUPSFILTERS is a foundational library developed by OpenPrinting that centralizes the document conversion logic—formerly handled by dozens of standalone CUPS filter executables—into a single, streamlined framework. Its primary purpose is to provide the "engine" for converting various input files (such as PDFs, images, or text) into printer-ready data formats like PWG Raster or PCLm, which are essential for modern driverless printing standards like Apple AirPrint and Mopria.

While the main CUPS repository has excellent CI testing (static analyzers and multi-distro build tests), LIBCUPSFILTERS lacks a similar automated system. As an essential part of OS infrastructure, every commit to this library needs to be verified automatically to prevent regressions or bugs that could break printing services across various Linux distributions.

Benefits to the Community

Implementing this project helps OpenPrinting in several key ways:

- **Quality Standards**: It brings LIBCUPSFILTERS up to the high standards of the main CUPS repository by replicating its successful CI model.
- **Infrastructure Reliability**: As an essential part of the OS, automated testing ensures every commit is stable and does not break the printing system.
- **Broad Compatibility**: It verifies that the C code builds and executes correctly across different Linux distributions and architectures.
- **Error Prevention**: By applying static analyzers and execution tests, the project prevents bugs from reaching the final operating system.

Project Plan

My plan follows a four-stage approach to move from local learning to global automation:

- **Stage 1: Local Environment Setup**: I will set up the project on WSL 2 (Ubuntu) and use the ***./configure*** and ***make*** processes to manually compile LIBCUPSFILTERS.
- **Stage 2: Basic Workflow Creation**: I will write the core GitHub Actions script (.yml). This script will act as a "robot" that triggers a build every time code is pushed to the repository. I will start with an Ubuntu environment to ensure a "Green Checkmark" appears on successful compilation.
- **Stage 3: Multi-Distro & Static Analysis**: I will expand the script so that it works on other environments like Fedora. I will also

integrate Cppcheck, to automatically scan the C code for memory leaks or logic errors.
- **Stage 4: Documentation & Final Polish**: I will document how the CI works so that future contributors know how to read the test results and add new tests if needed.

I will complete this project using an iterative testing method. I will perform all "trial and error" work on a personal fork of the repository. Only once the GitHub Action works perfectly on my fork will I submit a Pull Request to the main OpenPrinting organization. I will also maintain a weekly communication log with my mentors to share my progress and clear any "blockers" early.

The Tech Stack

To ensure the project is stable and professional, I will use:

- Language: C (to understand build errors and compiler warnings).
- Platform: GitHub Actions (the automation engine).
- Configuration: YAML
- Build Tools: Autotools (Autoconf) and Make (to manage the complex C compilation).
- Local Dev: WSL 2 (Ubuntu) and VS Code.
- Analysis Tools: Cppcheck (to catch bugs automatically).

Milestones

Milestone	Tentative Date	KPI
Week 1	25.01.2026-31.01.2026	Local Environment Setup (100% Build Success)
Week 2	1.02.2026-7.02.2026	The Hello World of CI (Green Checkmark)
Week 3	8.02.2026-14.02.2026	Multi-Platform Validation (Zero Portability Failures)
Week 4	22.02.2026-28.02.2026	Automated Quality Gate (100% Scan Coverage)
Week 5	1.03.2026-7.03-2026	Execution Testing (Binary Verification)
Week 6	8.03.2026-14.03.2026	Final Handover (Code Merge Readiness)

Deliverables

By the end of my 6-week contribution, I will have handed over the following items to the OpenPrinting team:

- The Workflow Scripts: I'll deliver the actual .yml files for GitHub Actions. These are the "instructions" that tell the GitHub runners exactly how to pull the code, install the dependencies we talked about, and run the compiler.
- Multi-OS Verification: A working setup that doesn't just check Ubuntu. I'll make sure the CI checks the code against Fedora as well, catching those annoying bugs that only show up on different Linux versions.
- The "Bug-Finder" Integration: A configured Cppcheck step in the pipeline. It'll be set up to "fail the build" if it finds things like memory

leaks or uninitialized variables—errors I've learned to watch out for in my C programming class.

- Clean Build Badges**:** I'll add "Build: Passing" badges to the main README. It's a small thing, but it shows everyone the library is stable and the CI is doing its job.
- Initial Cleanup Report: A list of any existing warnings my new tools found during the setup phase so the mentors can see where the code needs a quick polish.

Acknowledgement

I want to thank OpenPrinting for organizing this project. Finding an opportunity to work on something as significant as LIBCUPSFILTERS, where I can truly apply the C programming I am learning in class, is a wonderful opportunity for a first-year BTech student.

Special thanks to GDG IIIT Kalyani for hosting Winter of Code 5.0. Open source becomes much less intimidating for beginners like me because of programs like these. I doubt I would have known how to begin contributing to a professional organization if it were not for this event.

Additionally, it has been really beneficial to examine the current workflows in the LIBCUPSFILTERS repository. It has provided me with a clear roadmap on what constitutes high-quality testing, and I am eager to attempt to apply that same standard to this project.