Open Source Engineering

Self Hosting & Contributions Report

Sirisha Duba
ID - 2400031818

B.Tech
Computer Science and Engineering
2nd year
KL University
Vaddeswaram

Supervisor:

Dr. Sripath Roy Koganti

1 About the Linux Distro Used: Ubuntu

For this project, I used Ubuntu, a popular and beginner-friendly Linux operating system. Ubuntu is stable, secure, and widely used for development and open-source work.

Why I chose Ubuntu:

- Easy to install and use
- Long Term Support (LTS) version
- Strong community support
- Works well for programming, Git, and self-hosting

Key Features:

- GNOME desktop (simple and clean UI)
- Built-in security tools
- Regular updates and patches

What I used Ubuntu for:

- Running terminal commands
- Installing tools
- Self-hosting server
- Making my open-source PRs

2 Encryption and GPG

In my project, I explored how GPG encryption works on Linux and learned how it can be used to protect files and messages. Using the terminal in Ubuntu, I generated a basic GPG key pair and understood the difference between a public key and a private key. With GPG, I can encrypt any text file, protect sensitive information, and share encrypted data safely with others. I did simple tasks like checking my keys, exporting the public key, and trying sample encryption commands. Even though I did not use it heavily in the project, I understood how GPG can be used in real situations such as sending secure emails, protecting configuration files, and verifying the authenticity of downloaded software.

3 Sending Encrypted Email

As part of learning GPG, I also tried how encrypted emails can be sent from Linux. The process is simple: first I used my GPG public key to encrypt a sample text file, which converted it into a secure .asc file. This encrypted file can be attached to an email and sent safely because no one can read it without the matching private key. Even though I didn't use encrypted emails regularly in the project, I understood how the method works and how it can be used for sharing important information like login details, configuration files, or private messages in a protected way. The main idea is that only the intended person who has the correct private key can open and read the encrypted email content.

4 Any 5 Privacy tools

For my project, I checked different privacy tools listed on prism-break.org and explored a few that are useful for secure communication and safe browsing.

FireFox: Firefox is a privacy-friendly web browser that blocks trackers and gives users control over their data.

DuckDuckGo: DuckDuckGo is a search engine that does not store your search history or track your online activity.

Signal: Signal is a messaging app that uses strong end-to-end encryption for chats and calls, keeping conversations private.

ProtonMail: ProtonMail is a secure email service that automatically encrypts your emails so only the sender and receiver can read them.

Tor Browser: Tor Browser protects your identity online by routing your traffic through multiple servers, making your browsing anonymous.

5 About the Open Source License I Used

In my project, I selected the MIT License because it is one of the easiest and most developer-friendly open-source licenses available. The MIT License is known for being very short, simple, and flexible. It allows anyone to use my project for almost any purpose - they can download it, modify it, reuse it in their own work, or even use it commercially. This makes it a great choice for students, beginners, and open-source contributors who want their code to be easily accessible.

The main condition of the MIT License is that anyone who uses my code must keep the original copyright notice and the license text in their project. Other than that, there are no limitations. It does not force others to open-source their changes, and it does not restrict how the code is used. This level of freedom encourages collaboration and helps others learn and build on top of my work.

I added the license file in my GitHub repository so that the project is clearly marked as MIT licensed. This shows that my project follows open-source standards and can be safely reused by anyone without legal issues. Using the MIT License helped me understand how open-source licensing works and why it is important to include a license when publishing code publicly.

6 Self-Hosted Server: Lychee (PhotoHub)

For the self-hosting part of my project, I installed and set up a Lychee photo management server on my Linux system. I named my self-hosted server "PhotoHub" because it acts as a central place to store, manage, and view photos through a clean web interface. Lychee is a fully open-source photo gallery system that allows me to upload images, create albums, organize media, and access everything from any device on the same network. Running it on my own server gave me full control over my photos without depending on cloud platforms.

About Lychee and Why I Chose It:

Lychee is very lightweight, easy to install, and perfect for hosting personal photos privately. It runs on a simple stack using Apache and MySQL/MariaDB, which makes it suitable even for beginners. I chose Lychee because:

- It is open-source
- It has a beautiful and simple interface
- It supports albums, tags, descriptions, filters
- It can be accessed from any browser
- It gives full privacy since photos stay on my own system

By hosting PhotoHub, I learned how a real self-hosted web application runs on a Linux server.

Localized (Translated) Document:

To show localization support, I translated a small description of PhotoHub into Telugu and added it in my report.

Here is a translated text:

```
@@ -0,0 +1,52 @@
1  + ## Self-Hosted Project: Lychee
2  +
3  + ## ప్రాజెక్ట్ పేరు
4  + Lychee – ఫోటో మేనేజ్మెంట్ సిస్టమ్
5  +
6  + ## ఇది ఏమి చేస్తుంది?
7  + Lychee ఒక ఓపెన్ సోర్స్ ఫోటో మేనేజ్మెంట్ టూల్.
8  + మీ కంప్యూటర్ లేదా సర్వర్లో ఫోటోలను అప్లోడ్ చేసి,
9  + అవన్నీ ఆల్బమ్స్లో సింపుల్గా మేనేజ్ చేయడానికి ఇది ఉపయోగపడుతుంది.
10 +
11 + Google Photos లాంటి ఫీచర్లు మీ దగ్గర, మీ కంట్రోల్లో ఉంటాయి.
12 +
13 + ---
14 +
15 + ## ఫీచర్స్
16 + - ఫోటోలను అప్లోడ్ చేయడం
17 + - ఆల్బమ్స్ క్రియేట్ చేయడం
18 + - ఫోటోలను షేర్ చేయడం
19 + - ప్రైవేట్ మరియు పబ్లిక్ యాక్సెస్ సపోర్ట్
20 + - మొబైల్ మరియు వెబ్ బ్రౌజర్లో యాక్సెస్
21 +
22 + ---
23 +
24 + ## ఎలా ఇన్స్టాల్ చేయాలి (Telugu Steps)
25 +
26 + ### అవసరమైనవి
27 + - PHP (>= 8.0)
28 + - MySQL / MariaDB
29 + - Apache / Nginx
30 + - Git
```

```
22 + ---
23 +
24 + ## ఎలా ఇన్స్టాల్ చేయాలి (Telugu Steps)
25 +
26 + ### అవసరమైనవి
27 + - PHP (>= 8.0)
28 + - MySQL / MariaDB
29 + - Apache / Nginx
30 + - Git
31 +
32 + ### ఇన్స్టలేషన్ స్టెప్స్
33 + 1. GitHub నుండి Lychee ప్రాజెక్ట్ను క్లోన్ చేయండి:
34 +
35 + ```bash
36 + git clone https://github.com/LycheeOrg/Lychee.git
37 + cd Lychee
38 +
39 + అవసరమైన ప్యాకేజీలు ఇన్స్టాల్ చేయండి:composer install
40 +
41 + .env.example ఫైల్ని కాపీ చేసి .env పేరుతో మార్చండి:cp .env.example .env
42 +
43 +
44 + మీ డేటాబేస్ సెట్టింగ్స్ .env ఫైల్లో ఎడిట్ చేయండి.
45 +
46 + సర్వర్ రన్ చేయండి:php artisan serve
47 +
48 + ఇప్పుడు బ్రౌజర్లో http://127.0.0.1:8000 ఓపెన్ చేస్తే Lychee వెబ్ యాప్ కనిపిస్తుంది.
49 +
50 + https://drive.google.com/file/d/1T9L6zf4UqgGgcIy6Pxzwnq-2A_N7o-fu/view?usp=sharing
51 +
52 + https://www.linkedin.com/pulse/photo-hub-self-hosted-lychee-server-sirisha-duba-zt1de/?trackingId=K98Nco65SUW%2F298VJnmatgX3D%3D
```

Open Source Engineering

PhotoHub Self Hosted Server

- Photo Hub is a lightweight, open-source photo management server.It allows users to securely upload, organize, and share images via a web interface.
- Designed for speed and simplicity, it works perfectly on personal servers or small teams.

Highlights / Features

- Self-Hosted — Full control over your photos and data.
- Fast & Lightweight — Optimized for low-power devices and small servers.
- Secure Access — Private albums and authentication support.
- Web-Based Interface — Easy to manage from any browser.
- Database Support — Compatible with MySQL, PostgreSQL, or SQLite.
- Extensible — API and plugin support for developers.
- Automatic Image Optimization — Smart resizing and metadata handling.
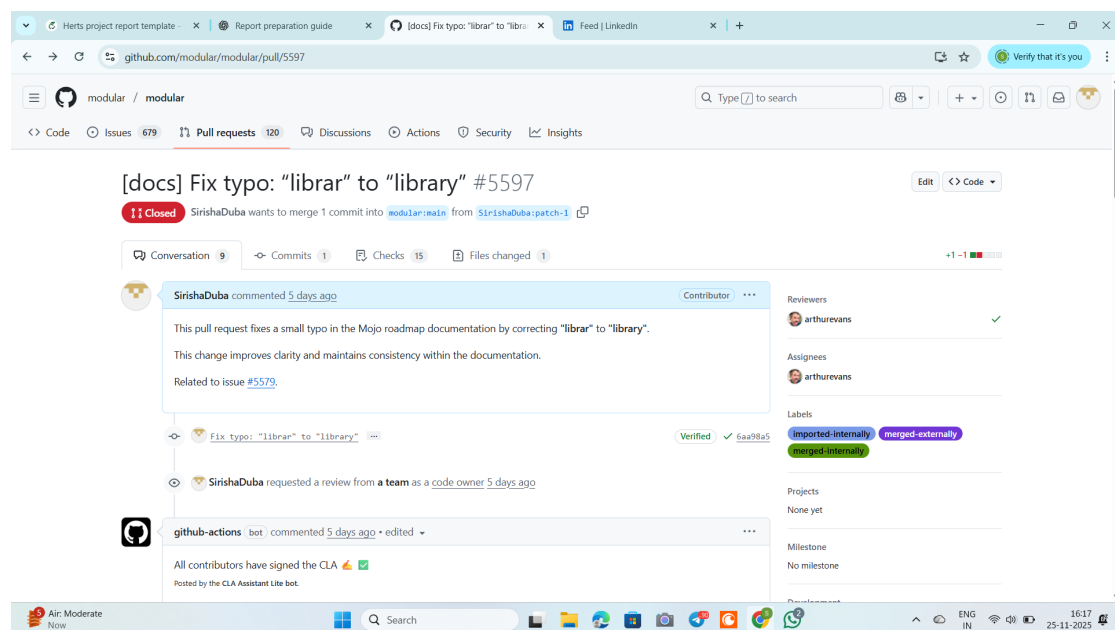
License :

MIT License

## Hosted by

Y. Navya Sree - 2400032695
D. Sirisha - 2400031818

9

7 Open Source Contributions

PR 1 — modular/modular I corrected a small spelling mistake in the documentation where "librar" was incorrectly written. I raised a PR with the corrected change, and it was reviewed and merged.
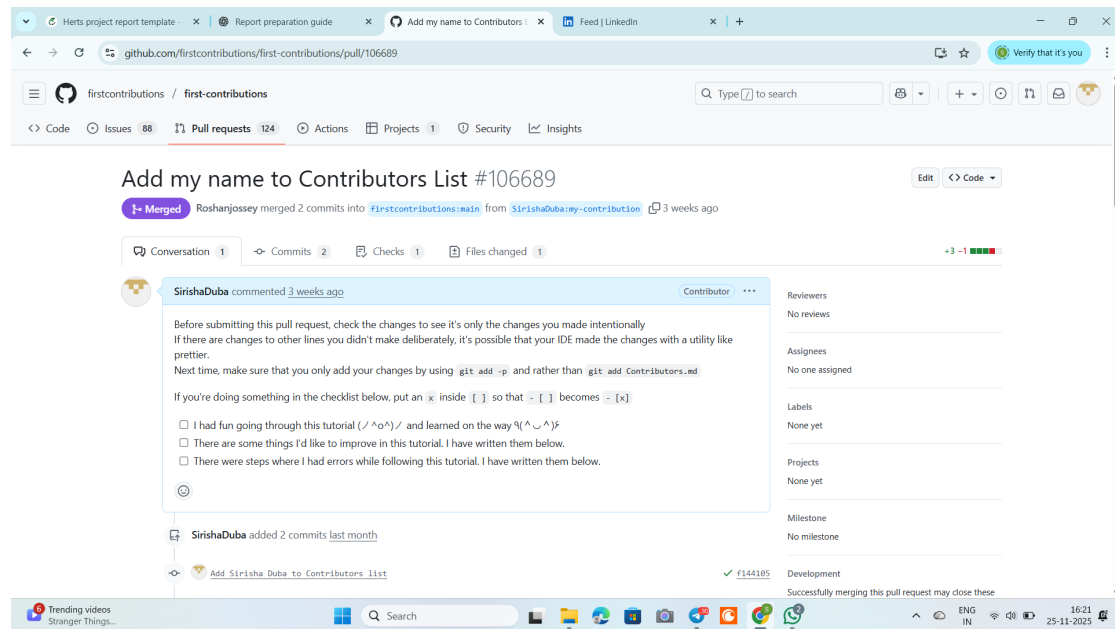

PR 2 — mnkiefer/repo-with-bugs In the CSS file, the property was wrongly written as border-botto. I corrected it to border-bottom which fixed the header styling issue.
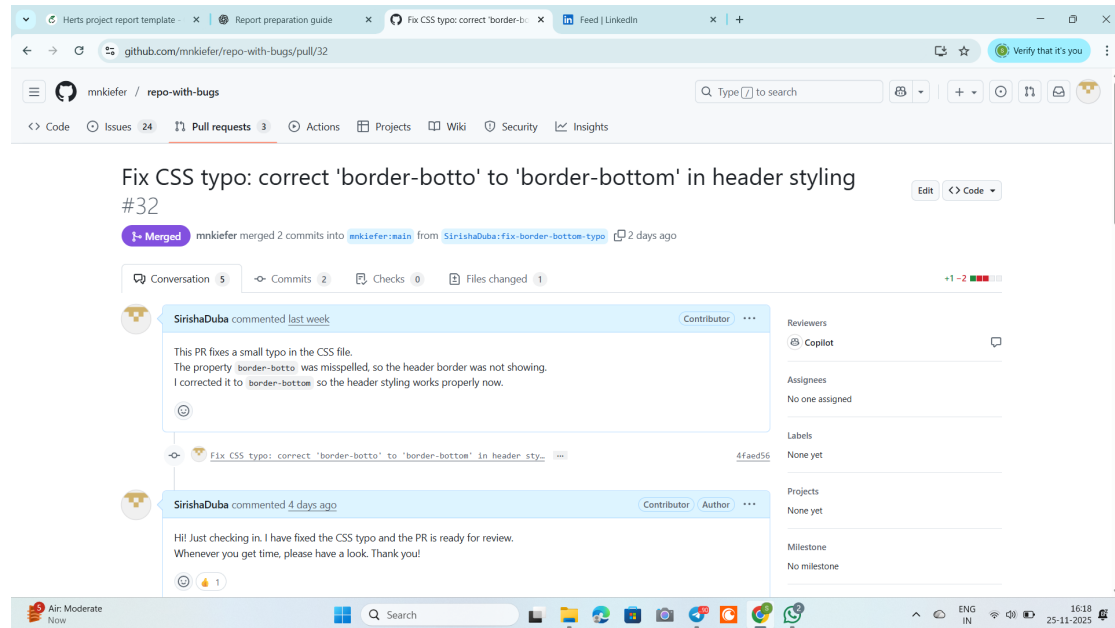
PR 3 — godotengine/godot I fixed a documentation typo in the Godot Engine project. Small documentation improvements help maintain clarity for developers.
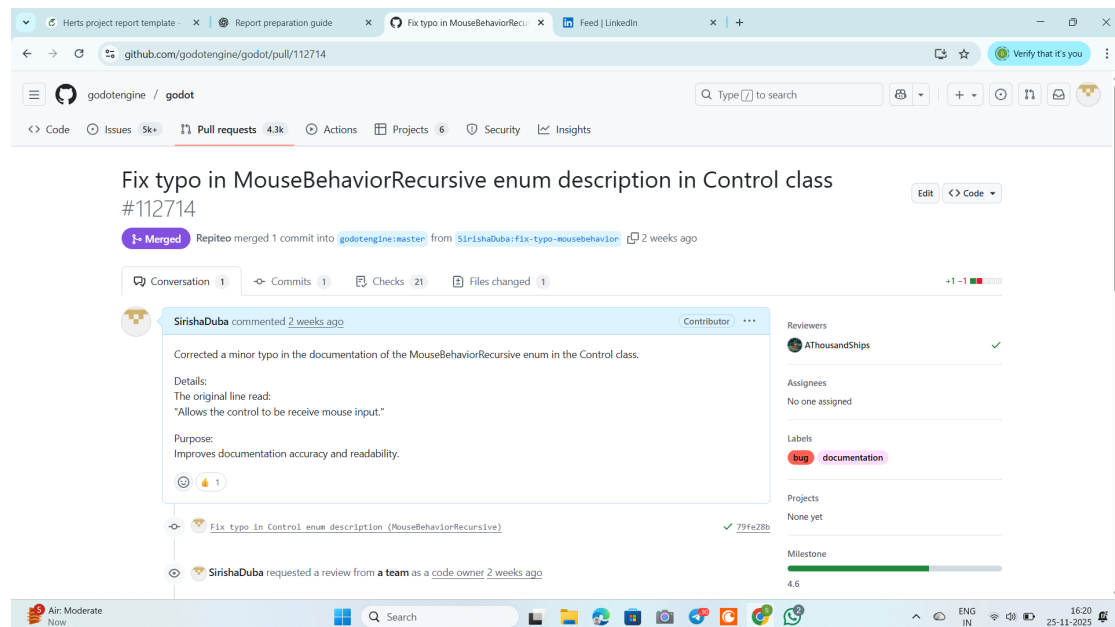
PR 4 — firstcontributions/first-contributions   I added my name "SirishaDuba" to the contributors list as part of a beginner-friendly open-source project.

8 LinkedIn Posts

SELF-HOSTING:

https://www.linkedin.com/pulse/photo-hub-self-hosted-lychee-server-sirisha-duba-zt1de/?trackingId=Hda

BLOG:

https://www.linkedin.com/pulse/building-skills-through-linux-self-hosting-open-source-sirisha-duba-zriff/?trackingId=HdaNlt0oSKyS%2Bb%2F8T%2BUKiQ%3D%3D

PR MERGES:

PR 1 — modular/modular
https://github.com/modular/modular/pull/5597

PR 2 — mnkiefer/repo-with-bugs
https://github.com/mnkiefer/repo-with-bugs/pull/32

PR 3 — godotengine/godot
https://github.com/godotengine/godot/pull/112714

PR 4 — firstcontributions/first-contributions
https://github.com/firstcontributions/first-contributions/pull/106689