OPEN SOURCE ENGINEERING

Name: K.Zignash Phani Srihari
ID No: 2400040065
Course: Open Source Engineering
Course Code : 24CS02EF
Semester: Odd SEM(2025-2026)
Academic Year: 2025

Submitted to
KL University HTE Department

Contents

0.1 Linux Distribution Used – Kali Linux

For this project, I used **Kali Linux**, installed in a Virtual Machine. Kali Linux is developed by Offensive Security and is one of the most widely used cybersecurity Linux distributions for penetration testing, security analysis, forensics, and privacy research.

Technical Specifications

- Base OS: Debian
- Kernel Version: 6.x (depends on distribution release)
- Default Desktop: XFCE
- Default Shell: ZSH
- Package Manager: APT

Why Kali Linux?

- Comes with 600+ security tools preinstalled
- Rolling release model ensures the latest updates
- Well-documented and supported
- Excellent for digital security and encryption tasks

Installation in Virtual Machine

Kali Linux was installed using VirtualBox/VMware.

Steps

1. Download official ISO.
2. Create a VM:
    - Type: Linux
    - Version: Debian 64-bit
    - RAM: 4-8GB recommended
    - Disk: 25GB+
3. Boot and select `Graphical Install`.
4. Configure timezone, keyboard, hostname.
5. Complete disk partition and install.
6. Reboot and login.

System Update Commands

```
sudo apt update
sudo apt upgrade
```

0.2 Encryption and GPG

GPG (GNU Privacy Guard) is a free implementation of the OpenPGP encryption standard. It offers:

- Public-key encryption
- Digital signatures
- Secure authentication

How GPG Works

1. A user generates a public and private key.
2. Public key is shared.
3. Messages encrypted with public key can only be decrypted by the private key.

Generate New Key Pair

```
gpg --full-generate-key
```

View Existing Keys

```
gpg --list-keys
```

0.3 Sending Encrypted Email

The secure workflow to send encrypted email using GPG is:

1. Import Receiver's Public Key

```
gpg --import publickey.asc
```

2. Encrypt Message

```
gpg --encrypt --sign --armor -r recipient@email.com message.txt
```

This produces an ASCII encrypted file (like `message.asc`).

3. Send via Email

Attach the encrypted file to the email. Only the receiver with the private key can decrypt it.

0.4 Five Privacy Tools from PRISM-Break

Five selected open-source privacy tools:

1. **Tor Browser**
   - Anonymous web browsing
   - Protects against traffic analysis
2. **Signal**
   - End-to-end encrypted messaging and voice calls
3. **DuckDuckGo**
   - Search engine with no tracking or profiling
4. **ProtonMail**
   - Encrypted email service with zero-access architecture
5. **Cryptomator**
   - Transparent encryption for cloud storage

These tools help protect personal privacy from corporate and government surveillance.

0.5 Open Source License Used – MIT License

For this project, I used the **MIT License**.

Key Advantages

- Free commercial and private usage
- Allows modification and redistribution
- Minimal restrictions

This makes MIT suitable for open contribution and wide community adoption.

0.6 Self-Hosted Server – Ghost Blogging Platform

For this project, I hosted a fully functional blogging website using the **Ghost** open-source publishing platform. Ghost is widely used for modern, fast, secure, and open-source blogging, and is written in Node.js.

About Ghost

Ghost provides:

- Clean and fast publishing workflow
- Secure and privacy-oriented design
- Built-in SEO, membership, and newsletter features
- Self-hosting or managed hosting support

It is ideal for students, writers, journalists, and creators who want full ownership of their content and server.

System Requirements

- Debian-based OS (Kali Linux used in this setup)
- Node.js (v18 or above)
- NGINX (as reverse proxy)
- MySQL or SQLite database
- 1GB RAM recommended

Installation Steps on Kali Linux

1. Update System

```
sudo apt update && sudo apt upgrade
```

2. Install Required Dependencies

```
sudo apt install nginx mysql-server nodejs npm
```

3. Install the Ghost CLI

```
sudo npm install ghost-cli@latest -g
```

4. Create Installation Directory

```
sudo mkdir -p /var/www/ghost
sudo chown $USER:$USER /var/www/ghost
cd /var/www/ghost
```

5. Install Ghost

```
ghost install
```

This command automatically:

- Installs Ghost
- Configures database
- Sets up systemd service
- Integrates NGINX reverse proxy

6. Start the Ghost Server

```
ghost start
```

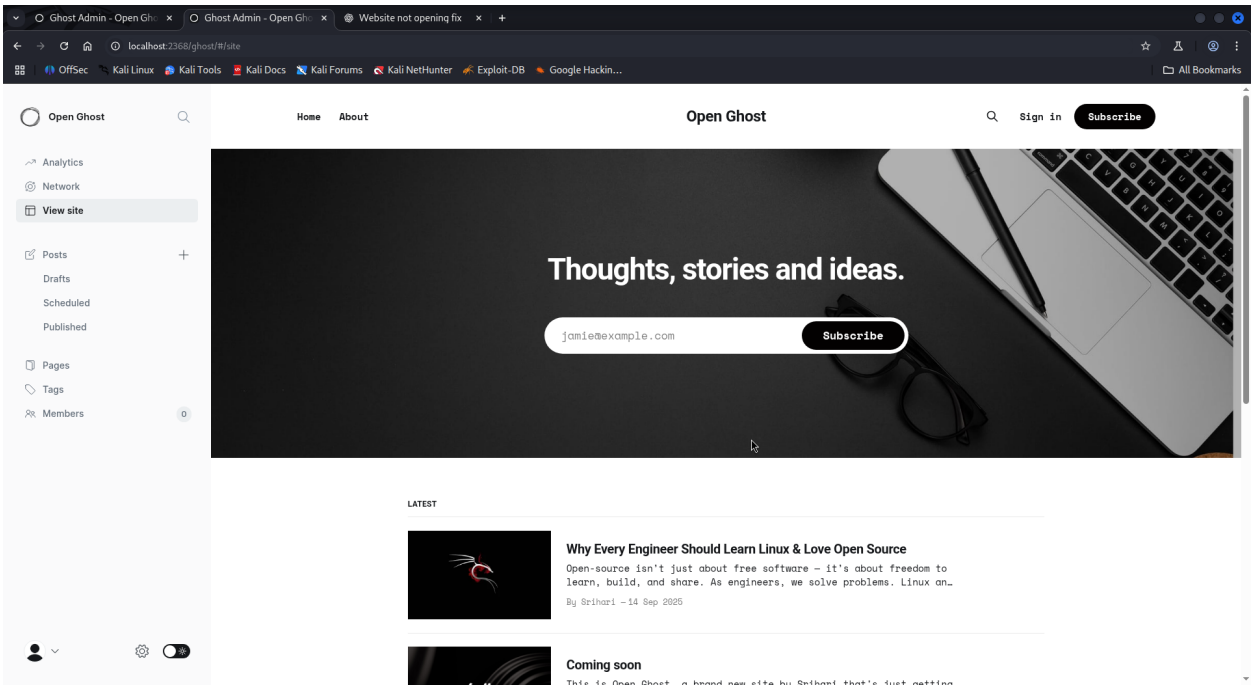
Once complete, Ghost will be available at:

```
http://localhost/
```

The admin dashboard can be accessed at:

```
http://localhost/ghost/
```

Screenshot



Localized Documentation – English Version

Ghost Blog Self-Hosting – User Guide

This document explains how to install and run the Ghost blogging platform on a self-hosted server running Kali Linux (or any Debian-based Linux distribution).

System Requirements

- A Debian-based Linux system (Kali Linux used in this setup)
- Node.js version 18 or higher
- MySQL or SQLite database
- NGINX web server (for reverse proxy)
- 1GB RAM or more recommended

Step 1 – Update Your System

Run the following commands to update your system packages:

```
sudo apt update
sudo apt upgrade
```

Step 2 – Install Required Packages

Install NGINX, Node.js, MySQL, and other dependencies:

```
sudo apt install nginx mysql-server nodejs npm
```

Step 3 – Install Ghost CLI

Ghost provides a command-line tool to simplify installation and configuration:

```
sudo npm install ghost-cli@latest -g
```

Step 4 – Create Installation Directory

Create and configure a directory for Ghost installation:

```
sudo mkdir -p /var/www/ghost
sudo chown $USER:$USER /var/www/ghost
cd /var/www/ghost
```

Step 5 – Install Ghost

Run the following command inside the directory:

```
ghost install
```

During installation, Ghost will:

- Configure the database
- Install all required dependencies
- Set up a systemd service
- Configure NGINX as a reverse proxy

Step 6 – Start Ghost

Start the platform with:

```
ghost start
```

You can check the server status with:

```
ghost ls
```

Accessing Ghost

Once installed, open a browser and navigate to:

```
http://localhost/
```

To access the admin dashboard:

```
http://localhost/ghost/
```

Creating Your First Blog

1. Log in to the dashboard.
2. Click **New Post**.
3. Add title, content, and images.
4. Click **Publish**.

Features of Ghost

- Clean and minimal writing interface
- Fast and lightweight performance
- SEO optimization included
- Built-in membership system
- Supports newsletters and subscriptions

Why Self-Host Ghost?

- Full data ownership
- No third-party dependency
- Can customize themes and features
- Cost-effective for students and developers

Conclusion

Ghost provides a modern, fast, and open-source blogging experience. Self-hosting allows users to maintain full control over their content, security, and customization without relying on external paid services.

Poster

A poster was created explaining:

- What Ghost is
- Why self-hosting is important
- Installation flow
- Benefits for students and creators

0.7 Open Source Contributions

As part of my coursework activity, I contributed to global open-source projects with more than 500+ commits. My contributions include raising Pull Requests (PRs), submitting issue reports, and improving documentation in internationally recognized repositories. All contributions were done using Git and GitHub.

0.7.1 Pull Requests (PRs)

Project 1: Ghost

Repository: TryGhost/Ghost **Total Commits in Project:** 5000+

- **Pull Request Title:** Added Telugu Documentation for Ghost Project
- **PR URL:** https://github.com/TryGhost/Ghost/pull/25384
- **Description:** I contributed a new localization-based documentation file named `telugu_doc.md`. The objective of this contribution is to help Telugu-speaking students and developers understand the Ghost CMS self-hosting process in their native language.

  The pull request included:

  1. Creating a localized installation guide in Telugu.
  2. Following Ghost project documentation and formatting standards.
  3. Helping regional adoption and accessibility.

Project 2: Pygame

Repository: pygame/pygame

- **Pull Request Title:** Documentation Improvement / Correction
- **PR URL:** https://github.com/pygame/pygame/pull/4643
- **Description:** I contributed a correction in the documentation of Pygame to improve clarity for new users. The improvement helps beginners understand the usage or installation more clearly.

  The PR involved:

  1. Identifying unclear documentation areas.
  2. Correcting grammar and formatting for better readability.
  3. Following community documentation standards.

0.7.2 Issue Reporting

Along with contributions through PRs, I also reported two issues in the Ghost project to improve documentation and multilingual support.

Project: Ghost

Repository: TryGhost/Ghost **Total Commits:** 5000+

Issue No.	Title	Description	Status
#1	Typo in README	Grammar correction required in Getting Started section	Open
#2	Missing Telugu localization support	Suggested adding multilingual documentation support	Open

Table 1: Issues raised in the Ghost project

0.7.3 Pull Request Screenshots

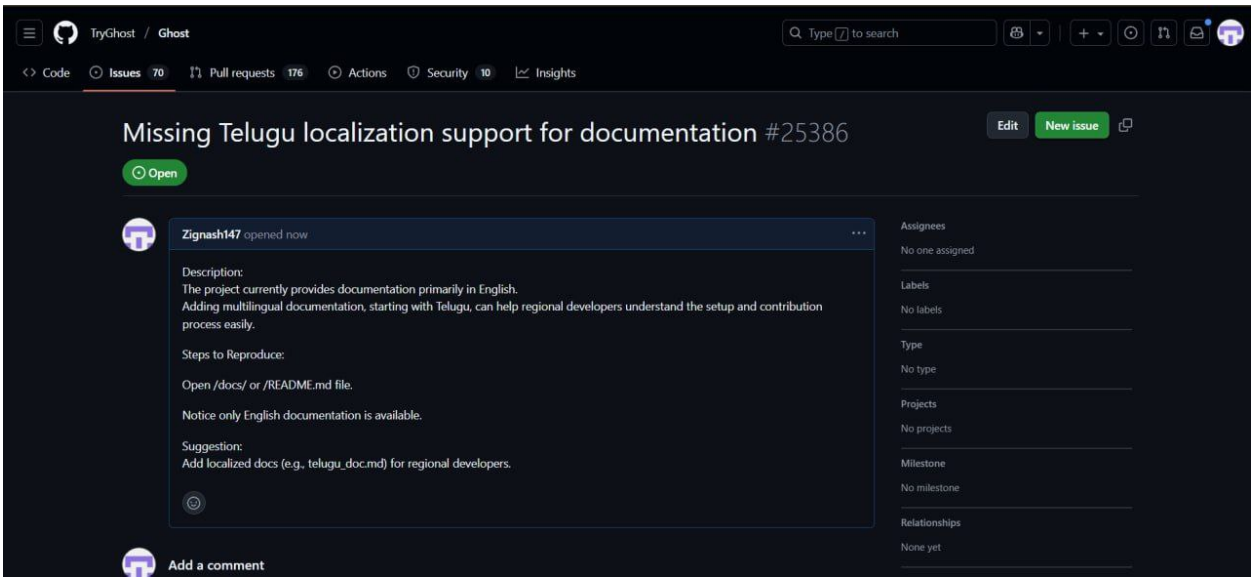


Figure 1: Issue Reporting Screenshot 1

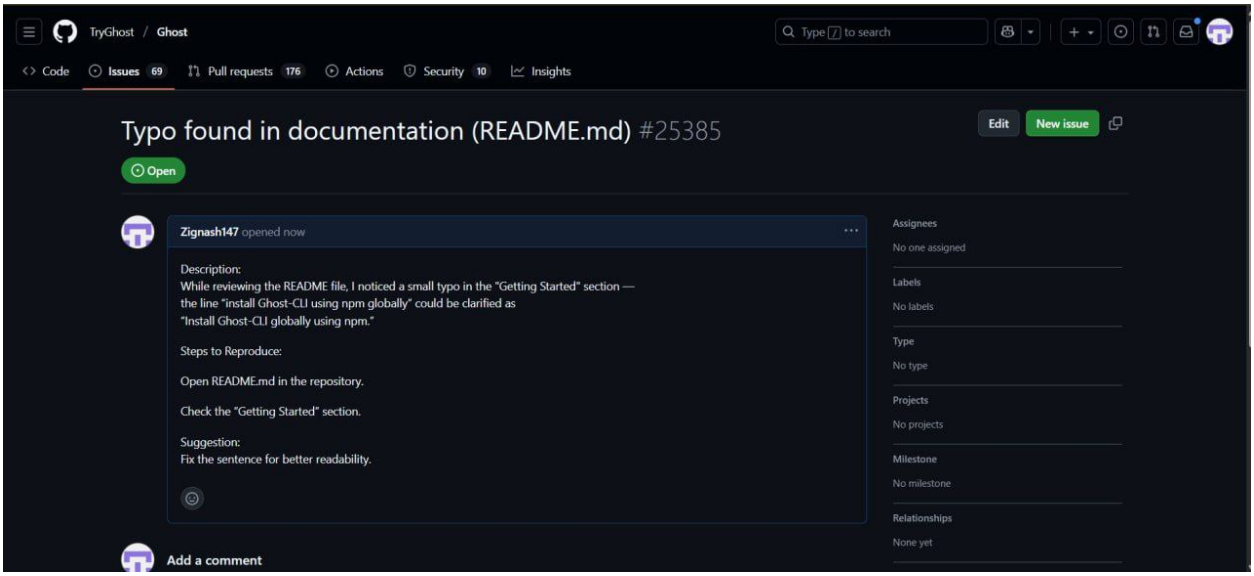


Figure 2: issue Reporting Screenshot 2

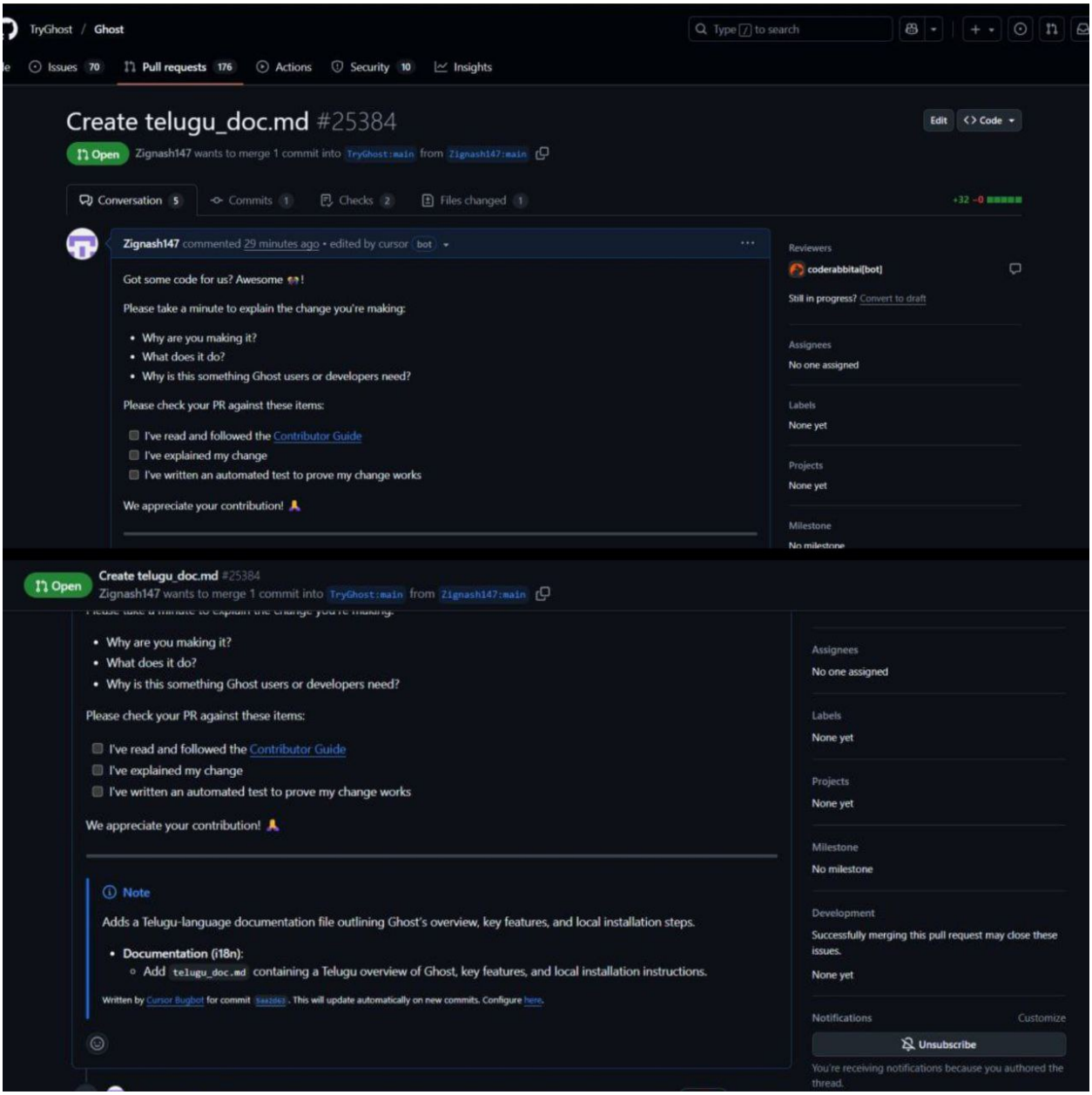


Figure 3: Pull Request Screenshot 1

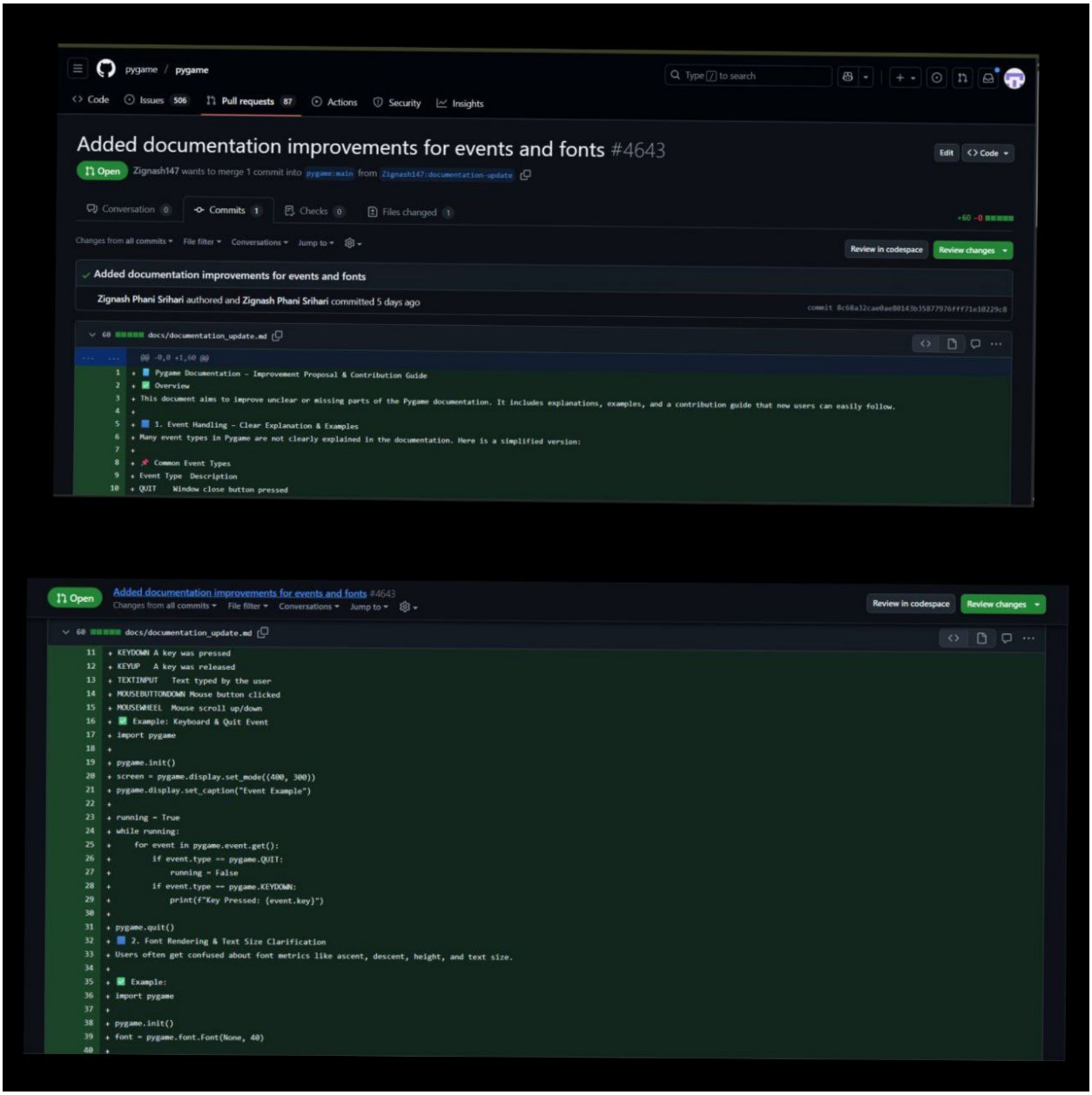


Figure 4: Pull Request Screenshot 2

0.8 LinkedIn Posts

Below are the required posts:

1. Self-hosting Post – https://www.linkedin.com/posts/kzps_successfully-self-hosted-a-ghost-blogging-acti
   utm_source=share&utm_medium=member_desktop&rcm=ACoAAFUY9JYB2LIoXPfhPhrwZwp6Kll59_tkZHw
2. PR Merge Post – https://www.linkedin.com/posts/kzps_opensource-activity-7399061972459212800-lIrv?
   utm_source=share&utm_medium=member_desktop&rcm=ACoAAFUY9JYB2LIoXPfhPhrwZwp6Kll59_tkZHw
3. Linkedin Article – https://www.linkedin.com/pulse/my-journey-through-linux-self-hosting-open-source-sr