OPEN SOURCE ENGINEERING PROJECT REPORT

Submitted by

P RAHUL

Roll No: 2400032050

Department: Honors Through Experiential Learning

Submitted to

Department of Computer Science Engineering

Academic Year 2024–2025

Contents

Chapter 1

About the Linux Distribution I Used

I have used Ubuntu 24.04 LTS which is a Debian-based open-source Linux distribution. When I first started using it I had no prior knowledge of Linux so everything felt completely new—from the terminal to the way files are organized. I chose Ubuntu because it is easy to install and has strong community support which makes it beginner-friendly. I explored the GNOME desktop environment and learned to install software using APT and Snap package managers. I also understood the importance of the built-in firewall called ufw which helps secure the system. Using Ubuntu gave me my first hands-on experience with Linux. Gradually I became more comfortable navigating and using the system for daily tasks. I learned how to manage files and directories, understand file permissions, and work with repositories. This experience helped me understand how Linux works, why it is widely used in the open-source world, and how it gives users more control compared to other operating systems.

Some frequently used commands:

- `sudo apt update` – to update the package lists for upgrades

- `sudo apt install gpg thunderbird git nodejs npm` – to install important tools and software

- `ls`, `cd`, `cat` – to list files, navigate directories, and view file contents

- `chmod`, `chown` – to modify file permissions and ownership

Using Ubuntu regularly helped me understand permissions, directory structure, repositories, package management and system control much better than before. It also improved my confidence in using the terminal, troubleshooting errors and exploring open-source tools which has been a valuable foundation for learning Linux and open-source development.

Chapter 2

Encryption and GPG

During this course I also learned about encryption and GPG (GNU Privacy Guard) which are important tools for securing data and communications. At first I had no idea how encryption worked but I gradually understood that it helps protect sensitive information so that only the intended person can read it. GPG uses a pair of keys: a public key which you can share with others and a private key which you must keep secret. Messages or files encrypted with your public key can only be decrypted using your private key ensuring privacy and security.

I practiced creating my own GPG key pair using the terminal and learned how to export my public key to share with others. I also tried encrypting and decrypting files on my laptop to see how the process works. For example I used commands like:

- `gpg --full-generate-key` – to create a new GPG key pair with more options

- `gpg --list-keys` – to see the public keys available on your system

- `gpg --export -o publickey.asc "Your Name"` – to save/export your public key to a file

- `gpg --import publickey.asc` – to import someone else's public key

- `gpg -e -r "Recipient Name" secret.txt` – to encrypt a file for a specific person

- `gpg -d secret.txt.gpg` – to decrypt a file you received

Using GPG helped me understand the basics of data security, encryption and digital signatures. It also showed me how open-source tools provide strong privacy protections without relying on proprietary software. Practicing encryption and decryption strengthened my confidence in handling sensitive information safely and gave me practical skills I can use in real-life digital communication.

Chapter 3

Sending Encrypted Email

After learning about GPG and encryption I also explored how to send encrypted emails. At first the process seemed complicated but I realized that using open-source tools makes it much easier. I used Thunderbird a free email client along with the Enigmail add-on to enable encryption.

The process involved creating my GPG key pair, sharing my public key with the recipient and importing their public key into Thunderbird. Once the keys were set up I could compose an email and encrypt it so that only the intended recipient could read it. Similarly I could receive encrypted emails and decrypt them using my private key.

Some key steps I practiced included:

- Configuring Thunderbird to use your GPG keys

- Importing the recipient's public key

- Encrypting outgoing emails before sending

- Decrypting received emails using your private key

This experience helped me understand how email encryption ensures privacy and security. I also realized how open-source tools allow users to protect their communications without relying on proprietary software. Practicing encrypted emails gave me hands-on knowledge of digital security and improved my confidence in using Linux for secure communication.

Chapter 4

Privacy Tools I Explored

During this course I also explored several privacy tools recommended by Prism-Break.org. These tools focus on protecting user data, improving security and maintaining privacy while using computers or the internet. I learned that using open-source tools can give users more control over their digital life. Some of the tools I found useful include:

1. Signal

Secure messaging application providing end-to-end encryption for chats and calls.

2. Tor Browser

Web browser for anonymous browsing, routing traffic through the Tor network to protect privacy.

3. ProtonMail

Email service that provides automatic encryption, keeping emails private and secure.

4. VeraCrypt

Tool for encrypting files, folders, or entire disks to protect sensitive data.

5. Nextcloud

Self-hosted cloud storage platform that allows private file storage and sharing, giving full control to the user.

Using these tools helped me understand the importance of digital privacy. I also realized that open-source software is a powerful way to keep data safe while giving users freedom and transparency that proprietary software often doesn't provide.

Chapter 5

Open Source License I Used

For my open-source work I used the GNU General Public License version 3 (GPL-3.0). This is a widely used license that allows anyone to use, modify and distribute software freely as long as the same license is applied to any derivative works. One of the main ideas behind GPL-3.0 is to ensure that the software remains free and open for everyone and any improvements made by others are also shared openly.

By using this license I learned how open-source licensing works and why it is important for protecting both the developer's rights and the community's freedom. It ensures transparency, encourages collaboration and allows other people to contribute to the project while keeping the software free and accessible. This experience helped me understand the legal and ethical side of open-source development

Chapter 6

Self-Hosted Server Setup

During this course I learned how to self-host a server on my own laptop which was completely new to me at first. Our project involved hosting the Passky server, an open-source password manager using code from GitHub. I worked together with my teammate Subash and we followed the step-by-step instructions provided in the repository.

- Installing Ubuntu on our laptops as the base operating system.

- Installing required software like Apache/Nginx, MariaDB/MySQL, and PHP to run the server.

- Cloning the GitHub repository of Passky and reading the `CONTRIBUTING.md` and setup instructions carefully.

- Configuring the server by following the installation guide, setting up databases, and creating admin accounts.

- Testing the server locally to make sure it worked correctly and could be accessed via the assigned IP address.

During this process I faced many challenges. Sometimes dependencies failed or configuration errors occurred and at one point I even had to reinstall Ubuntu to fix an issue. However by following the documentation carefully and retrying multiple times I learned how a local server works, how to troubleshoot errors and how open-source projects can be deployed.

Finally after several attempts both Subash and I successfully hosted the Passky server on our laptops. We tested saving and sharing passwords through our server and confirmed that it worked perfectly. This experience taught me not only how to self-host applications but also how valuable open-source code and collaboration can be in building real-world projects.

After this I accessed the local server in my browser created an account and tested password storage.

Chapter 7

My Open Source Contributions (PRs)

During this course I had the opportunity to contribute to **open-source projects on GitHub** by fixing beginner-friendly issues and raising **pull requests (PRs)**. These contributions helped me understand how collaborative development works in the open-source community and gave me practical experience in troubleshooting, coding, and following contribution guidelines.

PR 1 – Multipass macOS Uninstall Script Fix

In this PR, I fixed an issue in the **macOS uninstall script** of the Multipass project. The `multipass delete` command was failing due to authentication errors when run as root. I modified the script to run the command as the **currently logged-in user** using `sudo -u "$(logname)"`.

Steps I followed:

- Identified the issue by testing the uninstall script on macOS.

- Opened a related GitHub issue to describe the problem (#4484).

- Edited the script to run the deletion command under the logged-in user.

- Tested the script locally to confirm the fix worked.

- Submitted a PR (#4483), responded to code reviews, and signed the Canonical Contributor License Agreement (CLA).

This PR was successfully merged, and it taught me how to debug scripts, follow contribution protocols, and collaborate with maintainers.

PR 2 – GovDirectory Head of Institution Fallback Fix

For the GovDirectory project, I improved how the **head of an institution** is displayed when certain data is missing. I added fallback properties and updated the queries to handle missing information gracefully.

Steps I followed:

- Identified missing fallback logic in the queries.

- Added additional properties like chair, CEO, director, and secretary-general.

- Tested the changes locally on multiple country entries.

- Submitted the PR and addressed review comments by removing redundant COALESCE logic as requested by reviewers.

This contribution helped me understand how to work with queries, handle edge cases, and incorporate feedback from project maintainers.

PR 3 – Translating Image Alt Text in FirstContributions

In this PR, I translated **English image alt text into Latin** in the `README.la.md` file to improve accessibility.

Steps I followed:

- Identified all remaining English alt texts in the README file.

- Translated them carefully into Latin.

- Verified that changes were limited to the intended files.

- Submitted the PR and prepared it for review and merge.

This contribution taught me how open-source projects value **documentation improvements**, accessibility, and attention to detail.

Overall Experience

Through these three PRs I learned how to identify issues make effective code changes test locally follow contribution guidelines and communicate with maintainers. These experiences gave me confidence and inspired me to continue contributing to open-source projects in the future.

Chapter 8

LinkedIn Posts

Here are the links to my LinkedIn posts:

- Self Hosting Post: `https://www.linkedin.com/posts/rahul-shetty-437009368_opensource-kluniversity-foss-activity-7391051386370748416-ANbZ?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFsgPysBrzfyCRWVrMCQVf3dMoPoAUH-n3s`

- PR Merge Post: `https://www.linkedin.com/pulse/excited-share-my-open-source-contrib`

- Blog Article Post: `https://www.linkedin.com/pulse/open-source-engineering-journey-le`