



# Open Source Engineering Report

**Student Name:** BANGARU VENKATA KRISHNA MAHESH  
(2400040019)

**Branch:** B.Tech – Electronics and Communication Engineering (ECE)

**Course:** Open Source Software Engineering

**Academic Year:** 2025–2026

**Submitted To:**

*Dr. Arunekumar Bala / EL&GE / KL University*

# Contents

1	About Linux Distro Used: Ubuntu	2
2	Encryption and GPG	2
3	Sending Encrypted Email	3
4	Privacy Tools (PRISM-BREAK)	3
5	Open Source License Used – MIT	4
6	Self Hosted Server – Scribble	4
7	Open Source Contributions	7
8	LinkedIn Posts	9

# 1 About Linux Distro Used: Ubuntu

Ubuntu is one of the most popular Linux distributions used by developers, students and beginners. It is based on Debian and is known for its stability, regular updates and a friendly graphical interface. Ubuntu is widely used in software development, cloud computing and open-source learning labs.

Ubuntu provides thousands of free and open-source packages through the **apt** package manager. Using simple commands, we can install compilers, editors, servers and security tools. This makes it a very good choice for students who are just starting with Linux.

A key advantage of Ubuntu is its Long-Term Support (LTS) releases. LTS versions receive security and bug fix updates for five years, so they are trusted by companies and universities. Most major cloud platforms like AWS, Azure and Google Cloud support Ubuntu images by default.

In this course, Ubuntu helped me learn:

- Basic terminal commands for navigation and file handling
- Installing and updating software using **apt**
- Managing users, permissions and executable files
- Using Git and GitHub directly from the terminal
- Running and testing self-hosted services such as Scribble

Overall, Ubuntu gave me a strong foundation in using Linux as a development environment for open source engineering.

## 2 Encryption and GPG

GNU Privacy Guard (GPG) is a free and open-source implementation of the OpenPGP standard. It is used for encrypting files, signing data and verifying signatures. The main idea is public-key cryptography: each user has a **public key** (can be shared) and a **private key** (kept secret).

When someone wants to send us a secret message, they encrypt it with our public key. Only our private key can decrypt that message. In the same way, if we sign a file with our private key, others can verify the signature with our public key and confirm that it really came from us and has not been modified.

### Common GPG Commands

- `gpg --full-generate-key` – Generate a new key pair (public + private)
- `gpg --list-keys` – Show the public keys stored in our keyring
- `gpg --export --armor > publickey.asc` – Export our public key so that we can share it
- `gpg --encrypt --recipient <email> file.txt` – Encrypt `file.txt` for a specific user

- `gpg --decrypt file.txt.gpg` – Decrypt an encrypted file using our private key

In the lab we practised generating keys, exporting the public key and encrypting and decrypting sample files. This helped me understand how many open-source projects sign their releases and how users can verify authenticity.

## 3 Sending Encrypted Email

Normal email is like sending a postcard: anyone on the path can read the content. To protect privacy, we can combine email with GPG encryption. For this we can use tools such as Thunderbird with built-in OpenPGP support or browser plugins like Mailvelope.

### Steps for Encrypted Email

- Both sender and receiver generate their own GPG key pairs.
- Each person shares their **public key** with the other, usually as a `.asc` file or via a key server.
- In the email client, we import the other person's public key and mark it as trusted.
- While composing a mail, we select the option "Encrypt" (and optionally "Sign").
- The email body is encrypted with the recipient's public key and sent over the internet.
- The recipient opens the mail, enters their passphrase and decrypts the message using their private key.

This activity showed me how encryption is used in real life for secure communication and how public-key infrastructure works beyond theory.

## 4 Privacy Tools (PRISM-BREAK)

PRISM-BREAK is a community-driven website that lists privacy-respecting alternatives to many popular services. Its goal is to help users avoid mass surveillance and tracking by using open-source and decentralised software.

Some tools we explored are:

- **Signal** – An end-to-end encrypted messaging application for chats, voice and video calls.
- **Tor Browser** – A browser that routes traffic through multiple relays, hiding the real IP address and making tracking difficult.
- **KeePassXC** – A local password manager that stores all passwords inside an encrypted database file.
- **Jitsi Meet** – An open-source video conferencing platform that can also be self-hosted.

- **LineageOS** – A free Android-based operating system that removes bloatware and gives more control over permissions.

These examples helped me see that privacy is not only a theory topic. There are real open-source tools available for almost every daily use-case.

## 5 Open Source License Used – MIT

The MIT License is a permissive open-source license that allows:

- Free use, modification and distribution of the software
- Commercial use without extra restrictions
- Closed-source modifications and inclusion in proprietary projects
- Only requirement: include the original copyright and license notice

Because it is short and easy to understand, many open-source libraries and student projects select the MIT license. It encourages reuse and sharing while still giving credit to the original author.

## 6 Self Hosted Server – Scribble

Scribble is an open-source, self-hosted notes and knowledge management application. It is designed for personal productivity and privacy-friendly note-taking.

### Features

- Clean web interface for creating and editing notes
- Markdown support for rich text formatting
- Tagging and search to organise notes properly
- Runs completely on our own machine or server
- Can be shared on the local network so that it is accessible from multiple devices

### How I Self-Hosted Scribble

- Installed the required dependencies (Docker / Docker Compose and Git).
- Cloned the Scribble repository from GitHub to my Ubuntu system.
- Configured environment variables like port number and storage folders.
- Used `docker-compose up -d` to start the Scribble services.
- Accessed Scribble from the browser using `http://localhost:<port>`.

## Dashboard Screenshot



# OPEN SOURCE ENGINEERING



## Scribble Game

Scribble.rs is an open-source online drawing and guessing game, inspired by Skribbl.io. It allows multiple players to join rooms, take turns drawing words, and guess what others are sketching in real time through a simple web interface. When self-hosted, it enables friends, classrooms, or teams to run their own private, secure instance of the game without relying on public servers.

License: MIT License

- 🎨 Key Features of Scribble Self-Hosting Server
- Real-time multiplayer drawing & guessing: players draw and guess words together instantly in a shared game room.
- Self-hosted and private: run the server on your own machine or VPS, keeping games secure and under your control.
- Custom word lists: create your own word sets for themed games, learning activities, or inside jokes.

M Sai Nihhaar - 2400040002

B Venkata Krishna Mahesh - 2400040019

## 7 Open Source Contributions

GitHub Username: **bangarumahesh22**

During this course, I contributed to real open-source projects on GitHub. I worked on two meaningful pull requests: one in the popular **first-contributions** repository and another in the **HeyPuter/puter** project.

### List of Pull Requests

- **firstcontributions / first-contributions** – “Add bash syntax highlighting to Armenian README” (PR #106070)

In this pull request, I updated the Armenian version of the README by adding missing bash syntax highlighting. The tutorial helped me understand how Git, branching, committing, and pull requests work in practice.

This contribution improved my skills in:

- identifying documentation inconsistencies,
- making precise text/code formatting fixes,
- understanding PR review workflows,
- and communicating clearly through commit messages.

The PR was **successfully merged** by the maintainers.

- **HeyPuter / puter** – “Docs: improve clarity in README introduction” (PR #2010)

In this PR, I refined the README introduction by improving clarity and correcting minor language issues. I created a separate branch, made the documentation changes, and opened a PR to the main project.

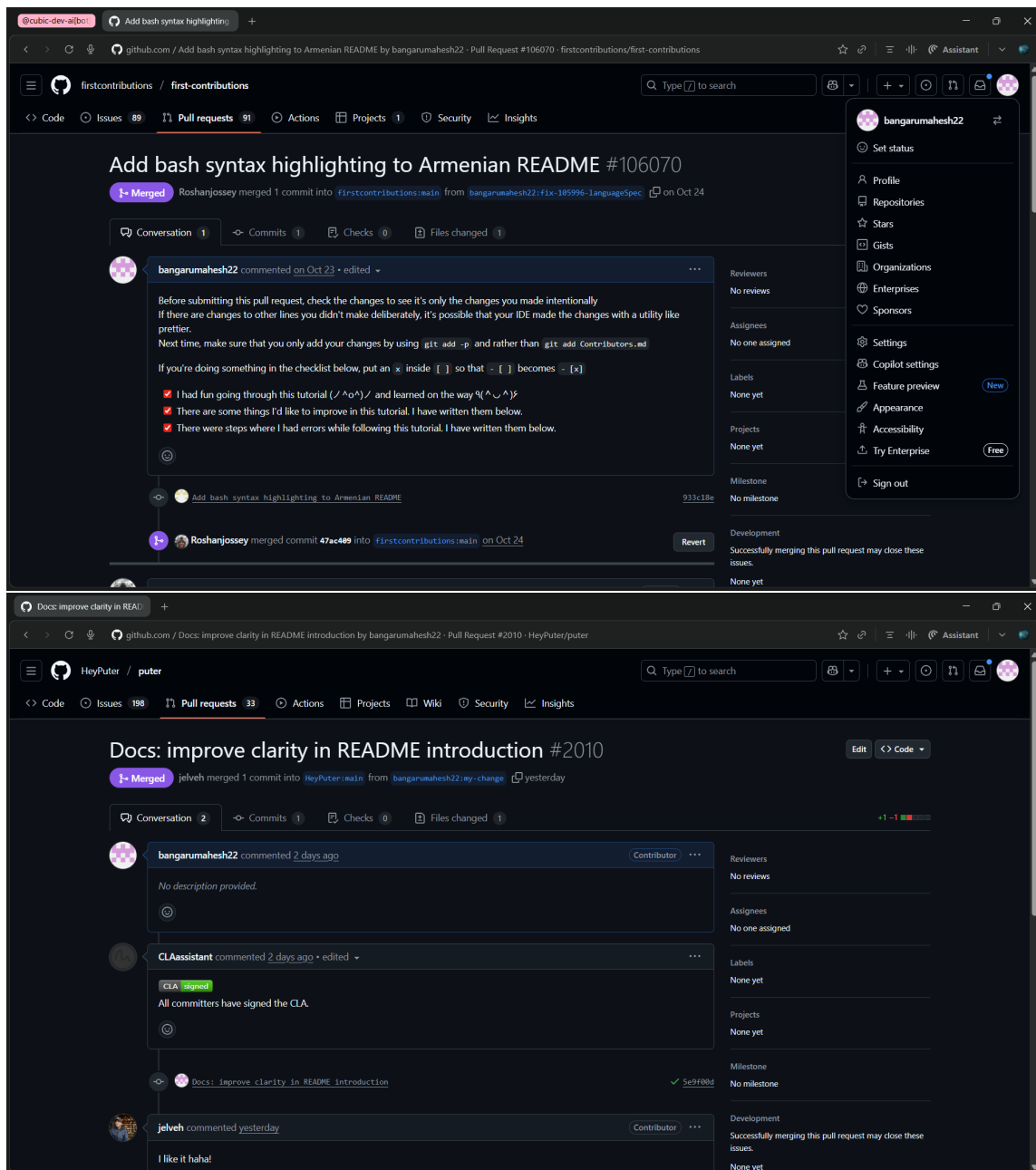
This contribution helped me learn:

- how to work with large repositories,
- how maintainers review documentation PRs,
- how CLA (Contributor License Agreement) automation works,
- and how small documentation improvements enhance user experience.

This PR was also **merged**, showing that my contribution met the maintainers’ quality expectations.



# Screenshots



## 8 LinkedIn Posts

I have shared my open-source journey on LinkedIn. These posts helped me explain my learning and connect with other developers.

- Post about my first merged PR in first-contributions –  
[https://www.linkedin.com/posts/bangaru-venkata-krishna-mahesh-88a441357\\_opensource-hacktoberfest-firstcontributions-activity-7399009303531012096-Ew\\_f?utm\\_source=social\\_share\\_send&utm\\_medium=member\\_desktop\\_web&rcm=ACoAAFjgL6oBe6lPBfQN-4pfDMUEpOLpmp\\_JnHM](https://www.linkedin.com/posts/bangaru-venkata-krishna-mahesh-88a441357_opensource-hacktoberfest-firstcontributions-activity-7399009303531012096-Ew_f?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAAFjgL6oBe6lPBfQN-4pfDMUEpOLpmp_JnHM)
- Post about self-hosting Scribble –  
[https://www.linkedin.com/posts/bangaru-venkata-krishna-mahesh-88a441357\\_opensource-foss-selfhosted-activity-7390934243679318016-4IQT?utm\\_source=share&utm\\_medium=member\\_desktop&rcm=ACoAAFjgL6oBe6lPBfQN-4pfDMUEpOLpmp\\_JnHM](https://www.linkedin.com/posts/bangaru-venkata-krishna-mahesh-88a441357_opensource-foss-selfhosted-activity-7390934243679318016-4IQT?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFjgL6oBe6lPBfQN-4pfDMUEpOLpmp_JnHM)