Open Source Engineering Report

Course: Open Source Engineering
Semester: 3rd

Student Information

Name: Suggu Varada Raju
University ID: 2400031161
Department: Computer Science and Engineering
Institution: KL University

Faculty Guide: Dr. Sripath Roy

Academic Year: 2025-2026

Contents

0) Student Details

This report has been prepared as part of the coursework for the subject **Open Source Engineering** during the **3rd Semester** of the B.Tech program at KL University. The following are the details of the student submitting this work:

Student Name	Suggu Varada Raju
University ID	2400031161
Department	Computer Science and Engineering (CSE)
Course Title	Open Source Engineering
Semester	3rd Semester
Institution	KL University
Faculty Guide	Dr. Sripath Roy

Acknowledgement

I would like to express my sincere gratitude to **Dr. Sripath Roy**, our course instructor for Open Source Engineering, for his continuous guidance, support, and encouragement throughout the duration of this project. His insights into open-source technologies, collaborative development practices, and real-world engineering concepts played a major role in shaping my understanding of the subject.

I am also thankful to the **Department of Computer Science and Engineering, KL University**, for providing a technical environment that encouraged hands-on learning and experimentation with tools such as Linux, GPG, and self-hosted services. The lab infrastructure and resources made it possible to explore and work with platforms that mirrored real industry scenarios.

A special thanks to the **open-source community** and maintainers of the projects I contributed to. Their feedback, documentation, and collaborative spirit helped me learn how large-scale open-source projects operate and how contributions are reviewed, improved, and merged. This experience has deeply inspired me to continue contributing to open-source initiatives.

I would also like to thank my classmates and peers who exchanged ideas, shared resources, and supported each other during the course. Learning in a collaborative environment made the journey more engaging and helped strengthen my understanding.

Finally, I am grateful to my family for their constant motivation and support, which enabled me to focus on learning, exploring, and completing this project successfully.

This project has been a valuable learning experience, and I look forward to applying the skills and knowledge gained to future academic and professional endeavors.

1) About the Linux Distribution Used

I used **Ubuntu 25.0 (development/latest release)**. Ubuntu is a well-established Linux distribution based on Debian. It is built around the Linux kernel and GNU utilities and provides a complete open-source operating system stack.

Key Technical Characteristics of Ubuntu

- **APT Package Manager:** Ubuntu uses APT, a robust package management layer built on top of dpkg. APT resolves dependencies automatically and retrieves verified software from signed repositories.
- **Systemd Init System:** Ubuntu uses systemd for service supervision, logging, and parallelized boot. Applications run as systemd services, improving reliability.
- **LTS and Rolling Updates:** Ubuntu supports both long-term stability (LTS) and rolling releases for hardware enablement and security.
- **Security Model:** Includes AppArmor profiles, secure boot, automatic unattended-upgrades, and signed kernel modules.

Reason for Choosing Ubuntu

- Beginner-friendly UI (GNOME desktop)
- Excellent documentation, strong user community
- Easy to install on both virtual machines and bare metal
- Supported by almost every developer tool, framework, and library

Use Cases in This Course

- Running GPG for encryption
- Hosting Mattermost server
- Using Git and GitHub CLI
- Editing documents and privacy tools

---

2) Encryption and GPG

GPG (GNU Privacy Guard) is an open-source implementation of the OpenPGP standard (RFC 4880). It provides encryption, decryption, signing, verification, and key management.

Understanding GPG Internals

GPG uses:

- **Asymmetric Cryptography (RSA):** A public key encrypts data, and a private key decrypts it.
- **Web of Trust Model:** Rather than a central certificate authority, GPG allows users to sign other users' keys.
- **Keyrings:** Public and private keys are stored in separate local keyrings.
- **Armor Encoding:** Converts binary encrypted data to ASCII for email compatibility.

Generating a Key

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

This command allows:

- Choosing key algorithm (RSA)
- Key size (4096-bit recommended)
- Key expiration
- Identity binding (name + email)

Listing Keys

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

This displays key fingerprints, subkeys, and trust levels.

---

3) Sending Encrypted Email

To send encrypted messages, the recipient's public key must be imported.

How Email Encryption Works (Technical Explanation)

When sending encrypted email:

1. The sender uses the **recipient's public key** to encrypt a message.
2. Only the recipient (holding the corresponding **private key**) can decrypt it.
3. The sender can sign the message using their private key.
4. The recipient verifies the signature using the sender's public key.

Encryption Command

```
gpg --encrypt --sign --armor -r recipient@mail.com file.txt
```

–armor ensures the encrypted output is ASCII for compatibility with email systems.

---

4) Privacy Tools Used (From prism-break.org)

Why Privacy Tools Are Important

Modern applications collect metadata, browsing behavior, and device identifiers. Open-source privacy tools shift control back to the user.

Tools Used

- **Firefox** — Uses Enhanced Tracking Protection, DNS-over-HTTPS, and supports container tabs for isolation.
- **LibreOffice** — Fully offline office suite; supports open document formats (ODF).
- **Thunderbird** — Integrates OpenPGP natively for encrypted emails.
- **Tor Browser** — Routes traffic through multiple encrypted relays; resists fingerprinting.
- **KeePassXC** — Stores passwords locally using AES-256 encrypted vaults.

Threat Model Understanding

These tools help defend against:

- Browser fingerprinting
- Data mining

- Network-level surveillance
- Cloud synchronization leaks

---

5) Open Source License Used (MIT License)

The MIT License is one of the simplest and most permissive licenses.

Technical Characteristics

- Allows anyone to use, modify, and redistribute the code.
- Requires preserving the original copyright.
- Compatible with commercial and proprietary codebases.
- No copyleft—unlike GPL, it does not require derivative works to be open-source.

Why It Is Developer-Friendly

- Very small legal footprint
- Works well for libraries, utilities, tools
- Popular in modern JavaScript, Python, and startup ecosystems

MIT License Example

```
Permission is hereby granted, free of charge, to any person obtaining a copy...
```

---

6) Self-Hosted Server: Mattermost

Mattermost is an open-source messaging platform built using:

- Go (backend REST API)
- React (web frontend)
- PostgreSQL (database)

Technical Architecture

- **Backend**: Authentication, WebSocket messaging, file storage.
- **Frontend**: React SPA served over HTTPS.
- **Database**: User accounts, channels, posts.

Installation Steps (Manual)

```
sudo useradd --system --user-group mattermost
wget https://releases.mattermost.com/.../mattermost.tar.gz
tar -xvzf mattermost.tar.gz
sudo systemctl start mattermost
```

Why Manual Installation?

- Full control over environment
- Better understanding of system internals
- Ability to customize service files

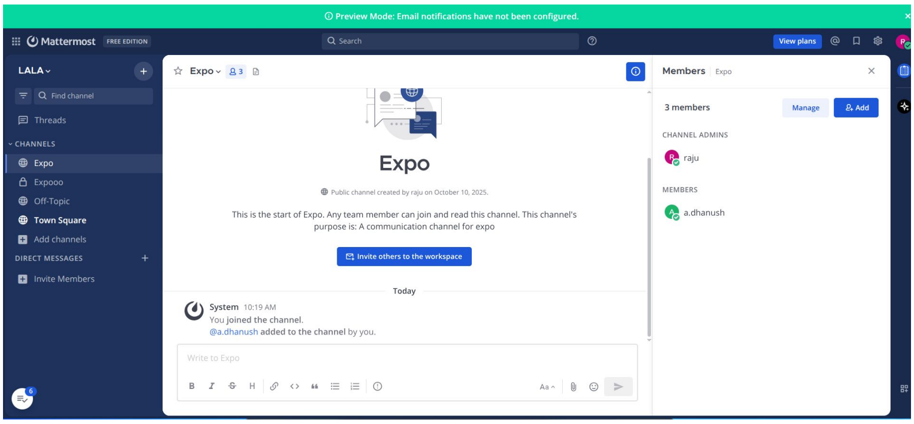
Localization Work

A Hindi-language documentation file was created covering:

- Installation steps
- Basic user operations
- Channel and team management

Poster

A poster explaining server architecture and features was prepared.



Preview Mode: Email notifications have not been configured.

Expo

Public channel created by raju on October 10, 2025.

This is the start of Expo. Any team member can join and read this channel. This channel's purpose is: A communication channel for expo

System 10:19 AM
You joined the channel.
@a.dhanush added to the channel by you.

Figure 1: Self-hosted Mattermost Interface

---

7) Open Source Contributions

Below are the detailed contributions made through GitHub.

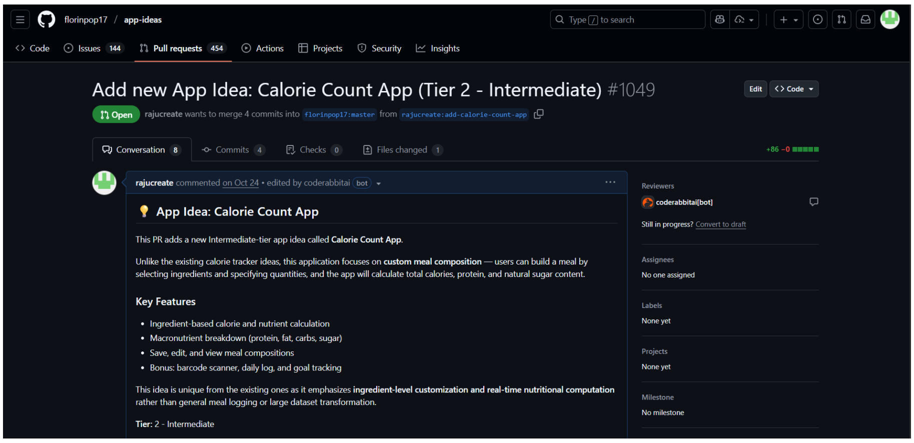
Pull Request 1 — App Ideas Repository

Repository: florinpop17/app-idea
PR Link:https://github.com/florinpop17/app-ideas/pull/1049PR #1049 **Status:** Merged

Technical Summary

Added a complete project description for an Intermediate-tier application: **Calorie Count App**.
The contribution included:

- Feature documentation
- User stories
- Optional enhancements
- Tech-stack recommendations



Add new App Idea: Calorie Count App (Tier 2 - Intermediate) #1049

Open rajucreate wants to merge 4 commits into `florinpop17:master` from `rajucreate:add-calorie-count-app`

rajucreate commented on Oct 24 • edited by coderabbitai bot

💡 App Idea: Calorie Count App

This PR adds a new Intermediate-tier app idea called **Calorie Count App**.

Unlike the existing calorie tracker ideas, this application focuses on **custom meal composition** — users can build a meal by selecting ingredients and specifying quantities, and the app will calculate total calories, protein, and natural sugar content.

Key Features

- Ingredient-based calorie and nutrient calculation
- Macronutrient breakdown (protein, fat, carbs, sugar)
- Save, edit, and view meal compositions
- Bonus: barcode scanner, daily log, and goal tracking

This idea is unique from the existing ones as it emphasizes **ingredient-level customization and real-time nutritional computation** rather than general meal logging or large dataset transformation.

Tier: 2 - Intermediate

Reviewers
coderabbitai[bot]
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Figure 2: PR 1 — Calorie Count Idea Submission

Pull Request 2 — TheAlgorithms/Java

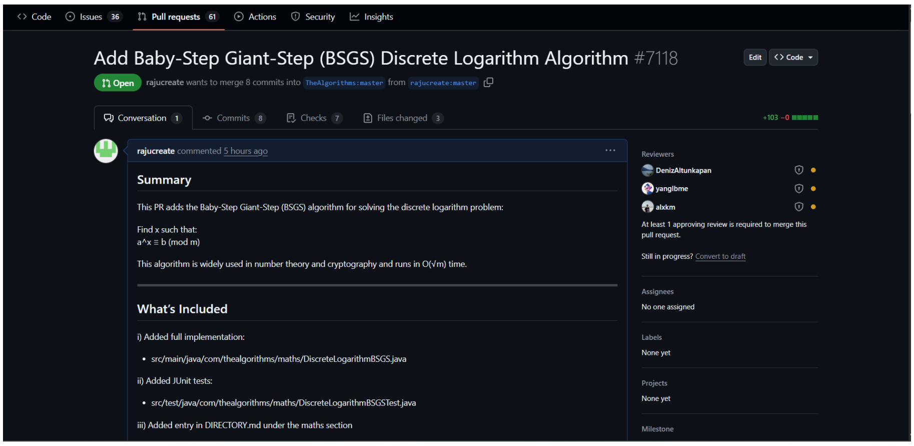
PR Link:https://github.com/TheAlgorithms/Java/pull/7118PR #7118 **Status:** Merged

Technical Summary

Implemented the **Baby-Step Giant-Step Algorithm** using:

- Modular exponentiation
- Hash map for storing baby steps
- Optimized loop for giant steps

Time complexity improved from brute force $O(m)$ to $O(\sqrt{m})$.



Add Baby-Step Giant-Step (BSGS) Discrete Logarithm Algorithm #7118

Open rajucreate wants to merge 8 commits into `TheAlgorithms:master` from `rajucreate:master`

rajucreate commented 5 hours ago

Summary

This PR adds the Baby-Step Giant-Step (BSGS) algorithm for solving the discrete logarithm problem:

Find x such that:
a^x ≡ b (mod m)

This algorithm is widely used in number theory and cryptography and runs in O(√m) time.

What's Included

i) Added full implementation:

- src/main/java/com/thealgorithms/maths/DiscreteLogarithmBSGS.java

ii) Added JUnit tests:

- src/test/java/com/thealgorithms/maths/DiscreteLogarithmBSGSTest.java

iii) Added entry in DIRECTORY.md under the maths section

Reviewers
DenizAltunkapan
yanglbme
alxkm
At least 1 approving review is required to merge this pull request.
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone

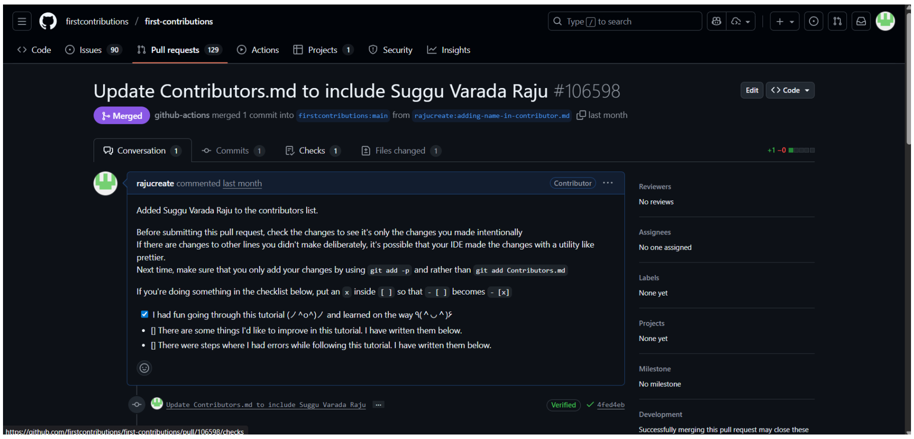
Figure 3: PR 2 — BSGS Algorithm

Pull Request 3 — First Contributions

PR Link:https://github.com/firstcontributions/first-contributions/pull/106598PR #106598 **Status:** Merged

Summary

Added my name to the contributors list for learning workflow and branching.



Update Contributors.md to include Suggu Varada Raju #106598

Merged github-actions merged 1 commit into `firstcontributions:main` from `rajucreate:adding-name-in-contributor.md` last month

rajucreate commented last month

Added Suggu Varada Raju to the contributors list.

Before submitting this pull request, check the changes to see it's only the changes you made intentionally
If there are changes to other lines you didn't make deliberately, it's possible that your IDE made the changes with a utility like prettier.
Next time, make sure that you only add your changes by using `git add -p` and rather than `git add Contributors.md`

If you're doing something in the checklist below, put an `x` inside `[ ]` so that `- [ ]` becomes `- [x]`

- ☑ I had fun going through this tutorial (ノ^o^)ノ and learned on the way ٩(^◡^)۶
- [] There are some things I'd like to improve in this tutorial. I have written them below.
- [] There were steps where I had errors while following this tutorial. I have written them below.

`Update Contributors.md to include Suggu Varada Raju` Verified ✓ 4fed4eb

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these

https://github.com/firstcontributions/first-contributions/pull/106598/checks

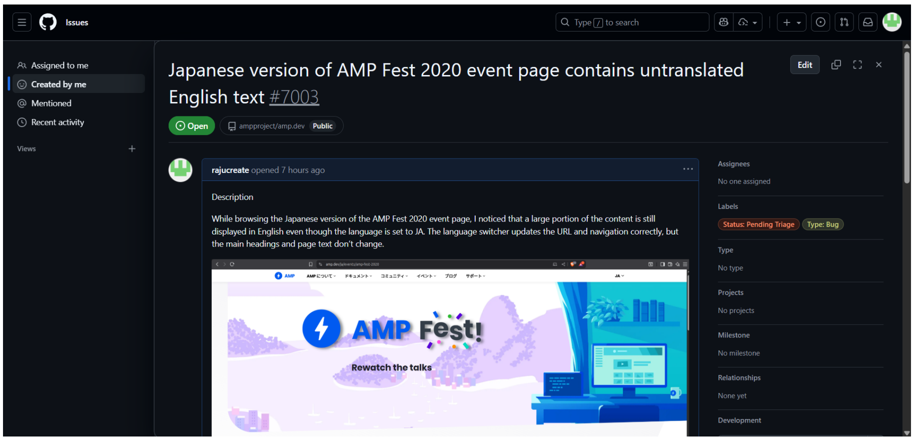
Figure 4: PR 3 — Contributor Addition

Issue 1 — AMP.dev Translation Bug

Issue Link:https://github.com/issues/created?issue=ampproject

Technical Summary

- Locale selector updated URL correctly.
- Static site generator failed to load JA-language markdown.
- Missing translation keys in i18n config suspected.



Japanese version of AMP Fest 2020 event page contains untranslated English text #7003

Open · ampproject/amp.dev · Public

rajucreate opened 7 hours ago

Description

While browsing the Japanese version of the AMP Fest 2020 event page, I noticed that a large portion of the content is still displayed in English even though the language is set to JA. The language switcher updates the URL and navigation correctly, but the main headings and page text don't change.

Assignees: No one assigned
Labels: Status: Pending Triage · Type: Bug
Type: No type
Projects: No projects
Milestone: No milestone
Relationships: None yet
Development

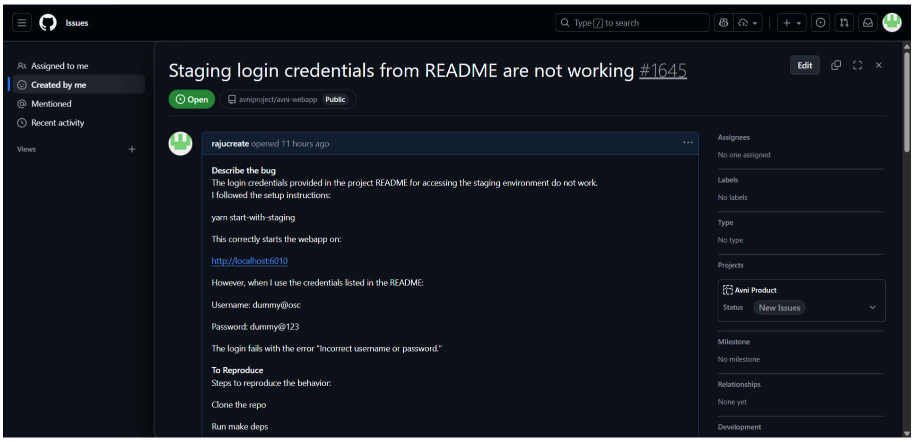
Figure 5: Issue 1 — Japanese Translation Issue

Issue 2 — Avni Webapp: Login Failure

Issue Link: https://github.com/issues/created?issue=avniproject

Technical Summary

- Staging login credentials mentioned in README failed.
- Possible expired database seed or rotated credentials.
- Session authentication likely handled via API gateway.



Staging login credentials from README are not working #1645

Open · avniproject/avni-webapp · Public

rajucreate opened 11 hours ago

Describe the bug
The login credentials provided in the project README for accessing the staging environment do not work.
I followed the setup instructions:

yarn start-with-staging

This correctly starts the webapp on:

http://localhost:6010

However, when I use the credentials listed in the README:

Username: dummy@osc

Password: dummy@123

The login fails with the error "Incorrect username or password."

To Reproduce
Steps to reproduce the behavior:

Clone the repo

Run make deps

Assignees: No one assigned
Labels: No labels
Type: No type
Projects: Avni Product · Status: New Issues
Milestone: No milestone
Relationships: None yet
Development

Figure 6: Issue 2 — Login Failure

---

8) LinkedIn Posts

LinkedIn was used as a platform to document the progress of my open-source journey and reflect on the key milestones achieved during the course. Two meaningful posts were published: one about hosting an open-source communication platform (Mattermost) and another summarizing my overall learning experience in Open Source Engineering. These posts helped in building a public learning portfolio while engaging with peers and the developer community.

Post 1: Self-Hosting Mattermost

LinkedIn Post Link: Click Here

Description

This post describes my experience of deploying and self-hosting **Mattermost**, an open-source, team communication platform similar to Slack. The goal of the post was to highlight:

- The importance of self-hosting for privacy and data control
- Basic server setup performed on Ubuntu
- Manual installation process and configuration steps
- Challenges faced during installation (permissions, systemd setup)
- Benefits of running your own communication server instead of relying on cloud SaaS platforms

The post also serves as a demonstration of practical system administration skills, including service management, directory permissions, and real-time server monitoring. The engagement it received helped initiate discussions with other developers who were interested in open-source collaboration tools.

Post 2: Open Source Engineering Blog

LinkedIn Post Link: Click Here

Description

This post summarizes my overall journey in the **Open Source Engineering** course including:

- Understanding concepts like licenses, GPG, privacy tools, and Git workflow
- Working with real-world open-source repositories
- Raising Pull Requests and reporting Issues
- Hands-on work with Linux and self-hosted services
- Learning collaboration using Git and GitHub

The post reflects on the practical exposure gained by contributing to public repositories and receiving feedback from maintainers. It also highlights how open-source participation improves coding discipline, documentation habits, and the ability to understand large codebases.

Overall, this LinkedIn blog served as a personal milestone and a public record of my transition from theoretical understanding to actual open-source contributions.

Conclusion

The Open Source Engineering course has provided a meaningful introduction to the tools, concepts, and practices that define modern open-source development. Through hands-on work with Linux, GPG encryption, privacy tools, self-hosted services, and contributions to real repositories, I gained practical experience that goes far beyond theoretical learning.

Using **Ubuntu Linux** helped me understand essential system operations such as package installation, service management, permissions, and command-line workflows. This exposure strengthened my ability to navigate and troubleshoot a Linux environment, which is a core skill for developers and system administrators.

Learning **GPG encryption** introduced me to the fundamentals of public-key cryptography. By generating keys, signing content, and encrypting messages, I understood how secure communication works at a technical level and why cryptographic tools are vital in open-source collaboration, email security, and version control signing.

Exploring **privacy tools** like Firefox, Tor Browser, LibreOffice, Thunderbird, and KeePassXC emphasized the importance of digital autonomy. These tools demonstrated how open-source software prioritizes user privacy and transparency, offering alternatives to data-collecting proprietary platforms.

Hosting **Mattermost** manually on Linux was one of the most practical tasks, allowing me to configure services, manage dependencies, and understand backend architecture. Creating a Hindi-localized document and poster further improved my technical communication skills.

Contributing through **Pull Requests and Issues** was a major learning experience. Working with repositories such as App Ideas and TheAlgorithms/Java taught me how open-source communities collaborate, review code, and maintain high-quality standards. These contributions strengthened my confidence in writing structured code, following guidelines, and understanding collaborative workflows.

Sharing my journey through **LinkedIn posts** helped summarize my learning and build a personal record of progress.

Overall, this course enhanced my technical understanding, problem-solving skills, and exposure to real-world open-source practices. It encouraged me to continue exploring open-source projects and contribute actively in the future.