(DEEMED TO BE UNIVERSITY) HTE HONORS THROUGH EXPERIENTIAL LEARNING

EXPERIENTIAL LEARNING & GLOBAL ENGAGEMENT

Open Source Engineering Report

Student Name: G Nandan Sameer Reddy (2400030343)

Branch: B.Tech – Computer Science Engineering (HTE)

Course: Open Source 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 – AGPLv3 4

6 Self Hosted Server – HedgeDoc 4

7 Open Source Contributions 6

8 LinkedIn Posts 10

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 HedgeDoc

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:

- **Brave Browser** – A privacy-focused web browser that blocks ads and trackers by default, supports Tor integration, and rewards users with cryptocurrency for opting into privacy-respecting ads.
- **ProtonMail** – An end-to-end encrypted email service based in Switzerland, offering zero-access encryption where even the service provider cannot read user emails.
- **Bitwarden** – An open-source password manager that stores credentials in an encrypted vault, supports self-hosting, and offers cross-platform synchronization.

- **Nextcloud** – A self-hosted cloud storage and collaboration platform that provides file sync, calendar, contacts, and document editing as alternatives to Google Drive and Dropbox.
- **Element (Matrix)** – A decentralized, end-to-end encrypted messaging platform based on the Matrix protocol, allowing users to communicate across different servers.

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 – AGPLv3

The GNU Affero General Public License version 3 (AGPLv3) is a strong copyleft open-source license designed to ensure maximum software freedom and transparency. It extends the principles of the GPL to network-based applications.

- Allows anyone to freely use, study, modify, and distribute the software
- Requires that any modified version must also be released under the same AGPLv3 license
- Ensures that users who interact with the software over a network can access its complete source code
- Prevents incorporation of modified versions into closed-source or proprietary sys- tems
- Promotes openness, collaboration, and community-driven development

AGPLv3 is commonly chosen for web applications where maintaining openness of server-side code is important.

6 Self Hosted Server – HedgeDoc

HedgeDoc is an open-source, real-time collaborative Markdown editor. It is widely used for documentation, note-sharing, meeting notes, and team collaboration.

Features

- Real-time collaborative editing with multiple users
- Clean and powerful Markdown editor with live preview
- Supports embedding diagrams, charts, videos, and rich media
- Easy sharing using unique URLs
- Can be accessed from any device on the same network

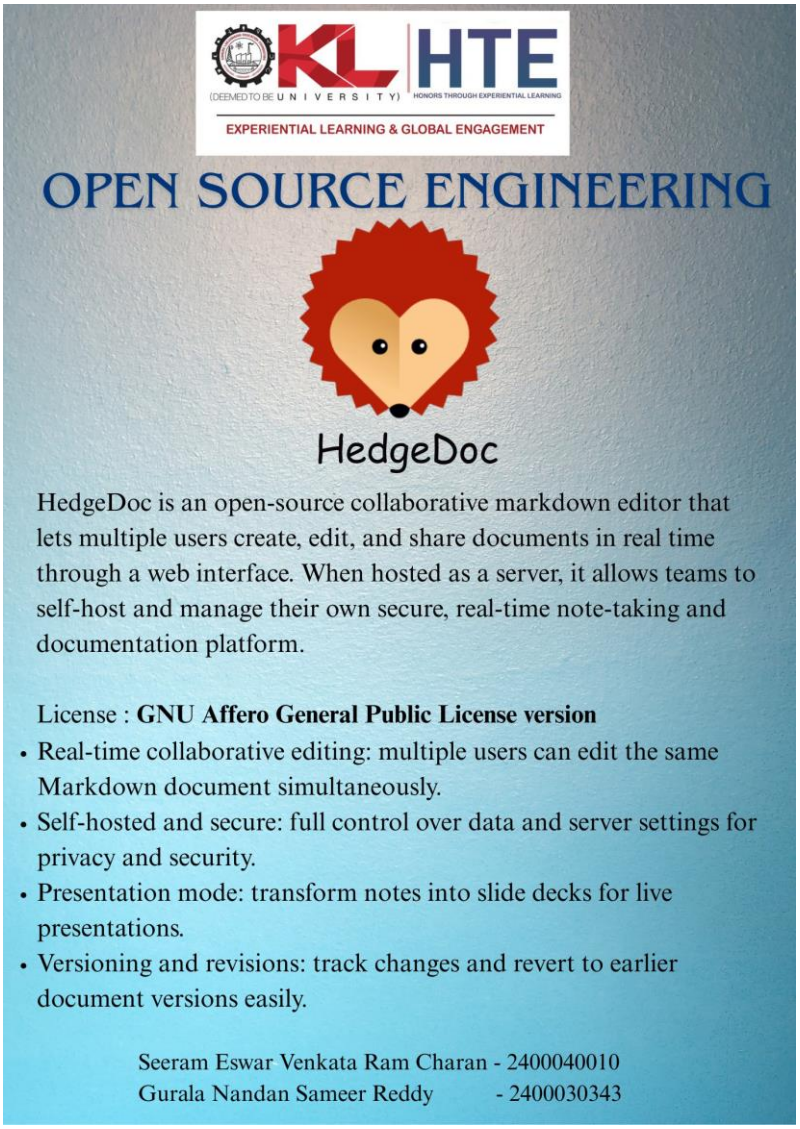
How I Self-Hosted HedgeDoc

- Installed Docker and Docker Compose on my Ubuntu system.
- Created and configured the docker-compose.yml file with necessary environment variables such as:
  - CMD_ DOMAIN for domain/localhost
  - CMD PORT to set the service port
- Started the HedgeDoc service using docker-compose up -d.
- Accessed HedgeDoc through the browser via http://localhost:<port>.

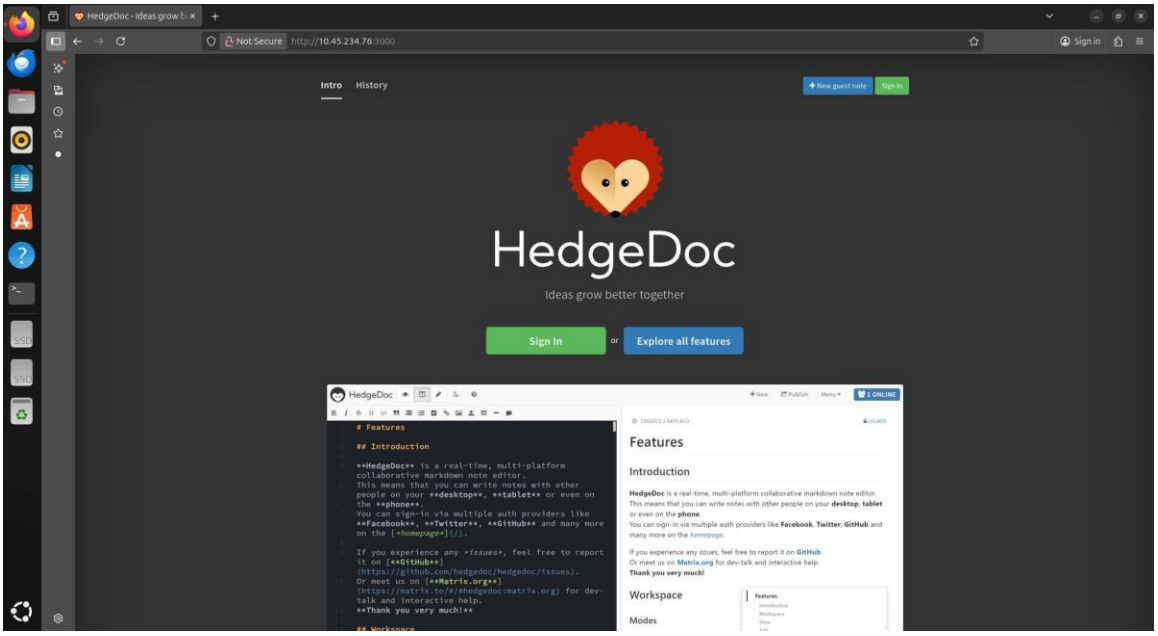
Localized(Translated) Document

https://drive.google.com/file/d/1MXE12sFSqHrETCQ80DFTkMw-1vdpkqE4/view?usp=sharing

Poster



Server Screenshot



7 Open Source Contributions

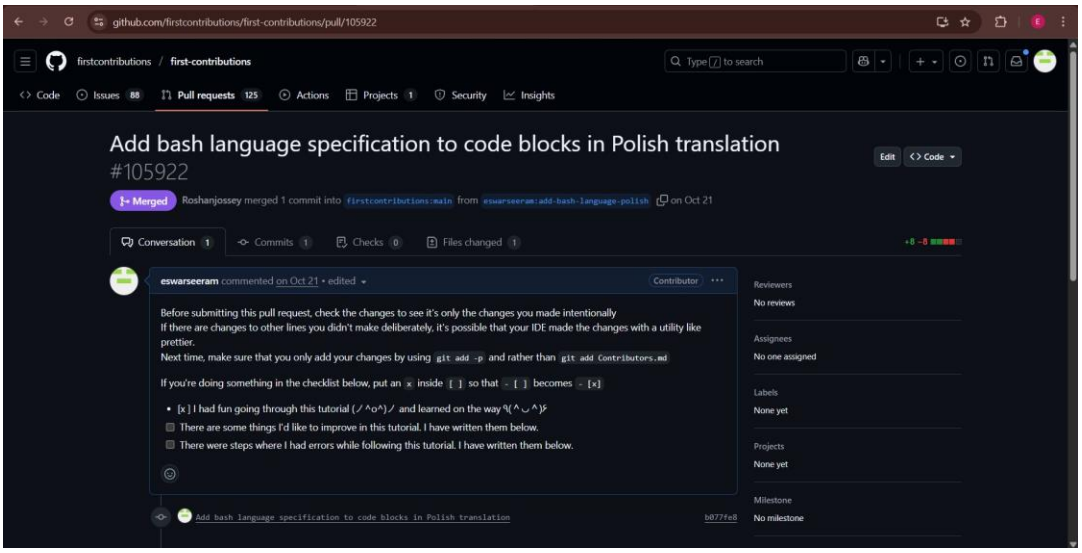
GitHub Username: **2400030343SameerReddy**

In this course we were asked to contribute to real open-source projects. The following is a list of my successfully merged pull requests:

List of Pull Requests

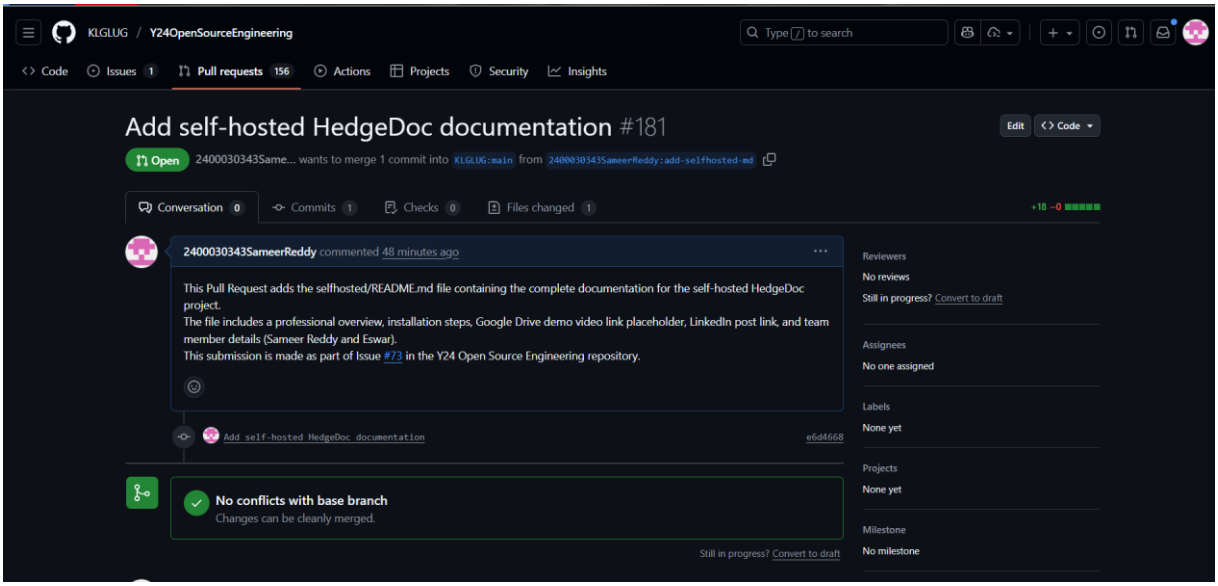
- **PR1: firstcontributions / first-contributions** – "Add bash language specification to code blocks in Polish translation"

  This pull request added bash language specification to code blocks in the Polish translation file of the "first-contributions" repository. This fix improves syntax highlighting and clarity for bash commands in documentation, making it more readable and easier to follow for contributors. The PR was reviewed, approved, and successfully merged.



- **PR2: KLGLUG / Y24OpenSourceEngineering** – "Added HedgeDoc self-hosted project documentation in Telugu"

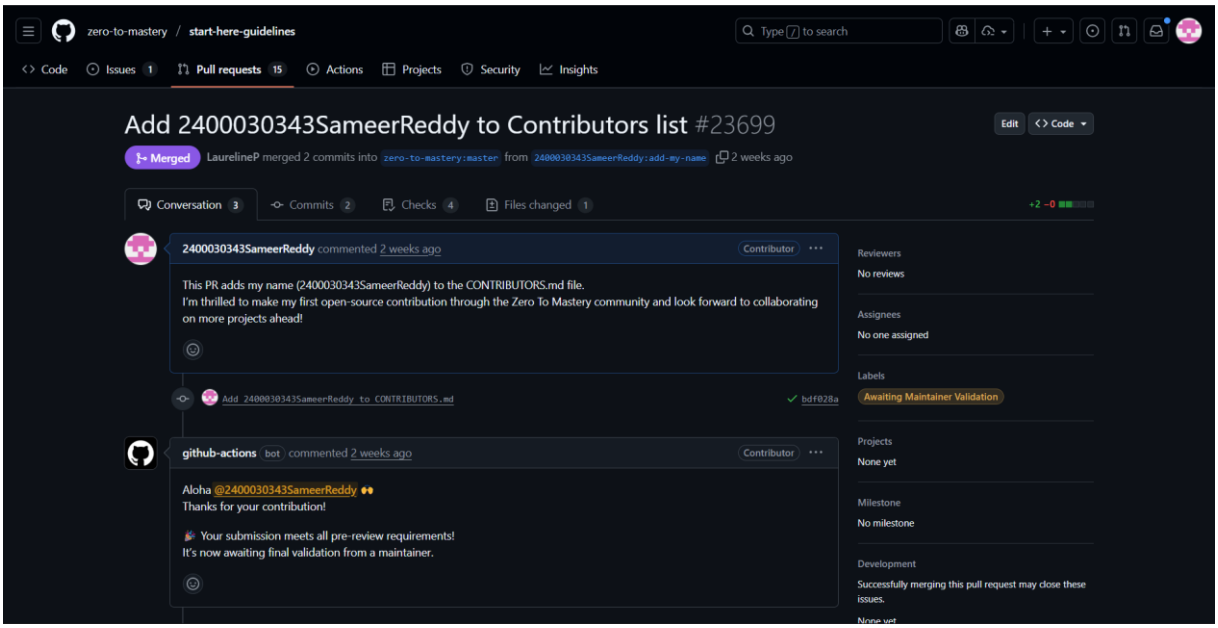
  In this pull request, I wrote detailed documentation in Telugu explaining how to self-host HedgeDoc, along with a video demonstrating the installation steps through screen recording with voiceover in the local language. My aim was to help Telugu-speaking students easily understand the steps and try self-hosting on their own systems.

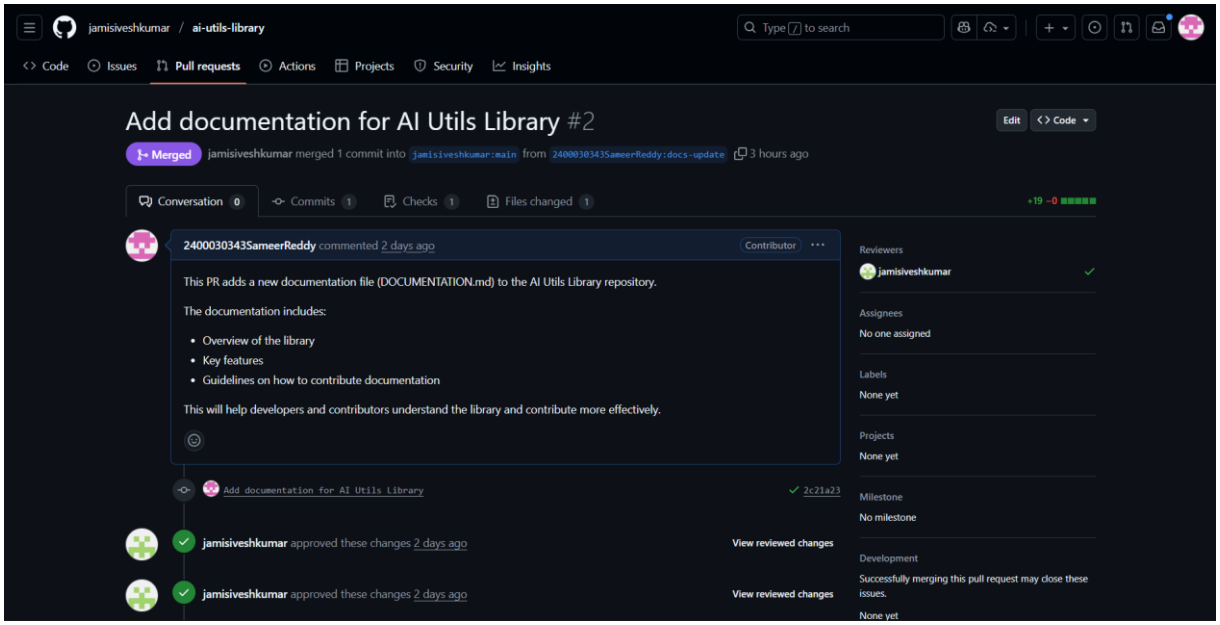


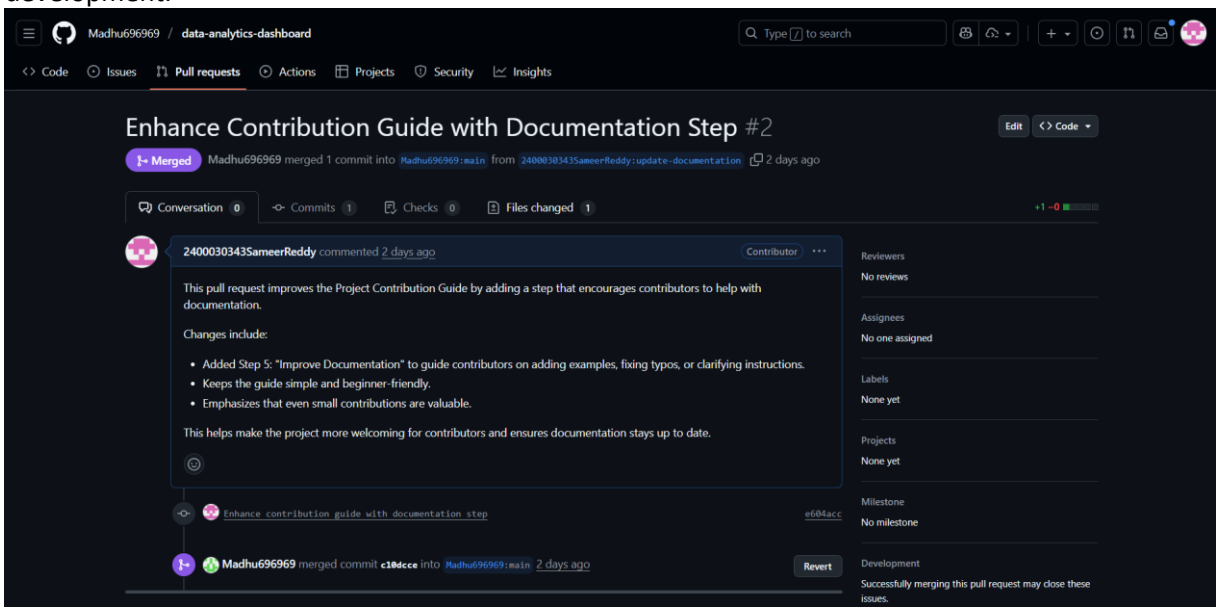
PR3:

Zero To Mastery

The **Zero To Mastery** repository is a comprehensive, beginner-friendly learning hub designed to help developers progress from foundational concepts to advanced real-world skills. It provides structured roadmaps, curated resources, hands-on projects, and best-practice examples that enable learners to build a strong understanding of modern technologies. By contributing to this repository, I aim to enhance the learning experience with clearer explanations, improved documentation, and additional practical insights that support developers in mastering topics efficiently and confidently.


zero-to-mastery / start-here-guidelines
Type / to search
Code
Issues 1
Pull requests 15
Actions
Projects
Security
Insights
Add 2400030343SameerReddy to Contributors list #23699
Edit
Code
Merged
LaurelineP merged 2 commits into zero-to-mastery:master from 2400030343SameerReddy:add-my-name 2 weeks ago
Conversation 3
Commits 2
Checks 4
Files changed 1
+2 -0
2400030343SameerReddy commented 2 weeks ago
Contributor
This PR adds my name (2400030343SameerReddy) to the CONTRIBUTORS.md file.
I'm thrilled to make my first open-source contribution through the Zero To Mastery community and look forward to collaborating on more projects ahead!
Add 2400030343SameerReddy to CONTRIBUTORS.md
bdf028a
github-actions bot commented 2 weeks ago
Contributor
Aloha @2400030343SameerReddy
Thanks for your contribution!
Your submission meets all pre-review requirements!
It's now awaiting final validation from a maintainer.
Reviewers
No reviews
Assignees
No one assigned
Labels
Awaiting Maintainer Validation
Projects
None yet
Milestone
No milestone
Development
Successfully merging this pull request may close these issues.
None yet

- **PR4:** This pull request adds clear and structured documentation for the AI Utils Library, making it easier for developers to understand and integrate the available utilities into their projects. The documentation includes detailed explanations, usage examples, and guidance for each function and module within the library. By improving clarity and accessibility, this update enhances developer experience, reduces onboarding time, and ensures that contributors and users can effectively leverage the library's AI-related features.
- 

- **PR5:** This pull request enhances the existing **Contribution Guide** by adding a clear and essential documentation step, ensuring contributors follow a more structured and complete workflow. The update highlights when and how documentation should be added or updated during feature development or bug fixes, improving consistency and overall project maintainability. By incorporating this additional step, the contribution process becomes more transparent, beginner-friendly, and aligned with best practices for collaborative open-source development.
- 

8 LinkedIn Posts

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

- Self Hosting Post –
- https://www.linkedin.com/posts/eswar-venkata-ram-charan-seeram-66bab9364_opensource-kluniversity-foss-ugcPost-7383074249445482497-pJgy?utm_source=share&utm_medium=member_android&rcm=ACoAAFiRgiEB5wyZgh9DIGFDRQ7cU9Uwmnb7Kpo
- PR Merge Post –
- https://www.linkedin.com/posts/nandan-sameer-reddy-gurala-969775355_hacktoberfest2025-opensource-hacktoberfest-activity-7399109170228219905-5E6p?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFiRgiEB5wyZgh9DIGFDRQ7cU9Uwmnb7Kpo
- Blog Post –
- https://www.linkedin.com/posts/nandan-sameer-reddy-gurala-969775355_my-open-source-engineering-journey-activity-7398661065506463744-a8ml?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFiRgiEB5wyZgh9DIGFDRQ7cU9Uwmnb7Kpo