



## **EXPERIENTIAL LEARNING & GLOBAL ENGAGEMENT**

# Open Source Engineering

## Student Details

**Name:** GORLE SUNIL KUMAR

**Roll Number:** 2400032539

**Department:** Computer Science and Engineering

**University:** KL University

Course: Open Source Engineering

Semester: ODD SEMESTER

Submitted to:

Dr. Sripath Roy

Department of Computer Science and Engineering  
KL University

## Contents

<b>1 Linux Distribution</b>	<b>4</b>
1.1 Distribution Used: Ubuntu 22.04 LTS . . . . .	4
1.2 Why Ubuntu? . . . . .	4
1.3 Key Features of Ubuntu 22.04 LTS . . . . .	4
1.4 System Specifications . . . . .	4
1.5 Installation Process . . . . .	5
<b>2 Encryption and GPG</b>	<b>5</b>
2.1 What is Encryption? . . . . .	5
2.2 Types of Encryption . . . . .	5
2.2.1 Symmetric Encryption . . . . .	5
2.2.2 Asymmetric Encryption . . . . .	5
2.3 GNU Privacy Guard (GPG) . . . . .	5
2.4 Installing GPG . . . . .	5
2.5 Generating GPG Keys . . . . .	6
2.6 Listing Keys . . . . .	6
2.7 Exporting Public Key . . . . .	6
2.8 Encrypting Files . . . . .	6
2.9 Decrypting Files . . . . .	6
<b>3 Sending Encrypted Email</b>	<b>6</b>
3.1 Email Encryption Overview . . . . .	6
3.2 Tools Used . . . . .	6
3.3 Setting up Thunderbird with GPG . . . . .	7
3.3.1 Installation . . . . .	7
3.3.2 Configuring OpenPGP . . . . .	7
3.4 Sending Encrypted Email . . . . .	7
3.5 Receiving Encrypted Email . . . . .	7
3.6 Best Practices . . . . .	7
<b>4 Privacy Tools from prism-break.org</b>	<b>8</b>
4.1 What is PRISM-Break? . . . . .	8
4.2 Tool 1: Signal - Encrypted Messaging . . . . .	8
4.3 Tool 2: Firefox - Web Browser . . . . .	8
4.4 Tool 3: ProtonMail - Encrypted Email . . . . .	9
4.5 Tool 4: Tor Browser - Anonymous Browsing . . . . .	9
4.6 Tool 5: VeraCrypt - Disk Encryption . . . . .	10
<b>5 Open Source License</b>	<b>10</b>
5.1 License Used: MIT License . . . . .	10
5.2 What is the MIT License? . . . . .	10
5.3 MIT License Text . . . . .	10
5.4 Why Choose MIT License? . . . . .	11
5.5 Other Common Open Source Licenses . . . . .	11
5.5.1 GPL (GNU General Public License) . . . . .	11
5.5.2 Apache License 2.0 . . . . .	11

5.5.3	BSD License . . . . .	11
<b>6</b>	<b>Self-Hosted Server: QloApps</b>	<b>12</b>
6.1	What is QloApps? . . . . .	12
6.2	Team Members . . . . .	12
6.3	Project Resources . . . . .	12
6.4	Why Self-Host QloApps? . . . . .	13
6.5	Installation Guide . . . . .	13
6.5.1	System Requirements . . . . .	13
6.5.2	Installation Steps . . . . .	13
6.6	Key Features of QloApps . . . . .	15
6.6.1	Hotel Management . . . . .	15
6.6.2	Booking System . . . . .	16
6.6.3	Customer Management . . . . .	16
6.6.4	Payment Integration . . . . .	16
6.7	Localization - Telugu Translation Documentation . . . . .	16
6.7.1	Telugu Documentation Title . . . . .	16
6.7.2	Key Sections Translated . . . . .	16
6.7.3	Translation Highlights . . . . .	17
6.7.4	Documentation Benefits . . . . .	17
6.8	Benefits of Self-Hosting QloApps . . . . .	17
6.9	Backup and Maintenance . . . . .	17
6.9.1	Database Backup . . . . .	17
6.9.2	Files Backup . . . . .	17
6.9.3	Useful Resources . . . . .	18
<b>7</b>	<b>Open Source Contributions Overview</b>	<b>18</b>
7.1	PR 1: update Contributors.md with my details . . . . .	18
7.1.1	Issue . . . . .	18
7.1.2	Solution . . . . .	18
7.2	PR 2: clean up code by deleting unused function . . . . .	18
7.2.1	Issue . . . . .	18
7.2.2	Solution . . . . .	18
7.3	PR 3: Add NASA Image and Video Library API to the Science & Math category . . . . .	19
7.3.1	Issue . . . . .	19
7.3.2	Solution . . . . .	19
7.4	PR 4: The repository lacked a basic Java example for beginners participating in Hacktoberfest. . . . .	19
7.4.1	Issue . . . . .	19
7.4.2	Solution . . . . .	19
7.5	PR 5: The repository did not have the additional file/content needed for the session learning materials. . . . .	19
7.5.1	Issue . . . . .	19
7.5.2	Solution . . . . .	20
7.6	PR 6: The comment used British spelling “behaviour” instead of the required American spelling “behavior.” . . . . .	20
7.6.1	Issue . . . . .	20

7.6.2	Solution . . . . .	20
7.7	PR 7:Corrected the grammar by changing “module” to “modules” in the comment. . . . .	20
7.7.1	Issue . . . . .	20
7.7.2	Solution . . . . .	20
7.8	PR 8: Update Contributors.md with New Contributor . . . . .	20
7.8.1	Issue . . . . .	21
7.8.2	Solution . . . . .	21
7.9	PR 9: Updated the spelling from “behaviour” to “behavior” in the DeprecationLogger comment. . . . .	21
7.9.1	Issue . . . . .	21
7.9.2	Solution . . . . .	21
7.10	Key Learnings from Contributions . . . . .	21
<b>8</b>	<b>LinkedIn Posts</b>	<b>22</b>
8.1	Post 1: Software Testing and Contributions . . . . .	22
8.2	Post 2: FOSS Culture at KL University . . . . .	22
8.3	Post 3: Blog Link to My Open Source Journey . . . . .	22
<b>9</b>	<b>Conclusion</b>	<b>23</b>

# 1 Linux Distribution

## 1.1 Distribution Used: Ubuntu 22.04 LTS

For this project, I have used **Ubuntu 22.04 LTS** as my primary operating system.

## 1.2 Why Ubuntu?

Ubuntu is one of the most popular Linux distributions for several reasons:

- **User-Friendly:** Ubuntu has an intuitive interface suitable for beginners
- **Long Term Support:** LTS versions receive 5 years of security updates
- **Large Community:** Extensive documentation and community support
- **Software Availability:** Wide range of packages through APT
- **Stability:** Reliable for both development and production

## 1.3 Key Features of Ubuntu 22.04 LTS

1. **Desktop Environment:** GNOME 42
2. **Kernel Version:** Linux 5.15 LTS
3. **Package Manager:** APT (Advanced Package Tool)
4. **Default Applications:** Firefox, LibreOffice, GNOME utilities
5. **Snap Support:** Built-in support for snap packages

## 1.4 System Specifications

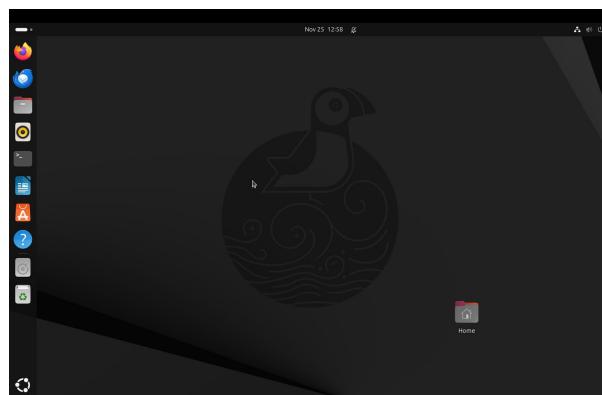


Figure 1: System Information

My system configuration:

- Operating System: Ubuntu 22.04 LTS

- Architecture: x86\_64
- Desktop Environment: GNOME
- Shell: Bash 5.1

## 1.5 Installation Process

The installation involved:

1. Downloaded Ubuntu 22.04 LTS ISO from official website
2. Created bootable USB using Rufus/Etcher
3. Configured dual boot with existing OS
4. Installed essential development tools
5. Configured system for open source development

## 2 Encryption and GPG

### 2.1 What is Encryption?

Encryption is the process of converting plaintext into ciphertext to protect data confidentiality. It ensures that only authorized parties can access the information.

### 2.2 Types of Encryption

#### 2.2.1 Symmetric Encryption

Uses the same key for encryption and decryption. Examples: AES, DES.

#### 2.2.2 Asymmetric Encryption

Uses a public-private key pair. Examples: RSA, ECC.

### 2.3 GNU Privacy Guard (GPG)

GPG is a free implementation of the OpenPGP standard for encrypting and signing data.

### 2.4 Installing GPG

```
1 sudo apt update
2 sudo apt install gnupg
3 gpg --version
```

## 2.5 Generating GPG Keys

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

Steps followed:

1. Selected RSA and RSA (default)
2. Key size: 4096 bits
3. Key validity: 1 year
4. Entered name and email
5. Created strong passphrase

## 2.6 Listing Keys

```
1 gpg --list-keys
2 gpg --list-secret-keys
```

## 2.7 Exporting Public Key

```
1 gpg --armor --export your-email@example.com > public-key.asc
```

## 2.8 Encrypting Files

```
1 gpg --encrypt --recipient your-email@example.com document.txt
```

## 2.9 Decrypting Files

```
1 gpg --decrypt document.txt.gpg > document.txt
```

# 3 Sending Encrypted Email

## 3.1 Email Encryption Overview

Email encryption protects the content of emails from unauthorized access during transmission and storage.

## 3.2 Tools Used

- **Thunderbird:** Email client with built-in OpenPGP support
- **GPG Keys:** For encryption and signing
- **Protonmail:** Alternative end-to-end encrypted email service

### 3.3 Setting up Thunderbird with GPG

#### 3.3.1 Installation

```
1 sudo apt install thunderbird
```

#### 3.3.2 Configuring OpenPGP

Steps followed:

1. Open Thunderbird
2. Go to Account Settings
3. Select End-to-End Encryption
4. Add existing GPG key or generate new one
5. Import recipient's public key

### 3.4 Sending Encrypted Email

Process:

1. Compose new email
2. Click on Security button
3. Select "Require Encryption"
4. Optionally add digital signature
5. Send email

### 3.5 Receiving Encrypted Email

When receiving:

1. Email appears encrypted
2. Thunderbird automatically detects encryption
3. Enter GPG passphrase
4. Email content is decrypted and displayed

### 3.6 Best Practices

- Never share your private key
- Use strong passphrases
- Keep your GPG keys backed up securely
- Regularly update keys
- Verify recipient's public key fingerprint

## 4 Privacy Tools from prism-break.org

### 4.1 What is PRISM-Break?

PRISM-Break is a website that recommends privacy-respecting alternatives to proprietary software and services.

### 4.2 Tool 1: Signal - Encrypted Messaging

**Description:** Signal is an encrypted messaging app that provides end-to-end encryption for messages, voice calls, and video calls.

**Key Features:**

- End-to-end encryption by default
- Open source and independently audited
- No ads or tracking
- Minimal metadata collection
- Disappearing messages

**Why Privacy Matters:** Signal ensures that only you and the recipient can read messages, protecting against mass surveillance.

### 4.3 Tool 2: Firefox - Web Browser

**Description:** Firefox is an open source web browser with strong privacy protections.

**Privacy Features:**

- Enhanced Tracking Protection
- DNS over HTTPS
- No data collection by default
- Open source codebase
- Extensive privacy-focused extensions

**Configuration Tips:**

- Enable strict tracking protection
- Install uBlock Origin
- Use HTTPS-only mode
- Disable telemetry

#### 4.4 Tool 3: ProtonMail - Encrypted Email

**Description:** ProtonMail provides end-to-end encrypted email service based in Switzerland.

**Key Features:**

- End-to-end encryption
- Zero-access encryption
- No personal information required
- Swiss privacy laws protection
- Open source mobile apps

**Use Cases:**

- Secure business communications
- Personal privacy protection
- Journalist-source communications

#### 4.5 Tool 4: Tor Browser - Anonymous Browsing

**Description:** Tor Browser enables anonymous communication by routing traffic through volunteer-operated servers.

**How It Works:**

- Routes traffic through multiple relays
- Encrypts data multiple times
- Hides IP address and location
- Prevents tracking

**Best Use Cases:**

- Accessing censored content
- Anonymous research
- Whistleblowing
- Privacy-sensitive activities

## 4.6 Tool 5: VeraCrypt - Disk Encryption

**Description:** VeraCrypt is a free open source disk encryption software.

**Features:**

- Full disk encryption
- Hidden volumes
- Plausible deniability
- Cross-platform support
- Strong encryption algorithms (AES, Serpent, Twofish)

**Use Cases:**

- Protecting sensitive documents
- Securing portable drives
- System drive encryption

## 5 Open Source License

### 5.1 License Used: MIT License

For my open source contributions and projects, I primarily work with the **MIT License**.

### 5.2 What is the MIT License?

The MIT License is a permissive free software license that allows users to:

- Use the software commercially
- Modify the software
- Distribute the software
- Use the software privately
- Sublicense the software

### 5.3 MIT License Text

```
1 MIT License
2
3 Copyright (c) 2025 GORLE SUNIL KUMAR
4
5 Permission is hereby granted, free of charge, to any person
6 obtaining a copy of this software and associated documentation
7 files (the "Software"), to deal in the Software without
8 restriction, including without limitation the rights to use,
```

```
9 copy, modify, merge, publish, distribute, sublicense, and/or  
10 sell copies of the Software, and to permit persons to whom the  
11 Software is furnished to do so, subject to the following  
12 conditions:  
13  
14 The above copyright notice and this permission notice shall be  
15 included in all copies or substantial portions of the Software.  
16  
17 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...  
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...
```

## 5.4 Why Choose MIT License?

1. **Simple and Easy:** Short and easy to understand
2. **Permissive:** Minimal restrictions on reuse
3. **Business-Friendly:** Can be used in proprietary software
4. **Popular:** Widely used and recognized
5. **Compatible:** Works well with other licenses

## 5.5 Other Common Open Source Licenses

### 5.5.1 GPL (GNU General Public License)

- Copyleft license
- Requires derivative works to be open source
- Used by Linux kernel

### 5.5.2 Apache License 2.0

- Permissive like MIT
- Includes patent grant
- Used by Apache projects

### 5.5.3 BSD License

- Very permissive
- Similar to MIT
- Used by FreeBSD

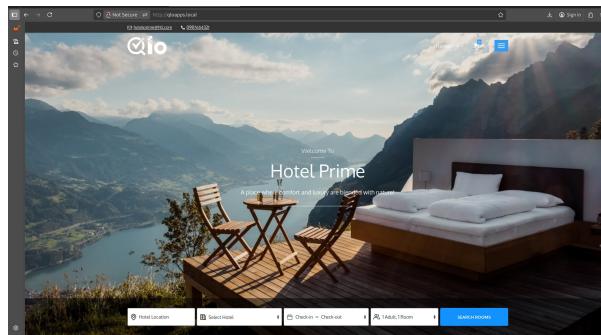


Figure 2: QloApps

## 6 Self-Hosted Server: QloApps

### 6.1 What is QloApps?

QloApps is a free, open-source hotel management and online booking system built on PrestaShop that allows you to manage:

- Hotel rooms and properties
- Online and offline reservations
- Pricing and availability
- Customer management
- Website booking engine
- Payment processing

### 6.2 Team Members

This project was completed in collaboration with:

- **sunil kumar** - Lead Developer
- **venkat ramana** - Developer and Tester

### 6.3 Project Resources

- **Demo Video:** [https://drive.google.com/file/d/1hf5ehgALdg1\\_b7oY3Xf3rbd0R6PAqmsb/view?usp=drivesdk](https://drive.google.com/file/d/1hf5ehgALdg1_b7oY3Xf3rbd0R6PAqmsb/view?usp=drivesdk)
- **LinkedIn Post:** [https://www.linkedin.com/posts/venkata-ramana-80a137369\\_opensource-opensource-kluniversity-activity-7382319162603282433-hZ7D](https://www.linkedin.com/posts/venkata-ramana-80a137369_opensource-opensource-kluniversity-activity-7382319162603282433-hZ7D)

## 6.4 Why Self-Host QloApps?

1. **Full Data Control:** All customer and booking data stays on your server
2. **Customization:** Modify templates, modules, pricing logic
3. **Cost Effective:** Core system is free and open source
4. **Independence:** No reliance on third-party hotel software subscriptions
5. **Learning Experience:** Install, configure, and maintain a real Property Management System

## 6.5 Installation Guide

### 6.5.1 System Requirements

Figure 3: QloApps Dashboard

### 6.5.2 Installation Steps

#### Step 1: Update System and Install Required Packages

```

1 sudo apt update && sudo apt upgrade -y
2 sudo apt install apache2 -y
3 sudo apt install mariadb-server mariadb-client -y
4 sudo apt install php php-cli libapache2-mod-php php-mysql \
   php-curl php-gd php-mbstring php-xml php-zip php-intl \
   php-bcmath unzip -y
5 sudo a2enmod rewrite
6 sudo systemctl restart apache2
7
8

```

#### Step 2: Secure MariaDB

```

1 sudo mysql_secure_installation

```

Recommended answers:

Component	Recommended Version
Operating System	Ubuntu 20.04 / 22.04 (64-bit)
Web Server	Apache 2.4 or higher
PHP	7.4 – 8.1
Database	MariaDB / MySQL 5.7 or higher
RAM	Minimum 2 GB
Storage	Minimum 2 GB free space

Table 1: QloApps System Requirements

#### Step 3: Create QloApps Database

```

1 sudo mariadb

```

Execute these SQL commands:

```

1 CREATE DATABASE gloappsdb;
2 CREATE USER 'gloappsuser'@'localhost' IDENTIFIED BY 'StrongPassword123!';
3 GRANT ALL PRIVILEGES ON gloappsdb.* TO 'gloappsuser'@'localhost';
4 FLUSH PRIVILEGES;
5 EXIT;
```

#### Step 4: Download QloApps

```

1 cd /tmp
2 wget https://github.com/Qloapps/QloApps/archive/refs/tags/v1.7.0.zip
3 unzip qloapps.zip
4 sudo mv QloApps-1.7.0 /var/www/html/qloapps
```

#### Step 5: Set Folder Permissions

```

1 sudo chown -R www-data:www-data /var/www/html/qloapps
2 sudo chmod -R 755 /var/www/html/qloapps
3 sudo chmod -R 777 /var/www/html/qloapps/upload
4 sudo chmod -R 777 /var/www/html/qloapps/download
5 sudo chmod -R 777 /var/www/html/qloapps/cache
6 sudo chmod -R 777 /var/www/html/qloapps/logs
```

#### Step 6: Configure Apache Virtual Host

```
1 sudo nano /etc/apache2/sites-available/qloapps.conf
```

Add this configuration:

```

1 <VirtualHost *:80>
2   ServerAdmin admin@localhost
3   DocumentRoot /var/www/html/qloapps
4   ServerName qloapps.local
5
6   <Directory /var/www/html/qloapps/>
7     Options Indexes FollowSymLinks
8     AllowOverride All
9     Require all granted
10  </Directory>
11
12  ErrorLog ${APACHE_LOG_DIR}/qloapps_error.log
13  CustomLog ${APACHE_LOG_DIR}/qloapps_access.log combined
14 </VirtualHost>
```

Enable the site:

```

1 sudo a2ensite qloapps.conf
2 sudo systemctl restart apache2
```

Add to /etc/hosts:

```
1 127.0.0.1 qloapps.local
```

**Step 7: Browser Installation**

Open in browser:

```
1 http://localhost/qloapps
```

or

```
1 http://qloapps.local
```

Follow the installation wizard:

1. Select language
2. Accept license
3. Enter database details:
  - Database name: qloappsdbs
  - User: qloappsuser
  - Password: StrongPassword123!
4. Complete installation

**Step 8: Post-Installation**

Remove install folder:

```
1 sudo rm -rf /var/www/html/qloapps/install
```

Access admin panel:

```
1 http://localhost/qloapps/admin
```

Optional PHP settings:

```
1 sudo nano /etc/php/8.1/apache2/php.ini
```

Modify:

```
1 upload_max_filesize = 16M
2 post_max_size = 16M
3 max_execution_time = 300
```

Restart Apache:

```
1 sudo systemctl restart apache2
```

## 6.6 Key Features of QloApps

### 6.6.1 Hotel Management

- Multiple property management
- Room type configuration
- Pricing and availability calendar
- Seasonal pricing rules
- Room inventory tracking

### 6.6.2 Booking System

- Online booking engine
- Offline/manual bookings
- Booking modifications
- Cancellation management
- Booking status tracking

### 6.6.3 Customer Management

- Guest profiles
- Booking history
- Customer communications
- Loyalty programs

### 6.6.4 Payment Integration

- Multiple payment gateways
- Online payment processing
- Invoice generation
- Payment tracking

## 6.7 Localization - Telugu Translation Documentation

As part of making QloApps accessible to Telugu-speaking users, I have created comprehensive documentation in Telugu language.

### 6.7.1 Telugu Documentation Title

#### QloApps

(QloApps Self-Hosting Documentation)

### 6.7.2 Key Sections Translated

The Telugu documentation includes:

1. **(Introduction):** Overview of QloApps and its capabilities
2. **(System Requirements):** Hardware and software requirements
3. **(Installation):** Step-by-step installation guide
4. **(Database Setup):** Database configuration instructions
5. **(Maintenance):** Backup and maintenance procedures

### 6.7.3 Translation Highlights

Key technical terms translated to Telugu:

Question	Answer
Switch to unix_socket authentication	Y
Remove anonymous users	Y
Disallow root login remotely	Y
Remove test database and access	Y
Reload privilege tables now	Y

### 6.7.4 Documentation Benefits

- Makes QloApps accessible to Telugu-speaking hotel owners
- Removes language barriers in technology adoption
- Promotes open source software in regional languages
- Helps local businesses implement self-hosted solutions
- Demonstrates importance of localization in software

## 6.8 Benefits of Self-Hosting QloApps

1. **Complete Privacy:** All guest data remains on your server
2. **No Commission Fees:** Unlike booking platforms, no per-booking fees
3. **Full Customization:** Modify any aspect of the system
4. **Data Ownership:** Complete control over all business data
5. **Educational Value:** Learn hotel management systems and web hosting
6. **Cost Savings:** No monthly subscription fees
7. **Scalability:** Expand as your business grows

## 6.9 Backup and Maintenance

### 6.9.1 Database Backup

```
1 mysql dump -u root -p qloappsdb > qloapps_backup.sql
```

### 6.9.2 Files Backup

```
1 tar -czvf qloapps_files_backup.tar.gz /var/www/html/qloapps
```

### 6.9.3 Useful Resources

- Official Documentation: <https://devdocs.qloapps.com>
- Demo Website: <https://demo.qloapps.com>
- Forum / Support: <https://forums.qloapps.com>

This section details all pull requests I have contributed to various open source projects, including screenshots and detailed descriptions of each contribution.

## 7 Open Source Contributions Overview

### 7.1 PR 1: update Contributors.md with my details

**Repository:** firstcontributions/first-contributions

**PR Number:** #106873

**Status:** Merged

#### 7.1.1 Issue

Your PR included unwanted or auto-formatted changes and you did not mark the checklist correctly.

#### 7.1.2 Solution

- Use git add -p to stage only your changes.
- Turn off auto-formatting before editing.
- Tick checklist items with - [x].

### 7.2 PR 2: clean up code by deleting unused function

**Repository:** thatithuripriya/goat-shriek

**PR Number:** #3

**Status:** Open

#### 7.2.1 Issue

Your repository has no clear description, topics, or detailed README, so contributors can't understand the project well.

#### 7.2.2 Solution

- Add a clear project description and topics in the repository.
- Improve the README with project details and contribution steps.

### 7.3 PR 3: Add NASA Image and Video Library API to the Science & Math category

**Repository:** public-apis/public-apis

**PR Number:** #4885

**Status:** Open

#### 7.3.1 Issue

The Science & Math category was missing the NASA Image and Video Library API.

#### 7.3.2 Solution

- Added the NASA API entry to the category, providing access to space images, videos, and audio.

### 7.4 PR 4: The repository lacked a basic Java example for beginners participating in Hacktoberfest.

**Repository:** sbmxc/hacktoberfest

**PR Number:** #772

**Status:** Open

#### 7.4.1 Issue

The repository did not have a beginner-friendly Java example for Hacktoberfest contributors.

#### 7.4.2 Solution

- Added a simple HelloWorld.java file.
- Program prints a greeting message for Hacktoberfest.
- Helps beginners make an easy first contribution.

### 7.5 PR 5: The repository did not have the additional file/content needed for the session learning materials.

**Repository:** Gorlesunilkumar/Y24OpenSourceEngineering

**PR Number:** #1

**Status:** merged

#### 7.5.1 Issue

The repository had incomplete content and was missing learner-friendly material.

### 7.5.2 Solution

- Added the missing file to complete the learning materials.
- Updated the repository to make it more helpful for learners.

## 7.6 PR 6: The comment used British spelling “behaviour” instead of the required American spelling “behavior.”

**Repository:** gradle/gradle

**PR Number:** #35806

**Status:** Open

### 7.6.1 Issue

The LockMode comment still used the British spelling ”behaviour”, which violates Gradle’s rule (ADR-0009) requiring American English spelling in all docs and comments.

### 7.6.2 Solution

- Updated ”behaviour” → ”behavior” to follow Gradle’s documentation standard.
- Made only a doc/comment fix with no functional code changes.

## 7.7 PR 7: Corrected the grammar by changing “module” to “modules” in the comment.

**Repository:** gradle/gradle

**PR Number:** #35807

**Status:** Open

**Tasks Completed:** 9

### 7.7.1 Issue

The LockMode comment had a grammar mistake: it used ”module” instead of the correct plural ”modules.”

### 7.7.2 Solution

- Changed ”module” → ”modules” to fix the grammar and improve clarity.

## 7.8 PR 8: Update Contributors.md with New Contributor

**Repository:** Gorlesunilkumar/freeCodeCamp

**PR Number:** #1

**Status:** Merged

### 7.8.1 Issue

The case-sensitivity lesson had a grammar mistake/missing comma, making the explanation unclear.

### 7.8.2 Solution

- Added the missing comma to fix the grammar and improve clarity.

## 7.9 PR 9: Updated the spelling from “behaviour” to “behavior” in the DeprecationLogger comment.

**Repository:** gradle/gradle

**PR Number:** #35805

**Status:** open

**Tasks:** 3

### 7.9.1 Issue

The DeprecationLogger comment used the British spelling “behaviour” instead of the required American spelling “behavior.”

### 7.9.2 Solution

- Replaced “behaviour” → “behavior” to follow Gradle’s American English documentation rule (ADR-0009).

English	Telugu ()
Installation	
Database	
Server	
Requirements	
Configuration	
Backup	
Maintenance	

Table 2: English to Telugu Technical Terms

## 7.10 Key Learnings from Contributions

- Git Workflow:** Mastered branching, committing, and pull request processes
- Code Review:** Learned to receive and implement feedback constructively
- Documentation:** Understood the importance of clear, comprehensive documentation
- Collaboration:** Experienced working with global developer communities

5. **Testing:** Learned to write comprehensive test suites
6. **Localization:** Appreciated the value of making software accessible in regional languages

## 8 LinkedIn Posts

### 8.1 Post 1: Software Testing and Contributions

**Link:** [https://www.linkedin.com/posts/g-sunil-kumar-68b045370\\_my-first-open-source-con](https://www.linkedin.com/posts/g-sunil-kumar-68b045370_my-first-open-source-con)  
`utm_source=share&utm_medium=member_android&rclm=ACoAAFvXqYsBQGTYHCu1im_4Q2nKbOCe1Pp8LII`

**Summary:** Shared experiences contributing to TheAlgorithms/Java repository, focusing on implementing comprehensive test suites and best practices in software testing.

**Key Points:**

- Importance of thorough testing in open source
- Test-driven development approach
- Learning from code reviews
- Contributing to educational repositories

### 8.2 Post 2: FOSS Culture at KL University

**Link:** [https://www.linkedin.com/posts/g-sunil-kumar-68b045370\\_opensource-opensource-kl](https://www.linkedin.com/posts/g-sunil-kumar-68b045370_opensource-opensource-kl)  
`utm_source=share&utm_medium=member_android&rclm=ACoAAFvXqYsBQGTYHCu1im_4Q2nKbOCe1Pp8LII`

**Summary:** Discussed the Free and Open Source Software culture at KL University and encouraged peers to participate in open source.

**Highlights:**

- FOSS courses and initiatives at university
- Personal journey in open source
- Community-driven development model
- Real-world learning experience

### 8.3 Post 3: Blog Link to My Open Source Journey

**Link:** [https://www.linkedin.com/posts/g-sunil-kumar-68b045370\\_journey-of-open-source-e](https://www.linkedin.com/posts/g-sunil-kumar-68b045370_journey-of-open-source-e)  
`utm_source=share&utm_medium=member_android&rclm=ACoAAFvXqYsBQGTYHCu1im_4Q2nKbOCe1Pp8LII`

**Summary:** Shared a comprehensive blog post documenting my open source journey, including challenges faced and lessons learned.

## 9 Conclusion

This report documents my comprehensive journey in open source software development, including:

- Setting up development environment with Ubuntu Linux
- Understanding encryption and privacy tools
- Self-hosting QloApps for hotel management
- Making meaningful contributions to open source projects
- Sharing knowledge through professional networking

Through these experiences, I have gained:

1. Technical skills in multiple programming languages and systems
2. Understanding of collaborative development workflows
3. Experience with version control and code review processes
4. Appreciation for open source philosophy and community
5. Professional networking and communication abilities
6. Practical knowledge of server administration and web hosting

The experience of self-hosting QloApps has been particularly valuable, providing hands-on experience with:

- LAMP stack configuration
- Database management
- Web server administration
- Security best practices
- Real-world application deployment

Open source contribution has been an invaluable learning experience, providing real-world software development exposure and connecting me with a global community of developers committed to building better, more accessible software for everyone.