

The Role of GitHub in Open Source Research and as a pedagogical tool in enhancing education

Binu Mol T. V.

1*Assistant Professor, Department of Computer Science, KKTM Govt.College, Pullut, Thrissur, Kerala, India
Email: binumoltv@gmail.com

How to cite this article: Binu Mol T. V. (2024). The Role of GitHub in Open Source Research and as a pedagogical tool in enhancing education. Library Progress International, 44(4), 150-155

ABSTRACT

Open-source software refers to a development model where the source code is freely accessible to the public and distributed under open-source licenses. It enables anyone to view, use, modify, and distribute the code. Research on open-source encompasses all phases of the process, including data, methods, software, results, and publications, all of which are made publicly available and accessible. GitHub is one of the largest collaborative platforms for hosting open-source projects, allowing developers, researchers, and organizations to manage, contribute to, and host projects, encouraging collaborative innovation. GitHub has as a powerful tool in education due to Its collaborative features and openness which make it ideal for active learning, and preparing students for real-world professional environments. GitHub promotes openness in education, where students can share their work with a global audience and educators can create and share course materials which we call Open Educational Resources (OER). IoT-based streamflow monitoring systems can be implemented using affordable microcontrollers like Arduino, Raspberry Pi, or Rock Pi. These devices can interface with sensors to monitor water flow, water levels, temperature, and other parameters, while providing real-time data collection, transmission, and visualization.

KEYWORDS: Open source, GitHub, IoT, open research, Open Education Resources

1. Introduction

The term open source refers to something that can be altered and shared because its design is openly available to the public [1]. They operate under a collaborative model. Proprietary software, also known as closed-source, is owned by a company or individual and is not made available to the public. Examples of open-source software include Linux, Mozilla Firefox, Apache Web Server, and GIMP. Open-source software is distributed under licenses that grant users the freedom to view, modify, and share the source code. The GNU General Public License (GPL) is the most widely used open-source license. Examples of open-source AI software include ClearML, PyTorch, and Google's TensorFlow. Free and Open Source Software (FOSS) in education fosters collaborative learning while offering cost-efficient solutions in academic settings. Integrating free and open-source software enables students to gain practical, real-world skills and experiences while building a portfolio to showcase to future employers [2]. GitHub is a popular software development platform that facilitates version control, collaborative development, and project hosting, and it is not exclusively limited to open-source projects. Integrating GitHub into the classroom provides students with the opportunity to gain hands-on experience with a widely used industry tool [3].

Methodology

2. Literature Review

Open-source software (OSS) grants users the ability to access its source code, enabling them to freely use, modify, debug, enhance, and distribute the software. Some researchers categorize OSS as a public good provided through voluntary

private contributions [4]. Table 1 [4] below clearly gives an idea of the difference between a proprietary and open source software.

Table 1. Examples of Open Source and Proprietary Applications

Application	Proprietary	Open Source
Web Browser	MS Internet Explorer, Netscape	Mozilla (based on Netscape)
Web Server	Microsoft IIS, Netscape	Apache
Application Server	WebMethods (includes JBoss application)	JBoss, Apache Tomcat, Enhydra
Office Suite	MS Office, Corel WordPerfect Office	OpenOffice.org (based on Sun's proprietary StarOffice)
E-mail/collaboration	MS Exchange, Lotus Notes	Ximian Evolution
Database	Oracle9i, IBM DB2, MS SQL	MySQL, PostgreSQL (both lack the power and depth of industrial-strength Oracle and DB2)

Examples of FOSS Tools for Students are LibreOffice, TuxMath and GeoGebra, Audacity, GIMP, and Inkscape. Scratch is a visual programming language developed by MIT to teach coding basics in an interactive, block-based interface that can be integrated into the curricula for creative programming.

Raspberry Pi OS is a Linux-based operating system specifically designed to run on Raspberry Pi hardware [5] and is a single board computer (SBC). Python is widely supported for coding. Pi-Top and Kano kits can be used for building hardware projects, like weather stations or robots. They are compatible with sensors and GPIO pins for integrating hardware and software.

Open-source software research can be divided into three key phases: Initiation Phase, Ongoing Project Phase, and Adoption/Deployment Phase. The Initiation Phase involves identifying, explaining, and validating the factors that influence the decision to launch an open-source project. The Ongoing Project Phase focuses on sustaining open-source projects, where users actively contribute to the development process. For example, GitHub plays a transformative role in the open source ecosystem by enabling global collaboration, enhancing software quality through automation, fostering transparency in research, and democratizing access to software development. It has become the backbone of modern open source development, driving innovation across industries and disciplines. The Adoption/Deployment Phase evaluates how open-source influences software quality, development efficiency, and long-term growth [4].

Additionally, students can showcase their projects publicly on GitHub, which employers frequently review during the hiring process. GitHub also helps students enhance their communication, collaboration, and critical thinking skills. Implementing GitHub in the classroom can be adopted as a best practice [3].

A beginner in Github can follow the steps below to create an account in GitHub [14].

Create an account with a username by visiting GitHub.

We have to select the field in which we work like student teacher or developer etc.

Then, select which fields we want to work with like contributing to open source projects or using GitHub Copilot, etc .

You can either use free access or apply for additional benefits. Figure 1 shows these options.

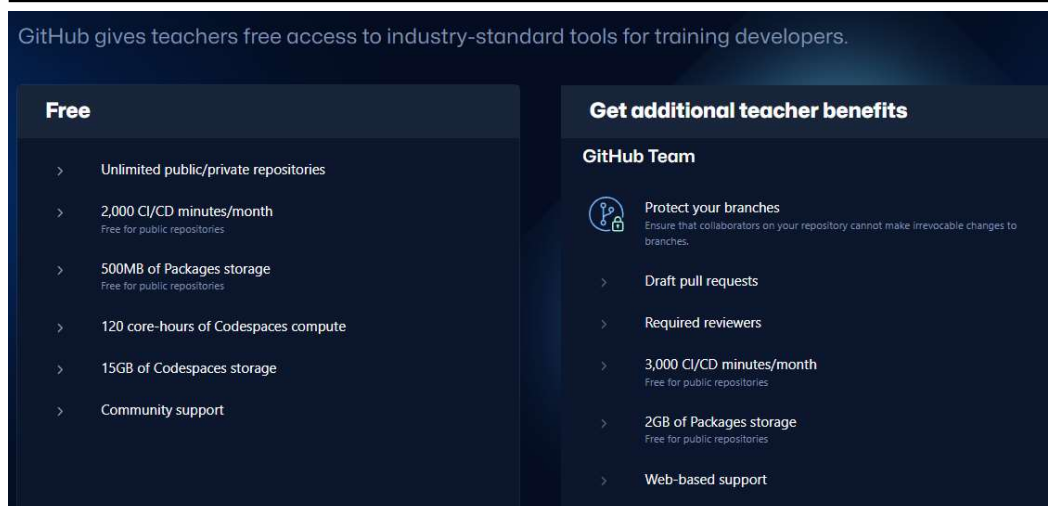


Figure1

Then we have a home page where we can share information about ourselves by creating a profile README.md, which appears at the top of our profile page.

Other facilities include -

Create or import a repository that includes all your project's files, version history, and discussions with collaborators for a new concept, or bring in an existing repository to continue contributing to it, with the option to set its visibility as public or private.

Create a site or blog from the GitHub repository with GitHub pages.

GitHub will give an introductory course that will help you to how to do things in GitHub.

Solve issues for projects or create our own projects.

GitHub Actions is a powerful tool to automate workflows directly in your repository. GitHub Copilot is an AI-powered coding assistant designed to help us write code more efficiently, reducing the effort required and enabling greater focus on problem-solving and collaboration and it has shown to boost developer productivity and speed up the software development process [14].

CI/CD Pipeline Overview: Here Continuous Integration (CI) automates building and testing of code after each change, Continuous Delivery (CD) Prepares code for deployment but waits for approval and Continuous Deployment (CD) fully automates the release to production.

GitHub's version control capabilities offer a unique way for students and educators to track progress over time.

Version History: Students can maintain a detailed history of their work, allowing educators to review the evolution of a project.

Progress Tracking: Educators can track students' contributions and activity over time, making it easier to assess individual effort in group projects.

Practical Skill Enhancement: Leveraging GitHub in education equips students with hands-on experience and essential skills that are highly sought after in the professional world.

Exposure to Industry Tools: Familiarizes students with Git, GitHub, and collaborative workflows used in professional software development.

Continuous Integration (CI): Students can automate testing and deployment using GitHub Actions, which introduces them to CI/CD pipelines.

Open Source Contribution: Encourages students to contribute to open source projects, enhancing their portfolios and employability.

GitHub's issue tracking and pull request system provide unique opportunities for active learning.

Problem-Solving with Issues: Students can use GitHub's issue tracker to document problems, brainstorm solutions, and discuss strategies with peers.

Learning Through Pull Requests: Students can submit pull requests to propose changes, allowing educators to provide feedback inline with the code.

Automated Feedback with Actions: CI tools can provide immediate feedback on code quality, formatting, and functionality.

Fostering a Culture of Open Science and Research: GitHub encourages students to engage in open research practices.

Open Science Repositories: Students can share data, methods, and results, contributing to the open science movement.

Reproducible Research: Encourages students to create reproducible research workflows, ensuring that their work can be validated by others.

Features for educators:

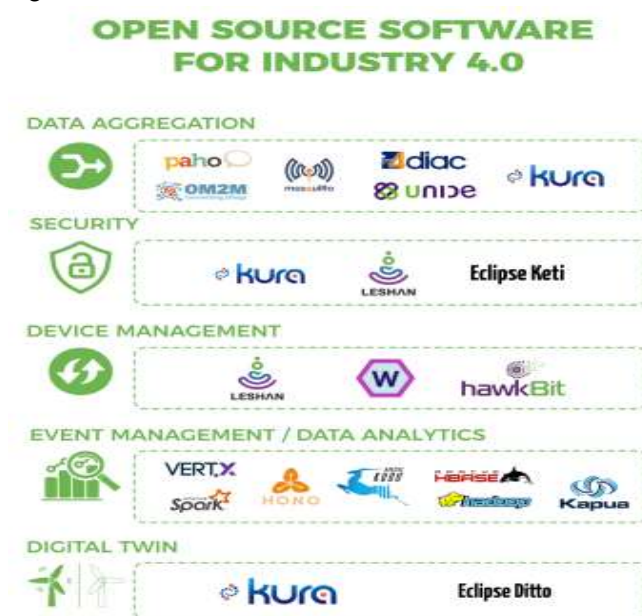
GitHub Classroom: A tool designed specifically for educators to manage assignments, grade submissions, and monitor progress.

Open Source Tools: Many educational resources and tools integrated with GitHub are freely available, reducing barriers to access.

Open source software ensures cost-effective IoT solutions by minimise ongoing maintenance and operational expenses. Open source software empowers IoT developers to leverage the power of community-driven innovation while avoiding vendor lock-in. Examples of real-world open-source IoT projects include Eclipse IoT, a community-led initiative offering open-source IoT frameworks and technologies, and the Zephyr Project, an open-source real-time operating system (RTOS) tailored for IoT devices with limited resources [7]. ThingsBoard is an open-source IoT platform that provides features such as data visualization, device management, and a rule engine for automated workflows [7]. The Internet of Things (IoT) technology involves embedding sensors into everyday objects and leveraging connectivity to enable the exchange of information, which is applied across various use cases and industries [8].

Integrating IoT technology into the manufacturing sector is referred to as Industry 4.0. Open-source communities such as Eclipse IoT and the Apache Software Foundation play a crucial role in developing production-ready Industry 4.0 solutions. Various open-source tools are utilized for purposes like data aggregation, security, device management, data analytics, and digital twins, as illustrated in the Figure 2 below [9].

Figure 2



Open-source tools available for data analysis and visualization include Orange, a popular platform recognized for its intuitive, visual programming interface, making data exploration and analysis accessible to users without extensive coding experience and it provides tools for machine learning, text mining, and network analysis [10]. Gephi is designed to visualize and analyze network data, such as social or biological networks. It supports real-time graph manipulation and exploration, making it suitable for discovering patterns and relationships in data. Gephi is highly customizable and user-friendly, even for those without programming experience [11]. Plotly provides powerful tools for creating interactive web-based visualizations, including graphs and dashboards. It integrates with Python, R, MATLAB, and JavaScript [12]. RAWGraphs is a lightweight, open-source visualization tool that transforms data from spreadsheets into various unique

graph types. It bridges the gap between raw data and polished visual presentations. Non-programmers can also use it easily [13].

IoT-based streamflow monitoring systems is proposed to use open source research to implement using affordable microcontrollers like Arduino, Raspberry Pi, or Rock Pi [15].

Below is a conceptual block diagram (Figure 3) describing the key components and their interactions in the system:

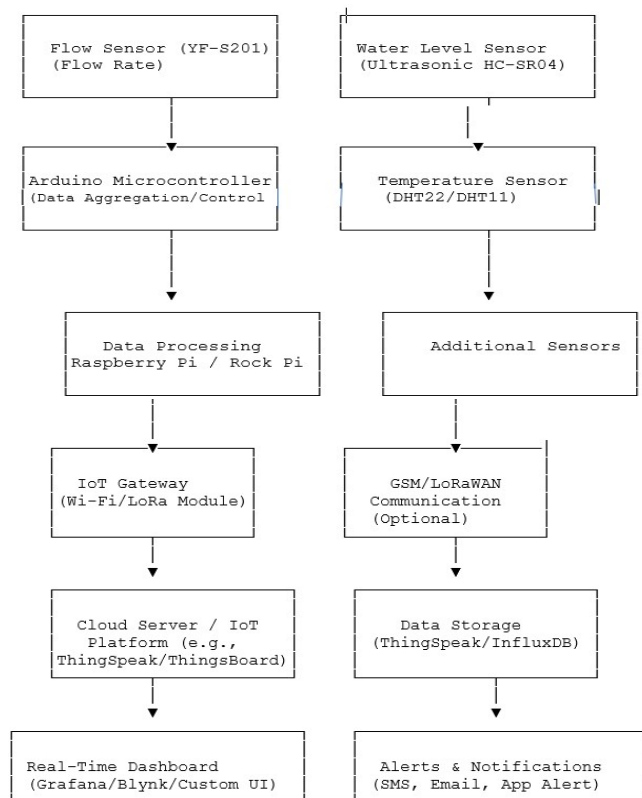


Figure 3

3. Findings

The comparison between various paid software packages and their open-source counterparts reveals the following insights: Paid software is extensively adopted due to strong brand recognition, driven by marketing strategies such as pre-installation on new PCs and laptops. In contrast, open-source software is a community-driven endeavor, where generating revenue is not the primary focus, resulting in limited marketing efforts. Additionally, most educational institutions tend to favor proprietary software due to the perceived advantages in flexibility and access to dedicated support [6].

4. Limitations

Students and educators may require initial training in Git and GitHub. Reliable internet access is necessary for full participation.

5. Conclusion

GitHub is a powerful pedagogical tool that enhances education by promoting collaboration, developing real-world skills, and fostering active learning. Its use in classrooms prepares students for careers in technology while supporting open science and research practices. Most servers on the internet are powered by open-source software and we are likely interacting with systems running on Linux-based machines [6]. As part of utilizing open-source research, an IoT-powered streamflow monitoring system using Arduino, Raspberry Pi, or Rock Pi is proposed, offering a cost-efficient and scalable approach to stream monitoring. By harnessing IoT sensors, communication technologies, and cloud services, these systems deliver critical data for flood mitigation, water resource optimization, and environmental conservation. GitHub serves as a collaborative hub where developers, researchers, and organizations can host, manage, and contribute to open-source initiatives. It has revolutionized the way open source software (OSS) is developed and maintained, fostering innovation and global collaboration.

References

- [1] <https://opensource.org/ai/open-source-ai-definition>
- [2] ICT Tools and Resources-A Study of Higher Education in India, Binu Mol T.V, IJISAE,2024
- [3] Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences: Findings from a Survey of Students and Teachers Courtney Hsing, Vanessa Gennarelli 2019
- [4] Understanding Open Source Software: A Research Classification Framework Matthew Nelson ,Ravi Sen s 2006
- [5] A Study on Raspberry PI, Arduino and IoT with Its Application in E-Controlling UPS, Binu Mol T.V, JES,2024
- [6] The impact of using open source software on the progress and quality of the educational process abdullah alzhahrani, yazeed al moaiad, yousef el-ebiary 2022
- [7] The Role of Open Source in the Growth and Adoption of IoT by Sufyan bin Uzayr,2023
- [8] Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0 Robin Chataut, Alex Phoummalayvane and Robert Ak 2023
- [9] Open Source Software for Industry 4.0 2017 <https://iot.eclipse.org>
- [10] <https://orangedatamining.com/>
- [11] <https://gephi.org/>
- [12] <https://plotly.com/>
- [13] <https://app.rawgraphs.io/>
- [14] <https://github.com/>
- [15] Design and Development of IoT Based Water Flow Monitoring for Pico Hydro Power Plant by Nurhening Yuniarti , Didik Hariyanto, Sigit Yatmono, and Muchlis Abdillah 2021