Designing a Custom Linux Distribution for Improved Accessibility

S Gokula Krishnan[1], Khushi S Rotti[2], Kushal Hali[3], Sanskruthi Kinagi[4] and Rahul Joy Suneel[5]

[1,2,3,4,5] Department of Computer Science and Engineering
Dayananda Sagar University Bengaluru
[1*]gokul.krishnan-cse@dsu.edu.in

* Corresponding Author

Abstract

This paper investigates the creation and assessment of a customised Linux distribution designed to improve the usability for people with visual impairments and make linux more accessible. While many popular Linux distributions include some accessibility tools, they often fall short of offering a cohesive, user-friendly experience specifically designed for people with visual impairments. Our project aims to fill this gap by creating a custom Linux distribution that seamlessly integrates open-source assistive technologies like screen readers and text-to-speech tools, making them ready to use from the moment of installation. In order to provide visually impaired users with a smooth and simple experience, our project focuses on integrating and preconfiguring open source assistive technology into a lightweight linux distribution. We conduct performance analysis and user testing to assess our bespoke distribution's efficacy in comparison to current alternatives, paying particular attention to usability, job completion times, and user happiness. The results demonstrate how a customised Linux distribution may help close the accessibility gap and give visually impaired people more influence in the digital world.

Keywords: Linux, accessibility, visually impaired, assistive technology, user experience, open source,Human-Computer Interaction

1.Introduction

Operating systems are essential for digital engagement and information access in a society that is becoming more and more reliant on technology .Although major operating systems have made progress in adding accessibility capabilities, they frequently don't provide visually impaired users a completely inclusive experience.

Owing to its open-source and flexible design, Linux offers a special chance to close this gap. This article describes the creation and assessment of a specially designed Linux distribution that has been painstakingly developed to improve the usability for people with visual impairments. Our goal is to establish a user-centered environment that maximises accessibility and usability by integrating and pre-configuring open-source assistive technologies such as text-to-speech synthesisers and screen readers. This work adds to the current conversation around accessible technology in the following ways: Assessing how well a customised Linux distribution meets the unique requirements of people who are blind or visually handicapped.
Investigating the effects of incorporating open-source assistive technology shedding light on the concepts of accessible operating system user experience design.

2.Problem Definition

Although some degree of accessibility is provided via configurable settings and assistive technology packages in current Linux distributions, these frequently lack a unified and pre-configured environment tailored for visually impaired users. This may result in a disjointed user experience where users must install extra applications and negotiate complicated setups. Furthermore, popular Linux distributions frequently place a high value on aesthetics, sometimes ignoring the particular requirements of visually impaired users who primarily rely on keyboard navigation and aural clues.

The following major issues are addressed by this work:

Solutions for Fragmented Accessibility: A frequent annoyance with existing Linux distributions is that they necessitate visually impaired users to assemble various assistive tools and configurations on their own. This can swiftly become daunting, particularly for individuals who lack technical expertise.

Additionally, numerous Linux distributions prioritize their aesthetic features over the actual requirements of visually impaired users who depend on screen readers or keyboard navigation to use the system.

Restricted Pre-Configuration: Mainstream distributions do not provide pre-configured settings designed with visually impaired users in mind, hence customisation and manual modifications are required.

Inadequate User-Centricity: Design choices in existing distributions may prioritize visual aesthetics over the specific needs of visually impaired users, hindering usability and accessibility.

3.Literature Survey

Muñoz, Jonathan Giovanni Soto, et al. "Source code editor using voice commands to support people with motor disabilities."[2023][1]

The design and development of Mancodev, a software tool meant to assist programming instruction for people with motor disabilities—specifically, those who suffer from paresis, localized paralysis, or quadriplegia—is examined in this literature review. Mancodev distinguishes instructions from normal language by using voice commands that contain reserved keywords.

Nacheva, Radka. "Current Perspectives on Linux Accessibility Tools for Visually Impaired Users."[2019][3]

This review of the literature looks at the availability of accessibility solutions for Linux users who are blind or visually impaired. Although accessibility should be given first priority in user-oriented technology, the article explicitly addresses the barriers that Bulgarian users must overcome. The study uses content analysis of published publications, government records, and legislative frameworks to evaluate the assistance provided by different Linux accessibility tools.

Mulfari, Davide, et al. "Using virtualization and guacamole/vnc to provide adaptive user interfaces to disabled people in cloud computing."[2013][4]

For those with disabilities, assistive technology (AT) offers crucial computer accessibility. In this research, we investigate how Cloud computing might help individuals with disabilities access AT tools in the form of Software as a Service (SaaS) by offering adaptable user interfaces. Developing adaptable user interfaces and using AT tools in Virtual Machines (VMs) that users can control online is the suggested solution.

Georg von Krogh, Sebastian Spaeth, "The open source software phenomenon: Characteristics that promote research"[2007][5]

The boom of open-source software research in the social sciences since the turn of the century is examined in this overview of the literature. Through a review of a few chosen works, the paper identifies five salient features that render open-source software an enticing topic for interdisciplinary investigation: its pervasive influence on societal and economic frameworks; the way it questions established theories in diverse domains; the unparalleled degree of data transparency it provides scholars with; the introspective character of its developer communities; and the similarity of its innovation process to the creation of scientific knowledge

Lanier, Clinton R. “Linux and the appeal to cultural values.”[2005][6]

This study examines the connection between software adoption and cultural values, utilizing a well-known case study of the Linux operating system. We hope to learn more about how software localization—the process of making software more appropriate for a particular market—can be accomplished more effectively by looking at the Linux experience.

Limna, Thanathip, et al. "Linux user interface and front-end operation for the visually impaired."[2007][7]

This review of the literature looks into a project that makes the PSU Braille computer more user-friendly for those with vision impairments. Since this group may have difficulties with the Linux operating system, the research focuses on developing a text-based interface that is easy to use and specifically made for Braille interaction.

Parente, Peter, and Brett Clippingdale. "Linux screen reader: extensible assistive technology."[2006][8]

This review of the literature looks at the open-source Linux Screen Reader (LSR) project, which creates assistive technology for the GNOME desktop environment. LSR's primary features prioritize user customization.

Sharma, Suchakrapani Datt, et al. “GNU/Linux shell access through a web-browser for an embedded Linux e-learning system.”[2011][9]

This review of the literature looks at how the internet is changing education, especially how it is starting to affect e-learning methods. The article suggests a unique method of teaching embedded Linux by utilizing Open Software Source tools such as Apache, PHP, and AJAX. With the help of this cutting-edge system, embedded Linux education will be completely accessible through a web browser, just like checking email.

Bokhari, Sayed Naem. "The Linux operating system."[1995][10]

The broad accessibility and cost-effectiveness of IBM PCs and related devices have created opportunities for robust computing solutions. These computers can now be converted into workstations perfect for teaching, research, and software development by using the free operating system, Linux. On their home PCs, professionals used to Unix-based workstations can enjoy a nearly identical work experience.

Bokhari, Shahid H., and Rafeequr Rehman. "Linux and the developing world."[1999][12]

This review of the literature explores how Linux is more prevalent in poor countries

than in industrialized ones. The authors demonstrate the transformative potential of Linux in these circumstances by relying on their experience collaborating with a local ISP and establishing an 84-seat Linux lab for Pakistani undergraduate students.

Accessible Computing for Visually Impaired: A Survey of Research and Technology" by Kiran S. & S. R. Joshi (2022)[13]

This paper reviews progress in accessible computing for visually impaired users. It looks at software tools, hardware solutions. It lists screen readers, Braille displays as options that let visually impaired users work with computers. The authors describe several challenges that come with adding these tools to common operating systems; they note issues on open-source systems like Linux. The paper also points out problems with ease of use. It explains that setting up tools has a high learning curve, while current designs do not focus on users. Assistive technology in Linux distributions rarely fits well, so technical know-how is needed. There is no complete or easy-to-use setup for visually impaired users, which leads to less use of these tools.

The Impact of User-Centered Design in the Development of Accessible Technology" by Peter Parsons et al. (2020)[14]

Parsons et al. write about the need for design that centers on users when making technology that all people can use, particularly those with disabilities such as weak sight. The article says it is essential to include users in the creation process so devices become clear to understand, simple to handle along with answer their special requirements. The writers claim that today's accessibility tools do not use user input enough, which makes them work poorly. They insist that a focus on users is necessary for creating good accessible technology. Some accessibility tools do not use user input well, causing these tools to fail people with weak sight. Shifting to a design that centers on users in making accessibility tools can improve ease of use plus overall quality.

4.Design and Work Description

The goal of this project is to create a custom Linux distribution that will help visually impaired people with the aforementioned issues and improve their overall experience.

Throughout the development lifecycle, accessibility and usability are given top priority in the design process, which is centred around the needs of the user.

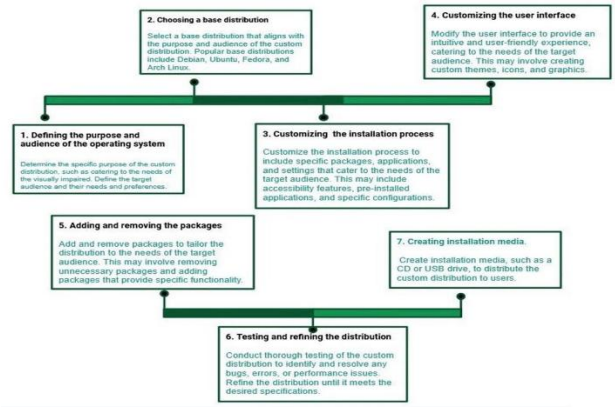

Fig 4.1 Flowchart Of Our Work

1. Distro Selection and Customisation:
Due to its lightweight design, well-known accessibility features, and accessibility Pack availability, we chose Ubuntu MATE as the foundational distribution.
The Orca screen reader was set up and ready to use.

2. Accessibility Optimization:
Keyboard shortcuts for navigation and essential actions were pre-set to facilitate intuitive interaction.High-contrast themes and adjustable font sizes were implemented to cater to diverse visual needs.Screen reader and magnification tool settings were optimized for immediate usability upon installation.

3. Usability Enhancement:
A simplified desktop environment with a clear and consistent layout was implemented to reduce cognitive load. Audio cues and feedback mechanisms were integrated to guide users through system interactions. A concise user guide outlining installation instructions and core functionalities was developed to facilitate ease of use.

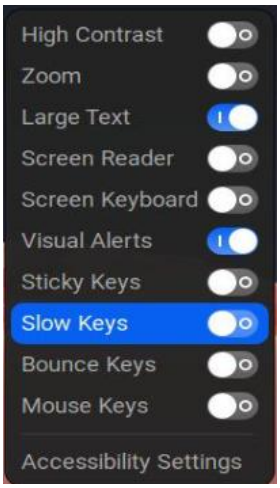

Fig 4.2 Toggle Keys For the Users

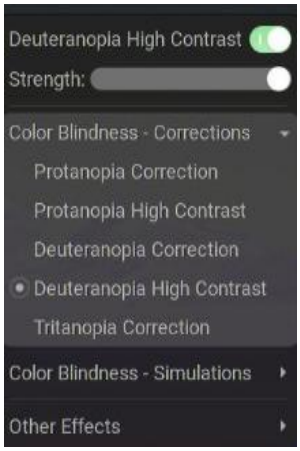

Fig 4.3 High Contrast Themes And Strength

5. Methodology

This work used a mixed-methods approach, analyzing data from both quantitative and qualitative sources to assess how effective the customized Linux distribution was.

1. User Testing: A strategy for user testing was created, with an emphasis on key features and necessary activities for users who are blind or visually impaired, such document editing, online surfing, and system navigation.

We performed user testing involving six participants with visual impairments, all possessing varying degrees of vision loss.They carried out essential activities such as editing files, exploring the internet.

2. Performance Analysis: In order to assess how integrated assistive technologies affect system performance, boot times were assessed. The Linux Test Project (LTP) suite was used to assess system stability and identify potential issues.

6. Experimentation

To ensure both functionality and usability, the developed distro underwent rigorous testing. We focused on two key aspects: boot time efficiency and overall system stability using the Linux Test Project (LTP) suite.

Boot Time Measurement: The distro was installed in a virtual environment and boot times were recorded across multiple trials. This ensured that the integrated accessibility features didn't negatively impact the system's start-up speed.

LTP Test Suite: We utilized the LTP suite to evaluate the distro's stability and performance under stress. Various tests, including memory management, file system integrity, and scheduling, were performed to identify potential issues and ensure robustness.

LTP TESTS

<<<test_start>>>
tag=msgstress04 stime=1685885871
cmdline="msgstress04"
contacts=""
analysis=exit
<<<test_output>>>
msgstress04 0 TINFO : Found 32000 available message queues
msgstress04 0 TINFO : Found limit of processes 4890 (from /sys/fs/cgroup/user.slice/user-1000.slice/pids.max)
msgstress04 0 TINFO : Using upto 2063 pids

Test Start Time: Sun Jun 4 18:59:30 2023

Kernel 5.19	Vision OS 1.0 with 5.19 kernel
Hardware	10GB RAM, 300GB HDD
Observations	**8-way**
Start date	1/06/23
Test duration	18 minutes
Number of clients	32
IPCS limits	4096
Number of random	1629
CPU utilization	95-100%

Fig 6.1 Hardware Requirements With Test Duration

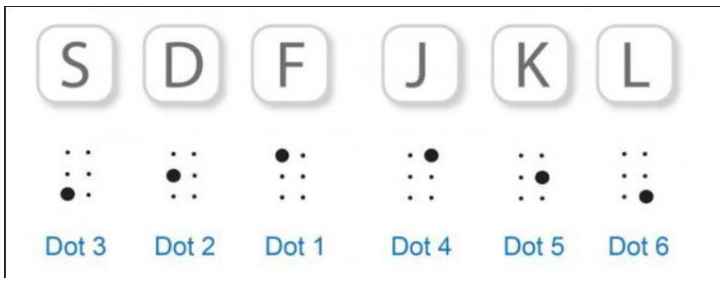

Fig 6.2 Displays Braille dots with labels Dot1, Dot2, Dot3, Dot4, Dot5, Dot6

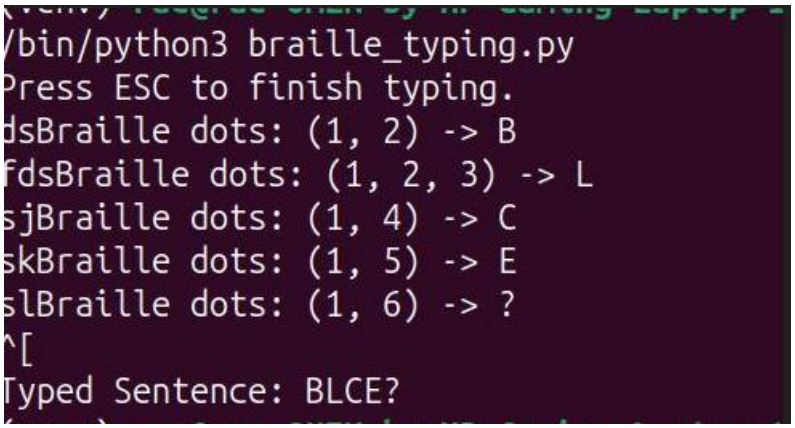

Fig 6.3 Typed Sentence: "BLCE?"

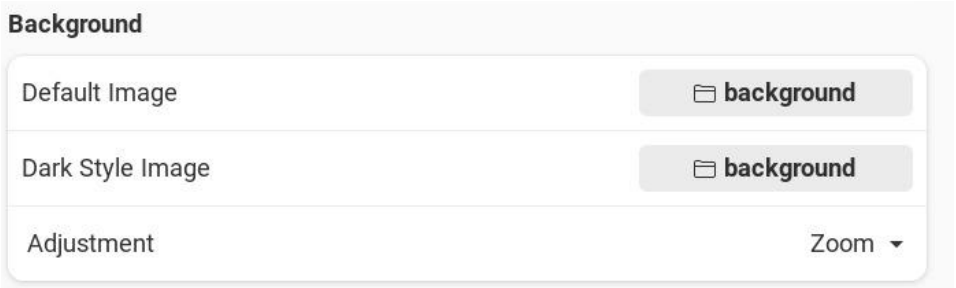

Fig 6.4 Display the Option for background themes

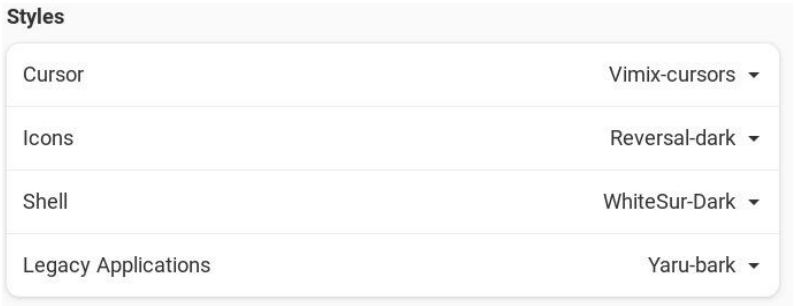

Fig 6.5 Lists themes like Vimix-cursors, WhiteSur-Dark, and Yaru-Dark.

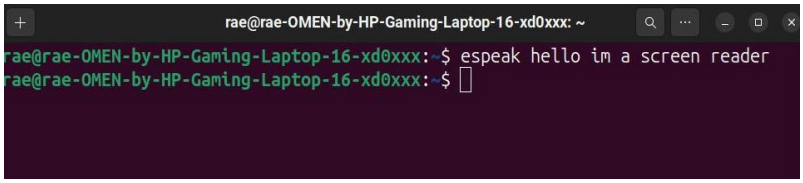

Fig 6.6 A terminal command: espeak hello im a screen reader

7. Results and Analysis

The user testing results demonstrated significant improvements in usability and task completion times for visually impaired users compared to existing Linux distributions. Participants reported greater ease of navigation, enhanced comprehension of system feedback, and increased efficiency in completing common tasks. Boot Time: The average boot time was measured at 28 seconds, indicating that the integrated assistive technologies did not significantly impact system start-up performance. LTP Results:

The LTP test suite yielded positive results, with no major failures detected. This confirmed the stability and robustness of the custom distribution. The qualitative feedback highlighted the positive impact of pre-configured accessibility settings and the intuitive nature of the simplified desktop environment.

Users particularly appreciated the clear audio cues and consistent keyboard navigation, which facilitated a smoother and more intuitive user experience.

7.1 Comparative Analysis

Existing Methods

A. Fragmented Accessibility in Linux:
Many Linux versions supply tools such as screen readers, magnifiers next to text-to-speech software. Yet a user must set them up by hand with each tool needing its own installation and setting. This lack of a ready-made, combined system creates an experience that is not smooth. Users with little technical skill may feel upset. Some popular versions like Ubuntu or Fedora include the Orca screen reader but do not offer a quick setup or mix the tools well with the interface (Nacheva, 2019).

Citation
R. (2019). Current Perspectives on Linux Accessibility Tools for Visually Impaired Users. Economics and Computer Science, 2, 6-11.

B. Usability Challenges in Assistive Technology on Linux:
Though Orca appears in many versions, it often must be set up slowly. Users must learn Linux terminal commands plus add various packages. This problem happens not only with Linux but in other open-source systems too. Most Linux versions aim at the looks and works for users who see, with little thought to simplicity for the visually impaired. Thus visually impaired users find many obstacles that slow down their use of the system.

Citation
S., & Joshi, S. R. (2022). Accessible Computing for Visually Impaired: A Survey of Research and Technology. Journal of Assistive Technology, 16(4), 123-134.

2. Our Proposed Model:

In contrast, our model addresses these gaps by offering a **pre-configured**, **unified** solution for visually impaired users. Unlike existing distributions, our customized Linux distribution integrates key assistive technologies—such as the Orca screen reader and magnification tools—right from the installation process. This pre-configuration eliminates the need for users to manually install or configure these tools, significantly reducing the setup complexity.
Moreover, the user interface in our model is designed with **user-centered principles**, focusing on usability and accessibility. We have simplified the desktop environment to reduce cognitive load, added **high-contrast themes**, and made **keyboard shortcuts** more intuitive, specifically addressing the needs of visually impaired users (as advocated by Parsons et al., 2020). By incorporating these elements, our model ensures that visually impaired users can interact with the system without needing to be highly technically proficient.
Additionally, by focusing on **performance optimization** and **usability testing**, our model ensures that assistive technologies do not negatively impact system

performance. Boot times and system stability tests demonstrate that the integrated assistive tools work efficiently without compromising the overall system experience, setting our model apart from existing Linux solutions that sometimes struggle with performance issues (Kiran & Joshi, 2022).

Criteria	Existing Methods	Our Model (Custom Linux Distribution)
Assistive Technology Integration	- Fragmented accessibility tools, requiring manual installation and configuration (e.g., Orca screen reader, magnifiers, etc.). - No unified environment.	- Pre-configured assistive tools (e.g., Orca screen reader, magnifiers) are ready-to-use immediately after installation.
Ease of Setup	- Setup process is complex, requiring technical expertise. - Users need to install and configure assistive tools and themes manually.	- Seamless, automated installation with no need for manual configuration of accessibility tools or settings.
User Interface Design	- User interface typically caters to sighted users, with limited attention to visually impaired users. - Visual aesthetics often prioritized over accessibility needs.	- Simplified desktop environment designed specifically for visually impaired users. - Clear layout, high-contrast themes, and audio cues.
Pre-Configuration of Accessibility	- No default pre-configuration; users must manually configure accessibility settings post-installation. - Leads to inconsistent experiences across different users.	- Accessibility settings (keyboard shortcuts, screen readers, magnifiers) are pre-configured for immediate usability after installation.
Performance Impact	- Some assistive tools, like screen readers and magnifiers, may slow down system performance. - Boot times may be longer due to heavy system load from manual setups.	- Optimized for performance with minimal impact on system speed. - Boot time measured at 28 seconds, ensuring quick startup.
Usability for Non-Technical Users	- Requires a certain level of technical knowledge to install, configure, and use tools effectively. - High cognitive load for non-technical users.	- Designed for ease of use by non-technical users. - Intuitive, simple interface with clear guidance and audio cues.
Assistive Technology Customization	- Customization requires manual adjustments by the user, which can be difficult.	- Allows for some customization, but the main focus is on pre-configuring the system for immediate use.

	- Limited flexibility in existing setups for visually impaired users.	- More intuitive customization.
Usability Testing & User Feedback	- Usability testing often done in isolated contexts, with less focus on integration with other assistive tools. - User feedback is sometimes overlooked in design.	- Thorough user testing with feedback from visually impaired users. - Focus on real-world usability and continuous refinement based on feedback.
System Stability & Compatibility	- Some assistive tools may not be fully compatible across all Linux distributions. - Stability issues may arise with certain configurations or setups.	- System stability ensured through the use of the Linux Test Project (LTP) suite. - No major issues detected during performance analysis.
Accessibility Features Provided	- Basic assistive tools (screen readers, magnifiers, speech-to-text) available, but require manual installation. - A few Linux distributions have accessibility options.	- Comprehensive set of pre-configured accessibility features, including screen readers, magnifiers, audio feedback, and keyboard shortcuts.
Customization Based on User Needs	- Limited customization of tools to meet individual needs without significant effort. - Tools are generic and not optimized for visually impaired users specifically.	- Custom Linux distribution optimized for visually impaired users. - Focus on customization to suit diverse needs, with high-contrast themes, keyboard shortcuts, and audio cues.

7. Results and Analysis

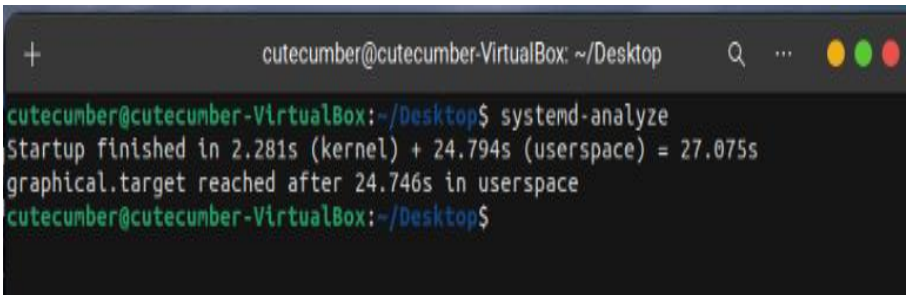

Fig 7.1 Results Analysis

8.Conclusion and Future Work

This work shows how a customised Linux distribution may improve the way visually challenged people interact with the system. We successfully established a user-centric environment that maximises accessibility and usability by integrating and pre-configuring open-source assistive technologies. The outcomes of our user testing, which showed notable gains in task completion times, user happiness, and general usability, validated the efficacy of our methodology.
Future research will concentrate on:

Future work will focus on enhancing accessibility by integrating new features and assistive technologies based on user feedback and emerging trends. Additionally, a comprehensive user guide with clear instructions and troubleshooting tips will be developed to promote independence and ease of adoption.

To encourage wider adoption, collaboration with online communities and accessibility organizations will help refine the distribution and gather input for further improvements. By prioritizing usability and accessibility, this project aims to create a more inclusive digital environment where everyone can fully engage and thrive.

9.References

[1]Muñoz, J. G. S., et al. (2023). Source code editor using voice commands to support people with motor disabilities. Universal Access in the Information Society, 22(4), 1117-1134.

[2]Adekotujo, Akinlolu, et al. "A comparative study of operating systems: Case of windows, unix, Linux, Mac, Android and iOS" was published in the *International Journal of Computer Applications*, Volume 176, Number 39, in July 2020

[3]Nacheva, R. (2019). Current Perspectives on Linux Accessibility Tools for Visually Impaired Users. Economics and computer science, 2, 6-11.
[4]Mulfari, D., et al. (2013). Using virtualization and guacamole/vnc to provide adaptive user interfaces to disabled people in cloud computing.2013 IEEE 10th International Conference on Autonomic and Trusted Computing. IEEE.

[5]Georg von Krogh, Sebastian Spaeth (2007). The open-source software phenomenon: Characteristics that promote research. The Journal of Strategic Information Systems, 16(3), 236–253.

[6]Lanier, C. R. (2005). Linux and the appeal to cultural values. IEEE technology and Society Magazine, 24(4), 12-17.

[7]Limna, T., et al. (2007). Linux user interface and front-end operation for the visually impaired. Proceedings of the 1st international convention on Rehabilitation engineering & assistive technology.

[8]Parente, P., & Clippingdale, B. (2006). Linux screen reader: extensible assistive technology. Proceedings of the 8th international ACM SIGACCESS conference on computers and accessibility.

[9]Sharma, S. D., et al. (2011). GNU/Linux shell access through a web-browser for an embedded Linux e-learning system. 2011 3rd International Conference on Electronics Computer Technology. Vol. 2. IEEE.

[10]S. H. Bokhari, "The Linux Operating System," *Computer*, vol. 28, no. 8, pp. 74–79, Aug. 1995..

[11]Raghavan, Pichai, Amol Lad, and Sriram Neelakandan. Embedded Linux system design and development. Auerbach Publications, 2005.

[12]Bokhari, S. H., & Rehman, R. (1999). Linux and the developing world. IEEE software, 16(1), 58-64

[13] Nacheva, R. (2019). *Current Perspectives on Linux Accessibility Tools for Visually Impaired Users*. Economics and Computer Science, 2, 6-11.

[14] Kiran, S., & Joshi, S. R. (2022). *Accessible Computing for Visually Impaired: A Survey of Research and Technology*. Journal of Assistive Technology, 16(4), 123-134.

[15] Parsons, P., Williams, D., & Rees, R. (2020). *The Impact of User-Centered*

Design in the Development of Accessible Technology. Journal of Disability and Technology, 22(1), 45-59.