



MSc in Computer Science - Team Project

Interim Report

Team Name:

Byte The Barrier

Joel Felix Quadras (D22125093 - DS)

Cheril Mariam John (D22124272 - ASD)

Mountdenyraj Chelladurai Nadar (D22124430 - ASD)

David Ayang (D21127639 - ASD)

Table of Contents

Introduction.....	4
1. User Scenario: The Characters.....	5
1.1. Who is our target user?.....	5
1.1.1. User Personas.....	5
1.2. Why Are They Important?.....	7
1.3. What Problem Are We Solving for Them?	8
2. Technical Problem: The Setting.....	9
2.1. Reasons For Building This Application	9
2.2. Core Technical Problems.....	10
2.2.1. Overview	10
2.2.2. User Interface/User Experience.....	11
2.2.3. Architecture and Hosting Platform	11
2.2.4. Document Optimization Features Selection.....	12
2.2.5. Review of Similar Tools.....	13
2.2.6. Comparison of Similar Tools to the Accessibilator	15
3. Technical Solution: The Plot.....	16
3.1. What does our system do?	16
3.2. How does the Accessibilator work?.....	18
3.2.1. Front-end: Technologies, User Interface Components and Mock-ups.....	19
3.2.2. Back-end: Technical components	25
4. Evaluation: The Reviews.....	32
4.1. What does success look like for our system?	32
4.2. How will we evaluate the system that we built?	33
4.2.1. User-Centric Evaluation:.....	33
4.2.2. Technical Evaluation:.....	34
5. Conclusion: The Plan.....	35
5.1. What is our project management strategy?	35
5.1.1. Team Communication.....	36

5.2. What are the biggest challenges we are currently facing?	36
5.2.1. Technical Challenges:.....	36
5.2.2. Team-Oriented Challenges:.....	37
<i>References</i>	40
<i>Appendix</i>	43

Introduction

In today's digital age, the concept of accessibility has extended beyond physical spaces to include digital platforms, making it an imperative need. The Accessibilator project was born out of this necessity, aiming to bridge the gap between standard document formats and the unique needs of individuals with learning disabilities, with an initial focus on Dyslexia.

Dyslexia is a neurological condition that affects roughly 10% of the global population and presents unique challenges in areas such as phonology, orthography, and syntax. This often results in prolonged reading times and reduced comprehension, highlighting the need for specialised document formatting. The project ideally aligns with the theme of "social good," as it not only enhances individual user experience but also contributes to creating a more inclusive digital society.

The primary function of the Accessibilator is to transform a variety of documents—ranging from PowerPoint presentations to Word documents and Excel spreadsheets—into formats that are highly accessible. Upon uploading a document, the system analyses it for potential issues and proactively suggests actionable improvements. Users have the final say in adopting these suggestions, ensuring that the document aligns better with their reading preferences.

Compliance with existing legal frameworks like the Americans with Disabilities Act (ADA), the European Accessibility Act, and the United Nations Convention on the Rights of Persons with Disabilities is a cornerstone of this project. The application aims to not just meet but understand the implications of these laws, thereby setting a standard in digital accessibility.

This report aims to offer a comprehensive overview of the Accessibilator, detailing its objectives, methodologies, and future directions. It serves as a lens into the world of digital accessibility, providing valuable insights into how technology can be leveraged for social good.

1. User Scenario: The Characters

1.1. Who is our target user?

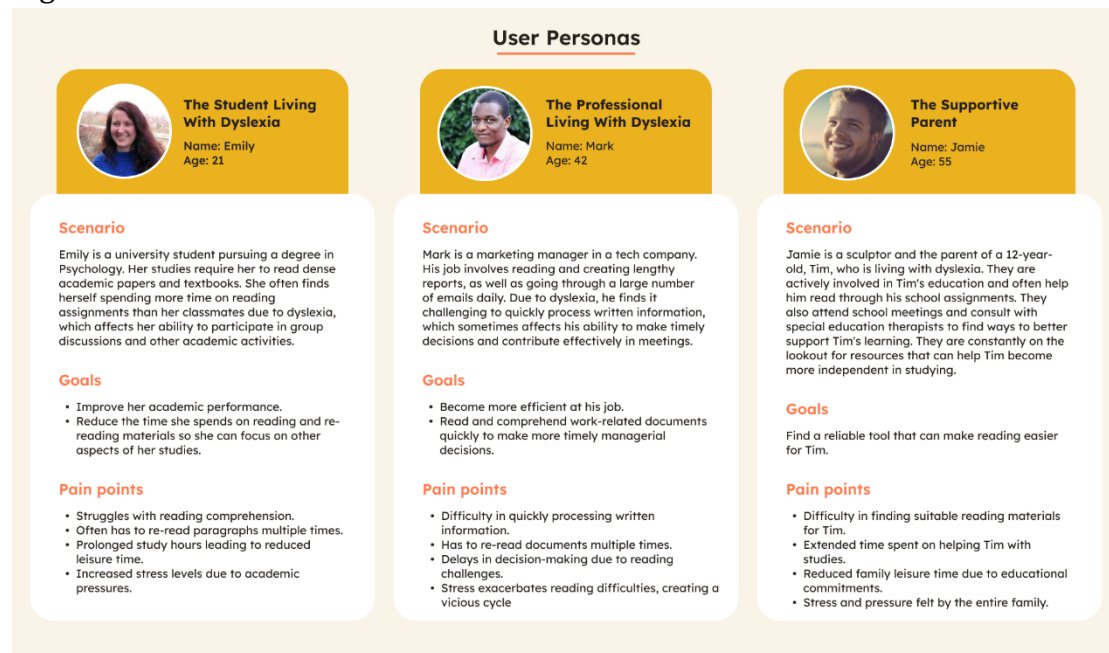
Our target users are primarily individuals with dyslexia, and also those who support them, such as parents, educators, and caregivers, as well as those who want to learn about that condition.

These user groups help in acknowledging and comprehending the unique demands of individuals who have dyslexia as well as those who may know or assist them. Our primary designated user group consists of members from the Dyslexia Association of Ireland. We have maintained ongoing communication with them to ensure their active involvement through surveys and feedback, in the development and testing of our system.

1.1.1. User Personas

Our user personas were identified through a comprehensive review of academic literature as well as seeking expert opinion from the Dyslexic Association of Ireland. For instance, Nevill and Forsey (2023) highlighted the unique challenges faced by dyslexic students in educational settings, such as reduced comprehension and prolonged reading times. Almgren Bäck *et al.* (2022) discussed the long-term experiences of dyslexic students using assistive technology, emphasizing the need for user-friendly and effective solutions. Another study by Kennecke *et al.* (2021) underscored the importance of accessible digital services, particularly for dyslexic users who often struggle with text-based interfaces. By taking these findings into consideration, we have created well-rounded personas like Emily, a university student; Mark, a marketing manager; and Jamie, a supportive parent, to ensure that the Accessibilator adequately serves the needs of its intended user base.

Fig 1: User Personas



Persona 1: Emily, The Student Living with Dyslexia

Age: 21

Occupation: University Student in Psychology

Scenario: Emily's studies require her to read dense academic papers and textbooks. Her dyslexia often leads her to spend more time on reading assignments than her classmates.

Goal: Improve academic performance and reduce time spent on reading.

Pain Points: Struggles with reading comprehension, prolonged study hours, and increased stress levels.

Persona 2: Mark, The Professional living with Dyslexia

Age: 42

Occupation: Marketing Manager

Scenario: Mark's job involves reading and creating lengthy reports and emails. His dyslexia affects his ability to process written information quickly.

Goal: Become more efficient at work and make timely managerial decisions.

Pain Points: Difficulty in processing written information, delays in decision-making, and stress.

Persona 3: Jamie, the Supportive Parent

Age: 55

Occupation: Sculptor

Scenario: Jamie is actively involved in their 12-year-old son Tim's education. They often help Tim read through his school assignments and consult with special education therapists.

Goal: Find a reliable tool to make reading easier for Tim.

Pain Points: Difficulty in finding suitable reading materials, extended time spent on helping Tim, and reduced family leisure time.

1.2. Why Are They Important?

Dyslexia comes in many different flavours, and pictured below are some common manifestations of it:

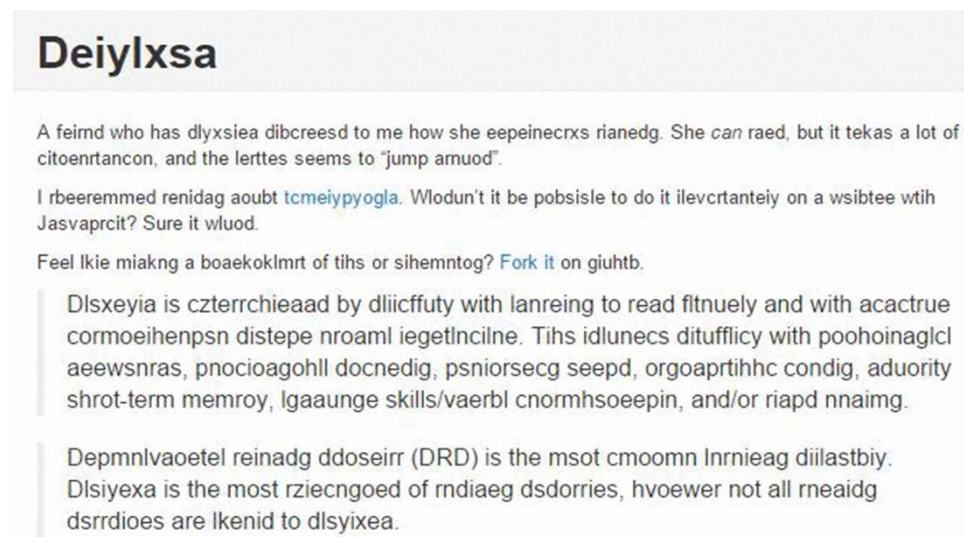


Fig 2: Type of dyslexia

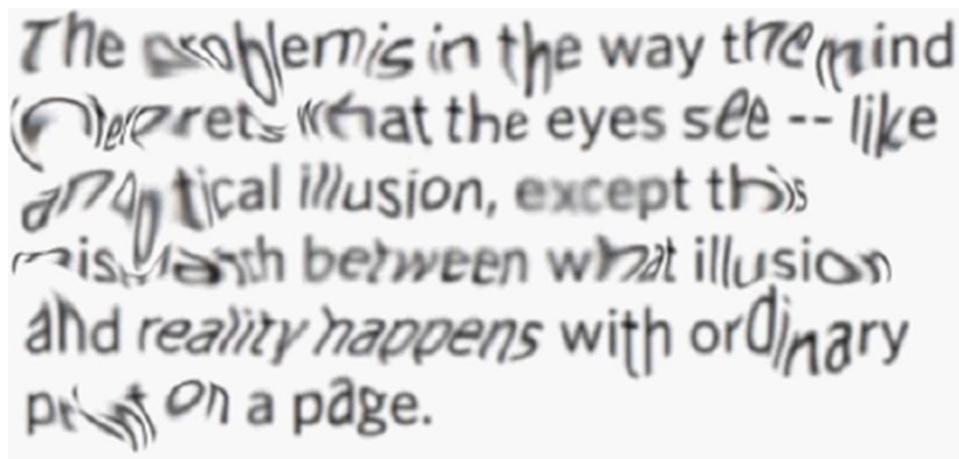


Fig 3: Type of dyslexia

Understanding the needs of these specific personas is crucial for the success of the Accessibilator. From our research, those who were living with dyslexia used practices of skipping and skimming sections of text that posed comprehension challenges, as well as the act of re-reading to solidify understanding. The survey participants with dyslexia consistently reported higher frequencies of these behaviours in comparison to their counterparts without dyslexia (Grusky *et al.*, 2020).

People living with Dyslexia like Emily and Mark represent a significant portion of the population who face unique challenges in academic and professional settings (Nevill & Forsey, 2023). By catering to their needs, we contribute to a more inclusive society where information is accessible to all, regardless of their neurological conditions. Jamie, the supportive parent, represents another crucial user group. Their role is often overlooked but is essential in the support system for those with dyslexia (Almgren Bäck *et al.*, 2022). By making resources more accessible for them, we indirectly improve the quality of life for the dyslexic individuals they care for.

1.3. What Problem Are We Solving for Them?

For Emily, the Accessibilator aims to alleviate the academic challenges highlighted by Nevill & Forsey (2023), such as prolonged reading times and

reduced comprehension. The application will offer features like font size, contrast and colour correction, advanced text optimizations like sentence chunking and a dyslexia ruler to improve her reading experience. Mark faces similar challenges but in a professional setting. The Accessibilator will help him become more efficient at his job by offering quick and effective document optimization solutions, thereby enabling him to make timely managerial decisions (Kennecke *et al.*, 2021). For Jamie, the platform will serve as a valuable resource for producing dyslexia-friendly reading materials for Tim, thereby reducing the emotional and time investment required from them (Almgren Bäck *et al.*, 2022).

By addressing these specific needs, the Accessibilator aims to set a new standard in digital accessibility, ensuring that no one is left behind. We have learned from our research, that the consequences of dyslexia extend beyond academic difficulties. Individuals with dyslexia often experience prolonged reading times and reduced comprehension levels, leading to the need for repeated readings. This struggle can lead to heightened levels of stress, anxiety, and depression. Moreover, the societal stigma surrounding dyslexia can exacerbate emotional and psychological burdens, affecting self-esteem and motivation. It is our aim to significantly improve this population's reading experience using various document optimization techniques. Our web application offers a user-friendly platform that allows individuals to upload documents with accessibility challenges. The system then automatically makes the necessary adjustments to optimize readability.

2. Technical Problem: The Setting

2.1. Reasons For Building This Application

In accordance with the definition provided by the *International Dyslexia Association* (IDA), dyslexia is identified as a specific neurobiological-based

learning disability. It leads to difficulties in recognizing words accurately and smoothly and in spelling and decoding. These issues are mostly caused by a phonological language impairment which causes individuals with dyslexia to face challenges in understanding what they read and may have a less fulfilling reading experience. The prevalence of dyslexia can differ based on its definition, with reported rates varying from 4% to 20% (Butterworth, B., & Kovas, Y., 2013; Coles, G., 1999; Shaywitz, S., 1996, 2005; Siegel, L., 2006; Snowling, M., 2010).

Even though there are many accessibility-enhancing systems in the market, all of them cater to real-time reading enhancements. One major difference between the existing systems is that our application provides the user with the ability to export the reformatted document which they can re-read at their own convenience. Most of the existing systems offer design-based formatting like font, layout etc. Our system makes use of Data Science techniques to transform the content of the document as well as make it more optimized for readers with dyslexia.

The concept of the Accessibilator was conceived as a response to the need to make digital content more accessible, especially for individuals with disabilities. The various formats of documents such as .doc, .docx, .pdf, .ppt, .pptx, .xls, and .xlsx are being used in professional and academic environments. These formats weren't designed with accessibility in mind and this realization that such a vast amount of information is still inaccessible to many people has provided us with the motivation to develop the Accessibilator.

2.2. Core Technical Problems

2.2.1. Overview

A few technical issues must be resolved to build the Accessibilator, including file parsing, maintaining content quality after conversion, creating an easy-to-use user interface, and selecting a scalable and effective architecture. We explore these questions in more detail in the sections that follow, setting the stage for the

solutions that come next. The main technical challenge is that dyslexia is a complex learning disability, which means no two people might have the same pattern of the disability. This variation in the behaviour of dyslexia for everyone is challenging to resolve.

2.2.2. User Interface/User Experience

Given that the very ethos of Accessibilator is to enhance accessibility, the UI/UX holds paramount importance. The primary challenge here is to ensure that the tool itself is optimally accessible, catering to individuals with various disabilities, be it visual, auditory, motor, or cognitive. The interface needs to be intuitive, with logical navigation and appropriate feedback mechanisms. Visual elements should be compatible with screen readers, and the platform should allow for keyboard-only navigation, among other accessibility features. Moreover, ensuring the tool provides a seamless process of file uploading, conversion, and downloading, with guidance at each step, is crucial. We opted for a desktop-first UI design instead of a mobile-first approach as the functionalities of Accessibilator will be more applicable on a larger screen. Features like file uploading, conversion settings, and detailed guidance will be easier for users to manage on a laptop/PC screen. Simultaneously, we aim to achieve a responsive design that optimizes the user experience across all platforms which ensures that the tool remains accessible as new devices and technologies emerge in the future.

2.2.3. Architecture and Hosting Platform

With the collection of document formats and the varying sizes they come in, the architecture for the Accessibilator needs to be robust and scalable. Cloud-based architecture can be apt, considering the benefits of scalability, flexibility, and easy maintenance. The hosting platform should offer high uptime, fast processing speeds, and tight security to protect user-uploaded documents. Further challenges include efficiently parsing the myriad of document formats and ensuring that the converted document retains its fidelity, especially when

transforming intricate elements like tables, charts, and images into more accessible forms.

2.2.4. Document Optimization Features Selection

In selecting the core and advanced features for document optimization, a multi-pronged approach was adopted. First, a comprehensive review of research articles was conducted. For instance, a study by Williams et al. (2022) emphasized the importance of alt-text for images and graphics, while another by Putra *et al.* (2023) highlighted the significance of logical content flow and simple language for enhanced readability. This informed our decision to include features like paragraph splitting, text summarization, and simplification of jargon. Likewise, multiple studies (Galliusi et al., 2020; Rello & Baeza-Yates, 2016; Rello et al., 2012) substantiated that the readability experience for individuals with dyslexia is significantly influenced by elements such as font style, background colour as well as text layout. Second, feedback from potential users, especially those with disabilities, was sought to understand the real-world challenges they face. Finally, tech feasibility studies were conducted to determine which optimizations could be automated and which might require manual inputs. This triangulated approach ensured that the chosen features were both evidence-based and user-centric.

Font Type	Reading Time			Font Type	Fixation Duration			Font Type	Preferences Rating		
	\bar{x}	$\bar{x} \pm s$	%		\bar{x}	$\bar{x} \pm s$			\bar{x}	$\bar{x} \pm s$	
Arial	24.22	28.35 \pm 12.39	100	Courier	0.22	0.22 \pm 0.05		Verdana	4	3.79 \pm 0.98	
OpenDys	23.81	29.17 \pm 15.79	103	Verdana	0.22	0.23 \pm 0.07		Helvetica	4	3.62 \pm 1.08	
CMU	26.06	29.58 \pm 12.05	104	Helvetica	0.24	0.24 \pm 0.06		Arial	4	3.60 \pm 1.13	
Courier	29.73	29.61 \pm 10.87	104	Arial	0.23	0.24 \pm 0.07		Times	4	3.45 \pm 1.15	
OpenDys It.	25.44	29.68 \pm 14.44	105	Times	0.24	0.25 \pm 0.07		Myriad	3.5	3.40 \pm 0.99	
Helvetica	27.18	31.05 \pm 15.04	109	Myriad	0.25	0.25 \pm 0.07		CMU	3	3.31 \pm 0.98	
Verdana	28.97	31.16 \pm 13.03	110	Times It.	0.25	0.26 \pm 0.06		Courier	3	3.14 \pm 1.39	
Times	29.30	31.68 \pm 11.81	112	OpenDys	0.24	0.26 \pm 0.07		Arial It.	3	2.90 \pm 1.10	
Times It.	28.55	32.38 \pm 12.34	114	OpenDys It.	0.26	0.26 \pm 0.07		Times It.	3	2.86 \pm 1.20	
Myriad	26.95	32.66 \pm 14.80	115	Garamond	0.25	0.27 \pm 0.07		Garamond	2	2.57 \pm 1.15	
Garamond	30.53	33.30 \pm 15.45	117	CMU	0.25	0.27 \pm 0.08		OpenDys	3	2.57 \pm 1.15	
Arial It.	29.68	34.99 \pm 16.60	123	Arial It.	0.28	0.28 \pm 0.08		OpenDys It.	2	2.43 \pm 1.27	

Table 1: Median, mean and standard deviation of Reading Time and Fixation Duration in seconds as well as the median, mean, and standard deviation of the Preference Ratings. We include the relative percentage for Reading Time, our main readability measure, with respect to the smallest average value, Arial.

Fig 5: Good Fonts for Dyslexia' by Rello, L. & Baeza-Yates, R., from Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, 2013

2.2.5. Review of Similar Tools

After reviewing a few similar tools, we discovered some similarities and few variations across their set of features. Here is a comprehensive review of two these tools and an analysis of these tools against the Accessibilator.

1. Microsoft Office Accessibility Checker:

a. Features:

- i. Identifies issues: The tool automatically scans your Microsoft Office document (be it Word, PowerPoint, or Excel) and highlights the parts in the content which might be difficult for people with disabilities to access.
- ii. Task Pane: The task pane is present on the side of the Office application. It lists the issues which were identified in real-time, and it allows you to make changes to the document as you create the content without disrupting your workflow.

b. Pros:

- i. Integrated into Office Products: One of the biggest advantages is its seamless integration with Microsoft Office applications. This eliminates the need to use a separate tool, streamlining the process.
- ii. Easy to Use: The user-friendly interface and the task pane guide you through the corrections, often with explanations and direct links to help resources.

c. Cons:

- i. Limited to Microsoft Formats: The tool is limited to Microsoft Office files.
- ii. Not as Comprehensive: While it does a decent job for basic accessibility issues, it does not cover all the Web Content Accessibility Guidelines (WCAG) standards and might miss some nuanced problems that specialized tools could catch.

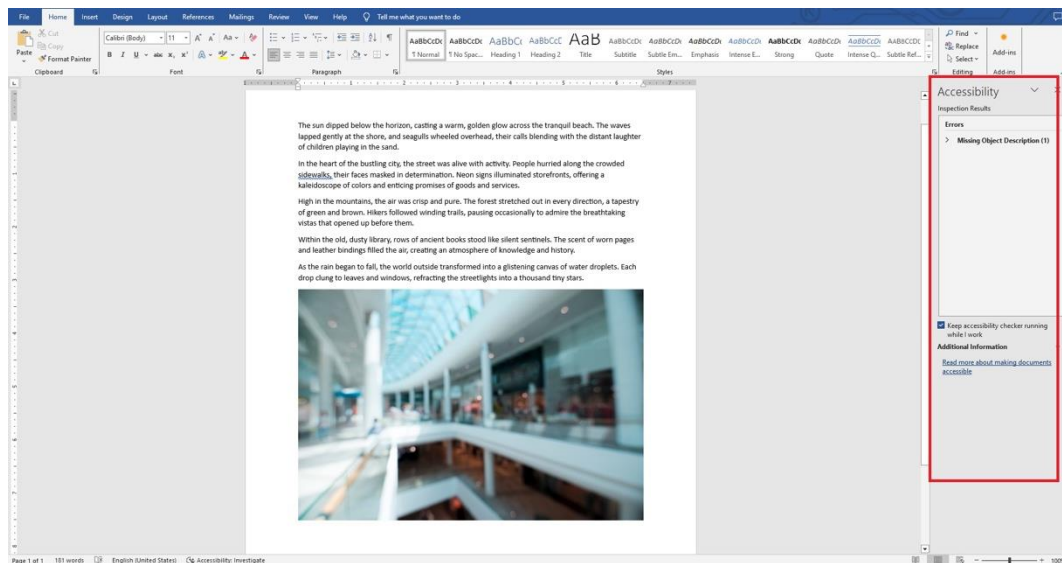


Fig 6: Microsoft Office Accessibility Checker Tool

2. WebAIM's WAVE Tool:

a. Features:

- i. **Web-based:** WAVE is a browser-based tool, allowing you to evaluate the accessibility of websites directly in your browser.
- ii. **Checks for Various Issues:** The tool looks for a range of accessibility issues like contrast errors, missing alt text for images, improper use of HTML elements, etc.
- iii. **Visual Representation:** It provides a visual overlay on the webpage, pointing out issues directly on the page elements.

b. Pros:

- i. **Free:** The tool is free to use, making it accessible to developers and content creators on a budget.
- ii. **Easy to Use:** With its visual interface, it is straightforward to spot and understand issues.
- iii. **Comprehensive:** It offers a more in-depth analysis compared to Microsoft Office Accessibility Checker and is designed to cover a broad range of WCAG guidelines.

c. Cons:

- i. **Limited to Web Content:** Unlike Microsoft Office Accessibility Checker, WAVE is specialized for web content and cannot be used for documents or other file types.
- ii. **False Positives:** Due to its automated nature, it may flag elements as issues that are not actually problems upon human review. Manual verification is usually required to sift through these.

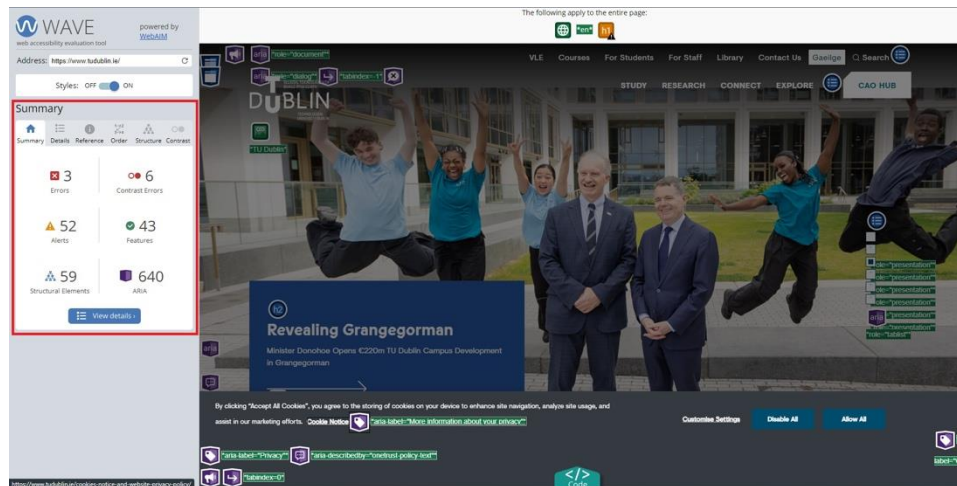


Fig 7: WebAIM's WAVE Tool

2.2.6. Comparison of Similar Tools to the Accessibilator

The above although good at what they do in increasing accessibility, here is a deeper comparison of their features against the features available with the Accessibilator.

1. **Comprehensiveness:** The Accessibilator suggests and implements changes such as colour contrast and font size through which it offers an operational level of comprehensiveness. It supports converting digital content to more accessible content. It is designed to work on multiple file types beyond Microsoft Office formats such as text documents and Rich text format documents (.rtf). The Microsoft Office Accessibility Checker is comprehensive for Microsoft formats, but it does not offer any checks for

other file formats. WebAIM's WAVE tool is highly comprehensive, but it is limited to web content, and it does not implement the suggested changes.

2. **Ease of Use:** The Microsoft Office Accessibility Checker is highly integrated and user-friendly within the Microsoft ecosystem whereas WebAIM's WAVE Tool is user-friendly but only limited to web browsers. The Accessibilator would be made available as a website which would be intuitive as we are developing, implementing, and analysing the UI/UX designs along with people who have dyslexia. It would be well-integrated into environments where it will be most used.
3. **Price:** The Accessibilator would be a free-to-use tool and it would be built and integrated with lots of features for people with an array of learning disabilities. The Microsoft Office Accessibility Checker is bundled with the price of Microsoft Office, so there is no additional cost whereas WebAIM's WAVE Tool is also free to use.
4. **Specialization:** The Accessibilator focuses on making changes to the documents for better accessibility, rather than just identifying issues whereas both Microsoft Office Accessibility Checker and WebAIM's WAVE Tool both specialize in identifying issues rather than fixing them. The Accessibilator is unique in that it not only identifies but also implements changes, making it specialized in a unique way to the other two tools. The significant advantage is that the implemented changes would be compliant with accessibility standards like WCAG.

3. Technical Solution: The Plot

3.1. What does our system do?

The Accessibilator is a comprehensive platform designed to enhance the accessibility of digital documents for individuals with dyslexia and other learning disabilities. The system offers a wide array of features, each meticulously designed to address specific challenges that dyslexic individuals face while interacting with digital content.

The platform allows users to upload various types of files, including but not limited to .doc, .docx, .xls, .xlsx, .ppt, .pptx, and .pdf. Once uploaded, the system performs an in-depth analysis to identify dyslexic-unfriendly elements such as font styles, sizes, and colours. It also examines the layout, including text justification and sentence length, and suggests improvements like adding more headings or a table of contents.

Moreover, the system employs machine learning techniques to offer advanced features like sentence chunking, abbreviation expansion, and jargon simplification. It even auto-generates a glossary for terms that cannot be simplified. Additional functionalities include a customizable dyslexia ruler, text-to-speech capabilities, and a focus mode for better readability.

The Accessibilator aims to make digital content not just accessible but also user-friendly, thereby empowering dyslexic individuals to engage with digital documents more effectively and efficiently.

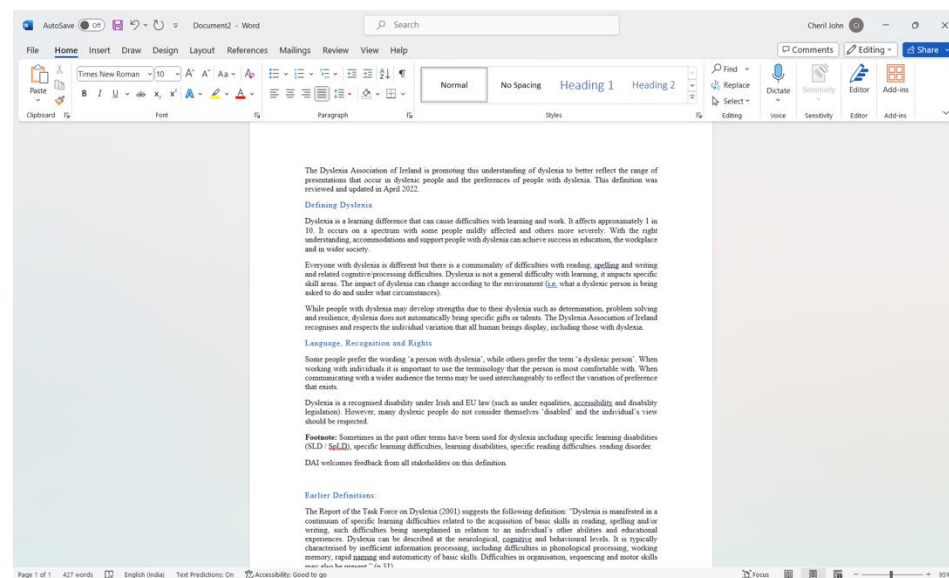


Fig 8: Document with poor accessibility

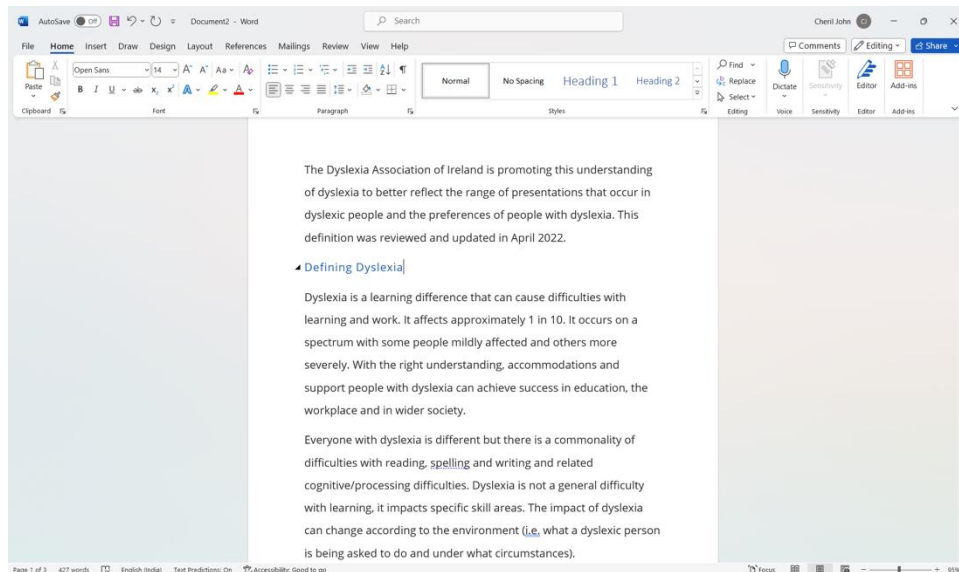


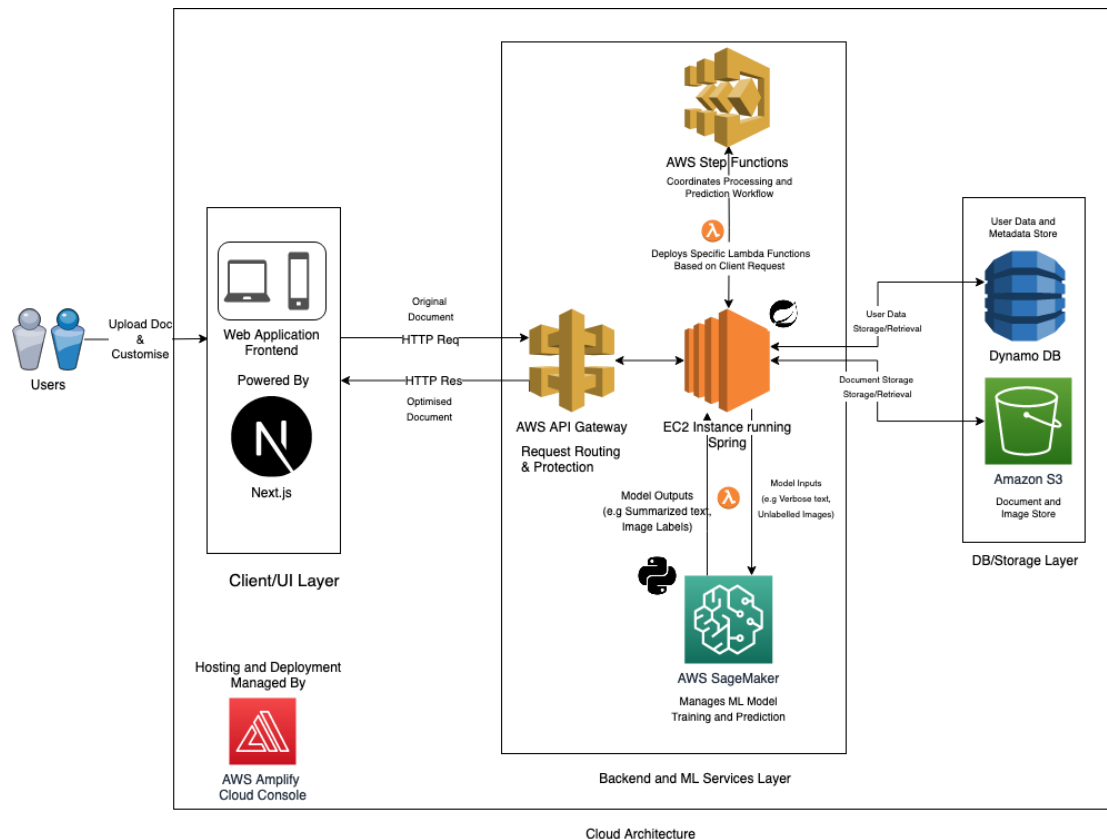
Fig 9: Document with improved accessibility

3.2 How does the Accessibilator work?

In our web application, the Accessibilator, the journey of a document from the user to the system and back is a well-orchestrated interplay of various tools and technologies. When a user uploads a document, it's initially sent to our Spring Boot application running on an AWS EC2 instance via a REST API. This backend is responsible for the initial parsing and analysis of the document, leveraging the Apache POI library for reading various file formats. Once parsed, the document's metadata is stored in an Amazon DynamoDB database, which allows us the flexibility to handle a wide range of document attributes.

For more complex tasks like machine learning-based optimizations, we employ AWS Step Functions to coordinate between different AWS services. The parsed document is sent to Amazon SageMaker, where Python-based machine learning models perform tasks like sentence chunking, heading suggestions, and more. After processing, the optimized document is stored in an Amazon S3 bucket, known for its high durability and security features.

Finally, the front-end, built on Next.js, TailwindCSS, and Headless UI, fetches the optimized document and presents it to the user. The user can then download the enhanced document, compare it with the original, and even provide feedback, all through a highly responsive and accessible interface. This end-to-end integration ensures a seamless and efficient user experience.



3.2.1 Front-end: Technologies, User Interface Components and Mock-ups

The front end of the Accessibilator is built using a blend of Next.js, TailwindCSS, and Headless UI. This technology stack was chosen after careful consideration of several factors, including performance, scalability, accessibility, and ease of development.

Starting with Next.js, our client framework, we chose it primarily for its server-side rendering capabilities. This feature is a boon for SEO and ensures that the initial page loads are swift, enhancing the user experience. Next.js also offers a plethora of built-in features like dynamic routing and API routes, making it a one-stop solution for our front-end needs. While it's a powerful tool, it can be a

bit of an overkill for simpler projects and may pose a learning curve for those new to React. Alternative frameworks like Create React App or Gatsby could be considered, but they do not offer the same comprehensive set of features as Next.js.

TailwindCSS was our choice for styling, and for good reason. Its utility-first approach to CSS allows for rapid development and prototyping. The framework encourages the use of reusable components, adhering to the DRY (Don't Repeat Yourself) principle. However, this utility-first paradigm can be a bit daunting for developers who are new to it. While alternatives like Bootstrap offer more structured, opinionated design systems, they don't provide the same level of customization that TailwindCSS does.

Our UI component, we integrated Headless UI for its unstyled but fully accessible UI components. This library gives us the freedom to implement our custom designs while ensuring that accessibility standards are met. One of the drawbacks is that it requires manual styling, which can be time-consuming. Also, the library offers a limited set of components compared to other options like Material-UI or Ant Design.

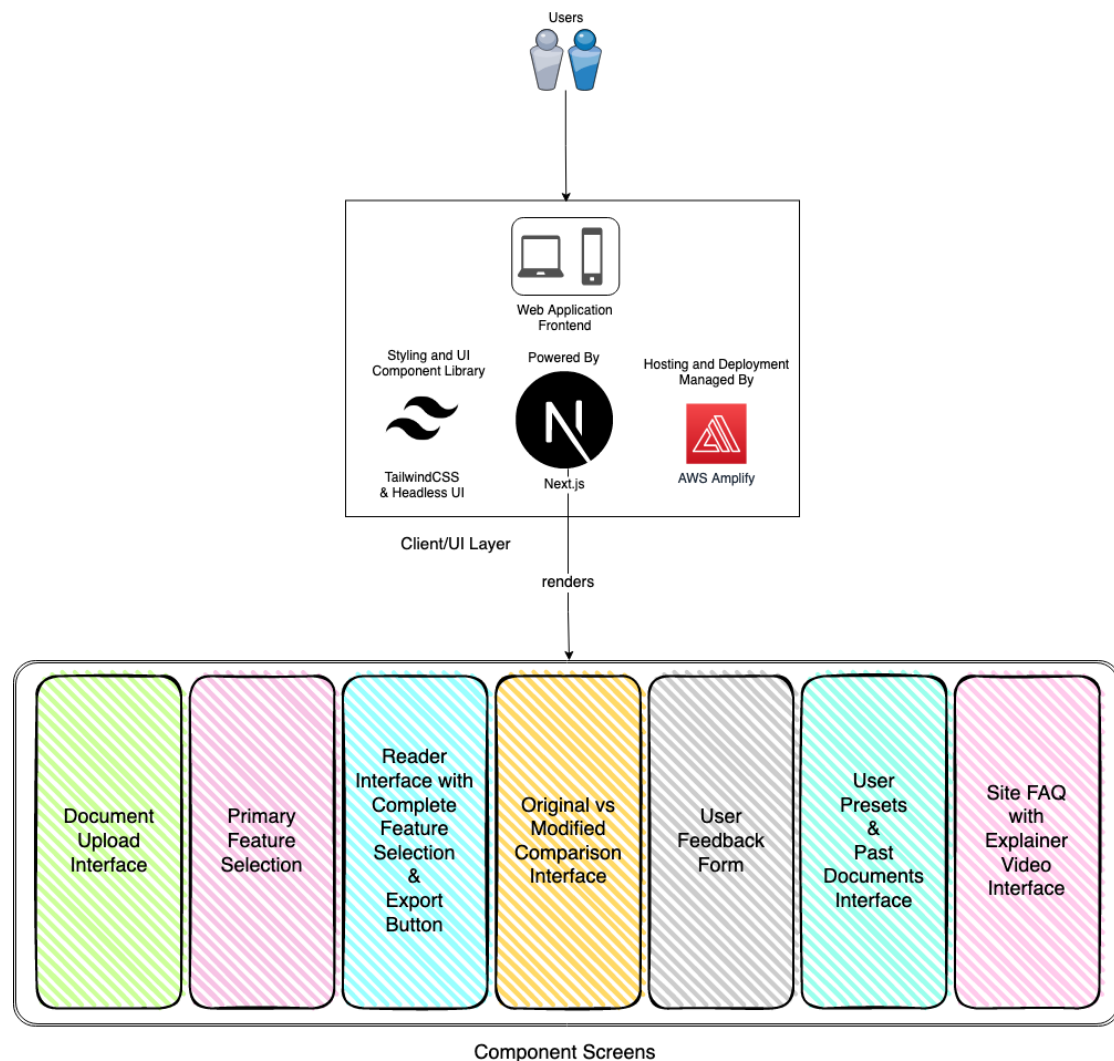


Fig 11: Front Technical Components

User Interface Components:

1. **Document Upload Interface:** This is the landing page where users can upload their documents.
2. **Primary Feature Selection:** Post-upload, users are directed here to select primary modifications such as font size, contrast etc.
3. **Reader Interface with Complete Feature Selection & Export Button:** The reader interface is at the heart of the application frontend, and it allows users to read the document, revert or select additional accessibility features, and export the modified document. It would display advanced features like Dyslexia ruler (with customizations), Screen reader, and

Text-to-Speech functionalities. Here is a brief description of these features:

- a. **Dyslexia Ruler (with Customizations):** The Dyslexia Ruler is a tool designed to assist users, particularly those with dyslexia or other reading difficulties, in focusing on one line of text at a time. It acts as a visual guide that runs horizontally across the screen, usually overlaying the document being read. Here are some key aspects:

Customizations

- i. **Colour:** Users can customize the colour of the ruler to contrast well with the document's background for optimal visibility
- ii. **Opacity:** The ruler can be made opaque, allowing the user to see text while not distracted by the other lines.
- iii. **Thickness:** Adjusting the thickness can help users find a balance between obscuring too much text and providing sufficient guidance.
- iv. **Positioning:** Users should be able to move the ruler up and down to guide their reading without having to use complicated commands.

Benefits

- i. **Focused Reading:** It aids in reducing the distraction caused by adjacent lines of text.
 - ii. **Ease of Tracking:** It helps users maintain their place in a document, preventing them from losing track of which line they are reading.
- b. **Screen Reader:** The Screen Reader is a software application that interprets what is displayed on the screen and presents it to the user through auditory or tactile feedback (for example, Braille). An immersive reader goes further by enhancing the reading experience through additional features.

Features

- i. **Word Highlighting:** As the text is read aloud, words could be highlighted to provide visual reinforcement.
- ii. **Simplified Layout:** Removes distracting elements from the screen to focus on the text.

Benefits

- i. **Accessibility:** Makes text-based content accessible to visually impaired users.
 - ii. **Comprehension:** Enhances understanding by synchronizing auditory and visual elements.
- c. **Text-to-Speech:** Text-to-Speech (TTS) technology converts written text into spoken words. This is particularly useful for those who may have difficulty reading text on a screen.

Features

- i. **Voice Customization:** Adjust the pitch, speed, and voice to fit the user's preferences
- ii. **Pause/Resume:** Users can pause and resume the reading at any point.
- iii. **Skip Forward/Backward:** Navigation features to jump to specific sections of the document.
- iv. **Customization:** Change the voice, pitch, and speed according to user preference.

Benefits

- i. **Multisensory Engagement:** Engages both visual and auditory senses, improving comprehension and retention.
 - ii. **Convenience:** Users can listen to the document while performing other tasks, making it easier to multitask.
4. **Original vs Modified Comparison Interface:** Users can compare the original and modified documents side-by-side.
5. **User Feedback Form:** A modal form to collect user feedback on the features and overall experience.
6. **User Presets & Past Documents Interface:** Users can save their preferred settings and view their past documents.

7. **Site FAQ with Explainer Video Interface:** A comprehensive FAQ section with an embedded explainer video to guide new users.

Mock-ups (Wireframes)

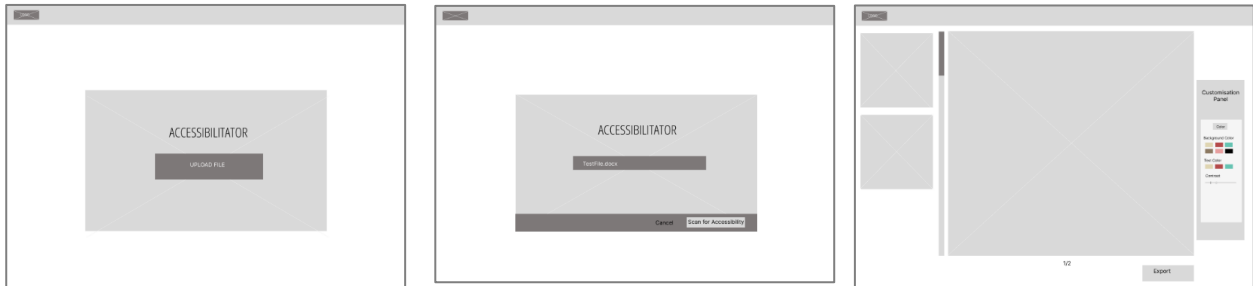


Fig 12: Landing Page and Reader Interface Wireframe

Medium Fidelity Prototypes

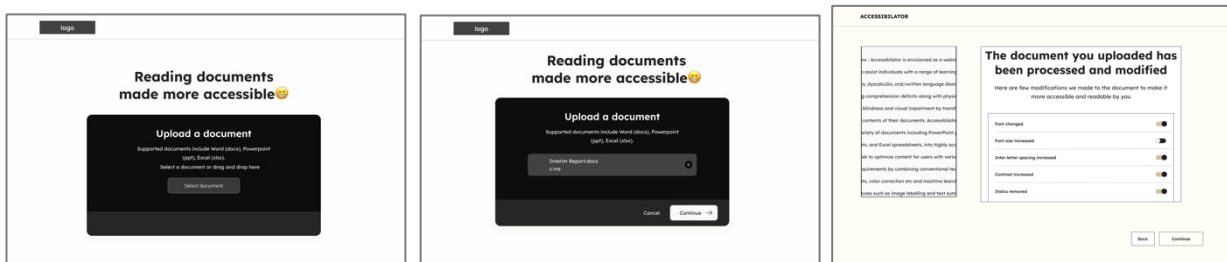


Fig 13: Landing Page and Primary Features Medium Fidelity

3.2.2 Back-end: Technical components

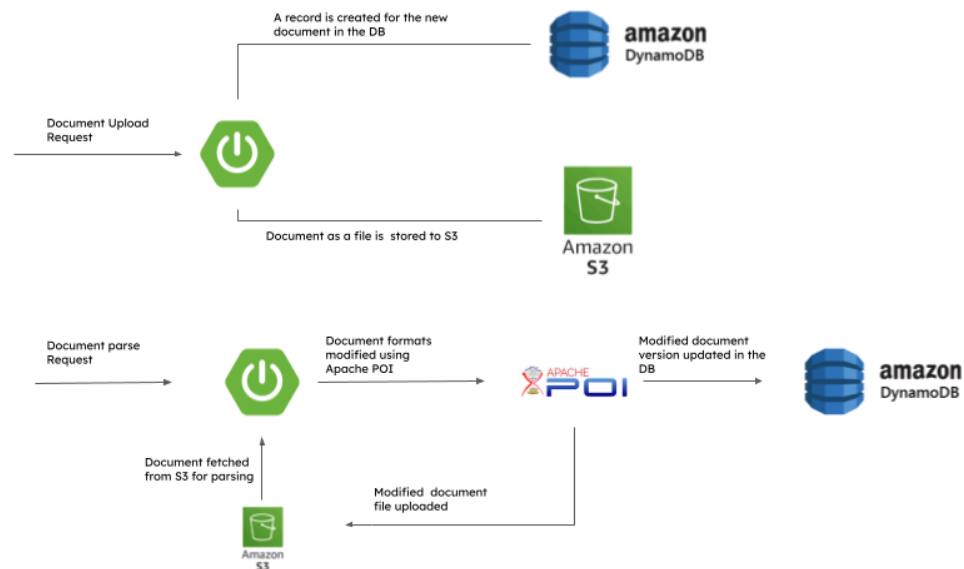


Fig 14: Backend Simplified Workflow

Java:

Our initial inclination was to utilize Node.js for our web-based application's backend, primarily due to its seamless integration with the frontend and its ease of development advantages. However, we encountered limitations in the availability of robust document parsing libraries for various file formats like DOCX, PPT, and Excel. Some available libraries had spotty maintenance, which raised concerns about long-term viability and support.

Upon careful consideration, we decided to transition to Java for the backend. Java offers many reliable and well-supported document parsing libraries, including Apache POI, which we ultimately selected. This library supports a variety of document types, from Word and PowerPoint to Excel files, and also enables efficient reading and editing functionalities. The vibrant community of developers around Apache POI and its comprehensive documentation ensures that we have access to extensive support and resources, aligning well with our needs for a versatile and reliable parsing solution.

Spring Boot Framework:

Spring Boot's convention-over-configuration approach allows developers to set up and run applications with minimal setup. The integration of Apache POI with Spring Boot is seamless, enabling the swift development of RESTful APIs tailored for document parsing. In addition to this Spring Boot has excellent exception handling mechanisms, with which you can handle errors gracefully, such as invalid file formats or corrupted files.

Amazon DynamoDB:

We've selected DynamoDB as our database. For the tasks associated with document parsing, we're convinced that a NoSQL database is the optimal choice due to its absence of a rigid schema. This flexibility allows us to accommodate a variety of metadata for individual documents without restructuring the database. For instance, one document could contain metadata about font style, while another focuses on color details. This approach ensures our database remains adaptable in line with the evolution of our application. Furthermore, since we're leveraging AWS cloud services for various application components, integrating DynamoDB was a logical decision

Amazon S3:

AWS S3 stands out as a prime choice for document storage. Boasting an impressive 99.999999999% durability rate annually, AWS S3 ensures utmost safety for stored documents. In addition to this S3 provides robust security measures, including bucket policies, seamless integration with AWS Identity and Access Management (IAM), and encryption while data is at rest. This comprehensive security approach guarantees the utmost protection for user data throughout document processing.

Document formatting Implementation

- a. **File Upload:** After receiving the file from the UI, it's first uploaded to an S3 bucket. Following this, Metadata including the file ID, filename, and S3 URL are recorded in Amazon DynamoDB.
- b. **Font Adjustment:** The document's font size can be fetched and modified using Apache POI. The default setting for the new font size is 14 points, although this can be customized by the user.
- c. **Font Style Customization:** Font styles such as Lexend, Open Sans, Comic Sans OpenDyslexic, Dyslexie can be applied as per user preference using Apache POI's built-in methods. The default font style will be Open Sans.
- d. **Background Colour Modification:** Apache POI allows you to fetch and change the background color for each paragraph in the document.
- e. **Layout Alteration:** The document layout can be set to left-aligned using Apache POI's built-in methods.
- f. **Contrast Enhancement:** The library allows you to fetch the current text colour for contrast analysis. Utilizing algorithms like the WCAG contrast ratio, you can assess and adjust the text and background colours for better readability.
- g. **Table of Contents Addition:** Headings in the document can be identified by iterating through its paragraphs using the Apache POI. A new paragraph is then created to list these headings in a numbered format, serving as the table of contents.

Datasets

Dataset Name	Source	Functionality	Description
Trending YouTube Video Statistics	Kaggle [Link]	For Headings Generation	This dataset contains statistics and metadata for trending YouTube videos, including their titles and category IDs. Useful for generating relevant headings for

			content or studying trends in video topics.
CNN-DailyMail News Text Summarization	Kaggle [Link]	For Headings Generation	Contains news articles and their summaries. Useful for training text summarization models and generating headlines based on the summaries.
English language- Obscene words	WorkWithData [Link]	Replace Bad Words	Contains a list of obscene words in English. Useful for filtering out or replacing offensive words to make text more appropriate for a wider audience.
CNN/Daily Mail	PapersWithCode [Link]	Text Summarization	Another dataset containing CNN and Daily Mail news articles and summaries. Commonly used in academic research for developing text summarization algorithms.
LScD (Leicester Scientific Dictionary)	Figshare [Link]	Simplify Jargons	Contains scientific jargon and their simpler explanations or definitions. Useful for translating

			complex scientific terms into easier-to-understand language.
English Word Dataset Synonyms and Antonyms	Kaggle [Link]	Simplify Jargons	Contains English words along with their synonyms and antonyms. Useful for simplifying complex words or jargon by replacing them with more common synonyms.
Dataset for classifying English words	Mendeley [Link]	Simplify Jargons	Can be used for various NLP tasks, including word classification. Useful for identifying and replacing jargon or complex terms to make the text more accessible.

Data Science Implementation for the Advanced Features

Heading Generation:

We In response to our comprehensive user survey, which revealed a demand for clear and concise paragraph headings, especially from individuals with dyslexia, we are embarking on a project to design, train, and deploy a Long Short-Term Memory (LSTM) model. Our primary objective is to automatically generate contextually relevant and easy-to-understand headings for individual paragraphs within textual documents. To ensure scalability and ease of integration, the trained LSTM model will be hosted on Amazon SageMaker. This

will facilitate seamless integration into existing applications, thereby providing users with automatically generated headings for easier navigation and comprehension of text.

Replace Bad Words

In order to ensure a clean and respectful user environment, we will employ text sanitization techniques to remove or replace inappropriate language. We will utilize Python-based profanity filtering libraries, such as profanity-check, profanity filter, and better_profanity, to systematically cleanse the text of any offensive or unacceptable terms.

Expand abbreviations

In order to enhance readability and understanding, our system will incorporate a feature to expand abbreviations within the text. Leveraging state-of-the-art word embedding techniques such as Word2Vec, FastText, or GloVe, the system will automatically identify abbreviations and replace them with their corresponding full forms. This will not only improve the text's clarity and help users with dyslexia but also make it more accessible to wider audience, including users who may not be familiar with certain abbreviations.

Simplify Jargons

Individuals with dyslexia find complex terminology and technical jargons challenging to comprehend. Therefore, we are committed to simplifying technical jargon within our content. To achieve this, we will utilize advanced natural language processing tools available in the Hugging Face Transformers library. Specifically, we will implement pre-trained sequence-to-sequence models like T5 and BART. These state-of-the-art models are engineered to transform complex and technical language into simpler, easily digestible terms.

Image labelling

In an effort to revolutionize image labeling, we will be leveraging OpenAI's state-of-the-art CLIP model, which is pre-trained to provide highly accurate labels by

analyzing both textual and visual data. One of the standout features of CLIP is its robustness, eliminating the need for a specialized dataset for training. To ensure ease of deployment and scalability, the CLIP model will be hosted on Amazon SageMaker. By combining advanced text and image recognition capabilities, CLIP will enable us to generate exceptionally accurate and contextually relevant image labels.

Glossary Auto-generation

We are implementing an automatic glossary generation feature within our application for technical terms that cannot be replaced in the text. Utilizing Term Frequency-Inverse Document Frequency (TF-IDF) algorithms, we will systematically identify key terms and phrases that hold significant importance within the document. Once these important words are identified, we will employ the WordNet API to fetch precise and contextually relevant definitions for each term. The result is an auto-generated glossary that not only enhances the readability of the document but also serves as a quick reference guide for readers, aiding in better understanding of the content.

Paragraph Chunking and sentence splitting

Given the intricate nature of paragraph chunking and sentence splitting functionalities, we've opted to leverage OpenAI's text completion and fine-tuning APIs to streamline the development process. While traditional methods could require considerable time and resources, using OpenAI's advanced natural language processing capabilities will allow us to efficiently break down lengthy paragraphs into more manageable chunks and split complex sentences into simpler structures. Although this approach may be computationally intensive, it offers a more effective and time-efficient solution for enhancing the readability and navigability of textual content.

Text Summarization

In a bid to make extensive textual content more digestible and user-friendly, we will implement an automated text summarization feature. We will employ

advanced deep learning architectures, including Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs), to generate concise yet informative summaries. These sophisticated models excel at capturing the semantic essence and key points of lengthy documents, transforming them into shorter versions that retain the original meaning. By integrating this feature, we aim to improve user engagement and comprehension, allowing for quick consumption of important information without the need to sift through voluminous text.

4. Evaluation: The Reviews

4.1. What does success look like for our system?

Success for our system is multi-faceted, encompassing both user-centric and technical dimensions. At the core, our primary aim is to fulfil the specific needs of our user personas as discussed earlier, which include individuals with dyslexia and those who support them.

From a user success standpoint, the system aims to educate users about dyslexia-friendly formatting rules, thereby empowering them to make informed decisions when interacting with digital content. For people living with dyslexia, the ultimate measure of success is making documents easier to read and comprehend. We aim to achieve this by offering automated tools that can adapt text, colour schemes, and layouts to meet individual needs. Furthermore, our system is designed to cater to a wide range of disabilities within the dyslexia spectrum, ensuring that our impact is as inclusive as possible.

On the technical front, success is gauged by the system's ability to efficiently and accurately parse and optimize a variety of document types. This involves seamless integration between the front-end and back-end components, from the moment a user uploads a document to when they download the optimized version. The backend, running on a Spring app on an EC2 instance, will work in tandem with AWS services like S3 for document storage and SageMaker for

machine learning tasks. The possible involvement of AWS Step Functions will ensure a well-coordinated, state-managed flow of these services.

Moreover, we aim for at least a 20% improvement in document accessibility scores, as measured by industry-standard tools. This quantifiable metric serves as a robust indicator of both technical and user success, aligning with globally accepted best practices in digital accessibility.

In summary, success for our system is not just about technical proficiency but also about making a meaningful impact on our users' lives. Achieving these objectives will affirm the system's effectiveness and its potential to become a valuable resource in the field of digital accessibility.

4.2. How will we evaluate the system that we built?

Evaluating the system we've built will involve a multi-modal approach that combines quantitative metrics with qualitative feedback to ensure both technical and user-centric success. Here's how we plan to go about it:

4.2.1. User-Centric Evaluation:

- a. **User Surveys:** Post-interaction surveys will be used to gather user opinions on the system's ease of use, effectiveness, and overall experience.
- b. **Case Studies:** We'll conduct in-depth interviews with a subset of users, especially those with dyslexia, to understand how the system has impacted their reading experience.
- c. **Usability Testing:** We'll employ a multi-method approach for usability testing. Think-aloud testing will provide insights into user navigation and perception, while Task-Based Testing will focus on the ease with which

users can complete specific objectives. These methods will be complemented by user surveys aimed at gathering information on user preferences, pain points, and overall experience.

- d. **Feedback Loop:** We have an in-app feedback form feature for users to quickly report issues or provide suggestions that can offer real-time qualitative data.
- e. **Inclusive Evaluation:** We aim to cater to a wide range of disabilities within dyslexia. Therefore, our evaluation will also focus on how well the system adapts to different user needs, including those who may not be well-versed in dyslexia-friendly formatting rules.

4.2.2. Technical Evaluation:

- a. **Front-end Evaluation:** Our frontend, built using Next.js, TailwindCSS, and Headless UI, will be rigorously evaluated for performance. We'll employ Google PageSpeed Insights to measure key metrics such as Time to First Byte (TTFB), First Contentful Paint (FCP), and overall Page Load Time. These metrics will help us understand the speed and responsiveness of our user interface, which is crucial for user engagement and retention. A slow-loading interface can lead to higher bounce rates, which would be detrimental to our goal of making documents more accessible.
- b. **Backend Evaluation:** Our central backend, running on a Spring Boot application hosted on an AWS EC2 instance, will be monitored using Amazon CloudWatch. This will allow us to track essential performance indicators like CPU utilization, disk I/O, and network throughput at five-minute intervals. We'll set up CloudWatch Alarms to notify us if any of these metrics fall outside predetermined thresholds, ensuring the system's reliability and robustness.
- c. **Data Handling Considerations:** We're using Amazon DynamoDB for our database and AWS S3 for document storage. Their performance and

security will also be consistently reviewed in line with GDPR regulations, especially considering the sensitive nature of user-uploaded documents. AWS provides robust IAM roles and policies to facilitate this.

- d. Code Quality and Functionality:** To ensure the highest code quality, we'll write unit testing across major sections of the codebase. This will be followed by functional testing to validate individual features and integration testing to ensure that all functionalities work cohesively when integrated into the larger system.

By employing this comprehensive evaluation strategy, we aim to build a system that is both technically robust and has a real-world impact, fulfilling our user personas' needs and making documents more accessible for persons with living dyslexia.

5. Conclusion: The Plan

5.1. What is our project management strategy?

Our project management strategy is rooted in Agile methodologies, specifically Scrum, to ensure that we can adapt to changes quickly and efficiently. We have divided the entire project into sprints, each lasting one week. At the beginning of each sprint, we conduct a planning meeting to prioritize tasks and set achievable goals. Daily stand-up meetings help us track progress and address any roadblocks. We also hold sprint retrospectives at the end of the sprint to assess what was accomplished, team morale, shout-outs and what needs improvement. We are using the robust Azure DevOps Boards for task tracking and GitHub for our mono-repo and version control management, which facilitates seamless collaboration among team members.

We also link work items like epics, user stories and tasks to GitHub commits and PRs to help us better track and investigate the progress and status of tasks

Additionally, we have set up CI/CD pipelines on AWS Amplify to automate our testing and deployment, ensuring that we can deliver features and fixes more rapidly and reliably.

5.1.1. Team Communication

Most team-related communication happens on a Slack workspace which provides a centralized platform for real-time meetings, sharing of documents, and asynchronous communication. The ability to create channels for different teams or project modules ensures that discussions are topic-specific and organized. Its rich set of features, including direct messaging, pinned items, and integrations with other platforms, allows for a cohesive communication strategy.

While digital tools provide convenience, face-to-face interactions are invaluable when it comes to understanding nuances, brainstorming, and building team camaraderie. Also, in situations that require urgent attention or a quick solution, we go the old-fashioned route by picking up the phone and giving ourselves a call.

5.2. What are the biggest challenges we are currently facing?

Any effective team must navigate a series of challenges to achieve its goals. These challenges often come in two flavours: technical and team-oriented. Addressing both is crucial for the successful completion of our project.

5.2.1. Technical Challenges:

- a. **Document Parsing:** Parsing various document formats like .docx, .pdf, .ppt and .xlsx is a significant hurdle. The challenge amplifies when we aim to maintain the original document's quality and structure while enhancing its accessibility. Moreover, each format might require a different set of libraries and languages to parse them optimally. For this

reason, we decided to start our system with the most universal document format, .docx and support other formats as we advance or in future work.

- b. **NLP/Machine Learning Implementation:** The use of natural language processing and machine learning for tasks like sentence chunking and image labelling, and glossary auto-generation presents challenges in both development and computational resources.
- c. **User Experience:** Creating an interface that is accessible to individuals with learning disabilities is not a cakewalk. It involves rigorous UX analysis and strict adherence to WCAG guidelines and accessible development standard. The challenge lies not just in the end product but also in the journey to get there.
- d. **Data Security:** Although AWS S3 is quite secure on its own, as users will be uploading potentially sensitive documents, implementing robust security measures is paramount. We plan to implement AWS K3 encryption at rest and inform test user groups not to upload sensitive data in the meantime.
- e. **Scalability:** Deciding between server-based and serverless architectures is a critical decision that impacts how well our system can scale to meet user demand. Both have their tradeoffs in terms of ease of configuration for server, and cold-starts and timeouts for serverless.
- f. **Complexity in Evaluating Success:** Given the multi-faceted nature of our project, evaluating success is not straightforward. We plan to use a combination of automated, manual and expert opinion methodologies to assess our progress comprehensively.

5.2.2. Team-Oriented Challenges:

- a. **Limited Data Science Expertise:** Having only one data science team member is causing some strain on that aspect of the project, especially when data-driven solutions are in play. The rest of the team is working hard to upskill in that area as well

- b. **Learning Curve for New Technologies:** Although we tried our best to select tools that are familiar to the team in reality having the entire team be familiar with every technology, we use is quite impractical.
- c. **Adherence to Processes and Ownership:** Ensuring everyone follows the defined process and takes responsibility is an ongoing challenge.

5.3. How will we use the time remaining to achieve a successful outcome?



Fig 15: Gantt Chart

With the time remaining, our focus will be on executing a well-defined plan that addresses both our technical and team-oriented challenges. Here's how we intend to use the remaining time to achieve a successful outcome:

Prioritization and Milestones: We will start by revisiting our project timeline and setting clear milestones. This will help us prioritize tasks that are crucial for the project's success. For instance, resolving the document parsing issues will be a top priority, given its foundational role in our application.

Technical Refinement: We will allocate specific time blocks for refining our machine-learning algorithms and enhancing the user interface. This includes running multiple iterations and tests to ensure that our algorithms are accurate and that the UI is intuitive and accessible.

Team Upskilling: Given the limited data science expertise within the team, we will organize short-term training sessions. These sessions will aim to bring everyone up to speed on the essential data science skills required for the project.

User Testing and Feedback: We will also allocate time for comprehensive user testing. This will involve both automated and manual tests, including think-aloud and task-based usability tests. Feedback from these tests will be used for iterative improvements.

Final Review and Deployment: The last phase will involve a rigorous review of all project components, followed by deployment. Post-launch, we will continue to monitor performance and user engagement metrics to make any necessary adjustments.

By adhering to this plan and making effective use of the time remaining, we strive to deliver a product that not only meets but exceeds our initial objectives.

References

- Almgren Bäck, Gunilla, Lindeblad, E., Elmqvist, C., & Svensson, I. (). Dyslexic students' experiences in using assistive technology to support written language skills: a fiveyear followup. *Disability and Rehabilitation: Assistive Technology*, 1–11.
<https://doi.org/10.1080/17483107.2022.2161647>
- British Dyslexia Association. (2018). *Dyslexia friendly style guide*. British Dyslexia Association. <https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide>
- Britton, D. (2019, July 22). *Dyslexia*. Daniel Britton. <https://danielbritton.info/dyslexia/>
- Butterworth, B., & Kovas, Y. (2013). Understanding Neurocognitive Developmental Disorders Can Improve Education for All. *Science*, 340(6130), 300–305.
<https://doi.org/10.1126/science.1231022>
- Coles, G. (1999). *Reading lessons: The debate over literacy*. Macmillan.
- de Santana, V. F., de Oliveira, R., Almeida, L. D. A., & Baranauskas, M. C. C. (2012). Web accessibility and people with dyslexia. *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility - W4A '12*.
<https://doi.org/10.1145/2207016.2207047>
- Galliussi, J., Perondi, L., Chia, G., Gerbino, W., & Bernardis, P. (2020). Inter-letter spacing, inter-word spacing, and font with dyslexia-friendly features: testing text readability in people with and without dyslexia. *Annals of Dyslexia*, 70(1), 141–152.
<https://doi.org/10.1007/s11881-020-00194-x>

- Grusky, M., Taft, J., Naaman, M., & Azenkot, S. (2020). Measuring and Understanding Online Reading Behaviors of People with Dyslexia. *In Submission*.
<https://par.nsf.gov/biblio/10170285>
- International Dyslexia Association. (2002). *Definition of Dyslexia*. International Dyslexia Association. <https://dyslexiaida.org/definition-of-dyslexia/>
- International Dyslexia Association. (2012). *JUST THE FACTS... Information provided by The International DYSLEXIA Association ® DYSLEXIA BASICS*.
<http://dyslexiahelp.umich.edu/sites/default/files/DyslexiaBasicsREVMay2012.pdf>
- Kennecke, A.-K., Wessel, D., & Heine, M. (2022). Dyslexia and Accessibility Guidelines – How to Avoid Barriers to Access in Public Services. *I-Com*, 21(1), 139–155.
<https://doi.org/10.1515/icom-2021-0040>
- Kumar Singh, N. (2021, June 6). *Essential elements of Agile Scrum*. Google.com.
<https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2Fagilemania%2Fessential-elements-of-agile-scrum-d5d5d8cafe8d&psig=AOvVaw2GVE1noaGkOvtJoOhkdbRb&ust=1695977762044000&source=images&cd=vfe&opi=89978449&ved=0CBMQjhqxqFwoTCKj5ttL3zIEDFQAAAAAdAAAAABAT>
- McCarthy, J. E., & Swierenga, S. J. (2009). What we know about dyslexia and Web accessibility: a research review. *Universal Access in the Information Society*, 9(2), 147–152. <https://doi.org/10.1007/s10209-009-0160-5>
- Musvuugwa, T., Dlomu, M. G., & Adebawale, A. (2021). Big Data in Biodiversity Science: A Framework for Engagement. *Technologies*, 9(3), 60.
<https://doi.org/10.3390/technologies9030060>
- Nevill, T., & Forsey, M. (2022). The social impact of schooling on students with dyslexia: A systematic review of the qualitative research on the primary and secondary education

of dyslexic students. *Educational Research Review*, 38, 100507.

<https://doi.org/10.1016/j.edurev.2022.100507>

Putra, J. W. G., Teufel, S., & Tokunaga, T. (2023). Improving logical flow in english-as-a-foreign-language learner essays by reordering sentences. *Artificial Intelligence*, 320, 103935. <https://doi.org/10.1016/j.artint.2023.103935>

Rello, L., & Baeza-Yates, R. (2015). How to present more readable text for people with dyslexia. *Universal Access in the Information Society*, 16(1), 29–49.
<https://doi.org/10.1007/s10209-015-0438-8>

Rello, L., & Baeza-Yates, R. (2016). The Effect of Font Type on Screen Readability by People with Dyslexia. *ACM Transactions on Accessible Computing*, 8(4), 1–33.
<https://doi.org/10.1145/2897736>

Rello, L., & Bigham, J. P. (2017a). Good Background Colors for Readers. *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17*. <https://doi.org/10.1145/3132525.3132546>

Rello, L., & Bigham, J. P. (2017b). Good Background Colors for Readers. *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17*. <https://doi.org/10.1145/3132525.3132546>

Rello, L., Kanvinde, G., & Baeza-Yates, R. (2012). Layout guidelines for web text and a web service to improve accessibility for dyslexics. *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility - W4A '12*.
<https://doi.org/10.1145/2207016.2207048>

Shaywitz, S. E. (1996). Dyslexia. *Scientific American*, 275(5), 98–104.

Shaywitz, S. E. (2003). *Overcoming dyslexia: A new and complete sciencebased program for reading problems at any level*. Knopf.

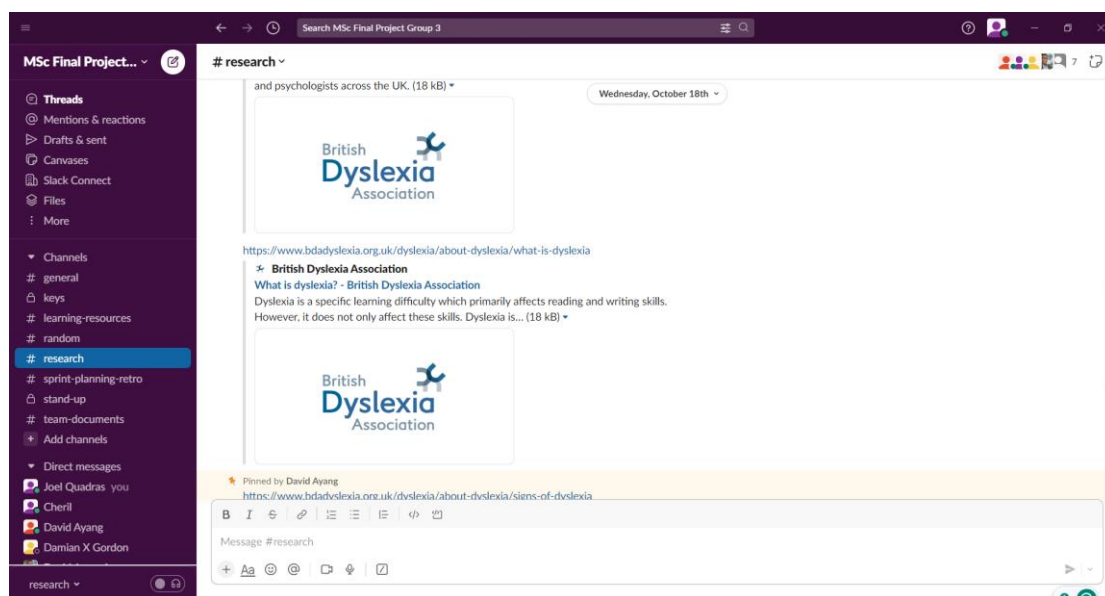
Siegel, L. S. (2006). Perspectives on dyslexia. *Paediatrics & Child Health*, 11(9), 581–587.

Snowling, M. J., & Hulme, C. (2011). Evidence-based interventions for reading and language difficulties: Creating a virtuous circle. *British Journal of Educational Psychology*, 81(1), 1–23.

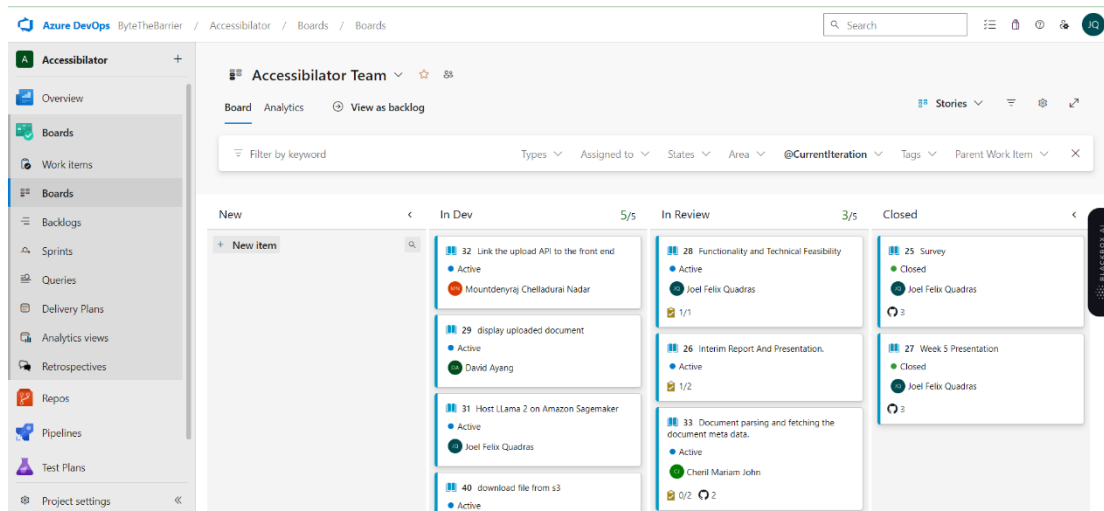
Williams, C., de Greef, L., Harris, E., Findlater, L., Pavel, A., & Bennett, C. (2022). Toward supporting quality alt text in computing publications. *Proceedings of the 19th International Web for All Conference*, 1–12.

<https://doi.org/10.1145/3493612.3520449>

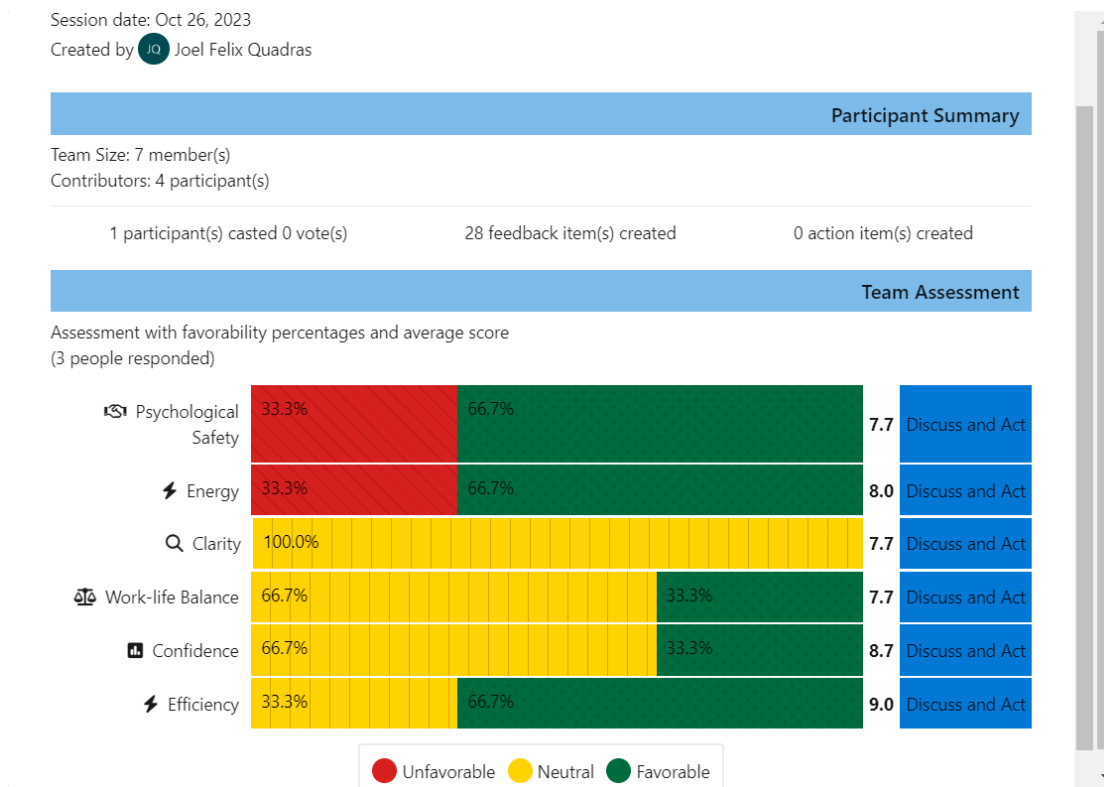
Appendix



Appendix A: Slack Workspace showing channels



Appendix B: Azure Boards Showing Active Sprint Work Items With Linked PRs



Appendix C: Azure Boards Retrospective Team Psychology

What didn't go well:

- Time management has to be given more importance. [0 votes]
- The team still has to increase its pace and more contributions are expected from the team members. [0 votes]
- We really kicked in our gears this week but the only thing is that we couldn't reach all of our goals which we set out for this week. [0 votes]
- The team needs to pay attention to deadlines and submissions from the start. [0 votes]
- Team members are not doing enough commits and PR requests to Github. [0 votes]

What went well:

- Co-ordination and collaboration was really good this week, with everyone chipping in their bits and pieces. I feel the communication between us has improved loads than when we began. [0 votes]
- The Team really started shifting gears and the team members are more available to each other in terms of collaboration and communication. [0 votes]
- the team members are helping out each other by sharing their knowledge and information and keeping all team members on the same page. [0 votes]
- The team energy and motivation at its highest so far [0 votes]
- The team collaborated extensively . Developed more rapport between the team members. Was able to learn a lot from each other [0 votes]

What needs to change:

- Team should start giving importance to following the process [0 votes]
- better time management by the team. [0 votes]
- More github commits and PR requests. [0 votes]
- The team needs to improve in ensuring that deadlines are met [0 votes]
- More commits to Github and equal assignment of tasks on time are a main issue with the team [0 votes]
- Every member should feel a sense of responsibility to the success of the project [0 votes]
- team members should focus on following the process along with the application development. [0 votes]

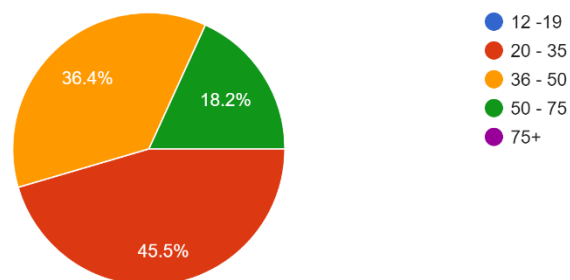
☀ Shout Outs:

- Appreciating the Individual efforts of the team members too to keep the application development pace in check. Cheers! [0 votes]
- To The entire team in their efforts in changing the pace of the project [0 votes]
- Shout Out to the team members for really picking up pace and getting serious in terms of what we will be achieving at the end of this project. [0 votes]
- David for the insights shared on collaborative session [0 votes]
- Amazing teamwork this week. Shout out to everyone! [0 votes]
- Collaborative efforts made it possible for us to finalize the UI design and get it vetted. Kudos to the team for the same. [0 votes]
- Shoutout to the front end team members for developing the Figma and the UI components. [0 votes]

Appendix D: Azure Boards Retro

Age Group

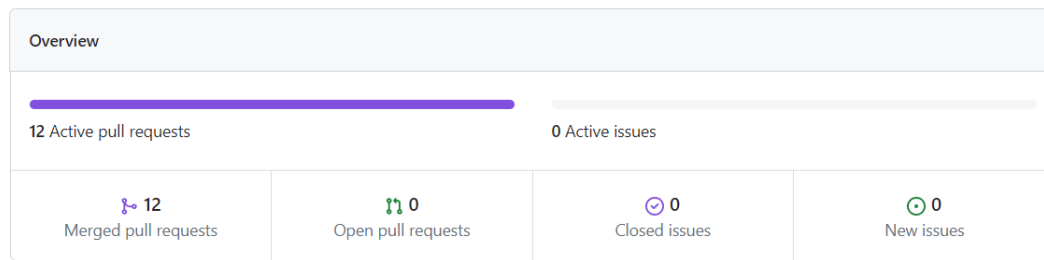
11 responses



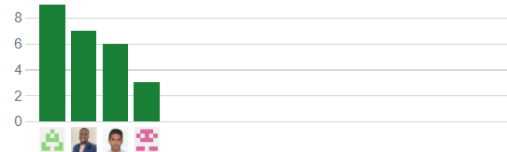
spective Summary

October 19, 2023 – October 26, 2023

Period: 1 week ▾



Excluding merges, **4 authors** have pushed **25 commits** to dev and **25 commits** to all branches. On dev, **29 files** have changed and there have been **1,794 additions** and **1,190 deletions**.



12 Pull requests merged by 4 people

- Implement Global Font Style, Configure TW Primary Color Palette, Touch Up Upload Screen (AB#37)
#39 merged 9 hours ago
- Survey [Fixes AB#25]
#36 merged 12 hours ago
- Week 5 Presentation [Fixes AB#27]

Appendix E: GitHub Weekly Insights

Dyslexia-Friendly Documents Survey

Help us make documents more accessible for people with dyslexia! Your insights will be invaluable in shaping a system that caters to the specific needs and preferences of readers with dyslexia. Whether you have dyslexia or know people with dyslexia, your feedback can pave the way for a more inclusive reading environment. Please take a few minutes to share your thoughts and experiences.

Age Group *

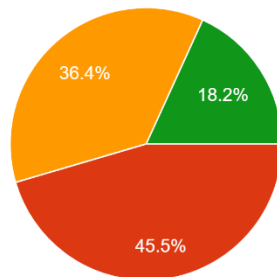
- ☐ 12 - 19
- ☐ 20 - 35
- ☐ 36 - 50
- ☐ 50 - 75
- ☐ 75+

Appendix F: Dyslexia-Friendly Accessible Document Questions

(<https://forms.gle/RQ4vvVYDcomNqP938>)

Age Group

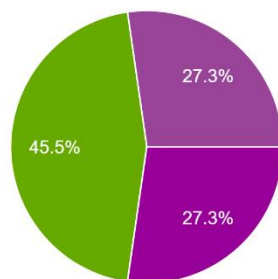
11 responses



- 12 -19
- 20 - 35
- 36 - 50
- 50 - 75
- 75+

Highest Level of Education:

11 responses

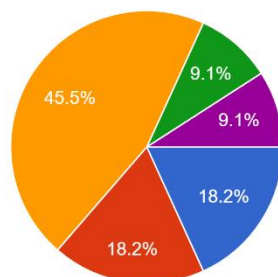


- Some high school
- High school graduate or equivalent
- Some college, no degree
- Associate's degree
- Bachelor's degree
- Post-Graduate-Diploma
- Post-Graduate Certificate
- Master's Degree

▲ 1/2 ▼

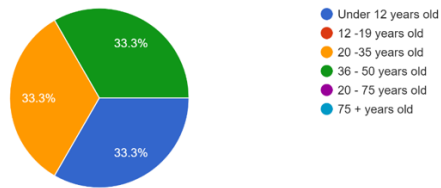
Are you a person with Dyslexia?

11 responses

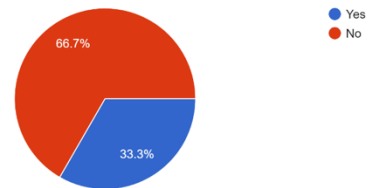


- Yes
- No, but I care for somebody with dyslexia
- No, but I know somebody with dyslexia
- I don't know if I have dyslexia or not.
- No

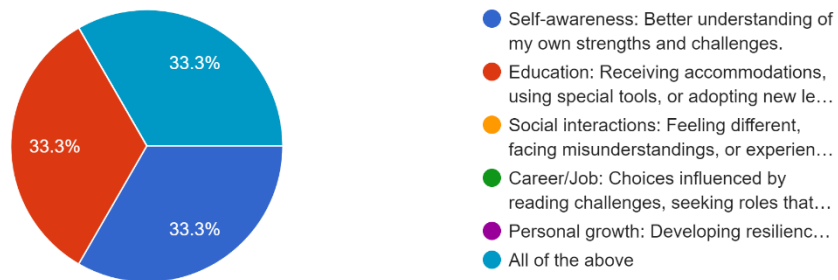
When were you diagnosed with dyslexia?
3 responses



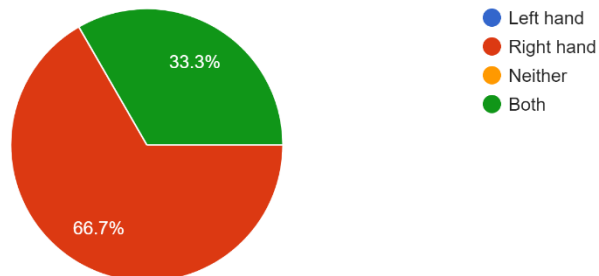
Have you received any formal training or therapy for dyslexia?
3 responses



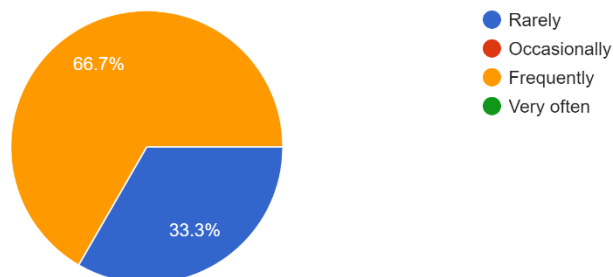
Which of the following areas of your life were influenced by your dyslexia diagnosis?
3 responses



Are you left handed or right handed?
3 responses

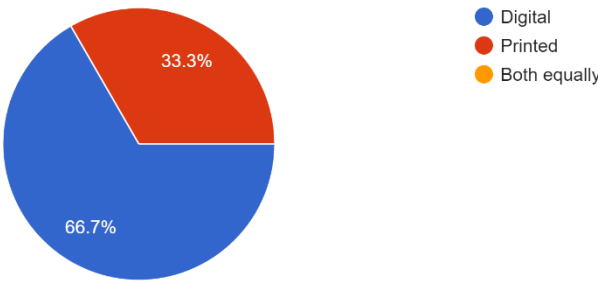


How often do you read documents (e.g., books, articles, reports) on a daily basis?
3 responses



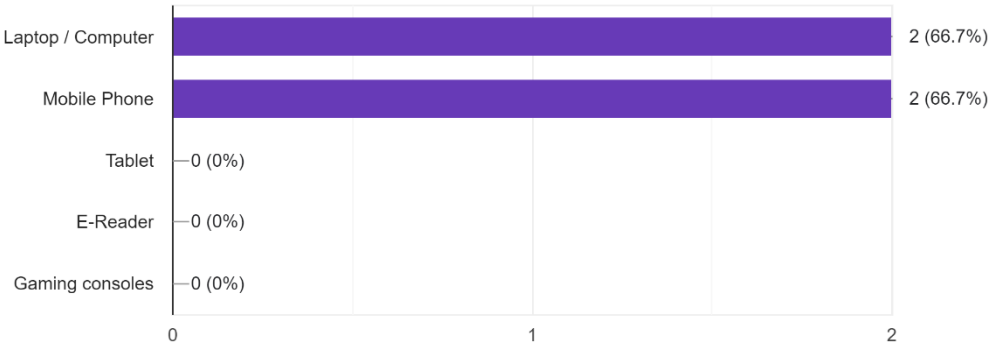
Do you prefer reading digital documents or printed ones?

3 responses



Which digital device(s) do you most frequently use for reading or accessing content?

3 responses



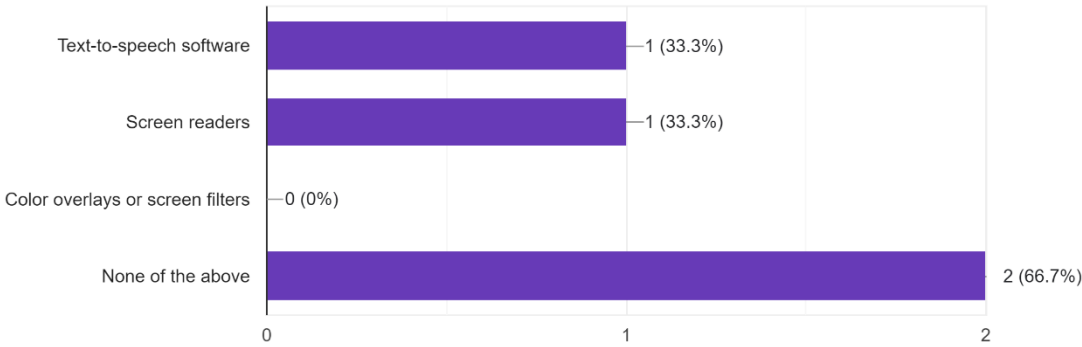
How would you rate your level of expertise with technology?

3 responses



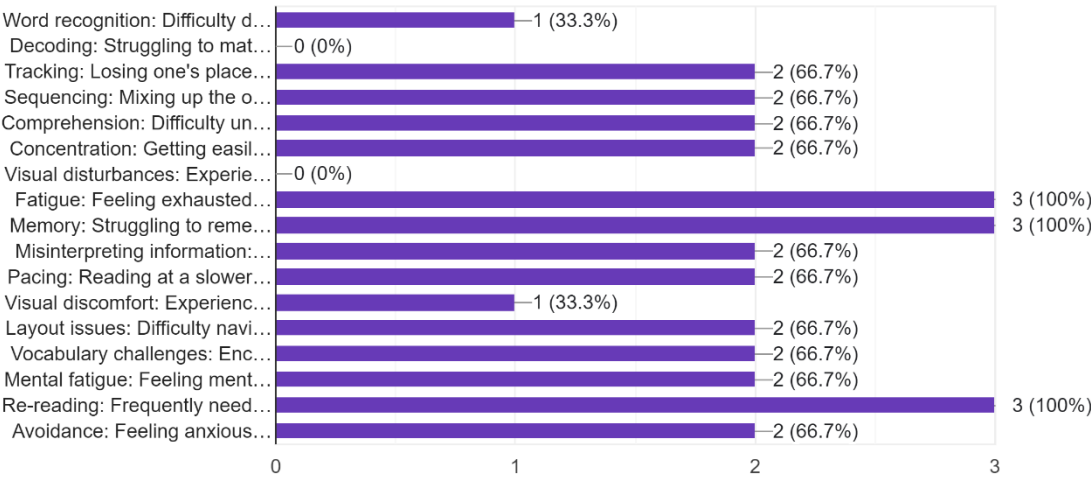
Do you use any of the following assistive technologies or tools to help you read?

3 responses



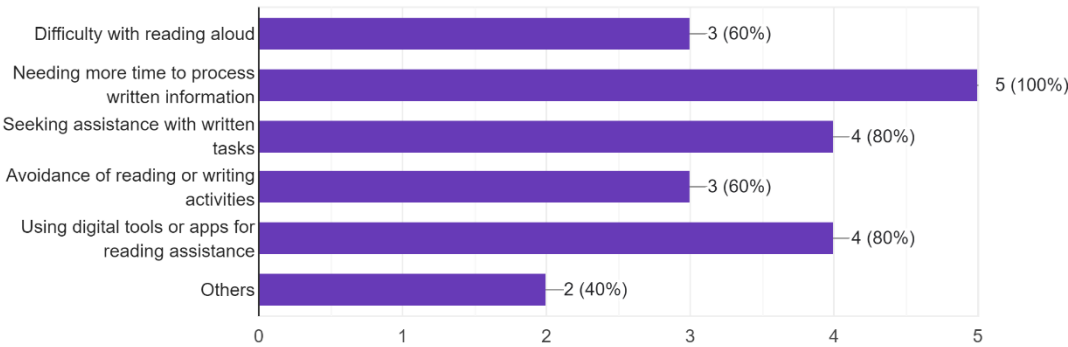
Which challenges do you commonly face while reading documents?

3 responses



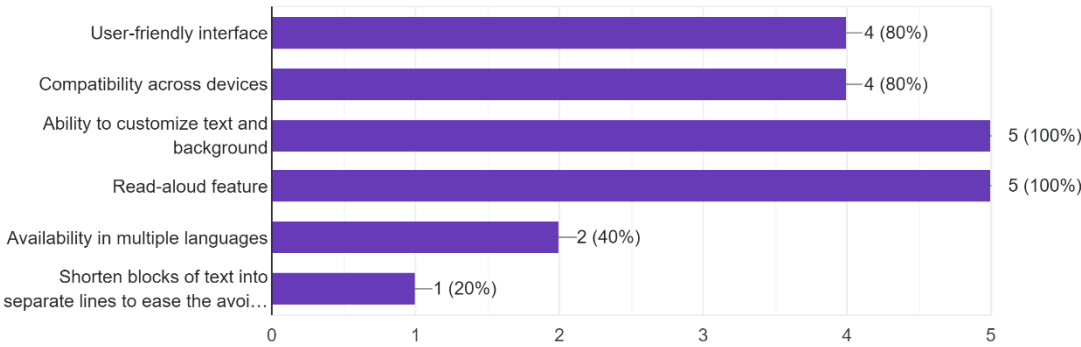
Based on your interactions, what challenges have you noticed the person with dyslexia facing?

5 responses



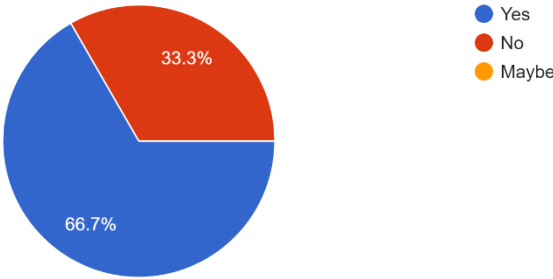
If you were to recommend a system designed to help individuals with dyslexia to someone you know, what features or qualities would you consider important?

5 responses



Would you be interested in a built-in feedback mechanism within the system to continuously share your insights and suggestions?

3 responses



Appendix G: Dyslexia-Friendly Accessible Document Survey Responses

