AI-Based Text Simplification for Cognitive Accessibility

Group 7
24210028 Dekka Muni Kumar
24250041 Kaloori Shiva Prasad
24250106 Sai Naga Venu Gopal Bhamidi
24510056 Kanupriya Mahaur
Indian Institute of Technology Gandhinagar
{muni.kumar, kaloori.shiva, venu.gopal, kanupriya.mahaur}@iitgn.ac.in

I. Introduction

Reading and writing sit at the core of academic learning, yet for many students these everyday activities involve disproportionately high effort. Learners with cognitive and learning disabilities—such as dyslexia, specific reading disorders, language-processing impairments, and Attention-Deficit/Hyperactivity Disorder (ADHD), experience persistent challenges while interacting with written text [1], [2]. Dyslexia affects the mechanisms that convert written symbols into meaningful sounds and units, causing difficulty with phonological decoding, visual discrimination, orthographic mapping, and reading fluency [2], [3]. ADHD influences the ability to sustain attention, manage working memory, and regulate executive control, which in turn disrupts the flow of reading, increases distractibility, and makes it harder to keep track of information in dense passages [4], [5]. Language-based learning disabilities further complicate the task by affecting grammatical interpretation and the extraction of meaning from complex sentences [6]. These students are often fully capable of understanding academic concepts, yet the act of reading itself becomes slow, tiring, and cognitively expensive [7].

Over the past decade, research in cognitive accessibility and human–computer interaction has converged on several mechanisms that support such learners [8], [9]. Studies in reading science show that reducing visual crowding, presenting text with clearer spacing, lowering sentence complexity, and providing auditory alternatives can significantly ease the cognitive load associated with decoding and comprehension [10], [11]. Adjustments to typography, such as increasing letter and word spacing, enlarging font size, or enhancing line height, help readers with dyslexia process visual information more accurately [8], [12]. Text simplification and summarisation reduce syntactic complexity and working-memory demands by restructuring long sentences, substituting difficult vocabulary, and clarifying the relationships between ideas [9], [13], [14]. Multimodal supports, particularly text-to-speech systems with adjustable pacing, offer alternate pathways to understanding when visual decoding is effortful or when attention fluctuates [15], [16]. Across multiple studies, these interventions have been shown to improve comprehension, increase reading speed, reduce fatigue, and elevate overall engagement among neurodivergent learners [8], [17].

While accessible-reading technologies have advanced considerably in English and a few high-resource languages, far less progress has been made for Indian vernacular contexts [18], [19]. Students in regional-medium schools often learn through scripts such as Hindi (Devanagari), Telugu, Gujarati, Bangla, and Marathi, each of which introduces a distinct set of visual and structural complexities [20], [21]. Indic scripts are built on akshara-based systems and frequently involve conjunct characters, non-linear matra placements, visually dense ligatures, and syllables whose written form does not map straightforwardly onto spoken components [21], [22]. These features create intensive visual demands that can be especially challenging for learners with decoding difficulties or attention limitations. In contrast to Latin-script languages, there are few validated dyslexia-friendly typographic guidelines for Indian scripts, and little evidence-based design work that addresses their unique visual characteristics [18].

The technological landscape mirrors this gap. Indian languages lack the large, curriculum-aligned simplified corpora that would enable high-quality text simplification; readability metrics adapted to local scripts are limited; and accessible multimodal tools—such as high-quality text-to-speech tuned for Indian languages remain uneven across regions [23], [24]. As a result, many Indian students with reading and attention difficulties encounter academic content that is visually dense, syntactically complex, and delivered in formats that do not accommodate cognitive diversity. This mismatch contributes to slower comprehension, weaker retention, and disengagement from reading tasks, even when linguistic understanding or subject knowledge is strong.

Addressing these challenges requires systems that integrate script-aware visual design, guided text transformation, and multimodal support in ways that suit the realities of Indian classrooms. Such systems must respect the structure of Indic scripts, reduce unnecessary cognitive effort, and make academic text more accessible without oversimplifying its educational value.

Motivated by these needs, our project explores how AI-enabled text simplification, accessibility-oriented presentation choices, and multimodal support can be combined to enhance reading experiences for students engaging with academic content in Indian vernacular languages.

II. Low-Fidelity Prototype

The low-fidelity prototype was created using Figma and served as the first concrete expression of our system's design vision. Its purpose was to explore user flow, verify accessibility assumptions, and rapidly test interface concepts before implementing any backend intelligence. The low-fi prototype covered two variants of the product: (1) a text-to-speech–focused prototype with summarisation, and (2) a broader reading-aid prototype offering summarisation, personalisation settings, readability controls, and multimodal support. Both prototypes were designed primarily for Indian vernacular languages, with Hindi used as the base horizontal slice for early testing.

The screens in the low-fi prototype included the home page, onboarding flow, language selection, login or guest access, text input/upload page, summarisation screen, and a detailed personalisation questionnaire. The readability configuration section showcased controls for font size, letter spacing, word spacing, line height, bold emphasis for ADHD users, and visual highlight options for dyslexic readers. Each control was represented with live preview text using pangrams to let participants see the impact of spacing, contrast, and text styling. Similarly, text-to-speech features such as play, pause, speed control, and pause insertion were laid out visually to simulate how users would interact with audio feedback.

To evaluate the prototype, a small Wizard-of-Oz user study was conducted with two healthy tenth-grade students and one participant with ADHD. These sessions were carried out through online meetings, where participants navigated the clickable wireframes while thinking aloud. A hidden facilitator acted as the "wizard," manually performing summarization, inserting simplified text, applying spacing adjustments, or simulating translations based on the user's requests. This setup enabled us to test the interaction logic and feature placement despite not having a running NLP model at that stage.

The user study generated several actionable insights. Participants preferred bullet-point summaries for quick scanning, and the ADHD participant reported higher cognitive load when both original and simplified text were shown together. This led us to show only the simplified text on the final screen in later iterations. The preference for larger spacing and adjustable text appearance informed the need for a personalization questionnaire, which was integrated directly into the low-fi design. Users also asked for clearer audio controls and easier navigation between language choices, prompting us to redesign these interface elements and ensure that core actions remained visually prominent.

The low-fidelity prototype helped solidify the structure of the final system, identify usability bottlenecks early, and confirm the necessity of multimodal and personalization features. Insights from this stage directly shaped the high-fidelity implementation, which carried forward the same flow, control groupings, and user-centered accessibility principles, but with fully functional backend summarization, translation, and TTS capabilities.

III. Design Goals and Requirements

The design goals and user requirements for our system were derived directly from observations made during the low-fidelity prototype evaluation and the Wizard-of-Oz study. These early tests allowed us to understand how learners interacted with text, which interface elements reduced cognitive effort, and what features were essential for supporting reading in English and Telugu. Based on these findings, we established a set of design goals that guided the development of the high-fidelity prototype.

A. Primary Goals

The low-fidelity study revealed four major goals that the final system needed to achieve:

- **Immediate readability:** The system should generate a simplified and easy-to-read version of any input text while preserving its semantic meaning.
- **Personalised presentation:** Learners should be able to quickly adjust font size, spacing, and other visual parameters to suit their individual reading preferences.
- **Multimodal access:** Users should have the option to consume content visually or through audio, with controls for playback speed and pacing.
- **Low cognitive load:** The interface should remain simple, with minimal steps, clear navigation, and unobtrusive controls.

B. User Requirements

Alongside these goals, the Wizard-of-Oz sessions helped us identify concrete user requirements. These requirements later became functional components of the Hi-Fi prototype:

- **Understandable content:** Users preferred concise summaries with optional bullet points for faster scanning and comprehension.
- **Readable display settings:** Adjustable font size, letter spacing, word spacing, line height, and theme options (light, sepia, dark) were essential for visual comfort.
- **Simple interactions:** Large buttons, minimal steps, and the ability to operate the system without logging in were important for reducing friction.
- **Audio support:** The system needed a text-to-speech engine with pause, play, and adjustable speed to support learners who benefit from auditory reinforcement.

- **Guidance and previews:** A short tutorial and real-time preview of reading settings were necessary to help users understand the impact of adjustments.
- **Multilingual support:** Users required seamless switching between English and Telugu, both for comprehension and for learning support.

These goals and requirements formed the foundation of the high-fidelity implementation. Every feature in the Hi-Fi prototype—ranging from translation and summarisation to spacing controls, bold emphasis, and audio playback—was directly derived from the needs identified during the low-fidelity testing phase. This ensured that the final system remained aligned with real user behaviours and cognitive accessibility needs.

IV. Accessibility and Reading Support Features

The system incorporates a set of carefully designed accessibility features intended to reduce cognitive load and improve readability for learners with dyslexia, ADHD, and low reading proficiency. These features are integrated directly into the interface of the High-Fidelity prototype and are fully operational within the Streamlit implementation. Rather than serving as static stylistic options, each feature is grounded in established principles of cognitive accessibility and reading science.

Font size adjustment is one of the core features provided by the system. Enlarging text has been shown to assist readers with visual strain, low vision, and decoding difficulties by increasing the perceptual clarity of individual characters. Users can increase or decrease the text size according to their comfort level, allowing them to read the simplified summary at a pace that suits their visual and cognitive needs. This flexibility ensures that the system adapts to a wide spectrum of reading abilities. In addition to font size, the system provides control over letter spacing. Increasing the spacing between individual characters has been widely demonstrated to improve word recognition for dyslexic readers by reducing visual crowding. Through the interface, users can gradually increase the distance between letters, creating clearer boundaries between adjacent characters and enhancing decoding accuracy. The feature is particularly effective for Telugu script readers, who often encounter difficulties due to dense ligatures and complex character clusters.

Line spacing control is another essential component of the system's accessibility toolkit. Adjusting the vertical distance between lines helps readers maintain their place in the text, prevents skipping or rereading errors, and reduces the overall density of the visual field. Increased line spacing supports readers with attention deficits by providing greater separation between lines, allowing the eye to track information more smoothly. The system enables users to modify line spacing dynamically, and the changes are rendered in real time within the Streamlit output.

Beyond visual adjustments, the system incorporates multimodal support through its audio playback functionality. The text-to-speech module includes a speed control mechanism that allows users to slow down or accelerate the narration. Slower playback helps learners who require more time to process spoken language, while faster speeds benefit advanced users who want a quick overview of the simplified summary. This flexibility allows the tool to accommodate diverse listening preferences and learning styles. The audio interface also includes play and pause controls, allowing users to stop the narration at any moment and resume when ready. This feature is particularly valuable for students who need breaks due to attention difficulties or cognitive fatigue. By providing fine-grained control over audio consumption, the system ensures that learners can engage with the content in a self-paced and manageable manner.

Another important accessibility feature integrated into the prototype is the option to apply bold emphasis to text. Bolding plays a significant role in guiding visual attention, especially for ADHD readers who may struggle to maintain focus on key elements of a sentence. By making certain portions of the text visually prominent, the bolding feature assists readers in identifying the most relevant terms or phrases. This method serves as a lightweight form of attention cueing and supports improved comprehension without altering the meaning of the content.

Finally, the system provides the ability to introduce additional spacing between words. Increasing word spacing reduces the likelihood of perceptual merging, where adjacent words appear visually fused, a common challenge for dyslexic readers. By spacing apart words, the system creates clearer segmentation within sentences, enabling smoother reading and reducing the cognitive effort required to parse complex text. The combination of adjustable word spacing, letter spacing, and line spacing results in a highly customizable reading experience tailored to the unique needs of individual learners.

Together, these visual and auditory features form a comprehensive accessibility framework within the Hi-Fi prototype. Each setting operates independently yet contributes to the larger goal of supporting cognitively diverse learners. By allowing users to personalize their reading environment, the system addresses the fundamental principles of inclusive design and provides a flexible, multimodal platform for improved reading comprehension.

V. System Implementation Details

The Hi-Fi prototype is powered by a lightweight but fully functional NLP pipeline implemented in Python and deployed through Streamlit. The core multilingual engine is the `facebook/nllb-200-distilled-1.3B` model, a 1.3B-parameter encoder–decoder Transformer capable of high-quality English–Telugu translation. The model uses a SentencePiece tokenizer and is loaded lazily to reduce memory overhead, occupying approximately 5.4 GB in FP16 format. This component enables the system to detect the input language, translate when needed, and maintain semantic consistency across languages. For Telugu-specific processing, we incorporate the Stanza POS pipeline, which provides sentence segmentation and part-of-speech tagging. These tags are used to identify and preserve key nouns during summarization and to enable noun highlighting in the final output. This step ensures that essential concepts remain intact, supporting readability and comprehension for learners with cognitive difficulties.

Text-to-speech output is generated using the gTTS engine, with optional speed modification handled by the `pydub` library. The audio is streamed directly to the interface as an in-memory MP3 file, supporting real-time playback with pause and speed control. Streamlit connects these components into an interactive interface, allowing users to paste text, view summaries, adjust spacing, and play audio with minimal latency. Despite using large multilingual models, the system runs smoothly on CPU-only machines, demonstrating that AI-assisted reading support can be delivered effectively even under practical resource constraints.

VI. High-Fidelity Prototype

The High-Fidelity (Hi-Fi) prototype represents the fully functional implementation of our proposed system and serves as a major progression from the low-fidelity conceptual design. While the low-fidelity prototype allowed us to explore user flow, overall layout, and cognitive accessibility-driven design choices, the Hi-Fi prototype integrates actual backend intelligence, natural language processing pipelines, bilingual translation, and dynamic text-to-speech synthesis. The system is implemented using the Streamlit framework, which provides a lightweight yet powerful interface for real-time interaction, rapid prototyping, and direct integration with Python-based AI models. The result is a working software environment where every functional component—summarization, simplification, translation, highlighting, and audio synthesis—is performed algorithmically rather than simulated.

The Hi-Fi prototype was designed to support both English and Telugu, reflecting the multilingual reading challenges faced by Indian students, especially those studying in state-board regional-medium schools. At the core of the language processing pipeline is the `facebook/nllb-200-distilled-1.3B` model, selected for its ability to handle low-resource languages such as Telugu while maintaining strong translation quality. Using this model, the system automatically detects the input text's script and converts it into the target language if required. This bidirectional translation ability enables users to receive simplified summaries in either English or Telugu, regardless of the input language. Such bilingual flexibility is essential for learners with different comfort levels in reading comprehension. The summarization process in the Hi-Fi prototype differs significantly between English and Telugu to better align with the linguistic properties and cognitive reading patterns of each language. English text is processed

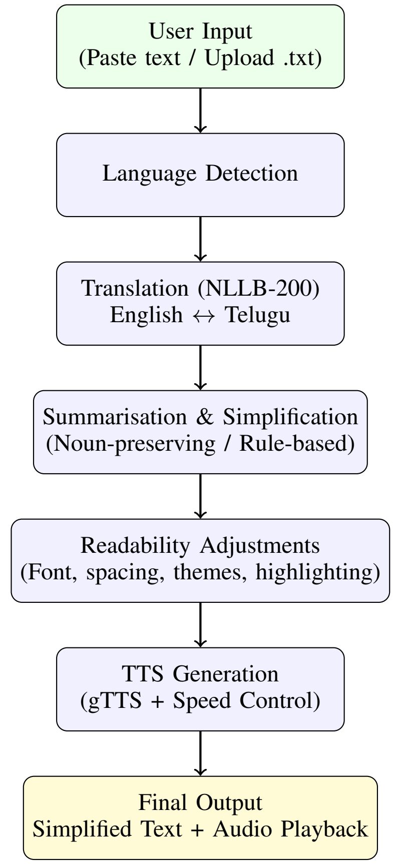

Fig. 1: System flow from user input to TTS output in the high-fidelity prototype.

through a basic extractive summarization mechanism that selects the most representative segments of the input based on sentence boundaries and density of meaningful content. However, Telugu requires additional linguistic considerations due to its agglutinative morphology and syntactic structure. Therefore, the system employs a noun-preserving extractive summarization method for Telugu. This method relies on Stanza's Telugu NLP pipeline, which performs sentence segmentation, tokenization, and part-of-speech tagging. All nouns and proper nouns are extracted, and each sentence in the input text is scored by counting the presence of these key nouns. The summary is constructed by prioritizing sentences with higher noun densities, ensuring that the essence of the passage—typically centered around people, objects, places, or ideas—is preserved. This approach aligns well with accessibility principles, particularly for readers with dyslexia or working memory challenges, who benefit from summaries that retain core semantic anchors.

Following summarization, an additional simplification stage is applied. This step removes complex vocabulary and restructures long or multi-clause sentences into shorter, more digestible segments. The vocabulary simplifier replaces academic or uncommon English terms with simpler synonyms and applies similar sentence restructuring techniques to Telugu by reducing unnecessary connectors and splitting long segments. The simplification functions also enforce a word limit defined by the user, ensuring that the output remains concise. This stage significantly reduces cognitive load and enhances readability, making the system suitable for students who struggle with dense academic passages. A unique accessibility-oriented feature included in the Telugu pipeline is the noun highlighting mechanism. The system uses POS tags to identify nouns and proper nouns in the simplified Telugu summary. These tokens are then visually emphasized in the final output, helping readers quickly locate the most important elements of the passage. Careful handling avoids highlighting errors caused by quotation marks or overlapping character spans. By selectively highlighting key nouns, the prototype enhances comprehension and supports attention retention, particularly for students with ADHD who often benefit from visually guided reading. Once the simplified summary is generated, users may optionally translate it into the other supported language. Translation is again carried out by the NLLB-200 model, ensuring semantic consistency across languages. This functionality opens the possibility for bilingual learning, enabling students to compare the same idea in two languages, thereby reinforcing understanding and improving confidence in reading skills.

The Hi-Fi prototype integrates a fully functional text-to-speech module implemented using Google's gTTS engine. The simplified or translated text is converted to speech and returned to the user as an audio stream. The system also provides audio speed control using the `pydub` library, which adjusts frame rates to generate slower or faster playback without distorting audio quality. This feature is designed to support learners who prefer auditory reinforcement, have difficulty with visual decoding, or require slower speech to maintain comprehension. All audio playback occurs directly within the Streamlit interface, creating a seamless multimodal reading experience. The user interface in the Hi-Fi prototype mirrors the structure of the low-fidelity design but replaces static placeholders with real, functional components. Users can type or paste text, upload files, select a target language, choose summary lengths, play audio summaries, and visually inspect highlighted tokens. Every interaction is instantaneous, with Streamlit refreshing the display to reflect newly generated content. The prototype therefore behaves as an end-to-end reading aid tool rather than a conceptual demonstration.

Taken together, the Hi-Fi prototype validates the technical feasibility of our system and translates design intentions into a fully working product. It demonstrates that AI-driven text simplification, when combined with multilingual processing and cognitive accessibility features, can be effectively implemented in practice. The prototype is robust enough to support preliminary usability testing and sets a strong foundation for future large-scale deployment and classroom evaluation.

VII. Evaluation Methodology

The high-fidelity prototype was evaluated through a structured peer-review study focusing on usability, readability support, summarization quality, audio clarity, and perceived cognitive effort. Because access to school-level users was limited, the evaluation was conducted with university-level peers who were capable of assessing interaction design, feature usefulness, and overall system behaviour. Participants used the working prototype and then completed a Google Form–based questionnaire. Details of the participant group and experimental procedure are provided in Section VII-A.

Participants performed core tasks, including entering text, generating summaries, adjusting readability settings (font size, spacing, and themes), and utilising the text-to-speech module with speed control. The form collected usability ratings using the System Usability Scale (SUS), capturing overall effectiveness, efficiency, and satisfaction. Summarisation quality was assessed through questions measuring clarity, coherence, and retention of key ideas. Readability features—including font adjustments, spacing controls, and bold-emphasis options—were evaluated for their usefulness in reducing visual effort. The audio section examined the clarity of the generated speech and the responsiveness of playback controls. Cognitive effort was assessed through a small set of self-reported questions on perceived ease or difficulty of completing tasks. Finally, participants provided open-ended comments on issues, suggestions, and opportunities for improvement.

This methodology provided a balanced mix of quantitative and qualitative data, offering a comprehensive understanding of prototype performance and informing areas for refinement.

A. User Study

A peer-based user study was conducted to evaluate the usability, readability features, summarisation quality, and audio capabilities of the prototype. A total of 10 university-level participants took part. All users were familiar with digital reading tools and comfortable with English and/or Telugu academic text. The evaluation was administered using a structured Google Form, completed immediately after interacting with the prototype.

Participants performed key tasks, including pasting text, generating simplified summaries, adjusting readability controls, switching themes, and using the text-to-speech playback feature. They then rated usability using the standard 10-item System Usability Scale (SUS), alongside additional questions evaluating summary clarity, readability improvements, audio clarity, and perceived cognitive effort. Open-ended responses captured qualitative impressions, usability challenges, and suggestions.

This study design enabled the collection of both numerical ratings and descriptive feedback, providing clear insights into real-world performance and indicating which system components may require further refinement.

VIII. Results

Participants' choices during onboarding provided meaningful signals about their reading preferences and accessibility needs. A noticeable proportion of users selected features associated

TABLE I: System Configuration of the Hi-Fi Prototype

Component	Specification
Model	NLLB-200 Distilled 1.3B for English–Telugu translation and script normalization.
Summarization	English: basic extractive. Telugu: noun-preserving extractive method using Stanza POS tags.
NLP Pipeline	Stanza (tokenization, sentence splitting, POS tagging).
Summary Length	User adjustable: 50–300 words (default: 120).
Simplification	Rule-based synonym replacement and optional clause splitting.
Assistive Features	Bold-first-grapheme (1–3), vowel bolding, letter-coloring, Telugu varṇa highlighting.
Font Size	14–30 px (default: 18 px).
Line Spacing	1.2–4.0 (default: 1.6).
Letter Spacing	0.00–0.30 em (default: 0.02 em).
Themes	Light, Sepia, Dark.
Text-to-Speech	gTTS with pydub-based speed adjustment (0.25x–2.00x).
Languages Supported	English and Telugu (fully interchangeable input/output).
Input Format	Paste input or upload ‘.txt‘ (UTF-8).
User Memory	Preferences stored using SQLite (font, spacing, assists, obstacles).

with reduced visual crowding, such as increased line spacing, larger font sizes, and wider letter spacing. Many users preferred the bold-first-varṇa option, suggesting that visual cueing helps with initial decoding. Theme preferences were split across Light and Sepia modes, indicating that high-contrast dark mode was less preferred for extended reading. Users also showed clear interest in audio support, with most participants adjusting speech speed rather than using the default setting. These choices collectively highlight the need for flexible, user-controlled accessibility settings rather than a single fixed presentation style.

A. System Usability Scale (SUS) Results

To assess overall usability, we administered the System Usability Scale (SUS), a standardized 10-item questionnaire widely used to evaluate interactive systems. Responses from ten participants were scored according to the standard SUS procedure, where positively worded items contribute $(\text{score} - 1)$ and negatively worded items contribute $(5 - \text{score})$, and the total is scaled to a value between 0 and 100.

The SUS scores obtained from participants were:

$$\{77.5, 52.5, 60.0, 67.5, 70.0, 50.0, 82.5, 50.0, 95.0, 100.0\}.$$

The average SUS score was **70.5**, which falls within the **“Good Usability”** range based on industry SUS benchmarks. This indicates that users generally found the system easy to learn, well-integrated, and comfortable to operate. Several participants rated the system exceptionally high (above 80), reflecting strong usability for a subset of users. Overall, the SUS results suggest that the prototype meets standard usability expectations and is suitable for further refinement and deployment.

Participant	SUS Score
P1	77.5
P2	52.5
P3	60.0
P4	67.5
P5	70.0
P6	50.0
Average	**62.9**

TABLE II: System Usability Scale (SUS) scores for the six participants.

To evaluate the quality of the summarisation component, we computed two widely used automatic metrics: ROUGE and BERTScore. ROUGE-2 captures bigram overlap between the system output and reference summaries, while BERTScore evaluates semantic similarity using contextual embeddings.

Metric	Precision	Recall	F1-score
ROUGE-2	0.6562	0.0674	0.1217
BERTScore	0.9489	0.8693	0.9071

TABLE III: Automatic summarisation evaluation results for ROUGE-2 and BERTScore.

The relatively higher precision but lower recall in ROUGE-2 indicates that the summariser preserves key bigrams while producing shorter outputs, which is typical of extractive noun-preserving methods. At the same time, the high BERTScore values show that, despite lexical reduction, the system maintains strong semantic fidelity to the reference summaries. Together, these results suggest that the summariser effectively balances brevity with meaning preservation, making it well suited for cognitive-accessibility applications where clarity and retention of essential ideas are more important than surface-level lexical overlap.

IX. Ethics and Risk Assessment

The study adhered to standard ethical guidelines for low-risk HCI research. Participation was voluntary, and no personally identifiable data were collected. All participants provided informed consent before engaging with the prototype or filling out the evaluation form. Since participants were university peers rather than minors or individuals with diagnosed disabilities, the study avoided sensitive populations and did not require institutional ethics approval.

The primary risks associated with the system relate to potential misinterpretation of simplified content, inaccuracies in machine translation, and over-reliance on automated summarisation. To mitigate these, the prototype displays both original and simplified text, allowing users to cross-verify meaning. Audio output is clearly marked as synthetic to avoid misattribution.

No user profile data are stored beyond temporary settings, and the system does not perform behavioural tracking or adaptive modelling. All text processing occurs on the user's machine through Streamlit, preventing third-party data exposure. These precautions ensure that the prototype remains safe and privacy-respecting during academic evaluation.

X. Future Directions

The current prototype represents a single horizontal slice of cognitive accessibility for Telugu, providing several user-facing features but limited technical depth and no broad expansion across multiple Indic scripts. Future work may progress along three key directions.

A. Horizontal Expansion Across Indic Languages

A primary direction for future development is to expand the system to support additional Indian languages, such as Gujarati, Bangla, Tamil, and Marathi. Achieving this will require script-specific preprocessing techniques, new summarisation datasets, and TTS pipelines tailored to the phonetic and orthographic characteristics of each writing system. Horizontal expansion would significantly increase the tool's relevance across diverse vernacular-medium classrooms in India.

B. Deeper Vertical Technical Development

Several accessibility-oriented features require deeper engineering efforts that were beyond the scope of this project. Future versions should integrate OCR for textbooks and scanned pages, implement grapheme-level dyslexia interventions for Indic scripts, and incorporate synchronised TTS highlighting. Advanced additions such as neural readability estimation and per-user adaptive simplification would further strengthen the system's technical depth and improve its alignment with cognitive accessibility goals.

C. Stronger Personalisation and Real-World Testing

A more mature system should include adaptive user profiles that remember preferences, automatically adjust spacing or summary length, and tune TTS pacing based on interaction behaviour. Beyond technical enhancements, real-world validation is essential. Evaluating the tool with clinically diagnosed dyslexic and ADHD learners, as well as testing on low-end mobile devices and unstable network conditions, will provide a more accurate understanding of how the system performs in actual classroom environments.

XI. Design Issues and Compromises

During the development of both the low-fidelity and high-fidelity prototypes, we encountered several design challenges that significantly shaped what could be implemented within the scope of this project. Although our design intentions were grounded in established accessibility principles, adapting them to Telugu and other Indic scripts introduced linguistic, technical, and interface-level constraints. The following subsections summarise the major challenges faced during the iterative design and implementation process.

A. Challenges in Implementing Dyslexia-Specific Letter Interventions

One of the earliest design challenges concerned our attempt to include dyslexia-oriented letter-level interventions, such as highlighting confusing grapheme pairs. While such adaptations are feasible in English, Indic scripts introduce far greater complexity. Individual characters often consist of multiple Unicode code points representing consonants, matras, and conjunct formations; modifying only part of an akshara risks breaking rendering entirely. Additionally, reliable OCR would be required to process uploaded scans or textbook pages, but current Telugu OCR tools struggle with combined aksharas and noisy inputs. Dyslexia itself is heterogeneous, and creating per-user confusion profiles or personalised glyph substitutions lies far beyond the project scope. These linguistic and technical constraints led us to exclude letter-level interventions and instead emphasise global readability improvements such as spacing, font size changes, chunked summaries, and integrated TTS.

B. Ensuring Cognitive Accessibility Across the Interface

Another major challenge was achieving cognitive accessibility across the entire UI, not just within the simplified text output. Ideally, every button, label, and instruction should feature clear contrast, adjustable font sizes, generous spacing, and minimal visual clutter. However, the Streamlit framework imposed limitations on how much we could customise built-in components. While the reading pane can be styled with accessible text, dropdown menus, sliders, and system labels cannot be resized or reformatted consistently. Balancing icons and text was particularly challenging for Telugu-medium users, who benefit from visual cues but still require clear labels for comprehension. These restrictions prevented complete adherence to accessibility guidelines, though TTS support helped mitigate some of these issues.

C. Discoverability Issues with Collapsible Readability Controls

A recurring design decision concerned the placement of readability controls for font size, spacing, letter spacing, themes, and bold emphasis. Displaying all options directly on the reading screen introduced clutter and increased cognitive load, but hiding them within a collapsible menu risked reducing discoverability. While collapsible menus keep the interface clean, some users may not immediately notice them or may struggle with the cognitive switching cost required to repeatedly open and close the panel. On small devices, the additional motor effort required to expand the menu also becomes noticeable. Despite these drawbacks, we adopted the collapsible design to maintain a distraction-free reading area.

D. Text-to-Speech Integration Limitations

Although TTS formed a core component of our prototype, it presented several practical limitations. High-quality neural voices are not consistently available for Telugu and other Indic languages, and the behaviour of speed and pause controls varies across accents. Users often expect syllable-by-syllable or word-synchronised highlighting, which remains technically infeasible within our implementation environment. While adjustable speed and replay features enhance accessibility, the overall multimodal experience still falls short of ideal standards for supporting individuals with dyslexia and ADHD.

E. Practical Challenges of Translation

Translation introduced additional complications, especially for users who requested bilingual or side-by-side summaries. Machine translation for Telugu may distort meaning, especially for academic vocabulary, and summarisation after translation can amplify these errors. Some concepts lack direct Telugu equivalents, and users sometimes prefer mixed bilingual text, which is difficult to generate automatically. Ensuring semantic consistency across translation and summarisation remains a challenge that exceeds the capabilities of our current pipeline.

F. Device and Bandwidth Constraints

Because the target users include students on low-end phones and government-issued tablets, the prototype had to be designed with strict performance constraints. This ruled out computationally heavy features such as dual-pane rendering, multiple simultaneous text views, large animations, or advanced web fonts for dyslexia iconography. Limited bandwidth also influenced how TTS and summarisation were triggered, encouraging minimalistic layouts and reducing the number of network calls. These constraints influenced the aesthetic and functional simplicity of the final interface.

G. Balancing Cognitive Simplicity with Feature Richness

Throughout development, we faced a tension between providing rich personalisation options and preserving cognitive simplicity. While users benefit from adjustable settings, too many visible controls increase cognitive load, especially for learners with ADHD. Designing an interface that remains simple, predictable, and uncluttered while still offering meaningful accessibility adjustments proved to be a challenging task. The final prototype strikes a balance but still contains multiple steps and menus that may be cognitively demanding for some users.

XII. Conclusion

This project examined how AI-based text simplification, translation, and multimodal reading support can be tailored to meet the needs of Indian vernacular-medium learners with attention and comprehension difficulties. Grounded in prior work on cognitive accessibility and supported by iterative low-fidelity testing with Wizard-of-Oz techniques, the study identified two primary user profiles: learners who rely on audio-dominant reading and those who benefit from visually clearer, summarised text. These insights informed the design of our two prototype directions and guided the development of the final high-fidelity system.

While the implemented prototype provides meaningful assistance through summarisation, translation, readability adjustments, and text-to-speech output, it represents only a single-language, horizontal-focused slice of the broader problem space. The system lacks horizontal expansion across the diverse Indic scripts used in Indian classrooms and does not yet incorporate deeper accessibility mechanisms such as dyslexia-specific grapheme manipulations or synchronised multimodal guidance. Nonetheless, the project demonstrates the feasibility and value of adapting cognitive-accessible reading technologies to vernacular contexts.

By integrating script-aware readability features with multimodal pathways for comprehension, this work represents an early step toward developing more inclusive reading tools for learners in India's diverse linguistic environments. Extending language coverage, deepening accessibility-oriented engineering, and validating the system with diagnosed learners of dyslexia and ADHD will be the next essential steps for achieving real-world educational impact.

Acknowledgments

We thank all participants in the Wizard-of-Oz sessions and the user study for their contributions to the HiFi prototype.

References

[1] S. E. Shaywitz and B. A. Shaywitz, "Dyslexia (specific reading disability)," *Biological Psychiatry*, vol. 57, no. 11, pp. 1301–1309, 2005.

[2] R. L. Peterson and B. F. Pennington, "Recent update on reading disability (dyslexia) focused on neuroimaging and interventions," *Brain Sciences*, vol. 11, no. 3, p. 304, 2021.

[3] M. J. Polonenko, J. F. Ghosh, et al., “Cognitive profiles of difficult-to-remediate children with developmental dyslexia,” *Journal of Learning Disabilities*, vol. 48, no. 5, pp. 486–500, 2015.
[4] M. J. Kofler et al., “Working memory and attention deficit/hyperactivity disorder: A meta-analytic review,” *Clinical Psychology Review*, vol. 40, pp. 91–107, 2019.
[5] R. A. Barkley, *Executive Functions: What They Are, How They Work, and Why They Evolved*. Guilford Press, 2012.
[6] H. W. Catts and A. Kamhi, “Language and reading disabilities,” *Allyn & Bacon*, 2006.
[7] E. S. Norton et al., “Functional neuroanatomical evidence for the double-deficit hypothesis of developmental dyslexia,” *Neuropsychologia*, vol. 50, no. 4, pp. 507–518, 2012.
[8] L. Rello and R. Baeza-Yates, “Good fonts for dyslexia,” 14:1–14:8, 2013.
[9] W3C Cognitive Accessibility Task Force, *Text simplification - cognitive accessibility*, https://www.w3.org/WAI/GL/task-forces/coga/wiki/Text_Simplification, 2021.
[10] M. H. Schneps et al., “Shorter lines facilitate reading in those who struggle,” *PLoS ONE*, vol. 8, no. 8, e71161, 2013.
[11] L. Rello et al., “Dyswebxia: A text accessibility model for people with dyslexia,” in *Proceedings of the 13th Web for All Conference*, 2016, 25:1–25:4.
[12] M. Zorzi et al., “Extra-large letter spacing improves reading in dyslexia,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 28, pp. 11 455–11 459, 2012.
[13] H. Saggion, *Automatic Text Simplification* (Synthesis Lectures on Human Language Technologies). Morgan & Claypool Publishers, 2017.
[14] M. Shardlow, “A survey of automated text simplification,” *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 1, pp. 58–70, 2014.
[15] S. G. Wood et al., “Use of text-to-speech technology to improve reading comprehension for students with reading disabilities,” *Journal of Special Education Technology*, vol. 33, no. 4, pp. 227–237, 2018.
[16] C. F. DiCarlo et al., “Assistive technology for reading: Text-to-speech programs,” *Journal of Technology and Teacher Education*, vol. 27, no. 2, pp. 203–228, 2019.
[17] A. S. Evmenova et al., “The impact of text-to-speech software on students’ performance,” *Journal of Special Education Technology*, vol. 25, no. 1, pp. 1–20, 2010.
[18] A. Dalvi et al., “Designing multi-script typography for indian languages,” *Design Issues*, vol. 36, no. 1, pp. 38–51, 2020.
[19] T. Azeez, *Tamil typography design and cultural expression*, Google Fonts Project, 2018.
[20] R. Prakash, *Reimagining indic scripts for the digital age*, AIGA Eye on Design, 2020.
[21] S. Nag et al., “Akshara-syllable mapping in brahmic scripts: Implications for learning to read,” *Writing Systems Research*, vol. 6, no. 1, pp. 54–68, 2014.
[22] P. Padakannaya et al., “An akshara to sound matching task for assessment of dyslexia in alphasyllabic languages,” *Dyslexia*, vol. 8, no. 3, pp. 162–172, 2002.
[23] A. Kumar and P. Joshi, “Text simplification for indian languages: Challenges and approaches,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–23, 2021.
[24] A. Baby et al., “Resources for indian language text-to-speech systems,” in *Proceedings of the 10th International Conference on Natural Language Processing*, 2016, pp. 37–43.

APPENDIX

This section presents the ten main screens of the high-fidelity prototype.

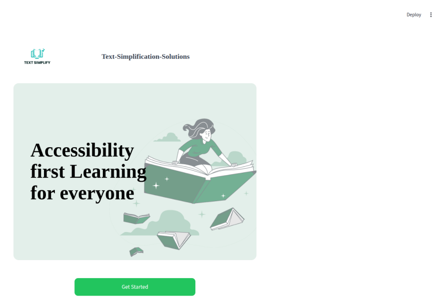

Fig. 2: Home page

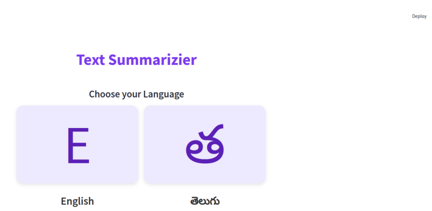

Fig. 3: Language Selection

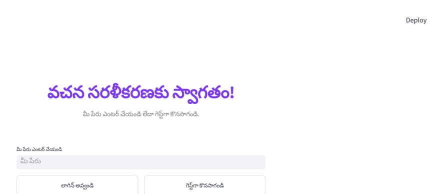

Fig. 4: Login Page

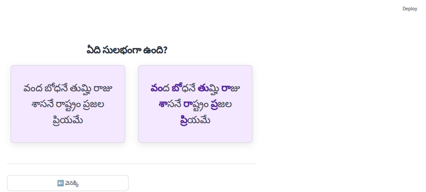

Fig. 5: Questionnaire 1 by bolding first varna

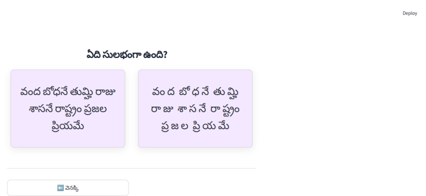

Fig. 6: Questionnaire 2 by more spacing

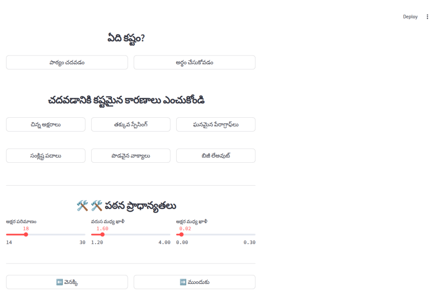

Fig. 7: User preferences

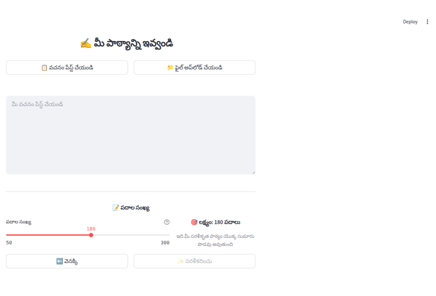

Fig. 8: Input text (paste or upload)

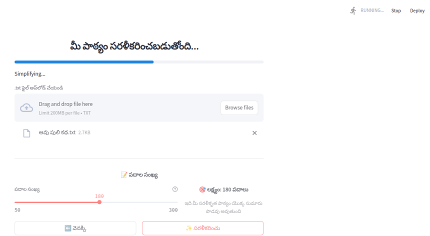

Fig. 9: Processing Page

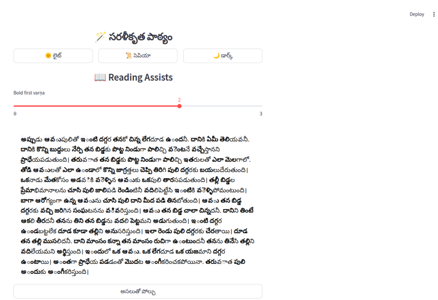

Fig. 10: Output text (simplified one)

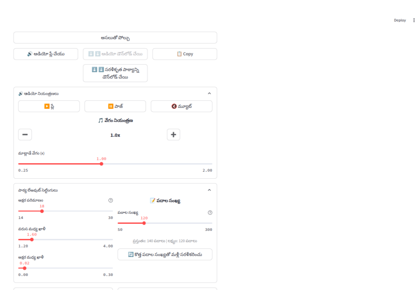

Fig. 11: Customisation of output text and TTS