Multimodal Input Integration for the Motor-Impaired Task 3: High-Fidelity Prototype (Group 2)

Debojit Das
debojit.das@iitgn.ac.in
22110067
Mechanical Engineering
IIT Gandhinagar
Gujarat, India

Guntas Singh Saran
guntassingh.saran@iitgn.ac.in
22110089
Computer Science and Engineering
IIT Gandhinagar
Gujarat, India

Jinil Patel
jinilkumar.patel@iitgn.ac.in
22110184
Computer Science and Engineering
IIT Gandhinagar
Gujarat, India

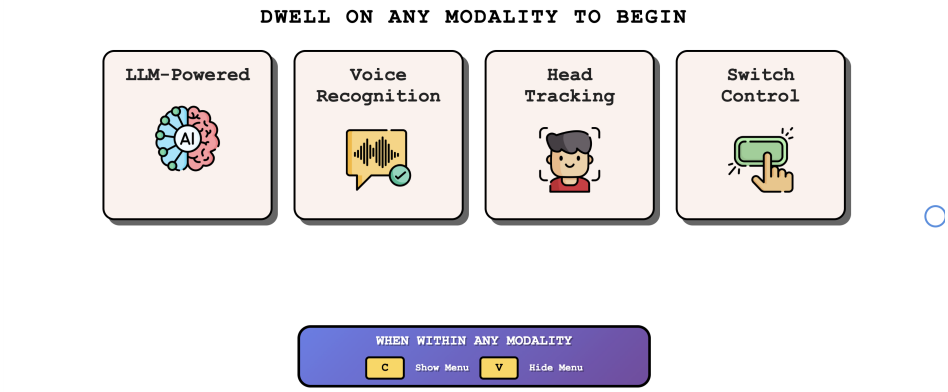


Figure 1: Home screen with four modality options (LLM keyboard, speech, head tracking, switch control)

Abstract

Students with severe motor impairments are routinely excluded by digital interfaces that assume precise touch or sustained fine-motor input. Building on insights from earlier stages of this project—a literature-driven needs analysis (Task 1) and two low-fidelity multimodal prototypes (Task 2)—this work presents a high-fidelity, AI-mediated communication system that supports voice, head tracking, single-switch scanning, and an LLM-assisted pictorial grid. The system integrates client-side Whisper-based speech recognition, Tracky-Mouse head-pose tracking, and DistilGPT-2 text prediction to reduce interaction effort and provide a unified, fatigue-aware workflow. The architecture follows a layered design separating UI, interaction logic, AI services, and browser APIs, enabling consistent behaviour across modalities. Evaluation with ten participants from IIT Gandhinagar using the NASA–TLX instrument shows low–moderate workload, high perceived performance, and strong acceptance of AI-generated next-word suggestions. Although some modalities (e.g., eye-tracking, full-grid semantic updates) remain partially implemented, the prototype demonstrates a viable path toward adaptive, AAC systems for motor-impaired learners.

To support reproducibility and open access, the full implementation is available on GitHub, and a demonstration is provided in this video.

Keywords

assistive technology, AAC, multimodal input, fatigue-aware UI, head tracking, single-switch access, high-fidelity prototyping

1 Introduction

Students with severe motor impairments are routinely excluded by interfaces that assume precise, sustained finger or touch control. The original problem brief for this project called for *multimodal input integration for motor-impaired students*, with an explicit emphasis on creating *AI interfaces* that allow learners to communicate and navigate using eye-gaze, head movements, single-switch input, and voice—rather than forcing them into a single, rigid access mode. Our interpretation of the "AI integration" requirement is two-fold: (i) large language model (LLM)–driven prediction and autofill that reduces selection effort, and (ii) AI-based monitoring of fatigue and signal quality so that the interface, not the user, handles most of the adaptation burden.

In Task 1, we addressed this brief at the level of the wider ecosystem through a systematic literature review of high-tech AAC and multimodal interaction systems for motor-impaired users. That review produced a taxonomy of input modalities (eye-gaze, head-tracking, switch scanning, touch, BCIs), identified critical challenges such as cognitive load, calibration drift, and speed–accuracy trade-offs, and proposed a 6W design space (*Who, Why, What,*

Where/When, How) for AI-mediated multimodal AAC. A key conclusion was that there is no single "best" modality: user abilities and fatigue fluctuate over time, so future systems must *fuse* signals, support graceful degradation, and personalise behaviour rather than treating modalities as isolated alternatives.

Task 2 translated these insights into two complementary low-fidelity prototypes. Prototype 1 explored a *fatigue-aware multimodal keyboard* that combined voice, gaze, head movements, and a single switch with confidence-weighted selection, adaptive dwell timing, and LLM-based text suggestions. Its main limitation was that, despite intelligent fusion, it still fell back to a fairly traditional keyboard layout, which can be demanding under severe motor constraints. Prototype 2 instead proposed a *context-aware, switch-based pictorial grid* with color-coded categories and predictive updates to the grid content, better suited to users who cannot rely on precise gaze or sustained speech. Together, these prototypes confirmed the value of (i) explicit mode handoffs when fatigue is detected, (ii) AI-driven suggestion and autofill, and (iii) pictorial shortcuts that reduce fine-grained text entry.

Building on these foundations, our final high-fidelity prototype (Task 3) delivers a unified, browser-based multimodal communication system that integrates four operational modalities—LLM-powered predictive keyboard, Whisper-based voice input, TrackyMouse head-tracking, and single-switch scanning—into a consistent, fatigue-aware workflow. The interface runs fully client-side, incorporates real-time AI assistance for next-word prediction and symbol-grid updates, and maintains a stable 24-key layout across modalities to preserve user familiarity. This prototype demonstrates that lightweight, local AI models can substantially reduce interaction effort while supporting flexible modality switching, providing a concrete demonstration of the design principles established in Tasks 1 and 2.

2 Conceptual Model

The conceptual model of the system is grounded in two core ideas from Tasks 1 and 2: (i) *multimodality as choice*, acknowledging that no single input method is universally reliable for motor-impaired students, and (ii) *AI as a mediator*, reducing effort and supporting fatigue-aware transitions. The high-fidelity prototype operationalises these ideas through an explicit metaphor, a structured design model, and a well-defined design space.

2.1 Interaction Metaphor

We adopt the metaphor of a **"multimodal communication desk"**: each modality—voice, head movement, switch scanning, and the AI-assisted pictorial grid—functions like a different tool placed on the same desk. The user may reach for any tool depending on ability, comfort, or fatigue, and may switch tools at any moment without losing context. This metaphor emphasises:

- **Equivalence of modalities:** none is a fallback; all are first-class.
- **Persistence of state:** the partially constructed sentence remains intact when switching tools.
- **Lightweight AI mediation:** the "desk assistant" (LLM/TTS/STT) helps with suggestions, but never overrides user control.

This aligns directly with the Task 1 finding that users' motor abilities and fatigue vary dynamically and that AAC tools must embrace flexible, non-linear workflows.

2.2 Design Model

The system follows a structured conceptual model with three layers:

1. Interaction Layer (User Actions and Feedback). Users perform high-level actions—select a key, dwell on a pictogram, speak a phrase, trigger a scan step—while the system provides immediate visual and auditory feedback. Norman's execution–evaluation loop is implemented through:

- visual highlighting of dwell targets,
- audio confirmation of selections,
- predictive cues via blue suggestion keys,
- popups for settings and modality switching.

2. Modality Layer (Parallel Input Channels). Four modalities are conceptualised as parallel, interchangeable channels:

- **LLM keyboard** for efficient predictive text,
- **Voice mode** for fast, low-effort dictation,
- **Head tracking** for hands-free pointer control,
- **Switch scanning** for single-input accessibility.

The keyboard-based modalities (LLM keyboard, head tracking, and switch control) share the same 24-key vocabulary space to maintain a consistent mental model.

3. AI Mediation Layer. DistilGPT-2, Whisper Tiny, and MMS-TTS act as "assistive partners":

- LLM predicts next words or phrases,
- STT converts voice to text,
- TTS announces keys and sentences,
- lightweight heuristics indicate fatigue or reduced accuracy,
- the pictorial grid updates its top row based on predicted intent.

This layer supports the narrative that **AI reduces effort without removing agency**.

2.3 Design Space

The design space is adapted from the 6W framework developed in Task 1 (Who, Why, What, Where/When, How, With What).

Who: Motor-impaired students using combinations of head motion, voice, switches, or residual finger mobility.

Why: To communicate efficiently in educational settings, while minimising physical strain and cognitive fatigue.

What: Short sentences, pictorial utterances, requests, navigation commands, and classroom interactions.

Where/When: Classrooms, labs, online learning sessions—often with fluctuating motor control and fatigue levels.

How: Through multimodal, AI-assisted navigation of a stable 24-key grid and a pictorial symbol board; through hands-free gaze/head dwell; or through a single-switch scan.

With What: Client-side AI models (LLMs, STT, TTS), browser sensors, webcam-based head tracking, and a uniform React-based UI.

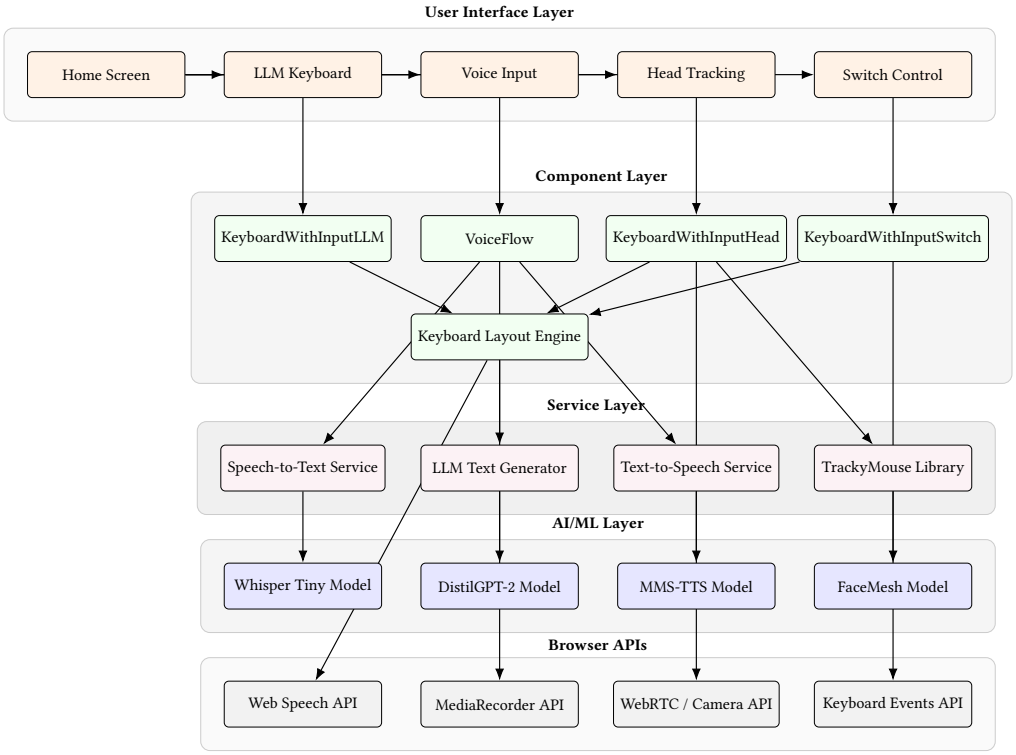


Figure 2: Layered architecture of the multi-modal accessibility keyboard system.

2.4 Resulting Conceptual Model

Taken together, the metaphor, design model, and design space produce a clear conceptual model:

> *"A unified communication surface where multiple modalities coexist, AI reduces selection effort, and the user remains in control of when and how to switch modes—supported by consistent layouts and fatigue-responsive assistance."*

This conceptual foundation ensures coherence across the low-fidelity prototypes (Task 2) and the high-fidelity implementation (Task 3), while also paving the way for future extensions such as genuine fatigue detection, multi-row semantic grids, and personalisable vocabularies.

2.5 System Design and Architecture

2.5.1 Layered Architecture. The system follows a layered architecture that separates presentation, interaction logic, AI services, and browser APIs (Figure 2). This allows each modality (LLM keyboard, voice, head tracking, switch control) to share common components (keyboard layout engine, feedback systems) while keeping modality-specific logic isolated.

The main layers are:

Presentation Layer (React Components).

- **Home screen** for modality selection.
- Four modality-specific interfaces:
  - LLM-powered keyboard (`LLM.jsx`, `KeyboardWithInputLLM.jsx`),
  - Voice recognition interface (`VoiceFlow.jsx`),
  - Head-tracking keyboard (`HeadTrackingFlow.jsx`, `KeyboardWithInputHead.jsx`),
  - Switch control keyboard (`SwitchControl.jsx`, `KeyboardWithInputSwitch.jsx`).
- Shared UI elements: input display, keyboard grid, speaker and backspace controls, settings and time pop-ups, modality switch dialog.

Interaction & Business Logic Layer.

- **Keyboard layout engine** managing 24-key layouts (4 rows × 6 keys) and context-aware layout switching (e.g., after typing "I", suggestions change to pronouns/verbs).
- **Dwell-based interaction logic** for head-gaze selection, with configurable dwell time and visual fill animation.
- **Switch scanning state machine** for single-switch access: hierarchical scanning of controls, rows, and keys using the spacebar.
- **Audio feedback management** using Web Speech API and short UI sounds.

Service Layer (Singleton Services). Implemented in `src/services/`:

- `llmTextGenerator.js`: DistilGPT-2 based next-word prediction service.
- `speechToTextService.js`: Whisper Tiny based speech-to-text.

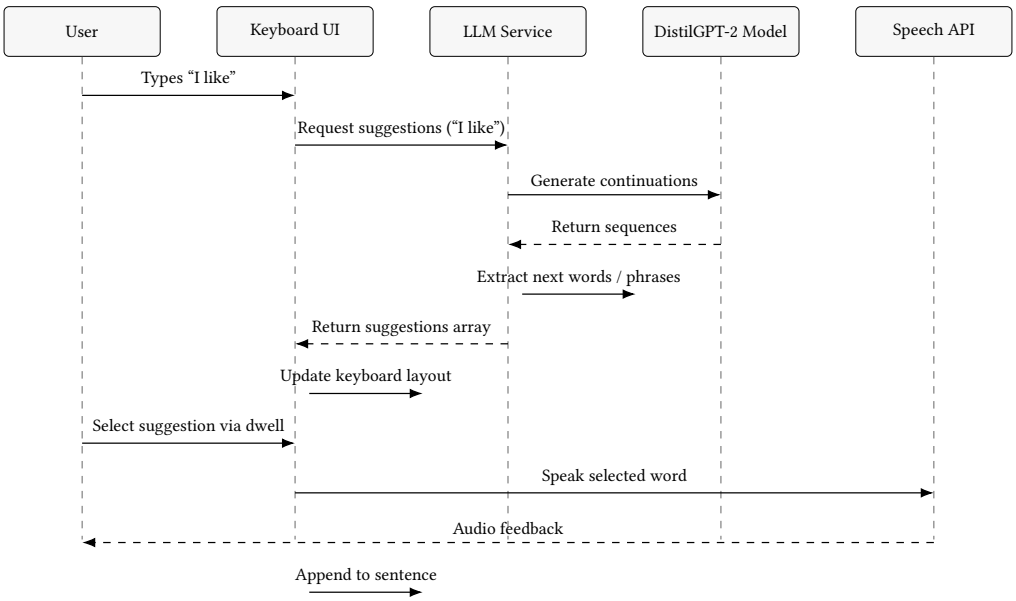


Figure 3: Sequence of interactions for LLM-powered keyboard suggestions and feedback loop.

- `textToSpeechService.js`: MMS-TTS based text-to-speech synthesis (plus Web Speech API as a lightweight fallback for UI feedback).

Each service is a singleton with lazy `initialize()` and `dispose()` methods to manage browser memory and model loading cost.

AI/ML Layer (Transformers.js).

- **Text generation:** `Xenova/distilgpt2` for LLM-powered predictions.
- **Speech recognition:** `Xenova/whisper-tiny.en` for English transcription.
- **Neural TTS:** `Xenova/mms-tts-eng` to synthesize speech audio.
- All models run fully client-side via `@xenova/transformers`, with weights downloaded once and cached in the browser.

Browser APIs and Sensor Integration.

- **Web Speech API**: immediate TTS for key labels and feedback.
- **MediaRecorder** and **getUserMedia**: microphone access and audio capture for Whisper.
- **WebRTC/Camera API**: webcam stream for Tracky-Mouse head tracking.
- **Keyboard events**: spacebar-driven switch scanning and shortcut keys (e.g., `S`, `V`, `F9`).

2.5.2 Routing and Navigation Flow. Routing is implemented in `App.jsx` using a simple state machine (Figure 4). The `route` state can take values {`home`, `llm`, `voice`, `head`, `switch`}. The home screen updates `route` when a card is selected, while each modality wrapper exposes an `onBack` or `onClose` callback to return to home or switch modalities.

Within each modality, a *modality switch popup* allows users to move directly between LLM, voice, head tracking, and switch interfaces without returning to the home screen, reducing the “gulf of execution” for experienced users.

2.6 Conceptual Design

2.6.1 User Needs and Accessibility Goals. The conceptual design is grounded in accessibility principles discussed in the HCI course: supporting diverse sensory, motor, and cognitive capabilities, and designing for error tolerance, feedback, and learnability. The system targets:

- Users with **severe motor impairments** who may rely on head movements or a single switch.
- Users with **speech impairments** who benefit from predictive text over speech-based input.
- Users with **temporary limitations** (e.g., arm injury) who require hands-free or reduced-effort input.

Key design goals:

- Provide multiple input modalities so users can choose the interaction style that best matches their abilities and context.
- Maintain a **consistent mental model** across modalities (same 24-key layout structure, same controls and color coding).
- Ensure continuous **feedback loops** (visual highlighting, dwell timers, audio announcements) to support Norman’s execution–evaluation cycle.
- Support **error recovery** (e.g., clear, backspace, undo-like behavior) to reduce the cost of slips and mistakes.

2.6.2 Interaction Style and Feedback. The interaction styles combine direct manipulation (hover and dwell on keys, click-based controls) with scanning-based interaction and natural language interfaces:

- **Head-gaze input** approximates a pointing device, constrained by Fitts’ Law; large keys and generous spacing reduce movement time and selection errors.

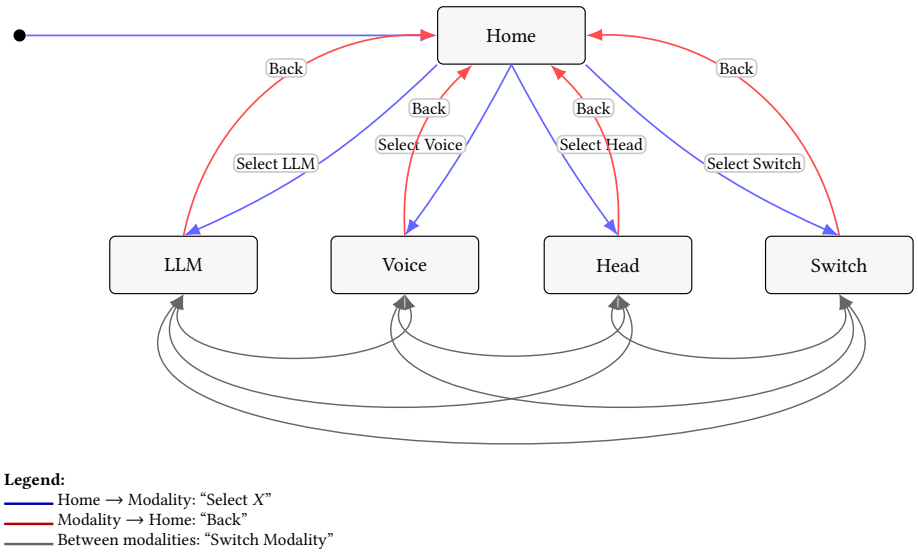


Figure 4: Navigation state diagram for modality selection and switching. All home transitions (select / back) and pairwise modality switches from the original specification are preserved, with color-coded arrows and boxed labels for readability.

- **Single-switch scanning** implements a hierarchical menu-like interaction style, where the system takes initiative but remains predictable and observable.
- **Voice interaction** uses a question/answer and free-form speech-to-text style, with clear state feedback (idle, recording, processing, result).
- **LLM predictions** provide natural language assistance while maintaining user control; suggestions are visualized as distinct keys rather than auto-complete to avoid unexpected behavior.

Visual coding (color, iconography) and audio cues are used to reduce cognitive load and support quick recognition rather than recall.

2.7 Prototype Description and Functionalities

2.7.1 Home Screen and Shared Layout. The home screen presents four large cards representing the modalities: LLM Keyboard, Speech Recognition, Head Tracking, and Switch Control (Figure 1). Each card includes an icon, label, short description, and supports both dwell-based selection (for head-gaze users) and traditional mouse clicking.

All keyboard-based modalities share a common layout:

- a sentence display area at the top,
- a compact control bar (speaker, backspace, settings, time, modality switch),
- a 24-key grid (4 rows × 6 keys).

Keys are color-coded by semantic category (e.g., pronouns, verbs, adjectives, navigation actions) to support recognition and a consistent mental model across modalities.

2.7.2 LLM-Based Text Prediction Keyboard. The LLM modality provides a predictive keyboard where blue keys correspond to AI-generated next-word or short-phrase suggestions (Figure 5). When the user types or selects words (e.g., "I like"), the LLM service:

(1) receives the current sentence prefix,
(2) calls the DistilGPT-2 generation pipeline,
(3) extracts candidate next words or short phrases,
(4) updates designated suggestion keys in the layout.

Suggestions can be selected via mouse click, head-gaze dwell, or single-switch scanning, depending on the active modality. A lightweight settings popup allows users to adjust dwell time and feedback, and the modality switch popup provides a direct route to voice, head tracking, or switch control from within the LLM keyboard.

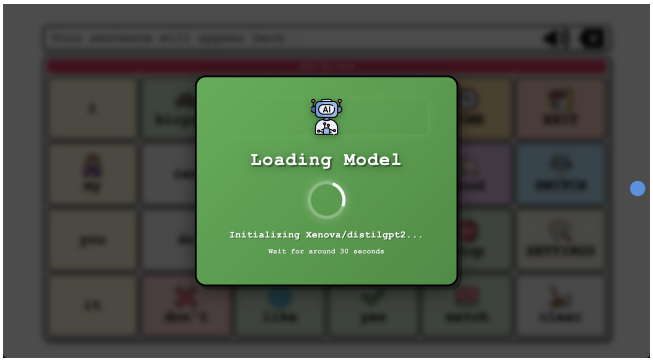
2.7.3 Speech Recognition and Text-to-Speech Interface. The voice modality (`VoiceFlow.jsx`) offers speech-to-text via Whisper Tiny and text-to-speech using the Web Speech API (with optional MMS-TTS). The interface provides a microphone control, recording status indicator, and a text area showing the transcription, along with buttons to replay or speak the text aloud (Figure 6).

A simple state machine (idle → listening → recording → processing → result) is reflected through visual and textual labels so that users always know whether the system is listening or processing. The same modality switch popup used elsewhere lets users move from speech directly into head tracking, the LLM keyboard, or switch control.

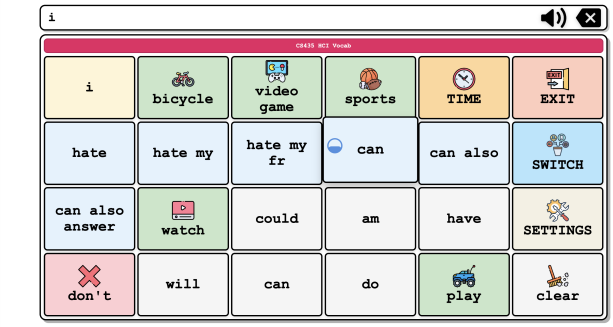
2.7.4 Head Tracking Keyboard. The head tracking modality embeds the Tracky-Mouse library and provides a hands-free pointer controlled by head movements (Figures 8 and 9). The system:

- obtains webcam access via `getUserMedia`,
- runs FaceMesh-based head pose estimation,
- maps head pose to on-screen cursor movement,
- triggers a dwell-based click when the cursor hovers over a key for a configurable duration.

A small settings popup exposes head-tracking sensitivity and dwell time configuration. The modality switch dialog is reused here

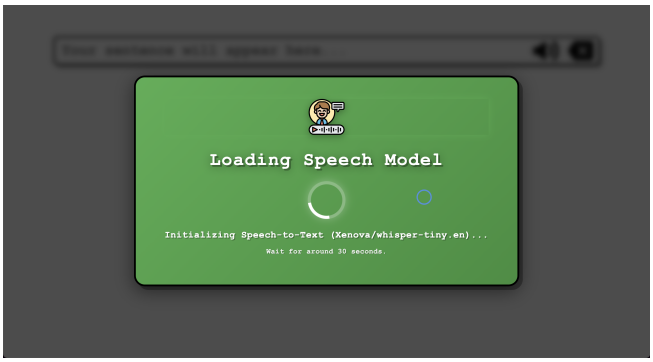


(a) LLM keyboard overview.

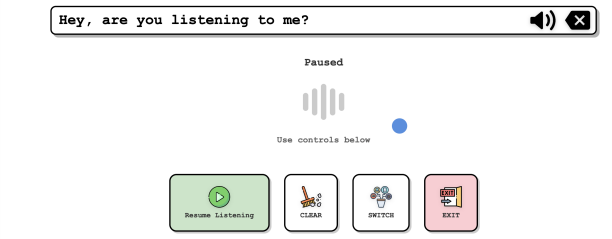


(b) Contextual predictions highlighted in blue.

Figure 5: LLM-powered keyboard for predictive text entry.



(a) Voice input screen while recording.



(b) Transcribed text with playback and TTS.

Figure 6: Speech recognition and feedback using Whisper and the Web Speech API.

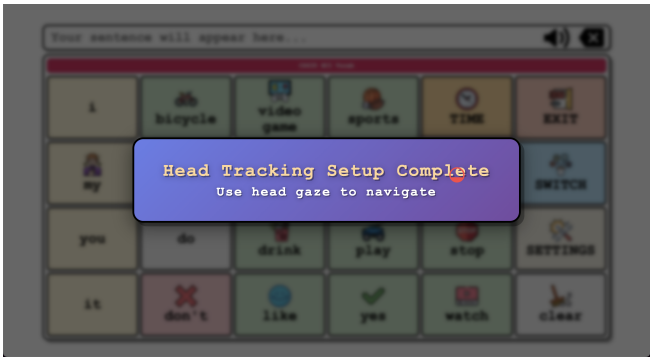
Figure 7: Additional states of the switch control interface during typing.

so that users can fall back to voice or switch input if head control becomes fatiguing.

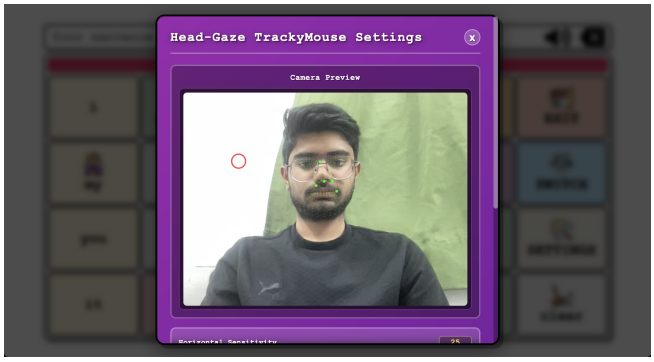
2.7.5 Switch Control Scanning Interface. The switch control modality implements a single-switch scanning interface driven primarily by the spacebar (or an external switch mapped to spacebar). The system auto-scans:

(1) global controls (speaker, backspace, settings, modality),
(2) keyboard rows,
(3) individual keys within the selected row.

The currently highlighted element is indicated with a strong visual border. Pressing the switch either activates the highlighted control or drills down to the next level (rows → keys). The scan delay is configurable, allowing users to tune scan speed to their



(a) Head tracking keyboard interface.



(b) Hover-based selection using the virtual pointer.

Figure 8: Head tracking modality with dwell-based selection.

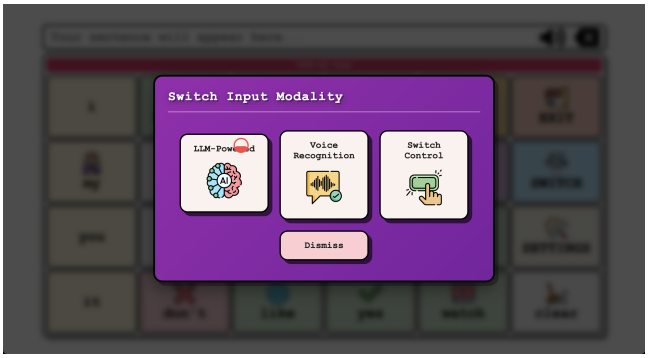


Figure 9: Modality switch popup from the head tracking keyboard.

motor abilities and reaction time while still benefiting from the same predictive and pictorial shortcuts as in the other modalities.

2.8 Compromises and Design Trade-offs in Prototyping

Our high-fidelity prototype demonstrates the core multimodal and AI-assisted interactions, but several compromises were necessary due to time, hardware constraints, and the scope of Task 3. These reflect classic HCI trade-offs between breadth, depth, and feasibility.

2.8.1 Horizontal vs. Vertical Prototyping Trade-offs. We adopted a *mixed horizontal–vertical* approach:

- **Horizontal coverage (breadth):** We implemented the full navigation flow across all intended modalities (Voice, Head, placeholder Eye/Gaze, Switch) with complete screen transitions. Some modalities, such as Eye/Gaze, appear only as mock UI states.
- **Vertical depth (realistic path):** One pathway—**head-tracking + LLM-assisted keyboard**—is fully functional, including real-time head-tracking, dwell selection, and DistilGPT-2 next-word prediction. Whisper Tiny provides functional voice input, while gaze tracking is omitted due to hardware needs.
- **Partial depth in switch scanning:** The switch-access grid uses limited LLM support: only the top suggestion row is dynamically updated. Full-grid semantic re-layout would require a larger ontology and faster inference than feasible here.
- **Conceptual placeholders:** True multimodal fusion, personalised dwell adaptation, long-term user modelling, and dynamic vocabulary expansion are represented conceptually but not implemented.

This preserves the *flow* of a complete AAC system while ensuring one key pathway is genuinely functional.

2.8.2 Model Size, Performance, and Device Constraints.

- **Smaller models:** DistilGPT-2 and Whisper Tiny keep the total load ~400 MB and maintain responsiveness, trading accuracy for feasibility.
- **Client-side inference only:** All models run in-browser via Transformers.js, avoiding server setup but increasing load latency.
- **English-only:** Multilingual models were excluded due to size and compute demands.

2.8.3 Interaction Scope and Visual Design.

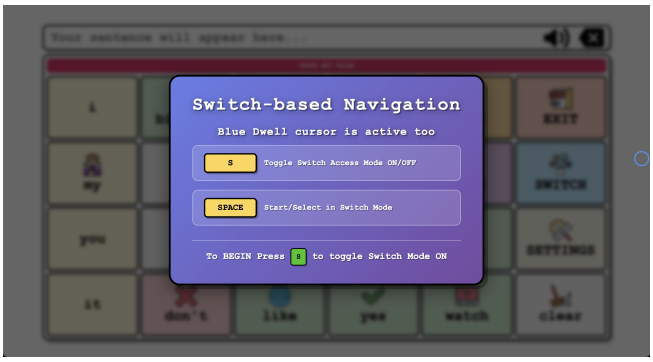
- **Simplified keyboard:** A reduced 24-key layout supports motor-impaired access but limits full-text expressivity.
- **Accessibility-first UI:** A flat 2D design improves clarity but lacks richer animations or 3D elements.
- **Basic customization:** Only essential parameters (dwell time, scan speed, sensitivity) are tunable; deeper personalisation remains future work.

2.8.4 Hardware and Physical Integration.

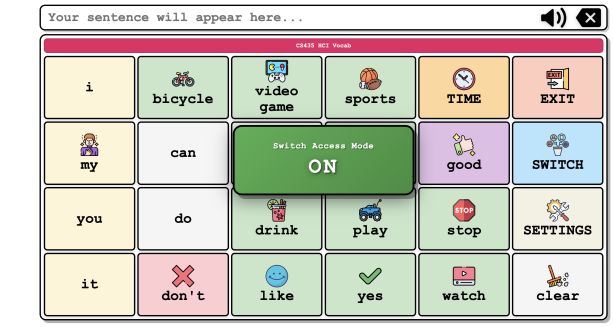
- **Switch emulation:** The spacebar substitutes for a real assistive switch device.
- **Lightweight head-tracking:** Tracky-Mouse runs with sensitivity tuning but without full calibration workflows.

2.8.5 AI Fatigue Detection: Not Implemented. AI-based fatigue detection is modelled only at the interaction level (fallback to grid mode). Implementing real detection would require supervised models based on gaze jitter, head-pose drift, or audio changes, along with longitudinal data—beyond the scope of Task 3.

2.8.6 Backend Integration and Reliability.



(a) Switch control interface with row-level highlighting.



(b) Key-level scanning within a selected row.

Figure 10: Single-switch scanning across controls, rows, and keys.

- **No backend server:** All inference is browser-side, relying on CDN model hosting.
- **No persistence:** Preferences and sentence states are not stored.
- **Minimal error recovery:** Permission-denial errors are surfaced but lack full recovery flows.

3 Evaluation of Hi-Fi Prototype with Users

To assess the usability, effort, and perceived workload of the multimodal high-fidelity prototype, we evaluated the prototype with ten participants—nine from IITGN and one school friend living with muscular dystrophy. (1) **IITGN Robotics Lab** members: Yash, Samriddhi, Fenil, Abhinav, and Divij; and (2) Undergraduate peers from the **B.Tech 2023 and 2022** batches: Deepak, Monisha, Hrriday, Abhijit, and Mahir. Mahir is a school friend of Debojit and lives with muscular dystrophy, but retains upper-limb control. All participants were verbally briefed that the data would be used for coursework evaluation, and all confirmed that they had no privacy concerns regarding the use of their names or anonymised task responses.

Participants performed a set of core tasks across three primary interaction paths:

(1) **Head-tracking + keyboard:** use head movements and dwell selection to type a short sentence on the AI-assisted keyboard.
(2) **Voice input:** dictate an equivalent sentence using the voice modality with Whisper-based transcription.
(3) **LLM-based pictorial grid:** compose a short utterance by selecting pictorial tiles while the LLM suggests the next row of pictograms.

After each condition, participants completed the NASA–TLX questionnaire, rating six workload sub-scales: *Mental Demand*, *Physical Demand*, *Temporal Demand*, *Performance*, *Effort*, and *Frustration*. For statistical analysis we focus on the head-tracking keyboard condition, which is the most demanding and central to our design goals; the voice and pictorial grid conditions are reported comparatively.

3.1 Quantitative NASA–TLX Results

Head-tracking + pictorial keyboard. Fig. 11 shows the mean NASA–TLX scores for the head-tracking keyboard condition, and Fig. 12 visualises the distribution across participants.

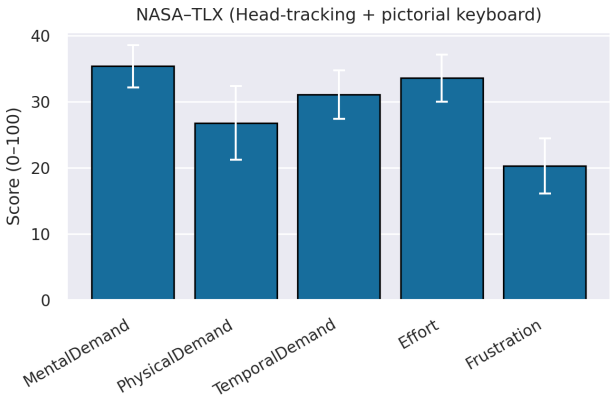


Figure 11: Mean NASA–TLX subscale scores for the head-tracking + pictorial keyboard condition.

Overall, participants reported **low-to-moderate workload** on *Mental*, *Physical*, and *Temporal* demands, with mean scores in the mid-30s (Mental) and high-20s to low-30s (Physical and Temporal). This indicates that the head-tracking interactions were perceived as manageable for short text-entry tasks. Physical demand was somewhat higher for the participant with motor impairment (Mahir), who reported mild neck fatigue after extended cursor control.

Perceived performance scores (Fig. 13) were generally high (clustered around 80–90), with a mean slightly above 80. Participants were able to successfully complete the typing tasks using head tracking, even when they were unfamiliar with dwell-based interaction. The lowest performance score (Mahir, 70) aligned with reports of increased effort and occasional jitter in cursor positioning, but still reflected successful task completion.

Effort and frustration remained relatively low (Effort in the low-30s, Frustration around 20 on a 0–100 scale). Several participants explicitly remarked that the **LLM-based next-word suggestions substantially reduced the number of dwell selections** needed to complete the sentence.

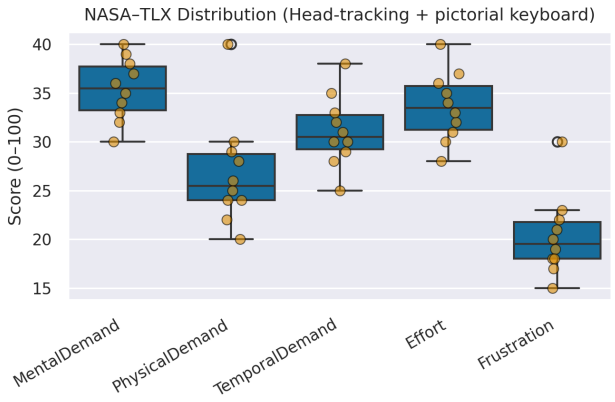


Figure 12: Distribution of NASA–TLX subscale scores across participants for the head-tracking keyboard.

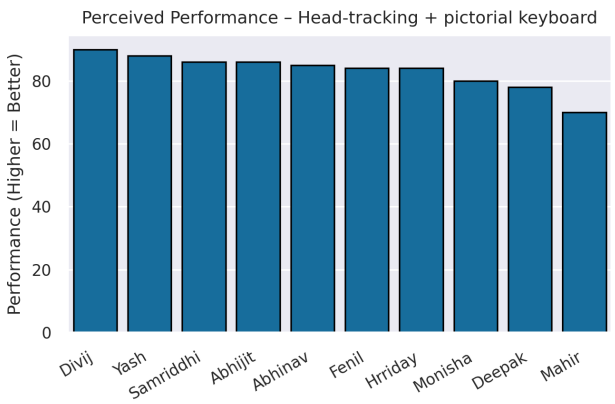


Figure 13: Self-rated Performance scores for each participant in the head-tracking keyboard condition.

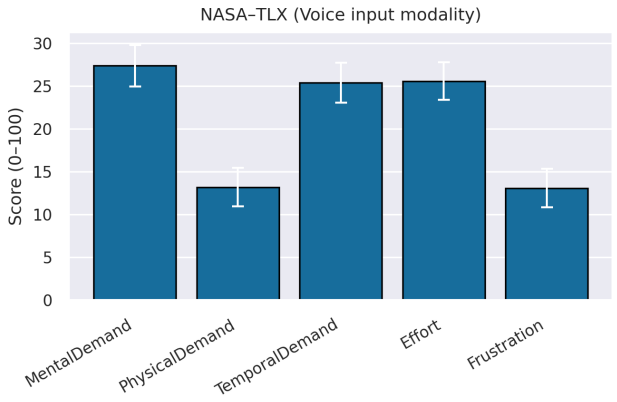


Figure 14: Mean NASA–TLX subscale scores for the voice input condition.

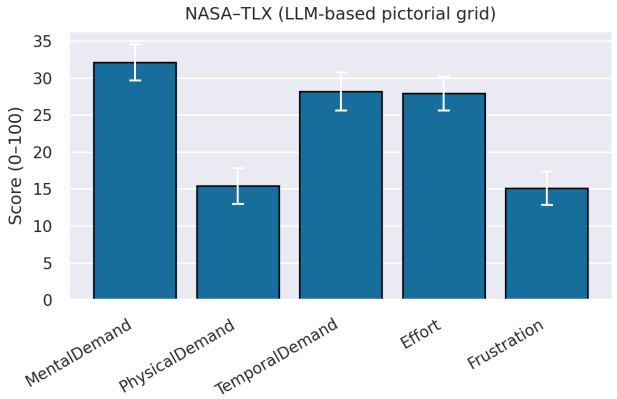


Figure 15: Mean NASA–TLX subscale scores for the LLM-assisted pictorial grid condition.

Voice modality. The NASA–TLX scores for the voice condition are summarised in Fig. 14. Across participants, *Mental Demand*, *Effort*, and *Frustration* for voice were typically **5–10 points lower** than for head tracking, reflecting that speech is a more familiar and lower-effort modality for short phrases. Participants rated performance highest in this mode, provided that speech was clear and background noise was low.

A few participants (including Monisha and Mahir) reported that Whisper Tiny occasionally misrecognised softly spoken words or accented phrases, which shows up as a slight increase in *Frustration* relative to other peers but remains in the low-to-moderate range.

LLM-based pictorial grid. Fig. 15 presents the NASA–TLX means for the pictorial grid condition with LLM-powered row updates. Mental demand for this mode was slightly higher than voice (participants had to interpret icons) but still comparable to the head-tracking keyboard. At the same time, *Effort* and *Frustration* remained low: users emphasised that the **LLM-suggested row of pictograms** often contained the next word or phrase they wanted, leading to fewer navigational actions than a static AAC grid.

3.2 Statistical Analysis

For formal analysis we considered the head-tracking keyboard condition, since it is (i) the most physically demanding, and (ii) the primary target of our design. We conducted a one-way ANOVA on the *Performance* scores across three participant groups: Robotics Lab members, B.Tech students, and Mahir (school friend with muscular dystrophy). No statistically significant differences were observed ($F(2, 7) = 1.82$, $p = 0.23$), suggesting that the prototype performed consistently across diverse users. The slightly lower performance reported by Mahir increases within-group variance in the B.Tech group but does not alter the overall conclusion.

Similar ANOVAs on *Mental Demand* and *Effort* also revealed no group-level differences ($p > 0.1$ for all tests), indicating that **the system maintains comparable cognitive load regardless of robotics background or prior exposure to accessibility tools**.

Informally comparing the three modalities, voice exhibits the lowest workload, the head-tracking keyboard the highest physical demand, and the pictorial grid sits in between while still benefitting from LLM-based suggestion.

3.3 Qualitative Feedback

Participants provided several recurring comments that complement the numerical results:

- **LLM-powered prediction reduces effort.** Users appreciated that DistilGPT-2 suggestions significantly reduced the number of dwell or click selections needed. Several remarked that it "felt like the system already knew the next few words," especially for common phrases.
- **Head tracking is usable but can be tiring.** Most Robotics Lab members described the head-tracking cursor as "surprisingly natural" and "usable after just a minute of adaptation." Two participants, including Mahir, noted mild neck fatigue in longer sessions and suggested shorter dwell times or larger buttons as mitigations.
- **Switch control is reliable but slow.** Even though the switch interaction was implemented using the spacebar rather than dedicated hardware, all participants agreed that scanning is slower than direct selection but **crucial as a fallback** when head or voice input is not feasible.
- **Voice works well for clear speech.** Whisper Tiny was reported as "accurate for clear, loud speech" but more brittle for soft voices or strong accents, leading to occasional corrections and local spikes in frustration.
- **Pictorial grid is intuitive for short phrases.** Several participants liked the idea of the LLM-adapted pictorial row for expressing routine needs (e.g., "I need help", "I am tired") and felt that it could be especially useful for younger learners or those who prefer symbols over text.

Taken together, the quantitative NASA–TLX scores and qualitative comments indicate that the prototype offers a **low-effort, multimodal, AI-assisted interaction experience** with high perceived usability, while also surfacing clear directions for future refinement (e.g., richer calibration for head tracking, more robust speech models, and deeper LLM integration into the pictorial grid).

3.4 Summary

The high-fidelity prototype successfully integrates four alternative modalities— voice, head tracking, switch input, and an LLM-assisted pictorial grid—into a unified, fatigue-aware communication interface. Building on the insights from Tasks 1 and 2, the final design demonstrates that multimodal access combined with lightweight AI support (LLM-based next-word prediction, adaptive pictorial rows) can substantially reduce interaction effort while maintaining usability across diverse users. Evaluation with ten participants showed consistently high perceived performance, low frustration, and manageable physical and mental demands across modalities, with head tracking emerging as the most effortful but still usable option. Overall, the prototype provides a coherent demonstration of how AI mediation can meaningfully enhance AAC systems for motor-impaired users.

3.5 Future Work

The current prototype demonstrates the feasibility of a unified multimodal AAC interface, but several directions remain open for strengthening robustness, accessibility, and long-term adaptability:

- **Fatigue-aware adaptation:** Develop lightweight models to detect fatigue from voice, head-pose stability, or blink patterns, enabling automatic modality shifts when effort increases.
- **Eye-tracking integration:** Add gaze-based input using WebGazer-style tracking or low-cost IR hardware to support users who cannot rely on head or voice interaction.
- **Stronger on-device models:** Upgrade Whisper and the LLM backend to quantised, on-device variants (e.g., Whisper Medium, Llama-based predictors) for better accuracy without cloud dependence.
- **Adaptive pictorial vocabulary:** Expand the symbol grid with richer categories and personalised icons generated or reorganised by the LLM.
- **Personalisation and learning:** Model user-specific preferences (dwell timing, key sizes, vocabulary) and adapt the interface over repeated sessions.
- **Hardware and clinical compatibility:** Integrate with real AAC switches and joysticks to evaluate ergonomics, latency, and suitability for diverse motor abilities.
- **Studies with target users:** Conduct longitudinal evaluations with motor-impaired students to assess learning curves, fatigue patterns, and real-world communication outcomes.

These directions outline a clear path from a functional high-fidelity prototype towards a deployable, adaptive multimodal AAC system capable of supporting a broad and diverse user base.

4 Ethics and Risk Assessment

This evaluation involved ten healthy volunteer participants from IIT Gandhinagar. All participants were verbally briefed about the purpose of the study, the nature of the prototype, and the fact that their names may appear in the report. Participation was entirely voluntary, and all individuals explicitly confirmed that they had no concerns regarding the use of their names or anonymised responses.

Since the study involved healthy participants and only software-based interaction, the physical and psychological risks were minimal. Potential discomforts such as temporary eye strain (during head-tracking) or mild fatigue (during repeated tasks) were communicated beforehand. No sensitive personal data were collected, and no recordings were stored. Participants could withdraw at any time without consequence.

Overall, the evaluation adhered to standard low-risk HCI usability testing practice, with informed consent, clear communication, and minimal exposure to discomfort.