

# **Project Concept Note (1M1B – IBM SkillsBuild AI + Sustainability Virtual Internship)**

## **Granite Accessible Assistant: An Inclusive AI Interface for Sustainable Cities and Public Services**

Student: Lohitaksha Patary

College: Amrita Vishwa Vidyapeetham, Bangalore

Email: bl.sc.u4cse24025@students.amrita.edu / patarylohitaksha06@gmail.com

GitHub: <https://github.com/lohitaksha06/granite-accessible-rag>

### **Introduction & Project Context**

Modern cities rely on self-service systems: kiosks at airports and hospitals, digital counters at banks, ticketing machines, and online government forms. These systems improve efficiency, but often assume users can see small text, follow complex menus, hear announcements, and ask staff confidently.

As a result, many people struggle to access essential services independently—especially people with disabilities (blind/low-vision, deaf/hard-of-hearing, cognitive/learning disabilities), elderly users, shy or socially anxious individuals, and non-native language speakers. In high-pressure places like airports, grocery stores, banks, hospitals, and government offices, the outcome is stress, longer queues, and dependence on staff.

This project treats sustainability as both environmental and social: inclusive systems reduce friction and make public services equitable, dignified, and usable for everyone.

### **SDG Alignment**

- Primary: SDG 11 – Sustainable Cities and Communities: inclusive access to public services and urban infrastructure.
- SDG 10 – Reduced Inequalities: reduces barriers based on disability, age, language, or confidence.
- SDG 4 – Quality Education: provides clear, step-by-step guidance that supports different learning needs.
- SDG 9 – Industry, Innovation, and Infrastructure: adds an accessibility layer to existing systems without full redesign.

### **Problem Statement**

How might we use AI to enable people of different abilities, languages, and confidence levels to independently access essential services in a sustainable and inclusive manner?

Current gaps:

- One-size-fits-all interfaces and instructions
- Over-reliance on visual or spoken interaction
- Limited personalization (language, simplified steps, calmer delivery)

- Stress and dependency on staff for routine guidance

## AI Solution Overview

Granite Accessible Assistant is an AI-powered accessibility layer (not just a chatbot) that provides grounded, user-adaptive help for public services.

It uses IBM Granite models for language understanding and response generation, and Retrieval-Augmented Generation (RAG) to pull verified, pre-approved information before answering. Responses adapt to a user's selected language and accessibility preference (for example: cognitive-friendly short steps, caption-first, or low-vision friendly structure).

This project does not train a new LLM; it responsibly applies existing models and retrieval of trusted content to improve reliability.

## How the System Works (Detailed Architecture)

### ### System Architecture Overview

```  
```  
```

### ### Step-by-Step Workflow

#### \*\*Step 1: User Interaction\*\*

The user approaches a kiosk, opens the mobile app, or accesses the web interface. They select their preferred language (English, Hindi, Spanish, French, German, Arabic, Mandarin, etc.) and accessibility mode (Blind/Low-vision, Deaf/Hard-of-hearing, Cognitive-friendly, or None).

#### \*\*Step 2: Query Submission\*\*

The user types or speaks their question—for example: "How do I check in for my flight?" or "Where is the nearest ATM?" The query is sent via HTTP POST to the FastAPI backend's `/ask` endpoint along with the selected disability type and language preference.

#### \*\*Step 3: Profile Resolution\*\*

The backend resolves the user's accessibility profile into specific instructions:

- \*\*Blind/Low-vision\*\*: "Provide step-by-step verbal instructions. Avoid visual-only references like colors or screen positions. Use clock-face directions (e.g., 'door at 3 o'clock')."
- \*\*Deaf/Hard-of-hearing\*\*: "Provide text-based instructions. Avoid audio-only references. Use visual cues and written directions."
- \*\*Cognitive-friendly\*\*: "Use simple, short sentences. Break tasks into small numbered steps. Avoid jargon and technical terms."

#### \*\*Step 4: Retrieval-Augmented Generation (RAG)\*\*

The RAG pipeline converts the user's query into a vector embedding using a SentenceTransformer model (all-MiniLM-L6-v2). This embedding is searched against a FAISS vector database containing pre-verified accessibility documents (airport navigation guides, bank procedures, grocery checkout steps, etc.). The top 2 most relevant document chunks are retrieved.

#### \*\*Step 5: Prompt Construction\*\*

The system builds a structured prompt that includes:

- Grounding rules (use ONLY retrieved context, say "I don't know" if context is insufficient)
- The user's disability-specific instructions
- The user's language preference
- Formatting requirements (short title, 3–6 numbered steps, short sentences)
- The retrieved context documents
- The user's original question

**\*\*Step 6: LLM Response Generation\*\***

The constructed prompt is sent to IBM Granite LLM (granite-4.0-micro), which generates a calm, structured, accessibility-aware response grounded in the retrieved information. The model does not hallucinate or add outside facts—it answers strictly from the provided context.

**\*\*Step 7: Output Delivery\*\***

The response is returned to the frontend and displayed in an accessible format:

- Clear text with numbered steps
- For blind users: optional Braille representation is shown alongside the text
- The interface uses high-contrast colors, large fonts, and screen-reader-friendly HTML

#### ### Key Technical Components

| Component      | Technology            | Purpose                            |
|----------------|-----------------------|------------------------------------|
| Frontend       | React + Vite          | Accessible kiosk-style UI          |
| Backend        | FastAPI (Python)      | RESTful API with CORS support      |
| Embeddings     | SentenceTransformers  | Convert text to vector embeddings  |
| Vector Store   | FAISS                 | Fast similarity search for RAG     |
| LLM            | IBM Granite 4.0 Micro | Grounded response generation       |
| Knowledge Base | Plain text documents  | Verified accessibility information |

#### ***Virtual Kiosk Interface (Screenshot)***

**Virtual Kiosk**

Granite Accessible RAG

|                                   |                                      |
|-----------------------------------|--------------------------------------|
| Disability                        | Language                             |
| <input type="text" value="None"/> | <input type="text" value="English"/> |

How can I help you today?

Type your question here

Ask Assistant

**Assistant Response**

Your answer will appear here.

## Target Users

Primary users:

- Blind or low-vision users
- Deaf or hard-of-hearing users
- Users with cognitive or learning disabilities

Secondary users:

- Elderly users
- Shy or socially anxious users
- Non-native language speakers
- General public in high-pressure environments (airports, hospitals, banks)

## Responsible AI Considerations

- Fairness: supports diverse needs via user-selected accessibility preferences.
- Transparency: indicates AI-assisted guidance and uses safe fallback when verified info is missing.
- Privacy: minimizes personal data; preferences can be used without identity details.
- Safety: reduces hallucinations by grounding responses in retrieved, curated content.

## Expected Impact

- Social: increased independence and dignity; reduced stress through calm, step-by-step guidance.
- Urban: faster service flow and more consistent instructions; reduced repetitive staff workload.

- Sustainability: scalable inclusive infrastructure that improves existing systems without major redesign.

## Metaverse & Future Scope

Here, “metaverse” means accessible virtual service interfaces (digital city-service spaces), not gaming and not VR-only.

### ### Upcoming Features (Roadmap)

#### \*\*1. Customizable AI Avatars\*\*

Users will be able to select from a variety of AI avatars that represent them during interactions:

- \*\*Diverse representation\*\*: Avatars of different genders, ethnicities, ages, and abilities
- \*\*Personalization\*\*: Users can customize avatar appearance, clothing, and accessories
- \*\*Emotional expressions\*\*: Avatars display appropriate emotions (calm, encouraging, attentive) to make interactions feel more human and supportive
- \*\*Accessibility indicators\*\*: Optional visual indicators showing the user's accessibility preferences

#### \*\*2. Interactive Sign Language AI Bot\*\*

A revolutionary feature where the AI assistant responds with animated sign language:

- \*\*Regional sign language support\*\*:
  - American Sign Language (ASL) – United States, Canada
  - British Sign Language (BSL) – United Kingdom
  - Indian Sign Language (ISL) – India
  - Japanese Sign Language (JSL/Nihon Shuwa) – Japan
  - Chinese Sign Language (CSL) – China
  - French Sign Language (LSF) – France, parts of Africa
  - German Sign Language (DGS) – Germany, Austria
  - Arabic Sign Language variants – Middle East, North Africa
  - Brazilian Sign Language (Libras) – Brazil
  - Korean Sign Language (KSL) – South Korea
- \*\*Real-time avatar signing\*\*: The AI avatar performs sign language in sync with text responses
- \*\*Bidirectional communication\*\*: Future integration with camera-based sign language recognition so deaf users can sign their questions
- \*\*Cultural adaptation\*\*: Signs and gestures are culturally appropriate for each region

#### \*\*3. Voice & Gesture Interaction\*\*

- \*\*Voice input\*\*: Speak questions naturally; speech-to-text converts to queries
- \*\*Voice output\*\*: Text-to-speech reads responses aloud for blind/low-vision users
- \*\*Gesture controls\*\*: Simple hand gestures to navigate, confirm, or repeat instructions
- \*\*Eye-tracking support\*\*: For users with motor disabilities, eye-gaze can control the interface

#### \*\*4. Multi-Modal Output Formats\*\*

- \*\*Braille display integration\*\*: Direct output to refreshable Braille displays

- **\*\*Haptic feedback\*\*:** Vibration patterns on mobile devices for navigation cues
- **\*\*Audio descriptions\*\*:** Detailed audio narration of visual elements
- **\*\*Simplified visual mode\*\*:** High-contrast, large-text, icon-based interface for cognitive accessibility

#### **\*\*5. Offline & Edge Deployment\*\***

- **\*\*Offline mode\*\*:** Core functionality works without internet using on-device models
- **\*\*Edge kiosks\*\*:** Deploy to standalone kiosks in airports, hospitals, banks with local processing
- **\*\*Low-bandwidth mode\*\*:** Compressed responses for areas with poor connectivity

#### **\*\*6. Learning & Adaptation\*\***

- **\*\*User preference memory\*\*:** Remember returning users' language and accessibility choices
- **\*\*Adaptive difficulty\*\*:** Automatically adjust explanation complexity based on user interactions
- **\*\*Feedback loop\*\*:** Users can rate responses to improve future answers

#### **### Long-Term Vision**

The ultimate goal is a universal accessibility layer that can be deployed across any public service system worldwide—creating truly inclusive smart cities where no one is left behind due to disability, language, age, or social anxiety.

## **Conclusion**

Granite Accessible Assistant focuses on practical sustainability by improving inclusion in public services. By responsibly using IBM Granite with RAG-grounded verified information, the solution is feasible, scalable, and human-centric—helping citizens access services independently while reducing stress and staff overload.