

AI-Learn: Comprehensive Testing Instructions for Judges (Hybrid Architecture)

This demonstration verifies the full functionality of the **AI-Learn Smart Assistant**, testing the instantaneous **on-device execution (Gemini Nano)** for core text tasks and the full capability of **Multimodal and Document AI** via the required locally running Flask backend.

1. Application Access and Setup

Requirement	Details
Extension Installation	Mandatory: Must be loaded via Load unpacked from the <code>Ai-Learn/extension</code> directory (as described in the GitHub README).
Backend Status	MANDATORY: You must run the Flask application locally on Port 5000 as detailed in the GitHub README's "Local Development Setup." The extension is configured to connect to <code>http://localhost:5000</code> for all remote services.
User Access	MANDATORY: Please click the Register button on the Side Panel's landing page and create a test account (e.g., <code>test@example.com</code> , password: <code>test</code>) to ensure the accessibility profile saving/loading functionality is active.
Local AI Status	CRUCIAL: To test the core technical function, keep Chrome flags enabled (as detailed in the README). If you encounter any issues with local AI functionality, simply disable the flags and reload Chrome to ensure Cloud Fallback is reliable.


2. Hybrid Features: Local-First Execution (Proofreader, Prompt, Summarize, Translate)

Goal: Verify that these text features use **Local AI for instant results** on short input, and gracefully **fall back to the local Flask server** for longer or more complex tasks (prompt length > 3000 characters).



Test A: Prompt and Proofreader (Local Execution Priority)

- Open any standard webpage (e.g., `https://www.wikipedia.org`).
- Proofreader Test (Local AI):**
 - Select** a short sentence (max 20 words) with a minor grammar error: e.g., "The cat chase the mouse very quick."
 - In the Side Panel, navigate to **Proofreader**. Click " **Check Grammar**".
 - Verification:** The corrected response appears **instantly (Local AI)** or quickly (Cloud fallback).

3. Prompt Test (Local AI):

- In the Side Panel, navigate to **Prompt API**.
- Enter a short, simple query (e.g., "What is the capital of France?").
- Click "  **Ask AI**".
- **Verification:** The answer returns **instantly (Local AI)** or quickly (Cloud fallback).


Test B: Summarize and Translate (Fallback Verification)



1. Open an extensive article. **Select a long passage (5000+ characters)** to guarantee the Cloud Fallback path is triggered.
2. **Summarize Test (Cloud Fallback):**
 - In the Side Panel, navigate to **Summarizer**. Select "**Medium**" length.
 - Click "  **Summarize**".
 - **Verification:** The summary is returned after a slight delay (latency confirms Cloud Fallback to the local Flask connection).
3. **Translate Test (Cloud Fallback):**
 - In the Side Panel, navigate to **Translator**. Select **Target Language: Spanish**.
 - Click "  **Translate**".
 - **Verification:** The Spanish translation is returned (latency confirms Cloud Fallback to the local Flask connection).

3. Cloud-Only Features (Local Flask Backend Required)

Goal: Verify functionality of all features that are designed to *only* run via the Cloud Gemini model due to complexity (Multimodal Vision, full webpage processing). **Network latency is expected for all tests below.**



Test C: Simplify and Simplify Web

1. **Simplify Text (Cloud-Only Execution):**
 - Open an extensive article. **Select a full paragraph (5+ sentences)** with complex language.
 - In the Side Panel, navigate to **Simplify**. Select "**Intermediate**" level.
 - Click "  **Simplify Text**".
 - **Verification:** The simplified text is returned. This task relies on the **Cloud Gemini model** routed through the local Flask backend.
2. **Simplify Web (Cloud-Only Execution - Webpage Replacement):**
 - Navigate to a complex article or news page (not Wikipedia).
 - In the Side Panel, navigate to **Simplify Web**. Select "**Intermediate**" level.



- Click " **Simplify This Page**".
- **Verification:** The current page content is **replaced** with the simplified, AI-rewritten version. Click " **Restore Original Page**" to confirm the original page returns.

Test D: Multimodal and Document AI

1. Screenshot Analysis (Cloud-Only Execution):

- Navigate to a page containing a complex chart or graph.
- In the Side Panel, navigate to **Screenshot AI**. Click " **Capture Screenshot**".
- In the query box, enter: Analyze the main trend shown in the chart.
- Click " **Analyze with AI**".
- **Verification:** The system returns an analysis of the visual content, confirming the use of **Gemini Vision** via the Flask backend.


2. OCR Translate (Cloud-Only Execution):

- In the Side Panel, navigate to **OCR Translate**.
- Click " **Capture Screenshot**" and capture an image containing text.
- Click " **Extract & Translate**".
- **Verification:** The extracted text and translation are returned.

3. Insights (Cloud-Only Execution - Analytics):

- In the Side Panel, navigate to **Insights**.
- Click "**Load Insights**" (or wait for automatic loading).
- **Verification:** The panel displays "**Sessions Analyzed**" and a text analysis of feature usage patterns generated by the Flask backend's AI route.

4. Document AI Assistant (Cloud-Only Execution):

- In the Side Panel, navigate to **Document AI Assistant**.
- Click "**Click to upload document**" and upload a **PDF** or **DOCX** file.
- Select the "**Both**" option. Click " **Process with AI**".
- **Verification:** The result container displays both a **SUMMARY** and **PROOFREAD RESULTS**.

4. Accessibility and UX Features (Client-Side)







Goal: Verify the application of dynamic CSS, user profile persistence, and client-side web APIs (TTS/Storage).

Test E: Profile Saving and Adaptive Styles

1. In the Side Panel, ensure **Accessibility Mode** is ON.
2. Click the **ADHD** profile button.

- **Verification 1:** The message "Profile saved..." appears.
3. **Apply Styles:** Ensure you are viewing a typical article.
 - **Verification 2 (ADHD Focus):** Observe the webpage. Distracting elements (sidebars, ads, footers) should be visibly **dimmed**. The **Reading Line** should follow the cursor's vertical position.
 4. Click the **Dyslexia** profile button.
 - **Verification 3:** The webpage font changes to a dyslexic-friendly font (e.g., 'OpenDyslexic', 'Comic Sans MS').
 5. Close and re-open the Side Panel.
 - **Verification 4:** The **Dyslexia** button remains selected, confirming profile persistence via Chrome Storage.

Test F: Voice Reader (TTS)

1. Select a paragraph of text on the webpage.
2. Open the Side Panel and navigate to **Voice Reader**.
3. Click " **Start Reading**".
 - **Verification:** The selected text is read aloud by the system TTS engine. The controls (" **Start Reading**", " **Pause**" / " **Resume**", and " **Stop**") function correctly.
4. **Response Voice Reader Check (Quick Test):**
 - Run the **Proofreader** test (Test A.2).
 - In the response container, click the " " (**Speaker icon**) button.
 - **Verification:** The corrected text is read aloud. The button cycles correctly through the