FAQs

1. What dataset (corpus & size) are you using to evaluate the tool?

The tool has been formatively evaluated on the MOSIP-OSU evaluation data, which in total comprised 20 screenshots over 13 use cases (for the Abi persona) and a resultant set of **122 annotated bugs**. Despite the relatively small size, it was able to achieve an impressive **84.2% recall in a zero-shot setting (unsupervised)**, meaning it was able to flag 84% of inclusivity bugs that the humans found in these sessions, without having seen any labeled examples or undergoing training.

We further **encourage you to use the tool internally**, record bugs it catches, misses, or misclassifies, and note any/all corresponding design fixes. This feedback loop helps us expand and refine the dataset, improving the reasoning model's performance over time.

---

2. Is the tool supervised or unsupervised?

The bug detection process is primarily **unsupervised**, as the model autonomously identifies inclusivity bugs/issues in the UI. However, the **decision rules** and **grounding mechanisms** (such as tying it to facets) make it **semi-supervised**. This ensures the model does not rely on vague heuristics or develop biases during inference/issue identification. The hybrid approach gives you the benefits of AI-driven exploration while maintaining foundational research-backed evaluation.

---

3. How many images can we upload into the tool?

There is **no hard-coded limit** on the number of images you can upload. Each screenshot is processed **asynchronously**, and the software scales based on your server capacity and available compute resources.

However, if your product workflows involve **interdependent UI states** (e.g., multi-step forms or modals triggered by previous actions), we may introduce **batch size constraints** or **ordering logic** in future updates to preserve performance and analytical accuracy.

---

4. Does the tool work for websites in languages other than English?

The tool could be modified to support **multi-lingual interfaces**, using the multi-lingual capabilities of reasoning models. This means it can process and understand content in a variety of languages.

However, a few important notes:

- For text-heavy analysis, **token usage and API costs** may **increase** due to **language translation overhead**. **Model performance** might also be affected. However, we expect things to get better and cheaper with the rapid evolution of these model capabilities.

- For **vision tasks**, such as analyzing UI structure, comprehending image components, language is not a barrier. These models can interpret textual content in screenshots regardless of language, making them highly scalable.

---

5. Can we add custom personas or decision rules?

Yes, both are supported via a **modular configuration**:

- **Decision rules** can be updated through the Decision Rules.csv file for the corresponding persona on the server side. Client-side toggles could be incorporated in future versions if needed.

- **Personas**: Can be added as the server supports extensibility to different personas. However, we advise strong caution against adding personas without backing them with foundational research.
    - LLMs can **amplify demographic or cultural biases**, and applying poorly designed personas can lead to **biased assessments** that could **negatively** AND **dramatically impact** your product/UX.
    - We can expand the tool in the future to consider personas for Socio-economic status, age, and neurodiversity, for which preliminary foundational research exists.

---

6. What is the value proposition of this tool? Does the tool decide what I should fix for the highest ROI?

The tool **automatically flags cognitive inclusivity bugs/violations** in software and will soon suggest **potential design fixes**. While **triaging** based on product constraints still requires human input, the tool **automates bug identification**, typically the most time-consuming part of UI evaluation, improving developer productivity. The result is **faster evaluations, iterations,** and **better UX**.

7. Can I switch between different LLMs (models)?

Yes. The tool is designed to be **model-agnostic** and is fully compatible with any LLM available on **Amazon Bedrock**, including **Claude**, **Mistral**, **DeepSeek**, and others—**no code modifications are required** for switching between these.

To switch models within Bedrock, follow these two steps:

1. **Add the model's inference token ID** to the list of supported models on the server side.

2. **Update the model invocation logic** on the server side to call the desired model at runtime.

If you prefer to use **OpenAI's GPT models**, you'll need **OpenAI credits/API** and will need to follow a **switch documented in the tool's README**, as AWS Bedrock doesn't support GPT models in it yet.

---

8. What are the technical requirements to run the tool?

To deploy and use the tool, you'll need the following stack:

- **Frontend**: JavaScript (React)
- **Backend**: Python server (Flask)
- **Model access**: AWS Bedrock or OpenAI token/API keys for inference

All dependencies and setup steps are detailed in the **README** file.

---

9. Are the flagged UI bugs meant for novice or expert users of my product?

The flagged violations cover issues that impact **both first-time and returning users**. The system, at its core, evaluates UI elements based on:

- **(UX-goal) Learnability** is a core usability dimension: the system identifies issues that hinder first-time and returning users (e.g., unclear labels, missing guidance).
- For **expert users**, it flags friction points that affect efficiency, discoverability, or consistency.

In short, the bugs flagged aim to improve the usability experience for **all user types**, ensuring both **ease of entry** and **efficiency of use**.