# Generative Interfaces for Language Models

Anonymous ACL submission

## Abstract

Large language models (LLMs) are increasingly seen as assistants, copilots, and consultants, capable of supporting a wide range of tasks through natural conversation. However, most systems remain constrained by a linear request-response format that often makes interactions inefficient in multi-turn, information-dense, and exploratory tasks. To address these limitations, we propose **Generative Interfaces** for Language Models, a paradigm in which LLMs respond to user queries by proactively generating user interfaces (UIs) that enable more adaptive and interactive engagement. Our framework leverages structured interface-specific representations and iterative refinements to translate user queries into task-specific UIs. For systematic evaluation, we introduce a multidimensional assessment framework that compares generative interfaces with traditional chat-based ones across diverse tasks, interaction patterns, and query types, capturing functional, interactive, and emotional aspects of user experience. Results show that generative interfaces consistently outperform conversational ones, with up to a 72% improvement in human preference. These findings clarify when and why users favor generative interfaces, paving the way for future advancements in human-AI interaction.

## 1 Instruction

A longstanding goal in computing is to design systems that not only respond to users but also adapt by dynamically reshaping interfaces to facilitate users' interaction and help them achieve their goals (Apple Inc., 1987; Lyytinen and Yoo, 2002). While recent advances in large language models (LLMs) have brought us closer to this vision by enabling flexible natural language understanding, the dominant interaction paradigm, which we call the conversational UI, remains static and linear: most LLM outputs are still rendered as long blocks of text, regardless of task complexity or user preference, limiting the model's ability to support the diverse ways users seek to learn, explore, and interact. At the same time, state-of-the-art LLMs have shown remarkable capabilities in automatically generating high-quality, functional webpages from designs and natural language descriptions (Si et al., 2024; Li et al., 2024; Xiao et al., 2024). Together, these developments raise an exciting research question: *How can LLMs go beyond conversational interfaces to enable adaptive, goal-driven interactions that meaningfully serve human needs?*

In this work, we introduce **Generative Interfaces**, a new paradigm that differs from conversational UIs. Rather than delivering static text responses within a predefined chatbot window, Generative Interfaces dynamically create entirely new interface structures that adapt to users' specific goals. While recent tools like OpenAI's Canvas (OpenAI, 2024b) and Claude's Artifacts (Anthropic, 2024) enhance user interaction by providing dedicated workspaces for documents, code, and visualizations, our approach extends this vision by seamlessly supporting deeper engagement and enabling richer, task-specific experiences. For example, as shown in Figure 1, when users pose questions such as "*I want to understand neural networks*", conversational interfaces typically return long blocks of text. In contrast, Generative Interfaces transform these queries into an interactive neural network animation or a piano practice tool that offers real-time feedback.

This paradigm shift presents two key challenges: (1) building the infrastructure to generate user interfaces on the fly in response to users' queries, and (2) rigorously evaluating whether such generated interfaces actually improve user experience. To address the first challenge, our framework introduces a structured interface-specific representation coupled with an iterative refinement procedure. The structured representation enables more controllable

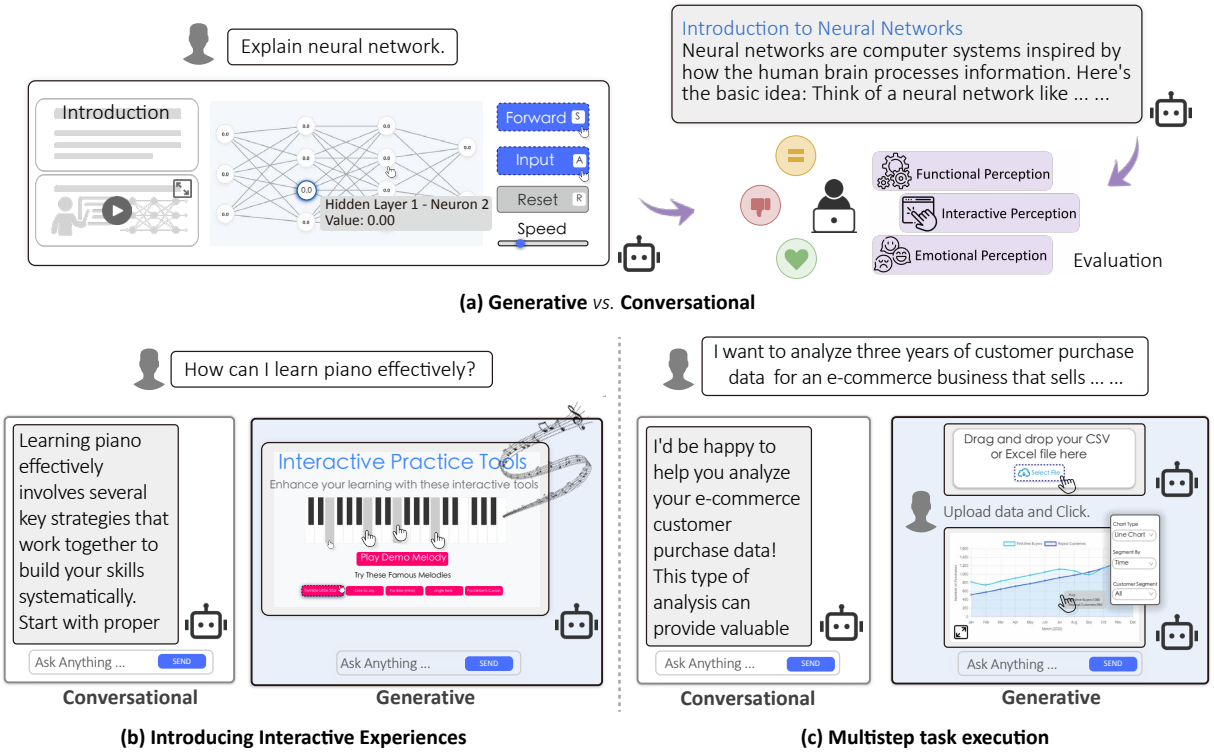



Figure 1: **Generative Interfaces compared to conversational interfaces.** (a) Conceptual framework showing how Generative Interfaces create structured, interactive experiences rather than static text responses, evaluated along functional, interactive, and emotional dimensions. (b–c) Example queries illustrate how Generative Interfaces transform user input into adaptive tools—such as interactive learning aids or multistep workflows—providing clearer organization and richer interactivity than conversational responses.

and interpretable generation by explicitly modeling high-level interaction flows, interface state transitions, and component dependencies, which we formalize using finite state machines (Shehady and Siewiorek, 1997; Wagner et al., 2006). The iterative refinement procedure further enhances output quality by prompting LLMs to generate query-specific evaluation rubrics and repeatedly refine interface candidates through generation-evaluation cycles until the system converges on a polished, context-appropriate solution. To address the second challenge, we establish a systematic evaluation framework for assessing language model interfaces across three key dimensions: functionality, interactivity, and emotional perception (Hartmann et al., 2008; Nielsen et al., 2012; Duan, 2025). Specifically, we construct a diverse prompt suite, ***U***ser ***I***nterface e***X***perience (UIX), that strategically covers diverse domains and prompt types to reflect real-world usage scenarios (Tamkin et al., 2024). For each user query, we recruit experienced annotators to interact with different interfaces and conduct pairwise comparisons. Beyond this fixed prompt suite, we further conduct a complementary evaluation involving real users and their self-reported queries, which substantiates the advantages of generative interfaces under more open-ended and authentic usage conditions. We also reveal *when* generative interfaces excel (in structured and information-dense domains) and *why* users prefer them (through enhanced visual organization, interactivity, and reduced cognitive load).

## 2 Generative Interface for LMs

We introduce the structured interface-specific representation (Sec . 2.1), outline the generation pipeline (Sec . 2.2), and finally describe the iterative refinement using adaptive reward functions (Sec . 2.3).

### 2.1 Structured Interface-specific Representation

Directly generating interfaces is challenging due to the vast search space and the complexity of interactive contexts. To address this, we translate user queries into a **structured interface-specific representation** that anchors and guides the generation process. This representation operates at two complementary levels: **(I)** *high-level* interaction

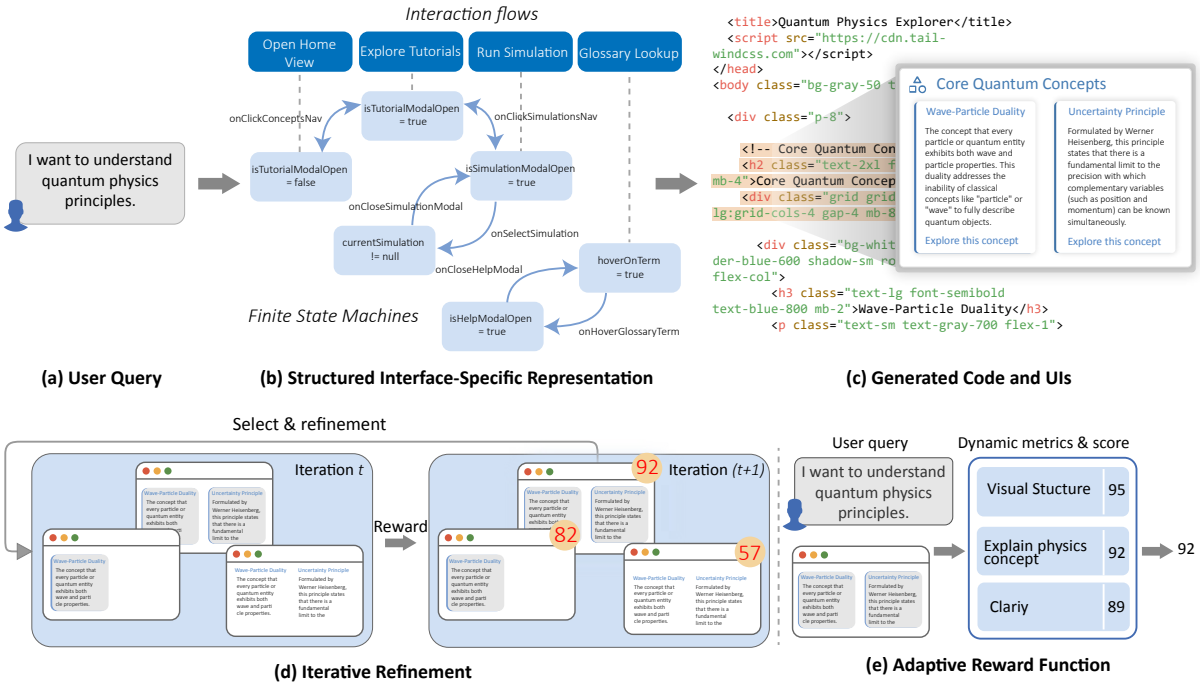



Figure 2: **Generative Interfaces infrastructure**: (a) User queries are first converted into (b) structured interface-specific representations that model interaction flows and component dependencies. This structured representation guides the generation of (c) functional code and user interfaces. The system employs (d) iterative refinement with (e) adaptive reward functions containing query-specific evaluation rubrics.

flows that capture user trajectories and task phases, and **(II)** *low-level* finite state machines (FSMs) that define component behaviors and UI logic.

**Interaction flows** The high-level interaction flow provides a symbolic abstraction of user behavior across primary interface stages. It represents user task progression as a directed graph, where transitions are triggered by UI events such as clicking. We denote this abstraction as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{T})$, where nodes $\mathcal{V}$ represent interface views or subgoals, and edges $\mathcal{T}$ denote possible transitions (See Appendix E for detail definition). In the example shown in Figure 2, the natural language query *I want to understand quantum physics principles"* is grounded into a coherent interaction trajectory: `Open Home View` →`Explore Tutorials`→`Run Simulation`→ `Glossary Lookup`". This abstraction captures the high-level intent and interaction logic of potential users, while concrete UI behaviors (e.g., state toggles and modal updates) are specified separately in the FSM.

**Finite state machines** We further use Finite State Machines (FSMs) to describe how individual UI modules respond to user actions and update their states accordingly. Formally, we model each UI component as $\mathcal{M} = (\mathcal{S}, \mathcal{E}, \delta, s_0)$, where $\mathcal{S}$ is the set of atomic interface states (*e.g.*, `isModalOpen=true`), $\mathcal{E}$ is the set of user-triggered events (*e.g.*, click, hover), $\delta$ is the state transition function, and $s_0$ is the initial state (See Appendix E). This structure explicitly defines how the interface should behave given a particular state and a triggered event.

## 2.2 Generation pipeline

**(I) Requirement specification:** First, we generate a requirement specification for the user query, capturing the main goal, desired features, UI components, interaction styles, and problem-solving strategies. This specification serves as a bridge between the user's natural language intent and formal interface design. **(II) Structured representation generation:** Second, we generate a structured interface-specific representation (Sec. 2.1) based on the requirement specification. This representation serves as a modular and interpretable scaffold for UI generation, where the hierarchy of interaction flows and finite state machines ensures that the resulting interfaces are both coherent and functional. **(III) UI generation:** To support the UI generation based on structured representations, we build a complementary codebase containing reusable im-

plementations of common UI elements (*e.g.*, clock, map, calculator, video player, code viewer, and chart). Additionally, a web retrieval module[1] gathers relevant UI examples and data sources. Finally, the entire context, including the natural language query, requirement specification, structured representation, predefined components, and retrieved content examples, is passed to an LLM to synthesize executable HTML/CSS/JS code, which is rendered into an interface, as illustrated in Figure 2.

### 2.3 Iterative UI Refinement

Generating an effective and well-structured user interface is usually an iterative process (Li et al., 2024). To this end, we introduce an adaptive, reward-driven iterative refinement procedure that progressively improves UI quality by generating evaluation metrics, scoring candidates, and regenerating interfaces through multiple cycles. **(I) Adaptive reward function:** To support task-specific and context-aware evaluation, we employ an LLM to construct a reward function tailored to each user query adaptively. As shown in Figure 2(e), for query *"I want to understand quantum physics principles,"* the system automatically generates a set of fine-grained evaluation metrics—such as *Visual Structure*, *Explain Physics Concept*, and *Clarity*—each with associated weights and verification rules. These dimensions are scored independently and aggregated to compute the final overall score, which ranges from 0 to 100. See Appendix F for examples of adaptive reward functions. **(II) Iterative refinement:** As depicted in Figure 2(d), at each iteration, multiple UI candidates are generated, then the adaptive reward function evaluates these candidates. In the next iteration, we will regenerate the UI using the highest-scoring candidate from the previous iteration, along with its evaluation. This feedback loop guides the LLM to address issues related to structure, semantics, or visual design. The process continues until a candidate reaches an overall score of 90 or higher, or until we have completed five iterations.

## 3 Evaluation Framework

To enable systematic evaluation, we developed a comprehensive evaluation framework, which includes a diverse user query suite named ***U***ser ***I***nterface e***X***perience (UIX), covering various scenarios, styles, and intents (Sec. 3.1); a set of multi-dimensional evaluation metrics (Sec. 3.2); and an integrated human study (Sec. 3.3).

### 3.1 User Queries

In UIX, we generate a test set of 100 user queries using Claude 3.7 that spans multiple domains, supporting varying specificity levels and capturing different query complexities. Specifically, we follow best practices from prior work around how people engage with LLMs as follows. **(I) Topic coverage**: Prompts are uniformly distributed across the ten domains defined in Clio (Tamkin et al., 2024), covering a wide range of real-world user scenarios. See Appendix B for category list. **(II) Query detail level**: Following Cao et al. (2025), each domain contains an equal split of concise and detailed prompts. Concise prompts express intent abstractly in fewer than 15 words (*e.g.*, "*Create a SWOT analysis for my small business*"), while detailed prompts provide explicit goals and rich context. **(III) Query Type**: As user queries shift from casual dialogue to actionable tasks, our design maintains a balanced mixture between general conversational prompts (*e.g.*, "*How can I improve my public speaking?*") and interactive, task-oriented queries (*e.g.*, "*I want to visualize my company's sales data*").

### 3.2 Evaluation Metrics

We evaluate interface quality across three perceptual dimensions adapted from Nielsen et al. (2012) and Hartmann et al. (2008). (I) *Functional Perception* measures Query–Interface Consistency (**QIC**), which is the alignment between the interface and user intent, and Task Efficiency (**TaskEff**), which is the effectiveness of users in accomplishing goals (Duan, 2025). (II) *Interactive Perception* assesses **Usability**, **Learnability**, and Information Clarity (**IC**) (Nielsen et al., 2012; Hartmann et al., 2008; Cao et al., 2025). (III) *Emotional Perception* covers Aesthetic or Stylistic Appeal (**ASA**) and Interaction Experience Satisfaction (**IES**) (Duan et al., 2024; Duan, 2025). Following Zheng et al. (2023); Si et al. (2024), for each query, we ask annotators to choose between two interfaces across all seven metrics and overall.

### 3.3 Human Evaluation

For pairwise comparison, we collected human judgments on Prolific [2] (Annotators are paid at the rate

[1] We use `exa.ai` as the search API.

[2] `https://app.prolific.com`

of $16/hour, see Appendix I for annotator demographics). Each evaluation instance consisted of a user query and two UI outputs (Example 1 and Example 2) generated by different methods. Annotators judged which output better satisfied the criteria across seven evaluation dimensions and overall (Example 1 wins, Example 2 wins, or Tie). We aggregated the three judgments per instance using majority voting to obtain a single final decision. Despite the inherent subjectivity of interface assessments, Fleiss' Kappa (Landis and Koch, 1977) reached 0.525, indicating moderate agreement. Besides human evaluation, we show the results of automatic evaluation in Appendix D.

## 4 Experimental Results

**Implementation details** Our system is built on OpenCanvas[3] and uses Claude 3.7 (Anthropic, 2025) as the default backbone LLM, given its strong performance in UI code generation (Si et al., 2024; Li et al., 2024). We refer to our approach as **GenUI** and compare it against two baselines: **(I)** ***Conversational UI (ConvUI):*** A traditional chat interface using either GPT-4o (OpenAI, 2024a) or Claude 3.7 (Anthropic, 2025). To reduce potential bias in human evaluation, we present a unified chat interface without disclosing the underlying model. For Claude 3.7, 26% of responses include artifact generation. We remove the artifacts and retain only the textual output to ensure a clean and fair comparison with other conversational systems. **(II)** ***Instructed UI (IUI):*** An interface generated by Claude 3.7 when explicitly prompted (query + "*Please help me solve it with UI*"). This prompt consistently triggers artifact generation, and the resulting artifact is taken as the system output.

### 4.1 Main Results and Findings

**Conversational *vs.* Generative Interfaces** As shown in Table 1, GenUI consistently outperforms ConvUI across all evaluation dimensions. Interestingly, ConvUI (GPT-4o) performs more competitively than ConvUI (Claude 3.7), suggesting that well-structured textual responses can still be effective in specific scenarios. Compared to ConvUI (Claude 3.7), GenUI achieves the most significant gains in ASA (+86.0%) and IES (+81.0%). Overall, its emotional appeal and interactive functionality are the primary drivers of its superior performance, resulting in an 84.0% win rate over ConvUI (Claude 3.7). These findings suggest that users clearly prefer GenUI for most queries.

User comments further support this finding. For example, one noted: "*GenUI provides the requested information in an **easy-to-understand manner**, laying out everything requested and anticipating what else may be needed.*" A small number of users did express a preference for the familiarity of traditional ConvUIs, as one remarked (see interface examples in Appendix Figure 5): "*Chatbot interface is **most people know already**, while GenUI is a somewhat complex and **unfamiliar** app.*" This suggests that some users remain attached to familiar formats due to habits or ease. However, such a preference did not override the broader recognition of GenUI's objective advantages, indicating strong potential for user adaptation and adoption in real-world deployments.

**Domain Analysis** As shown in Table 2, preferences for GenUI vary by domain. Users strongly favored GenUI in *Data Analysis & Visualization* (93.8%) and *Business Strategy & Operations* (87.5%), where tasks typically involve interpreting large amounts of structured information. By contrast, in *Advanced AI/ML Applications*, GenUI received 50.0% of preferences, suggesting that traditional linear text explanations remain effective in math-heavy contexts. Overall, these results indicate that domains characterized by complex information benefit most from GenUIs, whereas ConvUIs are still suitable for domains that rely on straightforward explanations.

**Query Analysis** As shown in Figure 3a, GenUI receives stronger preferences for certain query characteristics. It is particularly favored in interactive tasks (80.0%), underscoring the advantages of generative interfaces in scenarios where interaction is essential for task completion. In general conversations, users also show a clear preference for GenUI over ConvUI (73.0% *vs.* 23.0%). When comparing query detail level, GenUI is preferred more for detailed queries (80.0%) than for concise ones (73.0%), likely because simple conversational responses sometimes sufficiently address short queries, whereas GenUI may introduce unnecessary complexity.

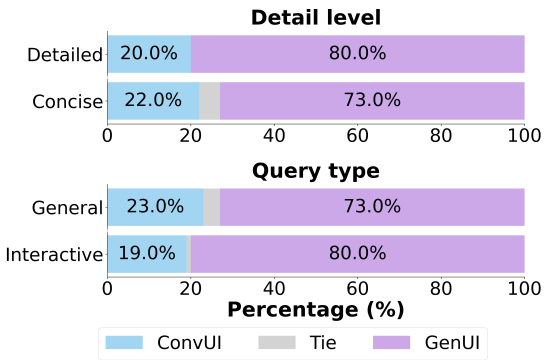
### 4.2 Ablation Study

**(I) Our Pipeline *vs.* Direct Instruct:** We compare our framework against IUI: directly instructing Claude 3.7 to generate a web interface with the artifact feature enabled, representing a highly

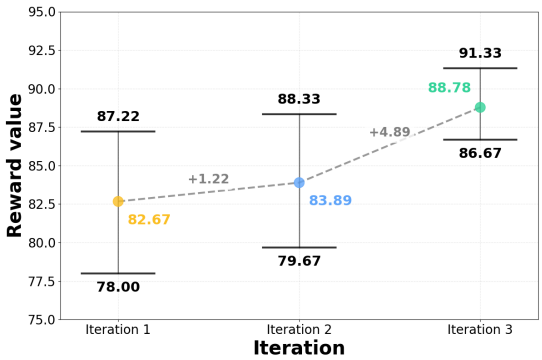
[3] https://github.com/langchain-ai/open-canvas

| Framework | Status | *Functional* | | *Interactive* | | | *Emotional* | | Overall |
|---|---|---|---|---|---|---|---|---|---|
| | | QIC | TaskEff | Usability | Learnability | IC | ASA | IES | |
| ConvUI (Claude 3.7) *vs.* GenUI | ConvUI | 11% | 14% | 13% | 10% | 9% | 3% | 6% | 12% |
| | Tie | 6% | 5% | 4% | 6% | 6% | 8% | 7% | 4% |
| | GenUI | 83% | 81% | 83% | 84% | 85% | 89% | 87% | 84% |
| ConvUI (GPT-4o) *vs.* GenUI | ConvUI | 32% | 41% | 28% | 35% | 38% | 13% | 24% | 30% |
| | Tie | 11% | 5% | 7% | 10% | 8% | 7% | 6% | 1% |
| | GenUI | 57% | 54% | 65% | 55% | 54% | 80% | 70% | 69% |
| IUI *vs.* GenUI | IUI | 13% | 17% | 16% | 14% | 16% | 20% | 14% | 17% |
| | Tie | 18% | 13% | 18% | 20% | 15% | 5% | 15% | 8% |
| | GenUI | 69% | 70% | 66% | 66% | 69% | 75% | 71% | 75% |

Table 1: **Human Evaluation of UI Framework.** Win, tie, and loss percentages of UI variants compared to our system (GenUI) across different perception dimensions: functional, interactive, and emotional.




(a) **Breakdown of query detail level and type.**




(b) **Ablation on number of iterations.**

Figure 3: Human evaluation results comparing GenUIs and ConvUIs. (a) User preference breakdown by query type and detail level. (b) Performance improvement across iterative interactions.

| Domain | GenUI(%) |
|---|---|
| Data Analysis & Visualization | 93.8 |
| Language Translation | 87.5 |
| Business Strategy & Operations | 87.5 |
| Education & Career Development | 83.3 |
| Academic Research & Writing | 79.2 |
| Content Creation & Communication | 75.0 |
| Digital Marketing & SEO | 75.0 |
| DevOps & Cloud Infrastructure | 75.0 |
| Web & Mobile App Development | 70.8 |
| Advanced AI/ML Applications | 50.0 |

Table 2: The ratio of GenUI being preferred across 10 query topics (Tamkin et al., 2024).

engineered baseline. Our system outperforms this strong baseline, achieving a 58.0% higher win rate (Table 1). Among the baselines, IUI shows better performance in emotional perception dimensions such as ASA, but it still lags behind GenUI overall. **(II) Natural Language *vs.* Structured Representation:** The natural language version provides a descriptive explanation of the UI based on the user query, without employing structured representations to define interface states formally. As shown in Table 3 (Row 1 *vs.* Row 2), structured representations outperform natural language, improving the win rate from 13% to 17% overall. **(III) One-shot Generation *vs.* Iterative Refinement:** As shown in Table 3 (Row 2 *vs.* Row 3), the iterative refinement process yields consistent improvements on human preference across all perception dimensions, resulting in a notable +14.0% overall win rate improvement compared to one-shot generation. Figure 3b further illustrates that each refinement round leads to a clear performance boost, with average LLM-based reward scores increasing by +1.2% and +4.9%, respectively. We illustrate an example of such iterative improvement in Appendix Figure 6, where each iteration incrementally enhances layout efficiency, usability, and user guidance, ultimately leading to a more informative and user-friendly interface through structured refinement. **(IV) Static *vs.* Adaptive Reward Function:** Table 3 (Row 3) highlights the effect of dynamic reward functions,

| Reward design | Generation paradigm | Representation | Status | *Functional* | | *Interactive* | | | *Emotional* | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | QIC | TaskEff | Usability | Learnability | IC | ASA | IES | |
| *Full GenUI: Adaptive, Iterative, Structured* | | | | | | | | | | | |
| Static | One-shot | Natural | Win | 8% | 16% | 11% | 19% | 15% | 10% | 13% | 13% |
| | | | Tie | 20% | 20% | 22% | 16% | 15% | 13% | 15% | 5% |
| | | | Loss | 72% | 64% | 67% | 65% | 70% | 77% | 72% | 82% |
| Static | One-shot | Structured | Win | 11% | 18% | 18% | 18% | 16% | 15% | 14% | 17% |
| | | | Tie | 20% | 12% | 12% | 15% | 10% | 10% | 13% | 5% |
| | | | Loss | 69% | 70% | 70% | 67% | 74% | 75% | 73% | 78% |
| Static | Iterative | Structured | Win | 28% | 30% | 30% | 27% | 27% | 34% | 27% | 31% |
| | | | Tie | 32% | 26% | 24% | 30% | 27% | 17% | 28% | 15% |
| | | | Loss | 40% | 44% | 46% | 43% | 46% | 49% | 45% | 54% |

Table 3: **Ablation study.** The control group is the full GenUI framework (*adaptive* reward, *iterative* generation, and *structured* representation). All ablations are compared against this full version, where "Loss" indicates that GenUI outperforms the variant. Note that "*Static*" refers to static reward design, "*One-shot*" denotes generation without refinement, and "*Natural*" indicates natural language representations.

| Concept | GenUI (%) |
|---|---|
| Visual Aesthetics & Engagement (23.4%) | 83.3 |
| Information Organization & Accessibility (14.9%) | 87.4 |
| Cognitive Load & Intuition (14.5%) | 78.5 |
| Actionability & Practical Utility (10.7%) | 81.8 |
| Information Richness & Comprehensiveness (10.4%) | 82.5 |
| Guidance & Learning Support (9.7%) | 74.5 |
| Content Relevance & Efficiency (6.8%) | 59.0 |
| Interactive Experience Quality (5.5%) | 89.7 |
| Perceived Credibility & Professionalism (3.3%) | 86.5 |
| Others (1.0%) | 71.9 |

Table 4: **Concept distribution.** We show the distribution of high-level concepts extracted from user comments using the pipeline described in Sec. 4.3. For each concept, we show the ratio of GenUIs being preferred.

which differ from the full version only by replacing adaptive scoring with a static baseline. The absence of dynamic rewards results in a 17.0% drop in overall win rate, with performance declining across all seven evaluation metrics. This comparison highlights the importance of dynamically adjusting evaluation criteria to capture the task-specific requirements inherent in each query instead of generic, fixed heuristics.

### 4.3 Human Preference Analysis

To understand the factors underlying human preferences, we collected textual justifications for 40% of pairwise comparisons and overall comments for the remaining 60%. Following Lam et al. (2024), we used Claude 3.7 to extract high-level semantic concepts from these qualitative responses and clustered them into themes (Table 4). This analysis reveals the main cognitive and perceptual drivers behind user choices beyond surface-level aesthetics. **(I) Cognitive Offloading as a Key Driver:** Cognitive offloading (Risko and Gilbert, 2016) emerged as a central reason for preferring GenUI. Among users mentioning *Cognitive Load & Intuition*, 78.5% favored GenUI, describing it as helping to "*break down categories into manageable steps*." GenUI thus acts as an external cognitive aid that simplifies complex information, while ConvUI remains preferred for straightforward, "how-to" queries where added structure may be unnecessary. **(II) Visual Structure Enhances Usability and Trust:** Within the "*Perceived Credibility & Professionalism*" theme, 86.5% of users preferred GenUI, citing its more organized, professional presentation—"*GenUI is more professionally written*," "*It offers sound advice*." Even when both interfaces conveyed similar content, users perceived GenUI as more trustworthy due to its modular layout, visual hierarchy, and polished formatting.

### 4.4 Real-User Query Evaluation

To further test generalizability, we conducted an additional study using real-user queries, where participants first provided five of their typical LLM queries (Bassignana et al., 2025) and then compared ConvUI and GenUI on these self-authored tasks (see Appendix K for more details). The collected queries encompass a diverse range of organically occurring user needs, from informational and planning tasks (e.g., travel, shopping, education) to personal, emotional, and creative requests. More importantly, this setup better captures authentic usage contexts and reduces query–annotator mismatch. Across 380 queries from 76 participants,

GenUI demonstrated a clear advantage (50.8% win, 8.2% tie, 41.1% loss) compared to ConvUI. Among all participants, 30.3% of them strongly preferred GenUI (in $\geq$80% of cases) even though this is their first time experiencing it, whereas only 18.4% strongly preferred ConvUI. The remaining users exhibited more query-dependent choices. Notably, for underrepresented domains in the UIX benchmark, such as personal and emotional tasks, GenUI still stands out, as users praised "*its visual appeal, interactivity, and personalized, tool-like experience that felt more engaging and immersive*". Overall, this study reinforces our main findings.

## 5 Related work

**Context-Aware and Adaptive Interface** Context-aware interfaces have been widely explored since the rise of ubiquitous computing, aiming to improve usability, reduce cognitive load, and better support user goals (Dey et al., 2000; Horvitz, 1999; Theng and Duh, 2008). As computing systems have become more complex and pervasive, the ability to adjust interfaces dynamically has been critical for creating more effective and accessible user experiences (Gajos and Weld, 2004; Gajos et al., 2007; Nichols et al., 2002, 2006a,b). Prior systems often adapted functionality through a finite set of predefined states. While effective in constrained settings, these approaches faced challenges with scalability and sometimes reduced predictability and user control (Findlater and Gajos, 2009). Recent advances in LLMs have enabled new forms of adaptive interfaces that dynamically generate interface elements in response to user prompts (Wu et al., 2022; Dibia, 2023; Cha et al., 2024; Cheng et al., 2024; Nandy et al., 2024), marking a shift from static outputs toward model-driven, interactive systems.

To improve interaction efficiency between humans and LLMs, prior studies (Jiang et al., 2023; Ma et al., 2024; Ross et al., 2023) have proposed combining text-based ConvUIs with Graphical User Interfaces (GUIs). For example, OpenAI Canvas enables users to directly edit documents and code on a canvas, avoiding repeated prompt inputs; Graphologue (Jiang et al., 2023) transforms lengthy and complex LLM responses into graphical diagrams to support information exploration and question answering. However, although these approaches leverage LLMs to generate displayed content, the UIs they employ are predesigned. In contrast, GenerativeGUI (Hojo et al., 2025) explores the usability of dynamically generated interfaces in clarifying question (CQ) interactions. ClarifyGPT (Mu et al., 2023) also introduces CQs, but in the narrower domain of code generation. Beyond CQ scenarios, DynaVis (Vaithilingam et al., 2024) proposes a system that combines natural language with dynamically synthesized UI widgets to support chart editing tasks, without exploring broader, general-purpose scenarios. Unlike prior systems that modify fixed UI components (Cao et al., 2025), our framework generates complete interfaces customized to diverse user queries.

**Automatic UI Generation** This direction has evolved from early vision-based approaches to reverse engineering mobile interfaces (Nguyen and Csallner, 2015) to neural network-based end-to-end synthesis systems (Beltramelli, 2018; Robinson, 2019; Aşıroğlu et al., 2019). Recent advances in LLMs have substantially improved UI generation from natural language descriptions (Laurençon et al., 2024), screenshots (Si et al., 2024), and sketches (Li et al., 2024) and iterative refinement via LLM-generated feedback (Li et al., 2024). Alternatively, our work requires no UI specifications from users. More fine-grained control of the UI code generation process encompasses diverse intermediate representation approaches: (I) graph-based representation to capture hierarchical relationships and dependencies between UI elements (Jiang et al., 2024), (II) UI grammar (Kong et al., 2008) to help LLMs for more intuitive and precise layout description (Lu et al., 2023), and (III) data schema-driven UI specification synthesis to guide subsequent generation processes (Cao et al., 2025). Similarly, our framework employs interaction flows and finite state machines to model the reaction to user actions and the evolution of interfaces.

## 6 Conclusion

We introduce **Generative Interfaces** for Language Models, a paradigm in which LLMs proactively generate adaptive, interactive interfaces to better support complex user goals. Our evaluation demonstrates clear advantages over traditional conversational approaches, particularly in structured and information-dense tasks. Our findings further clarify when generative interfaces are most effective and when conversational formats remain competitive. Future directions include integrating multimodal input, domain-specific templates, and collaborative multi-user environments.

## Limitations

First, the system only supports HTML/JavaScript frontends without backend logic, which restricts the complexity of generated interfaces. As tasks grow more complex, more expressive representations beyond interaction flows and finite state machines may be needed. Second, the iterative refinement process introduces latency of up to several minutes, which may be undesirable in real-time settings. Advances in model efficiency and infrastructure could help mitigate this issue. Third, the system generates interfaces for all queries, even when interaction is unnecessary. Future work could incorporate a classifier to determine whether an input requires interaction in context and selectively invoke the generative UI system.

## References


Anthropic. 2024. Claude artifacts.

Anthropic. 2025. Introducing claude 3.7 sonnet.

Apple Inc. 1987. Knowledge navigator. Concept video and vision for future technology.

Batuhan Aşıroğlu, Büşta Rümeysa Mete, Eyyüp Yıldız, Yağız Nalçakan, Alper Sezen, Mustafa Dağtekin, and Tolga Ensari. 2019. Automatic html code generation from mock-up images using machine learning techniques. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pages 1–4.

Elisa Bassignana, Amanda Cercas Curry, and Dirk Hovy. 2025. The AI gap: How socioeconomic status affects language technology interactions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18647–18664, Vienna, Austria. Association for Computational Linguistics.

Tony Beltramelli. 2018. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pages 1–6.

Yining Cao, Peiling Jiang, and Haijun Xia. 2025. Generative and malleable user interfaces with generative and evolving task-driven data model. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, page 1–20. ACM.

Yoon Jeong Cha, Yasemin Gunal, Alice Wou, Joyce Lee, Mark W Newman, and Sun Young Park. 2024. Shared responsibility in collaborative tracking for children with type 1 diabetes and their parents. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Ruijia Cheng, Titus Barik, Alan Leung, Fred Hohman, and Jeffrey Nichols. 2024. Biscuit: Scaffolding llm-generated code with ephemeral uis in computational notebooks. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 13–23. IEEE.

Anind K Dey, Gregory D Abowd, and 1 others. 2000. Towards a better understanding of context and context-awareness. In *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, volume 4, pages 1–6.

Victor Dibia. 2023. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. *arXiv preprint arXiv:2303.02927*.

Peitong Duan, Chin-Yi Cheng, Gang Li, Bjoern Hartmann, and Yang Li. 2024. Uicrit: Enhancing automated design evaluation with a ui critique dataset. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, page 1–17. ACM.

Shiyu Duan. 2025. Systematic analysis of user perception for interface design enhancement. *Journal of Computer Science and Software Applications*, 5(2).

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Leah Findlater and Krzysztof Z Gajos. 2009. Design space and evaluation challenges of adaptive graphical user interfaces. *AI Magazine*, 30(4):68–68.

Krzysztof Gajos and Daniel S Weld. 2004. Supple: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 93–100.

Krzysztof Z Gajos, Jacob O Wobbrock, and Daniel S Weld. 2007. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 231–240.

Jan Hartmann, Alistair Sutcliffe, and Antonella De Angeli. 2008. Towards a theory of user judgment of aesthetics and user interface quality. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 15(4):1–30.

Nobukatsu Hojo, Kazutoshi Shinoda, Yoshihiro Yamazaki, Keita Suzuki, Hiroaki Sugiyama, Kyosuke Nishida, and Kuniko Saito. 2025. Generativegui: Dynamic gui generation leveraging llms for enhanced user interaction on chat interfaces. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–9.

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166.

Jaehyun Jeon, Jang Han Yoon, Min Soo Kim, Sumin Shim, Yejin Choi, Hanbin Kim, and Youngjae Yu. 2025. G-focus: Towards a robust method for assessing ui design persuasiveness. *Preprint*, arXiv:2505.05026.

Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring large language model responses with interactive diagrams. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*, pages 1–20.

Yue Jiang, Changkong Zhou, Vikas Garg, and Antti Oulasvirta. 2024. Graph4gui: Graph neural networks for representing graphical user interfaces. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Jun Kong, Keven L Ates, Kang Zhang, and Yan Gu. 2008. Adaptive mobile interfaces through grammar induction. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 133–140. IEEE.

Michelle S Lam, Janice Teoh, James A Landay, Jeffrey Heer, and Michael S Bernstein. 2024. Concept induction: Analyzing unstructured text with high-level concepts using lloom. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–28.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. 2024. Unlocking the conversion of web screenshots into html code with the websight dataset. *Preprint*, arXiv:2403.09029.

Chunggi Lee, Sanghoon Kim, Dongyun Han, Hongjun Yang, Young-Woo Park, Bum Chul Kwon, and Sungahn Ko. 2020. Guicomp: A gui design assistant with real-time, multi-faceted feedback. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13. ACM.

Ryan Li, Yanzhe Zhang, and Diyi Yang. 2024. Sketch2code: Evaluating vision-language models for interactive web design prototyping. *Preprint*, arXiv:2410.16232.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2023. A dynamic llm-powered agent network for task-oriented agent collaboration. *Preprint*, arXiv:2310.02170.

Yuwen Lu, Ziang Tong, Qinyi Zhao, Chengzhi Zhang, and Toby Jia-Jun Li. 2023. Ui layout generation with llms guided by ui grammar. *arXiv preprint arXiv:2310.15455*.

Kalle Lyytinen and Youngjin Yoo. 2002. Ubiquitous computing. *Communications of the ACM*, 45(12):63–96.

Xiao Ma, Swaroop Mishra, Ariel Liu, Sophie Ying Su, Jilin Chen, Chinmay Kulkarni, Heng-Tze Cheng, Quoc Le, and Ed Chi. 2024. Beyond chatbots: Explorellm for structured thoughts and personalized model responses. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Fangwen Mu, Lin Shi, Song Wang, Zhuohao Yu, Binquan Zhang, Chenxue Wang, Shichao Liu, and Qing Wang. 2023. Clarifygpt: Empowering llm-based code generation with intention clarification. *arXiv preprint arXiv:2310.10996*.

Palash Nandy, Sigurdur Orn Adalgeirsson, Anoop K Sinha, Tanya Kraljic, Mike Cleron, Lei Shi, Angad Singh, Ashish Chaudhary, Ashwin Ganti, Christopher A Melancon, and 1 others. 2024. Bespoke: using llm agents to generate just-in-time interfaces by reasoning about user intent. In *Companion Proceedings of the 26th International Conference on Multimodal Interaction*, pages 78–81.

Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse engineering mobile application user interfaces with remaui (t). In *2015 30th IEEE/ACM international conference on automated software engineering (ASE)*, pages 248–259. IEEE.

Jeffrey Nichols, Brad A Myers, Michael Higgins, Joseph Hughes, Thomas K Harris, Roni Rosenfeld, and Mathilde Pignol. 2002. Generating remote control interfaces for complex appliances. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 161–170.

Jeffrey Nichols, Brad A Myers, and Brandon Rothrock. 2006a. Uniform: automatically generating consistent remote control user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 611–620.

Jeffrey Nichols, Brandon Rothrock, Duen Horng Chau, and Brad A Myers. 2006b. Huddle: automatically generating interfaces for systems of multiple connected appliances. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 279–288.

Jakob Nielsen and 1 others. 2012. Usability 101: Introduction to usability.

OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024b. Openai canvas.

Evan F Risko and Sam J Gilbert. 2016. Cognitive offloading. *Trends in cognitive sciences*, 20(9):676–688.

Alex Robinson. 2019. Sketch2code: Generating a website from a paper mockup. *ArXiv*, abs/1905.13750.

Steven I Ross, Fernando Martinez, Stephanie Houde, Michael Muller, and Justin D Weisz. 2023. The programmer's assistant: Conversational interaction with a large language model for software development. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 491–514.

Richard K Shehady and Daniel P Siewiorek. 1997. A method to automate user interface testing using variable finite state machines. In *Proceedings of IEEE 27th International Symposium on Fault Tolerant Computing*, pages 80–88. IEEE.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. 2024. Design2code: Benchmarking multimodal code generation for automated front-end engineering. *Preprint*, arXiv:2403.03163.

Alex Tamkin, Miles McCain, Kunal Handa, Esin Durmus, Liane Lovitt, Ankur Rathi, Saffron Huang, Alfred Mountfield, Jerry Hong, Stuart Ritchie, Michael Stern, Brian Clarke, Landon Goldberg, Theodore R. Sumers, Jared Mueller, William McEachen, Wes Mitchell, Shan Carter, Jack Clark, and 2 others. 2024. Clio: Privacy-preserving insights into real-world ai use. *Preprint*, arXiv:2412.13678.

Yin-Leng Theng and Henry Duh. 2008. *Ubiquitous Computing: Design, Implementation and Usability (Premier Reference Source)*. IGI Global, USA.

Priyan Vaithilingam, Elena L Glassman, Jeevana Priya Inala, and Chenglong Wang. 2024. Dynavis: Dynamically synthesized ui widgets for visualization editing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, and Peter Wolstenholme. 2006. *Modeling software with finite state machines: a practical approach*. Auerbach Publications.

Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. Promptchainer: Chaining large language model prompts through visual programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–10.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zixin Wang, Xinyi Xu, Wenxuan Wang, Zhiyao Xu, Yuhang Wang, and Michael R. Lyu. 2024. Interaction2code: Benchmarking mllm-based interactive webpage code generation from interactive prototyping. *Preprint*, arXiv:2411.03292.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

## A Potential Risks

While our work focuses on foundational research in generative user interfaces, it carries several potential risks. The ability to automatically generate interactive interfaces could be misused to create deceptive or manipulative content, and the generated interfaces may contain inaccurate or misleading information. There is also a risk of amplifying accessibility gaps if generative interfaces are optimized primarily for visually rich modalities, potentially disadvantaging users with visual or cognitive impairments. We advocate for future safeguards, such as provenance tracking, user consent mechanisms, and alignment with accessibility standards, to ensure the responsible and inclusive use of generative interface technologies.

## B Prompt Suite

To evaluate system performance across realistic user intents, we curated a prompt suite covering ten practical domains: `Web & Mobile App Development`, `Content Creation & Communication`, `Academic Research & Writing`, `Education & Career Development`, `Advanced AI/ML Applications`, `Business Strategy & Operations`, `Language Translation`, `DevOps & Cloud Infrastructure`, `Digital Marketing & SEO`, and `Data Analysis & Visualization`.

Each prompt belongs to one of four quadrants based on *detail level* (concise *vs.* detailed) and *type* (general *vs.* interactive), ensuring coverage of diverse user tasks and complexity levels.

**Example Prompts:**

- **Concise & General:** *"How can I learn piano effectively?"*
- **Concise & Interactive:** *"I want to create an infographic about water conservation."*
- **Detailed & General:** *"I'm writing a dissertation on the psychological effects of social media use among teenagers. I've collected survey and interview data but am struggling to integrate them in the analysis chapter. What methodological approach should I use to synthesize these data types rigorously?"*
- **Detailed & Interactive:** *"I'm developing a website for a local bookstore where customers can browse inventory, register for book club*

*meetings, and sign up for our newsletter. I want a cozy design but have no coding experience. The inventory is in Excel and updates weekly. What's the best approach to build this site?"*

## C Evaluation Metric Details

To assess the quality of LLM interfaces, we adopt a comprehensive set of evaluation metrics adapted from Nielsen et al. (2012) and Hartmann et al. (2008), capturing three core dimensions of user perception: functional, interactive, and emotional. *Functional Perception* includes **Query-Interface Consistency (QIC)**, which evaluates how well the generated interface aligns with and fulfills the user's query intent (Duan, 2025), and **Task Efficiency (TaskEff)**, which measures how efficiently users can achieve their goals with minimal effort or time (Nielsen et al., 2012; Duan, 2025). *Interactive Perception* comprises **Usability**, assessing interface clarity and actionable structure (Hartmann et al., 2008; Nielsen et al., 2012); **Learnability**, indicating how easily new users can begin using the interface without prior experience (Nielsen et al., 2012); and **Information Clarity (IC)**, which evaluates information organization, readability, and interpretability (Hartmann et al., 2008; Cao et al., 2025). Finally, *Emotional Perception* covers **Aesthetic or Stylistic Appeal (ASA)**, reflecting the visual consistency and attractiveness of the design (Hartmann et al., 2008; Duan et al., 2024), and **Interaction Experience Satisfaction (IES)**, capturing the user's overall satisfaction and engagement with the interface (Duan, 2025). This enables a comprehensive assessment of user experience by tracing the full perceptual process—"how users understand the interface" → "how they operate it" → "how they emotionally respond". Instead of using the traditional Likert scale, we adopt a pairwise comparison approach, following (Zheng et al., 2023; Si et al., 2024). That is, for each query, we present two interfaces to human annotators and ask for their preferences on all seven dimensions, as well as their overall preferences.

## D LLM Evaluation

User-centered evaluation remains the gold standard for UI assessment due to interfaces' fundamental purpose of facilitating human interaction and operation (Hartmann et al., 2008; Duan, 2025; Cao et al., 2025). However, generative interfaces requiring real-time synthesis and rapid iterative refinement cannot depend on user feedback, necessitating robust automatic evaluation frameworks.

Early approaches employed manually crafted behavioral prediction metrics (Lee et al., 2020), though these methods demonstrated limited generalizability and required substantial domain expertise. Recent research has increasingly leveraged LLMs for UI assessment, with Duan et al. (2024) employing LLMs to generate design feedback and quality ratings with bounding box annotations. Jeon et al. (2025) extends this paradigm to persuasiveness evaluation, achieving meaningful correlation with empirical A/B testing outcomes. In this work, we ask LLMs to judge the same dimensions that we ask human annotators, including some previously human-exclusive metrics such as task efficiency and learnability.

Specifically, we use a listwise ranker (Liu et al., 2023) to evaluate different interface variants of the same user query by presenting the LLM (Claude 3.7) with UI codes and screenshots, where the LLM assigns scores ranging from 0 to 100 for each evaluation dimension. We compare LLM evaluation scores with pairwise annotations from humans, which yields an agreement rate of 69.0%. While it suggests LLM as a reliable proxy for convenient and scalable evaluations, we also observe common issues like length bias (Dubois et al., 2024) which might favor GenUI.

## E Representation: Natural language *vs.* Structured

To present a more fine-grained comparison, we showcase two distinct representations of the same user intent. The user prompt used here is:

> "*I want to understand quantum physics principles.*"

A natural language representation includes the goal, salient features, technical requirements, and user preferences, which are expressed through multiple descriptive fields. This format provides rich detail about the UI requirements without imposing any constraints on the interface states or their transitions.

**Natural language representation**

```
{
  "mainGoal": "Create an
    interactive learning interface
```

| Framework | *Functional* | | *Interactive* | | | *Emotional* | |
|---|---|---|---|---|---|---|---|
| | QIC | TaskEff | Usability | Learnability | IC | ASA | IES |
| *- Score:* | | | | | | | |
| ConvUI (Claude 3.7) | 65.8 | 47.6 | 34.7 | 72.4 | 76.1 | 47.7 | 41.1 |
| ConvUI (GPT-4o) | 70.2 | 51.0 | 36.8 | 74.9 | 80.2 | 48.1 | 43.1 |
| IUI | 68.0 | 58.0 | 57.9 | 73.8 | 72.5 | 70.8 | 56.0 |
| GenUI | **86.1** | **84.2** | **87.0** | **84.0** | **88.5** | **88.9** | **87.2** |
| *- Relative Improvement (%):* | | | | | | | |
| *vs.* ConvUI (Claude 3.7) | 30.9% | 76.6% | 151.0% | 16.0% | 16.2% | 86.2% | 112.4% |
| *vs.* ConvUI (GPT-4o) | 22.7% | 65.1% | 136.2% | 12.2% | 10.4% | 84.8% | 102.3% |
| *vs.* IUI | 26.7% | 45.0% | 50.2% | 13.8% | 22.0% | 25.5% | 55.7% |

Table 5: **LLM-Based Evaluation Scores Across Perception Dimensions.** Automatic assessment (0–100 scale) of UI frameworks across functional, interactive, and emotional perception categories.

```
      for understanding quantum
    physics principles.",
  "keyFeatures": [
    "Step-by-step tutorials on key
     quantum physics concepts",
    "Interactive simulations
    demonstrating quantum
    mechanics principles",
    "Visual aids such as diagrams
    and animations to enhance
    comprehension",
    "Quizzes and assessments to
    test understanding and
    reinforce learning",
    "Discussion forums for peer
    interaction and support"
  ],
  "technicalRequirements": [
    ...
  ],
...
}
```

Structured interface-specific representation is state-oriented and descriptive. In table 6, we summarize concepts and symbols used in structured interface-specific representations.

**Structured representation**

```
{
  "description": "An interactive
    educational platform for
    learning quantum physics
    principles through tutorials,
    simulations, quizzes, progress
     tracking, and discussion
    forums. The platform offers
    step-by-step learning paths,
    visual demonstrations of
    quantum phenomena, and
    assessment tools to help users
     understand complex quantum
    physics concepts.",
  "metadata": {
    "title": "Quantum Physics
    Explorer - Interactive
    Learning Platform",
    "metaDescription": "Learn
    quantum physics through
    interactive tutorials,
    simulations, quizzes, and
    discussion forums. A
    comprehensive educational
    platform for understanding
    quantum mechanics principles."
  },
  "states": [
    {
      "name": "isMobileMenuOpen",
      "initialValue": "false",
      "description": "Controls the
     visibility of the mobile
    navigation menu on smaller
    screens."
    },
    ...
  "elements": [
    ...
    {
      "id": "helpButton",
      "parentId": "userControls",
      "elementType": "button",
      "content": "Help",
      "className": [
        "text-blue-600",
        "hover:text-blue-800",
        "focus:outline-none",
        "focus:ring-2",
        "focus:ring-blue-500",
        "rounded-full",
        "p-2"
      ],
      "functionality": "Provides
    access to help resources and
    tutorials.",
      "attributes": {
        "ariaLabel": "Get help"
      },
      "events": [
        {
          "type": "onClick",
          "handlerDescription": "
    Opens the help modal with
```

| Term / Symbol | Definition |
|---|---|
| **Interaction Flow** | A high-level abstraction over user interaction sequences, modeling task progression as transitions across interface views. |
| $\mathcal{G} = (\mathcal{V}, \mathcal{T})$ | Directed graph structure of the interaction flow: $\mathcal{V}$ is the set of views or subgoals, and $\mathcal{T}$ is the set of transitions between them. |
| $\mathcal{V}$ | Nodes in the interaction graph, each representing a specific UI view or a subgoal in the task. |
| $\mathcal{T}$ | Directed edges indicating possible user-triggered transitions between views, such as button clicks or link navigation. |
| **Finite State Machine (FSM)** | A formalism used to describe the behavior and state transitions of individual UI components based on user interactions. |
| $\mathcal{M} = (\mathcal{S}, \mathcal{E}, \delta, s_0)$ | Formal definition of an FSM: $\mathcal{S}$ is the state set, $\mathcal{E}$ the event set, $\delta$ the transition function, and $s_0$ the initial state. |
| $\mathcal{S}$ | Set of all possible atomic interface states (*e.g.*, `isModalOpen=true`, `activeTab=2`). |
| $\mathcal{E}$ | Set of discrete user-triggered events such as `click`, `hover`, `input`, etc. |
| $\delta$ | Transition function $\delta : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{S}$, defining how a component's state evolves given an event. |
| $s_0$ | The initial state of the UI component when the interface is first rendered. |

Table 6: Glossary of concepts and formal symbols used in structured interface-specific representation.

```
tutorials and resources.",
      "affects": [
        {
          "target": "
isHelpModalOpen",
          "action": "
updateState",
          "details": "true"
        }
      ]
    }
  ],
  "interactions": {
    "hover": {
      "className": [
        "text-blue-800",
        "bg-blue-50"
      ]
    },
    "focus": {
      "className": [
        "ring-2",
        "ring-blue-500"
      ]
```

```
        }
      }
    },
      "flows": [
    {
      "name": "Explore Tutorials",
      "description": "User
    navigates to and interacts
    with the tutorials section to
    learn about quantum physics
    concepts.",
      "steps": [
        "User scrolls down to the
    'Quantum Physics Tutorials'
    section or clicks on the '
    Tutorials' navigation item.",
        ...
      ]
    },
    ...
```

## F Adaptive Reward Function

The reward function consists of multiple evaluation metrics, each defined with four key fields:

- **`name`**: The high-level evaluation dimension.
- **`description`**: A brief explanation of the dimension's purpose.
- **`criteria`**: A list of granular human-interpretable evaluation checks.
- **`weight`**: The relative importance of the metric in the aggregated reward.

These metrics collectively guide the assessment of the interface from both functional and user-centered perspectives.

For example, for the user query "*I want to understand quantum physics principles,*" the adaptive reward metric includes a specific criterion stating that "*Interactive models effectively demonstrate phenomena like wave-particle duality.*", which provides intent-aware reward signals that move beyond generic usability (Figure 4b). In contrast, the static reward approach yields suboptimal results where particle distributions appear as incoherent clusters without proper interference visualization (Figure 4a).

**Adaptive reward function**

```
{
  "name": "generate_metrics",
  "args": {
    "metrics": [
      {
        "name": "Interactive
    Elements Quality",
        "description": "Measures
    the quality of user
    interaction with simulations,
    quizzes, and other dynamic
    components.",
        "weight": 0.15,
        "criteria": [
          "Animations and
    transitions are smooth and non
    -distracting.",
          "User actions (e.g.,
    answering quiz questions,
    changing simulation variables)
     receive clear and immediate
    feedback.",
          "Interactive components
    (simulations, quiz buttons)
    are responsive to user input
    .",
          "User flows through
    tutorials, simulations, and
    quizzes are intuitive.",
          "State changes (e.g.,
    quiz progress, simulation
    results) are accurately
    reflected.",
          "Error prevention
    mechanisms in quizzes (e.g.,
    guiding towards correct
    answers) are effective."
        ]
      },
      ...
    ]
  }
}
```

On the other hand, static rewards are defined by general UI rubrics without query-specific metrics.

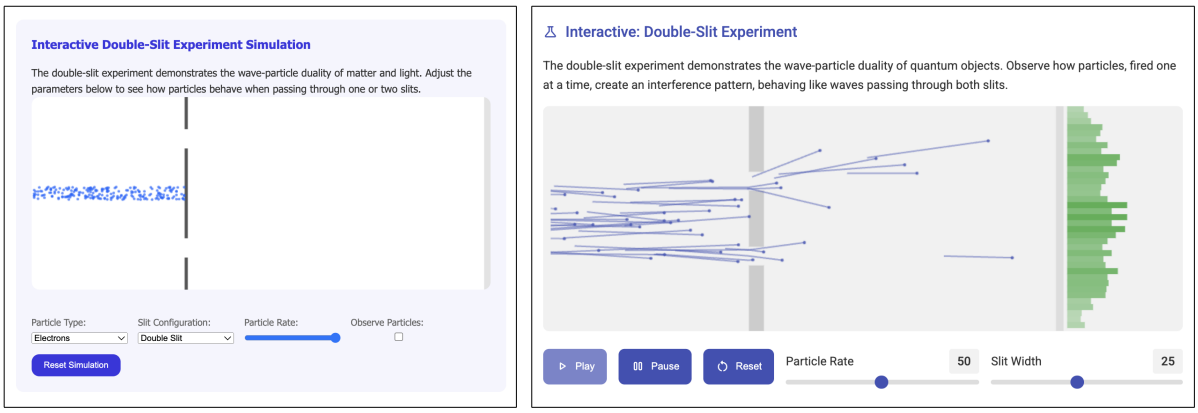
**Static rewawrd**

```
{
  "name": "generate_metrics",
  "args": {
    "metrics": [
      {
        "name": "Core
    Functionality and Web
    Stability",
        "description": "Evaluates
    whether the core functionality
     of the interface works
    correctly, the system is
    stable, and compatible across
    platforms.",
        "weight": 0.2,
        "criteria": [
          "Basic functionality
    working correctly",
          "System stability and
    error handling",
          "Cross-browser
    compatibility",
          "Mobile responsiveness"
        ]
      },
      ...
    ]
  }
}
```

## G Supplementary Examples

- Figure 5 compares GenUI and ConvUI in *Business Strategy & Operations* task.
- Figure 6 shows the iterative refinement process for a continuous integration dashboard. Each version progressively enhances usability and clarity through structure-aware feedback.
- Figure 7 demonstrates that the layout of GenUI significantly improves users' percep-

***User query:*** *"I want to understand quantum physics principles."*




(a) **Static reward.** The simulation fails to visualize wave-particle duality.

(b) **Dynamic reward.** The simulation successfully visualizes wave-particle duality.

Figure 4: **Visual comparison of static and dynamic reward settings.**

tion of clarity, trustworthiness, and professionalism.

## H Human Evaluation Questionnaire Interface

We show the annotation interfaces in Figure 8, 9, 10.

## I Annotator Demographics

All annotators held at least a bachelor's degree and were employed either part-time or full-time. They had extensive data annotation experience, each having completed over $1,000$ tasks with an approval rate exceeding $90\%$. All participants were native English speakers and regular users of AI chatbots (*e.g.*, ChatGPT) in their daily lives.

## J Human Annotation Filtering

To ensure the reliability of human annotations, we employed a multi-stage filtering process involving trap questions, consistency checks, and agreement rate evaluation.

- **Trap Questions.** Each annotation task contained 8 UI comparison questions. In some questionnaires, we embedded trap questions in which the "UI" was not a real interface with components, but rather a simple instruction such as "Select Example A for all options" or "Select Example B for all options." Annotators who failed to follow these explicit instructions were identified as inattentive, and their entire submissions were discarded.
- **Consistency Check.** We manually compared each annotator's multiple-choice selections with their accompanying textual comments. If a comment stated that Example A was better but the selected option was B, we considered this a clear inconsistency indicative of random selection. Such annotations were removed.
- **Manual Review.** We conducted a manual review for annotators who had low agreement with other annotators and determined whether the annotator's responses showed signs of random or careless selection. If so, all responses from that annotator were excluded.

Through this process, we ensured that the retained annotations were both attentive and internally consistent, thereby improving the overall quality of our evaluation.

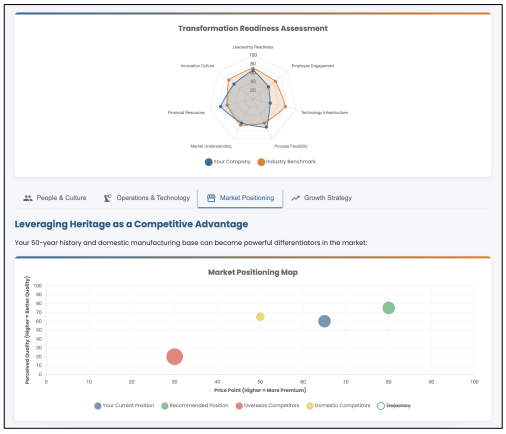
## K Real-User Query Evaluation

We first collect user queries by presenting the following survey.

**Query Collection Survey**

Our study evaluates the quality and effectiveness of user interfaces for AI chatbots.
In the screening survey, you will be asked to tell us five different AI chatbot queries from your daily usage.
In the follow-up survey, you will be presented with pairs of interfaces in response to your queries and you need to carefully compare these interfaces and determine which one better addresses your query.
**Your Chatbot Queries**

***User query:*** *"I'm a consultant working with a family-owned manufacturing business that's been operating for 50 years. They're facing increasing ... How should they approach strategic transformation while preserving their heritage and retaining institutional knowledge?"*




(a) **GeneUI** presents multiple charts and visual summaries.

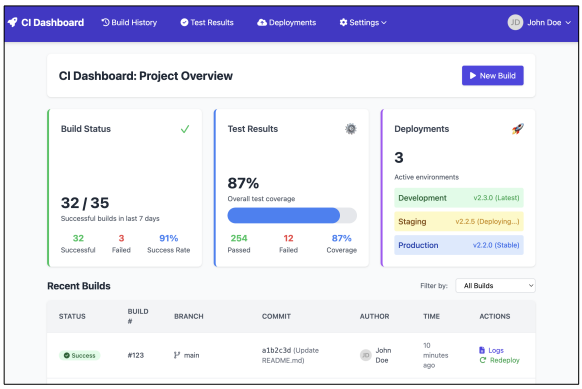
(b) **ConvUI** directly outlines strategy in a sectioned format.

Figure 5: **GenUI** ***vs.*** **ConvUI** in *Business Strategy & Operations* task.

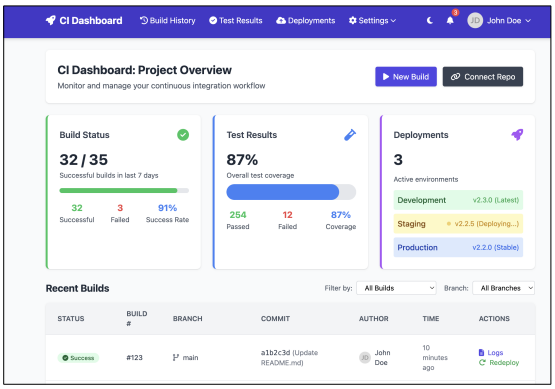
> Share 5 different **specific** queries you typically use ChatGPT, Claude, or other generative AI tools.
> **These will be used to generate personalized interface comparisons for you to evaluate.**
> For example:
> What are some attractions in New York City?
> How should I improve my tennis skills?
>
> Explain artificial intelligence in a simple way.

Note that the provided examples are randomly sampled from initial user submissions to ensure that users understand we are collecting specific queries, rather than high-level descriptions. To ensure the quality of collected queries, we set a minimal number of characters required for each query to be 35. We specified the exact requirements for participants as the human annotation for the UIX benchmark, ensuring that each user is a regular user of ChatGPT/Claude or similar products. Without any post-processing, all user queries are then used to generate one conversation interface and one generative interface. To allow for a high rate limit in this human study, we use Gemini-2.5-flash for both types of interfaces. These interfaces are then shown to the users (similar to Appendix Section H), and for each pair, we ask the user to provide an overall judgment of which kind of interface they prefer for their own query.

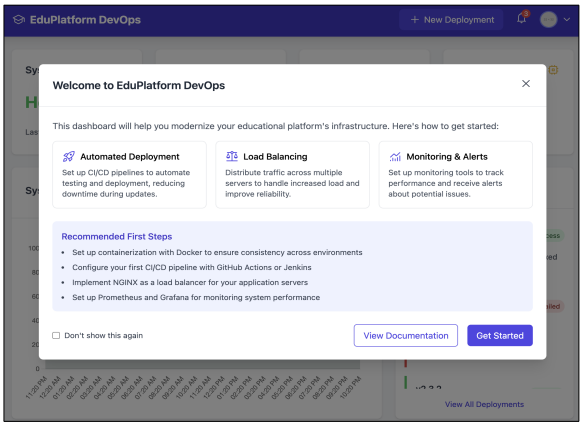
***User query:*** *"I want to set up a continuous integration workflow."*



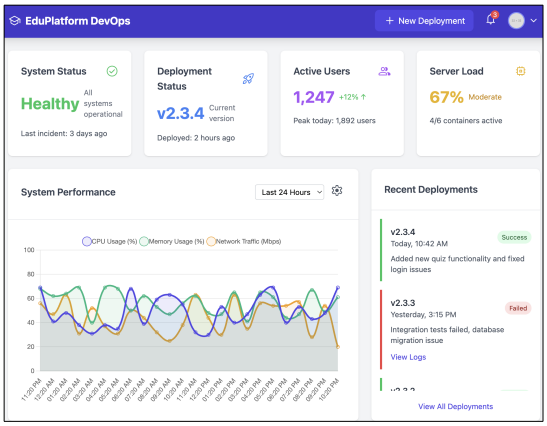

(a) **Iteration 1:** A basic CI dashboard with textual build/test summaries and limited interaction affordances.




(b) **Iteration 2:** Improves layout compactness by closing excessive gaps and clarifies the CI context with stronger visual grouping.



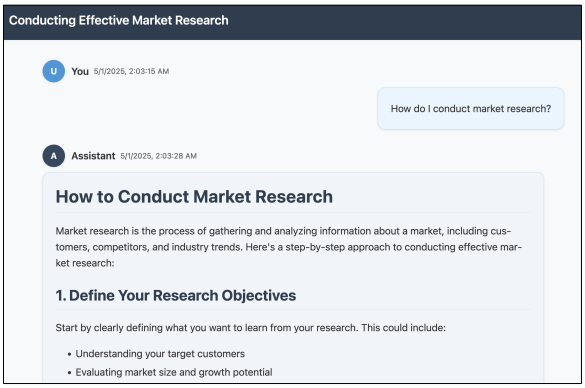

(c) **Iteration 3 (Onboarding page):** Introduces an onboarding modal outlining key components and recommended first steps.



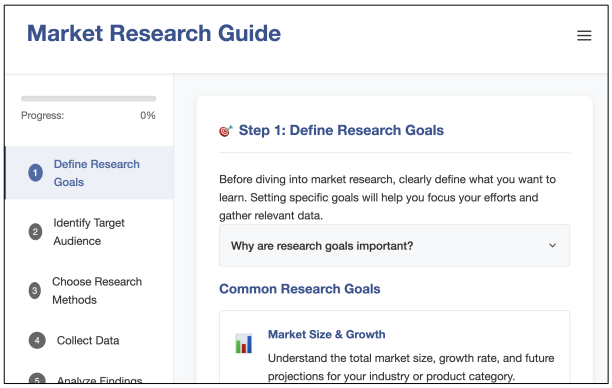

(d) **Iteration 3 (Main page):** Refactors layout to present deployment insights visually, using charts to highlight system status and activity trends.

Figure 6: **Evolution across UI iterations** for the *Continuous Integration Workflow* setup. Each version builds upon its predecessor by reducing visual clutter, providing onboarding guidance, and progressively enhancing the clarity of system performance and CI process feedback.

***User query:*** *"How do I conduct market research?"*

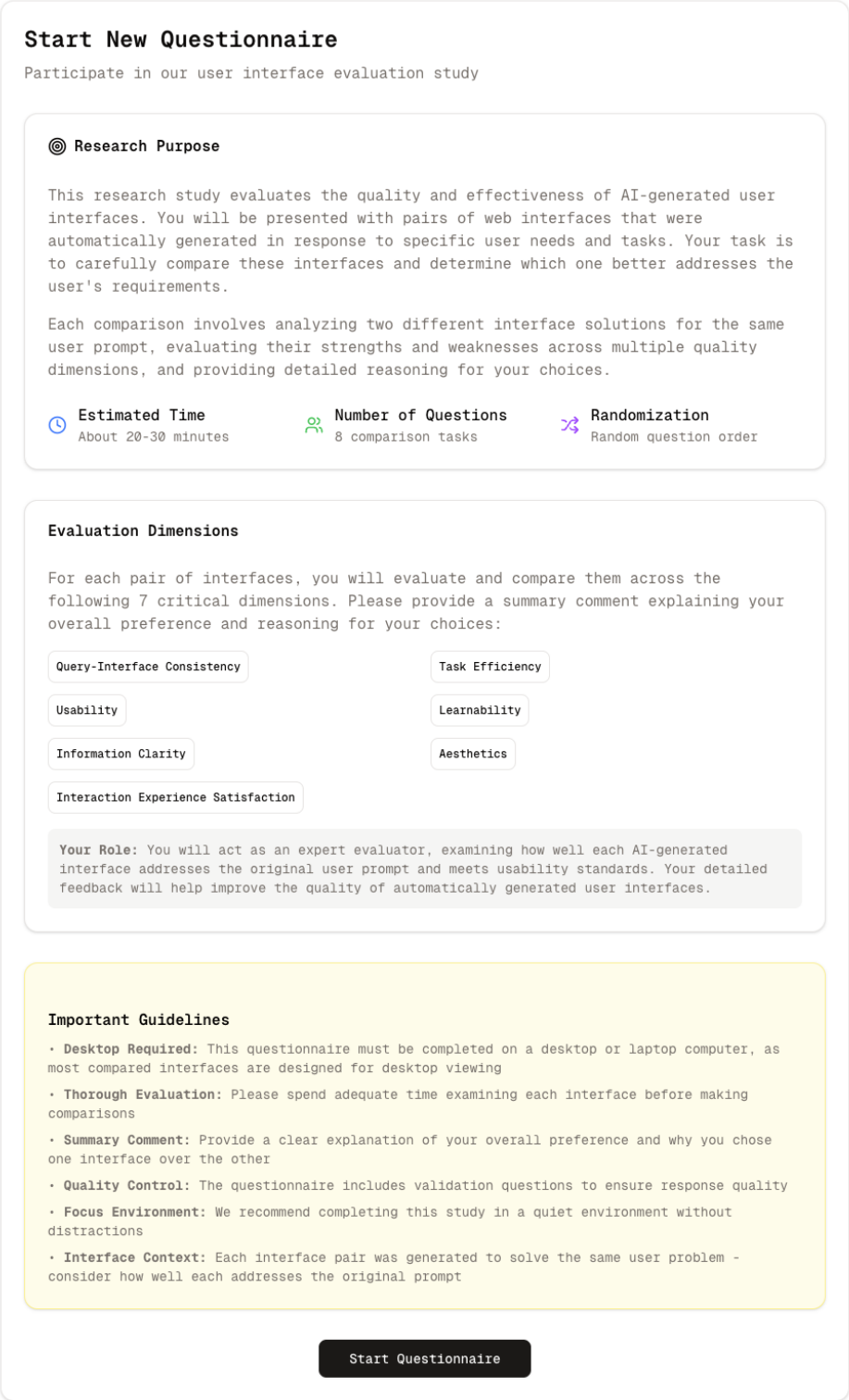



(a) **ConvUI.** Presents information as plain linear text without visual hierarchy, making it harder to navigate.




(b) **GenUI.** Organizes content into modular sections with clear structure, guiding users through the research process.

Figure 7: **Visual structure enhances perceived professionalism.** Despite conveying similar content, GenUI was consistently rated as more trustworthy and well-organized due to its structured layout and visual clarity.



## Start New Questionnaire

Participate in our user interface evaluation study

### Research Purpose

This research study evaluates the quality and effectiveness of AI-generated user interfaces. You will be presented with pairs of web interfaces that were automatically generated in response to specific user needs and tasks. Your task is to carefully compare these interfaces and determine which one better addresses the user's requirements.

Each comparison involves analyzing two different interface solutions for the same user prompt, evaluating their strengths and weaknesses across multiple quality dimensions, and providing detailed reasoning for your choices.

**Estimated Time**
About 20-30 minutes

**Number of Questions**
8 comparison tasks

**Randomization**
Random question order

### Evaluation Dimensions

For each pair of interfaces, you will evaluate and compare them across the following 7 critical dimensions. Please provide a summary comment explaining your overall preference and reasoning for your choices:

- Query-Interface Consistency
- Task Efficiency
- Usability
- Learnability
- Information Clarity
- Aesthetics
- Interaction Experience Satisfaction

**Your Role:** You will act as an expert evaluator, examining how well each AI-generated interface addresses the original user prompt and meets usability standards. Your detailed feedback will help improve the quality of automatically generated user interfaces.

### Important Guidelines

- **Desktop Required:** This questionnaire must be completed on a desktop or laptop computer, as most compared interfaces are designed for desktop viewing
- **Thorough Evaluation:** Please spend adequate time examining each interface before making comparisons
- **Summary Comment:** Provide a clear explanation of your overall preference and why you chose one interface over the other
- **Quality Control:** The questionnaire includes validation questions to ensure response quality
- **Focus Environment:** We recommend completing this study in a quiet environment without distractions
- **Interface Context:** Each interface pair was generated to solve the same user problem - consider how well each addresses the original prompt

Start Questionnaire

Figure 8: Human Evaluation Questionnaire Interface (a)

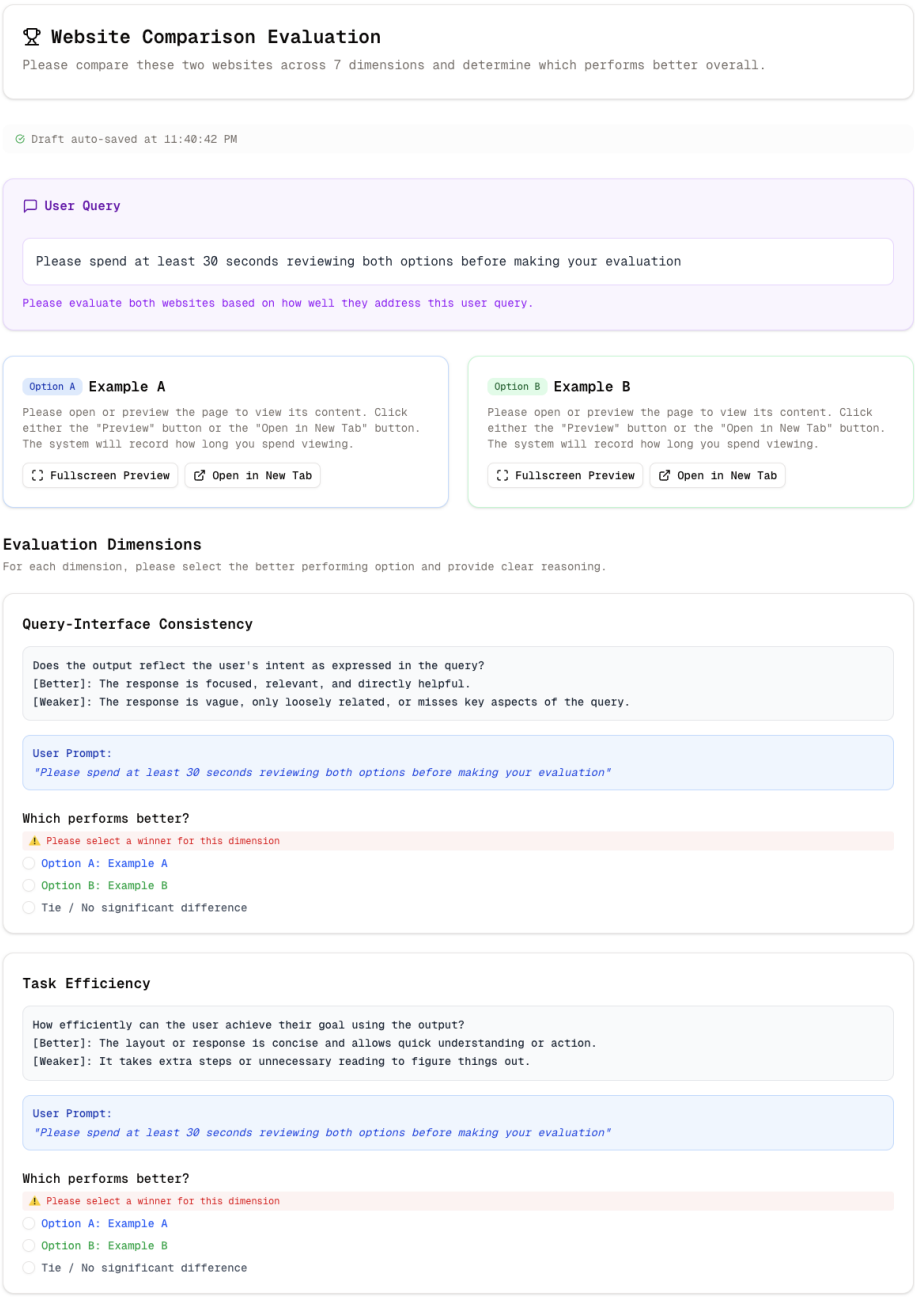



Figure 9: Human Evaluation Questionnaire Interface (b)

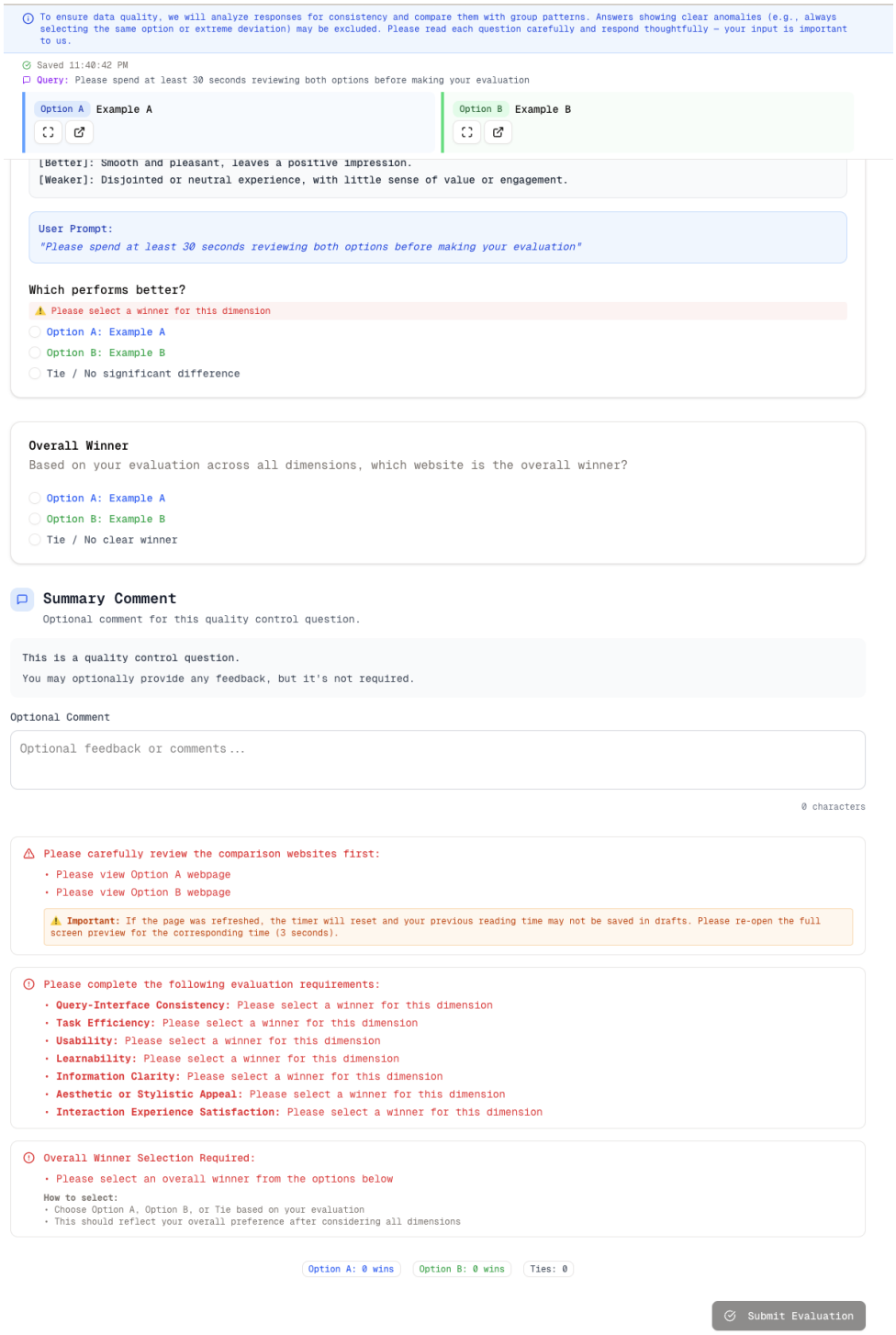



Figure 10: Human Evaluation Questionnaire Interface (c)