AI-Assisted Frontend Code Enhancement

Jason Hu
UIUC
Urbana, USA
jasonh11@illinois.edu

Nianze Guo
UIUC
Urbana, USA
nianzeg2@illinois.edu

Ziyue Zhuang
UIUC
Urbana, USA
ziyue14@illinois.edu

Chenhan Luo
UIUC
Urbana, USA
chenhan8@illinois.edu

ABSTRACT
While large language models such as ChatGPT, Claude, Gemini, and DeepSeek have demonstrated significant success in backend code generation, their ability to enhance frontend usability, aesthetics, and maintainability remains underexplored. This project investigates how LLMs perform in improving poorly designed web interfaces by transforming low-quality JavaScript code into cleaner, more usable versions while realizing features such as animation, table/graph creation, and implementation of specified style. We develop a comparative pipeline that evaluates AI enhanced UIs based on a structured rubric covering UI design, UX interaction, code quality, and AI-specific challenges such as template reliance and creativity limitations. We further validate our findings through simulated user ratings from 200 participants, enabling comparison between expert evaluations and end-user perceptions. Our results reveal distinct model strengths, by triangulating technical metrics with human-centered evaluation, we highlight both the potential and limitations of LLMs in frontend refinement, offering practical guidance for model selection in UI development workflows.

Author Keywords
AI-generated code; frontend development; UI design; usability evaluation; component libraries; human-computer interaction; maintainability.

CSS Concepts
• **Human-centered computing~Human computer interaction (HCI)**;

INTRODUCTION
Recent advances in large language models such as ChatGPT, Claude, Gemini, and DeepSeek have significantly improved automation in backend software development, particularly in tasks like bug fixing, logic synthesis, and code generation. However, the application of LLMs to frontend development remains relatively underexplored, especially in areas that emphasize usability, visual design, and user experience. Most existing research focuses on whether the code functions correctly, rather than whether the resulting interface is intuitive, visually appealing, or accessible.

This paper addresses that gap by asking an essential question: How can LLMs transform disorganized, developer-written frontend code into clean, user-centered interfaces? The focus is on e-commerce web pages with poor layout and structure, which are fed into various LLMs along with prompts that incorporate feedback from user surveys. The generated outputs are then evaluated based on qualities such as layout clarity, responsiveness, modularity, and accessibility.

To explore the capabilities of these models, a multi-step evaluation pipeline was designed. Starting from low-quality JavaScript codebases, user-informed prompts were applied to several LLMs to generate frontend designs for three distinct page types: homepage, item detail, and price tracking. These outputs were assessed using a custom scoring rubric grounded in user experience principles, and functional prototypes were created to support structured evaluations by both users and developers. Additionally, a simulated user evaluation was conducted to compare model outputs from an end-user perspective.

By comparing the behavior and output of different models under controlled conditions, this research highlights what current LLMs can do well, where they fall short, and how they respond to varying prompt strategies. By triangulating expert-driven and user-centered evaluations, this work contributes a holistic understanding of how LLMs behave in frontend refinement tasks. It also provides actionable guidance for integrating generative models into real-world UI development workflows.

RELATED WORK
Recent advancements in large language models have significantly impacted backend software engineering tasks, including logic synthesis, bug detection, and automated code repair [4]. In contrast, their application in frontend development, particularly in usability- and design-focused transformations, remains underexplored. This work addresses this gap by investigating whether LLMs can improve frontend design quality through user-centered design metrics rather than focusing solely on code correctness.

LLMs for Frontend Repair and Design
The intersection of LLMs and frontend development has begun to attract research attention, though most existing systems emphasize structural validity over user experience. For instance, DesignRepair employs a dual-stream LLM pipeline to reconcile frontend code with high-level design rules [12]. However, its evaluation primarily hinges on guideline compliance rather than subjective or aesthetic design quality. Similarly, InteractiveWeb supports iterative updates to HTML, CSS, and JavaScript based on user input

[13], but it centers on technical validity and functional alignment rather than perceptual layout quality or accessibility.

Other works, such as The Role of AI in Frontend Development [7], discuss LLMs' contributions to productivity, such as generating UI components and automating tests, but do not evaluate outcomes using experiential design metrics. Collectively, these systems demonstrate LLM feasibility in frontend contexts, yet they largely optimize for developer efficiency and syntactic precision at the expense of end-user satisfaction.

In contrast, this research investigates whether LLMs can convert dense, developer-written frontend code into modular, accessible, and visually coherent designs. This transformation is framed using structured UX evaluation criteria including layout clarity, responsiveness, visual hierarchy, and accessibility compliance, areas that have received limited attention in existing literature.

Benchmarks and Evaluation Frameworks

Several benchmarks support LLM evaluation in frontend tasks, typically emphasizing layout translation or general code editing. Design2Code [15], WebApp1K [1], and StructEval [3] contribute structure-aware frameworks for assessing model outputs, with StructEval particularly focusing on how well LLM-generated code adheres to intended semantic structures. ENAMEL contributes accessibility-focused annotations to assess LLMs' awareness of inclusive design, while CodeEditorBench evaluates editing capabilities across multiple programming languages and interface contexts.

However, few of these benchmarks integrate qualitative UX principles into their evaluation schemes. Most rely on syntactic correctness, structural preservation, or code similarity as proxies for design quality. In contrast, this work complements those efforts by introducing a user-centered evaluation rubric that incorporates direct user feedback alongside established design heuristics. The rubric assesses frontend outputs along dimensions such as visual clarity, layout cohesion, readability, and accessibility. These are factors that directly affect end-user experience but remain underrepresented in automated evaluation pipelines.

By situating this contribution at the intersection of code transformation and user-centered design, it provides a novel perspective that augments existing LLM benchmarks and systems while addressing key gaps in experiential and qualitative evaluation.

METHODOLOGY

To assess the capacity of large language models to enhance the quality of frontend code from a user-centered design perspective, we developed a structured experimental framework that integrates empirical user research, design heuristics, prompt engineering, and comparative model evaluation. This framework, illustrated in Figure 1, provides a systematic pipeline for investigating the responsiveness of state-of-the-art LLMs to real-world frontend development challenges.

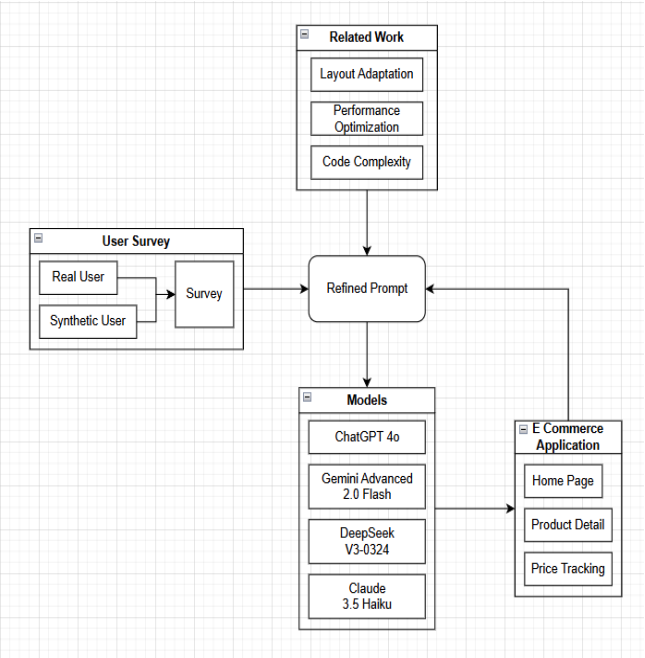

Figure 1. Experimental pipeline integrating user research and related work to generate refined prompts for evaluating LLMs across ecommerce frontend scenarios.

User Research and Persona Construction

The evaluation process commenced with a mixed-methods user research phase incorporating both real-user input and synthetic persona modeling. Real users were recruited to complete structured surveys evaluating critical e-commerce interface attributes, including layout clarity, visual hierarchy, usability, and accessibility. Responses were analyzed to identify recurring pain points and design preferences across different usage contexts.

To supplement this data and enhance the inclusivity of the evaluation criteria, we constructed a set of synthetic user personas representing diverse cognitive styles, accessibility needs, and device usage environments. These personas were informed by established UX research guidelines and accessibility taxonomies, allowing us to generalize user-centered design goals across a broader population while reducing potential evaluation bias.

Design Objectives and Prompt Engineering

Building on these user insights, we derived a set of design objectives informed by established literature on layout adaptability, semantic HTML practices, performance optimization, and frontend modularity. These objectives

were translated into scenario-specific prompts designed to guide LLMs toward generating frontend code that reflects both technical soundness and user-centered design principles.

Instead of relying on generic or pre-trained templates, the prompts were crafted using a combination of empirical findings and theoretical frameworks. Each prompt highlighted critical UX criteria such as responsiveness, semantic structure, and modular layout construction, ensuring alignment with real user preferences as well as recognized design standards. Recent methods like Self-Refine demonstrate the potential of leveraging iterative self-feedback to enhance LLM outputs, suggesting future opportunities to incorporate such mechanisms into frontend code refinement workflows [5].

Scenario Design and Model Selection

Three representative e-commerce interface scenarios were selected to reflect common design challenges and user expectations: a homepage, a product detail page, and a price tracking interface. Each scenario was associated with distinct design goals. The homepage scenario emphasized discoverability and grid-based responsiveness; the product detail page prioritized visual clarity and information modularity; and the price tracking interface focused on legibility, temporal data comparison, and adaptive layout behavior.

These prompts were uniformly applied across four state-of-the-art LLMs selected for their architectural diversity and public availability: ChatGPT-4o, Gemini Advanced 2.0 Flash, DeepSeek V3-0324, and Claude 3.7 Sonnet. Each model received identical developer-authored, low-quality frontend code inputs and the same scenario-specific prompts, thereby facilitating a controlled comparative evaluation.

Evaluation Protocol

The outputs generated by each model were instantiated into fully functional frontend prototypes. These prototypes were then evaluated using a multi-criteria rubric encompassing four key dimensions: layout coherence, accessibility compliance, responsiveness across devices, and modularity of code structure. Evaluation was conducted using a hybrid protocol combining automated analysis tools such as Lighthouse and axe-core, and structured user feedback collected via usability testing sessions.

The rubric was designed to ensure both the reliability of quantitative assessments and the interpretability of qualitative feedback. Scores were aggregated and compared across models and scenarios to analyze variation in design quality and adherence to user-centered principles. While most evaluation frameworks emphasize functional correctness, recent benchmarks such as ENAMEL highlight the need to also assess code efficiency in LLM outputs, revealing performance gaps even in state-of-the-art models [14].

USER RESEARCH

Recent research emphasizes the growing role of AI in UX workflows, particularly in supporting tasks such as persona creation, prototyping, and user feedback analysis [8]. Therefore, to ground the evaluation framework in user-centered design principles, we employed a dual-pronged user research methodology that combined empirical data from real users and with simulated insights derived from large language model-generated synthetic personas. This hybrid approach enabled the collection of both lived user experiences and a diverse range of simulated use cases, thereby supporting a more inclusive and comprehensive assessment of frontend design needs.

Real-User Data Collection

Empirical data were obtained through a structured survey administered to 60 undergraduate and graduate students at the University of Illinois Urbana-Champaign. Participants were recruited through departmental mailing lists and participated voluntarily. The survey aimed to identify common usability challenges in e-commerce interfaces and to elicit preferences regarding key frontend design attributes, including aesthetic clarity, interactivity, navigational efficiency, performance, and modularity.

Participants reviewed mockups of e-commerce pages and responded to a combination of Likert-scale questions and open-ended prompts. Quantitative items asked participants to rate the importance of various user experience dimensions on a five-point scale, with 1 being least important and 5 being most important. Qualitative prompts invited users to elaborate on frustrations and preferences encountered during typical web interactions.

To ensure diversity in device usage contexts, the survey captured participant habits across platforms, with 46.7% reporting primary use of desktop or laptop computers, and 47.9% relying on mobile phones. Notably, 78.8% of respondents identified as general users without formal training in web development or design, thereby providing an authentic end-user perspective.

Synthetic Persona Construction and Simulation

To extend the scope of user input beyond the demographic limits of the university-based sample, we generated 200 synthetic user personas using Gemini-2.0-flash in accordance with recent methodologies in HCI and persona simulation [2;6]. Each persona represented a distinct archetype, defined along dimensions such as technological proficiency, cultural background, device environment, user goals, and cognitive style.

Representative archetypes included Accessibility Advocate, Creative Visionary, Casual Browser, E-Commerce Power User, and many others. Persona construction followed a structured template and was guided by a combination of

HCI literature and user segmentation strategies. To ensure internal consistency, synthetic personas were instructed to maintain coherent behavioral and perceptual characteristics throughout the simulated survey process.

Each persona was prompted with the same set of questions administered to real users. Responses were generated using zero-shot or few-shot prompting methods, depending on the complexity of the response type, and were validated for thematic consistency. Temperature and response length parameters were constrained to simulate realistic variability without introducing excessive hallucination or redundancy.

Integrated Thematic Analysis

The combined corpus of 260 responses, with 60 real users and 200 synthetic personas, was analyzed thematically to identify recurring pain points, design expectations, and usability preferences. Using a grounded theory approach, data were coded inductively to allow key themes to emerge across both real and synthetic respondent groups.

Prominent themes included frustration with cluttered layouts, a strong preference for minimalist design, sensitivity to slow load times, and the need for consistent navigational structures. Accessibility-related issues such as insufficient font contrast, limited keyboard support, and missing ARIA labels were especially evident in feedback from accessibility-focused personas.

These insights guided the creation of both targeted prompt templates and a structured evaluation rubric for assessing LLM-generated frontend code. For instance, frequent mentions of navigational confusion and disorganized layout informed rubric items on visual hierarchy and information modularity. Similarly, concerns about interface responsiveness contributed to the inclusion of interaction feedback as a critical evaluation dimension.

By synthesizing input from both real and simulated users, the evaluation framework was designed to address a wide spectrum of needs, including typical usage scenarios and edge cases. This user-centered foundation enabled a more comprehensive and meaningful assessment of LLM capabilities in frontend design.

Survey Results

To inform the development of design evaluation criteria for LLM-generated frontend code, a structured user survey was conducted with a total of 260 participants. The sample included 60 real users, consisting of undergraduate and graduate students at the University of Illinois Urbana-Champaign, along with 200 synthetic user personas generated using gemini-2.0-flash. This dual-source strategy was intended to capture both authentic user experiences and a wide range of simulated usage scenarios, providing a diverse and inclusive foundation for evaluating design outcomes.

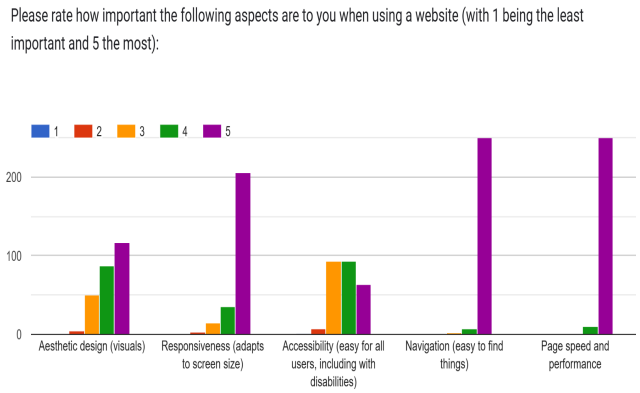

Figure 2. One of the user survey question results showing the perceived importance of the core aspects of website design.

The primary objective of the survey was to identify user priorities, frustrations, and preferences related to frontend web experiences, with a particular focus on e-commerce interfaces. Participants were presented with interface mockups and asked to rate or comment on key design attributes, including visual clarity, interactivity, navigational efficiency, page performance, and layout modularity. Responses were collected through a combination of five-point Likert-scale items and open-ended questions.

As shown in Figure 2, user responses revealed a strong consensus around several core aspects of effective interface design. A substantial majority of respondents of 78.8% self-identified as general users with no formal background in web development or UI/UX design. This characteristic of the sample ensured that the collected data reflected typical end-user perspectives rather than expert or technically specialized viewpoints.

Web usage patterns revealed a high level of engagement, with over 80% of participants reporting website interactions exceeding five times per day. Device preferences were relatively balanced, with 46.7% primarily using desktop or laptop computers and 47.9% relying on mobile phones. These findings emphasize the importance of responsive design and cross-platform compatibility as key considerations in frontend development.

Regarding visual style preferences, 62.7% of participants expressed a preference for minimalist and clean aesthetics. In comparison, 19.2% favored classic or traditional layouts, and only 13.1% preferred bold or colorful visual themes. This clear inclination toward visual simplicity supports the inclusion of minimalist layout principles as a central objective in the evaluation rubric.

When asked about common frustrations in web interaction, 90.8% of respondents cited slow loading times as the most problematic aspect of their experience. Other frequent

issues included cluttered layouts, confusing navigation, and inconsistent styling. The widespread concern with performance, both technical and experiential, informed the prioritization of page speed and load responsiveness in the assessment criteria.

Participants also rated the importance of various UX dimensions using a five-point Likert scale. Page speed and navigation clarity received the highest concentration of top ratings, underscoring their critical role in shaping overall user experience. In contrast, accessibility and visual aesthetics garnered more variable responses, indicating greater subjectivity and dependence on individual context.

Interaction responsiveness, defined as visual or functional feedback following actions like button clicks or form submissions, was also assessed. An overwhelming 99.7% of participants rated this feature as either "very important" (53.5%) or "somewhat important" (46.2%), further reinforcing its role as a key indicator of frontend usability.

In sum, the survey results highlight performance, visual clarity, and navigational simplicity as the most salient frontend design priorities across both real and synthetic users. The observed variability in responses to aesthetic and accessibility features further validates the inclusion of a hybrid user modeling approach in this study. These findings directly informed the construction of both scenario-specific prompts and the multi-criteria evaluation rubric used to assess LLM-generated frontend outputs.

SYSTEM AND PROMPT DESIGN

To assess the capacity of large language models to enhance the quality of front-end code from a user-centered design perspective, we developed a structured experimental framework. This involved designing a React-based e-commerce application testbed comprising three distinct pages, Home, Item Detail, and Price Tracking to evaluate LLM performance across different interface types: List, Image, Form/Chart views. We employed a detailed prompt engineering strategy, inspired by prior research, which included assigning an expert persona to the LLM, outlining core design principles and requirements derived from user research, providing project structure context, and specifying unique features for each page. We then conducted experiments by providing the initial low-quality code and these tailored prompts to four leading LLMs of ChatGPT-4o, Gemini Advanced 2.0 Flash, DeepSeek V3-0324, and Claude 3.7 Sonnet for each page. The resulting user interfaces generated by each model were captured and are presented for comparison.

System Design

To evaluate the capabilities of different large language models in enhancing front-end code quality from a user-centered perspective, we designed and implemented a React-based e-commerce web application as the experimental testbed. This application served as the foundation for testing how well LLMs could transform an initially poorly structured codebase into a more refined and user-friendly interface.

The application was structured into three distinct pages, each designed to test LLM performance on different common frontend views and interaction patterns:

1. Home Page: This page served as the primary interface for testing how LLMs generate and handle design challenges within a List view. The focus was on assessing improvements related to product discoverability, layout organization, and overall aesthetic presentation typical of an e-commerce landing page.
2. Item Detail Page: This page was used to evaluate LLM capabilities in managing Image views and associated interactions. Key challenges included organizing multiple product images, implementing interactive elements like image carousels or magnifiers, and ensuring visual clarity and information modularity for single-item presentation.
3. Price Tracking Page: This page focused on assessing LLM performance in generating and handling data-intensive components, specifically Form and Chart views. The tests involved generating interactive charts to display price history, integrating data visualization libraries, and ensuring the clear presentation of statistical information alongside product details.

This multi-page structure allowed for a systematic evaluation across diverse frontend scenarios, providing a basis for comparing model outputs on tasks ranging from basic layout structuring to complex data visualization and interaction design. The initial codebase provided to the LLMs for each page was intentionally designed with poor structure and styling to create a challenging transformation task.

Prompt Design

The design of our prompts was a critical step in guiding the LLMs to generate user-centered and technically sound front-end code. Our approach to prompt engineering was informed by established practices and tailored to the specific goals of our study. Recent efforts to catalog and structure prompt patterns in other domains, such as financial analysis, demonstrate the broader potential of prompt engineering to shape model performance through deliberate and domain-specific strategies [10].

We drew inspiration from the prompt structure detailed in the "CodeScope: An Execution-based Multilingual Multitask Multidimensional Benchmark for Evaluating LLMs on Code Understanding and Generation" paper by

Yan et al. [11]. Observing their successful methodology, we adapted a similar comprehensive approach for our own prompts.

The core structure of our prompt was designed to provide the LLMs with clear context and instructions, encompassing several key components:

1. Persona Assignment: We assigned the LLM the persona of an "expert front-end developer with deep knowledge of UI design, code style consistency, component library integration, and animation principles." This was intended to prime the model to adopt a professional and experienced perspective.
2. Problem Definition and Task Overview: The prompt clearly stated the objective: to transform an existing low-quality codebase into a "polished, professional, responsive e-commerce website." It emphasized improving visual consistency, spacing, alignment, typography, color usage, and responsiveness.
3. Core Requirements and Best Practices: We instructed the AI to apply professional UI/UX design patterns, ensure intuitive interactivity with accessible behavior, and prioritize smooth state changes and edge case handling. The prompt also encouraged the selection of modern charting or UI component libraries suitable for professional-grade requirements, optimized for responsiveness and maintainability. Furthermore, it highlighted the importance of modularity, maintainability, clarity through consistent naming, reusable components, minimal inline styles, and in-code documentation, all without modifying app-level files.
4. Project Structure Context: To enable the LLM to understand the existing architecture, we provided the current project's file and directory structure.
5. Page-Specific Feature Instructions: A crucial part of our prompt design was the inclusion of a section dedicated to page-specific features. This allowed us to tailor the requirements for each of the three pages:
6. Home Page: Prompts for this page focused on creating a user-friendly style, incorporating animations for items including hover animations, and other typical features of an e-commerce main page.
7. Item Detail Page: Instructions emphasized creating an appropriate style for a single item view, managing image galleries with navigation arrows and progress indicators, implementing a magnifier function for images, and adding animations for smoother user interaction.
8. Price Tracking Page: Prompts detailed the need for an interactive price history chart with toggleable time periods and visualization options like line, bar, and candlestick views, a product information panel, and detailed price statistics.
9. Output Format: The prompt specified that the response should only include the modified or new code files in full, ensuring consistent style conventions, and a list of any external libraries used.

This structured approach, combining a general expert persona and high-level design principles with detailed, page-specific functional requirements, was designed to elicit comprehensive and relevant code generation from the LLMs. The design of these prompts also reflected key findings from our user survey analysis, incorporating elements like modularity, accessibility, responsiveness, and clean layout, which were identified as highly valued by users.

Experiments and Results

To compare the frontend code generation capabilities of different LLMs, we conducted experiments using the system and prompts described previously. The initial low-quality code for each of the three page types, Home Page, Item Detail Page, and Price Tracking Page, was provided as input, along with the corresponding tailored prompt, to four different LLMs: ChatGPT-4o, Gemini Advanced 2.0 Flash, DeepSeek V3-0324, and Claude 3.7 Sonnet. The generated code was then rendered to observe the resulting user interface.

Home Page Results

The prompt for the Home Page focused on creating a user-friendly, visually appealing main page for an e-commerce site, incorporating standard features and item animations. The four models generated distinct visual layouts.

Item Detail Page Results

For the Item Detail Page, the prompt emphasized features like an image carousel with navigation, a magnifier function, and clear presentation of product information. The models implemented these features differently.

Price Tracking Page Results

The prompt for the Price Tracking Page required the generation of an interface with an interactive price history chart, product information panels, price statistics, and options to toggle chart views. The outputs varied significantly.

Screenshots depicting the prompts and rendered output from each model for all three page types are included in the appendix at the end of this paper for visual reference.

EVALUATION

To rigorously assess the structural and experiential quality of LLM-generated frontend code, this paper designed a

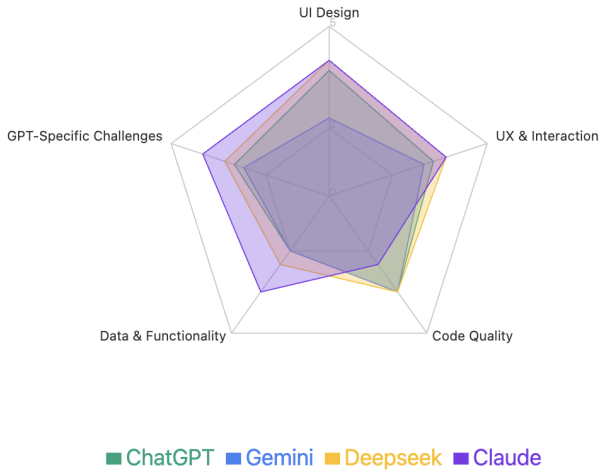

Figure 3. Comparative model performance across five evaluation dimensions: UI Design, UX & Interaction, Code Quality, Data & Functionality, and GPT-Specific Challenges.

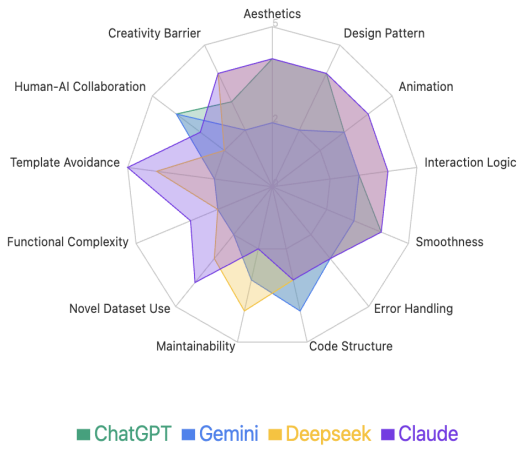

Figure 4. Overlay of model performance across all 13 evaluation criteria.

13-criterion evaluation rubric organized into five overarching dimensions: UI Design, UX & Interaction, Code Quality, Data & Functionality, and Model-Specific Challenges. These criteria were synthesized from empirical user feedback, HCI design heuristics, and contemporary frontend engineering practices.

Each output was evaluated on a five-point ordinal scale (0–5), with detailed scoring anchors defined for each criterion to promote consistency and reduce subjectivity. The UI Design dimension captured aesthetics, design pattern, and animation. UX & Interaction addressed smoothness, interaction logic, and error handling. Code Quality focused on code structure and maintainability. The Data & Functionality dimension assessed novel dataset use

and appropriateness of complexity. The final category, Model-Specific Challenges, evaluated known generative tendencies such as overuse of boilerplate templates, human-AI collaboration issues, or inadequate creativity.

To ensure a fair comparison, all four models—GPT-4o, Claude 3.7 Sonnet, Gemini Advanced 2.0 Flash, and DeepSeek V3—received identical inputs, including a low-quality HTML/CSS/JavaScript codebase and a unified prompt derived from user research themes. Model outputs were instantiated into interactive prototypes for hands-on review.

Quantitative results reveal that Claude 3.7 achieved the highest overall score, demonstrating strong performance across both visual design and structural innovation. It excelled in semantic layout construction, responsive interaction logic, and creativity, while also showing the highest score in avoiding template-based structures.

DeepSeek V3 followed closely, producing visually balanced and modular code with notable strengths in aesthetics, animation, and maintainability, though it was slightly less effective in managing complex functionality.

ChatGPT exhibited consistent competence in layout structuring and design conventions, producing modular, maintainable code. However, its outputs were occasionally constrained by conventional patterns, resulting in lower novelty and creativity scores. Gemini Advanced produced outputs with dynamic visuals and fluid animation, but suffered from structural inconsistencies, reduced semantic clarity, and frequent reliance on templated code structures, leading to the lowest overall score.

These results underscore distinct model tendencies, summarized in Figure 3&4: Claude 3.7 prioritizes structural clarity and innovation, DeepSeek balances stylistic coherence with modularity, ChatGPT emphasizes reliable architecture with moderate creativity, while Gemini favors expressive styling at the cost of structural discipline and maintainability.

DISCUSSION

To validate the consistency of our rubric-based expert evaluation with real-world perceptions, we conducted a user simulation study involving 200 participants. Each participant interacted with LLM-generated versions of three representative e-commerce pages and provided feedback on five quality dimensions: UI Design, UX & Interaction, Code Quality, Data & Functionality, and Overall Satisfaction. Ratings were collected on a 5-point Likert scale, and all model outputs were presented in randomized order to mitigate presentation bias.

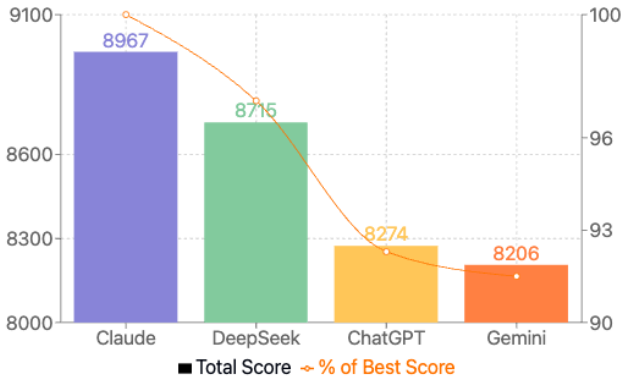

Figure 5. User evaluation results showing total satisfaction scores and relative percentage performance across LLMs (n = 200 participants).

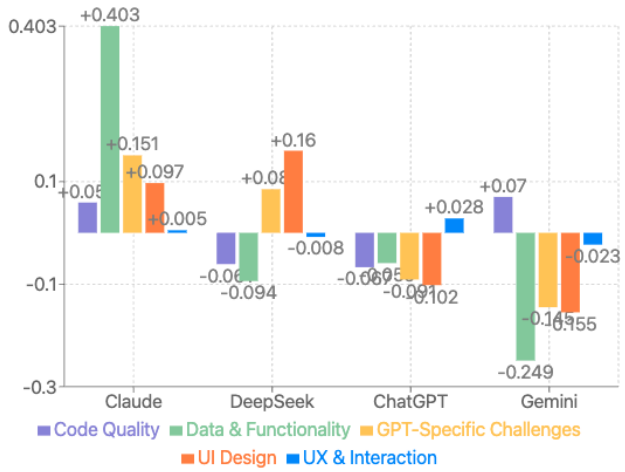

Figure 6. Normalized deviation from users mean ratings across five dimensions. Positive values indicate above-average user satisfaction in that dimension.

We visualized these user-driven results from two complementary perspectives. Figure 5 summarizes the total user satisfaction scores for each model, along with a line chart showing their relative performance percentage normalized against the best-performing model. Figure 6 illustrates the mean-centered deviations of each model's ratings from the category-wise average, highlighting model-specific strengths and weaknesses as perceived by users.

These results broadly mirror those of our expert rubric: Claude again achieved the highest aggregate score (8967), followed by DeepSeek, ChatGPT, and Gemini. However, the deviation plot reveals finer distinctions. For instance, Claude exhibits significant positive deviation in Data &

Functionality and GPT-Specific Challenges, reinforcing its strength in producing creative, modular outputs that go beyond templated layouts. DeepSeek receives high user marks in UI Design, reflecting its success in visual coherence and spacing. Meanwhile, Gemini, despite high animation fluency, underperforms in most categories, suggesting that aesthetic novelty alone does not guarantee user satisfaction.

Interestingly, ChatGPT's performance remains relatively neutral across all dimensions—consistent with its prior positioning as a model that favors structure and clarity over expressiveness. This confirms that while ChatGPT may lack innovation, its outputs are consistently perceived as usable and stable.

Putting all the results together, we find an important takeaway: clean and well-structured code does not always mean a better user experience. Users care more about things like clarity, responsiveness, and how easy the interface is to use—factors that don't always show up in traditional code quality checks. On the other hand, layouts that look flashy or creative ignores basic usability principles can actually make the experience worse, which is what we observed with Gemini.

This highlights the need to evaluate models from multiple angles. It's not enough to just check whether the code runs or follows technical rules. We also need to consider how real users react to the designs. By combining expert-based scoring with feedback from users, we get a more complete picture of how each model performs and when it's best to use each one.

FUTURE WORK

This work demonstrated how user-driven prompt design and an evaluation framework help reveal both the strengths and limitations of current large language models in frontend code generation. While these tools show promise, they also raise important questions and challenges that must be addressed through future research. Three key areas where deeper investigation is especially needed are identified: accessibility, design heuristics, and ethical considerations.

Accessibility is a critical concern that remains largely underexplored in the context of AI-generated user interfaces. Many interfaces produced by LLMs fail to meet the Web Content Accessibility Guidelines (WCAG), which are essential for ensuring that digital content can be used by people with disabilities. For example, generated code may lack appropriate semantic HTML elements, provide poor keyboard navigation, or omit alternative text for visual content. These gaps highlight the need for systematic evaluation of model outputs through an accessibility lens. Future work should explore how prompt engineering, fine-tuning, or post-processing techniques can be used to guide models toward producing more inclusive designs. There is also an opportunity to build benchmark datasets

and evaluation tools specifically for measuring accessibility in generative UI systems.

The second area is design heuristics, which refer to established best practices for usability and interface quality. While LLMs are capable of generating visually coherent layouts, it is unclear to what extent they understand or follow foundational principles such as consistency, feedback, error prevention, and user control. Research that compares AI-generated interfaces with well-known heuristic frameworks, such as Shneiderman's Eight Golden Rules of Interface Design [9], could provide insights into the "design intuition" of these models. Controlled user studies or expert evaluations can also help determine whether LLM-generated UIs improve or hinder user experience. Ultimately, this line of work can help bridge the gap between generative capability and user-centered design.

The third key area involves the ethical and social dimensions of using generative AI in UI development. As with other AI applications, LLM-generated code and design suggestions can embed social biases, reinforce stereotypes, or even inadvertently expose sensitive data. For example, if a model is trained on biased design examples or poorly anonymized datasets, it may reproduce harmful patterns without user awareness. Ensuring the ethical use of these tools will require robust frameworks for bias detection, transparency, and privacy preservation. It also calls for interdisciplinary collaboration between computer scientists, designers, and ethicists to define responsible usage guidelines and standards.

Addressing these research challenges is crucial for pushing the boundaries of what LLMs can do in frontend development. More importantly, it will ensure that the integration of AI into UI workflows is not only innovative but also inclusive, ethical, and aligned with real-world user needs.

ACKNOWLEDGMENTS

We thank all the staff and students of CS568: User-Centered Machine Learning at UIUC for their valuable feedback, collaborative spirit, and thought-provoking discussions throughout the Spring 2025 semester. We are especially grateful to Professor Ranjitha Kumar and teaching assistants Rizky Wellyanto, Ali Zaidi, and Matthew Weston for their guidance, support, and constructive commentary during class sessions, design reviews, and project development.

REFERENCES

[1] L. Cui, C. Liang, H. Zhang, X. Wang, Y. Wu, and Z. Liu. 2024. WebApp1K: A Practical Code-Generation Benchmark for Web App Development. arXiv preprint arXiv:2408.00019. Retrieved from https://arxiv.org/abs/2408.00019

[2] Gu, Heng (Eric), and Senthil Chandrasegaran. 2025. Synthetic Users: Insights from Designers' Interactions with Persona-Based Chatbots. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. Retrieved from https://doi.org/10.1017/S0890060424000283

[3] J. Liu, Y. Liu, Z. Xu, X. Xie, and W. Zhao. 2024. StructEval: Structure-Aware Evaluation for LLM-Generated Code. arXiv preprint arXiv:2404.03543. Retrieved from https://arxiv.org/abs/2404.03543

[4] K. Kumar, T. Ashraf, O. Thawakar, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, P. H. S. Torr, F. S. Khan, and S. Khan. 2025. LLM Post-Training: A Deep Dive into Reasoning Large Language Models. arXiv preprint arXiv:2502.21321. Retrieved from https://doi.org/10.48550/arXiv.2502.21321

[5] A. Madaan, P. Ammanamanchi, X. Jin, N. Madaan, A. Abid, P. Zhang, et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. NeurIPS 2023. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html

[6] R. Murray-Smith, J. Williamson, and R. Stein. 2024. Active Inference and Human–Computer Interaction. arXiv preprint arXiv:2412.14741. Retrieved from https://arxiv.org/abs/2412.14741

[7] Sarath Krishna Mandava. 2023. The role of AI in Frontend Development: From code suggestions to automated UI testing. International Journal of Enhanced Research in Management & Computer Applications, 12(08), 101–113. Retrieved from https://doi.org/10.55948/ijermca.2023.0819

[8] Prasadini Padmasiri, Pramukthika Kalutharage, Nethma Jayawardhane, and Chaminda Jagath Wickramaratne. 2023. AI-Driven User Experience Design: Exploring Innovations and Challenges in Delivering Tailored User Experiences. 2023 8th International Conference on Information Technology Research (ICITR), 1–6. Retrieved from https://doi.org/10.1109/icitr61062.2023.10382802

[9] B. Shneiderman and C. Plaisant. 2010. Designing the user interface: Strategies for effective human-computer interaction. 5th ed. Addison-Wesley Professional.

[10] Alex Xie and Yu Sun. 2024. Leveraging ChatGPT for Advanced Financial Analysis: A prompt pattern catalog. NLP & Information Retrieval, 79–87. Retrieved from https://doi.org/10.5121/csit.2024.140606

[11] W. Yan, H. Liu, Y. Wang, Y. Li, Q. Chen, W. Wang, T. Lin, W. Zhao, L. Zhu, H. Sundaram, and S. Deng. 2024. CodeScope: An Execution-based Multilingual Multitask Multidimensional Benchmark for Evaluating LLMs on Code Understanding and Generation. arXiv preprint arXiv:2311.08588. Retrieved from https://doi.org/10.48550/arXiv.2311.08588

[12] M. Yuan, J. Chen, Z. Xing, A. Quigley, Y. Luo, T. Luo, G. Mohammadi, Q. Lu, and L. Zhu. 2024. DesignRepair: Dual-Stream Design Guideline-Aware Frontend Repair with Large Language Models. arXiv preprint arXiv:2411.01606. Retrieved from https://doi.org/10.48550/arXiv.2411.01606

[13] K. Zhang, V. Kumar, and M. Zhang. 2025. Interactive Web: Leveraging AI-Driven Code Generation to Simplify Web Development Algorithms for Novice Programmers. International Journal of Computer Science & Information Technology, 17(2), 45–58. Retrieved from http://dx.doi.org/10.5121/csit.2024.150107

[14] R. Zhang, K. Zhang, S. Chen, Y. Song, and J. Yan. 2024. How Efficient is LLM-Generated Code? A Rigorous & High-Standard Benchmark. arXiv preprint arXiv:2406.06647. Retrieved from https://arxiv.org/abs/2406.06647

[15] W. Zhang, Z. Wang, D. Fried, Z. Zhou, and B. Yao. 2024. Design2Code: Benchmarking Multimodal Code Generation for Automated Front-End Engineering. arXiv preprint arXiv:2403.03163. Retrieved from https://arxiv.org/abs/2403.03163

A Appendix

A.1 Prompt Design

```
You are an expert front-end developer with deep knowledge of UI design, code style consistency, component library integration, and animation principles. Your task is to transform an existing low-quality codebase into a polished, professional, responsive e-commerce website.
1. Apply professional UI/UX design patterns to improve visual consistency, spacing, alignment, typography, color usage, and responsiveness across devices.
2. Ensure all user-facing components have intuitive interactivity with clear affordances and accessible behavior. Prioritize smooth state changes and edge case handling.
3. You may choose any modern charting or UI component library that best fits professional-grade requirements and is optimized for responsiveness and maintainability.
4. Maximize modularity, maintainability, and clarity through consistent naming, reusable components, minimal inline styles, and in-code documentation of complex logic—without modifying app-level files.
5. Current project structure:
C:.
| • App.css
| • App.js
| • App.test.js
| • index.css
| • index.js
| • logo.svg
| • reportWebVitals.js
| • setupTests.js
|•••
├──components
| • • • HomePage.js
| • • • ItemDetailPage.js
| • • • LoginPage.js
| • • • PriceTrackingPage.js
| • • • SignupPage.js
| • • • UserSettingsPage.js
|
├──data
| • • • itemData.json
|
├──utils
| • • fakeUserService.js

6. Return only the modified or new code files in full and ensure your code follows consistent style conventions for future maintainability.
7. Features to include: (sample)
8. Current file content: (to enter)
Respond should include the file names followed by their content in full, and followed by any external library used.
```

Figure 6. The General Prompt We Used for This Project.

```
1. Create a page style for a single item detail on a shopping website.
2. Put all the pictures together and show only one at a time. Users can view different pictures by clicking the arrows on the left and right sides.
3. Display several small dots arranged horizontally below the picture to provide users with a reference of "which page of pictures they are on now."
4. Allow users to use the magnifier function. The magnifier icon appears in the lower right corner of the picture element. When the user clicks the magnifier icon, the user's cursor can be used as a magnifier, providing the user with a square magnifier perspective.
5. Add some animations appropriately to make the user interaction smoother and more responsive.
```

Figure 7. Point 7 for Item Detail Page.

```
1. Create animation for each item.
2. Make the page with a user-friendly style.
3. It should have all the features that are on a shopping website's main page.
4. Each item should have animation when the mouse hovers on it.
5. Add more styles that are suitable for the shopping website's home page.
```

Figure 8. Point 7 for Home Page.

```
1. Build an interactive price history chart with toggleable time periods and visualization options
2. Include a product information panel showing ratings, current price, original MSRP
3. Display detailed price statistics, including current, average, lowest, and highest prices.
4. Enable users to toggle between line chart, bar chart, and candlestick view to better understand monthly price volatility and trend direction.
```

Figure 9. Point 7 for Price Tracking Page.

A.2 Results Screenshots

A.2.1 Item Detail Page

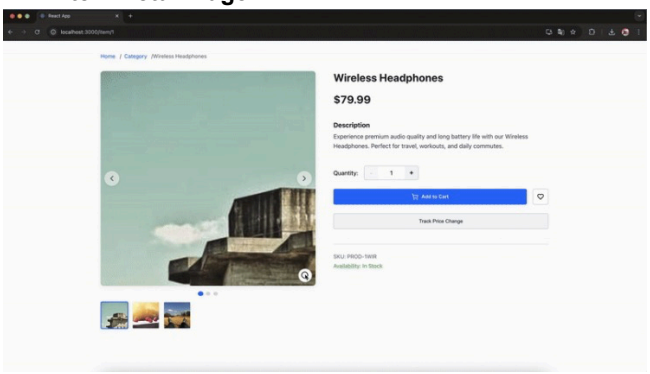

Figure 10. Claude-Generated Item Detail Page.

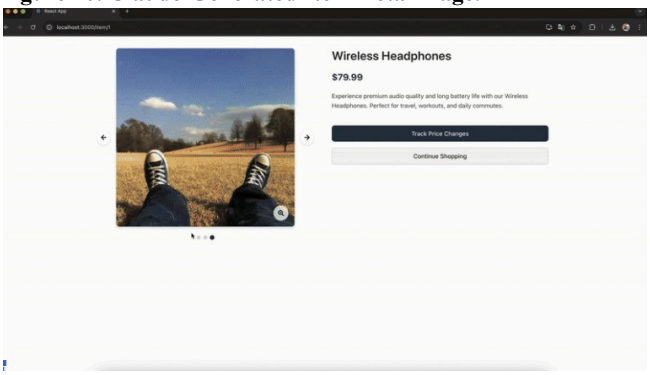

Figure 11. DeepSeek-Generated Item Detail Page.

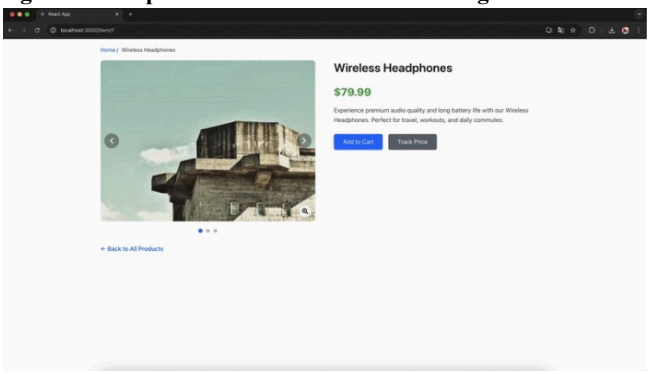

Figure 12. Gemini-Generated Item Detail Page.

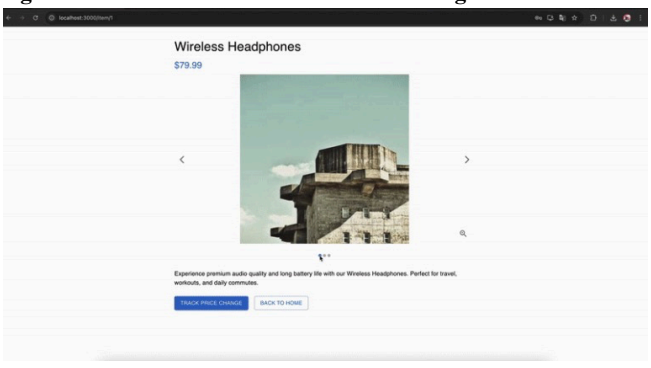

Figure 13. ChatGPT-Generated Item Detail Page.

A.2.2 Home Page

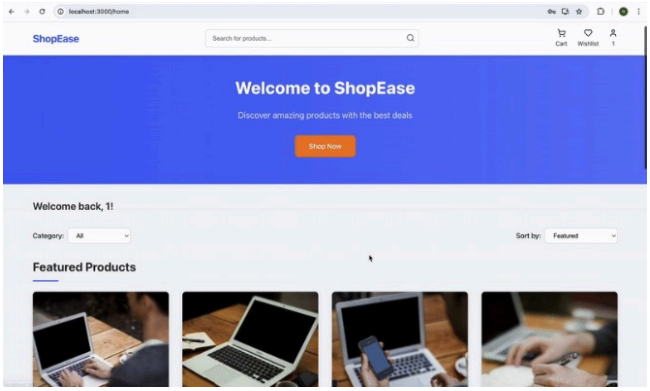

Figure 14. Claude-Generated Home Page.

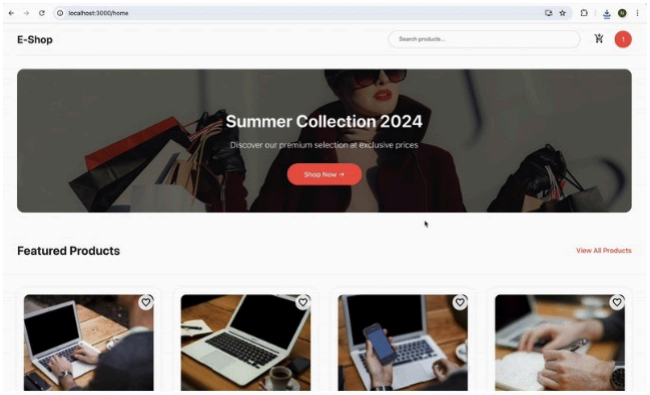

Figure 15. DeepSeek-Generated Home Page.

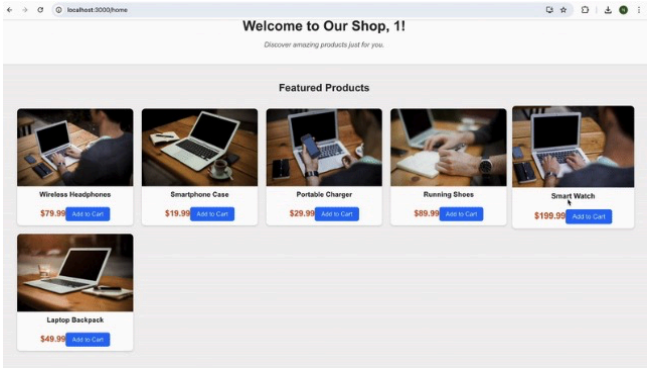

Figure 16. Gemini-Generated Home Page.

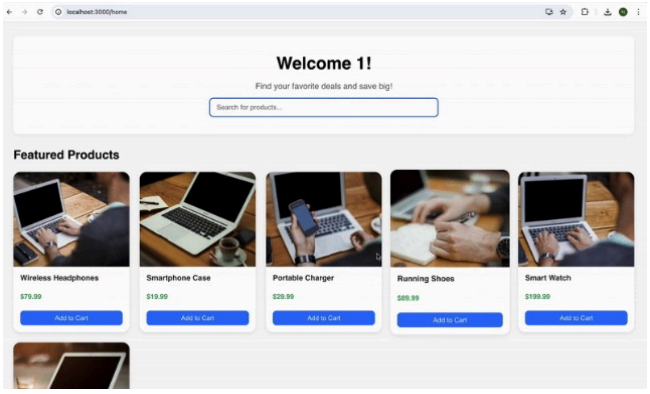

Figure 17. ChatGPT-Generated Home Page.

A.2.3 Price Tracking Page

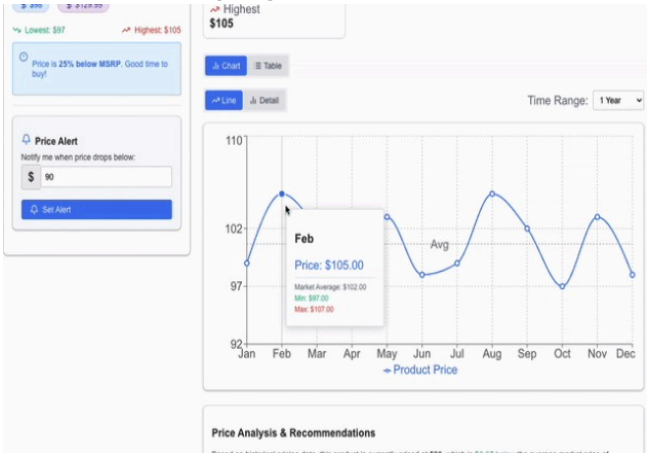

Figure 18. Claude-Generated Price Tracking Page.

Figure 19. DeepSeek-Generated Price Tracking Page.

Figure 20. Gemini-Generated Price Tracking Page.

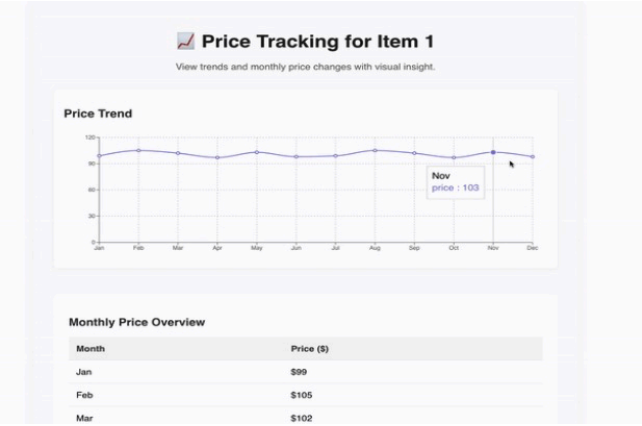

Figure 21. ChatGPT-Generated Price Tracking Page.