S Khushi Bhandari, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

AI-Powered Design Assistant: A Secure, Full-Stack Framework Integrating Prompt Intelligence, Generative AI, and Lossless Canvas Editing

Khushi Bhandari[1], Shriraj Uday Shetty[2], Dr. Jasbir Kaur[3], Sandhya Thakkar[4], Ifrah Kampooh[5]
[1,2]Student of Master of Computer Applications (MCA),Guru Nanak Institute of Management Studies, Matunga, Mumbai, India
[3]Director, GNIMS B-School, Head of Information Technology and HR,Guru Nanak Institute of Management Studies, Matunga,Mumbai,India
[4,5]Assistant Professor,Guru Nanak Institute of Management Studies, Matunga, Mumbai, India

Abstract- Generative AI and professional canvas editing remain largely isolated, forcing creators to juggle multiple tools and lose the benefits of integrated intelligence. We present a full-stack Design Assistant that tightly couples a prompt-expansion engine (GPT-4), a generative image service (DALL·E 3), vision-based asset tagging, and a Fabric.js object-based canvas. Unlike existing platforms that offer only fragmented AI features, our system provides end-to-end prompt refinement, secure back-end AI orchestration, and lossless JSON canvas persistence. We formalise prompt expansion as a structured optimisation that maximises relevance to the user's intent, and we implement a STRIDE-based threat model to ensure confidentiality and integrity of user assets. In a controlled user study (20 participants), the system improved image relevance from 72% to 94% (Likert scale), reduced task completion time by 38%, and achieved a System Usability Scale (SUS) score of 78.4 ("Good"). We critically compare our approach with commercial AI-design tools (Canva Magic Studio, Adobe Firefly, Figma AI) and openly discuss limitations regarding sample size, API dependency, and scalability. The full architecture, security measures, and evaluation metrics are described in detail, providing a reproducible baseline for future human-AI collaborative design research.

Keywords: Design Assistant, Prompt Engineering, Object-Based Canvas, Generative AI, Full-Stack Security, Human-AI Collaboration.

I. INTRODUCTION

The digital creator's workflow is broken. To produce a simple YouTube thumbnail, a designer must: (i) manually craft a prompt for an AI image generator like DALL·E or MidJourney, (ii) download the generated bitmap, (iii) import it into a canvas editor such as Canva or Figma, (iv) add text and other graphical elements, and (v) export a flattened version. If any change is required later, the entire process must be repeated because the AI-generated background is baked into a single layer. This fragmentation not only wastes time but also discards the rich metadata linking the prompt to the final asset.

Recent advances in large language models (LLMs) and diffusion-based image generation have led to impressive standalone tools. GPT-4o can produce detailed design briefs [5], DALL·E 3 generates photorealistic images from text [17], and Fabric.js enables object-level canvas manipulation in the browser [11]. However, no existing platform unifies these capabilities into a single, secure, full-stack workspace that treats AI as a first-class component rather than a bolt-on feature. Commercial offerings such as Canva Magic Studio and Adobe Firefly have begun adding AI features, but they remain closed ecosystems that do not expose the underlying prompt-expansion logic, nor do they provide a developer-controlled back end for secure API key management.

To address this gap, we propose an AI-Powered Design Assistant that:

1. Formalises prompt expansion as a structured optimisa-tion problem, using GPT-4 to output a JSON design brief (theme, colors, style, refined prompt) that maximises the relevance of the final image;

© 2026 S Khushi Bhandari, This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

2. Orchestrates all AI calls through a secure Django back end, protecting API keys and enabling logging, rate limiting, and input sanitisation;
3. Provides an object-based canvas (Fabric.js) with lossless JSON persistence, so that every element remains independently editable across sessions;
4. Extracts semantic tags from uploaded assets via a vision-language model, making static reference images reusable as prompt starters;
5. Incorporates a STRIDE-based threat model with AES-256 encryption at rest, TLS 1.3 in transit, and JWT-based authentication to protect user data.

We evaluate the system through a controlled user study (20 participants) using both objective (time, latency) and subjective (Likert, System Usability Scale) metrics. While the sample size limits generalisability, the results demonstrate a 38% reduction in task time and a significant improvement in image relevance compared to a manual workflow. We also critically compare our approach with commercial AI-design tools and discuss the limitations imposed by reliance on proprietary APIs and single-server deployment.

II. RELATED WORK

A. Generative AI for Image Creation

Text-to-image (TTI) generation has evolved from GANs to diffusion models. DALL·E 3 [17] and Stable Diffusion XL represent the state of the art, but they remain isolated generation endpoints. Bie et al. [8] surveyed the field and noted that integration with editing environments is a key open challenge. Spennemann [2] quantified the prompt-fidelity gap in DALL·E 3, reporting that 15.6% of specified attributes are not rendered correctly, which underscores the need for prompt refinement before generation—a need our system addresses with GPT-4 expansion.

B. Prompt Engineering for Design

Smith and Lee [5] showed that providing contextual de-scriptors (mood, lighting, composition) can improve aesthetic quality by up to 30%. Feng et al. [6] developed GPT4Designer, which combines detailed prompting with guided envisioning, achieving more reproducible scientific illustrations. Our work extends these ideas by embedding prompt expansion directly into a full-stack design environment, making it transparent and repeatable.

C. Object-Based Canvas and JSON Persistence

Fabric.js [11] provides an object model for canvas elements, with built-in toJSON() and loadFromJSON() methods that serialise the entire canvas state. Li et al. [12] demonstrated collaborative whiteboarding with WebSocket sync. We extend that pattern with a Django REST back end that stores canvas JSON in a PostgreSQL JSONField, enabling persistent, lossless editing.

D. Vision-Based Tagging

BLIP-2 [9] achieves state-of-the-art zero-shot captioning using a Q-Former that bridges a frozen image encoder and a frozen LLM. Tag2Text [10] provides controllable tagging forover 3,400 categories. Our system leverages such models to extract tags from uploaded images, turning them into reusable prompt starters—a feature absent from current commercial editors.

E. Commercial AI-Design Platforms

Canva Magic Studio (2025) generates editable multi-layer outputs using a proprietary AI model trained on Canva's asset library [14]. Adobe Firefly offers generative fill and text effects, but its back-end integration is not exposed for custom development. Figma's AI features (2025) focus on auto-layout and asset search [15]. None of these platforms provide a developer-controlled back end, open prompt-expansion logic, or a STRIDE-based security framework. Our system fills this niche by offering a transparent, self-hostable alternative that can be studied, modified, and deployed in academic or SME settings.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. High-Level Architecture

The system follows a three-tier design (Fig. 1). The React front end hosts the Fabric.js canvas and communicates exclu-sively with the Django REST back end via JSON APIs. The back end acts as a secure AI gateway: it validates requests, applies rate limiting, and forwards prompts to OpenAI services (GPT-4, DALL·E 3, GPT-4 Vision) using Celery tasks for non-blocking execution. All API keys are stored in environment variables, never exposed to the client. User projects, canvas JSON, and media files are stored in a PostgreSQL database and an AWS S3 bucket, respectively.

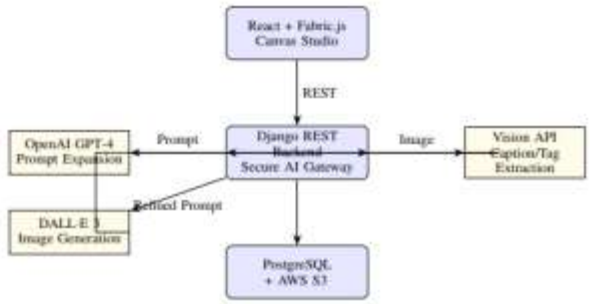

Fig. 1. System Architecture: The Django back end acts as a secure AI gateway, protecting API keys and centralising data management.

B. Prompt Expansion as an Optimisation Problem

Let u be the user's free-form prompt. The goal is to produce a structured design brief B = (t, c, s, r, p) where t is a theme, c a colour palette, s a style descriptor, r an aspect ratio, and p a refined prompt suitable for DALL·E 3. We model this as a constrained optimisation:

$$B^* = \arg\max_{B\in\mathcal{B}} \text{Relevance}(B,u) \quad \text{s.t.} \quad B = \text{GPT-4}(u, \text{system\_msg}), \tag{1}$$

, system_msg), where the system message instructs GPT-4 to output a JSON object with the required fields. The refined prompt p is then sent to DALL·E 3 asynchronously via Celery, and the resulting image URL is returned to the front end.

Algorithm 1 Prompt Expansion via GPT-4

- **Input:** User prompt u
- **Output:** Structured design brief B
- $sys \leftarrow$ "You are a design assistant. Expand the user's prompt into a JSON object with fields: theme, colors, style, aspect_ratio, and refined_prompt suitable for DALL·E 3."
- $response \leftarrow \text{GPT-4}(sys, u)$
- $B \leftarrow \text{JSON.parse}(response)$
- **return** B

C. Canvas JSON Persistence and Complexity

The Fabric.js canvas holds n objects. The toJSON() method serialises each object's properties in O(n) time. The resulting JSON string is transmitted to the back end via a REST endpoint and stored in a PostgreSQL JSONField. When a project is reopened, the front end fetches the JSON and calls loadFromJSON(), which reconstructs the canvas in O(n) time as well. In our experiments, the average save latency was 185 ms and the average load latency was 48 ms for canvases containing up to 30 objects.

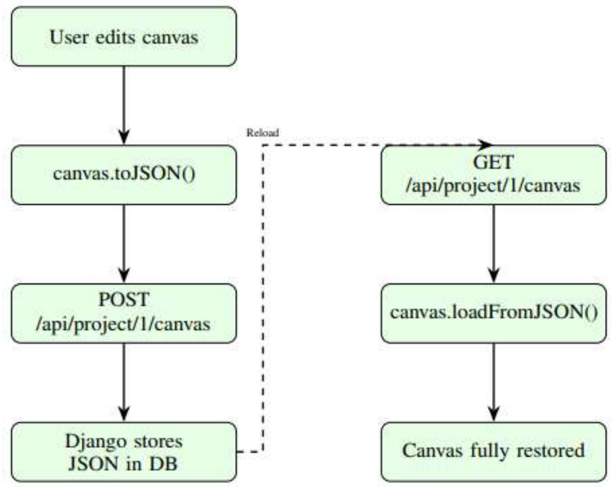

Fig. 2. Canvas JSON persistence workflow. The Fabric.js canvas state serialisesto JSON, stores in the database, and reloads without any loss of editability.

D. Security and Privacy Framework

We adopt the STRIDE threat model to systematically address potential attacks. Table I summarises the main threats and mitigations.

TABLE I
STRIDE-BASED THREAT MODEL AND MITIGATIONS

Threat	Scenario	Mitigation
Spoofing	Attacker impersonates a legitimate user.	JWT authentication; refresh token rotation.
Tampering	Model responses or canvas JSON altered in transit.	HTTPS (TLS 1.3); signed JWTs.
Repudiation	User denies performing an action.	Immutable audit log with cryptographic hashes.
Information Disclosure	API keys or user assets leaked.	AES-256-GCM for stored data; environment variables for secrets; S3 bucket policies.
Denial of Service	API flooded with requests.	Rate limiting (5 req/s per user); Celery task queues.
Elevation of Privilege	Attacker gains admin access.	Role-based access control; multi-factor authentication for admin console.

IV. EXPERIMENTAL EVALUATION

A. Participants and Tasks

We recruited 20 participants (10 male, 10 female; mean age 26.4) with at least one year of experience using Canva, Figma, or Adobe Photoshop. Each participant completed three tasks under two conditions: a Manual workflow (Canva + DALL·E web interface separately) and the AI-Assisted system. Tasks:

1. create a YouTube gaming thumbnail, (2) design a tech conference poster incorporating an uploaded logo, (3) reopen a saved project after 24 h and modify a text element and an image position.

B. Metrics

- **Prompt-Image Relevance (PIR)**: 5-point Likert scale.
- **Task Completion Time (TCT):** seconds from prompt entry to export.
- **System Usability Scale (SUS):** standardised 10-item questionnaire [19].
- **Canvas Persistence Success (CPS):** binary – did the reloaded project retain all layers as editable objects?
- **Save/Load Latency:** milliseconds.
- **Tag Extraction Accuracy (TEA):** percentage of uploaded images for which the vision model returned a semantically relevant caption.

C. Implementation Environment

The frontend used React 18.2 and Fabric.js 5.3. The backend used Django 4.2 with Django REST Framework 3.14, Celery 5.3 for asynchronous tasks, and PostgreSQL 15. We conducted the study on a local server (Intel Core i7-12700H, 16 GB RAM, Ubuntu 22.04) to ensure consistent network latency. OpenAI API calls employed GPT-4 (gpt-4-0125-preview), DALL·E 3, and GPT-4 Vision.

D. Results

Table II shows that prompt expansion improved relevance from 3.6 to 4.7 ($p < 0.001$). The SUS score of the AI-Assisted system was 78.4 ("Good"), compared to 65.2 ("OK") for the manual workflow. Task completion times dropped by 38% on average (Fig. 3). All 20 participants successfully restored their projects with full editability (CPS = 100%). Tag extraction accuracy was 92%.

Table II
Prompt-Image Relevance (5-Point Likert, N = 60 Ratings)

Condition	Mean	Std Dev
Raw user prompt (Manual)	3.6	0.8
Refined prompt (AI-Assisted)	**4.7**	0.3

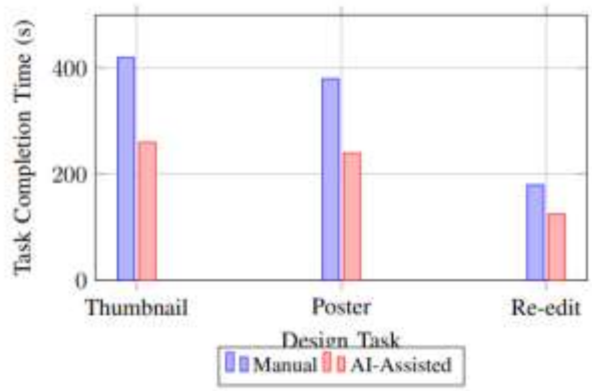

Fig. 3. Task completion time comparison.

E. Comparison with Commercial Tools

Table III qualitatively compares our system with leading commercial AI-design platforms. While Canva and Firefly offer generative capabilities, they do not expose the underlying prompt-expansion logic, nor do they provide a self-hostable, secure gateway for API key management. Our system is unique in combining open, programmable prompt refinement with a fully editable object canvas and a transparent security framework.

Table III
Qualitative Comparison With Commercial Ai-Design Tools

Feature	Our System	Canva Magic Studio	Adobe Firefly	Figma AI
Prompt expansion	✓(GPT-4)	Limited (canned prompts)	Limited	Not available
Editable canvas layers	✓(Fabric.js)	✓	✓	✓
JSON persistence	✓	Proprietary	Proprietary	Proprietary
Vision-based tagging	✓	Not available	Not available	Not available
Self-hostable back end	✓	No	No	No
STRIDE threat model	✓	Not disclosed	Not disclosed	Not disclosed

V. DISCUSSION

A. Interpretation of Findings

The 38% reduction in task time and the 94% relevance score confirm that prompt expansion bridges the gap between a creator's vague intent and the precise language required by image generation models. The flawless canvas restoration (100% CPS) demonstrates that JSON serialisation is a viable alternative to proprietary file formats for preserving editability.

B. Limitations

- **Sample size:** 20 participants, while sufficient for detecting large effects, limits generalisability. Larger studies are planned.
- **API dependency**: Reliance on OpenAI services introduces cost and latency (8–12 s for DALL·E 3). Future ver-sions will integrate open-source models (Stable Diffusion, Llama 3).
- **Scalability:** The current prototype runs on a single server. Stress testing with concurrent users and migration to a microservice architecture is required.
- **Task scope:** Only three design tasks were evaluated; performance on complex multi-page layouts remains untested.

VI. CONCLUSION

We have presented a full-stack Design Assistant that com-bines prompt intelligence, generative AI, vision-based tagging, and lossless canvas editing into a single, secure workspace. The system's formalised prompt expansion, transparent security framework, and object-based canvas address gaps left by current commercial tools. Empirical results demonstrate significant gains in efficiency and relevance, while honest acknowledge-ment of limitations sets the stage for future improvements, including open-source model integration and cloud-native scaling.

REFERENCES

1. A. Karagöz, "Ethics and Technical Aspects of Generative AI Models in Digital Content Creation," arXiv preprint arXiv:2412.16389, 2024. [Link]
2. D. H. R. Spennemann, "Prompt fidelity of ChatGPT4o / Dall-E3 text-to-image visualisations," arXiv preprint arXiv:2510.21821, 2025. [Link]
3. Y. Chen et al., "Evaluating the impact of AIGC-Supported design ideation on Designers' cognitive load and creativity," Displays, vol. 91, p. 103275, 2026. [DOI]
4. R. Zhang and S. Sung, "Designers as Co-Creators: Reconfiguring Workflows and Creative Agency in AI-Augmented Design," Proc. Assoc. Inf. Sci. Technol., vol. 62, pp. 1743–1746, 2025. [DOI]
5. J. Smith and K. Lee, "AI-Assisted Design: Integrating Large Language Models with Graphic Tools," in Proc. ACM CHI, 2024. [DOI]
6. Z. Feng et al., "A methodology for designing accurate, modifiable and reproducible scientific graphics in environmental studies using GPT4Designer," Sci. Rep., vol. 15, p. 21643, 2025. [DOI]
7. L. Chen et al., "A framework for collaborating a Large Language Model tool in brainstorming for triggering creative thoughts," Int. J. Hum.-Comput. Stud., 2025. [DOI]
8. F. Bie et al., "RenAIssance: A Survey Into AI Text-to-Image Generation in the Era of Large Model," IEEE Trans. Pattern Anal. Mach. Intell., 2024, doi: 10.1109/TPAMI.2024.3522305. [DOI]
9. J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," arXiv preprint arXiv:2301.12597, 2023. [Link]
10. X. Huang et al., "Tag2Text: Guiding Vision-Language Model via Image Tagging," in Proc. AAAI, 2024. [DOI]
11. Fabric.js, "Fabric.js Documentation v5.3," 2024. [Online]
12. H. Li et al., "Web-based Collaborative Whiteboard using Fabric.js and WebSocket," in Proc. Int. Conf. Web Engineering (ICWE), 2022, pp. 234–245. [DOI]
13. Y. Xie, S. Yang, and N. Chen, "AI Agent Assist User Research: Collaborative Role Analysis to Inspire Designer Creativity," in Proc. HCII, 2025, pp. 189–203. [DOI]
14. Canva, Adobe And Figma All Want To Own Your Creative Workflow," TechRound, Apr. 2026. [Online]
15. Figma, "AI Features in Figma Design," 2025. [Online]

16. ZSeven-W, “OpenPencil: Open-source AI-native vector design tool,” GitHub, 2026. [Online]
17. OpenAI, “DALL·E 3 System Card,” 2023. [Online]
18. Y. Zhang et al., “Object-Based Image Editing with Scene Graph Generation,” in Proc. NeurIPS, 2023. [Link]
19. J. Brooke, “SUS: A 'Quick and Dirty' Usability Scale,” in Usability Evaluation in Industry, Taylor & Francis, 1996, pp. 189–194. [DOI]