

RESEARCH ARTICLE

From Prompt to Polygon: Evaluating End-to-End Human–AI Workflows for Web-Based Generative 3D Authoring

Ray Fadly Hadi¹ * , Randi Proska Sandra² , Sri Rejeki³ 

¹Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang, Padang, **Indonesia**

²Faculty of Engineering, Universitas Negeri Padang, Padang, **Indonesia**

³Department of Informatics Engineering, Telkom University Purwokerto, Purwokerto, Central Java, **Indonesia**

✉ *Corresponding Author: 22343029@student.unp.ac.id

This article
contributes to:



ABSTRACT

Prompt- and image-driven 3D generation can lower the interaction barrier to digital asset creation, yet model-level advances do not establish whether non-specialists can operate a complete authoring workflow. This study designed and evaluated a web-based human–AI prototype integrating a React interface, FastAPI services, ComfyUI workflow orchestration, asynchronous job processing, persistent asset storage, and browser-based 3D visualization. We implemented four modes: Text-to-3D, Text-to-3D with ControlNet, Single Image-to-3D, and Multi Image-to-3D. Evaluation combined 14 requirement-derived black-box scenarios, cross-viewer GLB compatibility checks in three viewing environments, 35 completed generation jobs, task-based observation with seven participants completing 84 task executions, System Usability Scale (SUS) responses from a separate group of 12 students, and one expert assessment of implementation and output quality. All 14 functional scenarios produced the expected outcomes, and the tested GLB outputs opened in the integrated viewer, Blender, and 3dviewer.net. Mean server-side turnaround was 66.15 s; an exploratory Kruskal–Wallis test indicated differences among the observed workload distributions ($H(3) = 10.52$, $p = .015$, $\epsilon^2 = .243$), but did not support causal or pairwise speed claims. All observed tasks were eventually completed, but only 61 of 84 executions (72.62%) were independent; Tasks 5 (enter Studio) and 10 (find the generated model in My Projects) each required assistance for five of seven participants. Mean SUS was 81.67 (SD = 9.25; 95% CI [75.79, 87.55]), while generated-model quality received a diagnostic expert score of 3.50/5. The study contributes an end-to-end human–AI authoring architecture and a multidimensional evaluation logic that separates functional conformance, deployment turnaround, cross-viewer compatibility, observed interaction, perceived usability, and output quality. Findings support the feasibility of accessible web-based generative 3D authoring while identifying discoverability and geometric-quality constraints.

KEYWORDS

Generative 3D; human–AI interaction; interactive media; multimodal workflow; System Usability Scale; web-based 3D authoring

🕒 Received: Aug. 21, 2026; Revised: Sep. 18, 2026; Accepted: Sep. 28, 2026; Published Online: Oct. 31, 2026

How to Cite: Hadi, R. F., Sandra, R. P., & Rejeki, S. (2026). From prompt to polygon: Evaluating end-to-end human–AI workflows for web-based generative 3D authoring. *Journal of Hypermedia & Technology-Enhanced Learning*, 4(3), 202–223. <https://doi.org/10.58536/j-hytel.252>



1. INTRODUCTION

Generative artificial intelligence is moving 3D content creation beyond specialist modeling toward mixed-initiative authoring, in which users specify their intent and an AI pipeline generates a 3D asset. This shift is relevant to interactive media, visualization, prototyping, immersive environments, and technology-enhanced learning because it can reduce the procedural burden of mesh construction. A human-centered approach, however, requires more than successful generation: users must understand what the system can do, provide effective inputs, monitor uncertain processing, and recover usable outputs. Human-centered AI therefore emphasizes reliable and trustworthy operation [1], clear feedback and user control [2], and explicit support for understanding AI capabilities and limitations [3].

At the model level, denoising diffusion [4] and latent diffusion [5] established key synthesis foundations, while vision–language representation learning enabled semantic alignment between text and images [6]. Text-to-3D research then used 2D priors, including DreamFusion [7], Magic3D [8], and ProlificDreamer [9], which respectively emphasize diffusion-prior optimization, coarse-to-fine mesh refinement, and variational score distillation. VP3D further demonstrates how a 2D visual prompt can strengthen text-to-3D synthesis and address visual fidelity and cross-view consistency [10]. These studies advance generative quality, but their principal unit of analysis remains the generative method rather than the complete experience of non-specialist users operating an end-to-end authoring service.

Image-conditioned reconstruction offers a complementary route from one or more images to 3D representations. Zero-1-to-3 [11] synthesizes novel views from one image; pixelNeRF [12] predicts radiance fields from one or a few images; PIFu [13] uses pixel-aligned implicit functions for clothed-human digitization; and 3D-R2N2 [14] supports single- and multi-view reconstruction. Wonder3D [15] targets cross-view consistency in single-image 3D generation, whereas RichDreamer [16] uses normal-depth diffusion priors to enrich text-to-3D geometry. Yet benchmark-level improvements do not automatically resolve application concerns such as input guidance, long-running jobs, failure visibility, output storage, compatibility, or interface discoverability.

Recent user-oriented 3D authoring systems make this application-layer perspective increasingly important. Codeless3D evaluates a low-code 3D game-generation environment intended to reduce dependence on programming expertise [17], while MagicCraft integrates natural-language-driven generation with an interface for refining interactive 3D objects [18]. These systems reflect a shift from isolated generation models toward accessible authoring workflows. The present study differs by evaluating one web-based multi-stage 3D pipeline through multiple non-equivalent evidence streams—functional conformance, deployment turnaround, cross-viewer GLB compatibility, task observation, SUS, and expert quality assessment—rather than treating any one technical or usability measure as a proxy for overall system quality.

The potential educational relevance of accessible 3D authoring lies in its capacity to support rapid prototyping and visualization in design, computing, engineering, augmented and virtual reality, and interactive media activities. Students could use manipulable 3D artifacts to represent and explore concepts, while educators could prepare illustrative assets without relying on specialist modeling workflows. However, these applications represent plausible technology-enhanced learning use cases rather than educational outcomes demonstrated by the present study. Because learning gains, teaching

effectiveness, and classroom transfer were not measured, no claims regarding educational effectiveness are made.

Realizing this potential requires more than a capable generation model. A usable generative service must coordinate heterogeneous processing stages while communicating their status clearly to users. Such a service requires a browser-based interface, workflow selection, request validation, asynchronous execution, progress monitoring, persistent asset management, and interactive inspection of generated models. ComfyUI provides a node-based environment for composing generative workflows, while ComfyBench evaluates agents that design ComfyUI workflows [19]. Nevertheless, workflow automation and model-level benchmarks provide limited evidence about the experience of non-specialist users interacting with a deployed, multi-stage 3D authoring system.

Rather than proposing a new generative model, this study addresses the application-layer gap through a case study of system design and evaluation. Its practical objective is to make prompt-based and image-based 3D authoring accessible through a unified web workflow. Its research objective is to distinguish evidence of technical conformance from evidence of user experience and output quality. The prototype is therefore examined as an interactive authoring environment whose value depends on both computational orchestration and meaningful human use. All claims are limited to the evaluated prototype, participants, generation jobs, test scenarios, and viewing environments.

Three research questions guide the study. RQ1 examines whether the prototype supports prompt-based and image-based 3D authoring as a functional web workflow and whether its GLB outputs can be opened successfully across the tested viewing environments. RQ2 examines how server-side turnaround varies among the four generation modes under the observed deployment workload. RQ3 examines what task completion, assistance, error patterns, SUS responses, and expert assessment reveal about usability, interface discoverability, and generated-model quality. The study makes three contributions: an application architecture that connects multi-stage generation with asynchronous web interaction; a multidimensional evaluation framework that distinguishes functional conformance, turnaround, compatibility, observed interaction, perceived usability, and output quality; and empirical evidence of the difference between favorable overall usability ratings and the assistance required during specific tasks.

2. MATERIALS AND METHODS

2.1 Research Design, Questions, and Evaluation Logic

The study followed a software design-and-evaluation approach based on the Prototype Model of the software development life cycle [20]. Iterative prototyping was appropriate because the application joins computationally intensive generation, asynchronous job control, sequential workflow stages, and user-facing feedback. Development followed communication, quick planning, rapid design, prototype construction, and deployment-feedback cycles. The research evaluated the integrated application rather than training or benchmarking a new generative model.

The unit of evaluation differed across the technical measures. Functional evidence came from requirement-derived scenarios, compatibility evidence from opening exported files in independent viewing environments, and performance evidence from completed jobs. Human-centered evidence came from direct task observation, SUS responses, and one expert assessment. Separating these forms of evidence prevents a functional pass rate or favorable SUS score from being interpreted as proof of geometric fidelity, production reliability, or learning impact.

2.2 System Architecture and Development Process

The prototype uses a client–server architecture with asynchronous processing. The React frontend collects text or image inputs, displays job state, and renders generated 3D assets, while the FastAPI backend validates requests, selects a workflow, injects parameters, and coordinates result delivery. ComfyUI executes the selected generative graph, PostgreSQL stores application and job metadata, and ImageKit stores generated assets. [Figure 1](#) maps the user-facing path from input and job submission to workflow execution, persisted output, and browser-based inspection.

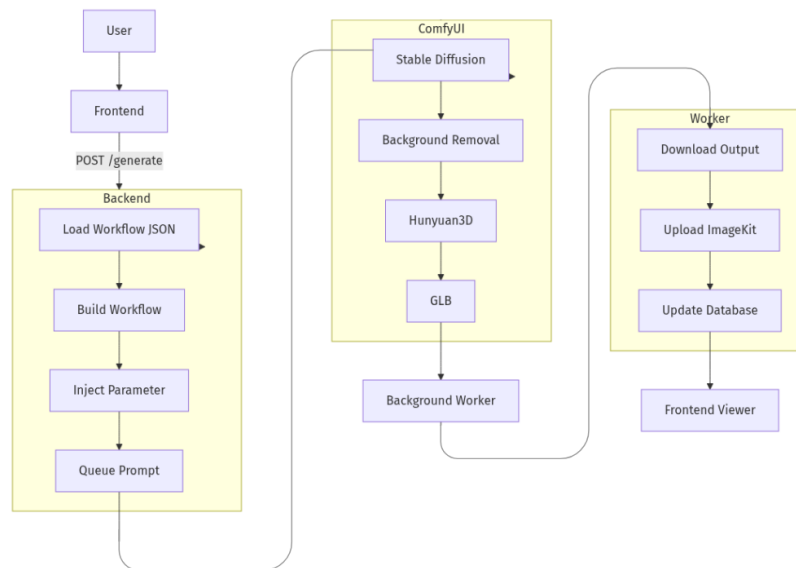


Figure 1. Web-based architecture and multi-stage human–AI generation workflow.

Evaluation drew on six evidence sources. Fourteen black-box scenarios examined expected input–output behavior, and compatibility checks opened exported GLB files in the integrated viewer, Blender, and 3dviewer.net. Performance analysis used 35 completed jobs: 20 Text-to-3D jobs and five jobs for each of the ControlNet, Single Image-to-3D, and Multi Image-to-3D modes. Task-based testing involved seven convenience-sampled students, five from Visual Communication Design, one from Civil Engineering, and one from Informatics, who each completed 12 tasks, yielding 84 task executions; SUS was completed by 12 respondents.

2.3 System, Software, Models, and Instruments

The implementation used React with Vite for the frontend; Python and FastAPI for backend services; Clerk for authentication; ComfyUI, PyTorch, and Diffusers for generative execution; Stable Diffusion for text-to-image synthesis; foreground removal before reconstruction; Hunyuan3D for image-to-3D generation; PostgreSQL with SQLAlchemy for application data; ImageKit for asset storage; and React Three Fiber, Drei, and Three.js for GLB visualization. ControlNet supplied an additional spatial-conditioning mechanism in the corresponding mode [21]. The broader technical context includes implicit occupancy representations [22], continuous signed-distance functions [23], differentiable volumetric rendering [24], sparse-input radiance fields [25], neural mesh rendering [26], and multiresolution hash encodings [27]. These works motivate component choices but are not treated as models newly proposed or retrained in this study.

The retained research record identifies the software families and deployed functions but does not preserve exact package versions, GPU model, CUDA version, random seeds, or dependency lock files. These omissions limit environment-level reproducibility and prevent hardware-normalized performance comparison. Turnaround is therefore reported only for the observed deployment; the article does not claim portable inference speed or exact computational reproducibility.

The expert validator was selected for subject-matter expertise in artificial intelligence, machine learning, and 3D animation. System implementation was assessed with 20 five-point Likert items (1 = strongly disagree to 5 = strongly agree) covering function and flow, input and validation, job management and status, AI workflow integration, multi-stage processing, prompt enhancement, output, visualization, viewer interaction, download, history, gallery, authentication, user-data separation, credits, interface clarity, and overall feasibility. Generated-model quality was assessed separately with 10 five-point items covering geometry, structural completeness, shape accuracy, detail, surface quality, multiview consistency, visual conformity, prompt/image correspondence, 3D-asset suitability, and GLB-output suitability. Appendix Table D1 reports system-implementation scores, and Appendix Table D2 reports generated-model quality scores. One validator completed both instruments; consequently, inter-rater reliability cannot be estimated and the results are diagnostic rather than a validated population-level quality estimate.

2.4 Multi-Stage Generative and Interaction Pipeline

The main Text-to-3D path converts a prompt into a 2D image, removes its background, reconstructs a 3D mesh, and exports the result as GLB. The backend selects a reusable workflow template according to the requested mode and injects prompt, negative prompt, seed, sampling steps, CFG scale, and image inputs where applicable. Each request is represented as an asynchronous job. After the workflow is sent to ComfyUI, the returned `prompt_id` links the ComfyUI execution to the application job record; the browser polls job state until completion or failure. On completion, the backend retrieves the GLB and thumbnail, uploads them to ImageKit, records persistent asset URLs in PostgreSQL, and makes the result available for interactive viewing and download.

Single Image-to-3D and Multi Image-to-3D accept user-provided images rather than a synthesized image. Multiple views can add geometric evidence: 3D-R2N2 addresses single- and multi-view reconstruction [14], while differentiable ray consistency uses multi-view observations to supervise single-view prediction [28]. Single-image methods such as Zero-1-to-3 [11], pixelNeRF [12], and PIFu [13] must infer unseen geometry. The four modes are therefore distinct input and conditioning paths that converge on the same job-management and GLB-delivery mechanism.

2.5 Functional, Compatibility, Performance, and Usability Measures

A functional scenario passed when observed behavior matched the requirement-derived expected result. The complete 14-scenario matrix is provided in Appendix Table A1, including a negative empty-prompt validation case and a simulated process-failure case. The retained record documents these tests as part of the integrated prototype evaluation but does not preserve an individual tester identity or an immutable build hash; accordingly, the paper does not claim a formally frozen release build.

Cross-viewer GLB compatibility was defined narrowly as the ability of the tested GLB output to open and display without a reported read/import failure in the integrated website viewer, Blender, and 3dviewer.net. This is not treated as full interoperability because material preservation, texture integrity, transforms, mesh validity, schema conformance, and repeated import–export behavior were not evaluated.

For performance, `processing_time` was defined as server-side turnaround rather than isolated model inference. It includes workflow monitoring, artifact retrieval from ComfyUI, and transfer to ImageKit. Mean, median, minimum, and maximum were reported by mode, and only jobs recorded as COMPLETED were available for analysis. One retained job required an internal retry before completion; an overall generation success rate is not calculated because the archive does not contain a complete denominator of all attempts, final failures, and cancellations.

Task observation used the 12-task script and completion records presented in Appendix Table B1. Each task execution was recorded as independent completion, completion with assistance, or failure. The retained task-level record provides independent and assisted completion counts and error events by task, as well as participant-level session times (Appendix Table B2); it does not preserve separate time-on-task values for each task or a sufficiently detailed intervention threshold to reproduce the facilitator's assistance decision exactly. Observed errors were categorized as input, navigation, interpretation, or procedural. Because the task-based and SUS groups were explicitly separate, no participant-level overlap is assumed.

The seven-person task study is treated as formative diagnostic usability testing rather than a summative population estimate. Early usability research showed that small formative samples can reveal recurrent interface problems with diminishing returns [29], while later evidence demonstrates substantial variability in what very small samples detect and cautions against treating a five-user rule as universally sufficient [30]. The present sample is therefore used to locate observed friction, not to estimate its prevalence in a broader population.

SUS used the standard 10-item scoring procedure [31]: odd-item responses were reduced by one, even-item responses were subtracted from five, the adjusted items were summed, and the result was multiplied by 2.5. The original study protocol classified scores of at least 70 as Acceptable; this label is used descriptively because SUS interpretation is context-dependent [32].

2.6 Statistical Analysis and Claim Boundaries

Because the performance groups were small and unbalanced, an exploratory Kruskal–Wallis test compared turnaround distributions across the four modes at $\alpha = .05$. Epsilon squared (ϵ^2) summarized omnibus effect magnitude, and no pairwise post hoc claim was made. Mode, input complexity, system load, workflow composition, and transfer overhead are not separable in the retained deployment data; therefore, statistical significance is not interpreted as evidence that generation mode caused the observed timing differences.

SUS results were summarized by the mean, sample standard deviation, median, range, and a two-sided 95% t confidence interval for the mean. Task executions were not treated as independent participants, so no confidence interval was attached to the 84 execution-level outcomes. Functional scenarios support conformance only for executed cases; compatibility supports file opening rather than geometric correctness; completed-job timing does not estimate failure probability; and the single expert assessment is an initial judgment rather than an inter-rater reliability study. No claim of production scalability, educational effectiveness, or state-of-the-art reconstruction follows from these measures.

3. RESULTS

Results are presented according to the research questions. Figure 2 shows the My Projects interface, where completed 3D generations and their visibility status are organized within the implemented

workflow. Functional testing, job records, task observation, SUS, and expert assessment were used to evaluate system operation, turnaround, interaction, perceived usability, and generated-model quality.

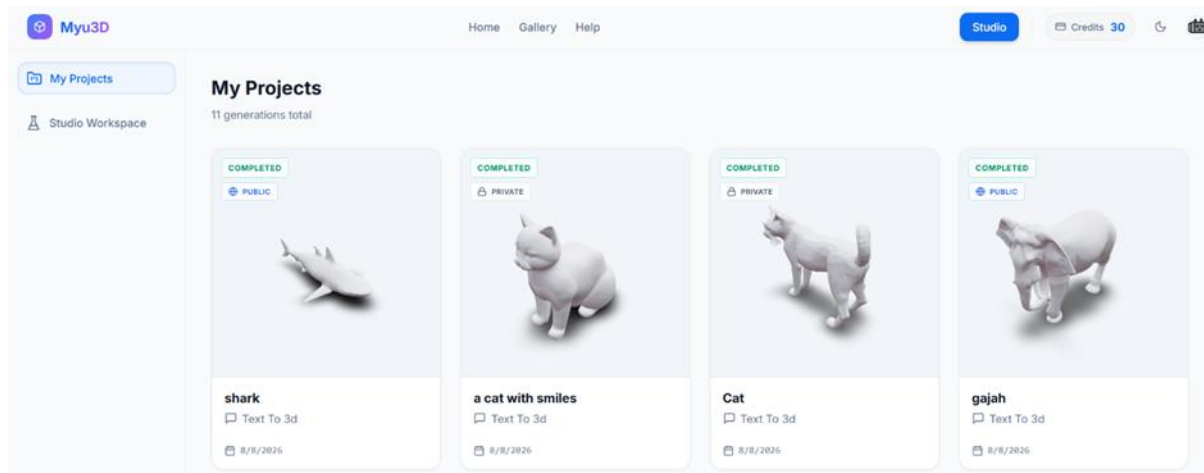


Figure 2. My Projects interface displaying completed 3D generations and their visibility status.

3.1 Functional Workflow and Cross-Viewer GLB Compatibility (RQ1)

All 14 requirement-derived black-box scenarios produced the expected outcomes (Appendix Table A1). The matrix covers prompt submission, empty-input validation, job creation, queued–processing–completed transitions, failure messaging, generation completion, viewer loading and interaction, download, Text-to-3D multi-stage execution, Single Image-to-3D, Multi Image-to-3D, ControlNet, and periodic status polling. These results demonstrate functional conformance for the executed cases, not exhaustive defect absence.

The tested GLB output opened successfully in the website viewer, Blender, and 3dviewer.net. This supports cross-viewer GLB compatibility, but not full interoperability because material preservation, mesh validity, transforms, schema conformance, and round-trip exchange were not evaluated.

3.2 Exploratory Deployment Turnaround by Generation Mode (RQ2)

The performance dataset contained 35 jobs recorded as COMPLETED: 20 Text-to-3D jobs and five jobs in each remaining mode. Table 1 summarizes server-side turnaround by mode. Because prompts and images were heterogeneous and mode counts were unequal, the statistics characterize the observed workload rather than a controlled model benchmark.

Table 1. Server-side turnaround of completed jobs by generation mode.

Generation mode	n	Mean (s)	Median (s)	Minimum (s)	Maximum (s)
Text-to-3D	20	63.10	63.44	50.57	72.70
Text-to-3D + ControlNet	5	82.94	77.52	66.54	108.39
Single Image-to-3D	5	48.79	57.86	10.87	62.86
Multi Image-to-3D	5	78.96	65.90	48.37	110.49
Total	35	66.15	63.65	10.87	110.49

Across all jobs, mean turnaround was 66.15 s and the median was 63.65 s. The exploratory omnibus test indicated that at least one observed workload distribution differed, $H(3) = 10.52$, $p = .015$, $\epsilon^2 = .243$. This result does not identify a superior mode or establish mode as the cause of timing differences. Input complexity, workflow composition, shared infrastructure, transfer overhead, and unrecorded system load may contribute to the observed variation.

Single Image-to-3D included one 10.87 s observation but otherwise clustered above 53 s, whereas ControlNet and Multi Image-to-3D extended beyond 108 s. These values are end-to-end observations from one deployment rather than hardware-normalized inference latency. The asynchronous architecture is nevertheless supported pragmatically by the observed tens-of-seconds turnaround because it allows the browser to return promptly and communicate status while the backend continues processing.

Figure 3 presents representative outputs from each generation path so that timing evidence is not mistaken for evidence of visual fidelity. The examples show the common GLB delivery format and outputs obtained through four conditioning routes. They are qualitative examples rather than a randomly sampled or blinded comparison set. Their role is therefore descriptive and is complemented by the expert scores reported later.

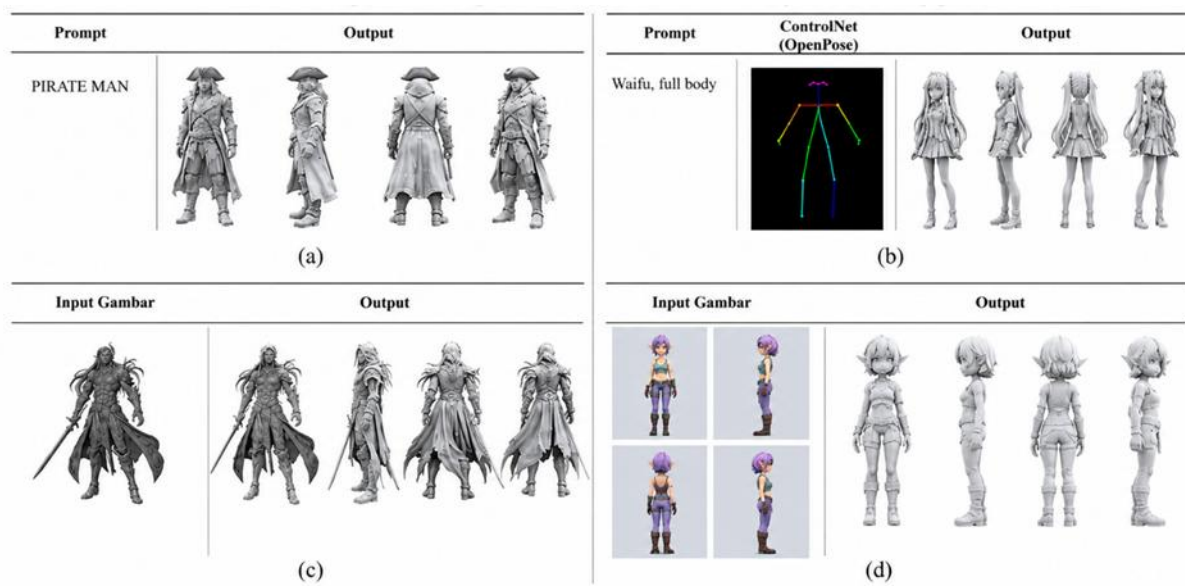


Figure 3. Representative outputs from (a) Text-to-3D, (b) Text-to-3D with ControlNet, (c) Single Image-to-3D, and (d) Multi Image-to-3D.

The examples retained recognizable characteristics of their prompts or input images, but geometric detail, structural completeness, and input correspondence varied. This variation is consistent with the expert assessment and cautions against equating successful file generation with consistently high-quality geometry. No reference mesh, multiview ground truth, or blinded perceptual comparison was available. The visual evidence therefore supports feasibility and identifies failure dimensions without quantifying reconstruction accuracy.

3.3 Observed and Perceived Usability (RQ3)

Seven participants each attempted 12 tasks, producing 84 task executions. Every task was eventually completed, but 61 executions (72.62%) were independent, and 23 (27.38%) required assistance. No execution was classified as a failure. Table 2 shows task-level independent and assisted completion, showing that assistance was concentrated in particular tasks.

Table 2. Task-level completion outcomes in the task-based usability study.

Task	Task description	Independent	Assisted	Failed
1	Sign in with Google	7	0	0
2	Open Gallery and find an interesting 3D model	5	2	0
3	Like the selected model	6	1	0
4	Download the selected Gallery model	7	0	0
5	Enter Studio	2	5	0
6	Generate a new 3D model from text	5	2	0
7	Inspect generated model from multiple viewpoints	6	1	0
8	Download the generated model	7	0	0
9	Make the generated model public	4	3	0
10	Find the generated model in My Projects	2	5	0
11	Find the newly published model in Gallery	6	1	0
12	Log out	4	3	0

Tasks 5 (enter Studio) and 10 (find the generated model in My Projects) each required assistance for five of seven participants, while Tasks 9 (make the model public) and 12 (log out) each required assistance for three participants. This task-level pattern makes the global 27.38% assistance rate actionable: the main issue was not inability to complete the workflow, but discoverability of navigation and result-management actions.

Forty-eight observed error events were recorded across the 84 executions: 28 input errors (58.33%), 16 navigation errors (33.33%), two interpretation errors (4.17%), and two procedural errors (4.17%). Error events were concentrated in Task 1 (12 events), Task 10 (9), Tasks 2 and 5 (8 each), with zero recorded events in Tasks 4 and 8. One task could contain more than one error event, so error counts are not equivalent to task failures. Appendix Table C1 reports error categories, Appendix Table C2 provides task-level counts, and Appendix Table C3 summarizes qualitative observations. Figure 4 juxtaposes assistance and error events by task.

Mean total time to complete the full 12-task session was 333.29 s per participant (median = 297 s; range = 275–423 s; Appendix Table B2). The study archive does not retain separate time-on-task values for each task, so timing is interpreted only at the participant-session level.

The 12 SUS respondents were a separate student group from the seven task-based participants. They were 21–23 years old, with nine men and three women; reported study backgrounds included Visual Communication Design (4), Informatics (2), Civil Engineering (1), Electronics Engineering (1), Informatics Education (1), and three unspecified study programs. SUS scores ranged from 67.50 to 95.00, with a mean of 81.67 (SD = 9.25), median of 81.25, and 95% CI [75.79, 87.55]. Eleven respondents met the study’s descriptive Acceptable threshold and one was classified as Marginal.

The favorable overall SUS score coexisted with assistance during more than one quarter of the observed task executions. This divergence is the central human-centered finding: perceived ease and successful eventual completion did not eliminate first-use discoverability friction. Qualitative observations were consistent with the task counts: participants asked how to find Studio, My Projects, and Logout, questioned prompt-language flexibility and style controls, and sometimes required direction when changing model visibility.

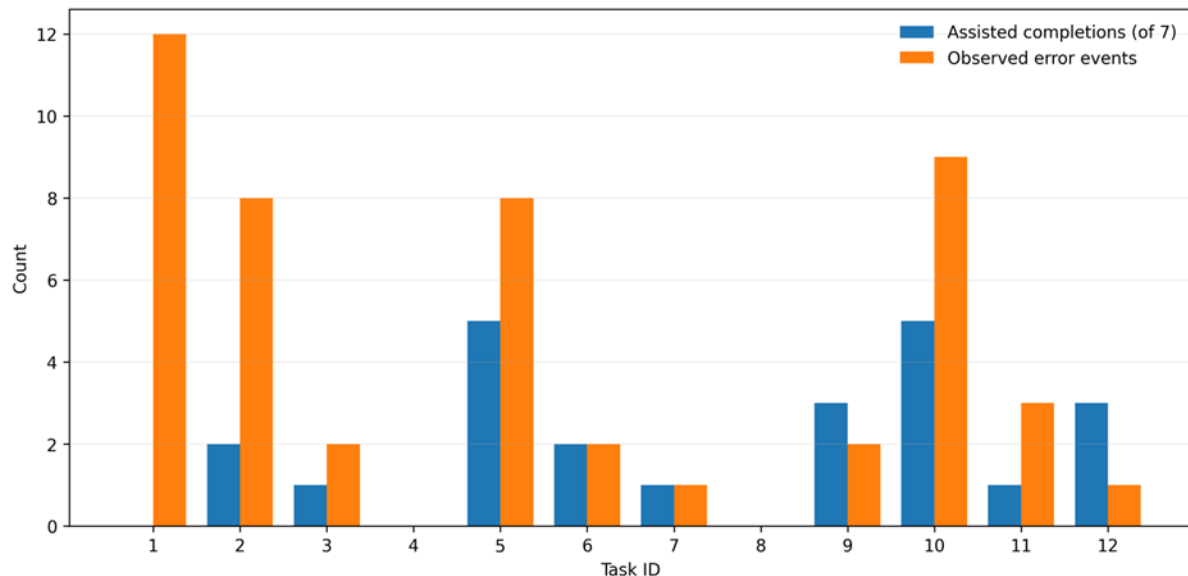


Figure 4. Task-level assistance and observed error profile across the 12 usability tasks.

3.4 Expert Assessment of Implementation and 3D Output

The two expert instruments are reported separately because implementation conformity and generated-model quality are not interchangeable outcomes (Table 3). The system instrument contained 20 five-point items and produced 81/100 (mean 4.05/5). The output-quality instrument contained 10 five-point items and produced 35/50 (mean 3.50/5). Complete item-level scores are provided in Appendix Table D1 (system implementation) and Appendix Table D2 (generated-model quality).

Table 3. Expert assessment of system implementation and generated-model quality.

Assessment	Indicators	Score	Mean
System implementation	20	81/100	4.05/5
Generated 3D quality	10	35/50	3.50/5

For system implementation, the highest ratings (5/5) were assigned to system-function conformity, generation-flow conformity, and prompt enhancement; 3D output and interface clarity were lower at 3/5. For generated-model quality, GLB output suitability received 5/5; detail, surface quality, and suitability as a 3D asset received 4/5; geometry, structural completeness, shape accuracy, multiview consistency, visual conformity, and prompt/image correspondence received 3/5. The pattern supports an operational prototype with usable export while locating the principal quality constraints in geometry, structural fidelity, and cross-view consistency. Because one validator completed both instruments, no inter-rater reliability statistic can be established.

4. DISCUSSION

The prototype succeeded as an integrated human–AI authoring workflow under the evaluated conditions, but the principal contribution is the separation of forms of evidence that are often conflated. Functional scenarios establish executed conformance, cross-viewer checks establish GLB readability across three environments, timing characterizes one deployment workload, task observation exposes interaction friction, SUS captures global perception, and expert review qualifies output quality. Reading these measures together produces a more defensible assessment than treating any single score as proof of overall system quality.

4.1 From Model Components to an Interactive Authoring Environment

For RQ1, the results show that established generative components can be assembled into a browser-accessible workflow without proposing a new generative model. The contribution is architectural and interactional: the system converts heterogeneous stages into a coherent sequence of input, queued execution, status feedback, persistent output, inspection, and download. This application layer complements diffusion synthesis [4], latent diffusion [5], vision–language alignment [6], DreamFusion [7], Magic3D [8], ProlificDreamer [9], visual-prompt conditioning [10], and image-conditioned reconstruction [11], [12], [13], [14], [15], [16]. It also operationalizes human-centered guidance on reliable control [1], feedback and correction [2], and AI-specific design challenges [3].

The application-level contribution is more directly comparable to systems such as Codeless3D [17] and MagicCraft [18], which likewise aim to reduce specialist barriers to 3D creation. The present study adds a specific claim–evidence structure for a web-based generative pipeline: orchestration reliability, file readability, deployment turnaround, observed independent use, perceived usability, and output quality are measured separately. This separation is important because a visually plausible model can still be embedded in a poor workflow, and a technically reliable workflow can still produce geometrically imperfect assets.

Educationally, this separation is useful because technology-enhanced-learning applications require both workable tools and understandable interaction. A student authoring activity may be disrupted if users cannot find the generation workspace or retrieve their results, even when the generative model is technically capable. Conversely, easy navigation does not guarantee that a generated asset has sufficient geometry for a learning task. The present study therefore motivates, but does not demonstrate, future classroom evaluation in which interaction measures are paired with learning or project outcomes.

4.2 Deployment Turnaround and Cross-Viewer Compatibility

For RQ2, the omnibus difference among the observed turnaround distributions is best interpreted as a deployment signal that merits profiling, not as evidence of an intrinsic speed hierarchy among generation modes. ControlNet introduces additional conditioning operations and multi-image reconstruction processes additional views, making longer or more variable workflows plausible, but the retained processing time also includes monitoring, retrieval, and storage transfer. Hardware load and input complexity were not controlled. A statistically significant omnibus result therefore cannot isolate generation mode as the causal factor.

Cross-viewer opening in the integrated viewer, Blender, and 3dviewer.net strengthens practical relevance because exported assets are not locked to the originating web interface. Nevertheless, the evidence is deliberately termed cross-viewer GLB compatibility. It demonstrates opening/readability,

not complete interoperability. Future compatibility testing should add GLTF/GLB schema validation, material and texture preservation, transform checks, mesh-manifold or integrity tests, file-size reporting, and repeated import–export round trips.

4.3 The Usability–Discoverability Gap

For RQ3, the combined task and SUS evidence reveals a discrepancy between overall perceptions and observed behavior. Mean SUS was favorable at 81.67, and all specified tasks were eventually completed; however, 23 of 84 executions required assistance. The task-level data show that this assistance was concentrated in navigation and result-management steps. Entering Studio and returning to a generated model through My Projects each required assistance for five of seven participants, while making a model public and logging out each required assistance for three. The evidence therefore points to discoverability rather than broad functional failure.

The 48 recorded error events provide further detail. Input and navigation accounted for 44 of 48 events, whereas interpretation and procedural errors were uncommon. Interface redesign should therefore prioritize information scent and primary-action visibility before adding new features: Studio and My Projects should be prominent persistent destinations; Gallery and visibility state should use explicit labels; logout should be consistently located; prompt input should state supported languages and examples; style controls should explain their effect; and job status should communicate expected duration, queued/processing state, completion, and recoverable errors. These recommendations are directly tied to the observed error and assistance pattern rather than being generic interface advice.

The difference between task observation and SUS also illustrates why global questionnaires should not replace behavioral evaluation. SUS captures overall perceived usability [31], while its interpretation and accumulated evidence are discussed more broadly in [32]. A favorable global score can therefore coexist with assistance on specific first-use tasks. Conversely, the task study’s seven-person formative sample is not large enough to estimate population-level incidence. The two methods are therefore complementary: task observation localizes friction; SUS summarizes perception in a separate sample.

4.4 Output Quality, Transferable Implications, and Limitations

The expert output score of 3.50/5 qualifies the favorable functional and SUS findings. GLB suitability was strong, but geometry, structural completeness, shape accuracy, multiview consistency, and correspondence to prompts or images were only moderate. Related reconstruction literature addresses relevant geometric and appearance challenges through neural surface reconstruction [33], texture fields [34], volume-rendered implicit surfaces [35], and generative radiance fields [36]. The present result supports the system as an accessible research prototype and authoring aid, not as evidence of state-of-the-art reconstruction fidelity or production-ready assets without further editing.

Three transferable implications follow. First, studies of interactive generative media should report orchestration and interaction outcomes alongside model outputs because users experience the entire workflow. Second, eventual completion and global perceived usability should be separated from independent completion, assistance, and discoverability. Third, output portability should be assessed independently from output fidelity. These principles provide a reusable multidimensional evaluation framework but do not establish computational reproducibility.

Several limitations constrain generalization. Timing data include only completed jobs, use unbalanced mode counts, combine heterogeneous prompts and images, and lack preserved hardware telemetry, package versions, random seeds, dependency locks, and system-load measurements. Task testing

involved seven convenience-sampled students; the separate SUS sample contained 12 students; prior 3D/generative-AI experience was not retained; and no baseline interface or longitudinal use was tested. The task archive provides task-level completion and error counts but not per-task time or a fully reproducible facilitator-intervention threshold. Functional testing includes a complete retained 14-scenario matrix, but the individual tester identity and immutable build identifier are not preserved. Only one expert rated implementation and 3D quality, so inter-rater reliability cannot be estimated. The study did not include production load, security/stress testing, a complete failure-rate denominator, objective geometric ground truth, blinded perceptual comparison, or educational outcomes.

The available ethics documentation is limited. The study was conducted as a non-sensitive, non-interventional software-usability evaluation within the undergraduate research process. Participation was voluntary and informed consent was obtained before task-based testing or SUS participation. According to the institutional requirements recorded for the study, formal ethics committee clearance was not required. The retained research record does not include a prospective ethics-exemption letter or identifier, so this determination cannot be independently substantiated from the archived documentation.

Future work should preserve containerized environment manifests and hardware telemetry, balance and randomize mode-specific workloads, retain failed and cancelled jobs, and profile each pipeline stage. Human-centered evaluation should recruit broader user groups, document prior experience, define assistance thresholds prospectively, retain per-task timing, compare revised interfaces, and examine learnability over repeated sessions. Output evaluation should add multiple blinded raters, inter-rater reliability, reference-based geometry metrics where ground truth exists, and perceptual comparison against alternative workflows. Educational extensions should test whether generated 3D assets improve specific learning, design, or communication outcomes rather than assuming benefit from usability alone.

5. CONCLUSION

This study developed and evaluated a web-based, multi-stage generative 3D workflow as a human–AI authoring environment. Under the evaluated conditions, all 14 functional scenarios produced the expected behavior; the tested GLB output opened in three viewing environments; and 35 completed jobs averaged 66.15 s of server-side turnaround. The four observed workload distributions differed in an exploratory omnibus analysis, but the study design does not support a causal mode-speed claim. All 84 task executions were eventually completed, while 61 (72.62%) were independent and 23 (27.38%) required assistance. Task-level evidence located the greatest assistance in entering Studio and recovering results through My Projects. SUS was favorable in a separate 12-student sample, but the assistance rate, input/navigation errors, and expert quality score exposed discoverability and geometry constraints that a global usability score alone would conceal.

The principal contribution is not a new generative model but an integrated user-facing workflow and a multidimensional evaluation logic in which functional conformance, deployment turnaround, cross-viewer compatibility, observed interaction, perceived usability, and output quality are treated as related but non-equivalent outcomes. This framing is relevant to interactive media and technology-enhanced learning because it clarifies what must be demonstrated before a generative 3D tool can be considered accessible to non-specialist users. The prototype provides a basis for further research, while stronger claims require balanced workloads, preserved environment metadata, comparative task

studies, multiple experts, objective geometry measures, and direct learning-outcome evaluation when educational use is intended.

DECLARATIONS

Author Contributions

Ray Fadly Hadi: Conceptualization; Methodology; Software; Investigation; Data Curation; Formal Analysis; Visualization; Writing – Original Draft. **Randi Proska Sandra:** Supervision; Validation; Writing – Review & Editing. **Sri Rejeki:** Writing – Review & Editing. All authors have read and approved the final version of the manuscript.

Acknowledgments

The authors thank the expert for validation and the Informatics Study Program, Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang, for academic and technical support.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing Interests

The authors declare no competing interests.

Ethics Approval

The study involved non-sensitive software-usability activities. According to the institutional requirements reported in the research record, formal ethical clearance was not required for this evaluation; participation was voluntary, and no medical, biological, or psychological intervention was conducted.

Informed Consent

Informed consent was obtained from all participants before task-based usability testing or completion of the System Usability Scale, as applicable.

Consent for Publication

Not applicable. The article does not report identifiable participant images or confidential personal records.

Data Availability Statement

Aggregate evaluation results are reported in the article. The corresponding author may provide anonymized participant-level records, de-identified job measurements, and supporting system documentation upon reasonable request, subject to institutional and privacy requirements.

Generative AI and AI-Assisted Technologies Statement

During manuscript preparation, the authors used ChatGPT (OpenAI) and Claude (Anthropic) for language refinement. These tools were not used to generate, modify, or analyze the research data, usability scores, performance measurements, or expert assessments. All AI-assisted text was checked against the thesis record and cited sources, and the authors accept full responsibility for the final manuscript.

Preprint and Prior Dissemination Statement

This article is derived from the first author's undergraduate thesis and is not under consideration elsewhere.

REFERENCES

- [1] B. Shneiderman, “Human-centered artificial intelligence: Reliable, safe & trustworthy,” *Int. J. Hum.-Comput. Interact.*, vol. 36, no. 6, pp. 495–504, 2020, doi: [10.1080/10447318.2020.1741118](https://doi.org/10.1080/10447318.2020.1741118).
- [2] S. Amershi et al., “Guidelines for human–AI interaction,” in *Proc. 2019 CHI Conf. Human Factors Comput. Syst.*, Glasgow, U.K., 2019, pp. 1–13, doi: [10.1145/3290605.3300233](https://doi.org/10.1145/3290605.3300233).
- [3] Q. Yang, A. Steinfeld, C. Rosé, and J. Zimmerman, “Re-examining whether, why, and how human–AI interaction is uniquely difficult to design,” in *Proc. 2020 CHI Conf. Human Factors Comput. Syst.*, Honolulu, HI, USA, 2020, pp. 1–13, doi: [10.1145/3313831.3376301](https://doi.org/10.1145/3313831.3376301).
- [4] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 10674–10685, doi: [10.1109/CVPR52688.2022.01042](https://doi.org/10.1109/CVPR52688.2022.01042).
- [6] A. Radford et al., “Learning transferable visual models from natural language supervision,” in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, pp. 8748–8763, 2021.
- [7] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “DreamFusion: Text-to-3D using 2D diffusion,” in *Proc. Int. Conf. Learn. Representations*, 2023.
- [8] C.-H. Lin et al., “Magic3D: High-resolution text-to-3D content creation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 300–309, doi: [10.1109/CVPR52729.2023.00037](https://doi.org/10.1109/CVPR52729.2023.00037).
- [9] Z. Wang et al., “ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation,” in *Advances in Neural Information Processing Systems*, vol. 36, pp. 8406–8441, 2023, doi: [10.52202/075280-0368](https://doi.org/10.52202/075280-0368).
- [10] Y. Chen, Y. Pan, H. Yang, T. Yao, and T. Mei, “VP3D: Unleashing 2D visual prompt for text-to-3D generation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2024, pp. 4896–4905, doi: [10.1109/CVPR52733.2024.00468](https://doi.org/10.1109/CVPR52733.2024.00468).
- [11] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, “Zero-1-to-3: Zero-shot one image to 3D object,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 9264–9275, doi: [10.1109/ICCV51070.2023.00853](https://doi.org/10.1109/ICCV51070.2023.00853).

- [12] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelNeRF: Neural radiance fields from one or few images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 4576–4585, doi: [10.1109/CVPR46437.2021.00455](https://doi.org/10.1109/CVPR46437.2021.00455).
- [13] S. Saito, Z. Huang, R. Natsume, S. Morishima, H. Li, and A. Kanazawa, “PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 2304–2314, doi: [10.1109/ICCV.2019.00239](https://doi.org/10.1109/ICCV.2019.00239).
- [14] C. B. Choy, D. Xu, J. Y. Gwak, K. Chen, and S. Savarese, “3D-R2N2: A unified approach for single and multi-view 3D object reconstruction,” in *Computer Vision – ECCV 2016*, Cham, Switzerland: Springer, 2016, pp. 628–644, doi: [10.1007/978-3-319-46484-8_38](https://doi.org/10.1007/978-3-319-46484-8_38).
- [15] X. Long et al., “Wonder3D: Single image to 3D using cross-domain diffusion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 9970–9980, doi: [10.1109/CVPR52733.2024.00951](https://doi.org/10.1109/CVPR52733.2024.00951).
- [16] L. Qiu et al., “RichDreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3D,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 9914–9925, doi: [10.1109/CVPR52733.2024.00946](https://doi.org/10.1109/CVPR52733.2024.00946).
- [17] C. Volioti, V. Martsis, A. Ampatzoglou, E. Keramopoulos, and A. Chatzigeorgiou, “Codeless3D: Design and usability evaluation of a low-code tool for 3-D game generation,” *IEEE Trans. Games*, vol. 17, no. 2, pp. 296–307, 2025, doi: [10.1109/TG.2024.3424894](https://doi.org/10.1109/TG.2024.3424894).
- [18] R. Kurai, T. Hiraki, Y. Hiroi, Y. Hirao, M. Perusquía-Hernández, H. Uchiyama, and K. Kiyokawa, “MagicCraft: Natural language-driven generation of dynamic and interactive 3D objects for commercial metaverse platforms,” *IEEE Access*, vol. 13, pp. 132459–132474, 2025, doi: [10.1109/ACCESS.2025.3587232](https://doi.org/10.1109/ACCESS.2025.3587232).
- [19] X. Xue, Z. Lu, D. Huang, Z. Wang, W. Ouyang, and L. Bai, “ComfyBench: Benchmarking LLM-based agents in ComfyUI for autonomously designing collaborative AI systems,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2025, pp. 24614–24624, doi: [10.1109/CVPR52734.2025.02292](https://doi.org/10.1109/CVPR52734.2025.02292).
- [20] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2014.
- [21] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 3813–3824, doi: [10.1109/ICCV51070.2023.00355](https://doi.org/10.1109/ICCV51070.2023.00355).
- [22] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4455–4465, doi: [10.1109/CVPR.2019.00459](https://doi.org/10.1109/CVPR.2019.00459).
- [23] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 165–174, doi: [10.1109/CVPR.2019.00025](https://doi.org/10.1109/CVPR.2019.00025).
- [24] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, “Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 3501–3512, doi: [10.1109/CVPR42600.2020.00356](https://doi.org/10.1109/CVPR42600.2020.00356).
- [25] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan, “RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 5470–5480, doi: [10.1109/CVPR52688.2022.00540](https://doi.org/10.1109/CVPR52688.2022.00540).

- [26] H. Kato, Y. Ushiku, and T. Harada, “Neural 3D mesh renderer,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 3907–3916, doi: [10.1109/CVPR.2018.00411](https://doi.org/10.1109/CVPR.2018.00411).
- [27] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Trans. Graph.*, vol. 41, no. 4, Art. no. 102, pp. 1–15, 2022, doi: [10.1145/3528223.3530127](https://doi.org/10.1145/3528223.3530127).
- [28] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, “Multi-view supervision for single-view reconstruction via differentiable ray consistency,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8754–8765, 2022, doi: [10.1109/TPAMI.2019.2898859](https://doi.org/10.1109/TPAMI.2019.2898859).
- [29] R. A. Virzi, “Refining the test phase of usability evaluation: How many subjects is enough?,” *Hum. Factors*, vol. 34, no. 4, pp. 457–468, 1992, doi: [10.1177/001872089203400407](https://doi.org/10.1177/001872089203400407).
- [30] L. Faulkner, “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing,” *Behav. Res. Methods Instrum. Comput.*, vol. 35, no. 3, pp. 379–383, 2003, doi: [10.3758/BF03195514](https://doi.org/10.3758/BF03195514).
- [31] J. Brooke, “SUS: A ‘quick and dirty’ usability scale,” in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194, doi: [10.1201/9781498710411-35](https://doi.org/10.1201/9781498710411-35).
- [32] J. R. Lewis, “The System Usability Scale: Past, present, and future,” *Int. J. Hum.-Comput. Interact.*, vol. 34, no. 7, pp. 577–590, 2018, doi: [10.1080/10447318.2018.1455307](https://doi.org/10.1080/10447318.2018.1455307).
- [33] Z. Li et al., “Neuralangelo: High-fidelity neural surface reconstruction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 8456–8465, doi: [10.1109/CVPR52729.2023.00817](https://doi.org/10.1109/CVPR52729.2023.00817).
- [34] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, and A. Geiger, “Texture fields: Learning texture representations in function space,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 4530–4539, doi: [10.1109/ICCV.2019.00463](https://doi.org/10.1109/ICCV.2019.00463).
- [35] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, “NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 27171–27183, 2021.
- [36] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, “GRAF: Generative radiance fields for 3D-aware image synthesis,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 20154–20166, 2020.



This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution, and reproduction in any medium or format, provided that appropriate credit is given to the original author(s) and the source, a link to the Creative Commons license is provided, and any modifications are indicated. Unless otherwise specified in a credit line, this article’s images or other third-party material are included under the Creative Commons license. If certain material is not covered by the article’s Creative Commons license and its intended use is not permitted by statutory regulation or exceeds the allowed usage, permission must be obtained directly from the copyright holder. <http://creativecommons.org/licenses/by/4.0/>.

AUTHOR BIOGRAPHIES



Ray Fadly Hadi is an undergraduate student in Informatics at Universitas Negeri Padang, Indonesia. His academic interests primarily include digital forensics, cybersecurity, security operations, and incident response. He also has interests in software engineering, artificial intelligence, and the development of web-based systems. His current undergraduate research focuses on developing and evaluating a web-based text-to-3D generation system using multi-stage generative modeling. This research has provided experience in integrating generative AI workflows, asynchronous processing, web technologies, and interactive 3D visualization into an end-to-end software system. His broader career and research interests are directed toward digital forensics and cybersecurity.



Randi Proska Sandra is a faculty member in the Department of Electronics Engineering, Universitas Negeri Padang, Indonesia. His research expertise includes programming education, human-centered computing, and social informatics. He has served as a Visiting Research Fellow at NTNU, Norway, and holds an academic background from National Central University, Taiwan. He has contributed as an Editorial Board Member for SAGE and as a reviewer for leading ACM conferences, including CSCW and SIGCSE. His scholarly work emphasizes the dissemination of educational technology and computer science research at the international level.



Sri Rejeki is a Master's student in Informatics at Telkom University, specializing in Data Science. She holds a Bachelor's degree in Informatics Engineering from Telkom University Purwokerto, with a concentration in Full-Stack Development and a GPA of 3.77/4.00. She has professional experience as a Freelance Software & Data Analyst, focusing on web and desktop application development, data analysis, database management, and machine learning. She also works as an Instructor at Dicoding Indonesia, where she mentors learners in machine learning concepts, model evaluation, and Python-based implementation. Her areas of expertise include software development, data science, machine learning, Python, SQL, data modeling, and data cleansing. Her current research interests focus on data science, machine learning, and the development of data-driven and intelligent technology solutions.



Publisher's and Journal's Note *Sagamedia Teknologi Nusantara, as the publisher and editor of the Journal of Hypermedia & Technology-Enhanced Learning (J-HyTEL), upholds the highest ethical standards in academic publishing. The journal remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. Authors are fully responsible for the originality, accuracy, and integrity of their work. Post-publication ethical concerns will be addressed through corrections, clarifications, or retractions as necessary. The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States.* <https://www.un.org/sustainabledevelopment>

APPENDIX A. COMPLETE BLACK-BOX FUNCTIONAL TEST MATRIX

The matrix below reports the complete retained 14-scenario functional record. Expected results are stated as the requirement criterion corresponding to each retained test scenario; observed results reproduce the recorded study outcomes. The study archive does not preserve the individual tester identity or an immutable build hash.

Appendix Table A1. Complete black-box functional test matrix.

ID	Function	Input/action	Expected result	Observed result	Status
BB-01	Text prompt input	Enter a prompt and press Generate	A generation job is created and a Job ID is returned	Job created and Job ID returned	Pass
BB-02	Input validation	Leave prompt empty and press Generate	Generation is blocked and a validation message is shown	Validation message shown; generation cancelled	Pass
BB-03	Job creation	Submit a valid prompt	Initial job state is QUEUED	Initial state recorded as QUEUED	Pass
BB-04	Job status update	Wait during inference	State transitions QUEUED → PROCESSING → COMPLETED	Recorded transitions matched expected sequence	Pass
BB-05	Process completion	Allow workflow to complete	A GLB model is produced	GLB model produced	Pass
BB-06	Error handling	Simulate process failure	State changes to FAILED and an error message is exposed	FAILED state and error message recorded	Pass
BB-07	3D Viewer	Open generated result	GLB loads in integrated viewer	Model loaded in 3D Viewer	Pass
BB-08	Viewer interaction	Rotate, zoom, and pan	All supported interactions respond	Rotation, zoom, and pan operated	Pass
BB-09	Download model	Press Download	GLB file downloads successfully	GLB downloaded	Pass
BB-10	Multi-stage Text-to-3D	Run Text-to-3D workflow	Pipeline completes through all stages	Pipeline completed	Pass
BB-11	Single Image-to-3D	Upload one image	A 3D model is produced	3D model produced	Pass
BB-12	Multi Image-to-3D	Upload multiple images	A 3D model is produced	3D model produced	Pass
BB-13	ControlNet mode	Provide a control image	ControlNet workflow completes	Workflow processed successfully	Pass
BB-14	Status monitoring	Keep generation page open during processing	Status updates periodically via polling	Status updated periodically via polling	Pass

APPENDIX B. TASK-BASED USABILITY PROTOCOL AND OUTCOMES

Seven task-based participants each completed the same 12-task script. The table reports the retained task-level completion counts. “Independent” denotes completion recorded without assistance; “Assisted” denotes eventual completion after assistance. The retained archive does not preserve a more granular facilitator-intervention threshold or per-task timing.

Appendix Table B1. Task-level completion outcomes across seven participants.

Task	Task instruction	Independent	Assisted	Failed	Total
1	Sign in with Google	7	0	0	7
2	Open Gallery and find an interesting 3D model	5	2	0	7
3	Like the selected model	6	1	0	7
4	Download the selected Gallery model	7	0	0	7
5	Enter Studio	2	5	0	7
6	Generate a new 3D model from text	5	2	0	7
7	Inspect generated model from multiple viewpoints	6	1	0	7
8	Download the generated model	7	0	0	7
9	Make the generated model public	4	3	0	7
10	Find the generated model in My Projects	2	5	0	7
11	Find the newly published model in Gallery	6	1	0	7
12	Log out	4	3	0	7

Appendix Table B2. Total time required by each participant to complete the 12-task protocol.

Participant	Total time for 12 tasks (s)
R1	297
R2	283
R3	275
R4	420
R5	286
R6	423
R7	349

APPENDIX C. OBSERVED ERROR DISTRIBUTION

Appendix Table C1. Error categories.

Error category	Count	Share of 48 events
Input	28	58.33%
Navigation	16	33.33%
Interpretation	2	4.17%
Procedural	2	4.17%

Appendix Table C2. Error events by task.

Task	Total error events	Participants with ≥ 1 error	Errors per task/7 participants
1	12	3	1.71
2	8	3	1.14
3	2	2	0.29
4	0	0	0.00
5	8	4	1.14
6	2	1	0.29
7	1	1	0.14
8	0	0	0.00
9	2	2	0.29
10	9	5	1.29
11	3	2	0.43
12	1	1	0.14

Appendix Table C3. Representative qualitative observations retained in the study record.

Category	Observation	Example participant utterance
Navigation	Difficulty locating Studio	“Where is the Studio?”
Navigation	Difficulty locating My Projects	“‘My Projects’? Where is that?”
Navigation	Difficulty locating Logout	“How do I log out?”
Interpretation	Uncertainty about style control	“Can I use this style?”
Procedural	Needed direction to make a model public	“Which one?”
Input	Questioned prompt-language flexibility	“Can I use either Indonesian or English?”

APPENDIX D. EXPERT-ASSESSMENT INDICATORS AND SCORES

Both expert instruments used a five-point Likert scale: 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, and 5 = strongly agree. One validator with expertise in artificial intelligence, machine learning, and 3D animation completed both instruments.

Appendix Table D1. System implementation assessment (20 items).

No.	Indicator	Score
1	System function conformity	5
2	Generation-flow conformity	5
3	Input acceptance	4
4	Input validation	4
5	Job management	4
6	Process status	4
7	AI workflow integration	4
8	Multi-stage pipeline	4
9	Prompt enhancement	5
10	3D model output	3
11	Model visualization	4
12	3D Viewer interaction	4
13	Model download	4
14	History/Recent Generation	4
15	Gallery	4
16	Authentication	4
17	User-data separation	4
18	AI-credit management	4
19	Interface clarity	3
20	System feasibility	4

Total = 81/100; mean = 4.05/5.

Appendix Table D2. Generated 3D quality assessment (10 items).

No.	Indicator	Score
1	Geometry quality	3
2	Structural completeness	3
3	Shape accuracy	3
4	Model detail	4
5	Surface quality	4
6	Multiview consistency	3
7	Visual conformity	3
8	Prompt/image correspondence	3
9	Suitability as a 3D asset	4
10	GLB output suitability	5

Total = 35/50; mean = 3.50/5.