



- (51) **International Patent Classification:**
G06N 3/0475 (2023.01) *G06Q 10/10* (2023.01)
- (21) **International Application Number:**
PCT/US2024/045382
- (22) **International Filing Date:**
05 September 2024 (05.09.2024)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
63/537,297 08 September 2023 (08.09.2023) US
- (71) **Applicant:** ZENFOLIO INC. [US/US]; 303 Twin Dolphin Drive, 6th Floor, Redwood City, CA 94065 (US).
- (72) **Inventors:** BARRACLOUGH, Keith; c/o Zenfolio Inc., 303 Twin Dolphin Drive, 6th Floor, Redwood City, CA 94065 (US). FERRIER, Michael; c/o Zenfolio Inc., 303

Twin Dolphin Drive, 6th Floor, Redwood City, CA 94065 (US). FAUST, George, Michael; c/o Zenfolio Inc, 303 Twin Dolphin Drive, 6th Floor, Redwood City, CA 94065 (US). LECHOW, Ross, Tyler; c/o Zenfolio Inc., 303 Twin Dolphin Drive, 6th Floor, Redwood City, CA 94065 (US).

(74) **Agent:** CRAWFORD, Robert, J.; Crawford Maunu PLLC, 1150 Northland Drive, Suite 100, St. Paul, MN 55120 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH,

(54) **Title:** METHODS AND APPARATUSES INVOLVING AUTOMATED TASK COMPLETION FOR WEBSITE REFINEMENT/CUSTOMIZATION

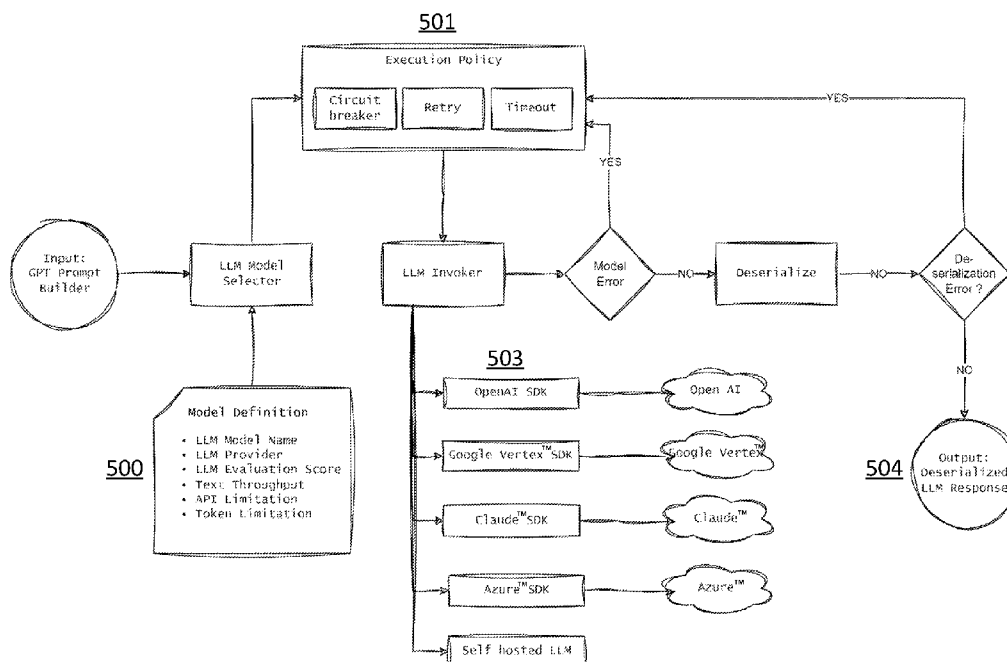


FIG. 5

(57) **Abstract:** In certain examples, methods and semiconductor structures are directed to a data-processing computer circuit and/or method to: determine a subset of multiple aspects or elements of an existing website to modify by analyzing user inputs, pertaining to desired changes in the existing website with respect to at least one of content and design; construct prompts for one or more large language models (LLMs) incorporating said user inputs and data structures characterizing the subset of multiple aspects or elements; drive one or more AI/ML algorithms of at least one computing-processor circuit, based on the constructed prompts, to output modifications of at least some of the subset of multiple aspects or elements; and generate a custom website, including the desired changes and based on the user input.

TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS,
ZA, ZM, ZW.

- (84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

**Methods and Apparatuses Involving
Automated Task Completion for Website Refinement/Customization**

BACKGROUND

[0001] Whether implemented manually or using high-technology tools, designing a website is often difficult. Becoming familiar with the types of data that typically goes into a data set for generating a website presents a significant challenge that is typically accompanied by a large learning curve. If one is attempting to use high-technology tools for generating aspects of the website, the challenge can be exponentially greater.

[0002] Another burdensome issue with designing a website is knowing how to make the appearance of the website attractive in terms of aesthetics, user comfort for its operations, and organizational efficiency. Further, a website design should be particular to the style of the entity (e.g., company, group or individual) that is to use the website. Among many other complications, the website needs to work well regardless of the type of computer-based communication tool which provides the UI (user interface) to access the website. This is true whether the computer-based communication tool is a desktop computer, a smart phone, or another type of user-engagement tool. Building, launching and refining customized websites traditionally requires extensive technical expertise involving manual efforts from web designers and developers. Users must communicate requirements, designers mockup page layouts, developers code the frontend and backend, and teams iterate often through multiple testing and feedback cycles. This complex, time-consuming and expensive process puts website creation out of reach for many individuals and businesses.

[0003] Moreover, as ongoing use of broadband data networks (e.g., the Internet) has continued to grow over the past decades, many decision makers behind website designs are becoming less familiar even with the basic technology specifics often used in generating websites and, at the same time, demand that such websites be designed in a manner that is highly customized based on their particular goals, preferences, needs and related uses.

[0004] Accordingly, there are needs for automated tools to develop websites in a manner that overcomes the above, and other, challenges, for instance, in a manner that can generate and execute tasks for a customized website aligned to a user's goals and preferences, without extensive manual intervention.

SUMMARY OF VARIOUS ASPECTS AND EXAMPLES

[0005] Various examples/embodiments presented by the present disclosure are directed to issues such as those addressed above and/or others which may become apparent from the following disclosure. For example, some of these disclosed aspects are directed to methods and devices that use or leverage from existing metadata, or other metadata provided by the entity seeking to customize and/or refine a website. Such aspects may be used in transforming user inputs, including or related to such existing or new metadata (and/or an existing website), automatically into such a customized and/or refined website. In more specific aspects, this automated transformation includes generating the full code, structure, content and/or assets for a complete customized and/or refined website (e.g., including media-related resources related to the user inputs) to align the website with goals and preferences with minimal and, in some instances, no manual intervention.

[0006] In certain specific examples, aspects of the present disclosure concern a system and related processes for upkeep of tailored websites with minimal human intervention, by employing natural language processing to drive a computer circuit configured with AI/ML (artificial intelligence and/or machine learning algorithm(s)) and in response, by automating creating, refining and/or optimizing one or more websites based on a set of high-level user input data. In one more-specific system or method, website code, structure, content and assets may be automatically synthesized so as to be tailored to a user's context by leveraging large pre-trained language models. According to another more-specific example, the system or method is implemented to enable one or a combination of: rapid generation of multiple design options; iterative feedback and refinement; reorganization of pages and galleries; analyzing, renaming, and recaptioning assets; content generation and proofreading; importing assets from external sources; and synthesizing designs from external sources. In this manner, the user is benefited by overcoming many complexities of website creation and enabled with automatic website refinement and maintenance without manual coding or design expertise. Users can rapidly deploy such customized websites for staying up-to-date through automated improvements over time.

[0007] In certain specific examples having related aspects or that may build on the above-discussed aspects, the present disclosure is directed to a data-processing computer circuit and/or method to: determine a subset of multiple aspects or elements (e.g., less than all of the multiple aspects or elements) of an existing website to modify by analyzing user inputs, pertaining to desired changes in the existing website with respect to at least one of content and design; construct prompts for one or more large language models (LLMs)

incorporating said user inputs and data structures characterizing the subset of multiple aspects or elements; drive one or more AI/ML algorithms of at least one computing-processor circuit, based on the constructed prompts, to output modifications of at least some of the subset of multiple aspects or elements; and generate a custom website, including the desired changes and based on the user input.

[0008] In more specific examples related to the above computer-implemented methods and/or computer-based devices and in connection with the present disclosure, it has been discovered that the best outcomes from the LLM model(s) result when the inputs used for processing of the data for the prompt construction, are (as an option) transformed from their native format (e.g., data structures) into a format that is conducive to an LLM input prompt such as one or a combination of text, attached files, and new merged/normalized data structures. In many example implementations in accordance with the present disclosure, this approach is advantageous in that by transforming the data from their native format as such, the process results in optimal outcomes from the deployment of the LLM model(s).

[0009] In certain specific examples having related aspects or that may build on the above-discussed aspects, the present disclosure is directed to a data-processing computer circuit and/or method to: rapidly generating diverse website designs and dynamically adapting them based on user feedback, by steps of processing user inputs, employing natural language processing, and iteratively steering custom websites, as selected, for design alterations. In a more specific example embodiment, this method involving rapid website generation includes: processing for a user, user inputs, existing user context, and external sources to generate a range of design variations for selection of one or more custom websites; employing one or more natural language processing techniques to solicit additional input or feedback from an entity corresponding to or on behalf of the user, and, in response, refining design options of the one or more custom websites; and enabling the entity real-time access to allow the entity to iteratively steer the one or more custom websites, as selected, for design alterations toward desired outcomes consistent with at least some of the user feedback and for generating one or more custom website designs.

[0010] In certain specific examples, the present disclosure is directed to aesthetic aspects (in whole or part, and with the same screenshot arrangement of sections and/or rearranged screenshot sections) of one or more of the GUIs such as shown in FIG. 1 (e.g., Tab 2, 101, etc.) and in FIGs. 6-11 (e.g., without some of the details and/or text such as in the smaller blocks).

[0011] The above discussion is not intended to describe each aspect, embodiment or every implementation of the present disclosure. The figures and detailed description that follow also exemplify various embodiments.

BRIEF DESCRIPTION OF FIGURES

[0012] Various example embodiments, including experimental examples, may be more completely understood in consideration of the following detailed description in connection with the accompanying drawings, each in accordance with the present disclosure, in which:

[0013] FIG. 1 is an example block diagram of a communications arrangement according to one example consistent with the present disclosure;

[0014] FIG. 2 is a diagram of an example design editor application service hosted on a cloud platform, according to certain exemplary aspects of the present disclosure;

[0015] FIG. 3 is a diagram of an example task system module, according to certain exemplary aspects of the present disclosure;

[0016] FIG. 4 is a diagram of an example request and response cycle, according to certain exemplary aspects of the present disclosure;

[0017] FIG. 5 is a diagram of an example model factory module, which intelligently selects one or more appropriate LLMs and providers for each content generation task, among other aspects, according to certain exemplary aspects of the present disclosure;

[0018] FIG. 6 is a diagram of an example design editor module, according to certain exemplary aspects of the present disclosure;

[0019] FIG. 7 is a diagram of an example approach for implementing website design changes via a dialog-type communications channel, according to certain exemplary aspects of the present disclosure; and

[0020] FIGs. 8-11 are screen shots of a display corresponding to a website, or aspects thereof, customized according to certain exemplary aspects of the present disclosure with: FIGs. 8 and 9 showing example screenshots to illustrate aspects concerning user communications surrounding a step of prompting a user (via a user device) regarding a description and/or type of background for a website, and with FIGs. 10-11 showing example screenshots to illustrate aspects, according to the present disclosure, to enable a user to change logo text.

[0021] While various embodiments discussed herein are amenable to modifications and alternative forms, aspects thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit

the disclosure to the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the scope of the disclosure including aspects defined in the claims. In addition, the term “example” as used throughout this application is only by way of illustration, and not limitation.

DETAILED DESCRIPTION

[0022] Aspects of the present disclosure are believed to be applicable to a variety of different types of apparatuses, systems and methods involving devices characterized at least in part by use of natural language modeling and extraction of user input and resources (e.g., media-related data and/or attributes) for driving large language models (LLMs) and AI/ML algorithms, and therein automatically determine and provide one or more optimal sets of content for website pages and/or content blocks tailored to needs of a user (e.g., company, group, individual, etc.). In this context, “optimal” refers to or includes a proposed best set of content given the available user inputs and resources, or an improved version over an earlier version in an iteration. While the present disclosure is not necessarily limited to such aspects, an understanding of specific examples in the following description may be understood from discussion in such specific contexts.

[0023] Accordingly, in the following description various specific details are set forth to describe specific examples presented herein. It should be apparent to one skilled in the art, however, that one or more other examples and/or variations of these examples may be practiced without all the specific details given below. In other instances, well known features have not been described in detail so as not to obscure the description of the examples herein. For ease of illustration, the same connotation and/or reference numerals may be used in different diagrams to refer to the same elements or additional instances of the same element. Also, although aspects and features may in some cases be described in individual figures, it will be appreciated that features from one figure or embodiment can be combined with features of another figure or embodiment even though the combination is not explicitly shown or explicitly described as a combination.

[0024] Exemplary aspects of the present disclosure involve computer-implemented methods and circuit-based arrangements directed to assembling automatically generated content into a website structure that is optimal to a particular user, in that the website structure is tailored to goals of the particular user and is consistent with attributes of media assets of the user.

[0025] Consistent with the above aspects, such a manufactured device or method of such manufacture may involve aspects presented and claimed in U.S. Provisional Application Serial No. 63/537,297 filed on September 8, 2023, to which priority is claimed. To the extent permitted, such subject matter is incorporated by reference in its entirety generally and to the extent that further aspects and examples (such as experimental and/more-detailed embodiments) may be useful to supplement and/or clarify.

[0026] Consistent with the present disclosure, computing data-processing devices and/or methods may be used for producing website content for user websites (as a website used by one or more “users” referring to entities such as companies, groups and/or individuals) which rely on images such as photography and art studios, and other entities benefited by frequently altering their websites based on attributes of metadata (e.g., multimedia metadata such as images, videos, hyperlinked sites, articles, screenshots, etc.).

[0027] Before turning to the drawing to be discussed in detail below, it is noted that each of the above (briefly-described) examples is presented in part to illustrate aspects of the present disclosure, as may be recognized by the foregoing discussion. As noted with the above discussion, and in connection with further more specific examples described below and in connection with the figures, such aspects include reference to AI and/or ML with AI and ML referring to artificial intelligence and machine learning, respectively, and appreciating that AI may be used interchangeably with ML, and to LLMs which are AI programs that use deep learning to analyze and understand text. An LLM is trained on large amounts of data (e.g., millions of gigabytes of text and/or metadata) from the Internet or from other large data sources, to recognize and generate phrases, interpret languages, discern contexts being used in communications, and understand how words, sentences, intonations and characters work together.

[0028] Turning now to the drawing, FIG. 1 shows a communications arrangement according to one example consistent with the present disclosure, which illustrates how, in certain examples a computer-based method and/or system may be implemented to generate a refined and/or customized website. This may be achieved by: determining a subset of multiple aspects or elements of an existing website to modify by analyzing user inputs, pertaining to desired changes in the existing website with respect to at least one of content and design; constructing prompts for one or more large language models (LLMs) incorporating said user inputs and data structures characterizing the subset of multiple aspects or elements; driving one or more AI/ML algorithms of at least one computing-processor circuit, based on the constructed prompts, to output modifications of at least some of the

subset of multiple aspects or elements; and generating a custom website, including the desired changes and based on the user input.

[0029] As shown in FIG 1, a user (and/or related entity) providing or receiving the user inputs, may be enabled to generate automated tasks through an application or web browser with an intuitive user interface (UI) on a desktop computer, mobile device, or other types of input and display devices 100. The user interacts with the UI, and the system allows users to generate websites through an intuitive user interface (UI) 101 to provide tasks, goals, and preferences to manipulate their site. The browser sends requests for such information to the user over the internet, passing through a content delivery network 102 (CDN) like Cloudflare® for security, caching and traffic management.

[0030] The requests arrive at the core application logic 103 hosted on a cloud platform, some examples of which include Microsoft Azure™, Heroku®, Kubernetes®, cloud servers, data center servers, and/or embedded devices. The application leverages various AI and ML services 104 to generate the website. The logic 103 sends user context to pre-trained LLMs such as OpenAI's GPT-3.5 or GPT4, Anthropic's Claude®, and Google Vertex AI® for generating text, code and assets. The application also utilizes computer vision APIs like Amazon Rekognition® and Google Vision® for intelligent media analysis.

[0031] In various specific embodiments, the requests arrive at the core application logic circuitry (or simply “logic”) such as depicted at block 103 (and below as module 203 of FIG. 2. The logic may be hosted by computing data-processing circuitry integrated with a set of one or more servers on a cloud computing platform. The logic leverages one or various AI/ML (AI and/or ML) services along with natural language modeling and media extraction tools to generate a refined/customized website. The media extraction tools, as part of the logic, are to extract resources (such as citations, images, videos, third-party sites, themes, and attributes of the same) as may be identified through the dialog with the user or an entity identified on behalf of the user. In one more specific example, the logic includes one or more of the following computer-based modules (e.g., implemented by the logic carrying out CPU-programming instructions): a site building services module, a media analysis service (optionally with a photorefine module), and a database. The logic sends user context to pre-trained large language models (LLMs) such as OpenAI's GPT-3.5 or GPT4 111, Anthropic's Claude 109, Azure AI 110 and Google Vertex AI 112 for generating text, code and assets. One or more such LLMs can be used for different purposes and/or can be differentiated in that they are disparately managed; for example, different ones of the LLMs may produce better and/or different results in terms of generating certain types of texts, video, and/or

graphics, and in some cases with different results which are useful for the logic to assess and compare to the user's input and media-based assets (e.g., for storage, categorization and/or previewing by the user or the above-noted user-related entity). In such examples, by having the logic configured to act on behalf of one or multiple different and/or disparately managed LLMS, the logic is effectively converted into an ultra-intelligent user-customized node through which the user may generate an ideal website for particular needs by providing minimal input and access to the media resources (e.g., images, videos and the like).

[0032] The core application compiles and processes the LLM and vision API outputs to assemble a fully-functional website personalized to the user's needs. The generated website is rendered back to the user's browser for preview. The user can iteratively improve the site through feedback prompts which trigger new generation cycles. All communications occur securely over TLS connections. The system automatically handles scaling of traffic and cloud resources to maintain performance.

[0033] This automated communication flow allows users to attain custom tailored websites built specifically to their business needs using cutting-edge AI without technical expertise. The intelligent services generate high-quality content, assets and code matching user requirements in a fast, automated manner.

[0034] FIG. 2 shows a Design Editor application service (via a module, Design Editor) hosted on a cloud platform 200. The Design Editor provides the user interface for collecting inputs from the user through UI elements, a conversational chatbot, voice commands, gestures via a camera, or other input sensors 201. The Design Editor maintains a database storing information about the user's website and preferences 202.

[0035] The Design Editor receives the user input data, and existing site configuration from the database, as the basis for building the customized site. The Design Editor also retrieves any required user media assets, such as images or videos, from connected media storage services.

[0036] With the user data and media gathered, the Design Editor invokes a request to a module (e.g., Task System 203 of FIG. 2) that assembles the final prompt to be sent to the LLM 204. This module (Task System) assembles the final prompt to be sent to the LLM 204. This includes the system instructions, rules, user context, and preferences.

[0037] The modular separation of components for user input, media handling, design variable definition, prompt creation and building enables automatically constructing a customizable website with minimal human input through leveraging predefined blocks tailored by LLM generated content.

[0038] FIG. 3 shows one example of the Task System module responsible for generating the style of a site and individual pages by leveraging a Large Language Model (LLM). The example follows a multi-step process:

[0039] First, the Design Editor 300 loads the user context, including properties, parameters, schemas, and descriptions of the user's existing design variables, and all other data related to their account, site, and assets.

[0040] Next, the Task Pre-Processor 301 combines the system prompt, user prompt, and user context.

[0041] This assembled prompt is sent to the chosen LLM provider 302. The system waits for the complete generated response, as LLMs produce content via continuous generation streams.

[0042] Once the full response is received, the Task Post-Processor 303 deserializes and validates the content to ensure proper formatting and completeness.

[0043] The validated LLM response is then used to construct previews of the changes 304. This type of preview can be accepted by the user, which saves the changes to the database.

[0044] FIG. 4 shows the request and response cycle. In specific example embodiments, the system carefully constructs prompts as at block 400 which are sent to one or more LLMs and with the expectation of specific response formats to enable website content generation tailored to the user's context. According to one example process in this regard and as depicted in FIG. 4, the prompt contains a system message 401 that defines the role of the LLM as a web building expert, (e.g., unlike a generic dialog box or a generic chatbot). This approach guides the LLM's mode of response. A JSON (JavaScript Object Notation) schema is injected that specifies the expected response structure so the output can be deserialized into strongly-typed data. Relevant user context and preferences are added to steer the LLM into suggesting personalized content fitting the user's needs. Sample prompt-response pairs are provided for few-shot learning 402. This improves response accuracy and conformance to the expected format. The prompt is assembled from sub-prompts for each task that requires content 403. These contain the property description, question for the LLM, and any expected response schema.

[0045] In response to the generation of the prompt, at block 404 the LLM outputs a field per task 405, adhering to the requested JSON structure. This phased prompting approach with guiding context and pre-training enables the LLM to produce website-specific content that

can be reliably deserialized into tasks for automated construction tailored to the user's specifications.

[0046] The system transforms generic LLM capabilities into an intelligent web content generation agent by strategically structuring prompts and responses. The modular prompt engineering unlocks AI for customized website building.

[0047] According to the present disclosure, different approaches may be used to transform and combine the different user inputs (or data inputs) and depending on the approach used, the outcome from the LLM model(s) may differ and/or be more or less aligned with such inputs (and/or related user-input goals). Based on significant experimentation in connection with the present disclosure, it has been discovered that the best outcomes from the LLM model(s) result when the user inputs used for processing of the data for the prompt construction, are (as an option) transformed from their native format (e.g., data structures) into a format that is conducive to an LLM input prompt such as one or a combination of text, attached files, and new merged and/or normalized data structures. In many example implementations in accordance with the present disclosure, this approach is advantageous in that by transforming the user input data from their native format as such, the process results in optimal outcomes from the use of the LLM model(s).

[0048] FIG. 5 shows the Model Factory (as one such module), which intelligently selects one or more appropriate LLMs and providers for each content generation task, optimizing for accuracy, cost, and/or speed. The Model Factory may consult the Model Definition registry 500 containing metrics on available LLM providers and models to choose the optimal one. Based on the complexity of the content needed, the Model Factory chooses an optimal model balancing accuracy, cost, and throughput. Simple content uses faster, affordable models, whereas complex requests use higher quality models. For example, for simple content like an "About Me" block, the Model Factory can select a fast, affordable model like GPT-3.5 with high throughput. For more complex tasks such as choosing artistic layouts, the Model Factory can leverage a slower but higher accuracy model like GPT-4 despite the higher cost.

[0049] The execution may extend through a Task Execution Policy 501 with "circuit breakers" to alternate models if errors occur frequently, retries for transient network errors, and timeouts to prevent blocked responses. This handles variability in LLM provider uptimes. The policy maximizes successful content generation with automated provider failover.

[0050] Once a response is received, the response is validated through deserialization. Invalid responses are handled via the execution policy by retrying with the same or different models 503.

[0051] For each prompt, this cycle determines the right LLM and iteratively improves resilience until valid content 504 is produced. This way the system dynamically matches the capabilities of different LLM models to the complexity of each content task. The system can use cheaper models where possible reduces costs, while leveraging accurate models for complex generation to maintain output quality.

[0052] The Model Factory may be implemented to optimally combine different LLMs into an ensemble tailored to the requirements of each website building task. This strategically applies AI to automate personalized web content creation.

[0053] FIG. 6 shows the design editor 600, where changes to the design and content can be generated by a user specifying their goal in natural language. The user accesses the design editor, and a dialog 601 asks for their input, voice commands, gestures, or other sensor inputs. After describing their goal and submitting it, the preview pane 602 is updated to show the changes. They can be iterated on with further prompts, or accepted and saved as the new design.

[0054] FIG. 7 is a diagram of an example approach for implementing website design changes via a dialog-type communications channel. In FIG. 7, a task agent design editing module 700 demonstrates making site design changes via dialog. The chatbot offers Large Language Model-generated page variations to select from, integrates requested features like palette generation and image search, and handles publishing and social promotion of the finalized page, enabling rapid website (via a mobile) creation. In a more specific example, such a process may be carried out by: rapidly generating diverse website designs and dynamically adapting them based on user feedback, by steps of processing user inputs, employing natural language processing, and iteratively steering custom websites, as selected, for design alterations.

[0055] In a more specific example embodiment, such rapid website generation includes: processing user inputs for a user/user-related entity, along with existing user context and external sources to generate a range of design variations for selection of one or more custom websites; employing one or more natural language processing techniques to solicit additional input or feedback from an entity corresponding to or on behalf of the user, and, in response, refining design options of the one or more custom websites; and enabling the entity real-time access to allow the entity to iteratively steer the one or more custom websites, as selected, for design alterations toward desired outcomes consistent with at least some of the user feedback and for generating one or more custom website designs. In different example implementations and protocols, such rapid generation occurs in sequence; as examples, with

not more than three or four alterations or inputs in terms of user input data, with only one or in some cases only two alterations or inputs in terms of user input data, and with no such alterations or inputs (or other human intervention) via the user interface of the user device (e.g., smartphone or desktop CPU).

[0056] In certain cases, such rapid generation occurs in sequence by limiting the time and/or amount of user intervention and/or space-based aspects of a website (e.g., quadrant, encircled portion around a logo, etc.), via the logic, between generating such alternative websites. As examples, this may be carried out by the logic indicating at the outset the categories or types of changes that may be provided, and/or limiting the number of such changes, and/or setting a time limit on receiving any further changes (after accepting a minimal threshold number). In this way, alterations or inputs in terms of user input data is minimized to enable the logic to rapidly generate such websites in response to such refining of design options of the one or more custom websites via selected alterations toward the desired outcomes.

[0057] As shown in FIG. 7, a task agent content-generation module 701 shows editing an existing page by allowing the user to request batch changes like modifying text content, and context-aware content generation. The chatbot uses AI to generate content based on the user's context, including time and location.

[0058] A task agent external sourcing module 702 exemplifies creating a page by importing data from one or more external sources, including text and images. The AI platform automatically imports semi-structured data, and also crawls that data for further external sources to use.

[0059] In certain of the above examples, the conversational interface is configured to mirror natural human interaction, collecting key inputs while handling complexity behind the scenes via intelligent automation. Users achieve custom site creation, editing and content generation on mobile devices without needing to navigate complex dashboards or options. This exemplifies the system's technical approach of infusing an intuitive chat-based interface powered by AI to liberate website building from desktop-centric constraints. Chatbot conversations may be used to drive strategic website updates and generation by channeling natural language instructions into focused under-the-hood optimizations.

[0060] In the discussion immediately following, several specific example embodiments are disclosed to aid in the understanding of how aspects and embodiments may be implemented at a more detailed level and/or in specific example (non-limiting) applications.

[0061] Example Embodiment 1: Design Editor

[0062] In order for a user to create a website, users can use the Design Editor to make changes to the data and design of their website. The Design Editor prompts the user via chatbot UI for their desired changes.

[0063] Additional embodiments could include:

- Creating and reorganizing all pages.
- Page-specific modifications to galleries, blogs, stores.
- Site-wide changes to settings, such as SEO values, sitemaps, contact information.
- Changing subscription settings, such as starting a new subscription, upgrading or downgrading an existing subscription, and changing payment information.

[0064] The structured input is stored to initialize the task completion agent for the user's specific needs. The system safely manages page creation and modification, user and site settings, subscriptions and payment, and media asset upload and manipulation, while collecting necessary context.

[0065] Example Embodiment 2: Rapid, Iterative, Multi-Option Redesign

[0066] An approach to rapidly generate diverse website designs and promptly adapt them based on user feedback is provided. Inputs can include existing user context and external sources, to create a variety of design variations for the user to choose from. Natural language processing techniques are employed to prompt for further user feedback and suggestions. The user can steer real-time previews towards their eventual goal. Generated designs can also be stored and cataloged to serve as a knowledge base for future design projects and continuous improvement.

[0067] Example Embodiment 3: Batch Asset Categorization

[0068] Once the user uploads media assets like images and videos, the files are sent to the AI Vision Client module for content analysis. Computer vision algorithms extract a variety of attributes including:

- Prominent colors in images for palette generation
- Detecting faces, objects and scenes through object recognition
- Assessing technical qualities like sharpness, resolution, brightness
- Identifying regions of interest or focal points in images
- Image similarity

[0069] The Media Processing Unit selects the optimal media assets to include based on analysis results and specified website goals. Only high quality, relevant images are chosen. Focal points are identified by detecting faces, products or other salient regions.

[0070] Additional embodiments could include:

- Categorizing images (portraits, products, nature, etc.)
- OCR text extraction from images for alt text
- Detecting logos, landmarks or custom objects

[0071] The extracted visual attributes are structured into metadata that informs prompt construction for the language models. The prompts describe images contextually such as "photo of bride and groom at wedding" or "image of person rock climbing outdoors".

[0072] An alternative embodiment uses the a client application ("PhotoRefine Client App") that runs locally on the user's device (Figure 10). This allows pre-processing media before uploading to analyze content, quality and focal points. The app reduces cloud costs and time by distributing workload.

[0073] In an embodiment, the PhotoRefine service can also be deployed in a self-hosted cloud environment managed by the user. This provides more control over data security and privacy compared to using a third-party public cloud provider. The self-hosted option also allows customization of the media analysis models and pipelines to better fit the user's specific use cases and datasets. Furthermore, running PhotoRefine as a self-hosted cloud service enables easy scaling of the media analysis workload by adding more servers to the back-end processing cluster. This allows the system to handle large volumes of media assets efficiently.

[0074] The multimedia analysis results are used throughout the system to optimize website design and content generation suited to the provided visual assets. Additional user feedback loops can be incorporated to improve extracted image attributes.

[0075] Example Embodiment 4: Content Generation using LLMs

[0076] The system utilizes a library of predefined tasks with parameters to generate instructions, such as making design changes, writing a blog post, or making subscription changes. Additionally, the system can construct completely new page types tailored to the user's website goals and context. Based on analysis of the user preferences, target audience and content needs, the system determines optimal parameters for custom pages:

- Number of columns, rows, layout format
- Style including colors, fonts, spacing
- Appropriate media types (image, video, etc.)

- Categorization and groupings of media
- Length and topics of auto-generated text content

[0077] These parameters are used to construct inputs such as prompts and supplementary data for the large language model (LLM) describing the context and requirements for the custom content. The LLM response populates the component properties to generate content for that specific user. Generated content approved by the user can be reused to further train and generate more relevant content in the future.

[0078] The flexibility to construct new content optimized for each user, combined with assembling predefined content, allows generating highly customized website designs and content flows. The system can build unique sites that match user needs more precisely than relying solely on pre-existing templates.

[0079] The system mixes and matches auto-generated content with configurable predefined content to balance flexibility, personalization and rapid iteration for users. This embodiment describes the core capability to automatically construct new content using AI that can extend system content on demand.

[0080] Example Embodiment 5: External Source Importer

[0081] The system can be tasked to fetch from external sources, and used to populate the user's existing site. This could include:

1. Social Media Feeds: Allowing users to import content from their social media profiles (such as Instagram, Facebook, Twitter) to help them curate and display their latest updates, photos, and interactions directly on their site.
2. Online Marketplaces: If users sell their work on platforms like Etsy, eBay, or other online marketplaces, the system could import product listings, invoices, and images, simplifying the process of showcasing and selling products on their site.
3. Image Hosting Services: Importing from popular image hosting services like Flickr, Imgur, or Google Photos can let users easily pull in images and captions from their existing accounts and galleries.
4. Behance, Dribbble, and Other Portfolio Platforms: Creative professionals often maintain profiles on other portfolio platforms. Allowing them to import their work from platforms like Behance and Dribbble can save time and ensure consistent content across different platforms.
5. Blogs and Articles: If users maintain a blog on platforms like WordPress, Medium, or Blogger, the ability to import blog posts and articles can help

them maintain a cohesive online presence by integrating their written content into their site.

6. **Email Contacts from Mailing Lists:** Importing from popular email marketing and customer relationship management (CRM) platforms, such as MailChimp, Constant Contact, or HubSpot, allows users to import their email contacts can enhance their outreach and engagement efforts. This feature would enable users to seamlessly import their subscriber lists and contact databases, making it easier to stay connected with their audience and promote their work. Users could then incorporate email signup forms or newsletter subscriptions directly on their site, leveraging their imported contacts for marketing and communication purposes.
7. **Professional Resumes:** Enabling users to import their professional resumes directly from platforms like LinkedIn can simplify the process of showcasing their career achievements and work experience on their site. This feature could save users time and effort by automatically populating their "About Me" or "Resume" pages with relevant information, including education, work history, skills, and accomplishments.
8. **Video Sharing Platforms:** Integration with platforms such as YouTube, Vimeo, or TikTok could enable users to showcase their video content directly within their site.
9. **SoundCloud or Spotify:** Musicians and audio creators might want to import their tracks or podcasts from platforms like SoundCloud or Spotify to feature their audio content on their site.
10. **Google Drive or Dropbox:** Enabling users to import files from their cloud storage accounts can make it easy to share documents, presentations, and other digital assets on their site.
11. **Stock Photo Websites:** Integrating with stock photo libraries could allow users to license and display stock images directly on their site.
12. **RSS Feeds:** Allowing users to import content from external RSS feeds can help them keep their audience updated with news, articles, or other content relevant to their niche.
13. **Event Calendars:** If users have event calendars on platforms like Eventbrite or Meetup, importing these events into their site could help them promote their activities.

[0082] Example Embodiment 6: External “Style Transfer” Importer

[0083] The system allows users to make a “style transfer” import of external sources, where the user inputs a target external source, and the system applies similar themes, layouts, colors, typography, and other design elements. Here are some cases where this feature could be used:

1. Inspiration from Competitors: Users often look at their competitors' websites for design inspiration. With style transfer, they could analyze the design elements, color schemes, typography, and layout of successful competitor sites and apply similar styles to their own portfolios.
2. Client Branding Alignment: Designers creating portfolios for clients could import the branding and visual style of the client's existing website. This would ensure a cohesive look and feel across the client's online presence, from their main website to their portfolio.
3. Matching Print Collateral: For creatives who have printed promotional materials like business cards, brochures, or banners with a distinct style, the style transfer feature could replicate that style on their online portfolio, maintaining consistency across both online and offline branding.
4. Uniform Design Language: Businesses with multiple websites or microsites catering to different audiences might want to maintain a unified design language. The style transfer feature could help them apply consistent styles across these different sites.
5. Theme Adaptation: Users who find a web design theme they like on another platform could use the style transfer feature to adapt the visual elements of that theme onto their site, ensuring a consistent look even if the original theme is not natively available on our platform.
6. Cross-Promotion on Multiple Platforms: Content creators and influencers who have profiles on various platforms could use style transfer to replicate the look of their primary website or social media profile on their site, strengthening their personal brand.
7. Event-Specific Styling: If users are creating portfolios for events, conferences, or exhibitions, they could import the visual style of the event's official website or branding materials to create a seamless experience for attendees.

8. Celebrating Trends: Designers who want to experiment with popular design trends can import styles from websites that showcase those trends effectively. This way, they can quickly adopt new aesthetics and techniques.
9. Adapting Design Awards: Professionals who have received design awards might want to replicate the styling of the awarding organization's website, creating a sense of authenticity and validation on their own portfolio.
10. Niche Aesthetics: For creatives working in specific niches, such as vintage design, futuristic aesthetics, or minimalism, the style transfer feature could help them quickly adopt the visual elements associated with their chosen niche.
11. Seasonal Themes: Users who update their portfolios to match different seasons or holidays could use style transfer to quickly refresh the design according to the occasion.
12. Localized Design Trends: Importing styles from websites popular in a specific region or culture could help users tailor their portfolio's aesthetics to appeal to a local audience.

[0084] In all cases, it is important for the style transfer feature to provide options for customization and fine-tuning so that users can adapt imported styles to their preferences and ensure that the transferred styles blend harmoniously with their existing content and branding.

[0085] Example Embodiment 7: Autonomous Agent for Managing Data and Configuration

[0086] The autonomous agent could perform tasks in the background for users, enhancing the user experience and streamlining various aspects of managing and maintaining a user's site. Here are some tasks that the autonomous agent could potentially perform:

1. Content Synchronization: The autonomous agent (or sometimes agent) could automatically synchronize (or sync) content from external platforms like social media, portfolio sites, and blogs, ensuring that the user's site remains up-to-date without manual intervention. This automatic content synchronization may be triggered in different ways (e.g., via a regular calendar schedule, by event driven occurrences such as change of logo, citation or product announcement referenced on the customized/refined website or a related website as indicated in a user-profile that is regularly monitored).

2. **Media Optimization:** The agent could optimize and resize images and media files uploaded to the site to ensure fast loading times and optimal display on various devices.
3. **Security Monitoring:** The agent could continuously monitor the website for potential security vulnerabilities, malware, or suspicious activities and take proactive measures to protect the site and user data.
4. **Performance Enhancement:** The agent could analyze website performance metrics and implement optimizations, such as caching, compression, and minification, to ensure optimal speed and responsiveness.
5. **SEO Improvements:** The agent could analyze the website's content and structure to provide SEO recommendations, such as optimizing metadata, suggesting keywords, and generating sitemaps.
6. **Backup and Restoration:** Regular automated backups of the website's content, design settings, and configurations could be performed, allowing for easy restoration in case of data loss or issues.
7. **Form Submissions Handling:** If the website includes contact forms or other interactive elements, the agent could manage form submissions, organize inquiries, and notify the user about new messages.
8. **Content Recommendations:** Based on the user's portfolio content and industry trends, the agent could suggest ideas for new content, such as blog topics, projects to highlight, or portfolio updates.
9. **Broken Link Detection:** The agent could periodically scan the website for broken links and alert the user to fix or update them, ensuring a seamless browsing experience for visitors.
10. **Traffic Analysis:** The agent could generate automated traffic reports and insights about visitor behavior, helping users understand their audience and tailor their content accordingly.
11. **Update Notifications:** The agent could monitor platform updates, plugins, and themes, and notify the user about available updates to keep the website running smoothly and securely.
12. **E-commerce Management:** The agent could manage inventory, track orders, send shipping notifications, and handle customer inquiries.

13. Integration Support: The agent could assist users in integrating third-party tools and services, such as domain registration, email registration, social media sharing, analytics tracking, or payment gateways.

14. Feedback Collection: The agent could prompt visitors to provide feedback or testimonials and assist the user in displaying these endorsements on their website.

[0087] By performing these tasks autonomously, the agent could help users save time, reduce manual effort, and ensure that their sites are optimized, secure, and engaging for visitors.

[0088] Example Embodiment 8: Data Discovery and Inference for Autonomous Content Integration

[0089] The autonomous task agent is equipped with advanced web crawling and data inference capabilities to automatically discover and integrate relevant user-related data from the web into the user's site. The agent systematically explores the web for various sources of data associated with the user, such as social media profiles, blogs, articles, and other online content. Upon discovering relevant data, the agent employs sophisticated inference mechanisms to determine the most suitable manner of integrating this data into the user's portfolio.

[0090] For instance, if the agent identifies an Instagram account associated with the user, the agent extracts images and relevant metadata. Utilizing image recognition and contextual analysis, the agent deduces the optimal placement for these images, intelligently creating gallery pages or updating existing ones. Similarly, if the agent finds articles authored by the user on external blogs, the agent extracts the content, links, and associated media. By leveraging natural language processing, the agent might suggest creating a dedicated section to showcase the user's written work or even integrate the articles into existing portfolio pages, maintaining a consistent narrative.

[0091] The embodiment includes continual monitoring to stay updated with the user's online presence and any new data. As new sources are discovered, the agent employs learning algorithms to adapt its inference strategies, enhancing its ability to accurately interpret and incorporate diverse types of data. This holistic approach streamlines the process of content integration, relieves the user from the burden of manual curation, and ensures that the portfolio remains current, comprehensive, and aligned with the user's evolving online presence.

[0092] Example Embodiment 9: Autonomous Agent for Pushing Updates to Multiple External Destinations

[0093] The autonomous agent can push data and assets from a user's site to external third-party sites, which opens up various opportunities for expanding the reach and impact of the user's content. Here are some use cases for this functionality:

1. Social Media Posting: The agent could automatically push new portfolio updates, blog posts, or projects to the user's social media profiles, ensuring a consistent online presence and saving the user time on manual posting.
2. Cross-Platform Promotion: Pushing content to external platforms like Behance, Dribbble, or other portfolio sites could help users reach a wider audience by showcasing their work on multiple platforms simultaneously.
3. E-commerce Integration: If the user is selling products on their site, the agent could push new products, promotions, and updates to e-commerce platforms like Shopify or Etsy to broaden their sales channels.
4. Content Syndication: The agent could syndicate blog posts or articles to content distribution networks or news aggregator sites, increasing the visibility of the user's expertise and driving more traffic back to their site.
5. Email Marketing: Automatically pushing new content to email marketing platforms like MailChimp or Constant Contact could help users maintain engaged subscriber lists and consistently send newsletters or updates.
6. Video Sharing: For users who create video content, the agent could push videos to platforms like YouTube or Vimeo, broadening the audience reach beyond their site.
7. Community Engagement: Pushing portfolio updates to online forums, niche communities, or industry-specific platforms could foster engagement and discussions around the user's work.
8. Event Promotion: If the user participates in events, exhibitions, or webinars, the agent could push event details and registration links to relevant event listing platforms.
9. Local Directories: Pushing business information to local business directories, such as Google My Business or Yelp, could help users attract local clients and customers.

10. Guest Blogging: If the user contributes guest posts to external blogs or publications, the agent could push the article content along with a link back to their portfolio.
11. Collaborative Platforms: Pushing collaborative projects or content to platforms like GitHub, Figma, or Google Drive could help users collaborate with others while maintaining a central hub on their site.
12. Portfolio Diversity: For freelancers or professionals with a diverse skill set, pushing relevant content to specialized platforms (e.g., GitHub for developers, SoundCloud for musicians) showcases their expertise in different areas.
13. Crowdfunding Campaigns: Pushing content related to crowdfunding campaigns to platforms like Kickstarter or Indiegogo can help users gather support and funding for their projects.
14. Industry-specific Sites: Pushing content to industry-specific platforms or directories can help users gain recognition within their niche and connect with peers and potential clients.
15. Press and PR: The agent could push updates to press release distribution services or PR platforms, increasing the chances of media coverage for the user's achievements.
16. Photography Software Integration: Pushing images or albums directly to photography software like Lightroom can streamline the workflow for photographers, allowing them to quickly access and edit their portfolio content.
17. Mobile Album Sync: Pushing portfolio updates to albums on mobile devices like iPhones can help users have quick access to their work on the go, making it easier to showcase their portfolio in person.

[0094] The autonomous agent possesses the capability to make intelligent decisions about content distribution based on the nature of the data it encounters. Leveraging advanced algorithms and pattern recognition, the agent can discern the characteristics of the content, such as identifying images as illustrations or products. When presented with an illustration, the agent can intuitively direct the illustration (or data and/or attributes therefrom) towards platforms optimized for showcasing visual artistry, such as illustration-specific websites or communities. Similarly, in the case of images recognized as products, the agent exports them towards e-commerce platforms where they can seamlessly integrate with existing inventory or storefronts. By tailoring its actions to the content type, the agent strategically disseminates

a user's work across a multitude of platforms while respecting their preferences and maintaining an overarching coherent online presence.

[0095] It is important to ensure that the autonomous agent provides users with control over where and how their content is pushed, allowing customization and flexibility to tailor the content for each platform. Additionally, the agent should respect privacy and permissions, obtaining user consent before pushing content to external sites.

[0096] Example Embodiment 9: Task Performance Optimization

[0097] The task agent measures task performance to optimize future task success rates and ensure user satisfaction. The system integrates quality indicators into the task agent's decision-making framework, allowing the system to learn from previous task outcomes and improve its performance over time.

1. Data Collection and Feedback Integration: The task agent receives task requests from users and executes these tasks in accordance with predetermined criteria. For each completed task, the user provides feedback, sourced from their actions (e.g., task acceptance or rejection) or other mechanisms (such as explicit feedback prompts). The quality indicators are derived from this feedback and provide insights into the user's satisfaction with the task outcome.
2. Quality Indicator Analysis: The system collects and analyzes the quality indicators associated with completed tasks to determine patterns, trends, and correlations between task attributes and successful outcomes. Quality indicators may include user ratings, completion times, user engagement, or specific user preferences related to the task's execution.
3. Adaptive Decision-Making: The technology employs machine learning algorithms to process the analyzed quality indicators and adapt the decision-making process of the autonomous task agent. The agent learns to prioritize tasks and make decisions that are more likely to lead to successful outcomes based on historical data.
4. Task Assignment Optimization: With the refined decision-making process, the task agent becomes more proficient in selecting tasks that align with user preferences and previous successes. The agent can utilize inferred quality indicators to match tasks with user profiles and effectively manage task execution strategies.

5. Iterative Improvement: As the task agent completes more tasks and accumulates additional quality indicator data, the task agent continually refines its decision-making process. The agent can dynamically adjust its task execution strategies based on real-time feedback, leading to improved task success rates and enhanced user satisfaction.

[0098] Example Embodiment 10: Voice Interface

[0099] The use of traditional chatbot interfaces may still pose limitations in terms of accessibility, user preference, and user engagement. These limitations are addressed by an alternative voice interface that allows users to interact with the task agent through voice commands, with this process:

1. Voice Input Reception: The system incorporates a voice recognition module that captures and processes user voice input. Users can issue commands or queries to the task agent using natural language voice interactions.
2. Intent Interpretation: Utilizing natural language processing (NLP) algorithms, the system interprets the user's voice input to derive intent and understand the context of the user's request. The system can identify keywords, phrases, and patterns to determine the desired task.
3. Task Execution: Based on the interpreted user intent, the system triggers the appropriate task execution. The task agent retrieves relevant data, assets, or information and performs the task autonomously.
4. Dynamic Interaction: The technology enables users to have dynamic back-and-forth conversations with the task agent using voice commands, allowing users to clarify or refine their requests in real-time.

[00100] The voice interface provides accessibility to users who may have challenges with text-based interfaces, making the task agent more inclusive. The voice interactions mimic human conversations, making the interaction more engaging, intuitive, and user-friendly. The interface also enables users to interact with the task agent hands-free, contributing to convenience and efficiency.

[00101] Example Embodiment 11: User Data Protection

[00102] To ensure that the task agent does not overwrite or modify data or content that the user wishes to preserve, several features and safeguards can be implemented in the system:

1. Content Locking: Allow users to manually lock specific content items, such as projects, images, or pages, to prevent the task agent from making changes to

them. Locking could be initiated through a simple toggle or command, ensuring that the protected content remains untouched.

2. **Version History:** Implement a version control system that maintains a history of changes made by the task agent. Users can revert to previous versions if they find any unintended modifications, giving them the ability to restore their content to a desired state.
3. **Selective Task Approval:** Introduce a mechanism where the task agent provides users with a preview of proposed changes before execution. Users can then approve or reject each task individually, ensuring that only authorized modifications are made.
4. **Task Simulation:** Allow users to simulate the outcome of a task before actual execution. This preview enables users to assess potential changes and prevent any unwanted alterations.
5. **Content Marking:** Enable users to label or tag content items that they want to be excluded from automated modifications. The task agent can then respect these labels and refrain from making changes to tagged content.
6. **Content Whitelisting:** Allow users to define a list of content items or categories that the task agent is allowed to modify. This approach ensures that the agent only interacts with specified content while leaving other data untouched.
7. **Task Categorization:** Classify tasks into different categories based on their potential impact. Users can set preferences for each category, specifying whether the task agent can execute them automatically or requires manual approval.
8. **Content Exclusion Lists:** Provide users with the ability to create exclusion lists that specify content or content types the task agent should avoid modifying. This list can be adjusted as needed to accommodate user preferences.
9. **Advanced Permissions:** Implement granular permissions for different content sections. Users can define roles and access levels, ensuring that the task agent has limited or no access to critical or sensitive content.
10. **User Feedback Loop:** Encourage users to provide feedback on the outcomes of executed tasks. This feedback loop can help refine the task agent's behavior and decision-making over time, aligning it more closely with user preferences.

11. User Training: Train the task agent to recognize patterns and preferences through user interactions. Over time, the agent can learn and adapt to the user's content preservation preferences.

[00103] By implementing these features, the system ensures that users retain control over their content and can protect specific data or content items from unintended modifications while benefiting from the task agent's automation capabilities.

[00104] FIGs. 8-11 are screenshots of respective user interfaces, with FIG. 8 showing an example screenshot to illustrate how a system, according to the present disclosure, may prompt user (via a user device or more specifically “graphic user device” or GUI) for a description, and the user asking for a pink background. FIG. 9 shows an example screenshot to illustrate how a system, according to the present disclosure, enables the user to preview the pink background (e.g., the background color which separates the images as shown).

[00105] FIG. 10 shows an example screenshot to illustrate how a system, according to the present disclosure, enables a user to ask the system to change the logo text, and FIG. 11 shows an example screenshot to illustrate how a system, according to the present disclosure, returns a preview of the new logo text to the GUI of the user device.

[00106] The above disclosure involving photography-specific screenshots are just a few of many examples of outputs of various example embodiments according to implementations of the present disclosure. In different example implementations, different outputs of the process include different types of websites that have significantly favorable characteristics which are generated automatically, and which can be modified in an iterative process by leveraging the prior version of the website. If there is no prior website and/or additional contextual data, the iterative process can take advantage of LLMs and other AI technology to output a website, web pages or modify the content therein to provide an end user experience which has a more favorable set of characteristics. Those characteristics include but are not limited to: website branding including color palette, logo, font and other branding elements; website look and feel including controls and positioning of control elements; information architecture of the website including site map; creation of descriptions for media and other content on the site; presentation of the content and media on the page (transforming the media into slideshows, video clips, audio clips, playlists and image transitions), and SEO (Search Engine Optimized) metadata for the website.

[00107] Furthermore, aspects of the present disclosure are directed to systems and methods that implement trained AI processing to further contemplate other types of (signal) data that may be collected through various host applications/services (e.g., pertaining to a

software platform). For instance, application of trained AI processing (e.g., one or more trained machine learning models) may be adapted to evaluate not only data and data sources integrating with an exemplary UX for assessing and serving operational opportunities, but other types of contextual data including past and/or current user actions, user preferences, application/service log data, etc., that are each associated with one or more users, entities, systems and/or endpoint devices. This additional signal data analysis may help yield determinations as to how (and/or when) to generate updated analytics (in real-time or near real-time) and/or reporting, as well as when and how often to present data insights and/or suggestions.

[00108] Non-limiting examples of signal data that may be collected and analyzed includes but is not limited to: device-specific signal data collected from operation of one or more user computing devices; user-specific signal data collected from specific tenants/user-accounts with respect to access to any of: devices, login to a distributed software platform, applications, services, etc.; application-specific data collected from usage of applications/services and associated endpoints (including third-party endpoints integrated within a software platform), data collected from disparate software platforms that provide disparate types of operational opportunities; or a combination thereof. Analysis of such types of signal data in an aggregate manner may be useful in helping generate contextually relevant determinations, data insights, etc. Analysis of exemplary signal data may comprise identifying correlations and relationships between different types of signal data specific to user usage of one or more software data platforms (e.g., communications software platforms), where telemetric analysis may be applied to generate determinations with respect to a contextual state of user activity with respect to different host application/services and associated endpoints. Analyzing of signal data, including user-specific signal data, may occur in compliance with user privacy regulations and policies.

[00109] In some examples, one or more components are configured to manage application of one or more AI models to enhance processing described in the present disclosure. Trained AI processing is applicable to aid any type of determinative or predictive processing including specific processing operations described with respect to determinations, classification ranking/scoring and relevance ranking/scoring. An exemplary component for implementation trained AI processing may manage AI modeling including the creation, training, application, and updating of AI modeling. Trained AI processing may be adapted to execute specific determinations described herein including those for analyzing specific data and data sources of a software data platform (e.g., a communications software platform)

and/or generating insights for data augmentation. For instance, an AI model may be specifically trained and adapted for execution of processing operations pertaining to analyzing features and functionality of an XCaaS offering including those non-limiting examples previously described. Non-limiting examples of AI implementation include but are not limited to: analyzing data (and metadata) associated with one or more software platforms including third-party integrations; generating contextual determinations for improving user experience, access to opportunities across disparate platforms (and for example rating or ranking such opportunities), performance, efficiency including contextual determinations for improving account access with respect to servicing opportunities, suggesting utilization of certain opportunities, and/or integrations including third-party integrations to enhance access; and prioritization of opportunities, actions, etc. to improve workflow and processing. Exemplary AI processing may be applicable to aid any type of determinative or predictive processing by any components of the present disclosure, via any of: learning for curating displays, learning for prioritizing opportunities, and learning for manners in which to assess and/or present respective opportunities, among other examples. In one example, trained AI processing comprises a hybrid AI model (e.g., hybrid machine learning model) that is adapted and trained to execute a plurality of processing operations described in the present disclosure. In alternative examples, trained AI processing comprises a collective application of a plurality of trained AI models (e.g., three trained AI models) that are separately trained and managed to execute processing described herein. In alternative examples, the present disclosure extends to integrating third-party AI modeling and further adapting and customizing said AI modeling to work with specific data and data sources of an exemplary software platform. For example, a third-party AI model may be adapted to work with a communications software platform including data, data sources, and integrations (e.g., APIs, web hooks, etc.) related to XCaaS features and functionality. In examples where a plurality of independently trained and managed AI models is implemented, downstream processing efficiency may be improved by an ordered application of trained AI models where processing results from earlier applied AI models can be propagated to subsequently applied AI models. For example, a trained AI model may evaluate opportunities and derive data correlations to improve processing and efficiency including suggestions for reallocation of resources and/or prioritizing of opportunities, which may then be utilized to suggest a re-prioritization of opportunities (and/or reallocation of resources as may be appropriate) to improve efficiency and quality of services provided.

[00110] Non-limiting examples of supervised learning that may be applied comprise but are not limited to: nearest neighbor processing; naive bayes classification processing; decision trees; linear regression; support vector machines (SVM) neural networks (e.g., convolutional neural network (CNN) or recurrent neural network (RNN)); and transformers, among other examples. Non-limiting examples of unsupervised learning that may be applied comprise but are not limited to: application of clustering processing including k-means for clustering problems, hierarchical clustering, mixture modeling, etc.; application of association rule learning; application of latent variable modeling; anomaly detection; and neural network processing, among other examples. Non-limiting examples of semi-supervised learning that may be applied comprise but are not limited to: assumption determination processing; generative modeling; low-density separation processing and graph-based method processing, among other examples. Non-limiting examples of reinforcement learning that may be applied comprise but are not limited to: value-based processing; policy-based processing; and model-based processing, among other examples. Furthermore, a component for implementation of trained AI processing may be configured to apply a ranker to generate relevance scoring to assist with any processing determinations with respect to any relevance analysis, such as that described herein. Scoring for relevance (or importance) ranking may be based on individual relevance scoring metrics described herein or an aggregation of said scoring metrics. In some examples where multiple relevance scoring metrics are utilized, a weighting may be applied that prioritizes one relevance scoring metric over another depending on the signal data collected and the specific determination being generated. Results of a relevance analysis may be finalized according to developer specifications. This may comprise a threshold analysis of results, where a threshold relevance score may be comparatively evaluated with one or more relevance scoring metrics generated from application of trained AI processing. Such approaches may include comparing to preset thresholds as part of the analysis step(s). Further and unless specified otherwise (such as by functionality or “plurality of ...”), the use of certain terms (e.g., circuit, LLM and algorithm) in the singular may be interchangeable with the plural use of the same (e.g., circuits or circuitry, LLMs and algorithms).

[00111] It is recognized and appreciated that as specific examples, the above-characterized figures and discussion are provided to help illustrate certain aspects (and advantages in some instances) of non-limiting examples of the present disclosure. Such disclosure is presented in the form of example methods, structures and devices described in connection with each of the figures or examples, and as each such described example

embodiment has one or more related aspects, such aspects may be modified and/or combined with aspects of the other examples as described hereinabove.

[00112] The skilled artisan would also recognize various terminology as used in the present disclosure. As examples, the Specification may describe and/or illustrates aspects useful for implementing the examples by way of various semiconductor materials/circuits which may be illustrated as or using terms such as layers, blocks, modules, device, system, unit, controller, and/or other circuit-type depictions. Such semiconductor and/or semiconductive materials (including portions of semiconductor structure) and circuit elements and/or related circuitry may be used together with other elements to exemplify how certain examples may be carried out in the form or structures, steps, functions, operations, activities, etc. It would also be appreciated that terms to exemplify orientation, such as upper/lower, left/right, top/bottom and above/below, may be used herein to refer to relative positions of elements as shown in the figures. It should be understood that the terminology is used for notational convenience only and that in actual use the disclosed structures may be oriented different from the orientation shown in the figures. Thus, the terms should not be construed in a limiting manner. Also, unless indicated otherwise, the term "website" is to be interpreted to encompass one or more of: multi-page websites; individual web pages; blogging sites; blog posts; landing pages; single-page applications (SPAs); web-based user interfaces; photography, videography and other digital media galleries; and other related forms of content presented through a web browser or similar internet-connected application.

[00113] Similarly, it will be apparent that various known devices may be used with the aspects and features described herein for example embodiments. As non-limiting examples, such devices may include one or more in combination of the following: devices including communications circuits such as servers, user-operable (e.g., network-enabled) devices such as computer processing circuits (e.g., smart phones and other personal assistant devices (aka user endpoint devices) with user interfaces, laptops, desk-based computer etc.). Such devices may be configured to provide services to other circuit-based devices. Moreover, various other circuit-related terminology is used in a similar context as apparent to the skilled artisan, as is the case with each such apparatus which refers to or includes otherwise known circuit-based structures. As one specific example, a personal assistant device may be implemented as a user endpoint device including a camera and/or graphic user interface (integrated directly with the user endpoint device or as a separate tool communicatively coupled (e.g., via a wired connection or a wireless connection) to the personal assistant device). Each such device (e.g., personal assistant device, endpoint device or other device) includes a communication circuit

and/or data-processing computer circuits and those with a communication circuit are configurable to establish a network connection (e.g., communication sessions with other such devices over the Internet or other network). Such networks include but are not limited to a broadband network (such as the Internet or a cellular communications network), local area networks, device-to-device connections (e.g., Bluetooth).

[00114] In certain embodiments, such devices (e.g., one with a processing circuit and communications circuitry), are configured to execute (e.g., after downloading over a network) a set (or sets) of instructions (and/or configuration data) which can be in the form of software stored in and accessible from a memory circuit, and where such circuits are directly associated with one or more algorithms (or processes), the activities pertaining to such algorithms are not necessarily limited to the specific flows such as shown in the flow charts illustrated in the figures (e.g., where a circuit is programmed to perform the related steps, functions, operations, activities, etc., the flow charts are merely specific detailed examples). The skilled artisan would also appreciate that different (e.g., first and second) modules can include a combination of a central processing unit (CPU) hardware-based circuitry and a set of computer-executable instructions, in which the first module includes a first CPU hardware circuit with one set of instructions and the second module includes a second CPU hardware circuit with another set of instructions.

[00115] Certain embodiments are directed to a computer program product (e.g., nonvolatile memory device), which includes a machine or computer-readable medium having stored thereon, instructions which may be executed by a computer (or other electronic device) that includes a computer processor circuit to perform one or more of the operations/activities disclosed herein (e.g., as in the various example embodiments, specific execution of an AI algorithm, using or generating a model, etc.). In certain example environments, these instructions may reflect activities or data flows as may be exemplified by way of figures, flow charts, and the detailed description.

[00116] Based upon the above discussion and illustrations, those skilled in the art will readily recognize that various modifications and changes may be made to the various embodiments without strictly following the exemplary embodiments and applications illustrated and described herein. For example, methods as exemplified in the Figures may involve steps carried out in various orders, with one or more aspects of the embodiments herein retained, or may involve fewer or more steps. Such modifications do not depart from the true spirit and scope of various aspects of the disclosure, including aspects set forth in the claims.

What is Claimed:

1. A computer-implemented method comprising:
 - determining a subset of multiple aspects or elements of an existing website to modify by analyzing user inputs, pertaining to desired changes in the existing website with respect to at least one of content and design;
 - constructing prompts for one or more large language models (LLMs) incorporating said user inputs and data structures characterizing the subset of multiple aspects or elements;
 - driving one or more AI/ML algorithms of at least one computing-processor circuit, based on the constructed prompts, to output modifications of at least some of the subset of multiple aspects or elements; and
 - generating a custom website, including the desired changes and based on the user input.
2. The computer-implemented method of claim 1, wherein the user inputs include different types of media, including image- or video-based media, the method further including extracting from metadata visual attributes and using the visual attributes to implement the prompts as constructed for the one or more LLMs, the method further including automatically integrating external content into user websites by carrying out one or a combination of: importing diverse content from one or more network-accessible external sources from among at least one style-related aspect or style-related element into one or more user portfolios associated with the user inputs, social media, marketplaces, portfolios, blogs, and email marketing; and dynamically incorporating content corresponding to at least one of: images, articles, videos, and audio tracks.
3. The computer-implemented method of claim 1, further including: extracting media-based data including at least one of metadata and visual attributes, from the user inputs; and in response, using the media-based data, as extracted from the user inputs, to drive the one or more LLMs.

4. The computer-implemented method of claim 1, further including:
transforming the user inputs from one or more formats native to the user input, into a format that is conducive to an input for processing via the one or more LLMs, and, in response, carrying out the constructing of prompts; and
presenting a proposed custom website, including the desired changes and based on the user inputs, to a user for review and approve.
5. The computer-implemented method of claim 1, further including causing the existing website to be at least one of: compared to the custom website, or replaced by the custom website.
6. The computer-implemented method of claim 1, further including testing operability aspects of the custom website.
7. The computer-implemented method of claim 1, further including validating the output modifications.
8. The computer-implemented method of claim 1, further including creating, via a chat interface, updated website pages and metadata relating to the updated pages.
9. The computer-implemented method of claim 1, further including employing natural language processing techniques to solicit user feedback and, in response, refining design options available for said generating the custom website.
10. The computer-implemented method of claim 1, further including enabling real-time previews that allow a user to iteratively steer design alterations toward their desired outcomes.
11. The computer-implemented method of claim 1, further including storing and cataloging generated website designs to establish a knowledge base for future design initiatives and ongoing enhancement efforts.

12. The computer-implemented method of claim 1, wherein said one or more AI/ML algorithms of at least one computing-processor circuit includes a first AI/ML algorithm of a first computing-processor circuit and a second AI/ML algorithm of a second computing-processor circuit, wherein the second computing-processor circuit is different than the first computing-processor circuit in terms of platform type and expected attributes to be generated via the respective output modifications.

13. The computer-implemented method of claim 1, wherein the user inputs include a plurality from among: content from media, content from communications related to the user inputs or to an entity designated on behalf of the user, content from or accessible through the existing website, purchase-related offerings or transactions, and behavior data derived from the existing website.

14. The computer-implemented method of claim 1, further comprising:
enabling a user associated with the user input to provide one or more target external sources for indicating use of at least one of the following in the custom website: a style, and for incorporating one or more themes from among the following: layouts, colors, typography, assets, design elements, one or more competitor-inspired designs: client-specific branding alignment, uniform design language across multiple sites, selected theme adaption, and one or more customization options for seamless adaptation of transferred styles to facilitate harmonious integration with content and branding-related data of the existing website.

15. The computer-implemented method of claim 1, further comprising integrating autonomous web data by one or more of:

equipping an autonomous task agent with web crawling and data inference capabilities;
automatically discovering and integrating user-related data from diverse web sources;
and
facilitating optimization of content integration, thereby relieving users of manual curation.

16. The computer-implemented method of claim 1, further comprising using autonomous content distribution to external destinations by carrying out the following steps:

enabling an autonomous agent to push data and assets between one or more user sites and diverse third-party platforms;

automating content dissemination to one or more of: social media, portfolio sites, e-commerce platforms, content networks, email marketing, video sharing, and collaborative platforms; and

empowering the autonomous agent with intelligent decision-making capabilities to identify content characteristics, thereby facilitating or ensuring tailored distribution based on content type and platform suitability.

17. The computer-implemented method of claim 1, further comprising using optimizing task performance by:

integrating quality indicators into an autonomous task agent's decision-making framework;

collecting user feedback and actions to derive quality indicators for completed tasks; and

utilizing machine learning to analyze quality indicators, adapt task decision-making, and enhance task success rates over time.

18. The computer-implemented method of claim 1, further including interacting via voice-based communications with a task agent by:

incorporating a voice recognition module for capturing and processing user voice input;

using natural language processing to interpret user intent from voice input; and

enabling task execution based on interpreted intent, fostering dynamic back-and-forth voice interactions and enhancing accessibility, engagement, and efficiency.

19. The computer-implemented method of claim 1, further including safeguarding content from unintended modifications, by performing one or more of:

providing content locking for specific items, version history for change tracking and content restoration and/or selectively approving one or more tasks related to at least one of the existing website and the custom website, and

providing simulation previews, marking content, whitelisting, categorizing tasks, and providing advanced permissions for controlled content interaction, and ensuring user control and content preservation.

20. The computer-implemented method of claim 1, further including using different media or content hosting environments linked to an entity related to the user inputs to drive the one or more LLMs, wherein the different media or content hosting environments are from among two or more of: websites, social media pages, social media sites, social media blogs and at least one other type of media or content hosting environment.

21. An apparatus comprising:

a data-processing computer circuit to:

determine a subset of multiple aspects or elements of an existing website to modify by analyzing user inputs, pertaining to desired changes in the existing website with respect to at least one of content and design;

construct prompts for one or more large language models (LLMs) incorporating said user inputs and data structures characterizing the subset of multiple aspects or elements;

drive one or more AI/ML (artificial intelligence and/or machine learning) algorithms of at least one computing-processor circuit, based on the constructed prompts, to output modifications of at least some of the subset of multiple aspects or elements; and

generate a custom website, including the desired changes and based on the user input.

22. A computer-implemented method, the method comprising:

rapidly generating diverse website designs and dynamically adapting them based on user feedback, by

processing for a user, user inputs, existing user context, and external sources to generate a range of design variations for selection of one or more custom websites;

employing one or more natural language processing techniques to solicit additional input or feedback from an entity corresponding to or on behalf of the user, and, in response, refining design options of the one or more custom websites; and

enabling the entity real-time access to allow the entity to iteratively steer the one or more custom websites, as selected, for design alterations toward desired outcomes consistent with at least some of the user feedback and for generating one or more custom website designs.

23. The computer-implemented method of claim 22, further including prompting a user interface to provide one or more limit on the set of the design variations and then carrying out the rapidly generating diverse website designs in sequence via at least one of the following circuits: a task agent content-generation circuit to indicate the design variations by way of batch changes, and a task agent external sourcing circuit to indicate the design variations by importing data from one or more external sources.

24. The computer-implemented method of claim 22, further including:
storing and cataloging the generated one or more custom website designs to establish a knowledge base for future design initiatives and ongoing enhancement efforts; and
generating one or more custom website designs.

25. The computer-implemented method of claim 22, further including batch-asset categorizing and optimizing aspects of the one or more of the custom website designs by causing a plurality of steps to be carried out from among the following:

initiating analysis of uploaded media assets, including at least one of one or more images and one or more videos, using an AI-vision client module;

extracting, via one or more computer-vision algorithms, a plurality of media-based attributes including prominent colors for palette generation, face, object, and scene recognition, technical quality assessment, and identification of regions of interest pertaining to at least one of the one or more custom websites;

selecting, via a media-processing module, certain media assets based on analysis of outcomes of certain uploaded media assets and website objectives indicated via user inputs;

identifying focal points by detecting faces, products, or other salient regions within the media assets;

categorizing images into groups including two or more of people, portraits, products, and nature;

extracting two or more data types from among: text, logos, landmarks, custom objects, and visual attributes;

structuring the extracted visual attributes into metadata and, in response, using the metadata to metadata and influence prompts for driving large language models (LLMs);
improving and/or customizing at least one of the one or more custom websites, based on multimedia analysis results, to generate content therein aligned with the media assets; and
enhancing at least one attribute of the at least one of the one or more custom websites by incorporating user-feedback loops, each of which drives user feedback to the computer-implemented method.

26. The computer-implemented method of claim 22, further including causing a plurality of steps to be carried out from among the following:

employing a library of predefined tasks with parameters to generate task instructions, encompassing at least one activity involving: altering design elements, composing blog posts, and adjusting subscriptions;

creating novel page types tailored to user website objectives and context;

analyzing user preferences, target audience, and content requirements to determine optimal parameters for customized pages, encompassing layout specifications, style attributes, media selections, and text content details;

generating prompts for at least one large language model (LLM) based on determined parameters indicated or provided by user inputs, encapsulating contextual context and content requisites;

utilizing LLM-based responses to populate component properties for generating bespoke content specific to the user or an entity designated on behalf of the user;

enabling reuse of approved generated content for training and generating more content that is pertinent to the user inputs;

integrating auto-generated content with configurable predefined content for a balanced approach to flexibility, personalization, and iterative content creation; and

constructing one or more custom website designs in response to at least one of the above steps.

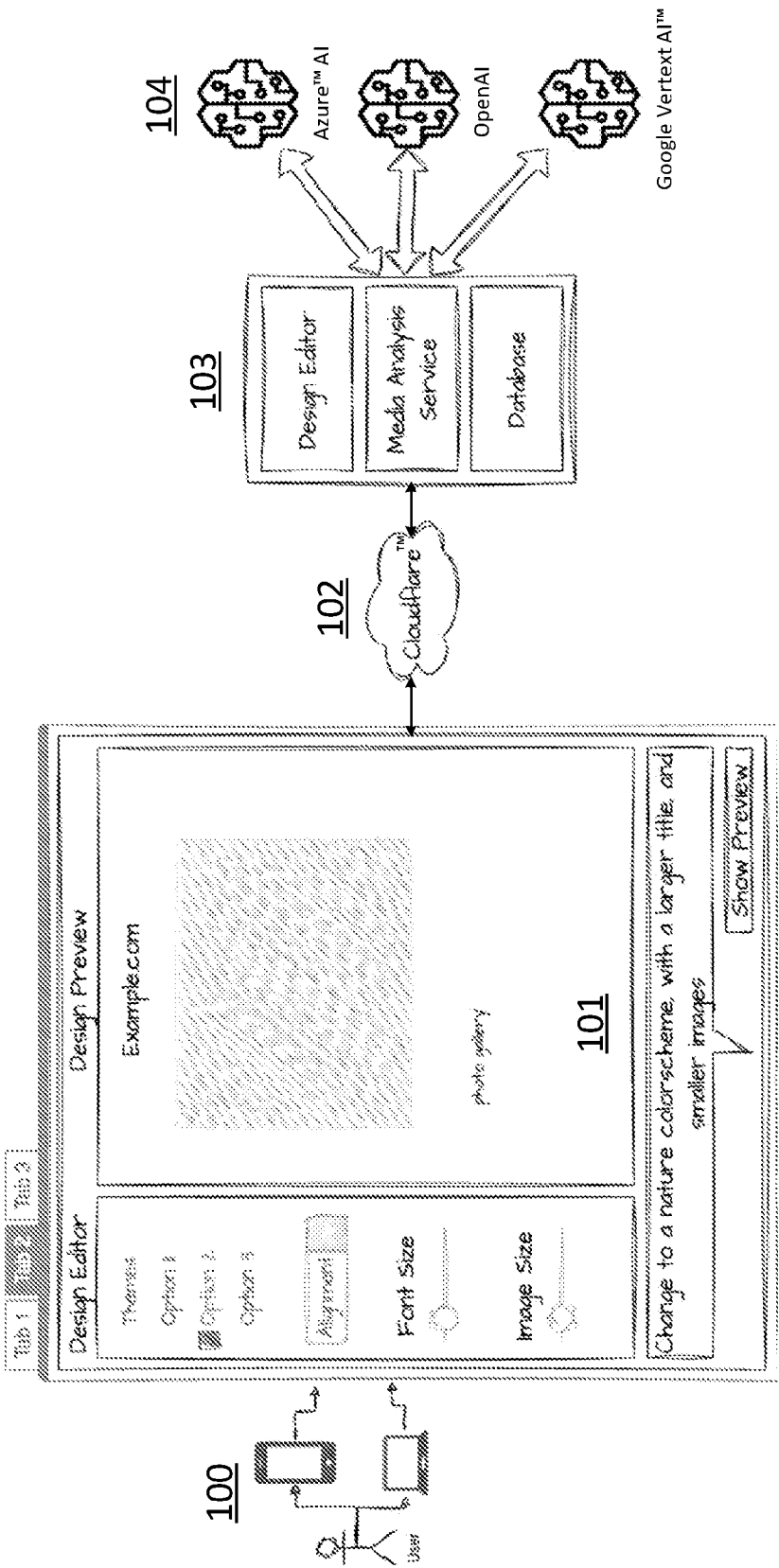


FIG. 1

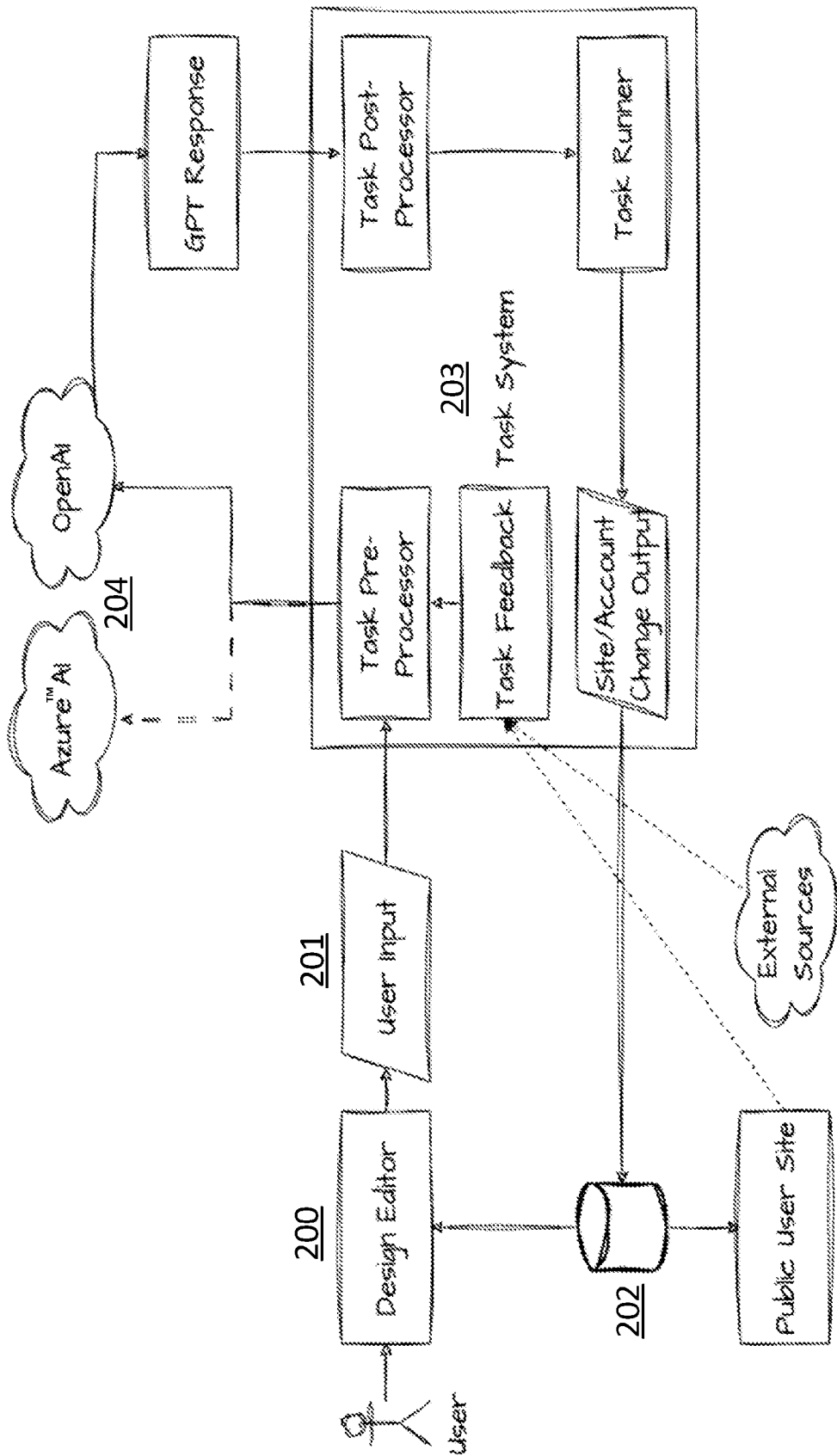


FIG. 2

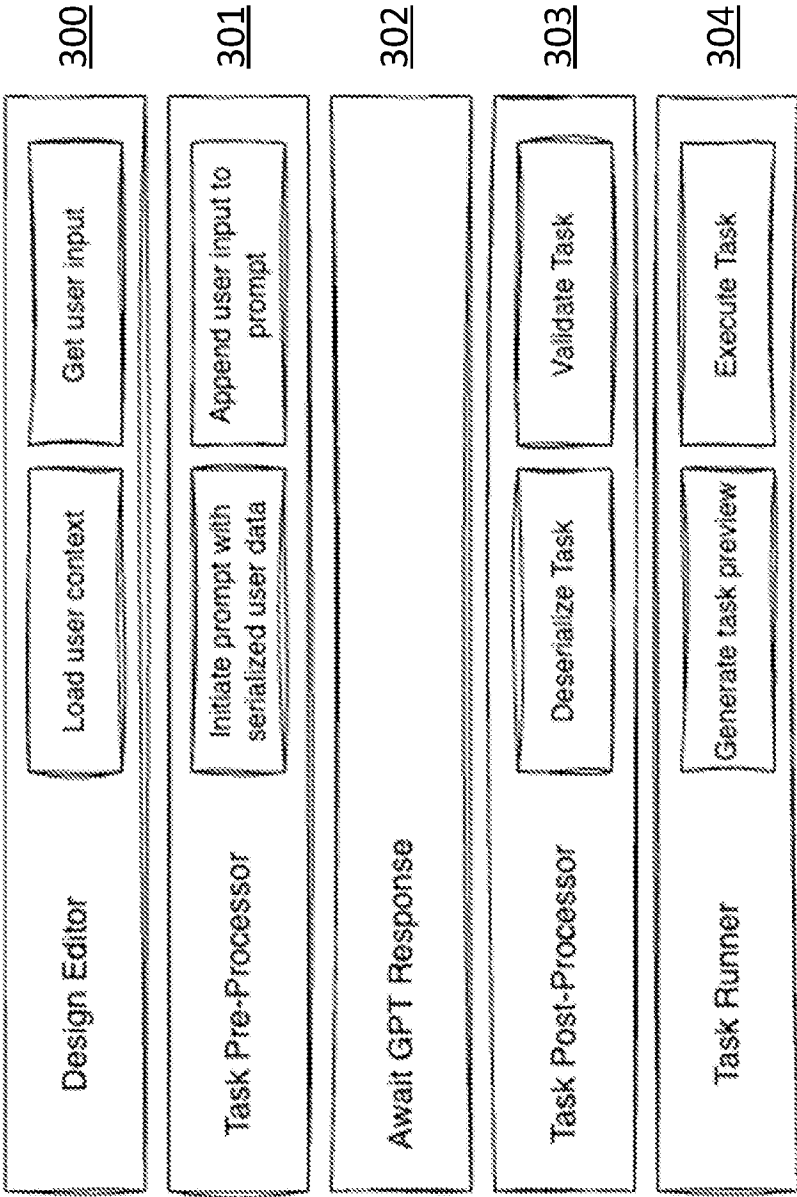
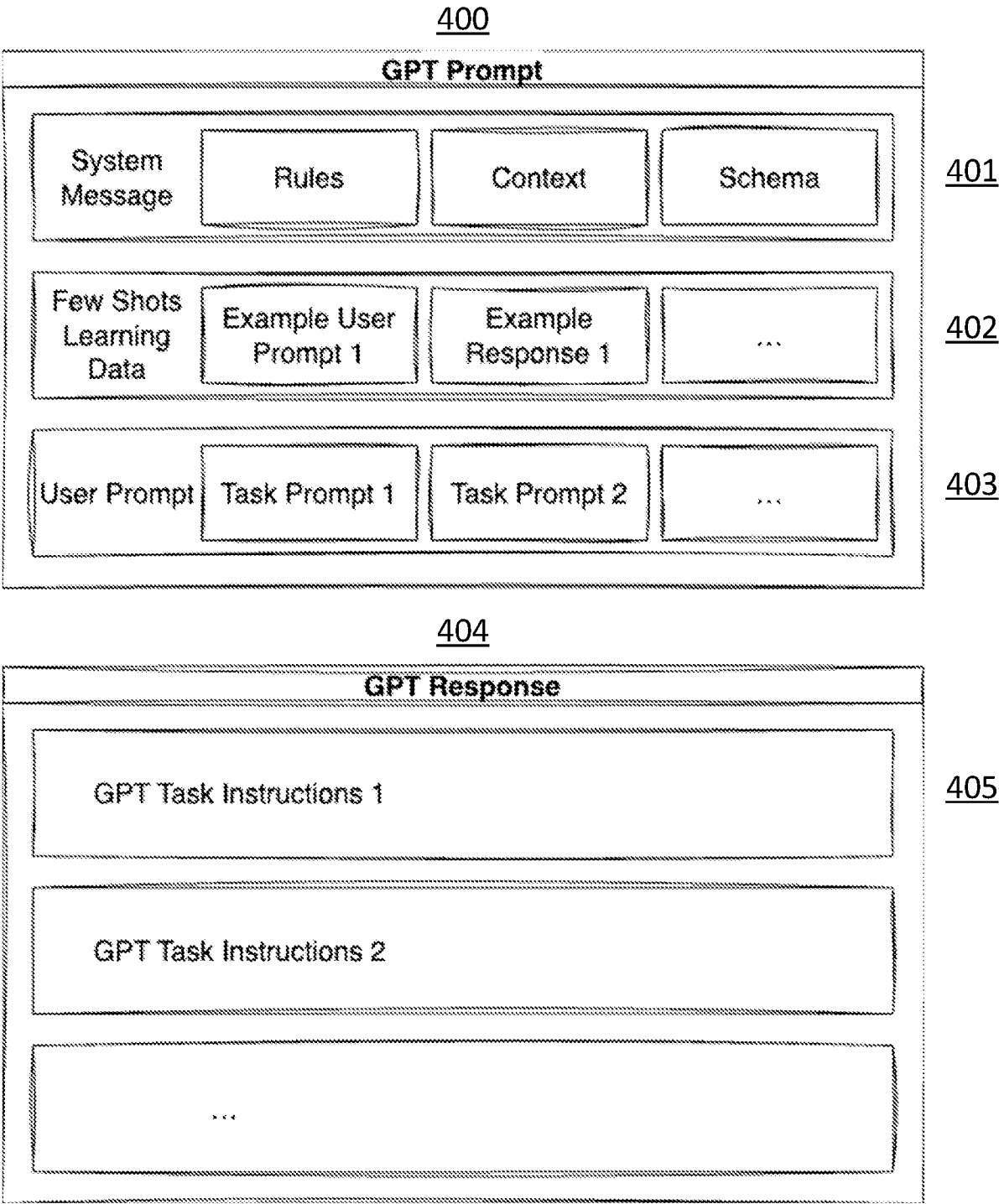


FIG. 3



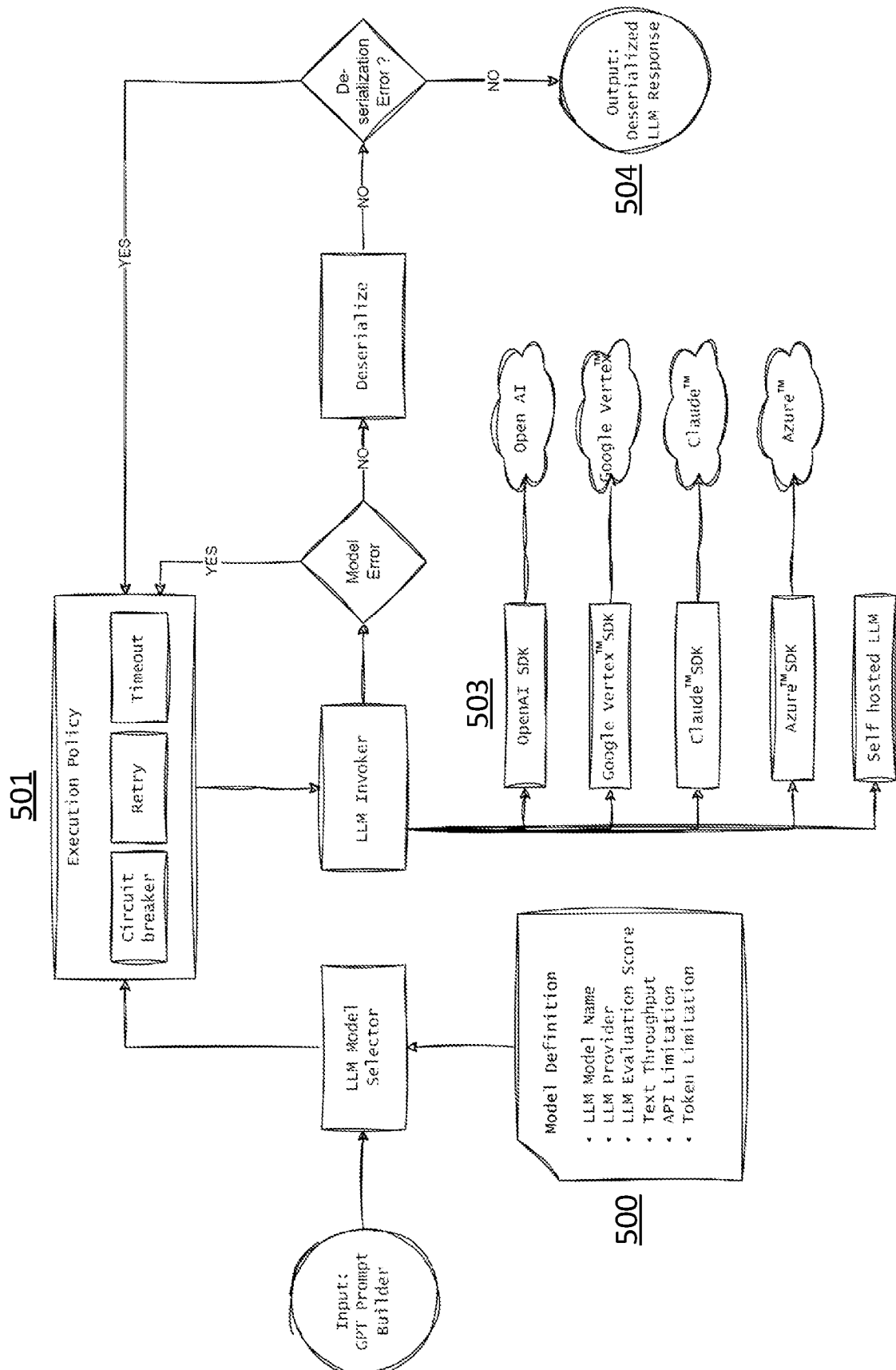


FIG. 5

600

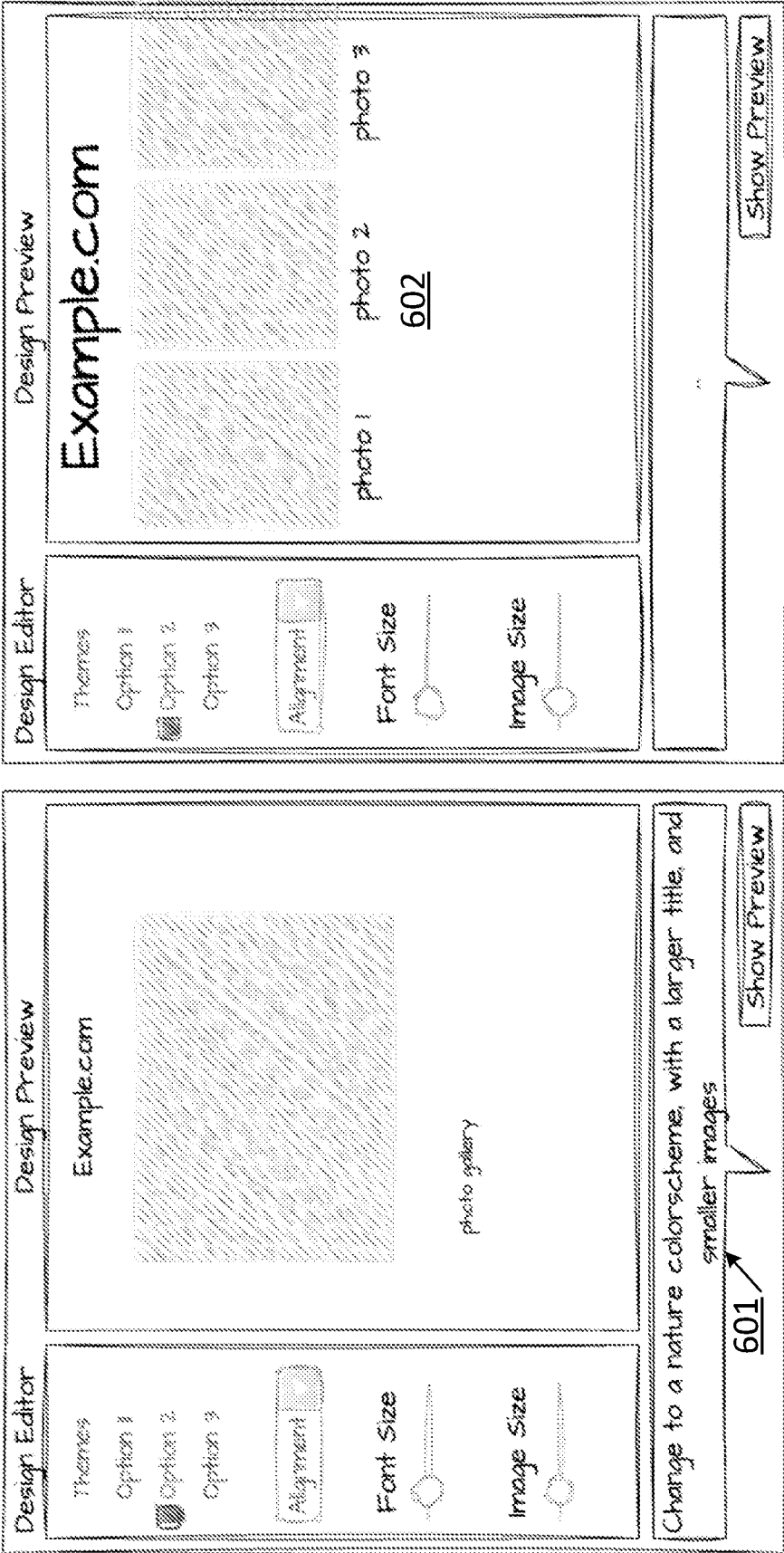


FIG. 6

702

Task Agent External Sourcing



I am a helpful assistant. What would you like to do?



Rewrite my About page, using my resume at [linkedin.com/company/you](#)



All set, here is your updated [About page](#).



I noticed your resume links to an Instagram account with new photos, would you like to import them?



Yes, get the last 50 photos on a gallery page.



You got it, 50 photos have been imported to [this gallery page](#).



701

Task Agent Content Generation



I am a helpful assistant. What would you like to do?



Make my store product descriptions funnier.



All set, here are your [new product descriptions](#).



Create discount codes for 10% off and a flat \$25 off for orders over \$100.



Here is a calendar and your sales are US-based, I've named them [HALLOWEEN](#) and [THANKSGIVING](#).



Create a blog post announcing these discounts.



700

Task Agent Design Editing



I am a helpful assistant. What would you like to do?



Give my site a natural, earth tone color scheme.



All set, click to try any of these palettes: [Blue Greens](#), [Rust Browns](#), [Terracotta Red](#)



Create a "Wild Photos" gallery with pictures.



You got it, click to preview your [Wild Photos](#) gallery. I've found and added 20 relevant pictures. Would you like to share on Facebook?



Yes, but rename the gallery to "Wildies" and enlarge the photos.



FIG. 7

8 / 11

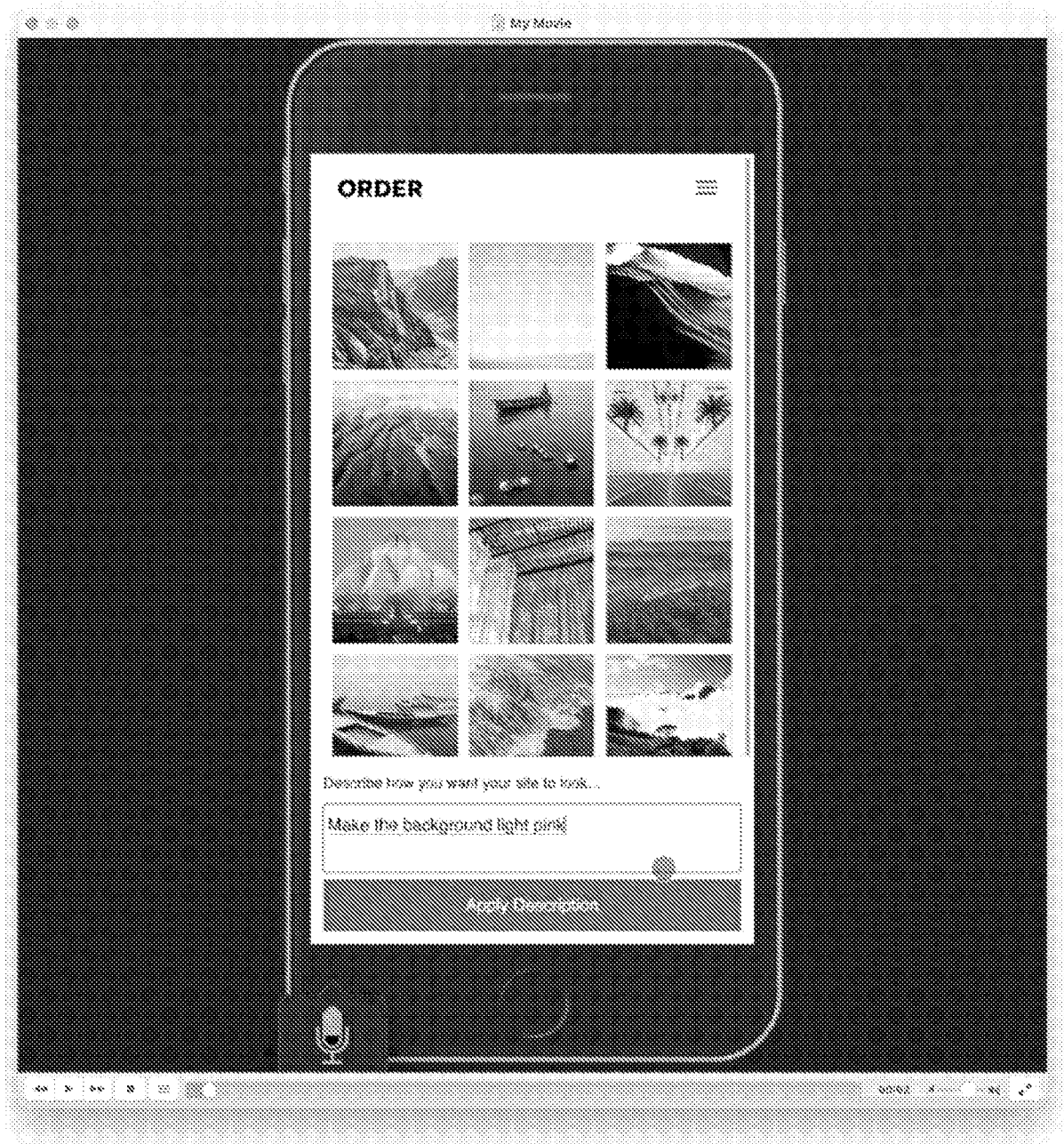


FIG. 8

9 / 11

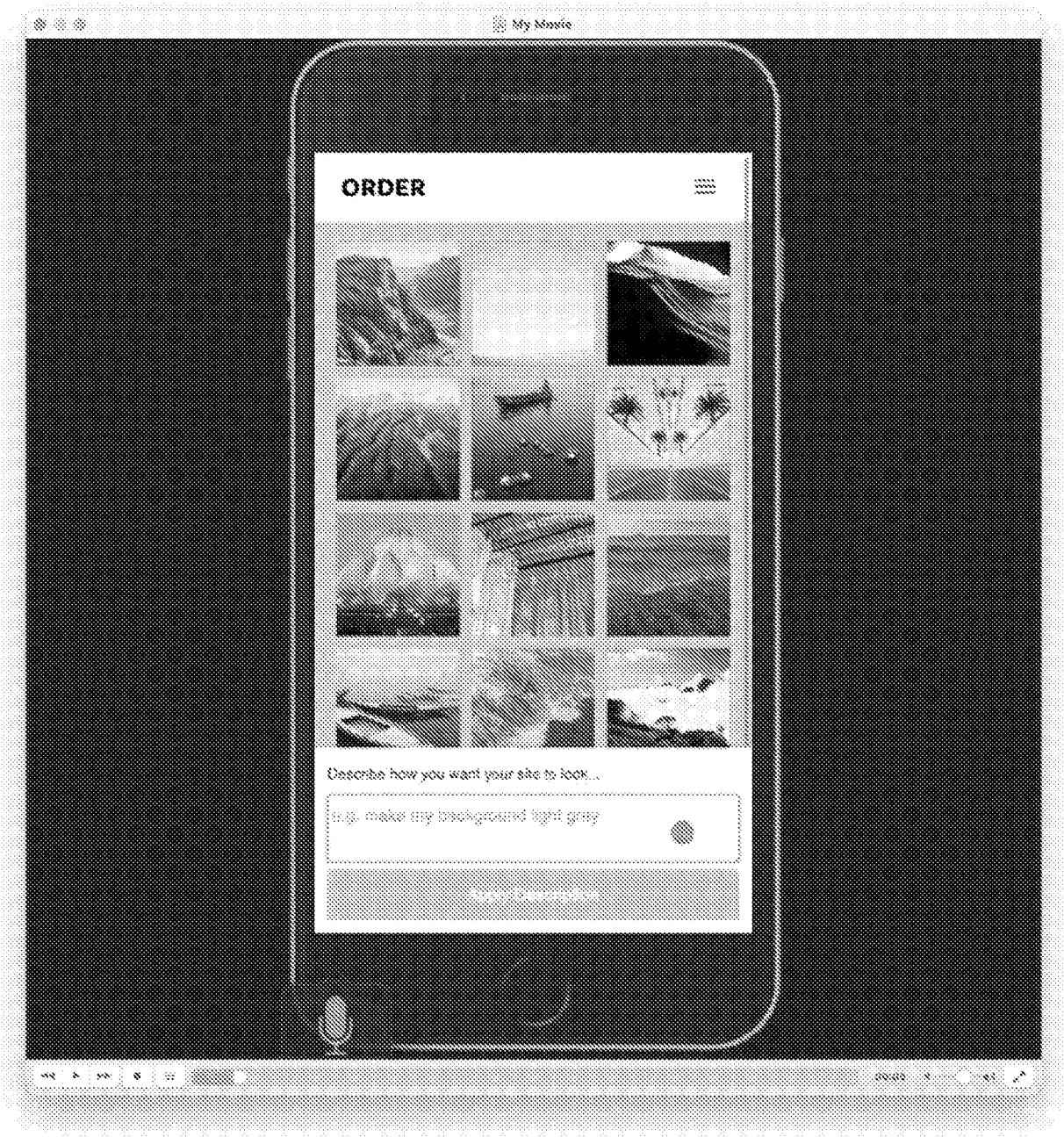


FIG. 9

10 / 11

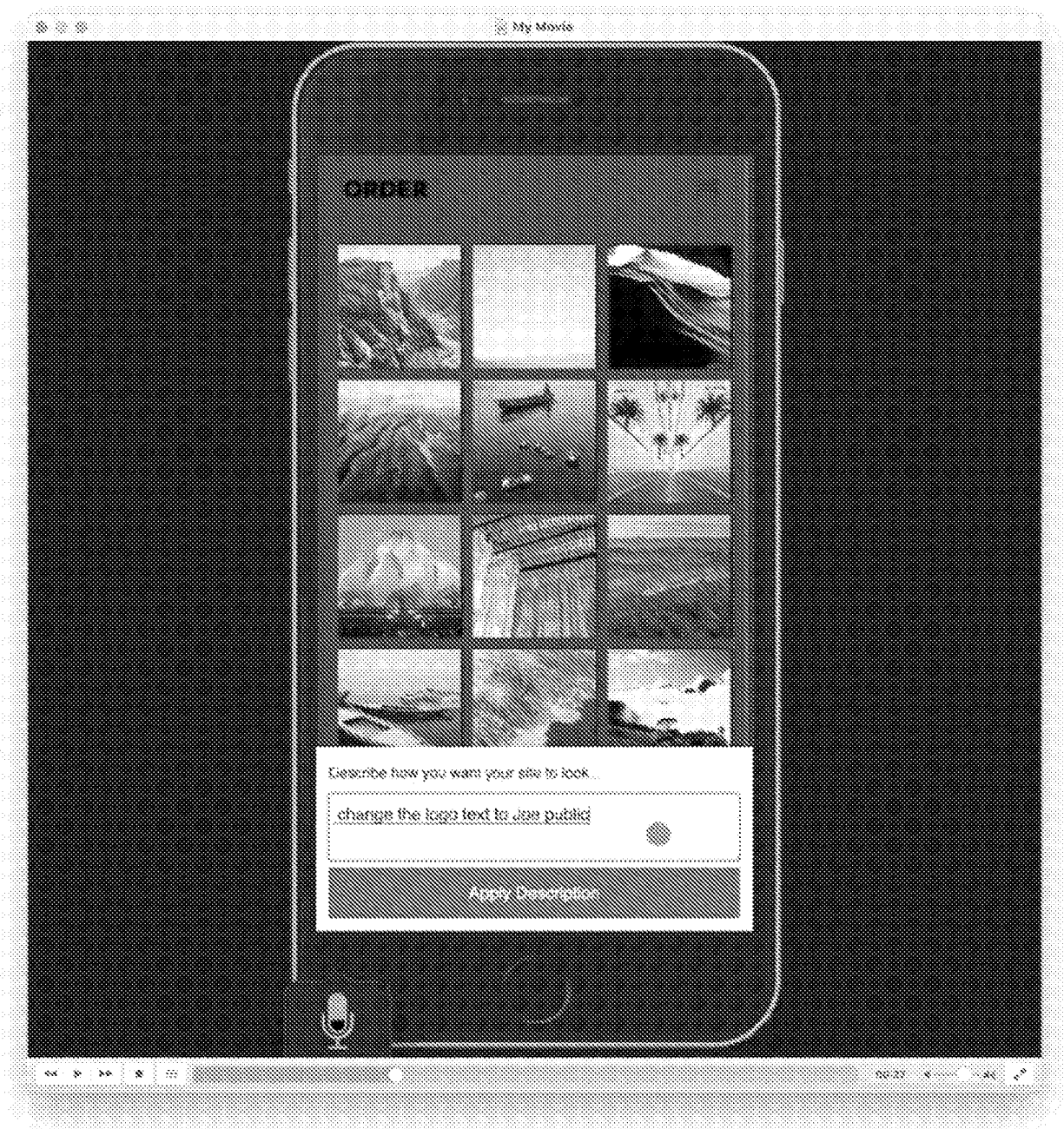


FIG. 10

11 / 11

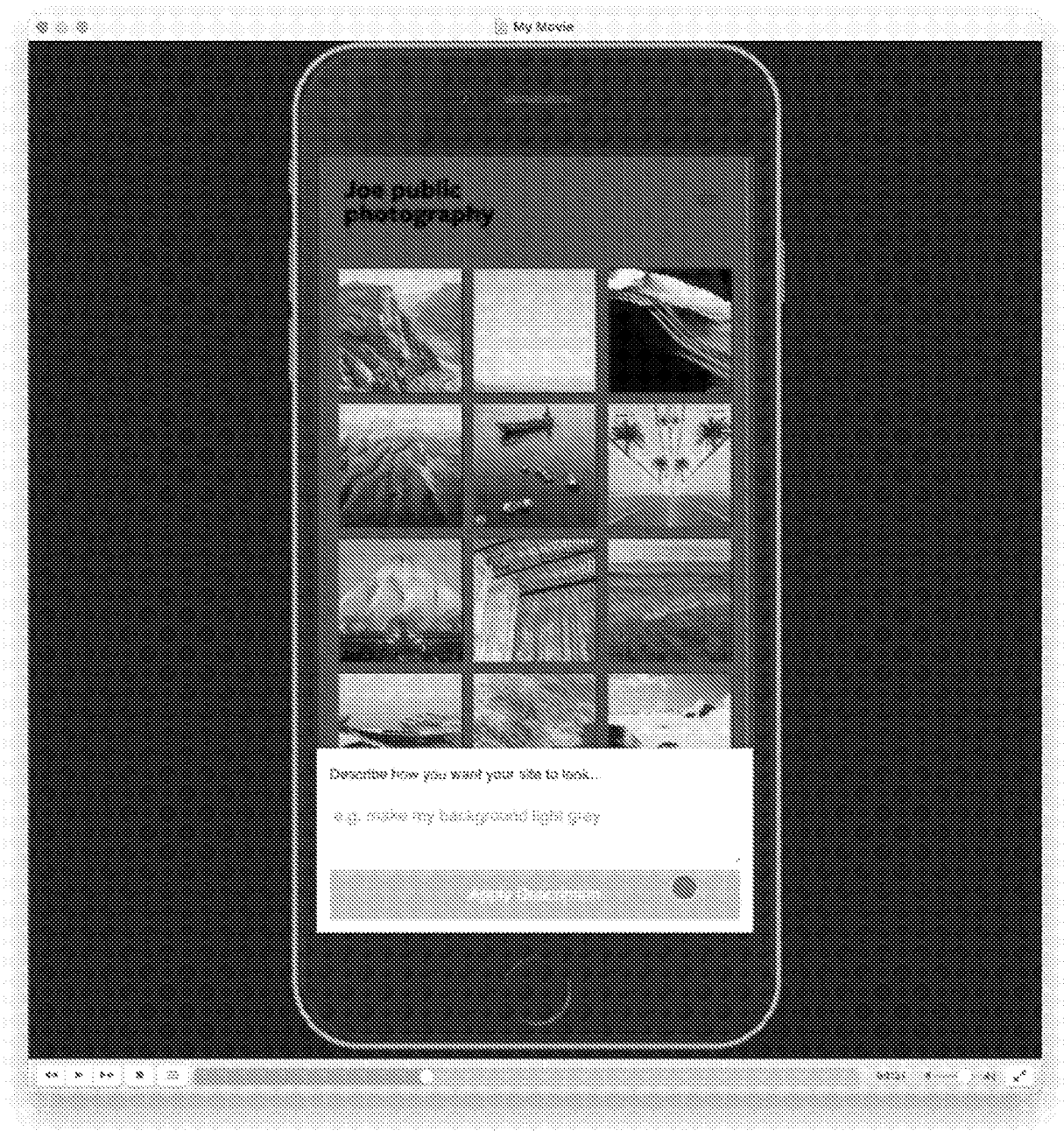


FIG. 11