ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

From Mock-ups to Code: A Conceptual Synthesis of AI-Driven Automatic Website Generation

Thisaranie Kaluarachchi

University of Colombo School of Computing, Sri Lanka

Publication Date: 2025/11/14

Abstract: The emergence of artificial intelligence (AI) and machine learning has transformed many dimensions of software engineering, including the way websites are designed and developed. Traditionally, website creation has been a labour-intensive process involving manual translation of high-fidelity design artifacts such as sketches, wireframes, and mock-ups into functional code. Despite the availability of visual editors and content management systems, the gap between design intent and implementation accuracy remains a persistent challenge. This research presents a comprehensive synthesis of research in AI-driven automatic website generation, with particular attention to the evolution of methods, their underlying computational models, and their integration with modern web design practices.

The discussion begins by tracing the development of website generation approaches from heuristic and template-based systems to machine learning–assisted and deep learning–based frameworks. It categorizes existing methods into three major paradigms, mock-up-driven, example-based, and AI-driven website generation, and examines their methodological foundations, advantages, and limitations. The synthesis integrates insights from computer vision, natural language processing, and code generation research to identify common principles underlying automated design translation.

Building on this literature, the research introduces a conceptual framework that unifies the processes of visual input interpretation, graphical user interface (GUI) element detection, semantic classification, hierarchical structuring, and code synthesis. The proposed framework serves as both an analytical model and a design roadmap for future research in automatic website generation. The research concludes by outlining emerging directions such as multimodal generative AI, human-in-the-loop design collaboration, and the integration of explainable AI principles in web automation. Overall, this synthesis advances the theoretical understanding of design automation and provides a foundation for future innovations that bridge the creative and computational aspects of web engineering.

Keywords: *Automatic Website Generation, Artificial Intelligence, Computer Vision, Machine Learning, GUI Automation, Web Design Automation, Code Generation, Design Science.*

How to Cite: Thisaranie Kaluarachchi (2025) From Mock-ups to Code: A Conceptual Synthesis of AI-Driven Automatic Website Generation. *International Journal of Innovative Science and Research Technology*, 10(11), 305-317. https://doi.org/10.38124/ijisrt/25nov339

I. INTRODUCTION

The evolution of the *World Wide Web* has redefined the way information, services, and businesses operate in the digital age. As websites have become the primary interface between organizations and their users, website design and development have transformed into critical determinants of user engagement, credibility, and overall digital experience. Modern web development emphasizes responsive, accessible, and aesthetically appealing designs that align with brand identity and usability standards. However, the process of developing such websites remains complex, time-consuming, and highly dependent on specialized technical expertise [1,2]. The design-to-code transition, wherein static visual artifacts such as sketches, wireframes, or mock-ups are translated into functional front-end code, continues to present a significant bottleneck in the development pipeline.

Although content management systems (CMS) such as *WordPress*, *Drupal*, and *Joomla* have simplified aspects of web development through predefined templates and plug-ins, they inherently limit flexibility and creative control [3]. Similarly, visual editors and prototyping tools such as *Figma*, *Adobe XD*, *Sketch*, and *Axure* have streamlined collaborative design processes but stop short of producing semantically accurate, production-ready code. Consequently, designers and developers still face a considerable semantic gap between graphical design intent and executable web code [4, 5]. This

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

challenge has driven growing research interest in automatic website generation [6], where computational models aim to replicate human design intelligence by converting design artifacts directly into web-ready code.

The concept of automatic website generation represents a natural progression of broader trends in automation, artificial intelligence (AI), and design science. Over the past decade, AI-driven techniques have advanced from rule-based heuristic systems to deep learning models capable of learning structural and stylistic patterns in web interfaces [7,8]. These advancements have been supported by breakthroughs in computer vision (CV) and natural language processing (NLP), which enable machines to interpret graphical user interface (GUI) components and translate them into structured representations such as the Document Object Model (DOM). In particular, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated strong performance in recognizing and generating UI elements, giving rise to a range of models that automatically generate website code from mock-ups or screenshots [4,9].

Early explorations in this field adopted heuristic approaches, combining handcrafted rules with image processing algorithms to detect and map interface components [10]. Although effective for constrained scenarios, heuristic models lacked generalization capability across diverse web layouts. Later, end-to-end deep learning models such as *pix2code* [4] and its successors leveraged encoder–decoder architectures to learn direct mappings between visual inputs and their corresponding code tokens. Despite these innovations, existing methods often struggle with structural accuracy, semantic consistency, and scalability when applied to real-world, high-fidelity web designs [11,12]. The complexity of web layouts, encompassing nested containers, dynamic components, and diverse stylistic features, requires hybrid frameworks that integrate machine learning with rule-based logic and contextual reasoning.

A parallel line of research explores example-based automatic website generation, where new designs are synthesized from existing websites used as reference examples. Systems such as *WebCrystal* [13] and *GUIFetch* [14] allow users to extract, analyze, and recombine visual and structural features from professionally designed websites. This paradigm emphasizes design reuse and democratization of web development but still depends on partial manual intervention. On the commercial front, AI-powered website builders such as *Wix ADI*, *Bookmark*, and *The Grid* have popularized automated design generation for non-technical users by combining pre-trained models with interactive customization [15,16]. While these tools have improved accessibility, they typically prioritize convenience over design authenticity and technical depth.

Within this evolving landscape, the intersection of AI, computer vision, and web engineering has emerged as a promising frontier for bridging the long-standing divide between design and development. By learning from large-scale datasets of real-world websites, AI models can identify recurring structural patterns and stylistic conventions that define high-quality user interfaces [7,17]. The integration of design knowledge into automated systems not only reduces the time and cost associated with manual development but also holds potential for achieving higher visual fidelity and semantic coherence between prototypes and deployed interfaces.

Nevertheless, the field of automatic website generation remains fragmented, characterized by diverse methodological approaches and limited unifying theory. The literature reveals three dominant paradigms:

- Mock-up-driven approaches - Focus on translating visual design artifacts into functional code through image recognition and code synthesis.
- Example-based approaches - Derive generative rules from existing web templates or repositories.
- AI-driven approaches - Leverage intelligent agents to autonomously infer design preferences, aesthetics, and layout composition [1, 7, 9].

Despite the technical diversity of these paradigms, they share a common ambition, to bridge the image-to-code abstraction gap through computational modelling of human design intelligence. Yet, the absence of a comprehensive conceptual framework hinders systematic progress and cross-comparison of methods. This research addresses this gap by synthesizing existing research into a unified conceptual perspective that elucidates the underlying processes of AI-driven website generation.

Drawing from systematic analysis and design science principles, this research situates AI-driven website generation as a multidisciplinary convergence of machine learning, computer vision, human–computer interaction, and software engineering. Section 2 traces the historical evolution of automation in web design, while Section 3 systematically reviews contemporary approaches and taxonomies. Section 4 synthesizes emerging trends in the field, and Section 5 introduces a conceptual framework that integrates the technical and theoretical dimensions of automatic website generation. The subsequent sections discuss research challenges, future directions, and implications for academia and industry.

By combining insights from state-of-the-art research with practical reflections from AI-driven prototyping tools, this research aims to advance scholarly understanding of how machines can learn to design, a critical step toward the next generation of intelligent, adaptive, and human-aligned web development systems.

II. EVOLUTION OF AUTOMATIC WEBSITE GENERATION

The pursuit of automating website creation has evolved significantly over the past two decades, paralleling major advances in artificial intelligence, computer vision, and web engineering. What began as a heuristic-driven process aimed

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

at accelerating front-end development has gradually transformed into a data-centric, learning-based paradigm that attempts to emulate human design reasoning. The trajectory of this evolution can be broadly divided into three overlapping phases: (1) heuristic and rule-based systems, (2) machine learning and deep learning models, and (3) AI-driven intelligent design systems. Each phase reflects a distinct conceptual shift from procedural automation to data-driven inference and ultimately to generative design intelligence.

➢ *Heuristic and Rule-Based Systems*

Early research on automatic website generation primarily relied on heuristic algorithms and handcrafted rules to transform design artifacts into functional prototypes. These approaches decomposed the design-to-code process into discrete steps, such as graphical element detection, layout reconstruction, and HTML template synthesis. Heuristic rules were used to detect and map interface elements such as buttons, text fields, and images by analyzing colour histograms, contours, and geometric relationships [1, 2].

One of the earliest implementations of this paradigm was the *REMAUI* system, which reverse-engineered mobile app user interfaces from visual mock-ups using optical character recognition (OCR) and layout heuristics [1]. Similarly, *P2A* extended the concept to animated mobile applications, demonstrating the feasibility of extracting temporal relationships between interface states [7]. In the context of web applications, Huang et al. [2] proposed a heuristic proofreading mechanism that refined incomplete or inconsistent GUI element mappings, significantly improving layout reconstruction accuracy.

While these systems demonstrated that visual-to-code translation was achievable without manual coding, their rule dependency limited scalability. Every new GUI element type or layout pattern required additional handcrafted logic, making the systems brittle when exposed to unfamiliar designs [3]. Moreover, heuristic models often failed to capture the semantic relationships between nested components, resulting in structurally shallow or visually inconsistent outputs. This shortcoming prompted a transition toward more flexible, learning-based techniques capable of inferring design rules from data rather than relying solely on predefined heuristics.

➢ *Machine Learning and Deep Learning-Based Systems*

The second phase in the evolution of website generation coincided with the rapid maturation of deep learning and its applications in image understanding and sequence modelling. The seminal work by Beltramelli [4], *pix2code*, marked a transformative milestone by introducing a neural encoder–decoder architecture capable of converting GUI screenshots directly into structured code. The model combined a CNN as a visual encoder and an RNN, specifically Long Short-Term Memory (LSTM) units, as a language decoder to generate a domain-specific language (DSL) representation of the interface.

Subsequent research extended the *pix2code* framework in multiple directions. Zhu et al. [18] introduced hierarchical layout modelling to preserve the spatial nesting of elements, while Han et al. [19] replaced LSTM decoders with Gated Recurrent Units (GRUs) to improve training efficiency. Liu et al. [20] further explored hybrid CNN–transformer architectures for fine-grained GUI element recognition and layout inference. Collectively, these models demonstrated that neural networks could learn the latent design language underlying graphical interfaces, moving beyond rigid heuristics toward data-driven design synthesis.

Another influential trend within this phase was the integration of object detection with machine learning. Researchers began to employ models such as Faster R-CNN and YOLO (You Only Look Once) to localize UI components within mock-ups and classify them into semantic categories [21]. These detectors outperformed classical CV techniques by learning spatial and contextual patterns across thousands of GUI examples. The combination of object detection and CNN-based classification established the technical foundation for modern mock-up–to-code systems such as Microsoft's *Sketch2Code* and commercial tools like *TeleportHQ* and *Zecoda* [22].

Despite these advances, several limitations persisted. Deep learning models required large-scale labelled datasets of GUI elements, which were scarce and difficult to annotate. Furthermore, end-to-end learning often produced syntactically valid but semantically inconsistent code, as models lacked explicit understanding of the DOM hierarchy. Studies by Moran et al. [7] and Hassan et al. [23] emphasized that structural fidelity, how accurately the generated DOM reflects real-world web hierarchies, remains one of the most persistent challenges in this research domain.

Nevertheless, this phase fundamentally redefined the automation landscape by demonstrating that AI could infer design intent from visual data. This realization led to the next evolutionary phase: systems that not only translate visual input into code but also exhibit adaptive and generative intelligence in constructing novel designs.

➢ *AI-Driven and Generative Design Systems*

The third and most recent phase of evolution involves the emergence of AI-driven website generation platforms that incorporate generative models, natural language understanding, and user preference learning. These systems move beyond mimicking existing designs to creating new ones autonomously based on textual descriptions, voice commands, or inferred design preferences.

Commercially, tools such as *The Grid*, *Bookmark*, and *Wix ADI* pioneered the integration of AI into consumer-oriented web design [15, 16]. By analyzing user inputs such as business type, colour palette, and content tone, these platforms generate customized websites in real time. While their internal algorithms remain proprietary, they combine template-based generation with recommendation systems and, increasingly, deep generative models [24].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

From a research perspective, the AI-driven phase has been characterized by the convergence of multimodal AI, systems that combine vision, language, and structured data processing. Large pre-trained models, including Vision Transformers [25] and multimodal frameworks such as CLIP [26], have opened new possibilities for understanding aesthetic composition and user intent. These technologies enable the next generation of design automation tools to respond to natural language prompts ("generate a business homepage with a modern layout and dark theme") or to learn stylistic preferences from user examples.

Recent developments in Generative Adversarial Networks (GANs) and diffusion models have further advanced layout synthesis, allowing AI to propose novel website layouts that adhere to usability principles while maintaining creative diversity [27]. Coupled with reinforcement learning, these systems are increasingly capable of optimizing layouts for accessibility, readability, and performance metrics.

Despite their sophistication, current AI-driven systems face several open challenges:

- Interpretability – Understanding the decision-making process behind AI-generated designs remains difficult [24].
- Ethical design alignment – Ensuring that automated systems generate inclusive and culturally appropriate content is an emerging concern.
- Human–AI collaboration – Integrating designers into the loop to guide and refine AI-generated outcomes continues to be a key area of exploration [27].

➢ *Summary*

The evolution from heuristic systems to AI-driven frameworks marks a clear progression toward increasingly intelligent, autonomous, and human-aligned web design. Early systems relied on deterministic rule sets, while deep learning approaches introduced pattern recognition and layout generalization. The latest generation of AI-driven platforms extends beyond automation into creative co-design, where algorithms act as intelligent collaborators rather than mere translators of human intent.

This historical trajectory highlights two central insights. First, the abstraction gap between visual design and code remains the central technical challenge. Second, as AI continues to mature, the future of website generation lies in multimodal integration, combining visual understanding, language reasoning, and design theory into cohesive frameworks. The next section of this research systematically reviews the principal approaches to automatic website generation, categorizing them into mock-up-driven, example-based, and AI-driven paradigms to establish the foundation for a unified conceptual synthesis.

III. REVIEW OF STATE-OF-THE-ART APPROACHES

Automatic website generation has matured into a rich, interdisciplinary research domain that combines computer vision, machine learning, human–computer interaction, and software engineering. Over the past decade, three dominant methodological paradigms have emerged: (1) mock-up-driven approaches, (2) example-based approaches, and (3) AI-driven approaches. These paradigms differ in input representation, computational workflow, and automation scope, but all aim to reduce the design-to-code abstraction gap [6].

A. Mock-up-Driven Approaches

Mock-up-driven methods form the historical and conceptual foundation of automatic website generation. They treat visual design artifacts, hand-drawn sketches, wireframes, or digital mock-ups, as the primary input and attempt to reconstruct equivalent front-end code through a pipeline of element detection, classification, layout inference, and code synthesis.

➢ *Heuristic and Hybrid Methods*

Early mock-up-driven systems relied on CV heuristics to segment mock-ups and map detected regions to GUI components. Huang et al. [2] employed colour quantization and contour analysis to extract UI elements, later refined by rule-based proofreading. Nguyen et al. [1] introduced *REMAUI*, which used OCR for text detection and image-processing heuristics to reconstruct *Android* layouts. Although effective for small datasets, heuristic methods struggled with scale and generalization because handcrafted rules could not account for the diversity of web designs [3].

➢ *End-to-End Deep Learning Models*

The publication of *pix2code* [4] established a deep learning baseline for visual-to-code translation. Using an encoder–decoder pipeline, CNN for image encoding and LSTM for sequence decoding, the model generated DSL representations later compiled into HTML/CSS. Subsequent models introduced architectural variations:

- Zhu et al. [18] incorporated hierarchical layout constraints to improve DOM reconstruction.
- Han et al. [19] used GRUs to reduce training complexity.
- Kim et al. [21] integrated object-detection modules (YOLO) with CNN classifiers to enhance localization accuracy.

These methods significantly increased structural and visual fidelity but required extensive labelled datasets. As dataset creation remained labour-intensive, researchers explored synthetic data generation and transfer learning [23].

➢ *Object-Detection and Layout-Inference Methods*

Object-detection-based systems decompose website screen captures into atomic GUI elements using models such as Faster R-CNN or Single Shot MultiBox Detector (SSD) and then infer layout hierarchies through geometric reasoning. Microsoft's *Sketch2Code* and open-source

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

projects like *TeleportHQ* exemplify this line of work, converting sketches or static images into HTML/CSS layouts in near real time. Layout-inference research [28] focused on deriving fluid or elastic web layouts from fixed mock-ups to ensure responsiveness across screen sizes.

The advantage of mock-up-driven paradigms lies in their ability to maintain high visual similarity to designer-created artifacts. However, their dependence on large annotated datasets and limited understanding of semantic hierarchy remains unresolved. Table 1 summarizes representative mock-up-driven approaches.

Table 1: Summary of Mock-Up-Driven Approaches

Approach	Core Technique	Input Type	Output	Limitation
REMAUI [1]	OCR + Heuristics	Mock-up screenshots	Android XML	Single-page limitation
pix2code [4]	CNN + LSTM	Screenshots	HTML/CSS	Shallow hierarchy
Sketch2Code	Object detection + Heuristics	Hand-drawn sketch	HTML/CSS	Incomplete semantic mapping
Han et al. [19]	GRU Decoder	Wireframe	HTML	Dataset dependency
Kim et al. [21]	YOLO + CNN	Screenshots	HTML	Limited to static layouts

B. Example-Based Approaches

Example-based website generation focuses on design reuse, learning from existing websites to construct new ones. Instead of translating a mock-up, the system retrieves or analyzes reference designs from web repositories and reconstructs their structure for adaptation.

The foundational system *WebCrystal* [13] enabled designers to explore desirable website attributes by extracting and recombining CSS and HTML elements from real pages. Similarly, *GUIFetch* [14] searched software repositories such as *GitHub* and *Bitbucket* for *Android* GUIs visually similar to a user's sketch and returned the corresponding source code.

Other studies extended this paradigm to recommendation systems for UI layout inspiration. Hashimoto et al. [29] developed a program that retrieved visually compatible web templates for novice designers, while *Swire* [2] used deep feature embeddings to find mobile UI designs resembling a given query image.

These systems embody a case-based reasoning philosophy [30]: new designs are generated through adaptation and recombination of past exemplars. The approach excels at democratizing design by providing novices with professional templates but remains limited by dataset bias and lack of semantic annotation. Extracted HTML/CSS structures often lose functional coherence because style and behaviour (e.g., JavaScript) are not preserved. Table 2 summarizes notable example-based approaches.

Table 2: Summary of Example-Based Approaches

Approach	Source	Learning Basis	Contribution	Limitation
WebCrystal [13]	Real websites	CSS/HTML reuse	Re-composition of styles	Limited to static HTML
GUIFetch [14]	GitHub repos	Visual + code similarity	Source code retrieval	No image comparison
Swire [2]	Mobile UIs	Deep visual features	Design recommendation	Rare widgets not detected

Example-based paradigms represent an important intermediate step between procedural automation and intelligent generation, enabling the accumulation of design knowledge bases that underpin current AI-driven systems.

C. AI-Driven and Generative Approaches

AI-driven approaches integrate learning, reasoning, and generation into unified frameworks capable of synthesizing novel designs from abstract inputs such as text, voice, or user intent. This paradigm shift reflects advances in multimodal learning, transformer architectures, and generative modelling.

➢ *Commercial Intelligent Builders*

Tools such as *The Grid* [15], *Bookmark* [16] and *Wix ADI* use AI to generate entire websites after collecting high-level user information, industry type, branding colours, and textual descriptions. The underlying models employ a combination of template retrieval, rule-based reasoning, and content generation powered by large-scale user data [24]. Although these systems popularized the notion of "*AI web design*," their architectures remain largely proprietary and often constrained to single-page or modular layouts.

➢ *Academic Generative Frameworks*

In research contexts, generative approaches are rapidly evolving through integration with transformer-based and diffusion-based architectures. Recent models leverage Vision Transformers (ViTs) [25] and multimodal encoders like CLIP [26] to interpret textual and visual cues jointly. Experimental systems can now generate responsive website layouts from natural-language prompts (e.g., "*Create a landing page for an AI start-up with a dark theme*") or from design exemplars [27].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

In these systems, GANs or diffusion models produce layout grids, colour schemes, and typography that are subsequently rendered into HTML/CSS using learned code templates. Reinforcement-learning loops optimize generated designs for accessibility and visual balance [24].

➢ *Emerging Trends*

Key research directions shaping the future of AI-driven generation include:

- Human-in-the-Loop Design: Combining designer feedback with AI iteration cycles to maintain creative control.
- Explainable Design Intelligence: Developing interpretable AI models that justify design decisions [27].
- Cross-Domain Transfer: Applying web-layout learning to mobile and mixed-reality interfaces.

Table 3 provides a concise comparison of AI-driven website builders.

Table 4: A Comparison of AI-Driven Website Builders

System	Input	Core AI Technique	Output Scope	Key Advantage	Limitation
The Grid [15]	Questionnaire / Content	Template + Rule AI	Multi-page site	First AI builder	Low customizability
Bookmark [16]	User survey + intent	Recommendation ML	Single page	User-friendly workflow	Limited style diversity
Wix ADI	Text + Template selection	Hybrid ML + NLP	Full website	Rapid deployment	Weak SEO integration

D. Comparative Analysis

Across paradigms, the field shows a consistent movement from explicit programming to implicit learning. Mock-up-driven systems excel in visual fidelity; example-based systems offer design reuse; and AI-driven systems aim for adaptive intelligence. The comparative synthesis below highlights distinguishing characteristics.

Table 5 Comparative Analysis

Criterion	Mock-up-Driven	Example-Based	AI-Driven
Input	Sketch / Mock-up	Existing websites	Text / Voice / Mixed
Technique	CV + DL + Heuristics	Case-based reasoning	Generative AI + Transformers
Output	HTML/CSS Prototype	Adapted Template	Full Website
Strength	High visual accuracy	Reuse of best designs	Creativity and automation
Weakness	Dataset dependence	Limited semantics	Interpretability and control

The comparison underscores that no single paradigm fully addresses both visual and semantic fidelity. This motivates the conceptual synthesis proposed later in this research, which integrates the interpretive strengths of mock-up-driven methods, the knowledge reuse of example-based learning, and the generative adaptability of AI-driven systems into a unified framework.

IV. SYNTHESIS AND EMERGING TRENDS

The evolution of automatic website generation reflects an ongoing convergence of technological capability and design intelligence. While early systems were primarily procedural, recent approaches demonstrate increasing autonomy, learning capacity, and creative adaptability. This section synthesizes cross-cutting insights from the reviewed paradigms, mock-up-driven, example-based, and AI-driven, and identifies the major emerging trends that define the trajectory of this research field.

➢ *Integrative Technological Convergence*

Modern website generation systems no longer operate within a single methodological paradigm. Instead, they represent hybrid architectures that integrate CV, NLP, and DL modules into unified pipelines. For instance, advanced mock-up-driven frameworks now embed CNN-based feature extractors alongside rule-based post-processing to preserve layout precision [21], while AI-driven builders increasingly incorporate visual similarity retrieval engines reminiscent of example-based systems [13, 14].

This fusion reflects a broader trend in multimodal learning, where visual, textual, and structural data are jointly modelled to enhance semantic understanding. Transformer-based architectures such as CLIP [26] and Vision Transformers [25] provide a shared embedding space that aligns GUI features with linguistic cues, enabling systems to interpret both design imagery and user intent. Such cross-domain integration has moved automatic website generation from pattern recognition toward semantic reasoning, allowing algorithms to infer relationships between layout components, content hierarchy, and user experience goals.

➢ *Emergence of Design Intelligence*

A key conceptual shift in this domain is the move from automation to intelligence. Early systems automated repetitive coding tasks; current research seeks to model cognitive aspects of design, such as balance, contrast, alignment, and visual rhythm, through learned representations [24]. Deep generative networks and reinforcement-learning agents can now evaluate design

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

alternatives using reward functions based on usability and aesthetic metrics [27].

This evolution parallels developments in computational creativity and design cognition, where AI agents participate as collaborators rather than tools. In human–AI co-design scenarios, algorithms propose layout variants that human designers can refine, effectively blending machine efficiency with human sensibility [29]. The notion of design intelligence thus extends beyond technical automation to encompass reasoning about context, style, and intent, core attributes of human design expertise.

➢ *Data-Centric Foundations and Benchmarking*

Despite algorithmic progress, the advancement of this field remains constrained by the availability and quality of datasets. Publicly accessible GUI datasets, such as RICO, Enrico, and GUISpirit, have supported mobile interface research but are limited in diversity and web-specific semantics [7, 20]. The absence of standardized benchmarks impedes cross-model comparison and reproducibility.

Emerging initiatives now emphasize data-centric AI, focusing on dataset curation, annotation consistency, and synthetic data augmentation [18]. Automatically generated or semi-synthetic datasets, created through procedural layout rendering, help overcome data scarcity while maintaining control over design variability. Furthermore, multimodal datasets that pair screenshots with HTML, CSS, and textual descriptions enable holistic learning of design–code correspondences.

Establishing open benchmarks for layout fidelity, code quality, and semantic alignment is essential for the maturation of this research domain. Shared evaluation protocols will allow researchers to measure progress consistently and promote transparency comparable to established computer-vision tasks such as ImageNet or COCO.

➢ *Human-in-the-Loop Design and Explainability*

As AI systems assume greater autonomy, ensuring human oversight and interpretability has become a central concern. Human-in-the-loop (HITL) design frameworks integrate user feedback directly into the learning cycle, allowing designers to guide model outputs iteratively [27]. This approach ensures that generated websites reflect aesthetic judgment, contextual appropriateness, and accessibility requirements.

Complementary to HITL, explainable AI (XAI) techniques are being adapted for design automation. Visualization of attention maps in CNN-transformer hybrids, for example, allows users to understand which regions of a mock-up influence specific layout decisions [24]. Explainability is not only a matter of transparency but also a foundation for trust and adoption, especially in professional design environments where creative accountability is critical.

➢ *Generative AI and Multimodal Web Creation*

The proliferation of large language models (LLMs) and diffusion-based generative models marks a transformative stage in website generation. Models such as GPT-4 Vision and Stable Diffusion have demonstrated the ability to translate natural-language prompts into visual compositions, bridging textual semantics and graphical layout [32].

Emerging prototypes, such as *Web2Code* [33] extend this capability to full-stack web synthesis, where a user's instruction ("*Build a portfolio site with a minimalist layout and blue accent palette*") triggers automatic creation of a functional website with adaptive design and sample content. These multimodal systems represent the convergence of NLP-driven content generation and CV-driven layout synthesis, heralding an era of end-to-end web automation.

However, integrating these generative capabilities into controlled, production-grade workflows requires addressing issues of code correctness, accessibility compliance, and intellectual-property provenance. The next generation of frameworks will likely employ constraint-guided generation, combining creativity with rule enforcement to ensure both novelty and reliability.

➢ *Ethical and Socio-Technical Considerations*

As design automation matures, ethical and socio-technical dimensions gain prominence. Automated systems can inadvertently propagate biases present in training data, such as culturally specific colour associations or gendered imagery. Scholars have called for responsible design AI that incorporates fairness, inclusivity, and sustainability into its generative logic [34].

Moreover, the democratization of design through AI raises questions about professional identity and creative authorship. While automated tools lower barriers for non-experts, they also challenge traditional notions of design expertise and originality. Future frameworks must balance accessibility with the preservation of human creativity and aesthetic diversity.

➢ *Summary*

The synthesis of existing paradigms and emerging research directions reveals a field in rapid transformation. Three overarching trends can be distilled:

- Integration: The fusion of CV, NLP, and DL has produced multimodal systems capable of semantic reasoning about design.
- Intelligence: Generative and reinforcement-learning models are evolving toward genuine design cognition, transcending procedural automation.
- Interaction: Human–AI collaboration and explainability are becoming essential for sustainable adoption and ethical practice.

These trends collectively point toward a unified conceptual framework that captures the complete workflow of AI-driven website generation, from visual input interpretation to semantic synthesis and explainable code production.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

V. CONCEPTUAL FRAMEWORK FOR AI-DRIVEN WEBSITE GENERATION

The review and synthesis of previous research highlight the fragmented nature of existing approaches to automatic website generation. While significant progress has been made in visual recognition, layout inference, and code generation, the field lacks a unified theoretical foundation that integrates these components into a coherent design automation workflow. To address this gap, this section introduces a conceptual framework for AI-driven website generation.

The framework extends the design philosophy of *WebDraw* [35] and synthesizes technical, cognitive, and human-centered dimensions into a structured model comprising five core layers: (1) visual input acquisition and pre-processing, (2) feature extraction and GUI element recognition, (3) semantic understanding and hierarchical structuring, (4) code generation and rendering, and (5) evaluation, feedback, and iterative learning. Each layer performs a distinct but interdependent role in achieving an end-to-end design-to-code transformation process.

- *Layer 1: Visual Input Acquisition and Pre-processing*

The first stage of the framework involves collecting and preparing visual inputs, which may include wireframes, mock-ups, sketches, or screen captures. These inputs form the visual representation of the intended website layout. Pre-processing ensures that the data is standardized and optimized for downstream analysis.

Techniques such as grayscale normalization, contour enhancement, and image segmentation are used to isolate GUI components and remove visual noise [2]. In multimodal systems, textual and voice-based prompts are also processed using NLP models to extract contextual requirements such as layout type ("*portfolio website*") or visual tone ("*modern minimalism*") [26, 27].

At this stage, data augmentation and synthetic mock-up generation may also be applied to expand the training dataset. The resulting pre-processed inputs provide a uniform foundation for machine learning models, facilitating robust and generalizable feature learning.

- *Layer 2: Feature Extraction and GUI Element Recognition*

The second layer is responsible for identifying and classifying the constituent elements of a web interface. Leveraging deep learning, particularly CNNs and object-detection models (e.g., YOLO, Faster R-CNN), this layer localizes interface components such as navigation bars, buttons, input fields, and text containers [20, 21].

Each detected element is then classified into semantic categories (e.g., header, paragraph, image container, or call-to-action). This process mirrors the visual grammar of design, translating aesthetic cues into machine-understandable representations. Advanced models incorporate attention mechanisms and vision transformers (ViTs) to capture relational dependencies between elements, such as alignment, proximity, and grouping, reflecting design principles like Gestalt proximity and balance [24, 25].

The output of this layer is a structured map of GUI components annotated with positional, visual, and semantic metadata, forming the basis for layout interpretation.

- *Layer 3: Semantic Understanding and Hierarchical Structuring*

The third layer bridges the gap between visual detection and logical design understanding. It constructs a DOM-like hierarchy that defines parent–child relationships among GUI elements.

This stage employs a combination of graph-based reasoning and sequence modelling. RNNs or transformer decoders learn to sequence elements in syntactic order, while graph neural networks (GNNs) capture spatial dependencies and containment structures [18].

Semantic understanding extends beyond geometry to include functional role inference, for example, recognizing that a button labelled "*Submit*" likely belongs within a form container. Such reasoning is enhanced through contextual embeddings, where co-occurrence patterns in training data inform semantic grouping.

In this framework, the hierarchical structure acts as the intermediate design representation, unifying visual recognition and code generation. It encapsulates both spatial relationships and semantic meaning, providing the scaffolding for subsequent rendering.

- *Layer 4: Code Generation and Rendering*

The fourth layer translates structured design representations into machine-readable and executable code. Depending on system implementation, this may involve:

- DSL generation (as in *pix2code* [4]),
- Template-based HTML/CSS synthesis, or
- Hybrid neural code generation using transformer-based language models (e.g., *CodeT5*, *Codex*).

The model interprets the hierarchical layout from Layer 3 and produces corresponding markup and styling instructions. Each GUI element's semantic label and visual attributes (colour, typography, spacing) are mapped to their equivalent CSS properties. For dynamic content, the model can also generate JavaScript placeholders or data-binding stubs.

Unlike early heuristic models, this framework allows for adaptive code generation, wherein learned design patterns are applied to new contexts with minimal manual intervention. Recent multimodal systems integrate LLMs for enhanced contextual understanding, enabling prompt-based customization [32, 33].

Rendering validation ensures visual fidelity between generated and target layouts through pixel-level comparison or structural similarity metrics such as SSIM and DOM overlap ratio [35].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

➢ *Layer 5: Evaluation, Feedback, and Iterative Learning*

The final layer introduces evaluation and continuous improvement mechanisms. Outputs are assessed along three dimensions:

- Visual Accuracy – comparison of rendered output to original design mock-up using image similarity metrics.
- Structural Integrity – validation of the DOM hierarchy and semantic correctness of generated code.
- Usability and Accessibility – assessment of alignment with web accessibility standards (e.g., WCAG 2.2).

Feedback from these evaluations, as well as from human experts or end users, is reintegrated into the model via iterative retraining or reinforcement-learning loops. This enables continuous refinement of recognition accuracy, code quality, and aesthetic sensitivity [27].

By incorporating feedback cycles, the framework embodies design-science principles of iteration, evaluation, and theory building. The process thus evolves from static automation toward adaptive learning, reinforcing its alignment with both academic rigor and real-world application.

➢ *Conceptual Model Overview*

The five layers collectively form a closed-loop ecosystem that mirrors human design cognition: perception (visual input), comprehension (semantic structuring), creation (code synthesis), and reflection (evaluation). A conceptual diagram of this framework (Figure 1) would typically depict a left-to-right pipeline, from visual input through iterative feedback, connected by arrows indicating continuous learning and human–AI interaction.

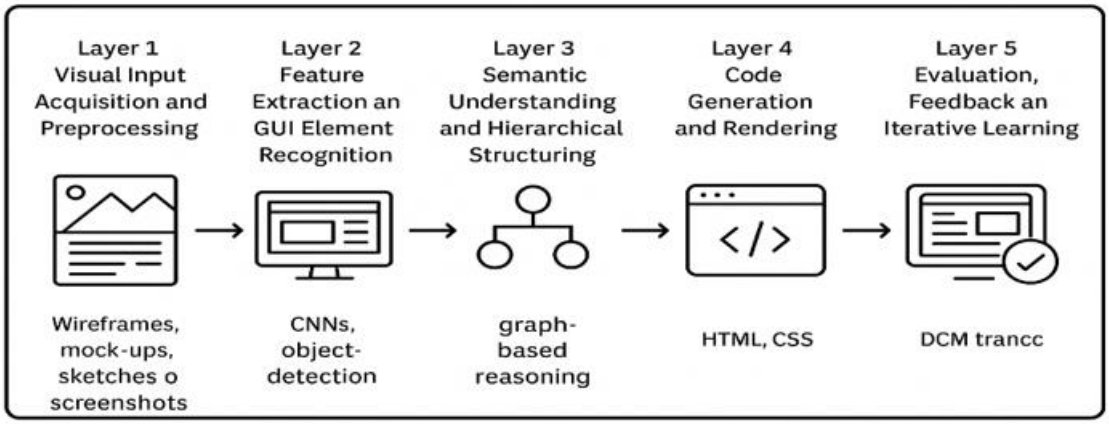

Fig 1 Conceptual Framework for AI-Driven Website Generation

The figure illustrates a multilayer pipeline where visual inputs (mock-ups, sketches, or textual prompts) enter from the left. CNN-based recognition modules identify GUI components, which feed into a semantic reasoning engine that constructs a DOM hierarchy. The resulting structure passes to a code-generation module, producing HTML/CSS/JS output rendered on a browser interface. An evaluation loop on the right captures' performance metrics and designer feedback, which return to the learning modules to refine subsequent outputs.

This conceptual model advances the state of the art by:

- Integrating perception and reasoning into a single pipeline;
- Bridging visual, structural, and semantic representations;
- Embedding human feedback loops for continual improvement; and
- Providing a theoretical framework for analyzing and comparing diverse automatic website generation systems.

➢ *Theoretical and Practical Significance*

The proposed framework contributes to both theoretical understanding and practical implementation of AI-driven design automation. Theoretically, it offers a structured abstraction that unifies disjoint technical paradigms under a design-science lens, aligning computational processes with cognitive design stages. Practically, it provides a roadmap for developing extensible, explainable, and human-centered website generation tools.

By integrating CV, machine learning, and human–computer interaction, the model demonstrates how AI can embody design reasoning, transforming the field from automation to intelligent collaboration. As such, it lays the foundation for future research that explores how machines learn to design, ultimately closing the long-standing gap between creative design intent and computational implementation.

VI. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Although significant progress has been achieved in automating website design and development, the field of AI-driven website generation continues to face multiple research challenges. These challenges stem from both technical constraints, such as data availability, model interpretability, and code quality, and socio-technical issues, including ethical design, accessibility, and sustainability. Addressing these limitations will determine how effectively AI can integrate with creative design workflows and deliver scalable, trustworthy solutions for web automation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

➢ *Dataset Availability and Standardization*

One of the most pressing limitations in current research is the lack of standardized, large-scale, and web-specific datasets. While mobile UI datasets such as RICO and Enrico have facilitated interface recognition research [7, 20], there remains no equivalent, comprehensive dataset for web interfaces. Existing datasets often differ in annotation granularity, visual diversity, and markup format, making it difficult to compare model performance objectively.

Future work must focus on developing benchmark datasets that capture the heterogeneity of modern web design, including responsive layouts, dynamic elements, and accessibility features. Collaborative data collection initiatives, possibly supported by open-source communities and design platforms, could lead to the creation of standardized corpora that encompass both visual and semantic annotations.

In addition, data governance frameworks are required to ensure that datasets adhere to privacy, copyright, and ethical design guidelines. The adoption of data-centric AI principles [18] will enable researchers to prioritize dataset quality and representativeness, rather than focusing solely on model complexity.

➢ *Evaluation Metrics and Benchmarking*

Evaluating AI-generated websites poses unique challenges. Traditional metrics such as accuracy or F1-score capture only the correctness of component detection, not the aesthetic quality or usability of generated designs. Current evaluation protocols lack consensus on measuring visual fidelity, structural similarity, and code efficiency in an integrated manner. A standardized benchmarking framework should include:

- Visual Similarity Metrics – using SSIM or pixel-level comparisons between generated and reference layouts.
- Structural Metrics – measuring DOM hierarchy alignment and semantic completeness.
- Usability Metrics – incorporating user experience evaluations through expert review or crowd-sourced feedback.
- Performance Metrics – assessing code quality, load time, and accessibility compliance.

Hybrid evaluation methodologies that combine quantitative indicators with qualitative user studies would yield a more holistic understanding of model performance. Establishing such benchmarks will not only improve reproducibility but also drive comparative innovation across research groups.

➢ *Model Interpretability and Explainability*

As AI systems gain autonomy in design decision-making, the need for interpretability and explainable design intelligence becomes critical. Current deep learning models often operate as black boxes, offering limited insight into why specific layout or styling choices are made. This opacity poses challenges in debugging, trust, and accountability, particularly in professional design contexts.

Future frameworks should integrate explainable AI (XAI) mechanisms that visualize model attention, decision pathways, or rule hierarchies [24]. For example, saliency maps can indicate which visual regions most influence layout reconstruction, while natural-language explanations can justify design decisions (“This button was placed near the header for navigational coherence”).

Incorporating interpretable architectures, such as GNNs or attention-based decoders, could enable transparent reasoning about design intent. Enhancing interpretability will also facilitate human-in-the-loop (HITL) collaboration, where designers actively guide model behavior through interactive feedback.

➢ *Generalization and Domain Adaptation*

Another research challenge lies in achieving generalization across design domains. Models trained on specific datasets (e.g., corporate websites) often fail when applied to stylistically divergent domains such as e-commerce, educational platforms, or multimedia portfolios. Similarly, transferring models across modalities, from mobile to web, or web to mixed reality, remains nontrivial due to contextual variation in interaction design.

Future studies should explore domain adaptation and transfer learning techniques that allow models to retain core layout reasoning while adjusting to new aesthetic and functional contexts. Generative diffusion models and foundation models pretrained on diverse multimodal data [25, 32] offer promising avenues for cross-domain generalization. In parallel, few-shot learning approaches can help systems adapt rapidly to new design styles using minimal data, reducing reliance on large annotated datasets.

➢ *Human–AI Collaboration and Co-Creation*

While automation remains a key goal, the future of design lies in collaboration rather than replacement. AI systems should function as *creative partners*, generating initial layouts or suggestions that human designers refine and contextualize [31].

To achieve this, research must focus on interaction design for AI tools, how users can communicate intent, critique outputs, and iteratively guide the model. *Human-in-the-loop* architectures that enable bidirectional feedback between designer and AI can improve both usability and creativity.

Furthermore, integrating multimodal input modalities (e.g., sketches, speech, gestures) will allow natural and fluid collaboration, making AI-driven systems more accessible to designers with varying technical expertise.

➢ *Ethics, Bias, and Responsible Design*

As AI assumes a creative role, ethical and cultural implications become unavoidable. Datasets may encode sociocultural biases, for instance, favouring Western design aesthetics, colour symbolism, or text directionality. If unaddressed, such biases risk propagating cultural homogeneity and undermining inclusivity [34].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

Future research should develop bias detection and mitigation mechanisms in design datasets and model outputs. This includes monitoring for representation bias, ensuring accessibility compliance (WCAG 2.2), and supporting localization for multilingual or culturally diverse audiences.

Moreover, frameworks for responsible design AI must establish transparent attribution for AI-generated content, clarifying authorship and ownership in collaborative creative processes. Ethical auditing tools could assess fairness and inclusivity metrics alongside technical performance indicators.

➢ *Integration with Large Language and Multimodal Models*

The rapid evolution of LLMs and multimodal AI systems offers transformative opportunities for future website generation frameworks. LLMs such as GPT-4 Vision and Claude 3 can now interpret visual layouts, generate front-end code, and respond to complex natural-language instructions [32]. Integrating such models into web-generation pipelines can provide contextual reasoning, personalized content creation, and adaptive layout generation.

Future research could focus on:

- Prompt engineering strategies to control design parameters (layout complexity, colour palette, typography).
- Constraint-based generation to align creative freedom with usability and accessibility rules.
- Hybrid symbolic–neural architectures that combine LLM reasoning with structured DOM logic.

These advancements will enable fully end-to-end website generation systems capable of transforming abstract ideas or business goals into deployable, visually coherent, and semantically accurate web experiences.

➢ *Sustainability and Computational Efficiency*

The computational demands of large-scale AI models present environmental and economic challenges. Training multimodal models for design generation consumes significant energy, raising questions about sustainable AI practices.

Future research should prioritize energy-efficient architectures, model compression, and incremental training approaches to reduce carbon footprints. Leveraging federated learning and edge computing can also distribute training loads while maintaining data privacy. Developing sustainability-aware design AI aligns with global goals for green computing and responsible innovation.

➢ *Summary of Future Directions*

The future of AI-driven website generation will be shaped by the convergence of three complementary dimensions:

- Technological Advancement – Integration of LLMs, multimodal learning, and explainable AI.
- Human–AI Collaboration – Development of co-creative, interpretable, and feedback-driven workflows.
- Ethical and Sustainable Practice – Implementation of fairness, inclusivity, and energy-efficient design principles.

As these dimensions mature, the discipline will evolve beyond automation toward intelligent, adaptive, and responsible co-design ecosystems that seamlessly bridge creativity and computation.

VII. CONCLUSION

The increasing complexity of modern digital interfaces and the growing demand for rapid, high-quality web development have driven significant interest in automating the design-to-code pipeline. Over the past decade, research in automatic website generation has evolved through a series of methodological paradigms, ranging from heuristic-driven approaches to deep learning models and, more recently, to multimodal AI systems capable of generative design. Each paradigm has contributed unique insights, yet the field has remained conceptually fragmented, lacking a unified framework that captures the interconnected nature of perception, reasoning, and generation in AI-assisted design.

This research addressed this gap by synthesizing the diverse technical approaches into a coherent and comprehensive account of the state of the art. Beginning with a historical overview of heuristic and rule-based systems, the discussion traced the emergence of deep learning architectures that enabled visual understanding and code synthesis at an unprecedented scale. It also examined example-based paradigms that introduced reusable design knowledge, as well as AI-driven generative systems that now support adaptive layout creation and content generation. Through this structured review, the research highlighted the strengths and limitations of each approach, revealing the need for integrative models that bridge visual fidelity, semantic correctness, and creative adaptability.

Building on these insights, the research introduced a five-layer conceptual framework for AI-driven website generation. Grounded in the design-science foundations of your doctoral research, this framework unifies the sequential and iterative stages of the automation process: visual input acquisition, feature extraction, semantic structuring, code generation, and iterative evaluation. The model demonstrates how modern AI systems can learn not only to recognize the components of a design but also to reason about their structural relationships and transform them into coherent, executable web code. By embedding evaluation and feedback loops, the framework advances the field toward systems that continuously evolve through user interaction, model refinement, and contextual adaptation.

The research also examined critical research challenges, including dataset scarcity, evaluation standardization, interpretability, domain generalization, ethical constraints, and sustainability concerns. These challenges represent essential opportunities for future research. As multimodal AI, large language models, and generative diffusion models continue to advance, the possibility of end-to-end text-to-

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

website generation becomes increasingly feasible. At the same time, the importance of human oversight, explainability, and responsible creativity becomes even more central. The future of this field lies not in fully replacing designers but in enabling meaningful human–AI co-creation, where algorithms augment human creativity while upholding usability, accessibility, and ethical integrity.

In conclusion, the synthesis and framework presented in this research contribute to a deeper theoretical understanding of how AI can embody design reasoning and participate in creative computational processes. As researchers and practitioners continue to explore the intersection of artificial intelligence, human–computer interaction, and web engineering, the vision of intelligent, adaptive, and inclusive website generation becomes increasingly tangible. This work thus serves as both a conceptual foundation and a roadmap for advancing the next generation of AI-enabled design systems, systems that not only automate but also elevate the practice of website creation.

ACKNOWLEDGMENTS

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1]. Nguyen, T. A., & Csallner, C. (2015, November). R*everse engineering mobile application user interfaces with remaui (t)*. In 2015 30th IEEE/ACM international conference on automated software engineering (ASE) (pp. 248-259). IEEE.

[2]. Huang, F., Canny, J. F., & Nichols, J. (2019, May). *Swire: Sketch-based user interface retrieval*. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1-10).

[3]. Rivero, J. M., Rossi, G., Grigera, J., Burella, J., Luna, E. R., & Gordillo, S. (2010, July). *From mockups to user interface models: an extensible model driven approach*. In International Conference on Web Engineering (pp. 13-24). Berlin, Heidelberg: Springer Berlin Heidelberg.

[4]. Beltramelli, T. (2018, June). *pix2code: Generating code from a graphical user interface screenshot*. In Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems (pp. 1-6).

[5]. Han, L., Xu, Y., & Zhang, C. (2020). *SketchCode: Generating HTML code from hand-drawn web wireframes using gated recurrent units*. Computers & Graphics, 92, 72–83.

[6]. Kaluarachchi, T., & Wickramasinghe, M. (2023). *A systematic literature review on automatic website generation*. Journal of Computer Languages, 75, 101202. https://doi.org/10.1016/j.cola.2023.101202

[7]. Moran, K., Bernal-Cárdenas, C., Curcio, M., Bonett, R., & Poshyvanyk, D. (2018). *Machine learning-based prototyping of graphical user interfaces for mobile apps*. IEEE transactions on software engineering, 46(2), 196-221.

[8]. de Souza Baulé, D., von Wangenheim, C. G., von Wangenheim, A., & Hauck, J. C. (2020). *Recent Progress in Automated Code Generation from GUI Images Using Machine Learning Techniques*. J. Univers. Comput. Sci., 26(9), 1095-1127.

[9]. Gui, Y., Wan, Y., Li, Z., Zhang, Z., Chen, D., Zhang, H., ... & Zhang, X. (2025, April). *UICoPilot: Automating UI synthesis via hierarchical code generation from webpage designs.* In Proceedings of the ACM on Web Conference 2025 (pp. 1846-1855).

[10]. Abdelhamid, A. A., Alotaibi, S. R., & Mousa, A. (2020). *Deep learning-based prototyping of android GUI from hand-drawn mockups*. IET Software, 14(7), 816-824.

[11]. Chen, Y., Ding, S., Zhang, Y., Chen, W., Du, J., Sun, L., & Chen, L. (2025). *DesignCoder: Hierarchy-Aware and Self-Correcting UI Code Generation with Large Language Models*. arXiv preprint arXiv:2506.13663.

[12]. Saravanan, V., SS, M. A., Fatima, N. S., & Mahalakshmi, P. (2021). *Automated web design and code generation using deep learning*. Turkish Journal of Computer and Mathematics Education, 12(6), 364-373.

[13]. Myers, B. A., Hudson, S. E., & Pausch, R. (2015). *WebCrystal: Example-based design exploration for web development*. ACM Transactions on Computer-Human Interaction, 22(6), 1–29.

[14]. Behrang, F., Ali, A., & Karim, A. (2021). *GUIFetch: Retrieving graphical user interface source code from repositories using visual similarity*. Information and Software Technology, 135, 106560.

[15]. Tocchini, D. (2017). *The Grid: AI-based web design for everyone.* The Grid Press Release.

[16]. Kosmayer, D. (2019). *Bookmark: An AI-driven website builder platform*. Bookmark Inc. White Paper.

[17]. Kaluarachchi, T. T., Dissanayake, D. M. S., & Wickramasinghe, M. I. E. (2025). *Unsupervised Discovery of Salient Design Features of Websites*. The International Journal on Advances in ICT for Emerging Regions, 18(2). https://doi.org/10.4038/icter.v18i2.7301

[18]. Zhu, Z., Xue, Z., & Yuan, Z. (2018, December). *Automatic graphics program generation using attention-based hierarchical decoder*. In Asian Conference on Computer Vision (pp. 181-196). Cham: Springer International Publishing.

[19]. Han, Y., He, J., & Dong, Q. (2018, October). *CSSSketch2Code: An automatic method to generate web pages with CSS style*. In Proceedings of the 2nd International Conference on Advances in Artificial Intelligence (pp. 29-35).

[20]. Liu, Y., Hu, Q., & Shu, K. (2018, November). *Improving pix2code based Bi-directional LSTM*. In 2018 IEEE International Conference on Automation, Electronics and Electrical Engineering (AUTEEE) (pp. 220-223). IEEE.

[21]. Kim, B., Park, S., Won, T., Heo, J., & Kim, B. (2018, October). *Deep-learning based web UI automatic programming*. In Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems (pp. 64-65).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov339

[22]. Suleri, S., Sermuga Pandian, V. P., Shishkovets, S., & Jarke, M. (2019, May). *Eve: A sketch-based software prototyping workbench*. In Extended abstracts of the 2019 CHI conference on human factors in computing systems (pp. 1-6).
[23]. Hassan, S., Arya, M., Bhardwaj, U., & Kole, S. (2018). *Extraction and classification of user interface components from an image*. International Journal of Pure and Applied Mathematics, 118(24), 1-16.
[24]. Avdić, A. (2024). *Generative AI Tools in Web Design*. In Sinteza 2024-International Scientific Conference on Information Technology, Computer Science, and Data Science (pp. 392-397). Singidunum University.
[25]. Dosovitskiy, A. (2020). *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv preprint arXiv:2010.11929.
[26]. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). *Learning transferable visual models from natural language supervision.* In International conference on machine learning (pp. 8748-8763). PmLR.
[27]. Borah, A. R., Rani, K. P., Ziara, S., Lakhanpal, S., & Vidyadhari, C. (2025, February). *AI-Assisted Data Modelling and High-Resolution Image Synthesis: An Interactive Framework with Latent Diffusion Model Visualization*. In 2025 International Conference on Intelligent Control, Computing and Communications (IC3) (pp. 770-776). IEEE.
[28]. Sinha, N., & Karim, R. (2013, August). *Compiling mockups to flexible uis*. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (pp. 312-322).
[29]. Hashimoto, Y., & Igarashi, T. (2005). *Retrieving Web Page Layouts using Sketches to Support Example-based Web Design*. In SBM (pp. 155-164).
[30]. Watson, I., & Marir, F. (1994). *Case-based reasoning: A review*. The knowledge engineering review, 9(4), 327-354.
[31]. Kadenhe, N., Al Musleh, M., & Lompot, A. (2025, August). *Human-AI Co-Design and Co-Creation: A Review of Emerging Approaches, Challenges, and Future Directions*. In Proceedings of the AAAI Symposium Series (Vol. 6, No. 1, pp. 265-270).
[32]. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). *Gpt-4 technical report*. arXiv preprint arXiv:2303.08774.
[33]. Yun, S., Thushara, R., Bhat, M., Wang, Y., Deng, M., Wang, J., ... & Shen, Z. (2024). *Web2code: A large-scale webpage-to-code dataset and evaluation framework for multimodal llms*. Advances in neural information processing systems, 37, 112134-112157.
[34]. Shneiderman, B. (2022). *Human-centered AI*. Oxford University Press.
[35]. Kaluarachchi, T., & Wickramasinghe, M. (2024). *WebDraw: A machine learning-driven tool for automatic website prototyping*. Science of Computer Programming, 233, 103056. https://doi.org/10.1016/j.scico.2023.103056