Beyond Code: Tackling Limitations in Website Builders with Sketch-to-Code and Alternative Tools

Boon Hian Lim[a]

[a] *College of Computing and Data Science, Nanyang Technological University* LIMB0061@e.ntu.edu.sg

Abstract

The rise of website builders has simplified online presence creation for individuals and businesses. However, existing platforms face limitations which hinder broader adoption. This study examines these challenges and proposes a novel proof-of-concept website builder that introduces cutting-edge features to enhance user experience, such as Large Language Model (LLM) powered Sketch-to-Code functionality, a Plugins-of-Plugins system, and Git-powered version control. Our solution prioritises usability, allowing users to focus on their business needs rather than technical complexities.

1. Introduction

The advent of technology has revolutionised the way businesses and individuals promote themselves online. With a well-designed website, companies can create a positive image, capture customers' attention, and deliver first-hand information. For individuals, custom websites serve as a platform for showcasing their portfolio. However, building a custom website can be complicated and requires well-versed technical expertise.

To address this challenge, website builders have emerged as a convenient alternative. These platforms provide drag-and-drop editors, templates, and plug-to-play plugins that enable users to create and maintain websites without extensive technical knowledge. Despite their ease of use, many users experience limitations with website builders, such as slow editors and unpleasant user experience.

This study investigates the limitations of current website builders and proposes a new proof-of-concept website builder that addresses these restrictions with cutting-edge technologies such as having LLM-powered full and partial Sketch-to-Code functionality, a Plugins-of-Plugins system, and Git-powered version control.

In this study, we define a website builder as a solution for designing and building websites without coding requirements [1]. In this study, we focus on website builders that do not restrict themselves to a specific type of content, such as Wix [2] or WordPress [3], and ignore those domain-specialised CMSs, such as Shopify [4], an e-commerce CMS or Blackboard by Anthology [5], a Learning Management System (LMS).

2. Related Work

Website Builder User Experience. Prior studies have identified limitations of website builders through user studies or development and have since proposed improvements. Oswal and Oswal identified the lack of accessibility in three website builders, Dorik, Relume and Wix [6]. Palmer and Oswal further argued that the websites generated by website builders lack considerations of disability as well [7]. Mamatha SK et al. suggested integrating an AI-powered recommendation engine to enhance user experience and support concurrent editing by multiple users [8]. Andy argued that the abstraction of website builders hinders customizability and suggested building simple features on website builders and having complex logic on custom-coded websites [9]. Jasper supported Andy's argument and further identified editor performance issues, lack of version control, and vendor lock-in of website builders [10].

Design to Frontend Code Generation. Design-to-frontend code generation has been a heavily studied topic. The first few pioneer research was done by Landay and Myers' SILK[11], Beltramelli's pix2code [12], and Azure [13]. These early studies leveraged computer vision techniques. More recently, researchers have explored machine-learning approaches, such as those demonstrated by Robinson [14], Jain et al. [15] and Soselia et al. [16]. Pix2Struct [17] is one of the examples, which is a vision model pre-trained to convert webpages' screenshots into HTML but can be finetuned to convert sketch to code as demonstrated in WebCode2M [18]. The latest studies often exploit the multimodal LLMs' capability to generate HTML instead, such as Design2Code by Si et al. [19] and Sketch2Code by Li et al. [20].

Contributions. The contributions of this paper are as follows:

- **Partial Sketch-to-Code**. To the best of our knowledge, this paper is the first attempt to explore the feasibility of such features with LLMs, where we generate code from users' minor modifications of the current website. We argue that this is a better-suited use case for non-technical users.
- **Dataset Creation and Evaluation.** Building upon the work in WebCode2M, we modify their dataset for the partial sketch-to-code task evaluation. We also adopt evaluation metrics proposed in Design2Code [19] and further extend them by adding the generation success rate to provide insights into the accuracy of LLMs.
- **User-Centric Applications.** We build proof-of-concept to explore cutting-edge features in website builders to identify potential improvements in user experience.

3. Literature Review

3.1 Market Analysis

Wix. Wix is a cloud-based, proprietary website builder platform that enables users to create and customise websites. It supports customisation through plugins or Wix Velo, a full-stack development platform that allows developers to build both frontend and backend Javascript code that is executed during web page load time [21]. Wix has also embraced AI as its solution. With Wix ADI, users can create a website by talking with a chatbot. The generated websites are templates populated with mock details, which can be further modified in the drag-and-drop editor. Users

can further generate text or even images with text input in said editor [22].

Wix has its limitations and restrictions. Its editor is sometimes slow and even crashes on its own [23], [24], [25], which may be due to Wix loading all web pages on the website at once when loading the editor [26]. Its built-in plugin is fundamental and does not support further customisation. As of January 2025, Wix Bookings Velo API does not support varied pricing for online payment, despite the plugin containing such functionality to set varied pricing for a booking service [27]. Wix's site history feature also merely consists of a snapshot of the frontend design, whereas changes done in its CMS are not traced.

10Web AI Builder. 10Web AI Builder [28] is a commercial website builder that provides WordPress hosting services and also allows the generation of WordPress websites with AI. Users are to fill in their requirements, which would be further enhanced and improved by some LLMs. Multiple websites will then be generated, and an overview of the layouts of these websites will be presented to the users. Users, however, cannot modify them; they can only provide styles or font sizes to be used on the websites.

3.2 Insights

As compared to the chatbot, we identify that users may have a visual of the frontend design, and it will be much more convenient for the user to draw out their prototype instead of describing it through text. Hence, we propose integrating sketch-to-code in the website builder. The limited functionalities of plugins can be mitigated by enabling them to support extensions. WooCommerce, a WordPress plugin, exemplifies this by allowing extensions to enhance its functionality [29]. Version controls are often a challenge for website builders as they do not natively support such features, and hence, a versioning solution with Git may solve the issue, similar to VersionPress [30].

4. Experiments

4.1 Benchmarking: Partial Sketch-to-Code Direct Generation

We perform benchmarking using multiple open-source and close-source LLMs to evaluate the effectiveness of these models towards the partial sketch-to-code generation task.

We create the test dataset by selectively choosing HTML pages from the WebCode2M training set [18] with less than 2048 tokens. A machine-learning-powered profanity check is then performed to filter harmful and explicit data. Among these web pages, a few have been picked to have some of their components replaced by sketches we manually draw. This is done by taking a full screenshot of the web page and then overlaying the sketch on top of the to-be-replaced components. We also remove the selected component from the HTML code, including all its relevant CSS.

The LLMs are then given both the modified, partial HTML and an image representing the screenshot of the web page merged with the sketch to generate the full HTML code. The response from LLMs is then parsed to extract the HTML and CSS code.

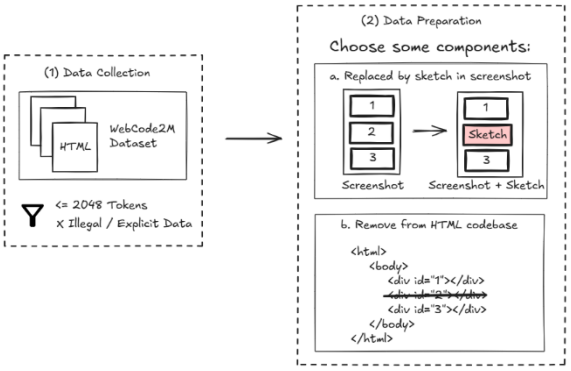

Fig. 1: The pipeline of constructing partial sketch2code dataset.

The generated web pages are compared against the original HTML and CSS code and evaluated according to metrics proposed in Design2Code [19]. In short, these metrics include:

- Block-Match: Evaluate if all visual elements are accurately replicated in the generated version.
- Text Similarity: Similarity between the corresponding text boxes.
- Position Similarity: How closely do the blocks match those in the original?

Additionally, we track the number of generations required to produce valid HTML. If it takes more than 10 attempts, we consider the generation unsuccessful, as shown in the table below.

Models	Block	Text	Position	Success Rate	Avg. # of Gen
Open-source models					
Gemma2: 27b	0.32	0.81	0.56	80%	1
Gemma2:9b	0.11	0.68	0.54	100%	2.47
Llama3.2-vision:11b	0.40	0.91	0.68	100%	1.2
Proprietary models					
GPT-4o	0.95	0.98	0.95	100%	1
Grok-2-vision	0.90	0.96	0.89	100%	1
Gemini-2.0-flash	0.96	0.99	0.95	100%	1

Table 1: The performance of three open-source and three commercial models on the Partial Sketch-to-Code Direct Generation task.

5. Analysis

Overall, we observe that proprietary models perform better than open-source models, especially Gemini-2.0-flash, which outperforms every other vision model. We realise that the open-source models sometimes refuse to perform the task, citing a lack of vision, legal restrictions or other limitations, and as such, have to regenerate until there is a successful attempt. Open-source models also often ignore the sketch part, and they tend to mess up the rest of the design, as reflected in their low position score and low block score.

Acknowledgements

I wish to express my gratitude to Dr Li Fang for her invaluable guidance and support in this project, as well as to Andy and Jasper for their foundational work in their previous studies, all of which significantly shaped the direction of this paper. I am grateful to Fun Of GrowinG (FOGG) for providing access to their Wix-built web pages for the literature review.

References

[1] A. Ko, "How to Create a Website From Scratch in 11 Steps (for Beginners)," Wix Blog. Accessed: Mar. 09, 2025. [Online]. Available: https://www.wix.com/blog/how-to-build-website-from-scratch-guide

[2] Wix, "Your website, your business, your future | Wix.com," wix.com. Accessed: Aug. 31, 2024. [Online]. Available: https://www.wix.com/

[3] "About," WordPress.org. Accessed: Aug. 31, 2024. [Online]. Available: https://wordpress.org/about/

[4] J. Hitchcock, "What Is Shopify and How Does It Work? (2025) - Shopify Singapore," Shopify. Accessed: Jan. 24, 2025. [Online]. Available: https://www.shopify.com/sg/blog/what-is-shopify

[5] Blackboard, "Blackboard Learn - Low-maintenance. Modern. A new kind of LMS. | Anthology." Accessed: Jan. 24, 2025. [Online]. Available: https://www.anthology.com/en-apac/products/teaching-and-learning/learning-effectiveness/blackboard-learn

[6] S. K. Oswal and H. K. Oswal, "Examining the Accessibility of Generative AI Website Builder Tools for Blind and Low Vision Users: 21 Best Practices for Designers and Developers," in 2024 IEEE International Professional Communication Conference (ProComm), Jul. 2024, pp. 121–128. doi: 10.1109/ProComm61427.2024.00030.

[7] Z. B. Palmer and S. K. Oswal, "Constructing Websites with Generative AI Tools: The Accessibility of Their Workflows and Products for Users With Disabilities," Journal of Business and Technical Communication, vol. 39, no. 1, pp. 93–114, Jan. 2025, doi: 10.1177/10506519241280644.

[8] Mamatha SK, Gurukiran AC, Basavaraj, and Meghana, "NO-CODE WEB DEVELOPMENT," IRJMETS.

[9] A. H. S. Ng, "What website builders can and cannot do," 2023, Accessed: Aug. 31, 2024. [Online]. Available: https://hdl.handle.net/10356/171901

[10] J. K. L. Yeo, "What website builders can and cannot do," 2024, Accessed: Aug. 31, 2024. [Online]. Available: https://hdl.handle.net/10356/175248

[11] J. A. Landay and B. A. Myers, "Sketching interfaces: toward more human interface design," Computer, vol. 34, no. 3, pp. 56–64, Mar. 2001, doi: 10.1109/2.910894.

[12] T. Beltramelli, "pix2code: Generating Code from a Graphical User Interface Screenshot," Sep. 19, 2017, arXiv: arXiv:1705.07962. doi: 10.48550/arXiv.1705.07962.

[13] Microsoft Azure, "Turn your whiteboard sketches to working code in seconds with Sketch2Code," Microsoft Azure Blog. Accessed: Mar. 09, 2025. [Online]. Available: https://azure.microsoft.com/en-us/blog/turn-your-whiteboard-sketches-to-working-code-in-seconds-with-sketch2code/

[14] A. Robinson, "Sketch2code: Generating a website from a paper mockup," May 09, 2019, arXiv: arXiv:1905.13750. doi: 10.48550/arXiv.1905.13750.

[15] V. Jain, P. Agrawal, S. Banga, R. Kapoor, and S. Gulyani, "Sketch2Code: Transformation of Sketches to UI in Real-time Using Deep Neural Network," Oct. 20, 2019, arXiv: arXiv:1910.08930. doi: 10.48550/arXiv.1910.08930.

[16] D. Soselia, K. Saifullah, and T. Zhou, "Learning UI-to-Code Reverse Generator Using Visual Critic Without Rendering," May 2023. Accessed: Feb. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/ea62c8802554a203f39d790959b7b75475413034

[17] K. Lee et al., "Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding," Jun. 15, 2023, arXiv: arXiv:2210.03347. doi: 10.48550/arXiv.2210.03347.

[18] Y. Gui et al., "WebCode2M: A Real-World Dataset for Code Generation from Webpage Designs," Feb. 22, 2025. doi: 10.1145/3696410.3714889.

[19] C. Si, Y. Zhang, R. Li, Z. Yang, R. Liu, and D. Yang, "Design2Code: Benchmarking Multimodal Code Generation for Automated Front-End Engineering," Feb. 09, 2025, arXiv: arXiv:2403.03163. doi: 10.48550/arXiv.2403.03163.

[20] R. Li, Y. Zhang, and D. Yang, "Sketch2Code: Evaluating Vision-Language Models for Interactive Web Design Prototyping," Oct. 21, 2024, arXiv: arXiv:2410.16232. doi: 10.48550/arXiv.2410.16232.

[21] Wix, "About Velo." Accessed: Jan. 26, 2025. [Online]. Available: https://dev.wix.com/docs/velo/articles/getting-started/about-velo

[22] Wix, "About Velo by Wix." Accessed: Jan. 26, 2025. [Online]. Available: https://dev.wix.com/docs/develop-websites/articles/getting-started/about-velo-by-wix

[23] "Wix Editor Constantly Crashing - Ask the community," Community Support Forum | Wix Studio. Accessed: Jan. 21, 2025. [Online]. Available:

https://forum.wixstudio.com/t/wix-editor-constantly-crashing/21617

[24] "Freeze or Crashing Editor - Ask the community," Community Support Forum | Wix Studio. Accessed: Jan. 26, 2025. [Online]. Available: https://forum.wixstudio.com/t/freeze-or-crashing-editor/7194

[25] "Wix Editor Crashing? - Ask the community," Community Support Forum | Wix Studio. Accessed: Jan. 26, 2025. [Online]. Available: https://forum.wixstudio.com/t/wix-editor-crashing/36384

[26] "Editor Performance: Best Practices | Help Center | Wix.com." Accessed: Jan. 26, 2025. [Online]. Available: https://support.wix.com/en/article/editor-performance-best-practices

[27] "Velo Wix Bookings Frontend Introduction | Velo." Accessed: Jan. 26, 2025. [Online]. Available: https://dev.wix.com/docs/velo/api-reference/wix-bookings-frontend/introduction

[28] "10Web WordPress AI Builder," 10Web. Accessed: Jan. 27, 2025. [Online]. Available: https://10web.io/wordpress-ai-builder/

[29] "WooCommerce - Open Source ecommerce Platform," WooCommerce. Accessed: Mar. 15, 2025. [Online]. Available: https://woocommerce.com/

[30] versionpress/versionpress. (Jan. 17, 2025). PHP. VersionPress. Accessed: Jan. 27, 2025. [Online]. Available: https://github.com/versionpress/versionpress

Appendix A: Examples of Dataset

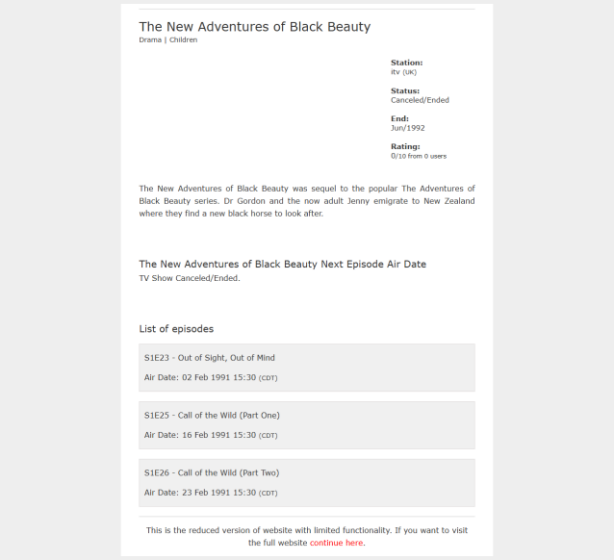

Fig. 2: A screenshot of the original web page fetched from the WebCode2M database.

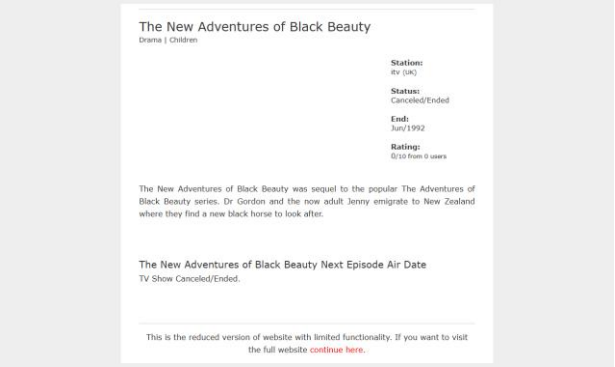

Fig. 3: The screenshot of the web page in Fig. 2 after removing the episode list section from it.

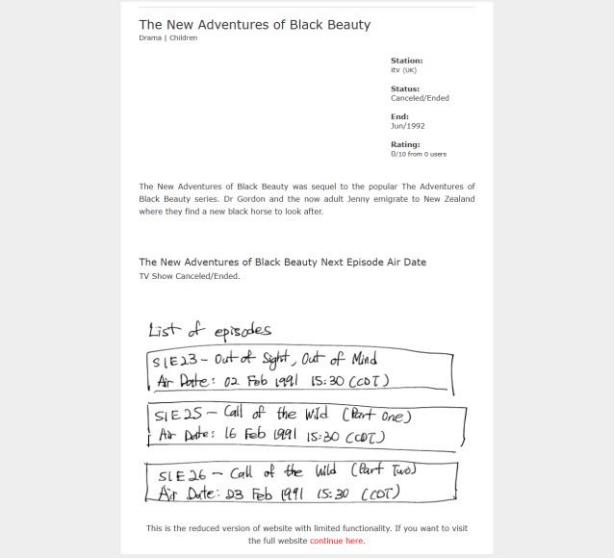

Fig. 4: The modified screenshot with a hand-drawn overlay replacing the episode list section.

Appendix B: Relevant Prompts

System Prompt: ''' You are an expert web developer who specialises in HTML and CSS. A user will provide you with the HTML code of the current webpage, as well as a screenshot of the new webpage design. Note that some components are in sketch format in the screenshot.
Your task is to convert the new design into HTML and CSS code based on the provided screenshot and the existing HTML code. You should modify only the necessary part, leaving the rest of the page unchanged.
Include all CSS code in the HTML file itself.
Do not hallucinate any dependencies on external files.
Pay attention to things like the size and position of all the elements, as well as the overall layout.
If you need to include new images, use "temp.jpg" as the placeholder name. As a reminder, the "temp.jpg" placeholder is very large (1920 x 1080). So make sure to always specify the correct dimensions for the images in your HTML code since, otherwise, the image would likely take up the entire page.
You should aim to make the new webpage as responsive as possible. '''

User Prompt: ''' Here is a screenshot of the new design for the webpage, as well as the existing HTML and CSS code. Please update the HTML and CSS code according to the screenshot. Make sure to maintain the overall layout and design consistency.

\n''' + <<partial_html_code>>

Listing 1: Prompts given to LLMs to generate the HTML and CSS based on the modified screenshot

Appendix C: Screenshots of Generated Web Pages

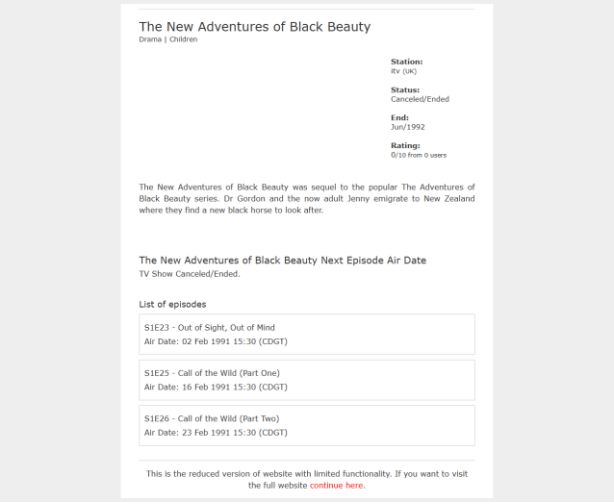

Fig. 5: A screenshot of the web page generated by GPT-4o based on the partial HTML code and Fig. 4.

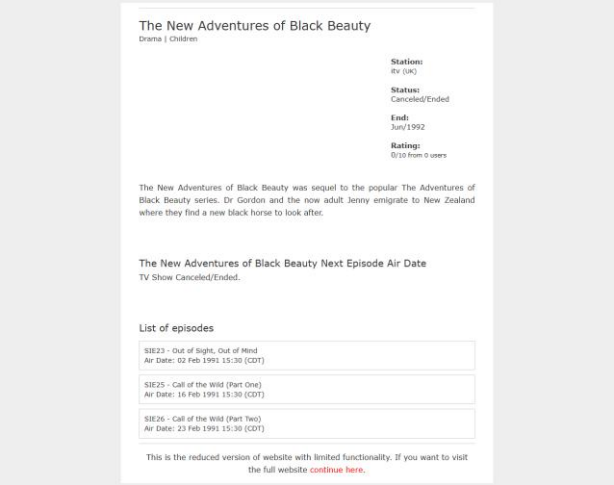

Fig. 6: A screenshot of the web page generated by Gemini-2.0-Flash based on the partial HTML code and Fig. 4.

Fig. 7: A screenshot of the web page generated by Llama3.2-vision:11b based on the partial HTML code and Fig. 4.

Appendix D: Screenshots of Proposed Proof-of-Concept Website Builder

We build a proof-of-concept using Grapes.js, an open-source, lightweight static website builder. We choose it for its simplicity and its ability to natively map drag-and-drop components to raw HTML code and vice versa.

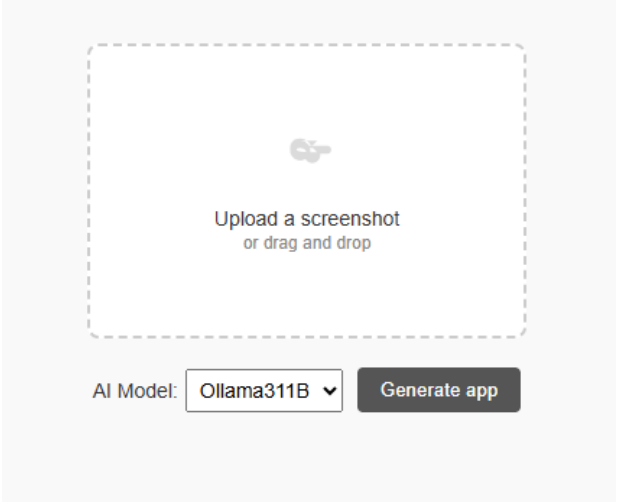

Fig. 8: A screenshot of the proof-of-concept website builder, asking the user to upload a screenshot.

Fig. 9: A screenshot of the proof-of-concept website builder using Grapes.js, showcasing its drag-and-drop editor interface.