International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

A No Code Website Builder Approach

Shreyansh Jagtap[1], Dhruv Yadav[2], Varun Varma[3], Abhishek Gore[4], Dr. Dhananjay Dakhane[5]

[1,2,3,4] B.TECH Student, DY Patil University, RAIT, Nerul, Maharashtra, India
[5]Professor, DY Patil University, RAIT, Nerul, Maharashtra, Indi

---

Abstract - The development and upkeep of websites have become essential components of both personal and business undertakings in today's digital landscape. But the complexity and technicality of website builders has traditionally presented a significant obstacle, restricting accessibility for those without specialist technological skills. In order to solve this pressing problem, our paper presents a website builder approach that is incredibly user-friendly and doesn't require any technical knowledge. This application gives schools a rare opportunity to efficiently teach website design principles while also empowering students. We analyze the state of website builders today through the prism of this paper, pinpoint the problems resulting from technological complexity, and suggest a game-changing fix.

Key Words: optics, photonics, light, lasers, templates, journals

1.INTRODUCTION

Websites are the cornerstone of modern digital presence, serving as the virtual storefronts, portfolios, and hubs for businesses, creatives, and communities alike. Their significance lies in their ability to transcend geographical boundaries, enabling entities to connect with audiences globally. By establishing an online presence, businesses can extend their reach far beyond local markets, tapping into a vast pool of potential customers accessible 24/7.

Moreover, websites are indispensable tools for marketing and branding. They provide platforms for storytelling, brand positioning, and product showcasing. With features such as search engine optimization (SEO), content marketing, and social media integration, websites serve as central hubs for various digital marketing strategies, driving traffic and engagement.

Customer engagement is another critical aspect facilitated by websites. Through features like contact forms, live chat, and social media integration, businesses can establish direct communication channels with their audience. Feedback mechanisms and customer support further enhance engagement, fostering relationships and loyalty.

For e-commerce ventures, websites are fundamental to sales and revenue generation. They serve as transactional platforms, facilitating purchases, processing payments, and managing inventory. The scalability of websites allows businesses to adapt and expand their online operations as they grow, accommodating increased traffic and functionality seamlessly.

Furthermore, websites offer valuable insights through analytics tools, providing data on visitor behavior, preferences, and demographics. This data-driven approach informs decision-making and optimization strategies, empowering businesses to continually refine their online presence for better performance.

In essence, websites are indispensable assets in the digital age, serving as essential tools for businesses, organizations, and individuals to establish their online footprint, engage with audiences, and achieve their objectives effectively.

Despite the pivotal role websites play in today's digital landscape, the process of creating one often presents a significant barrier for many individuals and businesses due to the technical knowledge required. Traditionally, building a website involved proficiency in coding languages such as HTML, CSS, and JavaScript, as well as familiarity with web development frameworks and tools. This technical complexity could be daunting for those without a background in programming or web development, posing a considerable challenge to anyone looking to establish an online presence.

This paper we aim to provide an overview on subsequent technologies, their problems and provide an approach towards making ones own website builder. Today one can make their own website builder through the use of strong templating capabilities of HTML,CSS and Javascript. Through this papper one might find a way to make their builder for others to use

2. Problem Statement

In today's digital age, establishing a compelling online presence is essential for individuals and businesses alike to thrive in a competitive landscape. However, the process of creating a website often presents significant challenges, particularly for those without technical expertise. Traditional website development requires proficiency in coding languages such as HTML, CSS, and JavaScript, posing a formidable barrier for individuals and small businesses lacking in-house development resources. Furthermore, staying abreast of evolving web technologies and standards adds another layer of complexity, requiring continuous learning and investment of time and resources.

Even with the availability of content management systems (CMS) like WordPress or Joomla, navigating the intricacies of these platforms and customizing websites beyond basic templates can be daunting for non-technical users. While CMS platforms aim to simplify website development with drag-and-drop interfaces and pre-designed templates, achieving a truly unique and tailored online presence often necessitates technical skills or the assistance of professional developers. This lack of accessibility and flexibility in website creation inhibits

individuals and businesses from fully realizing their online ambitions and reaching their target audiences effectively.

Furthermore, the cost associated with hiring professional developers or purchasing premium website templates can be prohibitive for budget-conscious users. As a result, many individuals and small businesses are left with suboptimal online presences that do not accurately reflect their brand identities or effectively engage their audiences.

In summary, the current landscape of website development presents a significant barrier for individuals and businesses looking to establish a compelling online presence. Addressing these challenges requires innovative solutions that simplify the website creation process, democratize access to web development tools and resources, and empower users of all backgrounds to create impactful websites that reflect their unique identities and objectives.

Therefore taking all the mentioned factors into consideration, the primary objective of this research is to address the following key questions:

Objectives:

1. To understand the current solutions, present for the no code web building, analyse them.
2. To evaluate the impact of the technical requirement for the entry webpage.
3. To provide options to people who do not want to manually sit through the web page creation process.

By achieving these objectives, this study aims to provide a comprehensive insights into the technological environment of the no code options regarding web developments , thereby building on it and create an approach to develop a more approachable and beginner friendly no code method to design a website.

3. LITERATURE REVIEW

1. WordPress

WordPress is a web content management system, usually referred to as WP or WordPress.org. Although it was first developed as a tool for blogging, it has since developed to enable the publication of other types of web content, such as more conventional websites, online stores, membership sites, email lists, and forums. The software can be downloaded for free and is open-source.

2. Weebly

Weebly is a web creation and hosting firm based in San Francisco, California, USA. It is a part of Block, Inc. The company's main goal was to develop software that simplified the process of creating personal web pages for individuals.

3. Google Slides

sing a drag-and-drop widget-based site builder, Weebly's website creation runs in a web browser. Although the service limits the size of each file, storage is limitless. Each website is instantly converted to a mobile format, blogging and e-commerce are supported, and storage is available with some file size limitations. Different domain endings are available to users, such as weebly.com,.com,.net,.org,.co,.info, or.us.

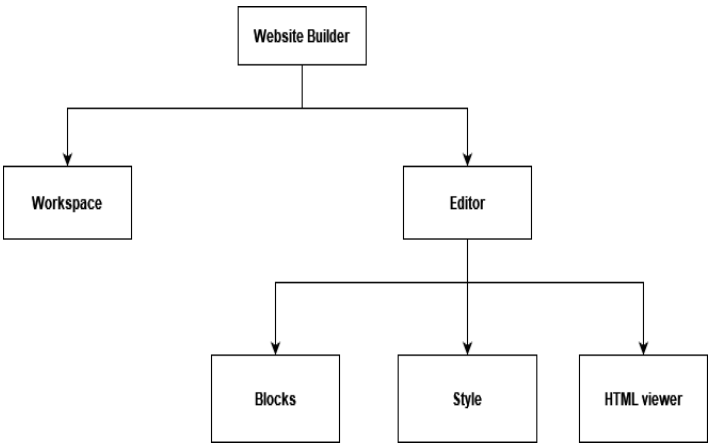
.

4. METHODOLOGY

The paper is a 'No Code Website Builder'. Websites today can be made through various methods using various coding languages. Our proposal is to provide a method to build websites without the use of any coding languages. This is possible though the power full functionality of HTML and JavaScript. The website can be built without any use of manual coding. Although it is powered by coding languages in the background. This results in the user being able to design websites without any prior technical knowledge. This user can then design websites with following common design practices.

There are 'blocks' provided to the user which are then manipulated by the user to create the web page. These blocks are the components which are used to build web document, These components are pre built and are only needed to be drag and drop. They are modular HTML code. They can be used in any web document. In our paper these blocks can be dynamically changed and these changes are viewed in the real time in the editor. The editor consists of blocks and the ability to change the styles and layout of the blocks dynamically. It is through the editor the user can edit the web document and its blocks without any code.

There are two components to the website builder. One is the workspace that is actually worked upon. The other is the editor through which the user can manipulate the workspace.



The workspace is but a blank document which is what the user want to edit. In the workspace the user can add blocks provided in the editor by drag and drop action. The user can also move these added blocks by the controller. The user can also remove these blocks from the workspace itself.

The ‘Editor’ has three key sections. The 'Style', 'Layers' and 'Blocks' are the present in the editor. The 'Blocks' module contains various blocks. The ‘Layers’ module contains the data of added blocks. The 'Styles’ contains the styles which is used to modify the blocks. The ‘Workspace’ module has the 'Editor' and 'Menu'. In 'Workspace' the blocks can be re-positioned as

well as be removed. In 'Menu' the HTML and CSS data can be exported.

The ‘Editor’ and ‘Workspace’ are two different HTML documents. Separating the ‘Editor’ and ‘Workspace’ into distinct HTML documents provides numerous advantages for system. This division adheres to the principle of separation of concerns, allowing each component to focus on specific functionalities. The ‘Editor’ can concentrate on user-friendly content creation, while the ‘Workspace’ handles layout and organization. This modular approach promotes code modularity and reusability, facilitating easier maintenance and updates across the paper.This separation also contributes to improved performance by reducing rendering and processing overhead. Additionally, the modular architecture streamlines debugging and testing processes, leading to more reliable software.

Linking the ‘Editor’ to the ‘Workspace’ and allowing access to the ‘Workspace’'s inner HTML simplifies the export process of HTML documents. This enables the extraction of content and styling information for seamless integration with external systems. Furthermore, the separation of makes it easier to export and import content between papers or environments. Thus, separating the ‘Editor’ and ‘Workspace’ into distinct HTML documents enhances manageability, promotes collaboration, improves performance, and streamlines the export process, resulting in more scalable and efficient web-based applications.

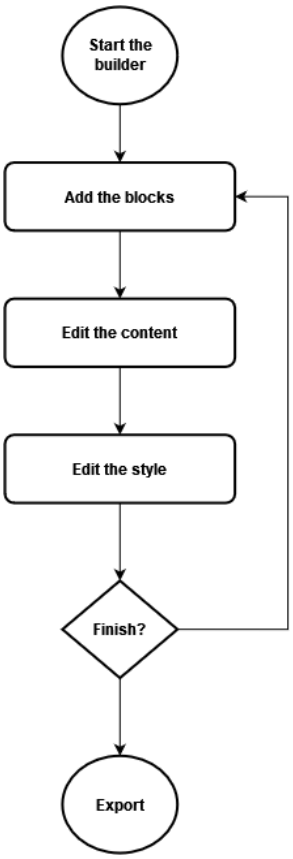
The ‘Layers’ and ‘Styles’ are where the CSS of the components in the workspace are to be edited. The ‘Layers’ module is used to manipulate the order or placement and the visibility of the components in the ‘Workspace’. This change in placement is not a change in position but is change in the placement on the HTML node tree. This helps to dictate the hierarchy of components in the ‘Workspace’.

On the other hand, the ‘Styles’ module serves as the creative engine for customizing the visual presentation of blocks within the ‘Workspace’. Here, users have the flexibility to modify various design attributes, including position, color, alignment, font, and more. By leveraging the ‘Styles’ module, users can infuse their papers with unique aesthetics and branding elements, transforming their vision into reality.

While the ‘Layers’ module focuses on the structural organization of components, the ‘Styles’ module is solely dedicated to visual styling. Unlike the ‘Layers’ module, which manipulates the HTML node tree, the ‘Styles’ module operates exclusively on the CSS of the blocks. This separation of concerns ensures that changes made within the ‘Styles’ module do not affect the underlying structure of the webpage but instead influence the presentation and appearance of individual components.

‘Blocks’ serve as the fundamental building blocks within the ‘Workspace’, constituting the core elements that compose the structure and content of a web page. These components are prebuilt, designed to offer users a diverse array of functionalities and design options to streamline the website creation process. Primarily, blocks encapsulate predefined HTML code, encapsulating common elements such as headers, paragraphs, images, buttons, forms, and more. These components are curated to cater to a wide range of use cases and design requirements, ensuring versatility and flexibility in content creation.

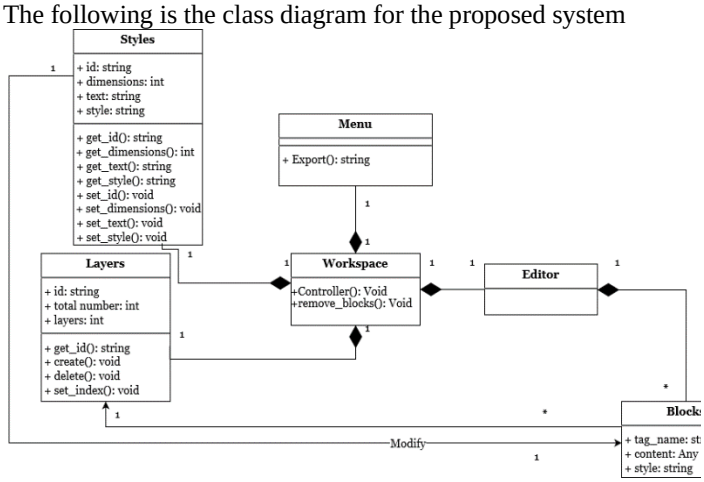
When users select a block within the workspace, it transitions into an active state, signaling that it's ready for customization. In this active state, the selected block's data is transmitted to the ‘Editor’, where it is prominently displayed within the ‘Styles’ tab. This integration streamlines the design process, offering users immediate access to the properties and attributes of the selected block.



The following is the general flowchart on how a the process of building a web document work using the website builder. After the start of the builder the blocks are added after which user can edit the style of the blocks. If the user feels sufficient then they can either export the HTML and CSS through the CSS viewer or they can continue building their document by adding more blocks.

5. SYSTEM DESIGN

The following is the class diagram for the proposed system



The 'Editor' has contained modules related to it. It is the central component of the system, consisting of interconnected modules crucial for content creation and customization. The 'Style', 'Layers' and 'Blocks' are the modules present in the editor. The 'Blocks' module contains various blocks. The "Blocks" module offers a diverse selection of elements or components that users can incorporate into their papers. These blocks serve as the foundational building blocks for constructing content within the editor. The 'Layers' module contains the list and order of the added blocks. It stores the data of added blocks, organizing them into a hierarchical structure for efficient management and manipulation. It enables users to access, rearrange, and modify their content seamlessly, providing clarity and coherence in the editing process. The 'Styles' modules contains the styles which is used to modify the blocks. Users can customize the visual presentation of components according to their preferences, enhancing the aesthetics and usability of their papers.

The 'Workspace' module has the 'Editor' and 'Menu'. In 'Workspace' the blocks can be re-positioned as well as be removed. In 'Menu' the HTML and CSS data can be exported.
The "Workspace" module serves as the cohesive framework that integrates the various functionalities of the editor. The "Editor" and "Menu" components, are in a compositional relationship with the Workspace. While the Editor and Menu can exist independently, the Workspace binds them together, providing a unified platform for content creation and management. Within the Workspace, users can reposition or remove blocks, offering flexibility in content arrangement and composition. Additionally, the Menu allows users to export HTML and CSS data, facilitating seamless transitions between the editor and external platforms or environments.

The Editor serves as the aggregate for the Styles, Layers, and Blocks modules. Without the Editor, these modules are non-functional within the system, highlighting the Editor's central role in facilitating content creation and customization. Both the Layers and Styles modules contain crucial data for the system. The Layers module stores information about the blocks used, including their IDs, and allows users to modify block positions. Meanwhile, the Styles module houses the CSS data used to style components, empowering users to customize the appearance of their content according to their preferences.

3. CONCLUSIONS

The paper successfully achieves its primary objective of provide a simple approach towards a no code website building. This approach ensures accessibility for users of varying skill levels, from beginners seeking to learn the basics of web development to experienced users looking for a streamlined solution for static site creation.

At its current iteration, the paper demonstrates considerable potential for further enhancement and expansion. This methos would only apply for static website.One area for improvement lies in the editor's functionality, where additional features could be introduced to facilitate the creation of more complex layouts and designs.

In addition to these enhancements, future iterations of the paper could explore opportunities for expanding its feature set, such as incorporating support for dynamic content, enabling e-commerce functionality, or integrating third-party services and APIs. These enhancements would further enrich the user experience and broaden the paper's utility, positioning it as a versatile and comprehensive solution for website creation and management. Overall, the paper successfully fulfils its initial objectives, there is some room for growth and improvement.

ACKNOWLEDGEMENT

REFERENCES

4. Alessandro Ortis, Michele Picardi, Alessandro Marrella, Fabio Di Lauro , The impact of no-code on digital product development by arXiv, 2023.

5. Amr Mohamed Abdel-Hamid, Abdel-Rahman Hisham Mohamed, Hala Abdel-Wahab Abdel-Fattah, The Scientific Comparison between Web-Based Site and Web-Builder (Open Source) Paper: Functionalities, Usability, Design and Security by ResearchGate , 2022.

6. Matthias Neubert, Manuel Leuthold, Daniel Schall, Alexander M¨adche, No-Code Development Platforms: Breaking the Boundaries Between IT and Business Experts by ResearchGates, 2022.

7. Zhaohang Yan,The Impacts of Low/No-Code Development on Digital Transformation and Software Development, 2022.

8. Sarah Gibbons and Andrew Keen, The Impact of No-Code Website Builders on Small Business Web Design and Development, 2022.

9. Mark Guzdial and Eric K. Roberts, The Future of No-Code Website Builders: A Study of Trends and Innovations, 2021.

10. John C. Thomas and Robert W. Baird, A Comparative Analysis of No-Code Website Builders: Features, Pricing, and Use Cases, 2020.