International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2(1) (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Pape

AI WEBSITE GENERATOR

[1]A Akhila, [2] S Khaja Pasha,[3] S Abdul Rahman, [4] P Keerthi Sri,[5] P Sneha Reddy
[1]AssistantProfessor, [2345]Students
Department of AIML
Siddhartha Institute of Technology & Sciences, Narapally
akhilaasaara@siddhartha.org.in, 24tq1a66f6@siddhartha.co.in, 24tq1a66g1@siddhartha.co.in, 24tq1a66e8@siddhartha.co.in, 24tq1a66h7@siddhartha.co.in

Abstract

In the modern digital era, the demand for responsive, user-friendly, and visually appealing websites has significantly increased across various domains. Developing such websites typically requires proficiency in multiple frontend technologies, including HTML, CSS, and JavaScript. This multi-technology requirement presents a considerable barrier for beginners and non-technical users, while even experienced developers often spend substantial time handling repetitive coding tasks, layout structuring, and design implementation. Consequently, there is a growing need for an intelligent and automated solution that simplifies the web development process and enhances efficiency.

This project introduces an AI-based Website Generation System that leverages the capabilities of Generative Artificial Intelligence to automate the creation of complete websites from simple textual descriptions. The system allows users to input natural language prompts specifying their website requirements, such as layout, features, and design preferences. Based on this input, the system generates well-structured, production-ready frontend code, including HTML for content structure, CSS for styling, and JavaScript for interactivity. The generated websites adhere to modern web design standards, incorporating responsive layouts, visually appealing gradient themes, and basic animations to enhance user experience.

The system is developed using Streamlit as the user interface framework, providing an interactive and intuitive platform for users to input their requirements and view outputs. It integrates with a locally hosted Large Language Model (LLM) via Ollama, ensuring efficient and secure code generation without reliance on external cloud services. Additionally, the system includes a real-time preview feature that enables users to instantly visualize the generated website, facilitating quick feedback and iterative improvements.

The primary objective of this project is to reduce the time and effort involved in website development, improve developer productivity, and democratize access to web creation tools. By minimizing the technical expertise required, the system empowers a broader range of users—including students, entrepreneurs, and small business owners—to build functional and aesthetically pleasing websites with ease.

I. Introduction

In recent years, Generative Artificial Intelligence (GenAI) has emerged as a powerful technology capable of creating content such as text, images, and code with minimal human intervention. One of its most impactful applications is in the field of software and web development, where it can automate complex and time-consuming tasks.

Website development, in particular, requires knowledge of multiple technologies such as HTML, CSS, JavaScript, and modern UI design principles, which can be challenging for beginners and non-technical users.

Traditional website development involves writing code manually, designing layouts, and ensuring responsiveness across devices. This process is not only time-consuming but also requires a certain level of expertise. Even experienced developers spend a significant amount of time on repetitive tasks such as structuring layouts, styling components, and adding basic interactivity. Therefore, there is a growing need for intelligent systems that can simplify and accelerate the website creation process.

This project introduces an AI-based Website Generation System that leverages Generative AI to automatically generate complete websites from simple user prompts. The system allows users to describe their website idea in natural language, and based on this input, it generates structured and production-ready code including HTML, CSS, and JavaScript. The generated websites follow modern design standards such as responsive layouts, gradient-based themes, animations, and interactive elements.

The system is developed using Streamlit, which provides an intuitive and user-friendly interface for interacting with the application. For the AI component, a locally hosted Large Language Model (LLM) is used through Ollama. This ensures faster response times, data privacy, and offline capability compared to cloud-based solutions. The system also includes a live preview feature that allows users to instantly visualize the generated website, making the process more interactive and efficient.

By integrating Generative AI into web development, this project aims to reduce the technical barrier for website creation and improve productivity. It is especially useful for students, beginners, and developers who want to quickly prototype or generate website designs without writing extensive code. Overall, this project demonstrates how AI can be effectively used to automate frontend development and transform the way websites are built.

II. Literature Survey

The rapid advancement of Artificial Intelligence, especially in the field of Generative AI, has significantly influenced modern software development. Generative AI enables systems to create human-like content such as text, images, and code, making it highly useful in automating complex tasks. In recent years, researchers and developers have focused on leveraging Large Language Models (LLMs) for code generation and application development. This section reviews existing technologies, tools, and research related to AI-based website generation systems.

GENERATIVE AI AND LARGE LANGUAGE MODELS

Generative AI refers to a class of artificial intelligence models that can generate new content based on learned patterns from large datasets. One of the most important breakthroughs in this domain is the development of Transformer-based architectures, which form the backbone of modern Large Language Models (LLMs). These models are trained on vast amounts of textual and programming data, enabling them to understand context and generate meaningful outputs.

Popular LLMs such as GPT and LLaMA have demonstrated strong capabilities in generating structured code from natural language prompts. These models can interpret user input and convert it into syntactically correct and logically consistent programming code. This capability has led to the development of intelligent systems that assist in coding, debugging, and application generation.

The use of LLMs in code generation has reduced the complexity of software development by automating repetitive tasks. However, challenges such as occasional inaccuracies, lack of optimization, and dependency on proper prompt design still exist. Continuous improvements in model training and prompt engineering are helping to enhance the reliability of these systems.

AI-BASED CODE GENERATION TOOLS

Several tools have been developed that utilize Generative AI for code generation. For example, AI-powered assistants can suggest code snippets, complete functions, and even generate entire applications based on user requirements. These tools significantly improve developer productivity and reduce the time required for coding.
Most existing solutions are cloud-based, requiring internet connectivity and raising concerns related to data privacy and security. Additionally, these tools may have limitations in customization and flexibility. Despite these challenges, AI-based code generation tools have become an important part of modern development workflows.

WEBSITE DEVELOPMENT APPROACHES

Traditional website development involves manual coding using HTML, CSS, and JavaScript. Developers need to design layouts, implement responsiveness, and ensure compatibility across different devices and browsers. This process requires technical expertise and can be time-consuming.
To simplify this process, website builders such as Wix and Squarespace provide drag-and-drop interfaces that allow users to create websites without coding. However, these platforms still require manual interaction and do not fully automate the design and development process based on user intent.
Recently, AI-powered website builders have emerged, which generate layouts and content based on user inputs. While these systems offer automation, they often lack flexibility and customization compared to manually coded websites.

LOCAL MODEL DEPLOYMENT USING OLLAMA

One of the key advancements in AI system deployment is the ability to run models locally. Tools such as Ollama allow developers to run Large Language Models on their own machines without relying on cloud services. This provides several advantages, including improved data privacy, reduced latency, and offline functionality.
Local deployment is particularly useful for applications where sensitive data is involved or where internet access is limited. It also allows developers to have better control over model behavior and performance. In this project, Ollama is used to integrate a local LLM for generating website code efficiently.

USER INTERFACE FRAMEWORKS – STREAMLIT

Streamlit is a popular Python-based framework used for building interactive web applications. It simplifies the development process by allowing developers to create user interfaces with minimal code. Features such as text input, buttons, and real-time output display make it ideal for AI-based applications.
In this project, Streamlit is used to build the frontend interface where users can enter their website ideas and view the generated output. It also supports embedding HTML content, enabling the system to display a live preview of the generated website.

LIMITATIONS OF EXISTING SYSTEMS

Despite the advancements in Generative AI and code generation tools, there are still several limitations. Generated code may sometimes contain errors or may not fully meet user expectations. Additionally, understanding complex user requirements accurately remains a challenge for AI models.

Many existing systems are dependent on cloud-based APIs, which may introduce latency and raise privacy concerns. Furthermore, customization options in automated systems are often limited compared to traditional development methods.

This project addresses these limitations by integrating a locally hosted AI model with a simple and interactive interface to generate complete websites. By combining Generative AI, local deployment, and user-friendly design, the proposed system aims to provide an efficient and accessible solution for automated website generation. Ultimately, this research highlights a critical shift toward decentralized AI development, prioritizing data sovereignty and developer autonomy. By bridging the gap between high-level automation and local execution.

III. System Analysis

The AI Website Generator system is designed to automate the process of website creation using Artificial Intelligence technologies. The system aims to simplify web development by enabling users to generate complete websites through simple text prompts and user requirements without requiring advanced programming knowledge. The proposed system integrates technologies such as Artificial Intelligence (AI), Natural Language Processing (NLP), Machine Learning (ML), and Generative AI to automatically create website layouts, content, design structures, and frontend code. Users can provide business details, preferred design styles, features, and functionality requirements through a conversational or form-based interface. The system analyzes user input and generates responsive website templates, navigation structures, images, and content dynamically. It reduces the complexity of manual website development and accelerates the design and deployment process. The architecture includes modules such as user interaction, AI processing, template generation, content generation, code generation, and deployment management. Machine Learning models continuously improve website recommendations and design quality based on user preferences and historical data. The system also supports customization, responsive design, and automated optimization features. Overall, the AI Website Generator improves development speed, reduces costs, and enables non-technical users to build professional websites efficiently.

Existing System

Traditional website development systems mainly rely on manual coding, web development frameworks, and professional designers for creating websites. In the existing approach, developers use programming languages such as HTML, CSS, JavaScript, PHP, and frontend frameworks to build websites from scratch. Website development often requires significant technical knowledge, design expertise, and development time. Businesses and individuals who lack coding skills depend heavily on web developers or website development companies, increasing project costs and delivery time. Existing website builders provide drag-and-drop interfaces and predefined templates, but they still require manual customization and configuration. Traditional systems also involve repetitive tasks such as layout design, content writing, responsive design adjustments, and SEO optimization. Many existing website

generation tools lack intelligent automation and cannot fully understand user requirements dynamically. Static templates provided by current systems offer limited personalization and flexibility. Manual website development also increases the chances of coding errors, design inconsistencies, and maintenance complexity. These limitations make traditional website development slower, more expensive, and less accessible for non-technical users.

Disadvantages of Existing System

- Requires technical coding knowledge and web development skills.
- Website development process is time-consuming.
- High dependency on professional developers and designers.
- Increased development and maintenance costs.
- Limited automation in website generation tools.
- Existing templates offer limited personalization.
- Manual content creation and design customization required.
- Higher possibility of coding and design errors.

Proposed System

The proposed AI Website Generator is an intelligent platform developed to automate website creation using Artificial Intelligence, Natural Language Processing, and Generative AI technologies. The system allows users to generate websites by simply describing their business, requirements, preferred design style, and functionality through text prompts or conversational interaction. The AI engine analyzes user input and automatically generates responsive website layouts, navigation structures, UI components, images, and dynamic content. Unlike traditional website builders, the proposed system uses Generative AI models to create customized and human-like website designs tailored to user preferences. The system also generates frontend code such as HTML, CSS, JavaScript, and framework-based components automatically. NLP techniques help the system understand user intent and convert natural language instructions into website features and design elements. Machine Learning algorithms continuously improve design recommendations and content quality based on user interactions and historical patterns. The system supports responsive design, SEO optimization, automated deployment, and easy customization features. A modular architecture ensures scalability, flexibility, and future integration with cloud services and advanced AI models. The proposed system reduces development effort, minimizes coding complexity, and enables users to create professional websites quickly and efficiently.

Advantages of Proposed System

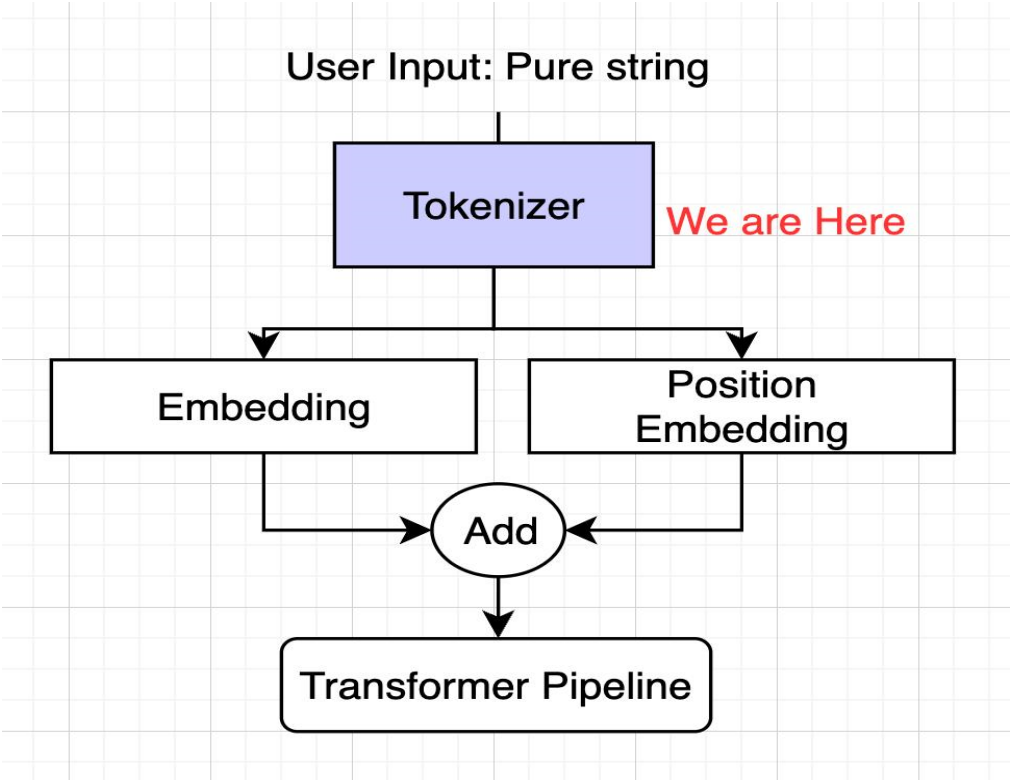
- Automates website creation using Artificial Intelligence.
- Reduces development time significantly.
- Allows non-technical users to create websites easily.
- Generates responsive and modern website designs automatically.
- Provides personalized website layouts and content.
- Reduces dependency on professional developers.
- Supports automated frontend code generation.
- Improves efficiency and reduces operational costs.

- Enables faster website deployment and updates.

IV. Methodology

The development of the AI Website Generator follows a structured methodology involving requirement analysis, AI processing, content generation, code generation, and website deployment. Initially, users provide website requirements such as business type, preferred theme, design style, features, color schemes, and content details through a conversational interface or input forms. The Natural Language Processing module analyzes user input and extracts meaningful information such as website purpose, layout preferences, and functionality requirements. The AI Design Engine processes this information and generates suitable website templates, UI layouts, and navigation structures dynamically. Generative AI models create website content including headings, descriptions, images, and sections based on user requirements. The Code Generation Module automatically generates frontend code using HTML, CSS, JavaScript, and framework-based components to build responsive web pages. Machine Learning algorithms optimize website recommendations and improve generation accuracy using previous design patterns and user feedback. The system also performs automated responsive design adjustments and SEO optimization for better usability and search engine visibility. Generated websites are previewed for users before final deployment. Testing and evaluation are conducted to ensure design quality, responsiveness, functionality, and performance. This methodology ensures efficient automation, intelligent website generation, and user-friendly website development.

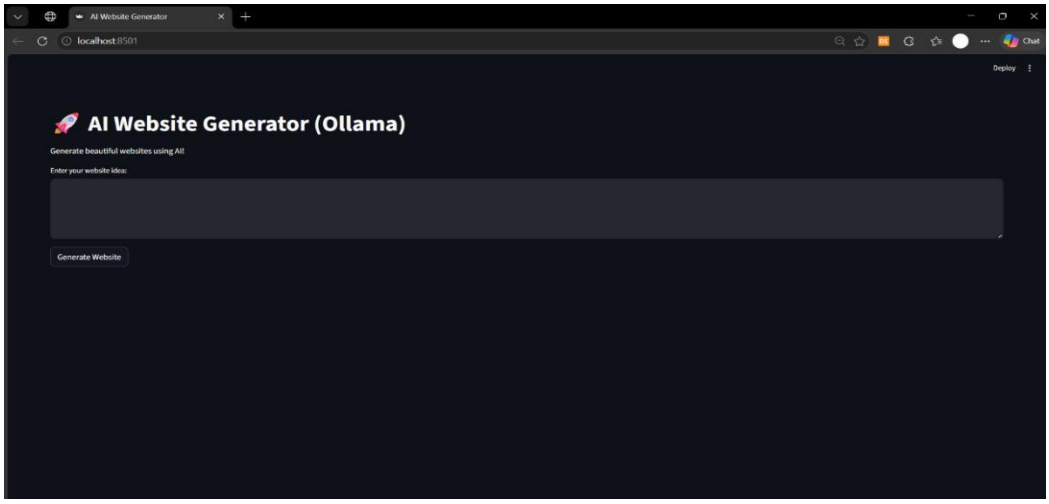
System Architecture

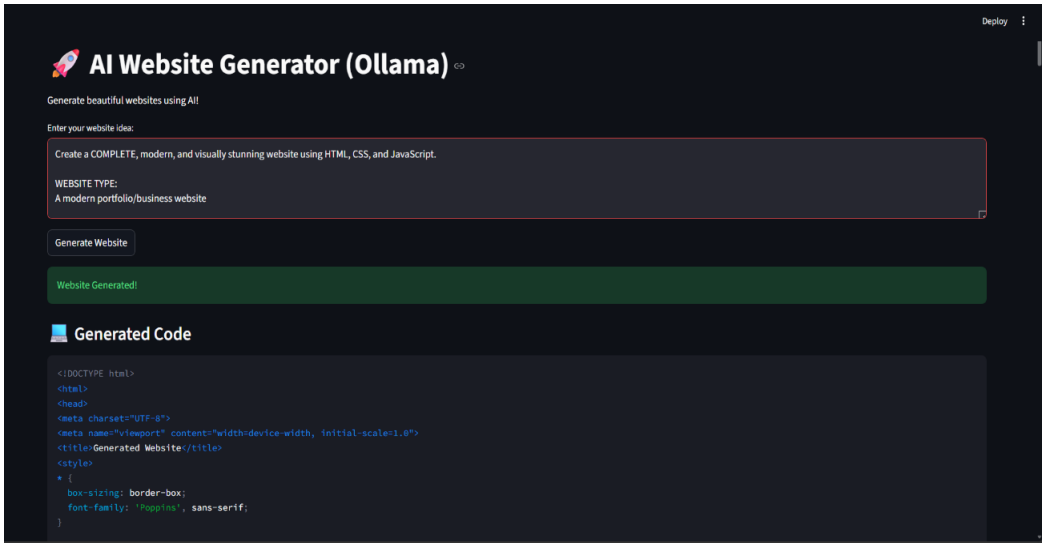


The system architecture of the AI Website Generator consists of multiple

interconnected modules that work together to automate website creation and deployment. The process begins with the User Interaction Module, where users provide website requirements through text prompts, forms, or conversational interfaces. The input data is passed to the Natural Language Processing (NLP) module, which preprocesses text, identifies user intent, extracts keywords, and understands design requirements. The processed information is then forwarded to the AI Design and Recommendation Engine, which generates website layouts, UI structures, navigation menus, and style suggestions dynamically. The Content Generation Module uses Generative AI models to create website content such as titles, descriptions, service sections, blogs, and call-to-action elements automatically. The Code Generation Module converts the generated designs into frontend code using HTML, CSS, JavaScript, and modern web frameworks. A Template and Asset Management System stores reusable templates, UI components, icons, and design resources for efficient website generation. Machine Learning models continuously improve design recommendations and personalization based on user preferences and historical data. The Preview and Customization Module allows users to modify generated websites before deployment. The Deployment Module publishes websites to cloud hosting environments or web servers automatically. Security and Database Management modules ensure secure data storage, project management, and efficient system performance.

V. Result and Output





VI. Conclusion

The AI-based Website Generation System successfully demonstrates the application of Generative Artificial Intelligence in automating frontend web development. The system allows users to generate complete and responsive websites using simple natural language prompts, eliminating the need for extensive coding knowledge and design expertise.

Throughout the project, the system was designed and implemented using Streamlit for the user interface and a locally hosted Large Language Model through Ollama for code generation. The integration of these technologies enabled efficient processing of user input and generation of structured HTML, CSS, and JavaScript code. The inclusion of a live preview feature further enhanced the usability of the system by allowing users to visualize the generated output instantly.

The results obtained from testing and analysis indicate that the system is capable of producing accurate, well-structured, and visually appealing websites. It performs effectively for a wide range of user inputs and significantly reduces the time and

effort required for website development. The use of prompt engineering played a key role in guiding the AI model to generate high-quality outputs.

Although the system has certain limitations, such as dependency on input quality and lack of backend functionality, it still provides a strong foundation for automated website generation. The project highlights the potential of Generative AI in transforming traditional development processes and improving productivity.

In conclusion, the proposed system achieves its objectives by providing a simple, efficient, and intelligent solution for website generation. It serves as a valuable tool for students, beginners, and developers, and opens up opportunities for further advancements in AI-driven software development.

References

[1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.

[2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.

[3] Sv satyakrishna, shirisha rangu ,bhargavi nalacheruve.(2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm

[4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.

[5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.

[6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.

[7] R. D. Kumar, V. N. S.Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.

[8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.

[9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.

[10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.

[11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.