A Project Report

On

AI WEBSITE BUILDER

Submitted in the partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Electronics & Communication Engineering

by

Akarsh Mishra (2023UGEC050)

Rahul Sharma (2023UGEC115)

Harsh Sundi (2022UGEC113)

Under the supervision of

Dr. Munendra Singh
(Assistant Professor)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR

JHARKHAND-831014, INDIA

April 2023

National Institute of Technology Jamshedpur
Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the B.Tech Project entitled, **“AI WEBSITE BUILDER”** Submitted by **“Akarsh Mishra , Rahul Sharma and Harsh Sundi”,** a B.Tech students of the ECE department of NIT, Jamshedpur, India is a record of bonafide project work carried out by him under our supervision. This project is submitted in partial fulfillment of the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering**. The work incorporated in this project has not been submitted to any other university or Institute for the award or any degree or diploma to the best of our knowledge.

Project Supervisor

The viva-voce exam has been held on …29/04/2025... in offline mode.

Faculty Advisor   Head of Department

CANDIDATE’S DECLARATION

I hereby declare that the presented report is as per my project titled **“AI WEBSITE BUILDER”** submitted by **“**Akarsh Mishra , Rahul Sharma and Harsh Sundi**”** and is uniquely prepared in the continuation of my project work at NIT Jamshedpur starting from 03 Jan 2023 to 24 April 2023 under the supervision of Dr. Munendra Singh, Assistant Professor, NIT Jamshedpur. This report is submitted in partial fulfillment of the credits for the degree of Bachelor of Technology in Electronics and Communication Engineering, NIT Jamshedpur during the academic semester January 2025 to May 2025. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Rahul Sharma (2023UGEC115)
Akarsh Mishra (2023UGEC050)
Harsh Sundi (2022UGEC113)

ACKNOWLEDGEMENT

It gives me immense pleasure to pay my sincerest gratitude to my Project Supervisor, Dr. Munendra Singh for their encouragement, support, and constructive criticism. I sincerely thank them for solving all my doubts with patience and persistence throughout the entire duration of my project. He helped me wholeheartedly with full cooperation during my research and also in reaching an efficient solution to my cause.

I am also immensely grateful to the Head of Department, ECE, NIT Jamshedpur, Dr. Dilip kumar for motivating me to present a project on the topic “AI WEBSITE BUILDER**”,** for the partial fulfillment of the requirements for the award of my Bachelor of Technology (Hons.) degree. I also extend my heartfelt gratitude to all the Evaluation committee members for the evaluation of this project and for providing constructive criticism on the overall project.

I also want to thank all my professors who have directly or indirectly helped me to complete this project. It is because of their contribution that I could come up with a decent solution for the project.

I am indebted to our parents, who have always stood beside me in every phase of my life. Whatever I am today and whatever I will achieve in the future I owe it to my parents.

Nothing would have been possible without the consistent affection and support from my friends, family, and well-wishers, so it seems obvious to extend our deepest thanks to them for being with us through all the good and bad times.

Rahul Sharma (2023UGEC115)
Akarsh Mishra (2023UGEC050)
Harsh Sundi (2022UGEC113)

TABLE OF CONTENTS

Abstract

The AI Website Builder is an innovative no-code platform designed to make professional web development accessible to everyone, regardless of technical background. It enables users to generate dynamic, fully functional React websites simply by providing natural language prompts. Built with cutting-edge technologies including Next.js, React, Sandpack, Convex, and Gemini (LLM), the platform transforms user input into real-time editable code while offering a live website preview within the browser.

Users can easily customize their websites through an interactive file explorer, edit code directly, manage their workspace securely, and export complete projects without writing a single line of code manually. The system ensures a smooth, responsive experience by leveraging sandboxed environments for code execution and real-time database syncing.

By abstracting the complexity of modern web development, the AI Website Builder dramatically reduces the time and effort required to build, test, and deploy production-ready websites. It empowers entrepreneurs, designers, content creators, and innovators to bring their ideas online quickly and effectively. This project represents a major leap forward in democratizing web creation, combining intelligent automation with modern frontend engineering to bridge the gap between creativity and technology.

Chapter 1

INTRODUCTION

The **AI Website Builder** is a cutting-edge no-code platform designed to democratize web development, making it accessible to users regardless of their technical background. It empowers individuals to create fully functional, production-ready websites based on the powerful React framework, simply by using natural language prompts. This approach removes the traditional barriers associated with learning to code, hiring developers, or working within the rigid limitations of prebuilt templates.

In today's fast-paced digital economy, the ability to quickly establish a strong online presence is critical for entrepreneurs, startups, content creators, and small businesses. However, conventional web development remains complex, expensive, and time-consuming. While existing no-code platforms offer visual editors and basic templates, they often lack the flexibility needed for fully customized, dynamic website creation.

The AI Website Builder addresses these gaps by leveraging modern frontend technologies like **Next.js**, **React**, and **Sandpack**, along with the intelligence of large language models such as **Gemini**. The system allows users to describe their ideas in plain English and instantly receive clean, editable React code, coupled with a real-time preview. Users can also manage files, edit code directly, and view changes live, creating an intuitive development workflow.

Beyond frontend capabilities, the platform also integrates backend functionalities through **Convex**, providing secure user authentication, real-time database storage, and personalized workspace management — all without the user writing a single line of backend code.

By abstracting complex development tasks and merging AI with state-of-the-art technologies, the AI Website Builder significantly reduces the time, cost, and technical effort required to launch a website. It offers a scalable, flexible, and user-centric solution that empowers anyone to turn their ideas into reality, representing a major step forward in the evolution of no-code and AI-driven software development.

Chapter 2

Literature review

The emergence of artificial intelligence in software development has significantly transformed the way applications and websites are built. AI-powered code generation tools, such as OpenAI Codex, Gemini, and GPT-4, have introduced intelligent assistance that enables users to write, optimize, and debug code more efficiently. These advancements have paved the way for a new generation of no-code and low-code platforms that aim to democratize software development for non-programmers.

Existing platforms like GitHub Copilot assist developers by suggesting code snippets in real-time, while visual builders like Webflow and Framer allow users to create websites through drag-and-drop interfaces. However, these solutions often either rely heavily on manual code editing or restrict users to predefined templates, limiting customization and flexibility.

The AI Website Builder addresses these limitations by integrating real-time AI text-to-code generation with a live editing environment. Unlike traditional visual editors, it outputs actual, editable React code, combining the flexibility of professional development with the accessibility of no-code tools. Technologies like Sandpack enable in-browser execution of React applications, while Convex provides seamless backend support for real-time database operations and authentication.

By blending intelligent automation with dynamic frontend tooling, this project moves beyond static website creation and toward true interactive, customizable web application development. The AI Website Builder thus represents a significant evolution in the literature of code generation and no-code platforms, positioning itself as a bridge between creative vision and functional software engineering for a broader audience.

Chapter 3

Problem Statements

Building dynamic and production-ready websites typically demands strong technical skills in coding languages such as HTML, CSS, JavaScript, and frameworks like React. For non-developers, this creates a steep learning curve and significant barriers to entry. Although no-code and low-code platforms have simplified web creation to some extent, they often rely on rigid templates and visual editors that restrict true customization and flexibility.

Meanwhile, AI-driven coding tools are primarily designed for users who already possess technical expertise, offering minimal assistance to beginners or non-programmers. As a result, individuals with creative ideas but no technical background struggle to create unique, functional websites quickly and affordably. There is a clear need for a more intuitive and flexible solution that empowers non-developers to build fully customized web applications without writing code.

Chapter 4

Ojectives

The primary objective of the AI Website Builder project is to simplify the process of website creation for users without any coding knowledge. This platform is designed to allow users to build production-ready, dynamic React websites simply by interacting with a natural language-based interface. Specific objectives include:

- Enable natural language-based website generation using AI models like Gemini, making web development accessible to non-programmers.
- Provide real-time code editing with a live preview feature using Sandpack, helping users instantly see the effects of changes.
- Ensure full code ownership by allowing users to export and manage the generated code files easily.
- Integrate secure user authentication and real-time database operations using Convex for reliable workspace management.
- Offer a flexible, scalable architecture that can evolve to support more frontend frameworks and backend generation in the future.
- Reduce time and cost barriers for individuals and businesses seeking rapid digital transformation.

By focusing on user-centric design, seamless AI integration, and dynamic React-based development, the AI Website Builder aims to revolutionize how websites are created, making it faster, more intuitive, and more affordable.

Chapter 5

Methodology

The AI Website Builder was developed through a systematic approach combining research, technology evaluation, prototype development, and iterative refinement. The methodology can be broken down into the following stages:

1. Research and Analysis:
   Initially, a detailed study of existing no-code platforms (Webflow, Framer) and AI code generators (Codex, Gemini) was conducted to understand their strengths, limitations, and user requirements.

2. Technology Stack Selection:
   Next.js and React were chosen for the frontend because of their efficiency in building dynamic applications. Sandpack was selected for live preview capabilities, while Convex provided a robust backend for real-time data management and authentication.

3. Design and Architecture:
   A modular, scalable system architecture was designed where the frontend interacts with Gemini for AI-driven code generation, and Sandpack for live code execution.

4. Development:
   The frontend components, AI integration, and backend services were developed iteratively. Key features like live editing, file management, and user authentication were implemented step-by-step.

5. Testing:
   Extensive testing was conducted to ensure smooth live previews, secure user sessions, and accurate AI responses.

6. Deployment:
   The final project was deployed with a focus on performance optimization and scalability for future expansion.

This structured methodology ensured the delivery of a robust, user-friendly platform that meets the needs of a broad audience.

Chapter 6

Implementation Details

The implementation of the AI Website Builder involves the seamless integration of frontend, backend, and AI systems:

- Frontend:
  Built using Next.js, React, and Tailwind CSS. Sandpack was integrated to render live previews of user-generated code directly in the browser. A dynamic workspace interface was developed, featuring file management, live code editing, and a real-time preview panel.

- Backend:
  Convex was used as a backend-as-a-service to manage database operations, file storage, and user authentication. It offers real-time data syncing through mutation hooks, ensuring that any workspace change is immediately saved.

- AI Layer:
  Gemini LLM was integrated to handle natural language prompts. The user's text input is processed, and the corresponding React code is generated and displayed in the editor. A chat-based AI assistant was also included for additional help and customization tips.

- Authentication and Authorization:
  Convex's built-in authentication modules were used to create secure user sessions, workspace isolation, and personalized storage solutions.

- Code Export:
  Users can download the full project code, ensuring they own their website and can deploy or modify it independently.

Overall, the project architecture ensures a lightweight yet powerful experience, supporting future extensions like API route generation and full-stack capabilities.

Chapter 7

Challenges Faced and Solutions

During the development of the AI Website Builder, several challenges emerged:

- Safe Execution of AI-Generated Code:
  Executing AI-generated React code in the browser posed a security risk.
  Solution: Sandpack was used, which runs code in a secure, isolated iframe, preventing unauthorized access to user data or the broader application.

- Real-Time AI Integration Without Lag:
  Generating and displaying code dynamically could introduce significant delays, especially with large prompts.
  Solution: Asynchronous communication with Gemini API and loading indicators were implemented to improve responsiveness and user experience.

- Workspace Scalability and Management:
  Managing multiple user workspaces dynamically required an efficient backend structure.
  Solution: Convex's real-time database and mutation hooks were used to ensure seamless syncing of workspace changes.

- Maintaining UI/UX Consistency:
  Designing a clean, intuitive, and responsive interface for complex operations like live coding and file navigation was challenging.
  Solution: Modular frontend components were built with React Context for global state management and Tailwind CSS for consistent styling.

These challenges were systematically addressed to deliver a high-quality, efficient, and scalable platform.

Chapter 8

Results and Discussions

8.1 Result

The AI Website Builder project successfully achieved its primary objectives of simplifying web development for non-developers and providing a fully functional, AI-driven, no-code platform. After completing the development and deployment phases, several key results were observed:

Real-Time Code Generation:
The integration of Gemini enabled the seamless conversion of natural language prompts into React code within seconds. Users could input website ideas in plain English and see instant code output with a live preview rendered by Sandpack.

Interactive Workspace:
The workspace supported full code editability and live updates, offering a highly responsive environment. Users could modify the generated code manually if desired, providing both automation and flexibility.

Secure User Authentication:
Using Convex, user login, session management, and workspace saving were implemented efficiently, ensuring personalized and secure user experiences.

Code Export Functionality:
Users were able to export the generated codebase, allowing independent hosting, deployment, or further customization outside the platform.

Performance:
The system performed smoothly even under heavy use cases, with AI response times averaging less than 2 seconds and code previews rendering within milliseconds.

8.2 Discussions

The project proved that combining AI with frontend and backend cloud services can lower the entry barrier to web development significantly. Unlike template-driven builders, users retained complete control over the code, resulting in higher customization potential. The choice of Sandpack over heavier alternatives like WebContainers also improved browser performance, making the platform lightweight and accessible even on mid-range devices.

However, the AI occasionally generated imperfect or verbose code, especially for complex prompts. This indicates the need for continuous fine-tuning of AI prompts and post-processing logic to enhance output quality

Chapter 8

CONCLUSION AND FUTURE SCOPE

6.1. Conclusion

The AI-powered web builder is a transformative solution that redefines the way frontend development is approached, making it accessible, efficient, and intuitive for both developers and non-technical users. Traditional frontend development requires mastering complex tech stacks, setting up environments, and writing extensive code—barriers that often hinder rapid prototyping and innovation. This platform eliminates those obstacles by leveraging the advanced capabilities of Gemini AI to generate clean, production-ready React code from simple natural language prompts.

One of the key strengths of this tool is its fully interactive development environment, which includes a live preview feature. This allows users to see their changes in real time, tweak designs, and refine functionality without switching between multiple applications or dealing with complex debugging processes. The integration of user authentication ensures that individuals and teams can work in personalized, secure environments, facilitating collaboration and project continuity.

By removing dependency on external tools, manual coding, and extensive debugging, this AI web builder significantly accelerates the development lifecycle. It empowers entrepreneurs, designers, and business professionals to quickly bring their ideas to life without needing deep programming expertise. Furthermore, it serves as an educational aid for aspiring developers, helping them understand React component structures and best practices through AI-generated examples

In essence, this platform bridges the gap between conceptualization and execution, enabling faster iteration, reducing development costs, and democratizing web creation. As AI continues to evolve, tools like this will play a crucial role in shaping the future of no-code and low-code development, making advanced web applications accessible to a much broader audience.

6.2 Future Scope of the Project

1.Full-Stack AI Integration - Expand beyond frontend to automate backend development with AI-generated APIs (REST/GraphQL), database schema design (Firebase/Supabase), and built-in authentication (OAuth/JWT) to enable complete app creation without separate backend developers, making end-to-end development accessible to all users.

2.Multi-Framework & Cross-Platform Support - Extend AI code generation to Vue.js, Svelte, and Angular while adding React Native for mobile apps, allowing users to choose their preferred framework and build for web and mobile simultaneously from the same intuitive prompt-based interface.

3.Design-to-Code AI (Figma/Image Input) - Implement Figma/screenshot uploads that auto-generate pixel-perfect React code by interpreting layouts, fonts, and components, bridging the designer-developer gap and accelerating prototyping by converting visual designs directly into functional code.

4.One-Click Deployment & Scalable Hosting - Integrate direct publishing to Vercel/Netlify/AWS with pre-configured CI/CD pipelines and serverless function support, enabling users to go from prompt to production in one click while handling everything from hosting to backend logic seamlessly.

Chapter 9

REFERENCES

- OpenAI. (2023). *Codex: An AI System for Code Generation*. Retrieved from https://openai.com/research/codex

- Google DeepMind. (2024). *Gemini: Advancing Multimodal and Code Generation AI*. Retrieved from https://deepmind.google/technologies/gemini/

- Vercel. (2023). *Next.js Documentation*. Retrieved from https://nextjs.org/docs

- Facebook Open Source. (2024). *React – A JavaScript Library for Building User Interfaces*. Retrieved from https://react.dev/

- Sandpack (CodeSandbox). (2024). *Sandpack: Live Running Code Editor for the Web*. Retrieved from https://sandpack.codesandbox.io/

- Convex Devs. (2024). *Convex – Backend for Frontend Developers*. Retrieved from https://docs.convex.dev/

- Webflow. (2024). *Visual Web Development Platform*. Retrieved from https://webflow.com/

- Framer. (2024). *AI-Powered Website Builder*. Retrieved from https://framer.com/