

# WebCraft.AI: An Intelligent MERN-Stack Platform for Automated Website Generation

Manish Sharma, Mayank Tyagi, Nitik Tyagi, Shlok Kumar, Ms. Swati Nagar

Dept. of Computer Science

Sunder Deep Engineering College, Ghaziabad, India

Msharma4121@gmail.com, maytyagi000@gmail.com

nitiktyagi3@gmail.com, shlok5141@gmail.com, Swati.nagar@sunderdeep.ac.in

**Abstract:** *WebCraft.AI is a platform that uses intelligence to create websites. It takes what you tell it. Turns that into a real website. You do not need to know how to code to use WebCraft.AI.*

*WebCraft.AI uses something called DeepSeek and Open Router to generate code for the website. It also uses Google Generative AI to make sure the website is what you wanted.*

*The platform is secure and safe to use. It uses Google OAuth 2.0 to authenticate users and Stripe to process payments. WebCraft.AI also has a system where you can buy credits to use the platform. This means that anyone can use it from people who just want a website to big companies.*

*The good thing about WebCraft.AI is that it saves a lot of time. It can create a website faster than a person can. The code is also good. The website will work well on different devices.*

*WebCraft.AI is a deal because it makes it easier for people to create websites. It does not matter if you know how to code or not. This platform can help people who want to start a business or just need a website.*

*The people who made WebCraft.AI want to keep making it better. They want to add features and make it easier to use. This means that WebCraft.AI will keep getting better and better.*

*Some important things about WebCraft.AI are:*

*it uses AI to generate websites*

*it uses the stack*

*it has a system for buying credits*

*it is safe and secure*

*it can create websites quickly*

*it is good for people who do not know how to code*

*it is good, for big companies and small businesses*

*it is a platform that will keep getting better..*

**Keywords:** *WebCraft.AI*

## I. INTRODUCTION

The internet has changed the way we do business. Now people want to have a website. Making a website is not easy. You need to know a lot about HTML, CSS and JavaScript which takes a time to learn. If you hire someone to do it for you it can be very expensive. Using templates to make a website does not give you a lot of freedom to make it your own way.

Recently computers have gotten very good at understanding what people are saying. Can even write code that works. This is very helpful when combined with tools that make it easy to build a website. This makes it possible for anyone to make a website even if they do not know a lot about computers.

This paper is, about WebCraft.AI, a website that can turn what you say into a website that you can use right away. The main things we did are:

Copyright to IJARSCT

[www.ijarsct.co.in](http://www.ijarsct.co.in)



DOI: 10.48175/IJARSCT-34616



- we made a way for the computer to understand what you say and turn it into a website
- we made a secure way for people to use our website using Google and Stripe to help with this.
- we made a system where people can pay for what they use so we can keep making our website better.
- we made it so you can see how your website looks on different devices like a phone or a computer while you are still making it.

## **II. RELATED WORK**

People have been working on generating code from natural language for a long time starting with simple rule-based systems [2]. Then models that use transformers came along. Made the code much better. For example: Codex [3] showed that models like GPT can create working JavaScript code from specifications. DeepSeek-Coder made it possible to generate code for files [4].

When it comes to building websites some systems like Wix ADI use rules to put together parts that are already made. Webcraft.AI is different because it generates the basic code for HTML/CSS/JavaScript, from scratch and you can customize it however you want. This gives you a lot freedom to design your website. Open Router [6] is also interesting because it provides a way to access many different language models, which helps you save money and avoids getting stuck with one vendor.

## **III. SYSTEM ARCHITECTURE**

WebCraft.AI is made up of three parts: the frontend that users see which is built with React.js, the backend that does all the work, which is built with Node.js and Express.js and the database that stores all the information, which is built with MongoDB. The people who made WebCraft.AI chose to use these tools because they all use JavaScript, which makes it easier for the different parts to work together and share information like validation rules with each other.

### **A. Frontend**

The React.js SPA includes prompt editor, real-time website preview panel, component property inspector, and project management dashboard. State management employs React Context API. Responsive layouts use Tailwind CSS utility classes.

### **B. Backend API**

The Express.js API has a lot of endpoints that are used for things. These endpoints are divided into four areas:

- /auth for Google OAuth,
- /generate for Artificial Intelligence requests,
- /projects for creating and managing projects
- and /billing for handling Stripe webhooks.

The Backend API uses a lot of Backend API middleware to make sure that users are authenticated with JWT authentication that database changes are handled correctly with credit atomicity and that the Backend API does not get many requests at the same time with rate-limiting.

### **C. Database**

MongoDB stores kinds of website data.

The main collections are:

- \* Users which include profile, OAuth, subscription and credits
- \* Projects which have prompts, code and URLs
- \* Transactions which have Stripe logs and credit allocations.



#### **IV. AI GENERATION PIPELINE**

The pipeline takes what people write and turns it into things that can be used on a website. It does this in four steps: it looks at what people write it uses a language model with something called OpenRouter it checks to make sure the answer is correct and then it sends the answer to the person who asked for it. The AI generation pipeline is what makes all of this work by taking the words people write and turning them into things that can be used on a website through these four steps.

##### **A. Prompt Engineering**

We add some details to the basic prompts. These details are like a set of instructions that tell the system what to do. They make sure the output is in a format like semantic HTML5 and that it is easy to use on a phone. They also help the system follow rules about accessibility like WCAG 2.1 so everyone can use it. This way Prompt Engineering helps the system give answers and not make things up that are not real, like non-standard APIs.

##### **B. DeepSeek via OpenRouter**

DeepSeek uses OpenRouter to send requests, to the deepseek-chat endpoint. This happens through OpenRouter's API. It works in streaming mode. The SSE channel is what lets us see things before they are completely done. We picked DeepSeek because it did really on the HumanEval-Web benchmark. It was also very cost efficient [4].

##### **C. Response Validation**

The HTML we generate goes through a check to make sure it is okay. We do this in three steps.

1. We look at the structure of the HTML using a tool called html-parser.
2. We take out the CSS. Check if it is valid according to the rules of the W3C.
3. We check the JavaScript code to see if it has any problems using a tool called Acorn.

If something goes wrong. We try again one time. This way the HTML works fine on the first or second try 97.3 percent of the time.

#### **V. AUTHENTICATION AND PAYMENT**

##### **A. Google OAuth 2.0**

I use Google OAuth 2.0 for authentication. It works with Passport.js to get rid of the hassle of passwords. When I log in Google gives me a code. This code helps get a short-term JWT that lasts for an hour and a long-term refresh token. These are stored safely in cookies that can't be accessed directly from the browser.

##### **B. Stripe Integration**

We use Stripe's API, for subscription billing. Each plan we offer is linked to a Stripe Product. When you checkout we use Stripe Checkout Sessions to handle payments in a way that meets PCI standards.

Our system also uses webhooks to give you credits and update your subscription level in our MongoDB database.

We make sure to handle events reliably by checking webhook signatures.

Stripe's test mode helped us test our integration thoroughly.

#### **VI. TIERED CREDIT SYSTEM**

The WebCraft.AI system uses a tiered credit system. This means it tries to make things fair for everyone while also keeping costs under control. When you use the AI to generate something it takes one credit. We store these credits in a place called MongoDB and subtract them one by one to avoid any problems.



TABLE I. SUBSCRIPTION TIERS

Tier	Credits/Mo	Features
Free	100	Basic AI generation
Pro	500	Faster Generation, editing
Enterprises	1000	Team collaborations, Dedicated support
Enterprise	Unlimited	White-label, API

If you need more credits you can buy them at any time using Stripe. This is useful if you need to do a lot of things at once and do not have credits. We show you how credits you have left at the top of the application.

## VII. EXPERIMENTAL EVALUATION

### A. Methodology

I evaluated WebCraft.AI across three areas.

1. I checked if it generated HTML, CSS and JavaScript code.
2. I measured how time it saved developers when creating a prototype.
3. I asked 45 users to rate how easy it was to use.

The test prompts included five types of web pages:

landing pages  
portfolios  
e-commerce sites  
SaaS dashboards  
event microsites

I used the System Usability Scale to get user feedback.

This helped me understand how well WebCraft.AI performed.

The evaluation was thorough. Covered various aspects of web development.

WebCraft.AI generated code that was valid and worked as expected.

The tool helped developers create prototypes quickly.

Users were satisfied with the usability of WebCraft.AI.

WebCraft.AI performed well in generating code.

It also helped improve developer productivity with WebCraft.AI.

The user satisfaction ratings, for WebCraft.AI were good.

### B. Results

The results show that things are working correctly 97.3 percent of the time when we check them the first or second time across 500 prompts. It takes 4.2 minutes to get a prototype with WebCraft.AI. It takes 28.6 hours to do it manually. That is an 85.3 percent reduction in time. The SUS score is 84.6, which is considered Excellent.

The time it takes for the API to respond is 8.3 seconds on average for a website. When it comes to MongoDB the transaction overhead is very low, below 12ms even when 500 users are using the website at the time, which is the 99th percentile. The WebCraft.AI results are very good. The MongoDB results are also very good.

## VIII. CONCLUSION

WebCraft.AI shows that using the language models with a good website system makes a platform that people can really use to make websites with artificial intelligence. The combination of DeepSeek and OpenRouter and Google OAuth 2.0



and Stripe billing and a credit system that is easy to understand makes a foundation that's safe and makes money and people can use it. The results of the tests show that people can make websites faster without making the code bad. Future work will look at making websites with pages that look the same and making search engines, like the website and letting people work on the website at the same time and using other company's tools. WebCraft.AI will use something called RLHF to make the website look nicer. WebCraft.AI and the language models will get better at making websites that people like.

#### **ACKNOWLEDGMENT**

We would like to thank Ms. Swasti Nagar for her helpful advice during this research.

We are also grateful to the people behind the stack OpenRouter and Stripe for making their tools available, for free.

This project did not receive any funding from sources.

#### **REFERENCES**

- [1]. T. Brown et al., "Language Models are Few-Shot Learners," NeurIPS, vol. 33, pp. 1877–1901, 2020.
- [2]. C. Deikel & C. Manning, "Natural Language to Code Generation," ACL Workshop, pp. 12–18, 2013.
- [3]. M. Chen et al., "Evaluating LLMs Trained on Code," arXiv:2107.03374, 2021.
- [4]. DeepSeek-AI, "DeepSeek-Coder: LLM Meets Programming," arXiv:2401.14196, 2024.
- [5]. Wix.com, "Artificial Design Intelligence (ADI)," Wix EngineeringBlog, 2018.
- [6]. OpenRouter Inc., "OpenRouter: Unified LLM Interface," TechnicalDocumentation, 2024.
- [7]. P. Liu et al., "Pre-Train, Prompt, Predict: Prompting Methods Survey," ACM Surveys, 2023.

