Artificial Intelligence Driven Website Composer

Sergio R. Cepeda Delgado
Dr. Jeff Duffany
Computer Science Department

Abstract

This study outlines the creation of an Artificial Intelligence (AI)-powered web composer that utilizes OpenAI's Application Programming Interface (API) models GPT-4o and DALL-E-2 together with Python and HTML code, to enable the automatic creation of web pages. This composer enables users to define their design preferences and content requirements, which are subsequently translated and implemented by the AI using sophisticated natural language processing and deep learning methods. The incorporation of these technologies eliminates the need for substantial technical expertise, therefore expanding the availability of web design to a wider range of users. Comparative assessments using traditional web development approaches demonstrate a substantial improvement in efficiency and user happiness, accompanied by notable decreases in production time. This research highlights the significant impact that AI may have on web development and lays the groundwork for future investigations into AI-powered design tools.

Introduction

The digital era demands efficient web development, but it remains a complex task requiring technical expertise. Recent AI advancements offer opportunities to streamline this process. The AI-driven web composer leverages OpenAI's API to translate user design preferences and content requirements into functional web pages [3].

Background

Web development has evolved from static HTML to dynamic applications, demanding aesthetic and technical skills [1]. Traditional methods are resource-intensive, limiting non-technical users. AI, particularly in natural language understanding and deep learning, offers potential to bridge this gap, enabling broader participation in web design [6].

Problem

A significant challenge in web development is the disparity between the need for quality web design and the skills available [4]. Current tools still require a solid understanding of design principles and manual adjustments. AI can transform this process but integrating it in an intuitive and effective manner for non-experts is complex.

Methodology

The System Architecture for this project uses a server-side environment where OpenAI API interacts with a Python backend and HTML frontend.

```
    </header>
    <h3>This tool uses the OpenAI API to generate a unique website as specified</h3>

    <form action="/generate" method="post">
        <label for="user_input">Enter your request:</label><br>
        <textarea id="user_input" name="user_input" rows="4" cols="50"
        placeholder="What website you like the AI to build?"></textarea><br>
        <input type="submit" value="Generate Website">
    </form>
</div>
```

Figure 1: HTML code for the AI Web Composer Homepage that allows the user to input a request to be sent for generation

API Integration [2]:

- User inputs are processed by Python scripts, generating design elements.
- GPT-4o model integration for natural language processing to generate web code .

```
@app.route('/generate', methods=['POST'])
def generate():
    """...

    if request.method == 'POST':
        user_input = request.form['user_input']

        #Generate HTML content using OpenAI API
        try:
            completion = client.chat.completions.create(
                #model="gpt-3.5-turbo",
                model="gpt-4o",
                messages=[
                    {"role": "system", "content": "Only return HTML code with embedded
                    styles, javascripts, CSS and photos. Create a complex and complete
                    website with multiple pages, containing all the content you want to
                    include. Do not include any markdown. Generate or find the photos
                    needed for the website."},
                    {"role": "user", "content": user_input},
                ]
            )
            html_content = completion.choices[0].message.content

        except Exception as e:
            print(f"Error generating completion: {e}")
            return make_response("Error generating website content.", 500)

        #Save the modified HTML content to a file
        file_name = "generated_website.html"
```

Figure 2: Section of Python code that implements the gpt-4o interaction by sending a system instruction and the received user request

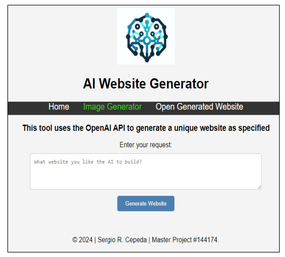
- DALL-E-2 model integration to create unique visual content.

```
@app.route('/generate_image', methods=['POST'])
def generate_image():
    """..."""
    if request.method == 'POST':
        user_input_image = request.form['user_input_image']

        #DEBUG : Print the generated HTML content to the console
        print (user_input_image)

        try:
            responseimg = client.images.generate(
                model="dall-e-2",
                prompt= user_input_image,
                n=1,
                size = "512x512"
            )
            image_url = responseimg.data[0].url

            #DEBUG : Print the generated image URL to the console
            print (image_url)

            response = requests.get(image_url)
```

Figure 3: Section of Python code that implements the dall-e-2 interaction by sending the specifications of the image and the user request (prompt)

Methodology

The UI Developed provides a user-friendly interface for users to specify what the AI will generate together with any other specification.



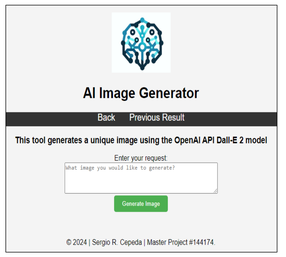


Figure 4 and 5: Home pages for the website generator and image generator where users can input their request.

Training data: The AI's ability to generate attractive and functional web designs uses training data for GPT-4o is currently up to October 2023 [3].

Results and Discussion

The AI-driven web composer effectively automates web design, translating user descriptions into HTML, CSS, and JavaScript code. Compared to traditional methods [5], it reduces development time by up to 50% and minimizes errors. DALL-E 2 enhances the visual content, creating custom images that fit the design context. The tool offers a simpler, template-based solution compared to other platforms like Wix and Jimdo, making web design more accessible.

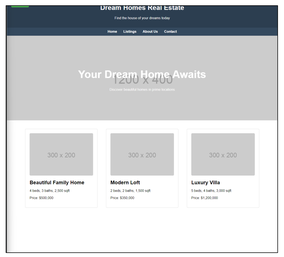


Figure 6 and 7: Example results of the Website Generator

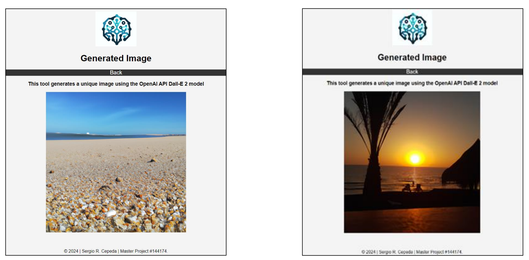


Figure 8 and 9: Example results of the Image Generator

Conclusions

The AI-powered web composer demonstrates AI's potential to democratize web development, allowing non-experts to create high-quality web designs. By integrating OpenAI's API with Python and HTML, it significantly enhances efficiency and creativity in web creation. Future work should focus on improving code generation, enhancing visual content, and incorporating user feedback for continuous improvement.

Future Work

- Improving Code Generation: Upgrade to newer GPT models for more complex and functional websites.
- Enhanced Visual Content: Use advanced DALL-E models for higher resolution and contextually relevant images.
- User Feedback Integration: Implement a systematic feedback loop for iterative AI performance enhancements.

Acknowledgements

I would like to extend my deepest gratitude to my advisor, Dr. Jeff Duffany, for his invaluable guidance and support throughout the duration of this project. His commitment to academic excellence and his dedication to nurturing my professional and personal growth have been profoundly inspiring. I am truly thankful for his mentorship and encouragement.

References

[1] "The web and web standards - Learn web development | MDN," MDN Web Docs, Oct. 08, 2023.https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards

[2] "OpenAI Platform." https://platform.openai.com/docs/api-reference/introduction

[3] "OpenAI Platform." https://platform.openai.com/docs/models

[4] I. Education and I. Education, "AI code-generation software: What it is and how it works," IBM Blog, Sep. 19, 2023. https://www.ibm.com/blog/ai-code-generation/

[5] C. Giattino, E. Mathieu, V. Samborska, and M. Roser, "Artificial Intelligence," Our World in Data, Jan. 31, 2024. https://ourworldindata.org/artificial-intelligence#all-charts

[6] "OpenAI Cookbook." https://cookbook.openai.com/