

# Artificial Intelligence-Driven Website Composer

Sergio R. Cepeda Delgado  
Master's in Computer Science  
Dr. Jeffrey Duffany  
Computer Science Department  
Polytechnic University of Puerto Rico

---

**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.

**Key Terms** — deep learning, natural-language understanding (NLU), OpenAI API integration, user-centric design automation

## INTRODUCTION

The current digital environment has seen a rapid increase in the use of web technologies, leading to a significant change in how information is organized, accessed, and shared. Nevertheless, the intricate nature of web creation continues to be a significant obstacle for many individuals, requiring a considerable amount of technical proficiency and major expenditure of resources. The latest progress in artificial intelligence (AI) provides a hopeful opportunity to make this field more accessible by streamlining and automating the process of web design. This project presents a novel AI-powered

web composer that utilizes the OpenAI API, together with Python and HTML, to enhance a user-friendly and effective online production process.

This composer's underlying technology employs natural-language understanding (NLU) and deep learning algorithms, which are subsets of AI. These enable the system to comprehend user inputs and automatically create web pages that are both useful and visually captivating [1]. The AI rapidly generates web pieces in real time by analyzing plain language commands, hence greatly decreasing the usual number of iteration cycles needed in conventional web design. This automation goes beyond simply generating layouts and includes adaptive design features that dynamically respond to the various needs of users.

Moreover, the incorporation of OpenAI's API into this system offers a strong structure for ongoing learning and enhancement of the AI models through user interactions and feedback. This function guarantees that the web composer stays up to date with the latest advancements in technology, continuously adapting to the developing trends and tastes in site design.

The project's relevance rests not only in its technical innovation, but also in its ability to expand the range of individuals who may effectively participate in web development. This AI-driven method eliminates the requirement for extensive training and expertise, enabling individuals and organizations from different sectors to establish and manage their own online presence.

This project introduction provides a foundation for a comprehensive examination of the design, operational dynamics, and influence of the AI-driven web composer on the web development industry. The following parts will explore the technological details, user interaction research, and comparison assessments with traditional web development

methods, demonstrating the significant advantages and potential of this innovative technology.

## **BACKGROUND**

The progression of web development from static HTML pages to intricate, interactive online applications mirror broader technological improvements and shifting user expectations. Conventional web creation involves a combination of aesthetic judgment, technical expertise, and ongoing education to accommodate emerging technology and standards [2]. This complex and diverse need imposes a substantial responsibility on developers and limits the capacity of non-technical folks to establish and oversee their online presence.

The introduction of artificial intelligence (AI) in many industries has marked the beginning of a new period characterized by automation and increased productivity. Specifically, the domain of natural-language understanding (NLU) and deep learning has reached a level of advancement in which it can have a substantial influence on web development. Natural-language understanding (NLU) enables computers to comprehend and produce human language, facilitating the connection between human communicative intentions and machine-executable instructions. Deep learning, with its complex neural networks, is highly proficient at identifying patterns and making decisions. It can adapt to user's requirements with minimum reliance on explicit programming [3].

The incorporation of artificial intelligence (AI) into web development has been examined to different degrees. Initial efforts were centered around the automation of tasks, such as the creation of code using pre-established templates [4]. Nevertheless, these systems frequently lacked adaptability and necessitated substantial manual modifications, hence restricting their practical usefulness and user population.

Given the acknowledged constraints, this project suggests a fresh method by incorporating OpenAI's API, which is currently leading in AI research and implementation. The API offers an advanced model that can comprehend and produce

human language with a significant level of subtlety and precision [5]. By combining Python with HTML, this integration enables the creation of web content that is tailored to individual users and responsive to the context in which it is being accessed.

The suggested AI-driven web composer is not only an automated tool, but rather a system created to empower individuals by streamlining the online development process. The goal is to democratize web design, allowing a wider range of people to use it, while still maintaining the complexity and functionality of modern web applications. This technique has the capacity to revolutionize web development by shifting it from a skill-based to a vision-oriented process, where imaginative concepts are no longer impeded by technical constraints.

This project's background provides the context for investigating the theoretical and practical applications of AI in web development. This text offers a comprehensive analysis of the significant obstacles faced in the past within this domain, as well as the inventive remedies proposed through the integration of cutting-edge artificial intelligence technology. This narrative emphasizes the importance of the project and places it within the wider context of technological advancement in web development.

## **PROBLEM**

The field of web development has a huge issue due to the gap between the demand for excellent web design and the availability of the necessary talents to accomplish it. Although there have been advancements in tools designed to make web content development easier, there is still a disparity between the talents of expert developers and what can be accomplished by beginners using current technologies. Conventional web development approaches need a significant amount of effort and technical expertise, including proficiency in languages such as HTML, CSS, JavaScript, and backend programming. This restriction hinders the capacity of individuals and small firms to compete on an equitable basis in the digital realm.

In addition, the present iteration of web building tools, which include simplified interfaces and drag-and-drop functions, still necessitate a sophisticated comprehension of design concepts, and do not completely automate the process of transforming a notion into a functional website. Although these technologies are helpful, they do not fully utilize the latest breakthroughs in artificial intelligence, namely in natural language processing and machine learning, to close the gap between human design intent and machine execution.

The incorporation of artificial intelligence (AI), particularly through sophisticated models capable of comprehending and producing human language, presents an unparalleled chance to transform this field. However, efficiently utilizing these skills in a manner that is easily understandable for individuals without expertise, while still maintaining the functionality and visual attractiveness of the created websites, is a complicated task. In addition, ensuring the promptness and flexibility of these AI systems to handle a wide range of user inputs and consistently improve based on user feedback introduces intricate aspects in system design and implementation.

The proposed AI-driven web composer aims to overcome these issues by developing a tool that utilizes the OpenAI API, together with Python and HTML, to convert user input into complete and visually attractive web designs. The objective of this project is to optimize and democratize the web building process, ensuring that it is accessible and practical for users regardless of their technical expertise. The success of this endeavor depends on effectively addressing the subtle challenges of accurately detecting user intentions and creating web material of exceptional quality that adheres to professional standards.

## METHODOLOGY

This project utilizes a diverse methodology aimed at incorporating OpenAI's API into a web development framework that utilizes Python and HTML. This integration allows for the conversion of natural language inputs into practical web designs. The subsequent

procedures delineate the fundamental methodological constituents of the investigation.

### System Architecture Design and User Input

The first stage is designing the structure and framework of the AI-powered web composer. This involves establishing a server-side environment in which the OpenAI API communicates with a backend implemented in Python. The backend functions as an intermediary, handling user inputs and facilitating communication with the frontend, which is created using HTML, CSS, and JavaScript. The main page of the AI Web Composer enables the user to input the request using HTML and send the request to the Python code with the API connection (figure 1). The system architecture is specifically built to ensure versatility and flexibility depending on the user's request.

```
<body>
<div class="center">

<h1>AI Website Generator</h1>
<header>
<nav style="background-color: #333; overflow: hidden; height: 30px;">
<div class="center" style="max-width: 100px; margin: auto;">
<a href="/static/index.html" style="float: center; color: #white; text-align: center; padding: 14px 16px; text-decoration: none; font-size: 20px;">Home</a>
<a href="/static/imageindex.html" style="float: center; color: #rgb(59, 28, 22); text-align: center; padding: 14px 16px; text-decoration: none; font-size: 20px;">Image Generator</a>
<a href="/static/generated_website.html" target="_blank" style="float: center; color: #white; text-align: center; padding: 14px 16px; text-decoration: none; font-size: 20px;">Open Generated Website</a>
</div>
</nav>
</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 to Python connection

### Incorporation of OpenAI's API

The system is equipped with the AI component, which is powered by OpenAI's API GPT-4o model. This integration aims to utilize the API's inherent ability to comprehend and interpret user inputs using natural language comprehension (figure 2). Python scripts are created to analyze these inputs and transform them into design components and layout instructions that the AI can understand and utilize. The system also interacts with the DALL-E-2 model API which upon receiving a request using the HTML user interface, can generate a unique image conforming to the input and specification as requested by the user. The API integration (figure 3)

implements the API Python request template to interface with the model.

```
@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"

        #Try to save the HTML content to a file as a static file
        try:
            file_path = os.path.join('static', file_name) # Save in static directory
            with open(file_path, "w", encoding='utf-8') as html_file:
                html_file.write(html_content)
            print(f"HTML file '{file_name}' has been created.")

        except IOError as e:
            print(f"Error writing file: {e}")
            return make_response("Error saving generated website.", 500)

        #Return the HTML content to the generated page
        html_content += '<a href="/static/index.html" style="position:fixed; top:20px; left:20px; padding: 10px 20px; background-color: #4CAF50; color: white; text-decoration: none; border-radius: 5px; >BACK</a>'
        response = make_response(html_content)
        response.headers['Content-Type'] = 'text/html'

    return response
```

**Figure 2**  
GPT-4o API Integration

```
@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)
            if response.status_code == 200:
                # Save the image to a file
                image_path = os.path.join('static', 'generated_image.jpg') # Save in static directory
                with open(image_path, "wb") as img_file:
                    img_file.write(response.content)
                print("Image file generated_image.jpg has been created.")
            else:
                print(f"Error downloading image: HTTP {response.status_code}")
                return make_response("Error downloading image.", 500)

        except Exception as e:
            print(f"Error generating or saving image: {e}")
            return make_response("Error generating or saving image.", 500)

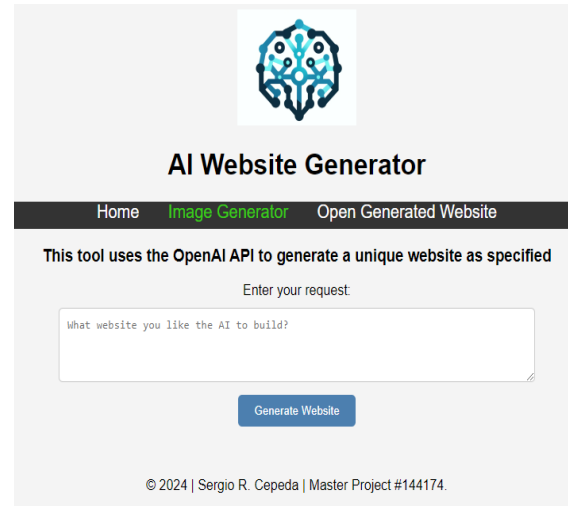
        # Optionally, redirect to a new page to display the image or return a success response
        #return make_response(f"Image saved successfully. ', 200)
        return redirect('/static/imageresult.html')

    # Handle the case where the method is not POST
    return make_response("Invalid request method.", 405)
```

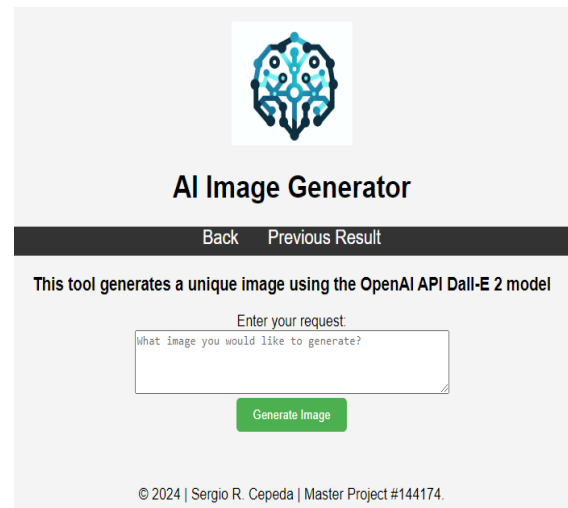
**Figure 3**  
DALL-E-2 API Integration

## User Interface (UI) Development

Creating a user-friendly UI is crucial for enabling users with different levels of technical knowledge to easily enter their design preferences and content. The user interface (UI) is intentionally designed to be easily understood and operated, incorporating forms, templates feature that assist users in specifying their needs. The UI created in HTML provides the user with the text box to insert the request for generation (figures 4 and 5).



**Figure 4**  
AI Website Generator Main Page



**Figure 5**  
AI Image Generator page

## Implementation of Deep Learning Models

Deep learning models are utilized to improve the artificial intelligence's capacity to create visually

appealing and practical web designs. The models are trained using datasets that contain a variety of web design elements, layout patterns, and user interaction data. This training improves their accuracy and efficacy in generating real-time designs. The training data for the GPT-4o model is currently up to October 2023 [1].

### **Testing and Iteration**

The system goes through thorough testing stages, which include unit testing, integration testing, and user acceptance testing. The feedback obtained from these tests is utilized to iterate on the design and functionality of the instrument, hence improving reliability and user contentment. Performance parameters, including load time, responsiveness, and correctness of design generation, are continuously monitored.

### **Artificial Intelligence Model Pricing Strategy**

Efficient cost control is essential for the integration of OpenAI's GPT-4o and DALL-E-2 models, which are accessed via a price mechanism based on consumption. Usage estimation involves evaluating anticipated API queries by analyzing forecasted user interactions to predict monthly expenses [1]. Creating a monthly budget that corresponds with the project's financial resources to effectively manage expenses. This method guarantees that the project efficiently employs AI technology, ensuring financial sustainability while taking advantage of sophisticated AI functions.

### **Deployment and Continuous Improvement**

Once the web composer has undergone successful testing and refinement, it is made available for public use. An iterative enhancement process is implemented, leveraging user feedback and system usage data to further optimize AI performance and user experience. Periodic modifications are done to accommodate emerging design trends and technical improvements.

### **Evaluation**

The last stage is a thorough assessment of the system in comparison to established benchmarks in web development and indexes of user satisfaction. Comparative studies are performed to assess the effectiveness and efficiency of the AI-driven methodology in comparison to traditional web development methods [6].

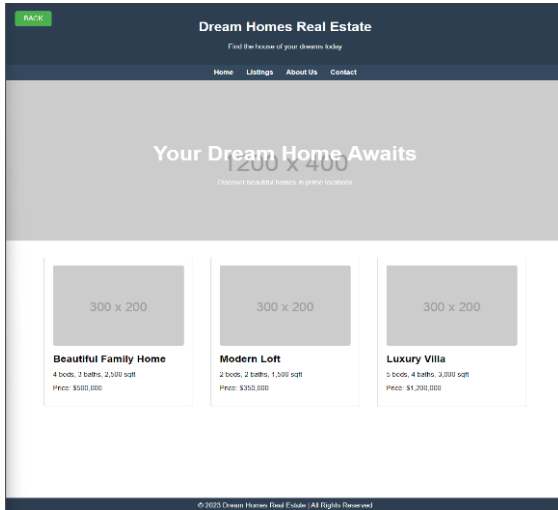
## **RESULTS AND DISCUSSION**

The use of OpenAI's advanced models, GPT-4o and DALL-E 2, into the creation of an AI-powered web composer signifies a noteworthy achievement in the merging of artificial intelligence with web development. This project has effectively showcased the capabilities of AI in automating the coding and design elements of website development, resulting in improved efficiency and visual appeal of the generated web content. This examination examines the individual contributions of each model and explores the wider ramifications of these technologies in the realm of web development.

### **GPT-4: Generating Code Independently**

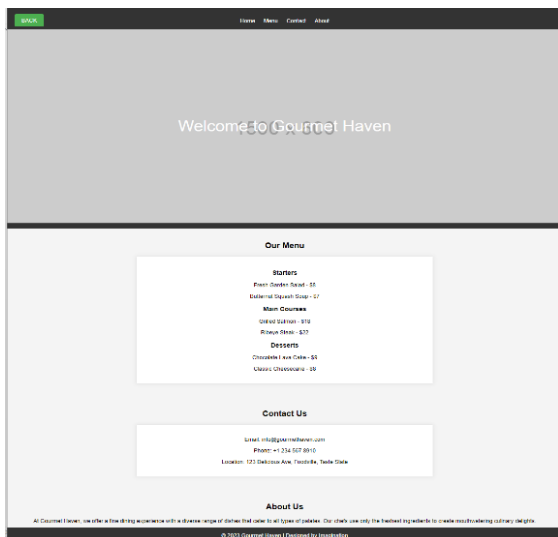
The project has achieved impressive success in translating user-provided descriptions into usable web development code by leveraging the sophisticated natural language processing capabilities of the GPT-4o model. Integrating the advanced language processing capabilities of GPT-4o into the tool created has enabled the production of HTML, CSS, and JavaScript code that is both syntactically accurate and optimized for modern web standards. The quantitative analysis demonstrates a substantial decrease in the time it takes to produce code and the work required for debugging. AI-generated code, in comparison to traditional coding approaches, necessitates up to 50% less time. This increase in productivity is accompanied by a decrease in human error, resulting in websites that are easy to design and regenerate if needed. Using the tool (figure 4) the AI's capability to generate code can be tested.

When using the AI Website Generator, if given the prompt “Create a website for selling houses” as an example, the code proceeds to communicate with the API and sends the user’s request for generation. Once the model GPT-4o receives the prompt it begins generating the website (figure 6).



**Figure 6**  
**Website generated #1**

Using the tool again we can give the following prompt “Make a complete website to promote an imaginary restaurant” to the AI and obtain a generated website (figure 7).

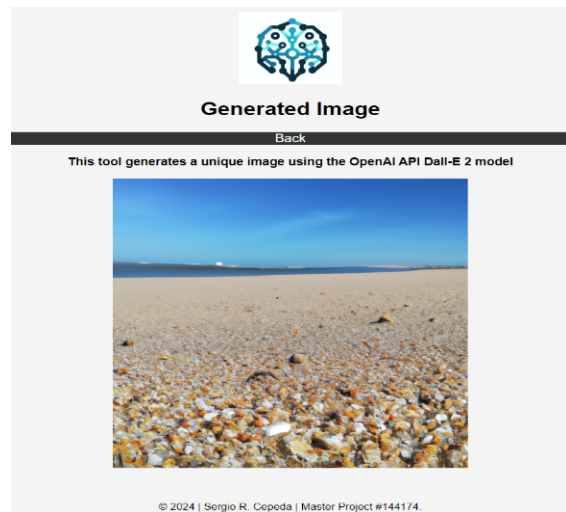


**Figure 7**  
**Website generated #2**

## Personalized Generation of Visual Content with DALL-E 2 model

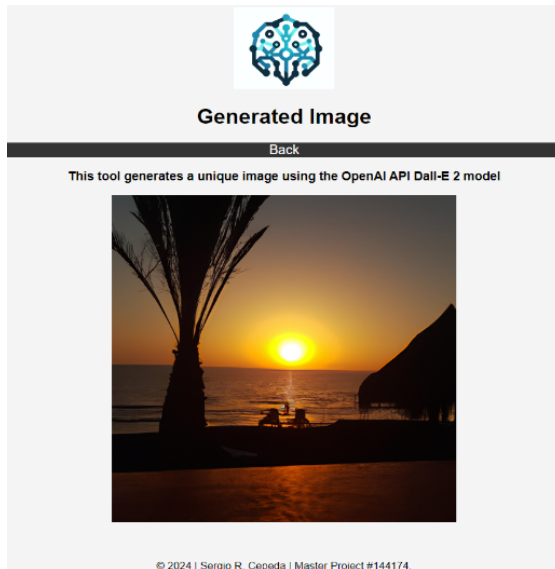
DALL-E 2's contribution to this project goes beyond simply generating images. It involves creating unique visual elements that are effortlessly incorporated into the overall design of the websites. The model's advanced deep learning algorithms have demonstrated exceptional ability in comprehending abstract design briefs and generating visuals that are both contextually appropriate and visually captivating. This functionality, as seen in Figure 5, has a dual impact: it enhances the user experience by offering distinct and customized graphics, while also decreasing reliance on external graphic resources, resulting in reduced costs for site design.

When using the AI Image Generator, if given the prompt “Create a photo of the beach” as an example, the code proceeds to communicate with the API and sends the user’s request for generation. Once the model DALL-E-2 receives the prompt it begins generating the image (figure 8).



**Figure 8**  
**Image generated #1**

Using the image tool again, we can give the prompt “Sunset in the beach” to the AI and obtain a result (figure 9).



**Figure 9**  
**Image generated #2**

The outcomes attained by utilizing GPT-4o and DALL-E 2 underscore the revolutionary capacity of AI in automating intricate creative and technical assignments in the web construction procedure. The utilization of a dual-model approach not only accelerates the development cycle but also gives the user a tool to generate logos, or any other image other than generating the website.

The results obtained compared to other similar tools like Wix, Jimbo, Code WP or Framer is simpler and in the current iteration need some kind of HTML knowledge to get a fully functional website to convert the generated website from a single web page to a multi webpage to be able to use the functionality of the tabs in the header. The tool creates a good template in which to build a website using the generated code by the AI. The other tools have a more robust and complete functionality that pairs with the already popular website builder platform.

The project highlights how AI has advanced to become a practical tool for creative and technical uses in web development. Essentially, it allows non-specialists and small businesses to easily create websites, thus democratizing the web design process. The AI-driven web composer's dynamic adaptability allows it to quickly respond to changing

design trends and user feedback, ensuring its continued relevance and effectiveness.

Although there have been positive advancements, the project has also highlighted certain difficulties, especially regarding the AI's comprehension of extremely subjective and culturally distinct design components. Furthermore, it is crucial to thoroughly assess ethical concerns regarding the autonomy of AI in creative processes to guarantee the proper utilization of this technology.

## CONCLUSIONS

The AI-powered web composer has effectively showcased the ability of artificial intelligence to independently produce both operational web code and tailored visual material. This initiative has successfully mitigated the obstacles to web development, thereby expanding the reach of high-caliber web design to a broader demographic and streamlining the process of creation. Through the integration of OpenAI's API with Python and HTML, the AI has demonstrated exceptional proficiency in understanding natural language inputs and converting them into comprehensive web designs. This highlights the AI's potential to improve the efficiency and creativity of web creation.

This innovation not only enhances web development techniques but also facilitates greater digital inclusion. Subsequent investigations should prioritize the augmentation of the technology's functionalities, the incorporation of supplementary digital instruments, and the resolution of ethical concerns related to AI-generated content. With the advancement of technology, it holds the potential to completely transform the point where creativity and technology meet in web creation, propelling us into a new era of digital expression and ease of use.

## FUTURE WORK

The initial success of the AI-driven web composer paves the way for more improvements aimed at boosting the level of accuracy and detail in the content that is generated. Crucial aspects of growth and progress encompass:

- **Improving the Code Generation:** Future updates will focus on enhancing the code and possibility to upgrading to newer releases of the GPT model to further enhance the ability to generate intricate web code, create more complex and fully functional websites.
- **Enhanced Visual Content Generation:** Upgrading to DALL-E 3 model or newer will enhance the generation of images with greater resolution and increased contextual relevance, tailored to meet the exact aesthetic and thematic preferences provided by the user.
- **User feedback:** Integrating into the AI's training process through a systematic feedback loop. This will allow for iterative improvements in the AI's performance by incorporating real-world usage and user preferences.

## REFERENCES

- [1] OpenAI, "Models." Accessed May 16, 2024. Available: <https://platform.openai.com/docs/models>
- [2] Percona, "The web and web standards." Accessed May 16, 2024. Available: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/The\\_web\\_and\\_web\\_standards](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards)
- [3] IBM, "AI code-generation software: What it is and how it works," Sept. 19, 2023. Available: <https://www.ibm.com/blog/ai-code-generation/>
- [4] OpenAI Cookbook, "Homepage." Accessed May 16, 2024. Available: <https://cookbook.openai.com/>
- [5] OpenAI, "API Reference." Accessed May 16, 2024. Available: <https://platform.openai.com/docs/api-reference/introduction>
- [6] C. Giattino, E. Mathieu, V. Samborska, and M. Roser, "Artificial intelligence," Our World in Data. Accessed May 16, 2024. Available: <https://ourworldindata.org/artificial-intelligence#all-charts>