

AI-based Front-End Generation

Advice Report.

Internship at
iO Digital

Internship Assignment Document

Luuk Briels
467020
0.0.1

Table of Contents.

1. Context	4
1.1 Introduction	4
1.2 Purpose of report	4
2. Recommendations	5
2.1 Interface	5
2.2 Functionalities	5
2.3 Models	7
2.4 Safety	8
3. Conclusion	9

Context.

L.

1.1 Introduction

During the project, I came across a lot of interesting ideas and insights for the proof of concept. I took notes on everything that seemed useful or promising. To help make sure the project keeps moving in the right direction and becomes even more successful, I've put together this advice report. In this report I've outlined some recommendations for different parts of the project. These include suggestions for the interface, the functionalities, the models, and safety.

1.2 Purpose of Report

The purpose of this report is to share the ideas and insights I gathered during the project. I want to help make sure that the project stays on the right path and can continue to be improved. This report holds my recommendations for different parts of the project/poc. My goal is to give useful advice that can help improve the project and hopefully make it an implemented tool for iO and its clients.

Recommendations.

2.1 Interface

The interface has been well tested to find any potential issues related to the user experience. The interface in its current state is regarded to be easy and very straightforward in use. However, there still are a few improvements that can be made to further the user experience.

Template Previews

Show previews of the templates. Because without it, users can only choose a template based on the title of it. Seeing what a template looks like might help choose the best one for their needs.

Tone of Voice Examples

Add examples for the tone of voice selector. This means showing how each tone of voice would create the same sentence and displaying this in the interface. This would help users better pick one that fits their needs.

Prompt Examples

Regarding the frame in which the page is generated, it could be helpful to add a few prompt examples along with a preview of that page that that prompt would generate. This helps the users create their own prompts and help them visualize what sort of page they would get back.

Header Tags

Inside the the generated pages, it might be helpful for some users to be able to view the header tags (H1, H2, etc.). This could be in the form of a toggable option.

Integrations Info Button

Add an info button on the Storyblok and Magento integrations explaining how users can get their tokens. These tokens are needed to create the connection to the platforms. However, it is not explained how these tokens are obtained.

2.2 Functionalities

In terms of functionalities, there are a lot of great and useful features that would help further improve the tool.

Generated Pages History

The ability to save generated pages to a users history. This would allow the user to go back and make changes anytime they like. However, this would likely also mean that an authentication functionality would need to be added, where pages are saved to an account.

Drag/Move Sections

Being able to drag and move sections around to rearrange the page. As well as being able to add and remove sections to a generated page. This would further allow users to customize the page to their liking.

More Image Control

Give users more control over the images inside a generated page. This can be improved in two ways: allowing users to manually add their own images, and allowing users to regenerate images with their own prompts.

Link URLs to Buttons

Another improvement in terms of user control, would be to let users link specific buttons to desired URLs and to be able to add their own forms.

Further Template Integration

Looking at the templates, it is recommended to implement a way to create, add and edit templates directly into the tool, as these currently need to manually be added on the backend.

Further Branding Integration

In terms of branding, there is still a lot that can be done to more easily let users input their brand style into the tool. For one, the ability to upload a CSS file or Tailwind config. Another one would be the ability to import a Storybook library. However, this will need modification of Storybook and you would likely need to create your own Storybook plugin that converts the components to HTML for this to make work properly. An even more hands free way would be to let an AI analyse the website of the user, then scan for the brand colors, tone of voice and typography to include in the generated page. However, this would require a lot of testing and development to make work properly.

Recommendations.

Color Picker

A relatively easier way for this would be to add a color picker for a variety of different elements like: primary, secondary, text, and background color. The same goes for typography, as it could be useful to let users upload their fonts that the tool can then use to generate a page. This would allow users to integrate their brand colors and fonts without using a template. However, keep in mind that this would result in unpredictable layouts that most companies likely want to prevent.

Modularity

To allow for easier further development of the tool, it is recommended to refactor the integrations of the tool into modules. Ideally, the tool would be fully modular and allowing users to pick their own modules. This way, you can theoretically have unlimited integrations without needing an interface with all the options. Instead, users can pick which modules they need, and the interface changes based on which modules are enabled.

Publish to CMS

In terms of integrations, it might be helpful for users to be able to publish pages directly into their own systems. For example, have the ability to publish directly to Magento or Storyblok. Another possibility would be to create an export functionality that can be used more widely, like a JSON file containing all the sections, images and content. However, as mentioned previously, this would also be a great thing to make modular. So that publish functionalities are added based on the modules enabled by the user.

Bulk Generation

To be able to use this tool on a bigger scale, it is recommended to look at the possibility of bulk generation of pages for different keywords or product variations. This way, a campaign can be launched even more quickly.

Add CMS Options

It was mentioned that it might be useful to add more CMS related features to the tool like metadata, slug editing, and OG images.

Page Goal Selector

Add an option that allows a user to select a specific goal of to page (for example sales, event registration, etc.), this might help the AI in creating more targeted content/pages.

Automatic Title

Currently when a new page is generated, the title of the page is "Untitled Page". It would be easier for users if this title was automatically generated by the AI as it already knows the context and content of the page.

2.3 Models

Update AI models

The models used for the development of the tool are GPT-4 and DALL-E 3. These both work well, however, these models are being improved every day. Each release will likely be better and better. It is advised, to test and evaluate new released OpenAI models in the tool. This can help get even better output without needing to refactor code as I made the OpenAI models hot swappable. For example, with the recent release of GPT-4o, the model is more capable, faster and even less expensive to use. These are great improvements without having to do much work to integrate them.

Experiment with new AI features

New models can also be used to expand the tool. GPT-4o for example, now has the ability to "see". Something that this could be used for is as previously mentioned to gather the brand colors of a website. Another thing could be to let GPT-4o look visually at the page it generated, it might then even better understand how to improve the page and more it look better.

Experiment with models from other creators

It is advisable to in the future to try to integrate different models from different creators. AI is becoming really big and new models are created very frequently. In the future, models by OpenAI might not be preferable anymore for the tool, and there might be better models on the market then. So it is advised to always keep experimenting with new models to get good insights in how the tool can be further improved.

Integrate multiple models

To give more flexibility to the users, you could also consider adding a variety of different models to the tool that users can pick from. Of course, only models that have been well tested and which have shown to be effective. This also opens up the possibility to generate a page concurrently using different models. A bit like the AI playground of Vercel, the user could then compare the different outputs and pick the one that is most fit for their use.

Recommendations.



2.4 Safety

During the project, I did an ethical analysis on the AI models used inside the tool (GPT-4 and DALL-E 3). Here was discovered that these AI models still have quite some problems regarding safety. In this analysis, I also proposed some solutions that I think would be good to also mention in this advice report.

First, it is recommended to integrate Guardrails AI. This is a Python framework that helps build more reliable AI applications. It works by changing the pipeline used for AI generation. Instead of the normal pipeline where the prompt goes straight to the model and the output is returned, the input first goes through a “guard” that checks/filters the content before its sent to the AI. Then the same happens to the output of the AI model, where it first goes through the “guard” pipeline and only returns if it passes all the checks.

Another recommendation is to always iterate on the system prompts of the application. These prompts tell the AI, how to read input, use the input, and what and how to output. These system prompts can help you shield against certain things. Another way to do this, is to change the safety settings inside of Azure where the AI models are deployed. Azure has already made a curated list of safety prompts/content filters and the amount of “safety” can be changed ranging from Low to High.

Lastly, it is recommended to implement automated testing for the tool. As new models updates are added to the tool, it's important to make sure that everything continues to work smoothly. Automated tests can help catch bugs and issues early on which saves time and effort in the long run. These tests can cover various aspects like:

Unit Tests: Test individual components or functions to make sure they work as expected.

Integration Tests: Test how different parts of the tool work together, especially when a new model is integrated.

Performance Tests: Test how the tool performs under different condition like generation speed and resource usage.

Conclusion.

2.

4 Conclusion

While the development of the tool has made great progression and proves the possibility of creating temporary theme pages using AI, there are still so many possibilities of making the tool even better. To further develop the tool, I recommended first implementing the improvements that better the user experience the most. Then you can start looking at expanding the tool further in terms of functionalities.

This advice report has laid out several recommendations to improve the project and make sure it stays on the right path. By focusing on improvements in the interface, functionalities, models and safety, I believe the project can become even more successful and useful for iO and its clients.

Overall, the goal of these recommendations is to make the tool more user friendly, flexible, and reliable. I hope this advice helps guide the project towards further success and implementation.