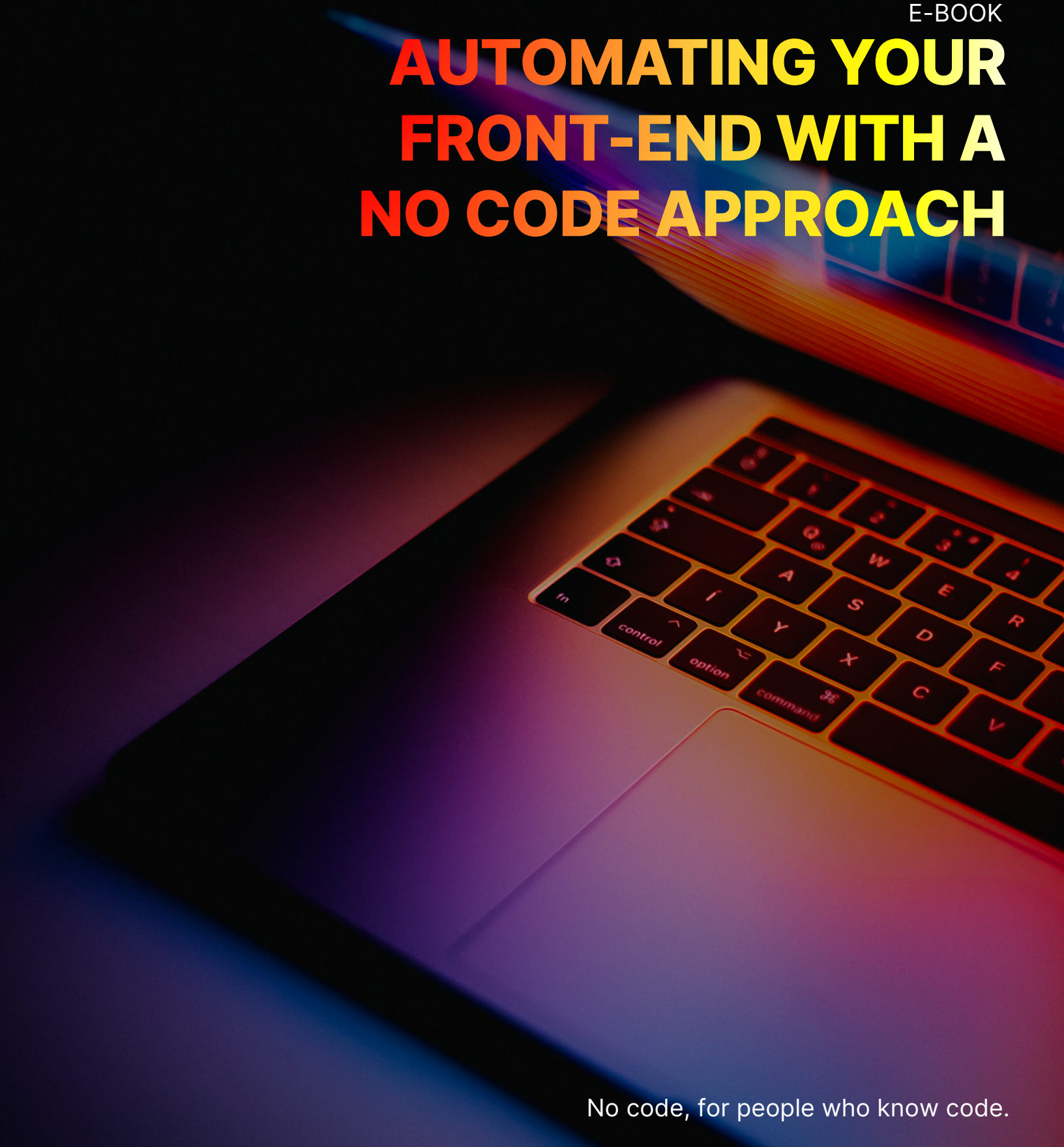
KLEEEN SOFTWARE

E-BOOK

AUTOMATING YOUR FRONT-END WITH A NO CODE APPROACH



No code, for people who know code.

Table of Contents

Introduction

A product's front-end may be the most important element of the user experience. It's the component users interact with to navigate a product and accomplish their tasks or goals, and it also gives a product its brand identity.

So what exactly is the front-end?

Essentially, the front-end is the code (typically the HTML, CSS, and JavaScript) that makes the product's interface work and keeps it running. This type of development is also known as client-side development, because it applies to everything the client, or the end-user, sees and experiences. People often mistakenly equate the front-end with the user interface, but the UI is just one part of the front-end. It also includes the infrastructure, data, and additional components required to ensure the product serves its function and its users' needs.

Front-end development most often utilizes one of three methods: a **manual** approach, a **low code** approach, or a **business intellgence (BI)** approach. While these approaches have proven successful, they are quickly becoming outdated and inefficient. The biggest issue with these approaches is that companies with exabytes of data aren't able to maximize how they leverage that data. In other words, businesses can't create the most effective product for their users because they're sinking so much

time and so many resources into developing and maintaing their front-end.

This is why you should consider two questions when creating your front-end:

1. What is the time to value?
2. How open is the stack?

Kleeen is a no code software designed to generate effective front-ends while offering fast time to value and an open stack. Our software uses AI to construct workflows, design UI, and generate production code, all while giving companies the flexibility to customize thier products as needed.

What is Front-End?

The front-end is the part of a product that users see and engage with. From the users' perspective, this means the product's UI. From the developers' perspective, this means the code and the tools used to keep the product up and running, as well as the data collected as users navigate the product.

Front-end development can be broken down into three main factors:

- **Components** – What goes into front-end development to create a working front-end?
- **Workflow** – Who are the stakeholders in developing and delivering a site's front-end?
- **Maintenance** – What happens after a front-end is deployed?

COMPONENTS

When it's broken down in the simplest terms, a front-end is made up of common features, application scaffolding, and API integrations.

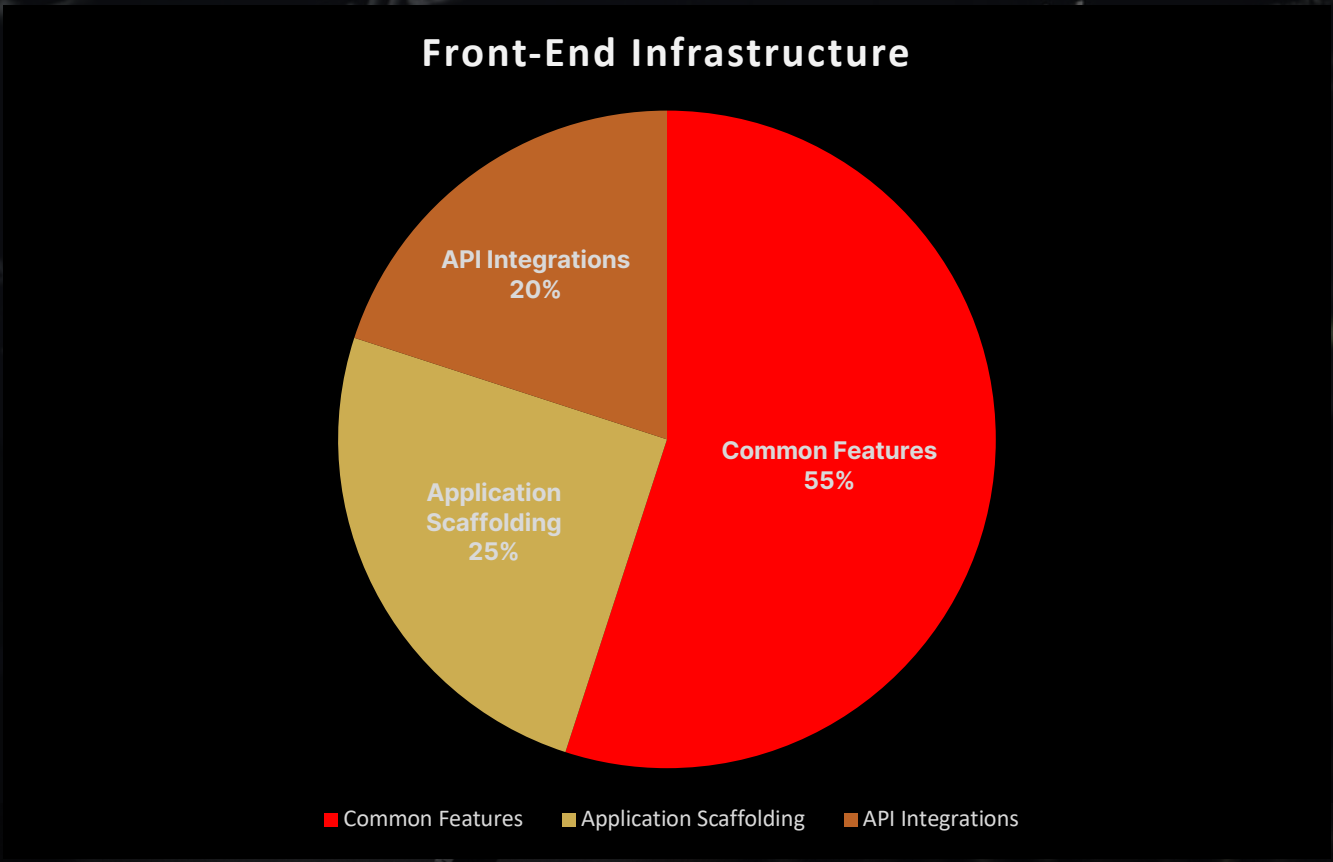
Common Features. These are the UI components users interact with directly, like navigation, dashboards, and buttons.

In traditional front-end development methods, UI designers and developers must hand-craft each feature and instruct it on how to behave (for example, telling the system what to do when users click **Save**). Specifying every interaction is a time-consuming investment for these teams.

Application Scaffolding. This is the piece that gives the front-end its functionality. In other words, it's the component that makes the front-end actually work. Application scaffolding includes the web application architecture, layouts, and state management.

The features control specific actions within the front-end, but application scaffolding tells all these features how to interact with each other. For example, consider a text field and a **Save** button. These are two separate features that may be connected through application scaffolding. So, when a user fills in the text field, the **Save** button is enabled, allowing the user to save their changes. Like defining the common features, application scaffolding requires significant time and resource investments from design and engineering teams.

API Integrations. This is the conversation between the front-end and the back-end, including data schema validation, API call validation, and data flow integration. Any time a back-end update is implemented, the front-end must also be updated to accommodate the change and ensure the site continues running as expected.



WORKFLOW

Crafting a front-end's components, application scaffolding, and API integrations requires the efforts and coordination of multiple teams. Product managers, UI designers, front-end developers and engineers, DevOps, and QA teams must all work in tandem to develop, test, and implement a product's front-end successfully. The development and overhead is a significant investment of time and resources, even if the process is smooth. Typically, however, the process is iterative, with bug fixes implemented and multiple rounds of QA performed. Ultimately, a traditional front-end development cycle can take months to deliver and depends heavily on unpredictable human factors. Often, this means front-end implementation can take up to 150% more time than the original project scope.

MAINTENANCE

Like the development process, maintaining a front-end is an iterative process. Each new feature or enhancement to an existing feature requires the efforts of all teams associated with the front-end, and may take weeks to develop, test, and roll out using traditional front-end development methods.

Bug tracking and fixes are also critical to maintaining the front-end. After its initial deployment, the product must be consistently monitored and maintained, and this continues for the lifetime of the product (or at least the lifetime of that product version).

Challenges with Traditional Front-End Development

One of the largest issues with traditional front-end development is that, since the initial workload to get the front-end up and running is so high, many data-intensive applications aren't able to harness all that user data they're collecting.

Without the bandwidth to perform a deep analysis on this data, product development becomes reactive instead of proactive and innovative. Having the ability to analyze product data and then act on it almost immediately gives businesses the edge they need to rise above competitors and provide intelligent solutions to their customers' problems.

To this end, front-end development can basically be broken down into two elements: the time to value and the openness of the stack.

Time to value (TTV). This is the amount of time it takes for customers to find value in a product. Think of it as the customers' "aha" moment or their return on investment for using a product. Ideally, this timeframe is as short as possible, because most customers are looking for fast solutions when using products.

Open Stack. This is the technology or solutions stack the product uses. It's the product's technological infrastructure, or the full list of services used to run the **Open stacks** allow companies to control their data and give access to the source code used to build their front-end, while **black box** solutions are closed in that companies do not have the flexibility to control their data or view their product's source code.

CHALLENGES WITH MANUAL, LOW CODE, AND BI DEVELOPMENT

Each tradditional method of front-end development comes with its own set of challenges.

- **Manual Creation**. A manual approach gives businesses the most control and flexibility in creating a customized front-end using an

open stack, but this comes at the cost of slow time to value, and without the benefit of a live prototype before deployment.

- **Low Code Platforms.** Low code platforms are appealing since they are less code-intensive and offer fast time to value, but this development approach comes with a number of constraints. It uses a black box solution that limits companies' control over their data, source code, and content.
- **Business Intelligence Tools.** BI tools let companies use working prototypes to test their front-end, but like the low code approach, they use a black box solution that restricts control over the data and source code. Embedded BI offers more flexibility than standard BI, allowing businesses to customize their content, UI, and branding, but both options provide slow time to value.

AI-based no code development approaches like Kleeen's offer the best of all worlds, with fast time to value, an open stack, and full access for companies to review their data and code as well as the ability to customize their front-end to their business needs.

Kleeen Automates Your Front-End Without Code

Imagine a solution that cuts development time significantly and allows front-end teams to focus their energy and resources on the bigger picture – creating not just a good product, but the best product on the market. Kleeen has the ability to automate up to 90% of business' UI and UX design. Likewise, application scaffolding and API integration requirements can be reduced to just 3% and 2%, respectively.

This level of automation means that teams can create and deliver an engaging, functional front-end in a fraction of the time traditional front-end development methods require. Businesses can reduce their implementation scope from months to just weeks, and this ease of use carries into the maintenance phase as well, with new features, enhancements, and bug fixes reduced from weeks of planning to just days (or even hours).

> This is the type of world Kleeen Software is making a reality. Taking larger amounts of the front-end stack off the plate of designers, engineers, and product managers.

Kleeen is specifically designed to support data-intensive applications, lightening the load of front-end development so teams can focus instead on mining and acting on the data they collect to build efficient, effective, user-centric products.

Kleeen is an action-based platform that ensures every feature put into a product is thoughtful and purposeful. To create a product with Kleeen, businesses input what they want customers to do and why. As a result, Kleeen isn't low code – it's a no code approach to creating front-end. Kleeen's AI constructs workflows, designs the UI, and generates production code ready for API integration. The software doesn't rely on mockups or previews to give companies a sense of what their product might look like – it actually builds a working prototype with simulated data based on customer specifications. Our efficient software lets businesses build and test a product in days or weeks, instead of months.

While the platform doesn't require development teams to perform any coding up front, products still needs code to run the front-end. Kleeen generates this code for based on business' input and checks the code into a GitHub repository that they have full control over, so internal teams can make any customizations or changes needed to the code.

Think of Kleeen as the most productive developer on your team. You have the flexibility to add custom widgets, pages, and branding, all in less time and with fewer resources. Kleeen frees up your teams and resources so you can focus on what's really important – solving your customers' problems faster and more creatively.

Conclusion

The front-end is arguably the most important piece of a product. It's the part users interact with and put data into, it's where users determine a product's time to value, and it's how companies can cultivate their brand through the product's design. It can also be one of the most timely components to roll out. Some of the more well-established front-end development methods force companies to choose between faster deployments and time to value or customization and control over the product's data and source code. Kleeen Software uses AI to offer a no code solution with fast time to value and an open stack that gives companies full control over their front-end data, source code, and customizations.

See the power of Kleeen for yourself with a free trial.

START TRIAL