Professional Portfolio

Process Book 2

MYDESIGN92

WEB DATA GRAPHICS ENGINE

JM Crawford

Professional Portfolio

Process Book 2

MYDESIGN92

WEB DATA GRAPHICS ENGINE

JM Crawford

Table of Contents

Thank you for taking the time to check out my portfolio! Connect with me on social media, or contact me directly through my personal website accessible through the link provided below.

www.johncrawforddesign.com

www.linkedin.com/in/john-michaelcrawford

https://www.behance.net/jmcrawforddesign

https://github.com/JMC818386

MYDESIGN99

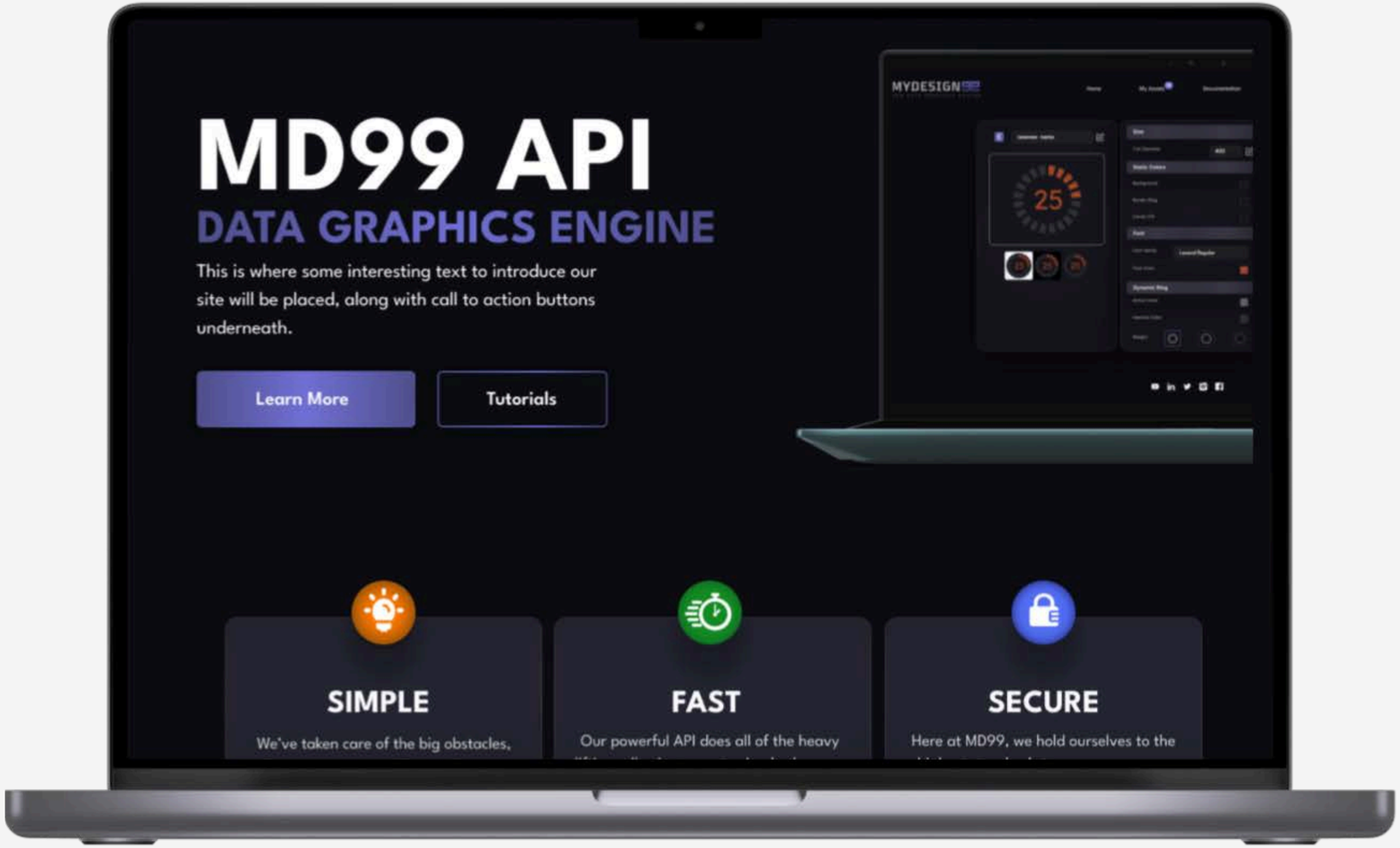
WEB DATA GRAPHICS ENGINE

Figma, Adobe Illustrator, VSCode, Github

HTML/CSS/JS + ReactJS

Concept

MD99 is a cutting-edge web-based platform designed to revolutionize the way developers create and deploy dynamic KPI graphics. It bridges the gap between data visualization and web development, offering a seamless way to generate, customize, and integrate performance metrics with minimal effort. Built with flexibility and efficiency in mind, MD99 enables both front-end and back-end developers to visualize key performance indicators instantly, without needing extensive design experience.



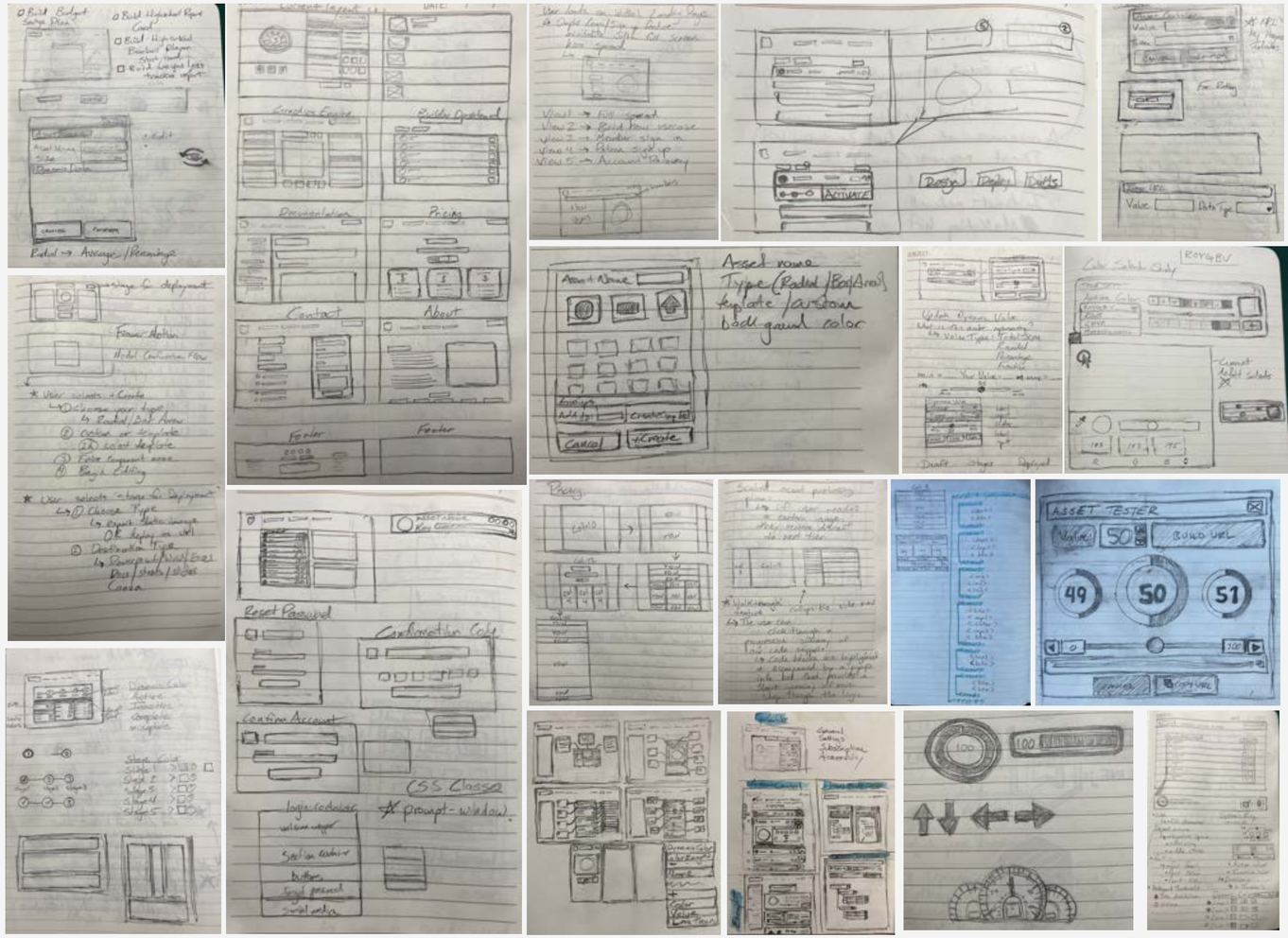
Project Conception

First Stage

MD99 began as a passion project between myself and a fellow colleague who I was working with on another project. We had been working with data visualizations, and wanted to see if we could build and use our own instead of relying on a third party service to serve them up to our web and native application pages. We wanted to focus on KPI (Key Performance Indicator) graphics, starting with a radial progress graphic as our test asset to begin attempting a proof of concept.

Goal

Our goal for this first stage of the project was to see if we could render our own custom dynamic KPI graphic to the browser page. This graphic would need to be able to re-render and change based on the number value that we sent with it.

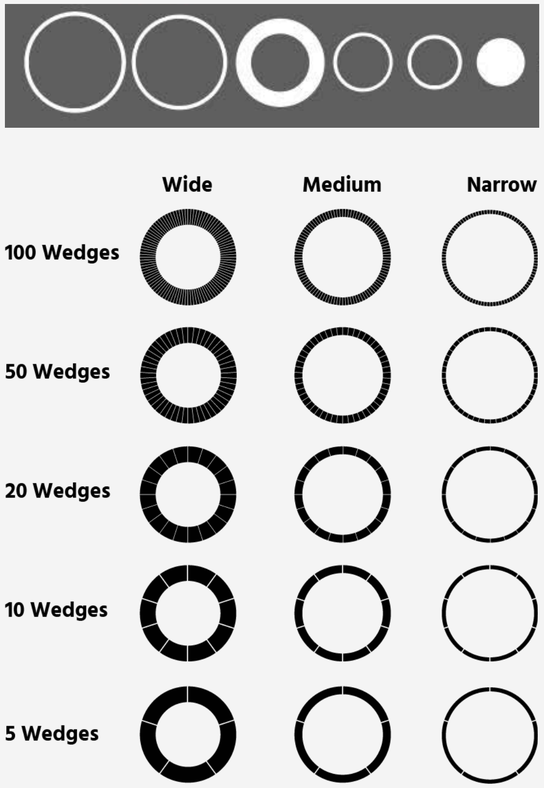


Radial Basic

The First KPI Asset

Our approach at this point could be best characterized as “vertical slicing” - where we would implement a basic design into a testing environment where we would ensure that what I was building on the front end of the website could successfully communicate with my colleagues backend servers, rendering a live dynamic KPI to the browser. For each step of this early design and development, we used this vertical slicing method to successfully debug and test each feature throughout the core functionality of the site.

To the right is an example of what I would deliver to my colleague - dividing each KPI into the different division amounts and line weights that would eventually become the different options for user’s to choose from while customizing their own assets. Each KPI that I created went through this process, ensuring that my colleague had every possible version of each asset for him to use to begin rendering them on our backend server.



Dynamic Ring Thickness + Divisions (full graphic)

Thickness: wide/medium/narrow
Divisions: 100/50/20/10/5

wide-100 wide-50 wide-20 wide-10 wide-5

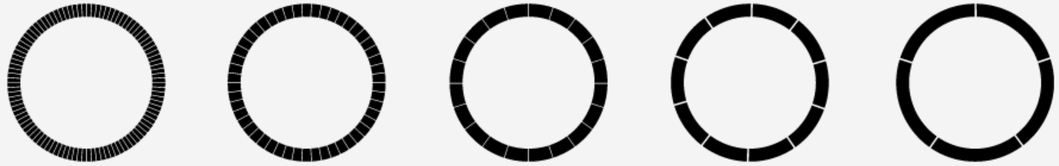
medium-100 medium-50 medium-20 medium-10 medium-5

narrow-100 narrow-50 narrow-20 narrow-10 narrow-5

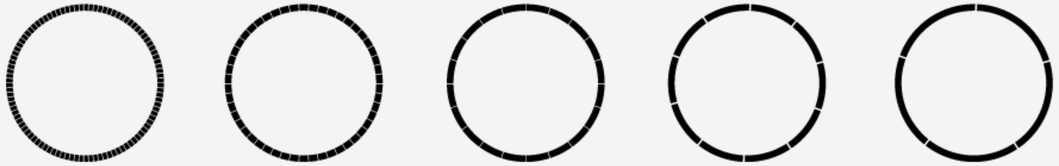
Dynamic Ring Thickness + Divisions (dynamic ring only)

Thickness: wide/medium/narrow
Divisions: 100/50/20/10/5

wide-100 wide-50 wide-20 wide-10 wide-5



medium-100 medium-50 medium-20 medium-10 medium-5



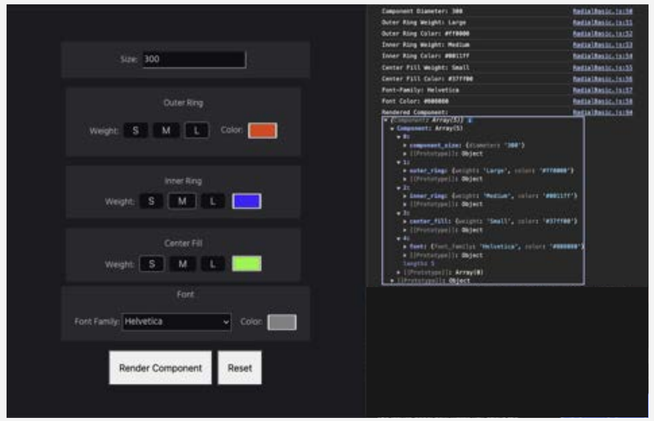
narrow-100 narrow-50 narrow-20 narrow-10 narrow-5

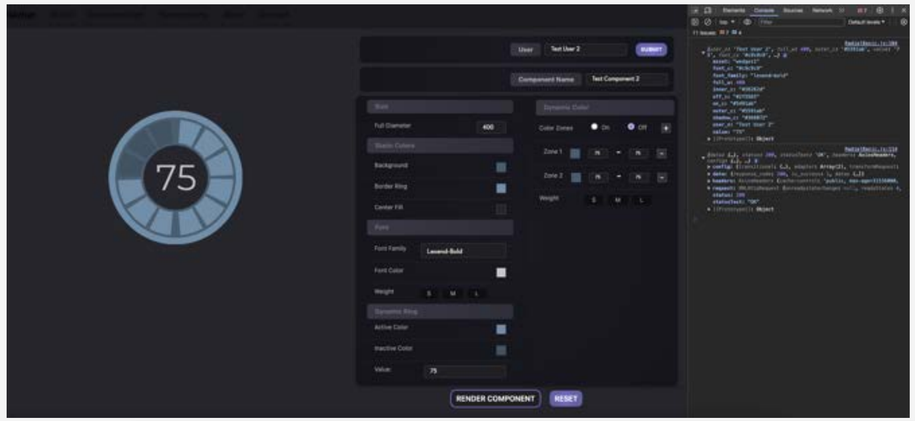
Proof of Concept

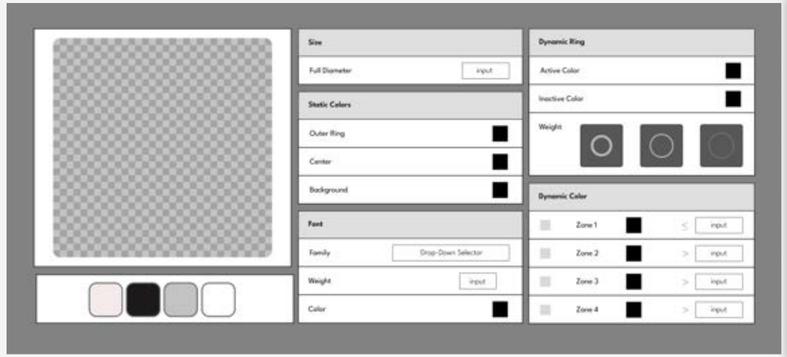
Building the Graphics Editor

The next step in this process was to construct a simple set of front end controls that would use stored state values to represent each option that the user could select, sending each selection as a key-value pair object through an API call to our backend. The main goal at this stage was to successfully demonstrate the core functionality of the graphics editor, allowing the user to select between different style options while seeing their custom KPI graphic update dynamically in the browser in real-time.

This is where I began working in figma to begin designing the refined graphics editor UI - building out the front end in stride with the successful demonstration of each feature. Taking this approach was both daunting and exciting, jumping between designing and coding, and seeing the site take form organically as we steadily made progress through the core functionality of the site.



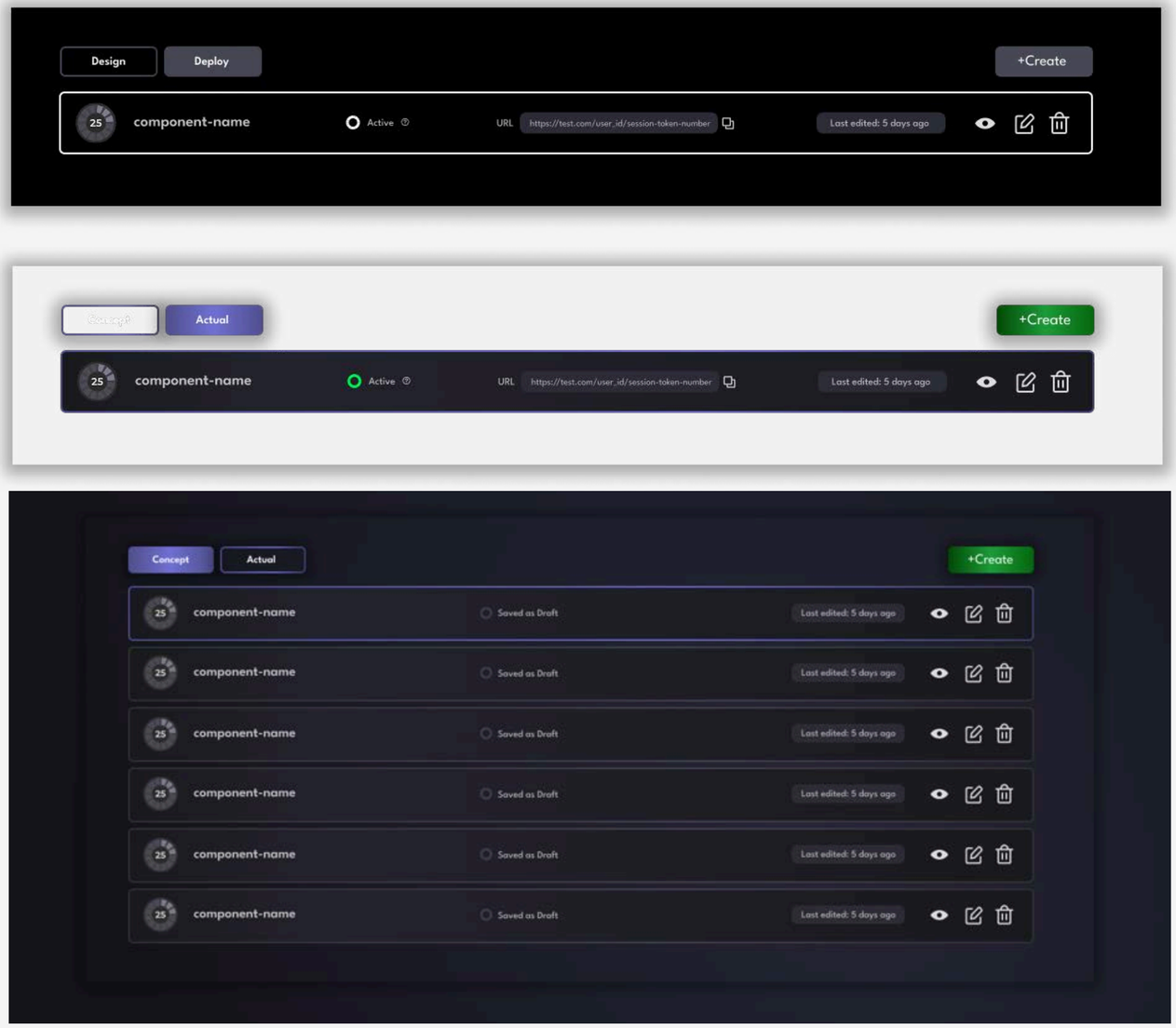




Asset Manager

Organizing Assets

Once we were able to successfully demonstrate a MVP version of the graphics editor, our next step was to construct a central location where the user would be able to create and manage all of their KPI assets. The design for this was pretty straight forward, displaying each of the user's created assets as a single element occupying it's own row, organized into a list, displaying the thumbnail, name, status, and view/edit/delete button from left to right. I wanted to leave a lot of negative space between groups of elements in each asset container so that when I began working on the responsive design there would be plenty of room for the contained elements to scale down to tablet and mobile widths without needing to hit too many breakpoints and major screen changes from one screen size to another.

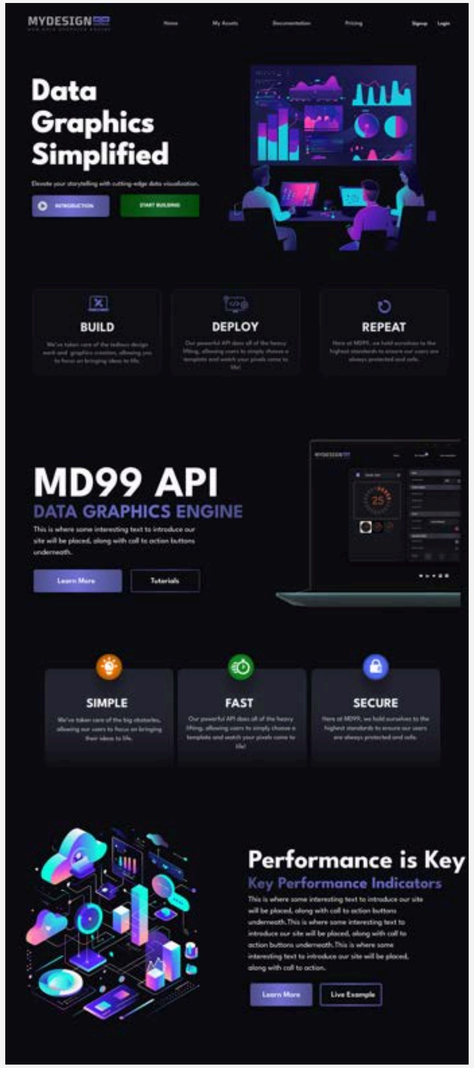


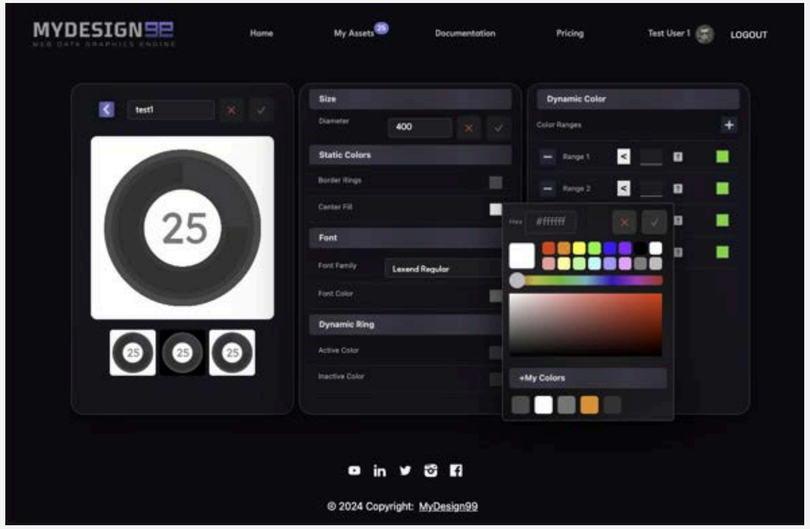
Site Design

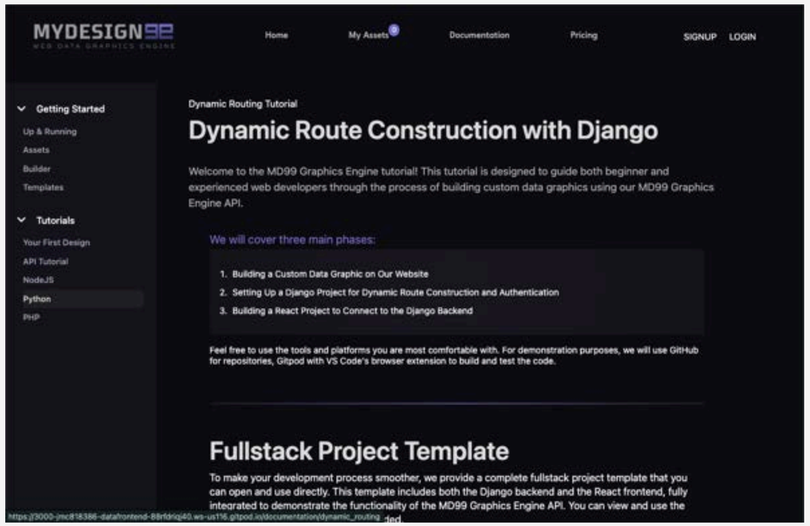
Building Backwards

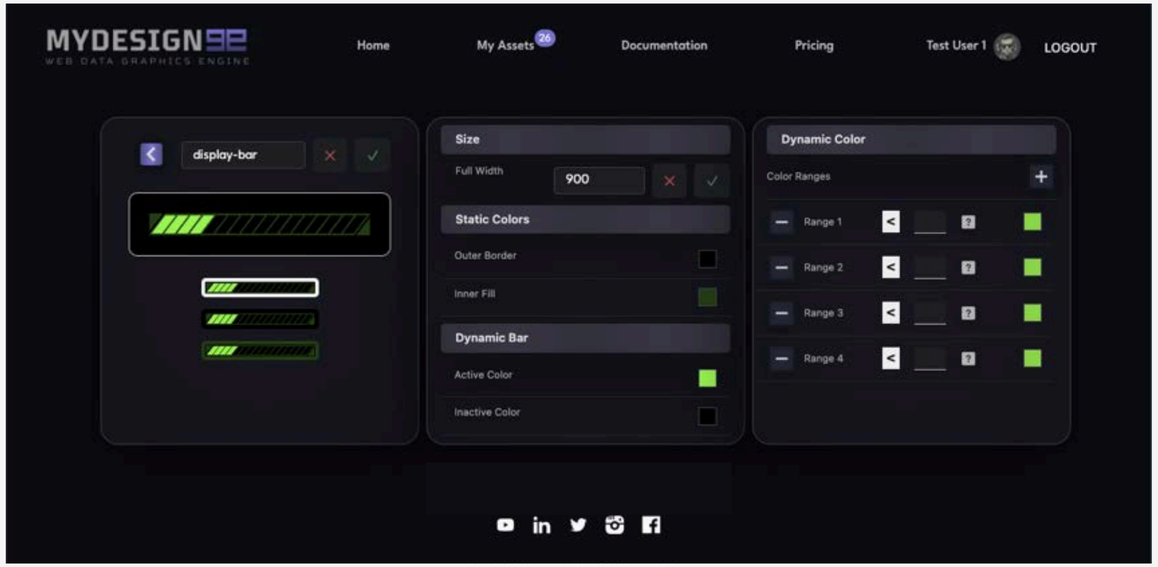
Once we were able to successfully demonstrate the core functionality of our two main features (graphics editor and asset manager), I began constructing the rest of the front end of the site - starting with the navigation and home page, then moving to documentation and pricing. My goal for the documentation was to make it as simple and easy to follow as possible, using indentation and typographic hierarchy to help guide the user from left to right and top to bottom through the documentation with each section clearly defined and organized in a way that would be easy for the user to work through and understand regardless of skill level.The challenge here was to design a documentation system that would be functional for both experienced developers as well as beginners.

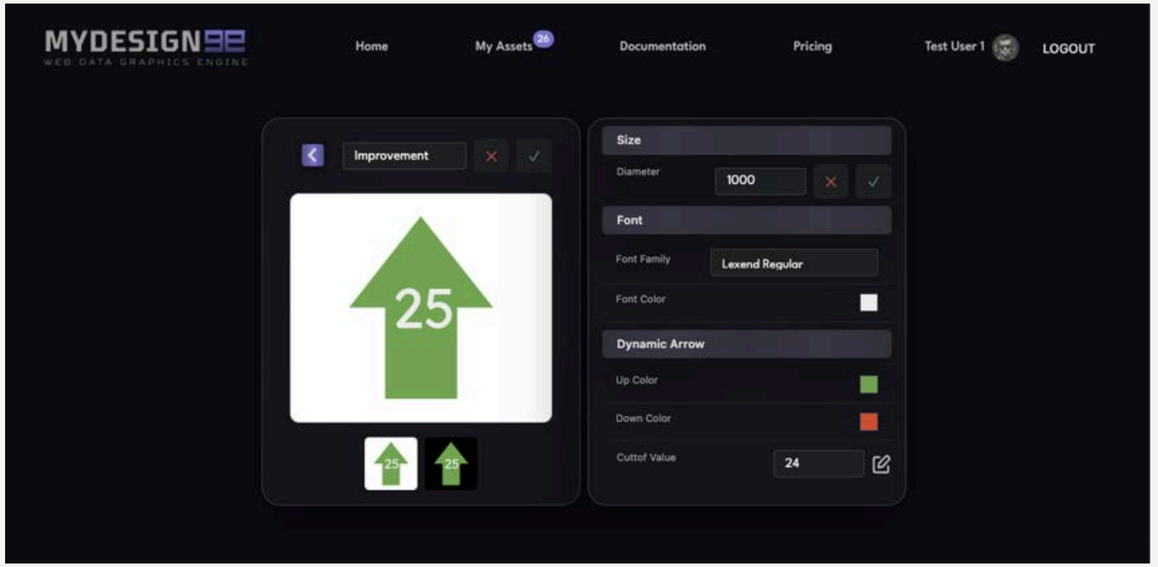
Once the general structure of the website was built, we began focusing on implementing the next 2 KPI types - a horizontal progress bar, and dynamic arrows. We designated these three first KPI's as radials, bars, and arrows - collectively serving as the foundation for a much more extensive collection of KPI's that would follow.



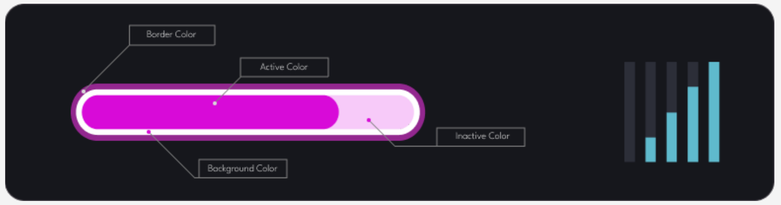





MYDESIGN99
WEB DATA GRAPHICS ENGINE
Home
My Assets 26
Documentation
Pricing
Test User 1
LOGOUT
display-bar
Size
Full Width
900
Static Colors
Outer Border
Inner Fill
Dynamic Bar
Active Color
Inactive Color
Dynamic Color
Color Ranges
Range 1
Range 2
Range 3
Range 4


MYDESIGN99
WEB DATA GRAPHICS ENGINE
Home
My Assets 26
Documentation
Pricing
Test User 1
LOGOUT
Improvement
25
Size
Diameter
1000
Font
Font Family
Lexend Regular
Font Color
Dynamic Arrow
Up Color
Down Color
Cuttof Value
24


Inactive Color
Active Color
75%
Background Color
Dynamic Value
25%
50%
75%

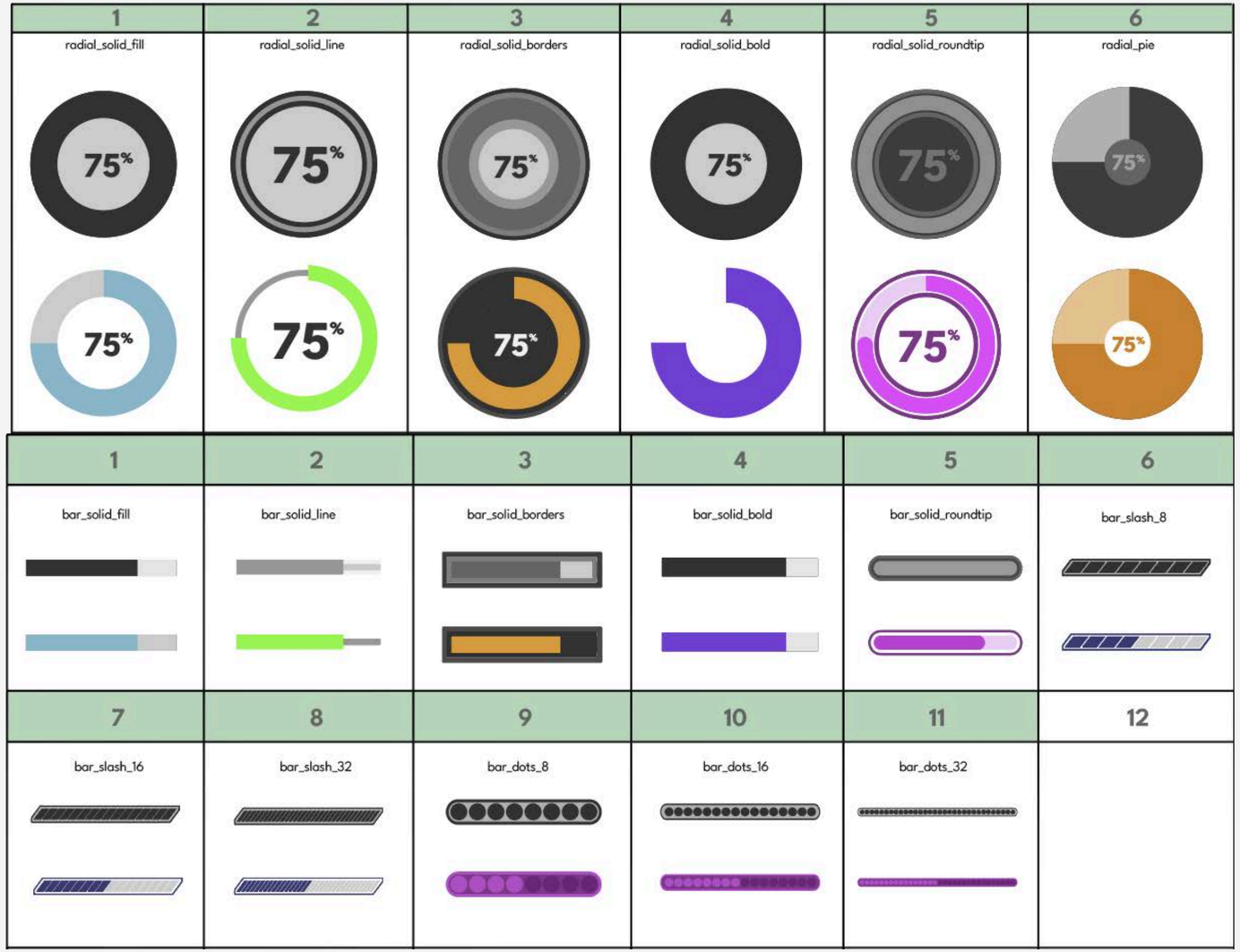

Border Color
Active Color
Inactive Color
Background Color


Up Color
Down Color
25
25
Dynamic Value
Positive Indicator
Negative Indicator
AAPL 48.91M 160.77 1.35%
MSFT 5K 61.25 1.98%

New Asset Designs

Approach

Following the same strategy as the first radial KPI, I began creating new variations of radials, bars and arrows to grow our asset library and begin establishing the different tiers that users would be able to access - each tier unlocking more assets and more features. My focus at this stage was to ensure that I didn't stray too far from the JSON structure we had established with our first assets, making the construction of this next set much easier to integrate into our existing source code without having to reinvent a unique object structure for each new design. In the long-run, following this convention has allowed us to create and integrate new designs with relative ease.

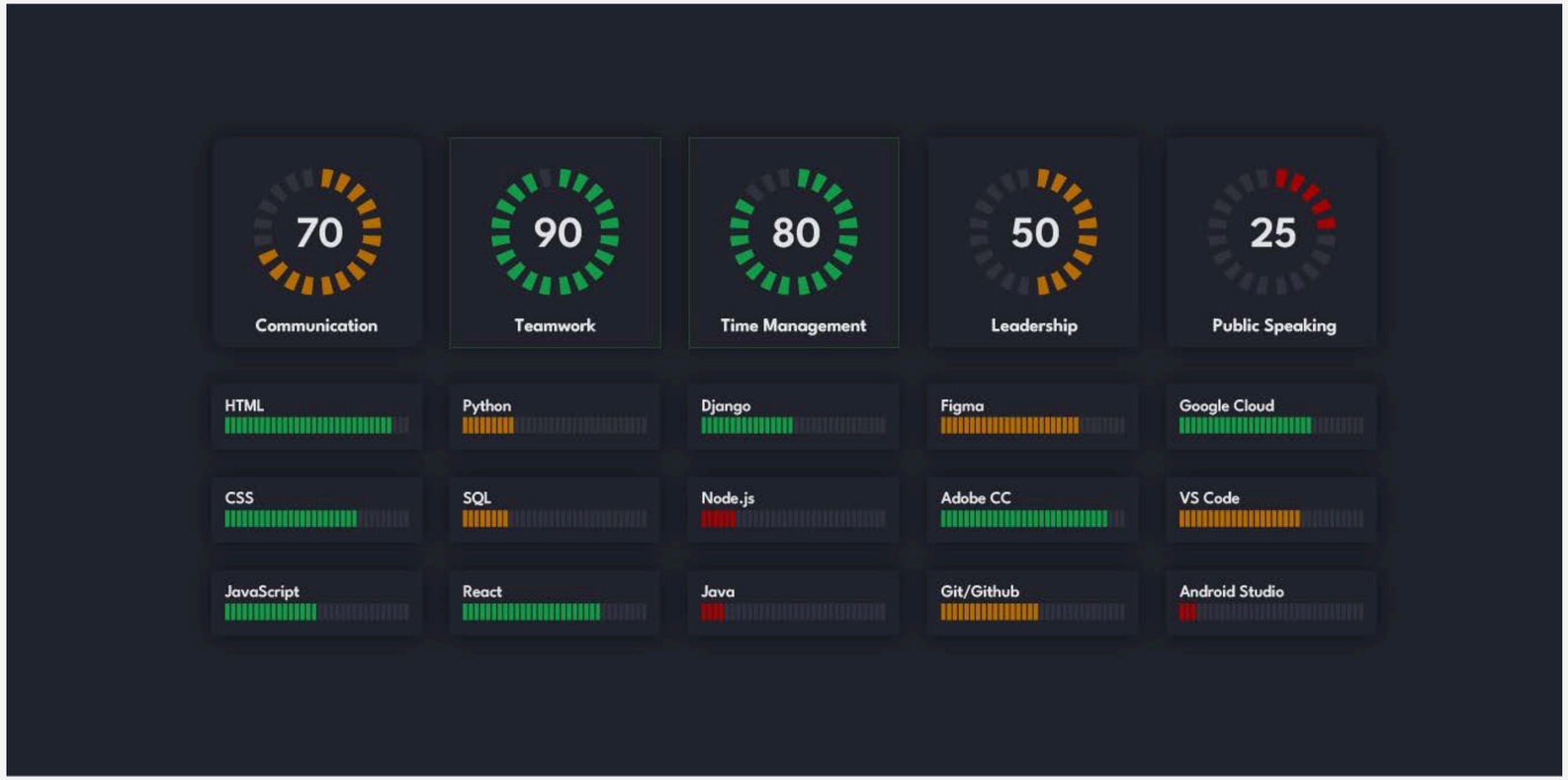


First Look

Rendering Live Assets

Our first live examples aim to demonstrate the way our technology can be used, showing a combination of different KPI types to provide a quick at-a-glance view of complex and detailed data displays. In this example we are using a React.js front end to display a collection of attributes that reflect the skills and abilities of an individual developer. This example uses specific colors that can be set by value ranges to reflect high, middle, and low score ratings, helping to make it instantly apparent which areas need improvement in contrasted to the areas that show a higher ability.

The key take-away from this example is that it shows the API rendering multiple KPI's of different types to the browser based on unique data stored on our backend - allowing users to connect their projects to a data base and then render dynamic value-based KPI graphics instantly with minimal set-up and coding.

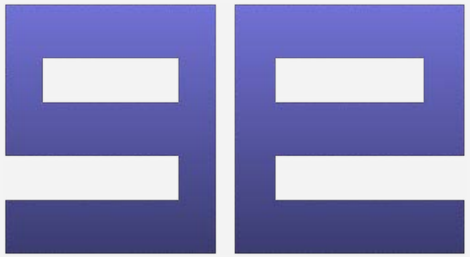


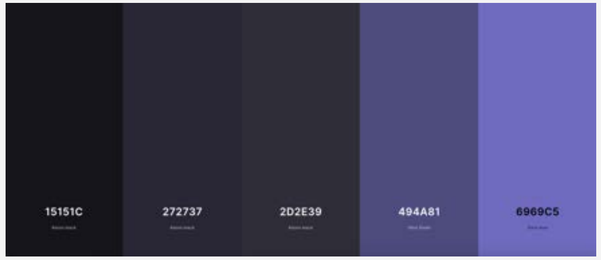
Conclusion

Initial Design

With our approach to rapidly develop with platform through experimental iterations, the brand and style aesthetic has evolved as the project has progressed. The challenge for this brand was to find a way to incorporate the “99” into the brand-mark in a simple, and meaningful way. The reflected “99” is a nod to my colleague and myself both working together on completely different ends of the project, working entirely remotely. The 99 can be seen as two individuals positioned back-to-back, each working on a keyboard. The 99 is also a nod to the developer community, revealing that the original name that my colleague wanted was an available domain name, so we added 99 to the end to ensure that we had a unique name. There is also a rhythmic quality and rhyme to the name when spoken out-loud (my-desIgn-nInety-nIne), making it memorable and giving it some character.

This project has been an incredibly valuable experience, being able to work side-by-side with an experienced senior software engineer, challenging my abilities and providing me with hands-on experience working on both design and code in a rapid-development project environment. This project has been instrumental in my growth as a designer and developer.





www.johncrawforddesign.com

Thank you for taking the time to check out my process book for Clover! You can connect with me on social media, or contact me directly through my personal website accessible through the link provided below.

www.johncrawforddesign.com

www.linkedin.com/in/john-michaelcrawford

https://www.behance.net/jmcrawforddesign

https://github.com/JMC818386