**Responsive Web Design Development**

ASSESSMENT 2: Multi-Device Application

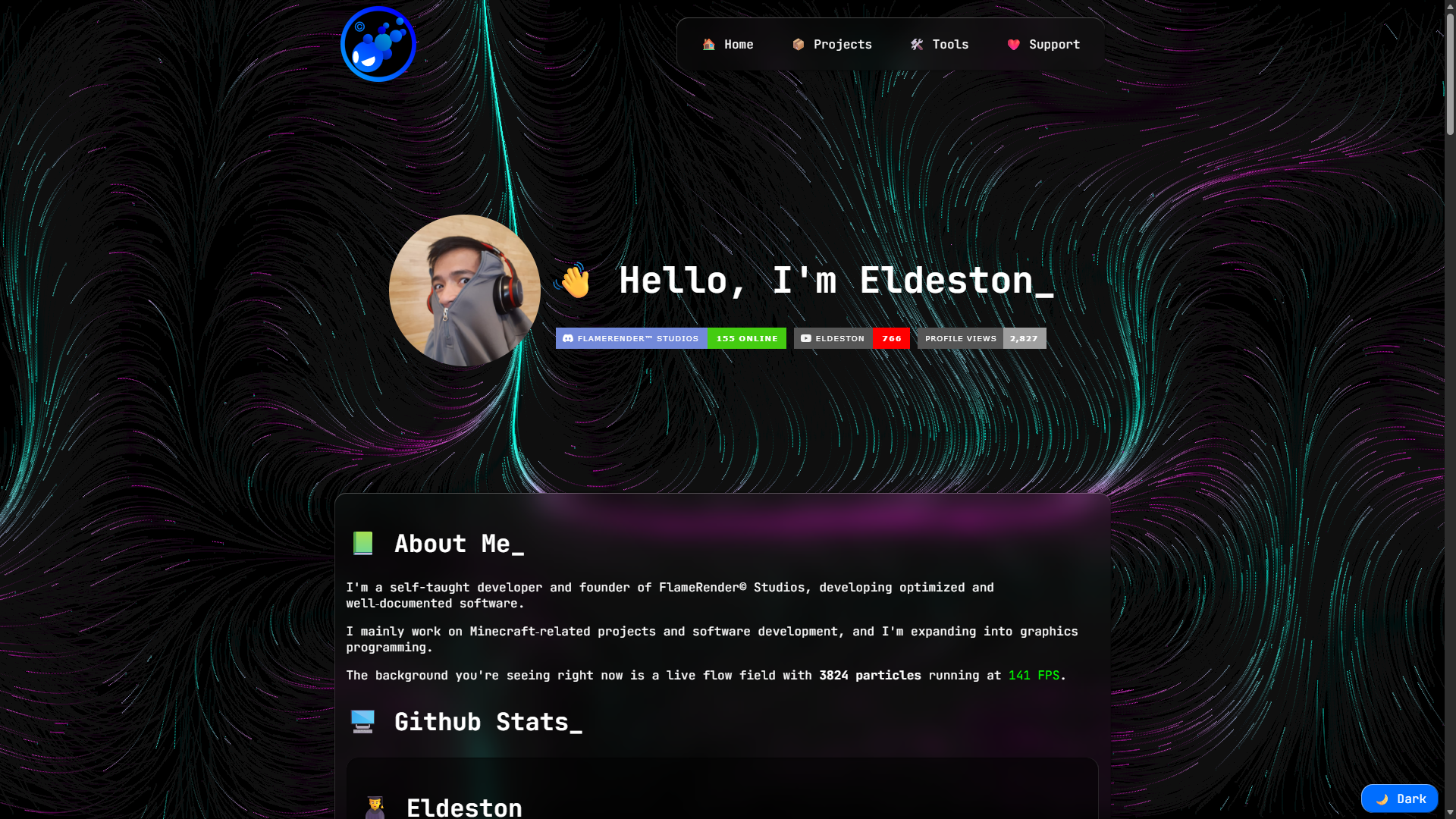
|  |  |
| --- | --- |
| **Student Name** | Frem Fredric Baring Patalinghug |
| **Assignment Brief** | My portfolio portal web application is a web application made with Node.js used to display my skills as a programmer.  It includes Github and Discord APIs to display my statistics and some fancy p5.js graphics. |
| **Github Repository Link** | [Eldeston/Personal-Website-Application](https://github.com/Eldeston/Personal-Website-Application) |
| **Walkthrough Video Link** | [Web Application Showcase - Responsive Web | Creative Computing L5](https://www.youtube.com/watch?v=_SFinxq5cqY) |
| **Repository Screen Shot** |  |

**DEVELOPMENT DOCUMENT**

**BRIEF**

My multi-device web application is a portfolio portal to display all my projects and skills as a web and graphics developer. This portal is also an upgrade to my previous static web portfolio hosted on Github pages as the upgraded web application now uses Node.js and is deployed on Vercel, a hosting service for deploying web applications.

It displays my Github and Discord server statistics using an API in the back end and a p5.js-powered background in the front end. It also features a forum system that users can write and submit comments that can be highlighted later. It targets mobile, tablet, and desktops and optimization is carefully considered for each device.



Web application screenshot, also a portfolio portal.

**DEVELOPMENT BREAKDOWN**

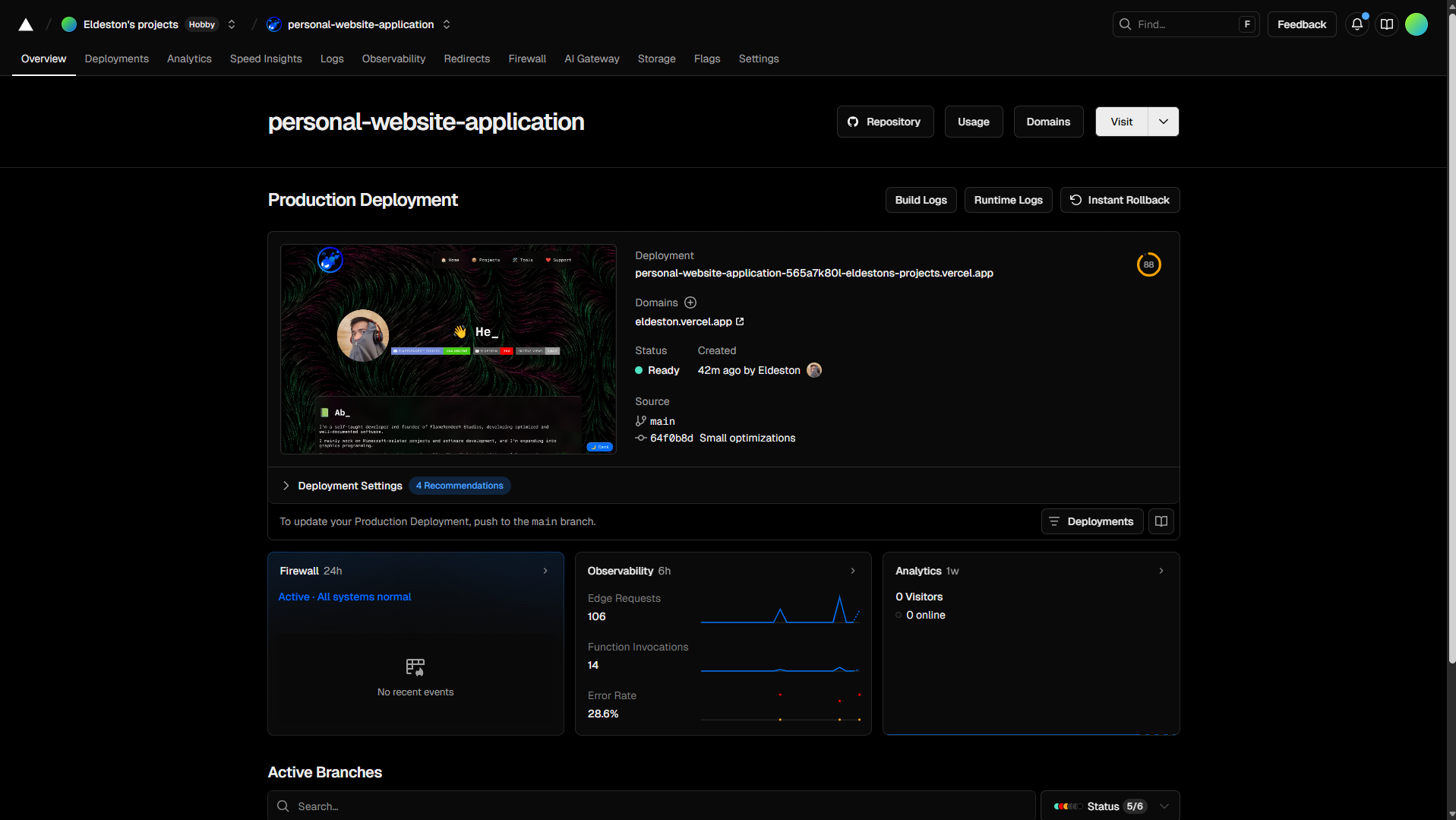
Along with taking the previous static web portfolio as our base, the steps to making this web application are as follows:

* Build the web application on Node.js.
  + Utilize Github API to fetch and display user statistics.
  + Utilize Discord API to fetch and display Discord server statistics.
* Write the portfolio contents in a single HTML page with appropriate semantics.
* Utilize Syntactically Awesome Style Sheets for styling the website.
  + Style the page with mobile-first approach in mind for responsiveness.
* Improvise background graphics and improve performance on various devices.
  + Abandon p5.js and write a custom background graphics script that draws on the canvas.

**WEB APPLICATION SETUP**

The web application is built on Node.js and requires the following packages:

* Express : For the web application’s framework.
* Dotenv : For managing environment variables, useful for managing and hiding private tokens.
* Octokit (REST) : For accessing Github’s REST API for fetching user and repository statistics.
* Discord.js : For accessing Discord’s API used to create discord bots, which is also used to fetch Discord server statistics.
* Vercel (Speed Insights) : For web application statistics to be viewed on Vercel’s web application panel (more on this later).
* SASS : For styling with SCSS (mostly used client side).
* MongoDB : For a simple forum system to submit comments.



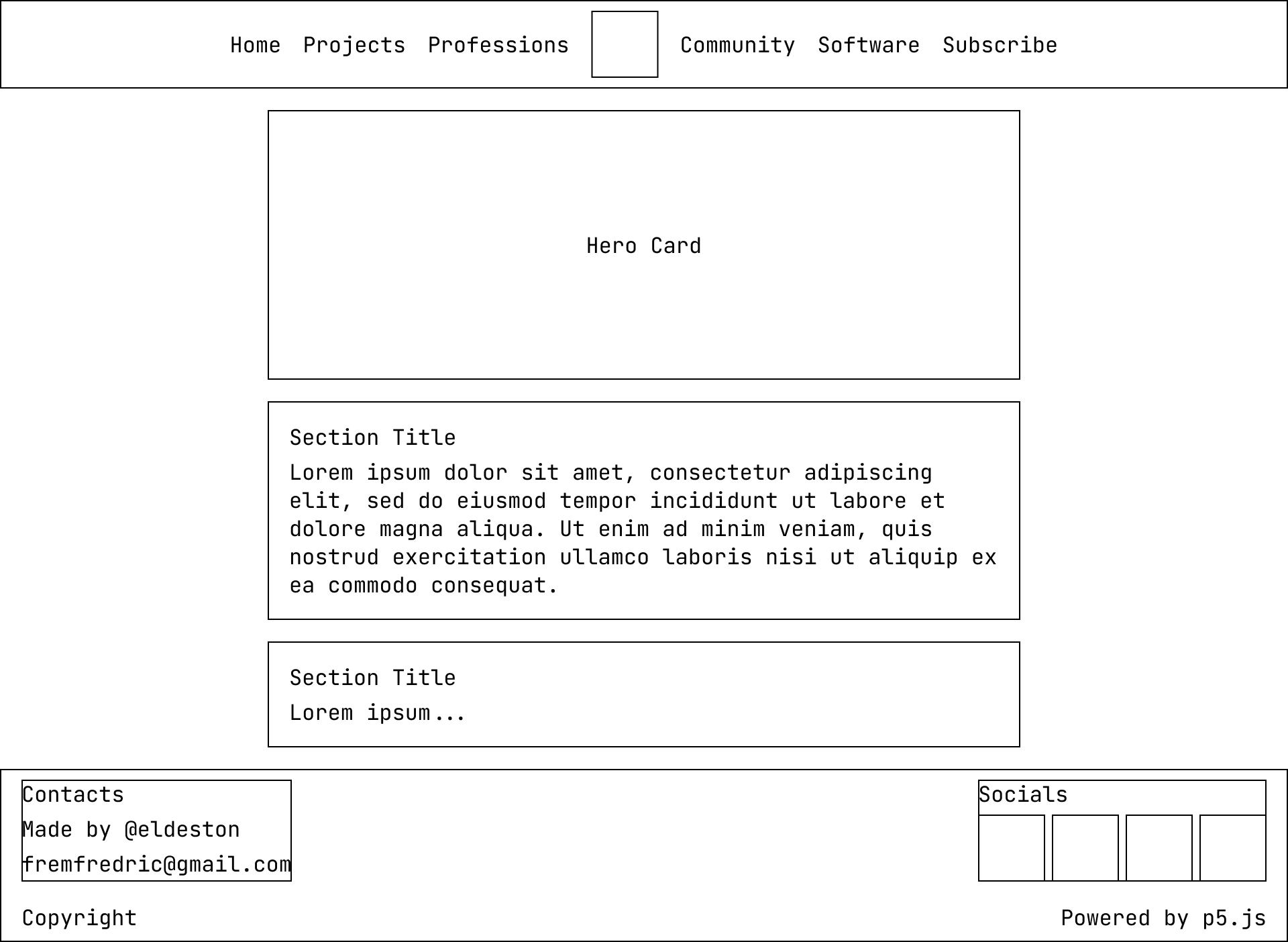
The project panel for this web application on Vercel.

Additionally, this web application is deployed on Vercel, a free web application hosting service, to demonstrate the web application’s practical usage. The web application is also initially uploaded to Github for Vercel to source from and deploy the web application. Vercel also lets you manage private tokens and view web application statistics.

The private tokens used for this web application are stored in a .env file. This is to avoid security breaches and is excluded from this web application’s the Github repository.

Vercel also lets you manage and connect MongoDB to your web application as an integration. Vercel will automatically create a database for you through the MongoDB Atlas integration and you will be provided a private token to access the database.

**WIREFRAME LAYOUT**

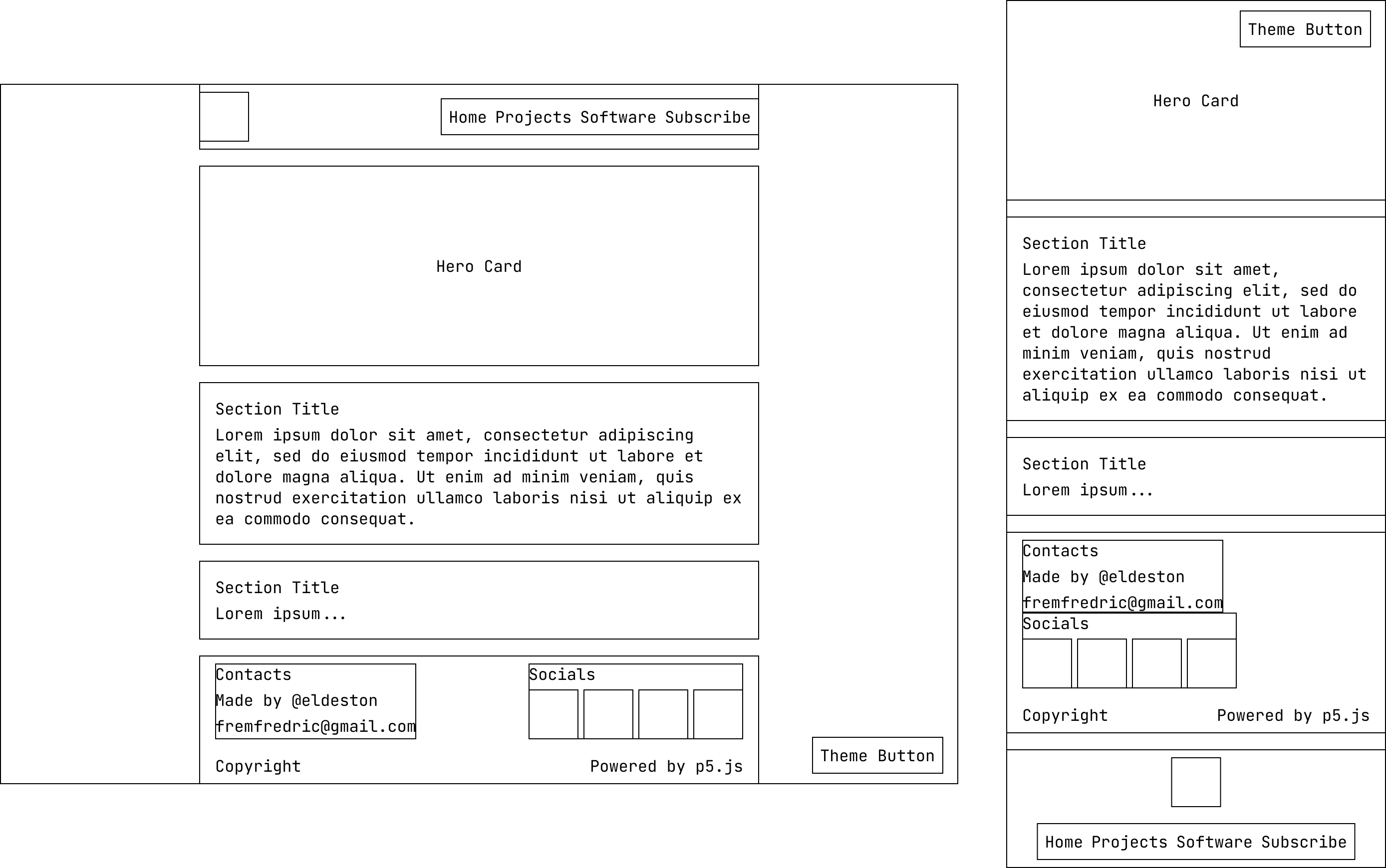


Previous iteration of my web portfolio as a wireframe from <https://eldeston.github.io/>

As mentioned earlier, this web application is an upgraded iteration of my previous website portfolio hosted on Github. The older layout is initially built with a desktop-first approach in mind and failed to be responsive on smaller screens.

The new page layout, while it technically still does a desktop-first approach since it is uses the old layout as inspiration, has a much cleaner and more modest design. The margin and padding for this layout is also more generous and consistent.

The page’s structure is divided into 3 main parts: a header, main, and footer. The new layout now properly uses HTML semantics such as the <header> to contain <nav> and logo, <ul> to list down navigation links, <main> for containing key contents, <section> for dividing contents, <article> for containing related contents, and <footer> to contain other additional information such as my email and my social links, and this page’s copyright information.



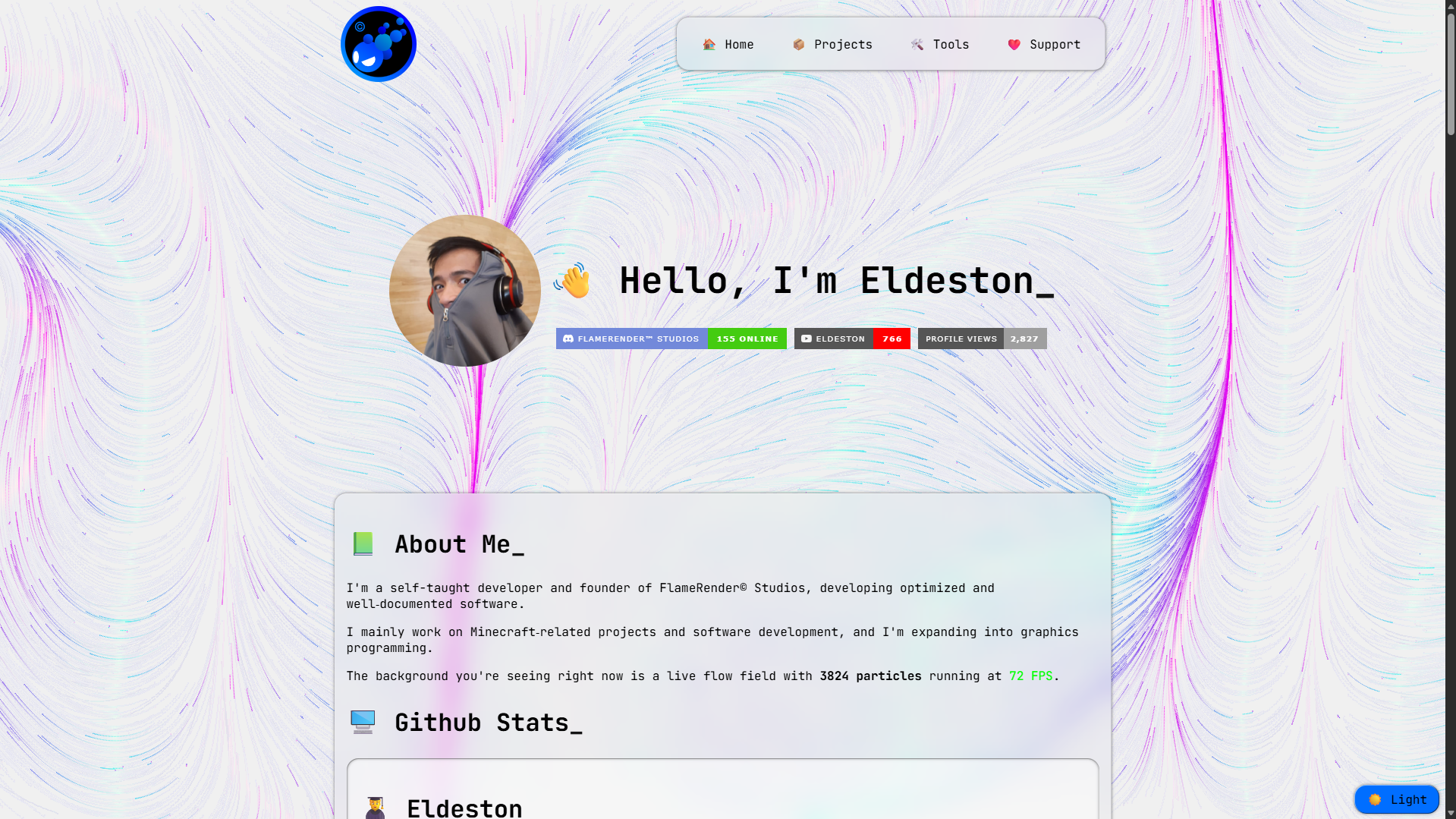
Web application’s desktop and mobile layout

The mobile layout has the header stick to the bottom of the screen instead. This is to give the user a different experience when on mobile devices and make it feel more like an application on your mobile. This is also to give more immersion since mobile screens can only offer so much space.

One additional change that is added on top of the previous iteration’s content is the forums section. Users who visit the website can leave comments that I can later read and manage inside the MongoDB Atlas panel. For security reasons, only highlighted comments will be displayed on the web application.

The page now also uses SCSS (Sassy Cascading Style Sheets) for styling and creating more modular code and reduce repeated style rules. This is later compiled into CSS for the page to use.

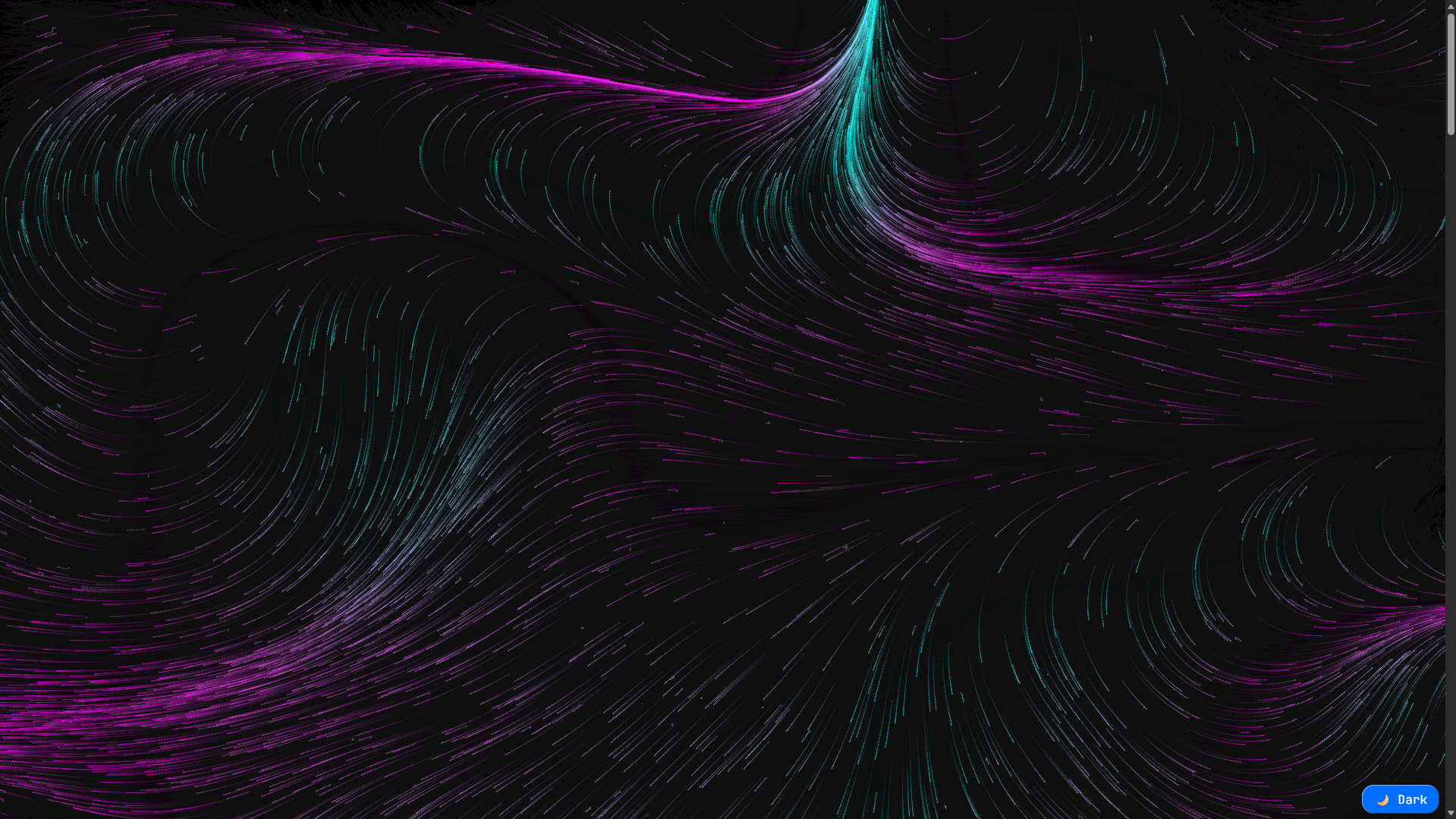
Lastly, the page starts in dark mode as it was in the previous iteration. The application additionally features a light mode that can be toggled via the theme button. Light mode mostly changes the element’s color, however for some elements such as the animated background (more on that later) and the white icons from CurseForge, X (formerly Twitter), and Github are instead using the inverted filter.



Light mode version of the web application.

**CANVAS GRAPHICS**

The page features an animated background to display my skills as a graphics programmer. The previous iteration of my web portfolio used p5.js to power the graphics. This time, the web application’s page does not use p5.js but instead uses a custom script utilizing the canvas element as a background. The reason for this change is to bring in more optimization and performance especially for weaker and outdated devices.



Standalone screenshot of the animated background.

The background features a simulation of a flow field of particles using value noise to change the direction and flow of the particles. The number of particles present in the simulation depends on the maximum length of the device for optimization. Additionally, this information is displayed in the “About Me” section of the page as well as an FPS counter of the simulation.

Finally, the background can be interacted with by the user. Clicking on the background will cause all the other elements except the background and the theme button to be invisible.

**Critical Reflection (400 words~)**

Many improvements have been implemented into this application as the upgrade iteration of my last work. Regrettably however, some of the features from the previous iteration and planned features have not been implemented due to the frequent festivities at the time of this development.

This project, however, did reap results in my learning and gave me ideas on how a web application works. The web application needs to run on a server to host the application live. Technically it can run on my local device but not forever.

Additionally, thanks to my prior experience in developing a Discord bot with Node.js, implementing Discord.js’ API was not too hard. However, since Discord.js is a package for running Discord bot applications, it means the Discord bot will have to stay active for as long as the web application is active. It may also pose security risks to my Discord server as the API in my web application can and will interact with the server as a Discord bot. Something I will have to look out for in the future.

Speaking of security, the forum section is prone to spam and other cyber security issues. There is no request limit and the post highlight system is not ideal for filtering posts.

Prior to this project, I was also working on my Python application. The concept of the project is similar as both applications use an API. Much of the knowledge I gained from making my Python application is transferred and applied here in this application.

The design process for the website is also an improvement compared to the previous iteration. It has improved in responsiveness and layout for smaller devices. I also learned that desktop and mobile screens should always have a completely different experience because of the fact that desktops start landscape and have bigger screens, and mobile devices in portrait with smaller screens. The styling process for the website also has become more modular thanks to SCSS. SCSS brought familiar tools to CSS such as functions and mixins to make code more modular.

Overall, this web application not only served as a practical web portfolio, but also gave me insights on how to develop an application and deploy a web application in a web application hosting service like Vercel. This development gave me insights on frameworks other than Node.js that may provide more flexibility or features.

**Appendix**

* The Web Application’s Main GitHub Repository & Vercel Link :
  + <https://github.com/Eldeston/Personal-Website-Application>
  + <https://eldeston.vercel.app/>
* The GitHub Web Portfolio’s (Previous Iteration) Github Repository & Github.io Link :
  + <https://github.com/Eldeston/eldeston.github.io>
  + <https://eldeston.github.io/>