Project: Eravate!

「An educational project on stellar objects」

Developed by Eravate.

For educational purposes only.

Index – Where to find what.

Quick introduction – A few words from me.

The main goal behind this project was to create a space that would serve as a repertoire of known stellar objects, as well as something that could be used for educational purposes, while making it feel like actual care and hard work were put into it.

I have spent more time than I anticipated on this, and to say it is finished would be, to a certain degree, an overstatement. But I have reached a point at which I can say that what I currently have is something somewhat worthy of publication.

The first thing you will see is how much this project differs from every other one, from the code to the documentation, and that is because I decided to take a completely different approach, one that would, at least in my opinion, make more sense in the real world, and one that would make the final product feel more “realistic”.

It is also a more direct approach to how an application should look, instead of the basic navigation bar and some interactive bits that almost every website has, this project has more of a desktop application feel to it, which is what I was aiming for since the very beginning.

The reason I decided to centre my application around space is the fact that I consider that it is something that we do not give the justice it deserves in the educational field, and it is something that could be especially important in the near future.

I always felt the lack of easy to understand, quick to access information about stellar objects from the general sources that we use, it usually is either too complex for a normal person to understand or takes too much time to get to that information.

And, in the end, it is not like I have a passion for many things, nor I have that much passion for space, but with this I could at least keep some motivation to work on this project and to make sure it turned into something decent.

Requirements & Reasoning - A quick look at the app.

With this project, there was no clear client or employer to work with, so all the requirements have been set by me, the first of which, and the most important one, is to make sure that the interface is as basic as possible, to ensure that the end user is never overwhelmed with information and can easily navigate throughout the entire application.

My focus was mostly set on ensuring that the main theme of the app, space, was present in most of its parts, and making the design consistent with it, from the background stars to the moving nebula in log in, from the input design to the colour scheme.

One thing about working with 3D Objects is the fact that they are not easy to render at all, they require more than the standard computational power to run smoothly. Still, one of the requirements set from the beginning was to make the objects as optimized as possible, so that the app can be accessed from as many devices as possible.

The reason I decided to exclude mobile devices out of this project at the start was that their screen would not be big enough to fit even half the app's information, and I decided to stick to a complete and full computer experience. But after some changes, the app can be accessed via mobile devices if desired, although not fully optimized.

On the other hand, the reason I decided to make this app purely in English is the fact that I want it to serve a wider audience. The main goal of this app is for its users to not feel excluded, and I feel like the use of any other language would achieve the contrary.

And the last requirement was to make the app self-sufficient, or as I like to call it, make sure the app runs smoothly without an admin looking over it, there are several techniques I used within the app to make sure that this requirement is met.

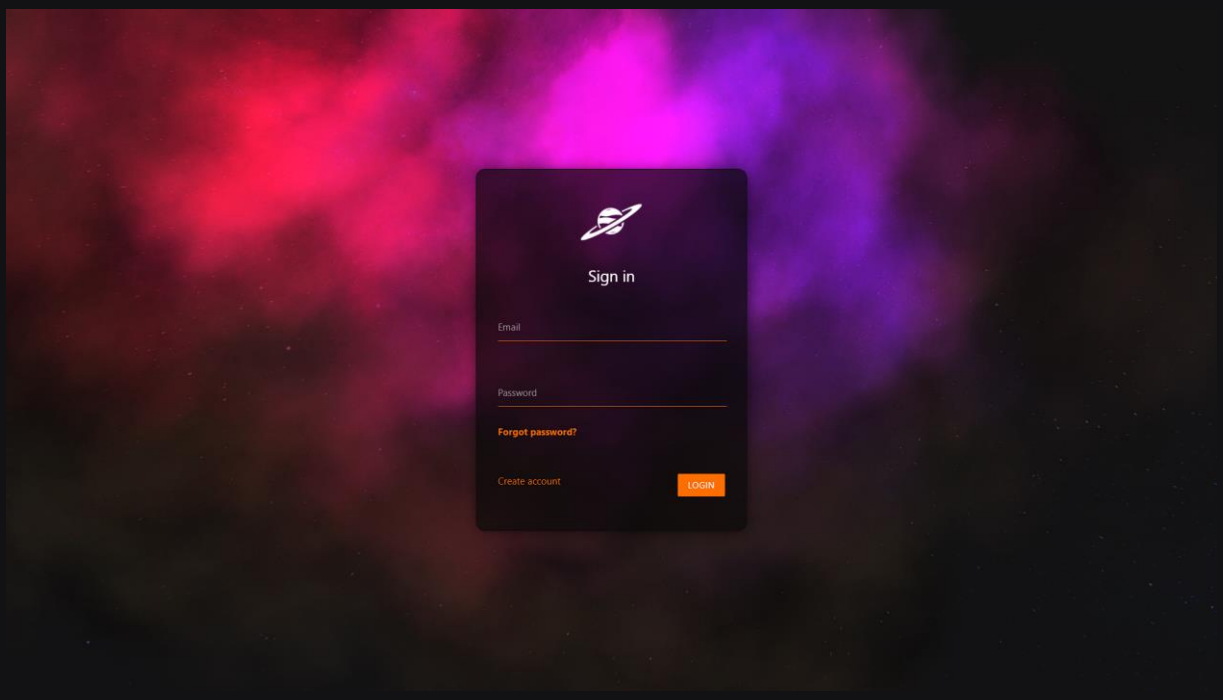
Everything that will be shown in the next few pages uses the final design as a reference - to see how the app looked in the preliminary stages, refer to the next part of the document: Planification & Development on page 8.

Log In - First impressions are the most important.

As established before, the space theme had to be present in most parts of the app, and that includes what is its most important part when it comes to impressions, since it is the landing page of our app, and it is the first page that most users will see.

I decided to use a procedurally generated nebula (each time it is loaded, the composition is different) in colours that would fit the design the most - A mix of Purple, Blue and Red, with a considerably basic window to fill in in order to log-in, reset password or create an account.

All of it is topped off with the use of Sweet Alerts to ensure the user knows what they are doing at all times and does not get lost in the process.



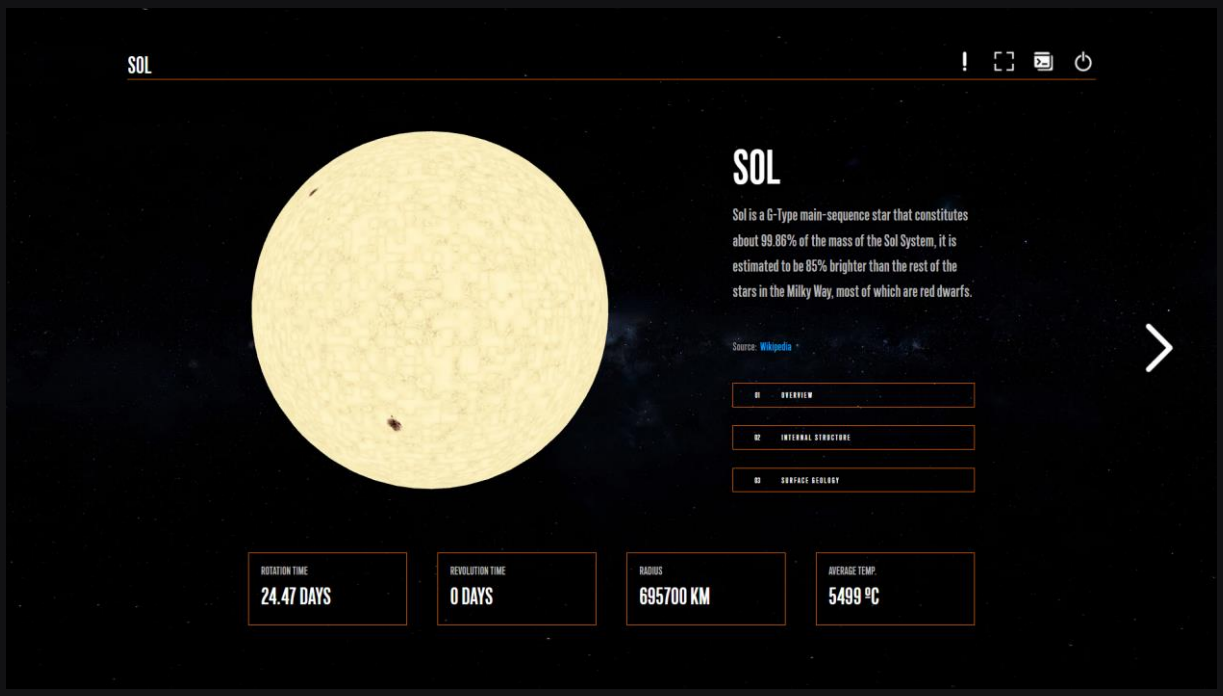
The log in site in full view.

To ensure that the right user resets the password, a confirmation is sent to their email address with a token, and only with that token will the user be able to reset their password, which is similar to the way the account activation process works.

Main – Here the magic happens!

After logging into our account, we are greeted by the main page and the main star in our system, Sol, as well as the menu around it, that allows us to move between the different objects in the Sol System, including the planets and moons, and the information they contain.

Everything is smoothly animated to ensure that the user experience is as satisfactory as possible, including the change of page, change of object, or change of information type. It also includes the stellar objects, which can be rotated with the mouse to check on different spots around them.



The main site after logging in, showing the most important object of the Sol System, its host star Sol.

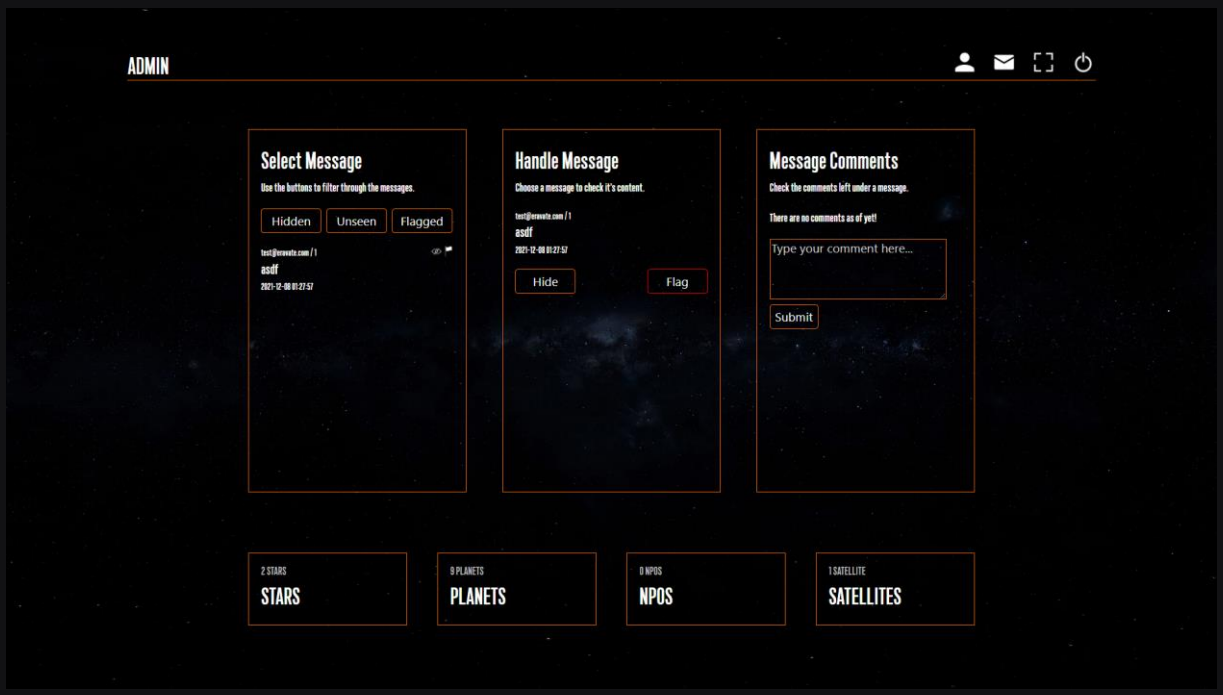
The menu also contains different buttons, which are, from left to right:

- Notify of Error (Opens box with message input that is directly sent to the Admin page).
- Make the app go Fullscreen.
- Open the Command Prompt (It can be used to move between object via text rather than the menu).
- Log Out (Sends us back to the Log In page).

Admin - I know I can fix it...

This part of the app has been created for the average admin user, where you want to oversee the app and its information without the hassle of knowing how to code, or change basic information while being away from the server and having no means of access.

It comes in six different parts, four parts are reserved for the different types of objects we store, those objects can be deleted, altered, or even created directly from this page. We also have message handling, for the errors that different users report, as well as the users tab, only accessible by Super-Admins, which allows you to create new users without verification, and check the logs of each user.



The admin site after accessing the message page.

Apart from everything explained previously, the app comes with a basic way of communication between Admins under messages in the form of comments, so that each Admin can know the stage of a reported error.

Finally, in the user's page, you can grant Admin privileges to already existing users, as well as remove those privileges to limit the access to the Admin app. It is important to remember that the Super-Admin role comes with great power, and it should only be given to trusted personnel.

Planification & Development – Where do we go from here?

This part will be exclusively referencing the preliminary stages of development and planification, and not the current state of the app, if you are not interested in these stages, you can skip directly to page 15.

As far as planification goes, I do not really plan a lot of stuff beforehand, as I have learned that if you do plan most of the things early on, you will have to compromise a lot of technicality and functionality within the app in order to achieve your end goals.

Of course, the most basic concepts, the overall app design, and the technologies to use must be planned before taking the first steps, so from early on I set myself a few points and guides to follow so that, in the end, I would know what I am looking for.

From the very first day I knew that the app would be written in a mix of PHP and JavaScript, with the use of three main libraries: PHPMailer, jQuery & ThreeJS, as well as some secondary libraries, like Swal & jQuery Editable Select.

As far as development goes, I usually develop in four phases, and this app was no different from how I usually do things, those phases are usually:

- Phase 1 (December-January): Early phase concepts, designs and bare minimum before coding can begin, no code is written during this part.
- Phase 2 (March-June): Base working app, with the bare minimum information and most of the design parts finished, so that I can see what I am working with and what I am looking for.
- Phase 3 (October-December): Full working app, most of the information has already been added, but not polished yet, some bugs and glitches can, and will, appear.
- Phase 4 (December-January): Polishing and Debugging, the testing phase, both black box and white box, making sure that everything is working as intended.

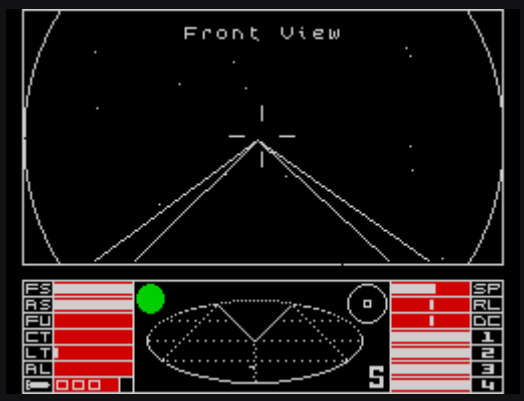
When it comes down to the time I spent on each phase, I lost count by June, and I did not record the time spent on both the third phase and the fourth phase. If I had to make a rough estimate, it probably would be between 600- & 900-hours total, ≈70% of which belongs to the third phase alone (Note: an estimate was made via Github data; 680 hours).

Early Concepts and Designs – How it all started.

I have started development on this app in late December / early January 2021, though the first month was mostly brainstorming ideas for the app. Since early on I knew I wanted to do something space-themed, but what would be both fun to program & useful at the same time?

While playing visual novels, I came across Planetarian, which really fascinated me with how little apps there are that give us easy access to common information about our local stellar objects, hence, the idea of a planetarium, where I take that information, and put it all in one place.

But Planetarian is only a small part of it, since a lot of inspiration came from 1984's Elite, a game that came out for the Commodore 64 and the first Atari that won several game awards. It was a game truly ahead of its time, which made me focus on making something that would make people think exactly that; "This is ahead of its time."

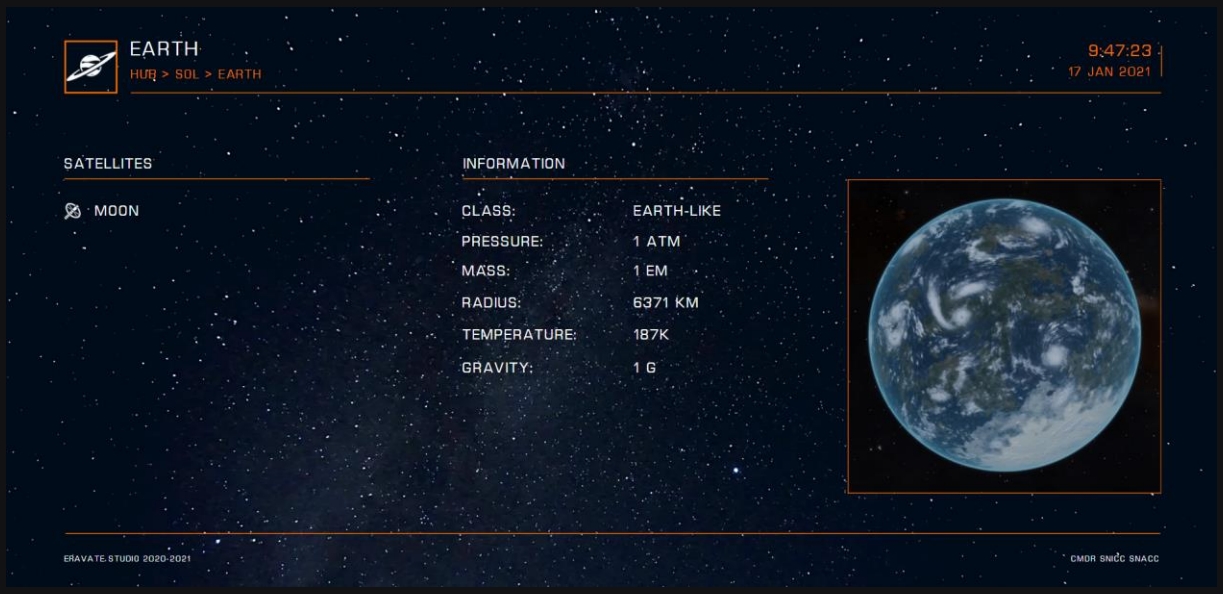


A 1984 ZX Spectrum port of 1984's Elite

Elite is also one of the inspirations behind the project name. Eravate has two origins, one of them is a Solar System within the game that does not post any significant importance, and the other meaning is a translation from Italian; I was often told "Eravate ancora giovani" which means "You were still young" in Italian.

But, back to the topic, the first step I needed to take was to reshape the meaning of "Web Application". Most Web Applications I came across were nothing but mere web pages with some interactive bits put into them, and that would not pass in my case, I did not want a web page. What I wanted was a true web application.

In order to achieve that, I made sure that most of the design is completely original, and no templates have been used to make it. Almost every part of this web app has been made from the ground-up, as I did not want people to see this as another web page with which you could play.



The first design, very vague and containing almost no information at all, done via Canva.

This is how I came up with this design, which was mostly made to visualize the various parts of the app, and how I wanted the information to display. It was made mid-way through January, and I would not advance onto the next phase until early March.

As you can see, it is extremely basic and does not resemble the final product at all, which goes to show how much changes throughout development and how important it is to not "lock-in" yourself with a design, and leave that part as open as possible, because later on you may struggle to adapt it.

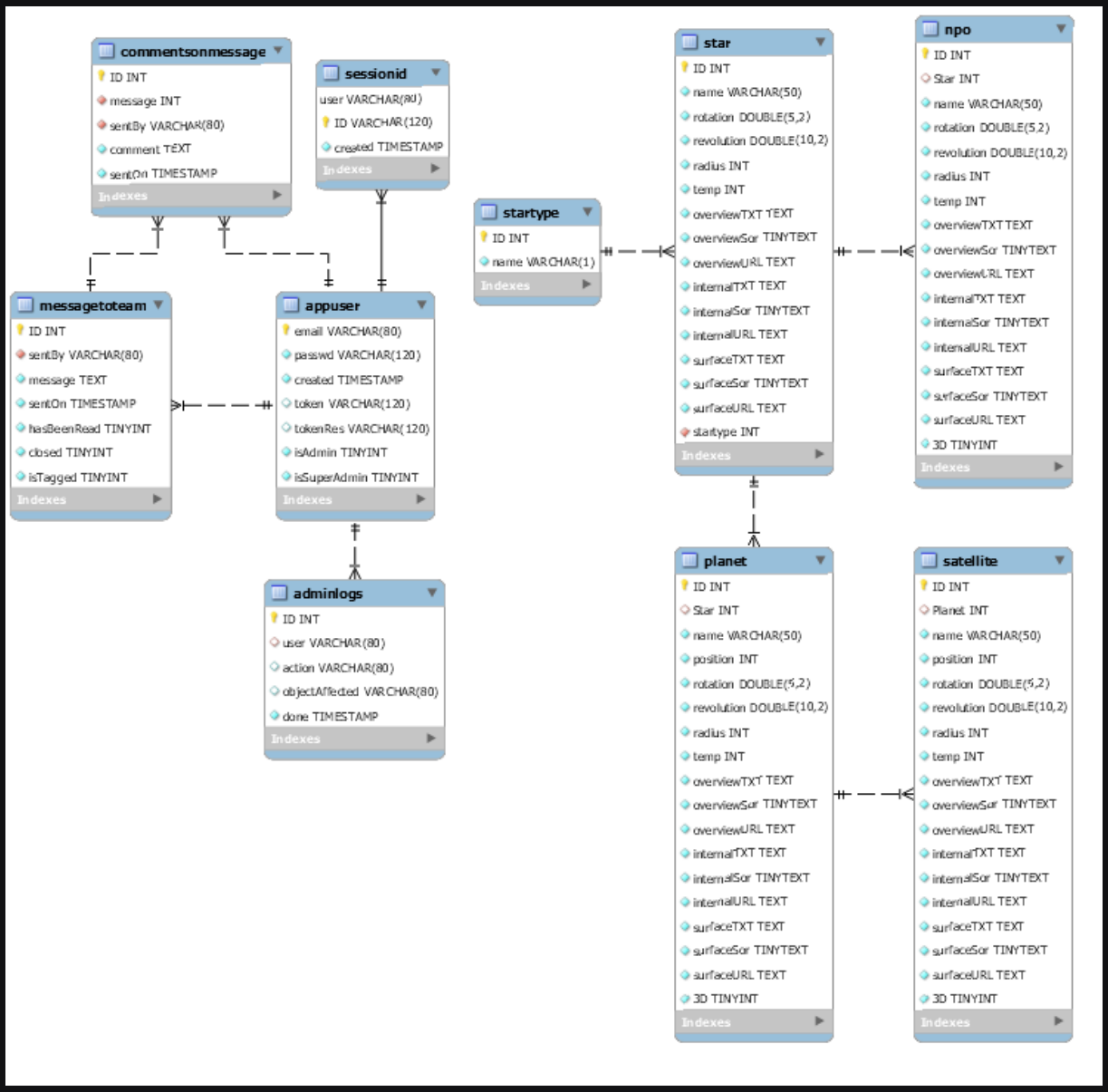
I also needed to make sure of one thing. It is important to make an application available on as many devices as possible, and that does include mobile phones and tablets. This only added to the "Main objectives" list, and remained at the bottom of said list. But it was, in the end, achieved.

But let us not get ahead of ourselves, because first, I need to answer a few questions to myself, like "What information do I want to store?" or "How do I display the objects themselves?" before I could start working full-on on this project, and the answer to said questions lies in the next phase.

Base working app – Database and Information.

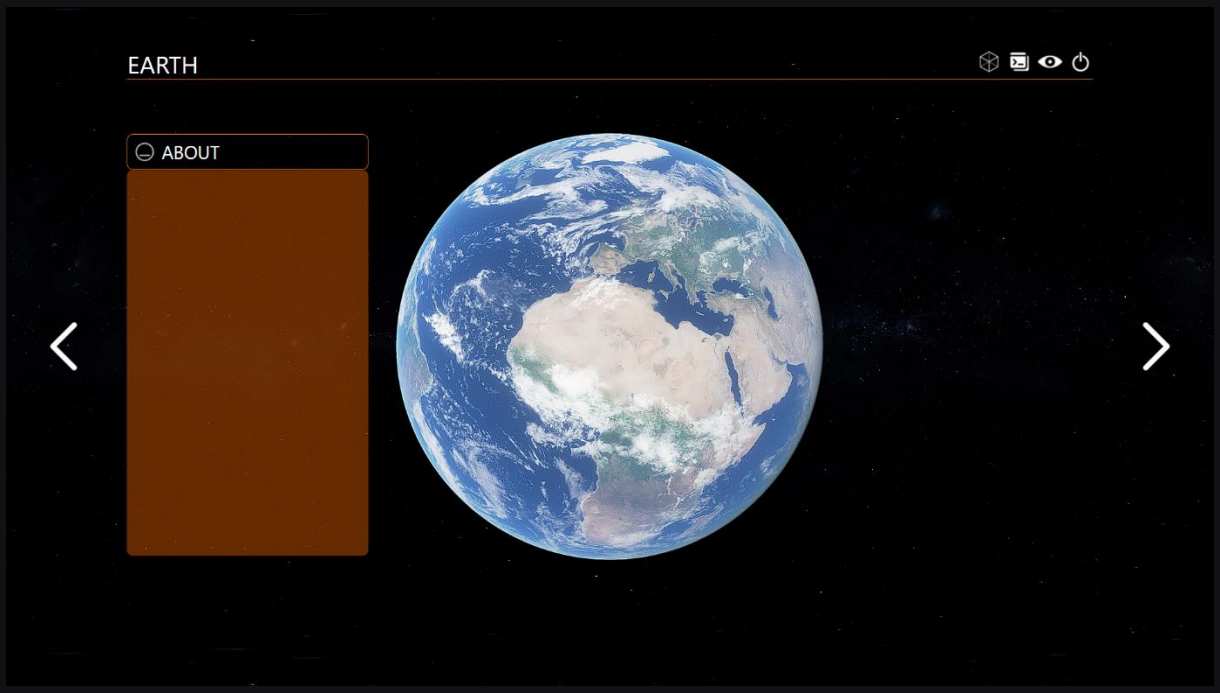
After having the project on hold for around 2 months, I picked it back up in early March, and I started by thinking what information I wanted to store, and how I wanted it to be stored, and not only the information related to the stellar objects, but also the users.

Apart from this, I also wanted to get down some code to know where I am standing, and what I am exactly looking for in this app, but not too much of it, since I was still deciding on how to exactly display that information and make the best use of the space given to me.



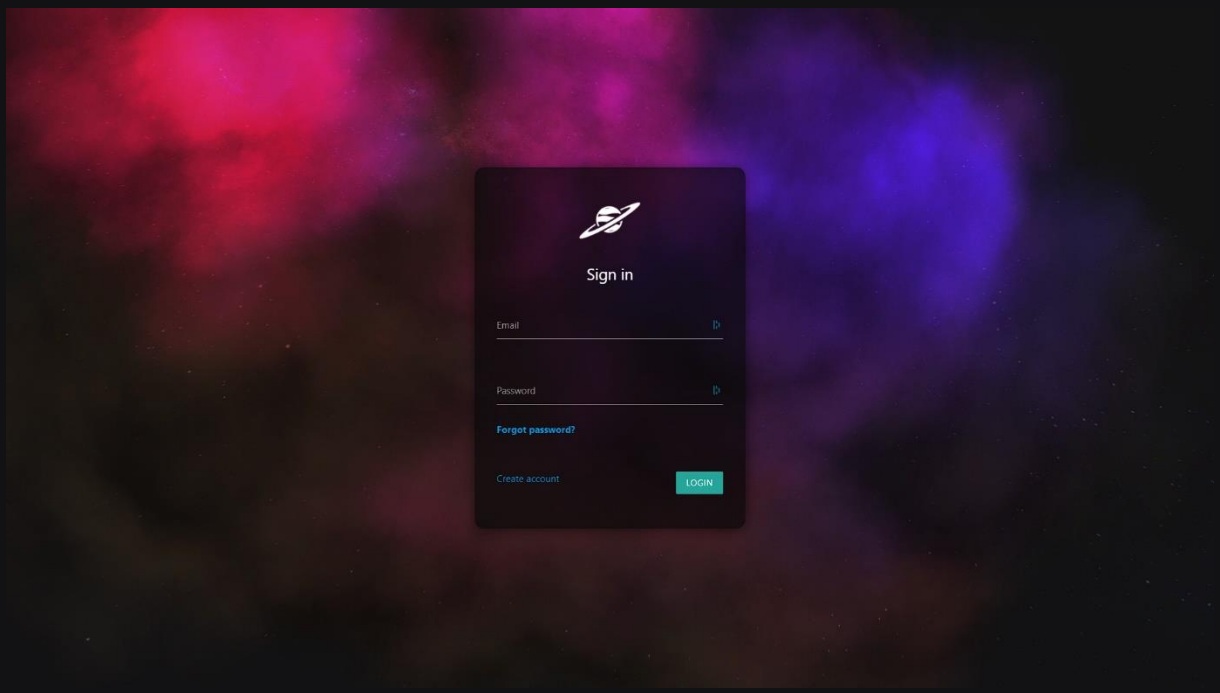
The finished MySQL database main design from early June.

When it comes down to the code, I created the backbone of what would be the final design used in the app, but with some minor changes happening to it in the later stages, since like I said before, I did not want to lock myself too early with the design.



The backbone of the app, the earliest instance of the currently used design, done exclusively with JavaScript, HTML and CSS.

When it comes to the Log In site, it was already mostly done by this stage, and only a handful of things changed at the later stages apart from the colours used, while the admin site was not yet created.



Early instance of the Log In site.

Full working app – In Detail.

There is not much to say about this phase, since it was perfectly explained in the Requirements & Reasonings stage, but I can still clear up a few things on how I got there.

While the previous phase ended mid-way through June, this one has not started until early October, which meant that during summer break, no progress has been made, Since I expected the second phase to be the last one, I gave up for the next 3 months, but I decided that I cannot show a half-way done project, even if I wasn't that invested in programming since.

This is the most time-consuming phase, since I spent around 6 days a week working on it for over 3 months straight, and each workday would be between 8 and 12 hours of coding, which in the end would make this project much bigger than anticipated, although most of those hours would be brainstorming solutions and analysing different routes.

The main reason it took this long to finish it was the fact that I used a new JavaScript library that I have never used before – ThreeJS, which helped me with both the Log In site and the Main site, to load both the nebula and the stellar objects.

As far as how it works – The Log In nebula has fifty white cloud objects loaded onto the canvas (the background of the page), and three lights (Red on the left, Purple on the middle and Blue on the right) change their colour accordingly, each cloud also has a random rotation to simulate the behaviour of a nebula.

The stellar objects have three event listeners placed on them – once the mouse is pressed down on the object, it allows rotation, which is handled by the mouse move event, and once the mouse is released from the object, it blocks rotation, and slowly returns to its previous rotation.

The Admin site was created using the backbone and design from the Main site, yet it is the only site to not include the ThreeJS library, and its only purpose, in the end, was making sure that normal users could handle the data inserted in the Database, without any MySQL knowledge.

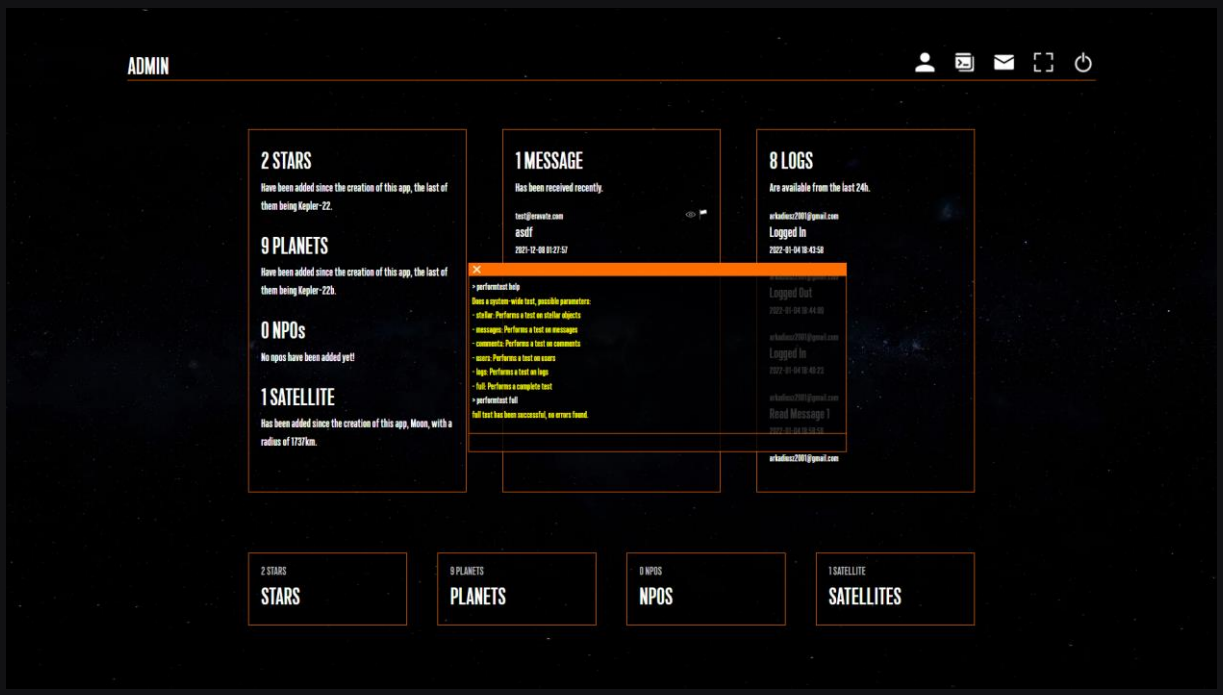
Polishing & Debugging - The testing phase.

For the final phase of the app, we have to make sure that everything is working exactly as we intended it, and for that reason, we have to perform a series of tests. There are two types of tests that can be done: black box (with code) & white box (without code).

Both tests have been performed on this app. For black box testing, I gathered ten individuals that would use the app for approximately thirty minutes to an hour each, after which I would receive reports of design inconsistencies and flaws within the client-side environment.

Some notable changes to the app from this testing method were: On hover borders appearing on Textareas (specially the console window), Admin User page not using full width in the middle window & colour inconsistencies within the Log In site.

On the other hand, white box testing is done within the code, to ensure that server-side errors do not occur, for which I have prepared a console window within the admin site to allow each admin to be able to perform either a full or individual test on each query done to the server.



A test being run on the admin site.

Installation – Just in case you want to go local.

So, you have looked through the documentation, analysed every point made in it, and made sure this app will suit your needs, excellent choice if I do say myself, but before you get overly excited, a few things need to be stated before you can get to work on your own version.

This app is, and will always be, completely free of charge, and completely free to use. The only thing I require from anyone using it is some sort of credit within their own implementation of the app if you choose to go local or choose to host it on your own servers. And as far as structure goes, check the next page.

For the requirements, you will need a server deployed that can handle PHP (I used Apache for the most part), as well as a database that is compatible with MySQL 8.0.21, otherwise you will have to adapt the eravatecreation.sql file accordingly.

As far as installation goes, everything is available directly on my GitHub (Eravate) under the project "ProjectEravate." From there you can either fork it onto your own repository or download the code directly onto your own server and work from there. You can also use the CD provided with this documentation.

Most PHP files within the PHP folder need to have a connection to the Database, the one included within the app is an example localhost connection, remember to adjust every PHP file with the correct Database information.

For database, copy the code included in eravatecreation.sql and run that query inside your database. By default, it comes with basic data that you can use, though, you are not forced to, it is not essential (It is important to note you need at least one star system to be able to access the main page).

Additionally, the files forgotPassword.php & process.php come with a basic implementation of PHPMailer, make sure to adjust it to your own needs as well, especially the mail Username and mail Password variables, which are set to "SECRET" by default.

If you are going to keep your own code public on GitHub, make sure that you, and I cannot stress this enough, censor your PHP Database connection & mail configuration, you do not want that information to go public. It can lead to extremely dangerous consequences.

Structure – For when you want to make your own changes.

Well, after implementing the app, you noticed that something is lacking, or that something can be improved upon, and since you already went with the trouble of setting it up, you want to fix it yourself, but you do not understand the structure that much. Fret not, as I will explain you everything right here.

I went through the trouble of documenting all of the code, from the php files to the script files, as well as trying to keep everything as easy to understand as possible. That being said, you will need some basic idea on how JavaScript, PHP and some of their libraries work.

On the main directory, you will see three main php files (Index for the main site, admin for the admin site, and login for the log in site), as well as four secondary php files (activation for activating an account, cookiepolicy for the cookie policy, forgotten for re-establishing a password and loader as a sort of transition between the php files). The PHP files in the main directory should not contain any code except the necessary session code.

Each main php file comes with 3 JavaScript files associated with it (included in the script folder); A main with the same name, a start JS file, and an end JS file. They also make use of another JavaScript file called commands.js, which contains all of the functions. As for 3D functionality, login makes use of clouds.js, while index makes use of 3dplanets.js.

You will see that the script folder contains more files as described here. That is because I make use of several libraries within the app (Like ThreeJS, Editable Select, GSAP or DragControls). Make sure you do not touch those script files.

The php folder serves the purpose of containing php code used exclusively to read and/or write data on the database. As explained previously, most PHP files inside this folder come with their database connection, and some come with their PHPMailer connection, make sure to fill them with your own data.

Apart from this, we also have some standalone folders; Font for containing the font used within the app, CSS for containing CSS for the main pages (each page has its own CSS, but they can also use additional ones), Icons for the Icons used within the app, Music containing the original score (Not used in the final product, but it is included) and the Models folder, which contains the planetary textures (and where new ones will be saved).

Credits – My final few words.

You have reached the end of the document, and so, the next few lines will not be containing any information about the project, they will just contain a couple of my thoughts, experiences and emotions related to it, so you can feel free to skip them if you so desire.

There is one question I always ask myself at the end of every project I do, I think that it is one that plenty people do, and it is whether I have learned anything, and if learning that thing was worth the time, I spent on doing so, since there is not any other way I would profit from this.

After giving it a lot of thought, I would probably say that I have not learned much from this, but I do not consider the time that I put into this project as time wasted. After all, I was not forced to do it nearly as advanced as I ended up doing it, but I still did it.

I also learned that programming is not really something made for me, or at least I am not ready to make a living out of it just yet, and for the time being, I want to pursue something different in my life, specially that I have not been doing anything different for over 4 years.

Still, I had a blast sharing this project with some of my friends and showing them my "spaghetti" code that took over a year to be written fully, and seeing their faces of terror when they could not make up the meaning of a single line, so maybe not all of it ended up so badly?

But enough of this philosophical monologue, if you ever feel intrigued by this project, struggle with a part while trying to deploy your own version or need any help in general related to this project, or your own project, feel free to contact me on Twitter @Erawaito or on discord Snicc Snacc#7272.

Lastly, I wanted to thank you for taking your time to read this, and for showing any interest at all in this. Any interaction you have with this project means the world to me and shows me that all the time I wasted was not for nothing.

「The starry night sky, It really is beautiful, do you not think so?」