Technologies research

Personal project

22/09/2023

Saamie Vincken

Version control

Name	Version	Date	Description	Details
Saamie	0.1	11-09-2023	Initial setup	Add introduction, ideal strategy and questions.
Saamie	0.2	14-09-2023	Development framework	Add research into development frameworks
Saamie	0.3	15-09-2023	3D modeling	Add research into 3D modeling
Saamie	0.4	25-09-2023	Database research	Add research into astronomical datasets
Saamie	1.0	26-09-2023	Final touches	Final additions and layout adjustments to document
Saamie	2.0	16-10-2023	Comparison tables	Change pros and cons tables to comparison tables for more valuable Information
Saamie	2.1	14-11-2023	Value neutral writing (feedback)	Remove value neutral writing after feedback
Saamie	2.2	14-11-2023	Small adjustments (feedback)	- Main and sub questions - Define more precise scope limitations
Saamie	3.0	14-11-2023	Conclusion JavaScript framework	Rewrite conclusion about JavaScript framework after feedback

Contents

Introduction

For the individual project this semester, a focus will be on developing a web application about stellar evolution. It will introduce users to information about the different stellar phases. The information will include calculation details and visual representation of the phases using 3D modeling.

This document's purpose will be to compare multiple development frameworks which can be used to build the application, a suitable library or tool for the 3D models. In addition to that, this project needs a detailed data set for future additions to the application, like a data table where similar stars and their locations are shown.

The goal is to create a better understanding and overview of the available options and make an informed decision about the choice of technology.

Questions

The following questions have come up during exploration of the possibilities of this project.

Main question

What are the most suitable technologies for developing this application and its specific requirements?

Sub questions

1. What technologies should be used for development of the web application?
2. What technologies should be used for development of the 3D models?
3. What data sources should be used for future implementation of the data table?

What technologies should be used for development of the web application?

For this project, a JavaScript framework will be used which supports the required features for this project. The reason for this choice is because JavaScript frameworks often offer high performance, organized code structure, built-in security features, and have large communities and documentation available. There are many different possibilities regarding JavaScript frameworks, in this document some available option will be explored.

What technologies should be used for development of the 3D models?

To visualize the different phases of a star, 3D models of each phase will be made. There are many different platforms and libraries offering 3D modelling options, which is why the most common and highly rated options will be explored.

What data sources should be used for future implementation of the data table?

In future development, this web application will also have a data table for analyzation of larger stellar datasets. This data should be accurate and come from a reliable source which is why the different available datasets should be evaluated.

Project Objectives and Scope

The ideal situation for this project would be to find the right framework and tools before starting the development phase, to ensure that everything is secure, future oriented and has high performance.

The primary objective of this web application project is to create an educational tool that introduces users to the various phases of stellar evolution. The project's goal is to provide users with information about the life cycles of stars, including calculation details and visual representations of these phases through 3D modelling.

The goal of this research is to find technologies that can be used for the project. With a main focus on exploring the available technologies for developing the web application and 3D models.

Scope

The scope of this project is the development of the web application itself, including the creation of a user-friendly interface for accessing stellar phase information and interacting with 3D models. The application's content will focus on the most influential stellar evolution stages, from protostars to supernovae, offering an interactive learning experience for users interested in astronomy and astrophysics.

Limitations

While the project is informative and interactive, it does not aim to provide all stellar phases. For this project, 10 stellar phases have been selected to use for the modeling, because they are a few of the most influential and common phases of a star. It should present simplified models and visualizations to create an accessible platform for a variety of users.

Additionally, the application's scope is limited to a web application, and it does not include plans for native mobile apps or other specialized platforms at this stage.

Legenda

To compare each technology, comparison tables will be used. The following legenda will accompany each table in this document.

Positive impact on project

Moderate impact on project

Negative impact on project

Development framework

Finding a suitable framework for the web application is essential for achieving the project's goals. Three JavaScript frameworks have been considered for this purpose:

Table 1: JavaScript framework comparison

Framework → / Criteria ↓	**React**	**Vue**	**Svelte**
Architecture Reusability	High (Component-based with reusability)	Medium (Component-based, single-file components)	High (Component-based, with compile-time reusability)
Learning Curve	Medium (Moderate)	High (Simple)	High (Simple)
Documentation availability	High (Many libraries and tools)	High (Many libraries and tools)	High (Many libraries and tools)
Integration into Existing Projects	Medium (Suitable for new projects)	High (Easy integration into existing projects)	Medium (Suitable for new projects)
Performance	Medium (Good performance)	Low (Not optimized for 3D visualization in this project)	High (Due to component compiling)
Ecosystem Flexibility	High (Abundant libraries and tools)	Medium (Fewer libraries and tools)	High (Abundant libraries and tools)
Community Support	High (Large and experienced community)	Medium (Limited expertise, language barriers)	Medium (Limited expertise, relatively new framework)

React

React is a popular JavaScript library for building user interfaces. It offers a large option of libraries and tools that could help the development of the web application. Reacts component-based architecture is particularly suited for creating interactive and data-driven applications. [1] [2]

Vue

Vue is another JavaScript framework known for its simplicity. It provides a straightforward way to build user interfaces. Vue's performance is generally good for most web development tasks but might not be optimized for 3D applications. Its strong suit lies in easy integration into existing projects. [3] [4]

Svelte

Svelte is a relatively new JavaScript framework that compiles components into highly efficient JavaScript at build time. Its approach to building web applications could result in a lightweight and high-performance platform, which is crucial for 3D visualization. [5] [6]

[1] (React, n.d.)
[2] (Ragala, 2023)
[3] (Vue, n.d.)
[4] (Tobiasz, 2022)
[5] (Svelte, n.d.)
[6] (Bhushan, 2023)

3D Libraries

For creation of the 3D models of each stellar phases, a suitable technology is needed. This choice can significantly impact the quality and performance of the web application, as well as the complexity of designing it. The following options have been considered for creating the 3D models:

Table 2: 3D library comparison

Framework → / Criteria ↓	**Three.js**	**Babylon.js**	**A-Frame**	**P5.js**
Learning Curve	Medium (Complex for beginners)	High (User-friendly and easy to learn)	High (Very beginner-friendly)	High (Very beginner-friendly)
Customization	High (Offers high control)	Medium (Supports customization)	Low (Limited customization)	Low (Relatively limited capabilities)
Suitability for Complex 3D Models	High (Suitable for complex models)	Medium (Less suitable for complex models)	Low (Not suitable for complex models)	Low (Not suitable for complex models)
Ease of Integration	High (Easy integration)	Low (May require additional integration)	High (Built on top of Three.js)	Low (May require additional integration)
Community & Documentation	High (Large community and extensive documentation)	High (Large community and extensive documentation)	Medium (Good community and documentation)	High (Large community and extensive documentation)
Ideal for Fast Prototyping and Simple Models	Medium (Less ideal for fast prototyping and simple models)	Low (Not ideal for fast prototyping and simple models)	High (Ideal for fast prototyping and simple models)	Low (Not ideal for fast prototyping and simple models)

Three.js

A popular and often recommended JavaScript library for the creation of 3D models. It offers a medium-level learning curve, which can be more complex for beginners. However, it provides many customization options and control over the 3D models. Three.js has a large community with extensive documentation, making it a good choice for a wide range of 3D applications. [7] [8]

Babylon.js

Another JavaScript framework for building 3D applications. It is known for it's ease of use and is a great choice for interactive models. It provides moderate availability in customization options and control over the 3D models. However, it is less suitable for very complex 3D models.[9] [10]

[7] (three.js JavaScript library, n.d.)
[8] (Abidi, 2023)
[9] (Rahi, 2023)
[10] (Welcome to babylon.js 6.0, n.d.)

A-Frame

An easier option for creating 3D models, without needing to use too much JavaScript or 3D graphics concepts. It is often used for game design and creating 3D experiences with minimal coding. However, its simplicity makes it also less useful for more complex 3D models. [11] [12]

P5.js

P5.js is a lightweight and beginner-friendly framework for creating basic 3D models. It is particularly well-suited for artistic projects, offering an open and creative platform for experimenting with 3D art and interactive visualizations. However, it is not easy to integrate and is less useful for more complex models because it lacks the advanced features and capabilities required. [13]

Astronomical databases

When considering data sources for future implementation of a data table containing stellar data, it's essential to find reliable and open-source databases or datasets that provide correct information about stars. The following options have been considered for the data table:

Table 3: Astronomical database comparison

Framework → / Criteria ↓	**SIMBAD Astronomical Database**	**The Open Exoplanet Catalogue**	**Gaia Data Release**	**NASA API Portal**	**Caltech/NASA Exoplanet Archive**
Customization	High (open-source API)	High (open-source API)	Low (no open-source API)	High (open-source APIs)	High (open-source APIs)
Data quality	High (precise data on millions of stars)	Medium (focuses on exoplanets)	High (precise data on millions of stars)	Low (relatively low information on stellar data)	High (detailed exoplanet and star data)
Data update frequency	High (frequent updates)	High (frequent updates)	High (frequent updates)	Medium (less frequent updates)	High (frequent updates)
Data diversity	High (stars, planets, galaxies)	Medium (primarily exoplanets)	High (stars, planets, galaxies)	Low (focused on other astronomical objects)	High (exoplanets and stars)
Data Licensing and usage	Medium (open access with some restrictions)	High (open access for most data)	Medium (open access with some restrictions)	High (open access for most data)	Medium (open access with some restrictions)
Data complexity	High (complex data structure and format)	Medium (moderate data complexity)	High (complex data structure and format)	Low (simple data structure and format)	Medium (moderate data complexity)

[11] (A-Frame, n.d.)
[12] (Entity-Component-System, n.d.)
[13] (KofiGroup, n.d.)

SIMBAD Astronomical Database

SIMBAD is a widely used astronomical database with information on over 12.5 million objects. It offers an open-source API and is well-maintained, providing reliable scientific references. However, its data structure and format can be complex. [14] [15]

The Open Exoplanet Catalogue

This dataset focuses on exoplanets and their host stars. It features an open-source API, easy data access, and is well-maintained. However, it's limited to stars with known exoplanets, and not all stellar types are covered. [16] [17]

Gaia Data Release

The Gaia mission maps billions of stars in the Milky Way, offering high-precision data on positions, distances, motions, and brightness. While it provides valuable data, there's no open-source API, and it may require advanced tools and knowledge for analysis. [18]

NASA API Portal

NASA offers a set of open-source APIs covering various aspects of astrophysics. They are well-maintained and well-documented, but there are no datasets specifically for stars. [19]

Caltech and NASA Exoplanet Archive

This database contains information about confirmed exoplanets and their host stars. It provides detailed stellar information and data visualization tools. It also offers open-source APIs, but, like the others, it's limited to stars with known exoplanets and may not cover all stellar types. [20]

[14] (NASA, Set of identifications, measuremenets, and bibiliography for Astronomical data (SIMBAD), n.d.)
[15] (Strasbourg, n.d.)
[16] (Hanno Rein, 2020)
[17] (Open exoplanet catalogue, n.d.)
[18] (Gaia, n.d.)
[19] (NASA, NASA APIs, n.d.)
[20] (NASA, NASA APIs, n.d.)

Conclusion

This document has provided an overview of the technological resources needed for this project. Several key questions have been addressed and a comparison between available options have been made.

JavaScript framework

Regarding the choice of technologies for development of the web application, three JavaScript frameworks have been considered. After research into each framework, a comparison table has been made. In the table React, Vue and Svelte have been considered. Because of the overall most positive impact on the project (*Table 1: JavaScript framework comparison),* the choice has been made to use Svelte. This choice is based on the high performance of the framework, component compiling, simplicity and readability in syntax, and available documentation to get started.

3D modeling library

When it comes to 3D modeling, the choice of technology could significantly impact the quality and performance of the application. After examining options like Three.js, Babylon.js, A-Frame, and P5.js, Three.js came out as the most suitable choice (*Table 2: 3D library comparison*) It provides a balance between complexity, customization, control, and community support, making it ideal for creating 3D representations of stellar phases.

Database

For the data sources required for the future implementation of a data table, it's crucial to select reliable and open-source databases. The SIMBAD Astronomical Database appears to be a strong option due to its extensive data on millions of stars and open-source API. It provides a solid foundation for the creation of a detailed and accurate data table for analyzing stellar datasets. (*Table 3: Astronomical database comparison*)

In summary, this document sets the starting point for the development of the application. The success of this project will further depend on effective implementation, a user-friendly design and informational content.

Bibliography

Abidi, A. (2023, 6 2). *The Three.js supremacy*. Retrieved from dev.to: https://dev.to/ayoub3bidi/the-threejs-supermacy-4a45

A-Frame. (n.d.). Retrieved from aframe.io: https://aframe.io/

Bhushan, A. (2023, 6 27). *Svelte vs React: Featurs, Performance and more*. Retrieved from kinsta.com: https://kinsta.com/blog/svelte-vs-react/

Caltech. (n.d.). *NASA Exoplanet archive*. Retrieved from exoplanetarchive.ipac.caltech.edu: https://exoplanetarchive.ipac.caltech.edu/

Entity-Component-System. (n.d.). Retrieved from aframe.io: https://aframe.io/docs/1.4.0/introduction/entity-component-system.html

Gaia. (n.d.). *Gaia data release screnario*. Retrieved from cosmos.esa.int: https://www.cosmos.esa.int/web/gaia/release

Hanno Rein, I. f. (2020). *Open exoplanet catalogue tables*. Retrieved from github.com: https://github.com/OpenExoplanetCatalogue/oec_tables/

KofiGroup. (n.d.). *P5JS and 5 things you need to know about it*. Retrieved from kofi-group.com: https://www.kofi-group.com/p5js-and-5-things-you-need-to-know-about-it/

Kramer, N. (2020, 8 10). *Svelte pros and cons, ecosystem overview and top resources*. Retrieved from daily.dev: https://daily.dev/blog/building-with-svelte-all-you-need-to-know-before-you-start

NASA. (n.d.). *NASA APIs*. Retrieved from https://api.nasa.gov/

NASA. (n.d.). *Set of identifications, measuremenets, and bibiliography for Astronomical data (SIMBAD)*. Retrieved from science.nasa.gov: https://science.nasa.gov/astrophysics/astrophysics-data-centers/set-of-identifications-measurements-and-bibliography-for-astronomical-data-simbad

Open exoplanet catalogue. (n.d.). Retrieved from openexoplanetcatalogue.com: https://www.openexoplanetcatalogue.com/

P5.JS. (n.d.). Retrieved from p5js.org: https://p5js.org/

Ragala, B. K. (2023, 9 18). *What are the pros and cons of react*. Retrieved from knowledgehut.com: https://www.knowledgehut.com/blog/web-development/pros-and-cons-of-react#cons%C2%A0%C2%A0

Rahi, M. (2023, 2 24). *Comparing Three.js and Babylon.js: Which JavaScript 3D library is right for you?* Retrieved from mohammadrahi.medium.com: https://mohammadrahi.medium.com/comparing-three-js-and-babylon-js-which-javascript-3d-library-is-right-for-you-7196ef21949e

React. (n.d.). *React - the library for web and native user interfaces*. Retrieved from react.dev: https://react.dev/

Strasbourg, U. d. (n.d.). *SIMBAD Astronomical database - CDS (Strasbourg)*. Retrieved from simbad.cds.unistra.fr: https://simbad.cds.unistra.fr/simbad/

Svelte. (n.d.). *Svelte: cybernetically enhanced web apps*. Retrieved from svelte.dev: https://svelte.dev/

three.js JavaScript library. (n.d.). Retrieved from threejs.org: https://threejs.org/

Tobiasz, F. (2022, 4 28). *Using Vue: Pros and Cons*. Retrieved from thecodest.co: https://thecodest.co/blog/pros-and-cons-of-vue/

Vue. (n.d.). *The progressive javascript framework*. Retrieved from vuejs.org: https://vuejs.org/

Welcome to babylon.js 6.0. (n.d.). Retrieved from babylonjs.com: https://www.babylonjs.com/