

Front End Engineering (FEE) Final Project

# StellarVisions (APoD) Web App

Devasheesh Kaul - 2110992006

# Project Overview

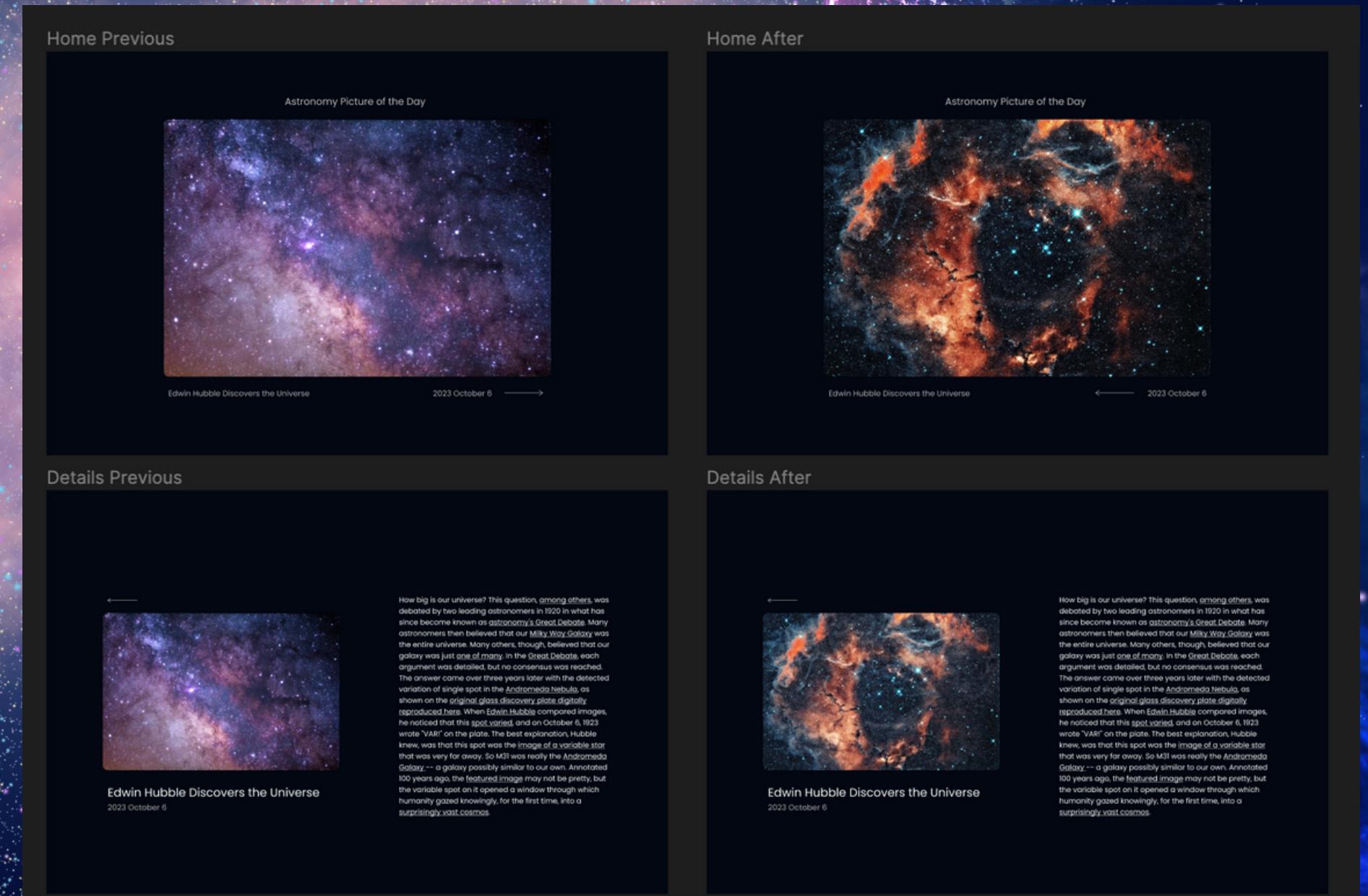
- The primary goal of this project is to create a user-friendly web application that brings the captivating world of astronomy closer to users.
- I wanted to provide a daily dose of wonder and knowledge by displaying NASA's Astronomy Picture of the Day (APoD).
- The web app allows users to explore the APoD on the homepage, providing a brief glimpse into the cosmos each day.
- Users can also input a specific date to view an APoD from that day, fostering exploration and learning.



# Project Design

- Utilized Figma for prototyping and visualizing the user interface.
- Ensured a responsive design for seamless accessibility on various devices.
- Iteratively refined our design based on user feedback.
- Ensured the final design met user expectations.

# Figma Design



# Project Structure

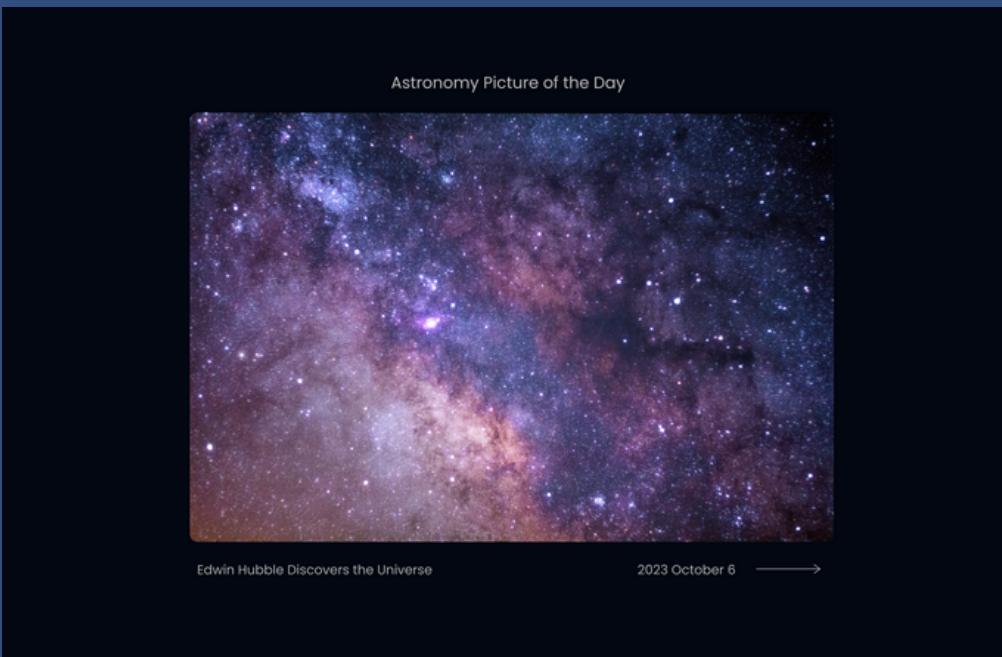
- **src/**: The core source directory, housing components, constants, context, pages, and types.
- **components/**: Contains essential React components for building the app.
- **constants/**: Stores constants like the months used in the project.
- **context/**: The directory responsible for managing the app's state using the Context API.
- **pages/**: Where the main pages of the app, Home and Info, are located.
- **types/**: Houses type definitions for the data used in the project.
- **App.tsx**: The entry point of the app and routing for pages.
- **index.css**: Global styles applied throughout the app.
- **main.tsx**: Renders the React app into the DOM.

stellar-visions

```
├── src/
│   ├── components/
│   │   ├── Details.tsx
│   │   └── Hero.tsx
│   ├── constants/
│   │   └── months.ts
│   ├── context/
│   │   └── APIDataContext.tsx
│   ├── pages/
│   │   ├── Home.tsx
│   │   └── Info.tsx
│   └── types/
│       ├── API.ts
│       ├── App.tsx
│       ├── index.css
│       ├── main.tsx
│       └── vite-env.d.ts
... (other config files and assets)
```

# Components

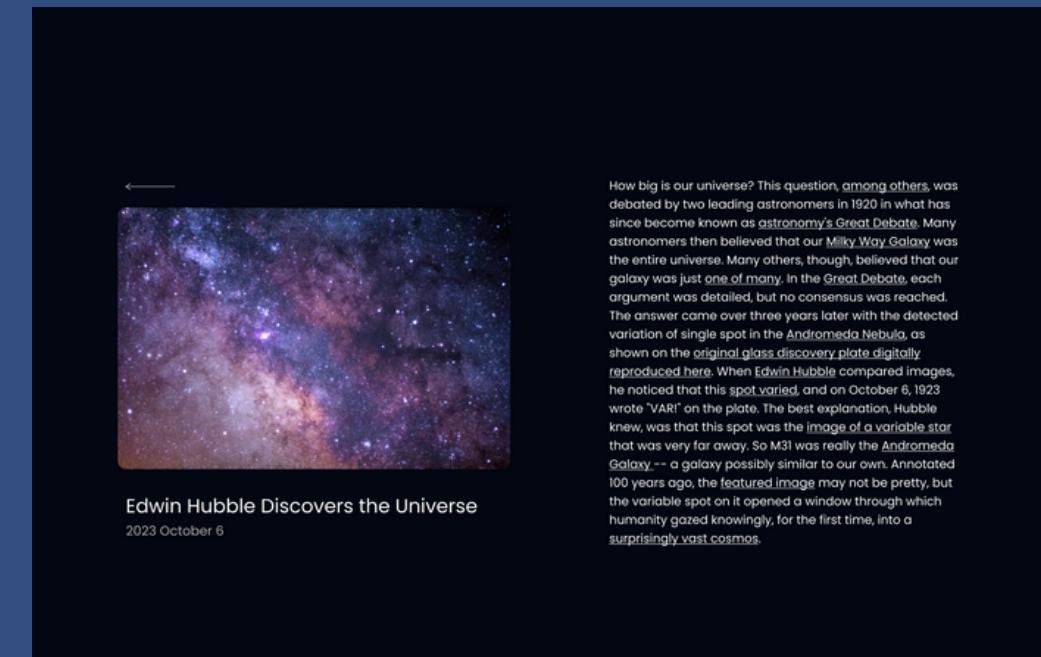
## Hero



- The Hero component is responsible for displaying the Astronomy Picture of the Day (APoD) on our homepage.

- It's the visual centerpiece of our app, offering a captivating glimpse into the cosmos.

## Details



How big is our universe? This question, among others, was debated by two leading astronomers in 1920 in what has since become known as astronomy's Great Debate. Many astronomers then believed that our Milky Way Galaxy was the entire universe. Many others, though, believed that our galaxy was just one of many. In the Great Debate, each argument was detailed, but no consensus was reached. The answer came over three years later with the detected variation of single spot in the Andromeda Nebula, as shown on the original glass discovery plate digitally reproduced here. When Edwin Hubble compared images, he noticed that this spot varied, and on October 6, 1923 wrote "VARI" on the plate. The best explanation, Hubble knew, was that this spot was the image of a variable star that was very far away. So M31 was really the Andromeda Galaxy -- a galaxy possibly similar to our own. Annotated 100 years ago, the featured image may not be pretty, but the variable spot on it opened a window through which humanity gazed knowingly, for the first time, into a surprisingly vast cosmos.

- The Details component is where users can explore in-depth information about the APoD.

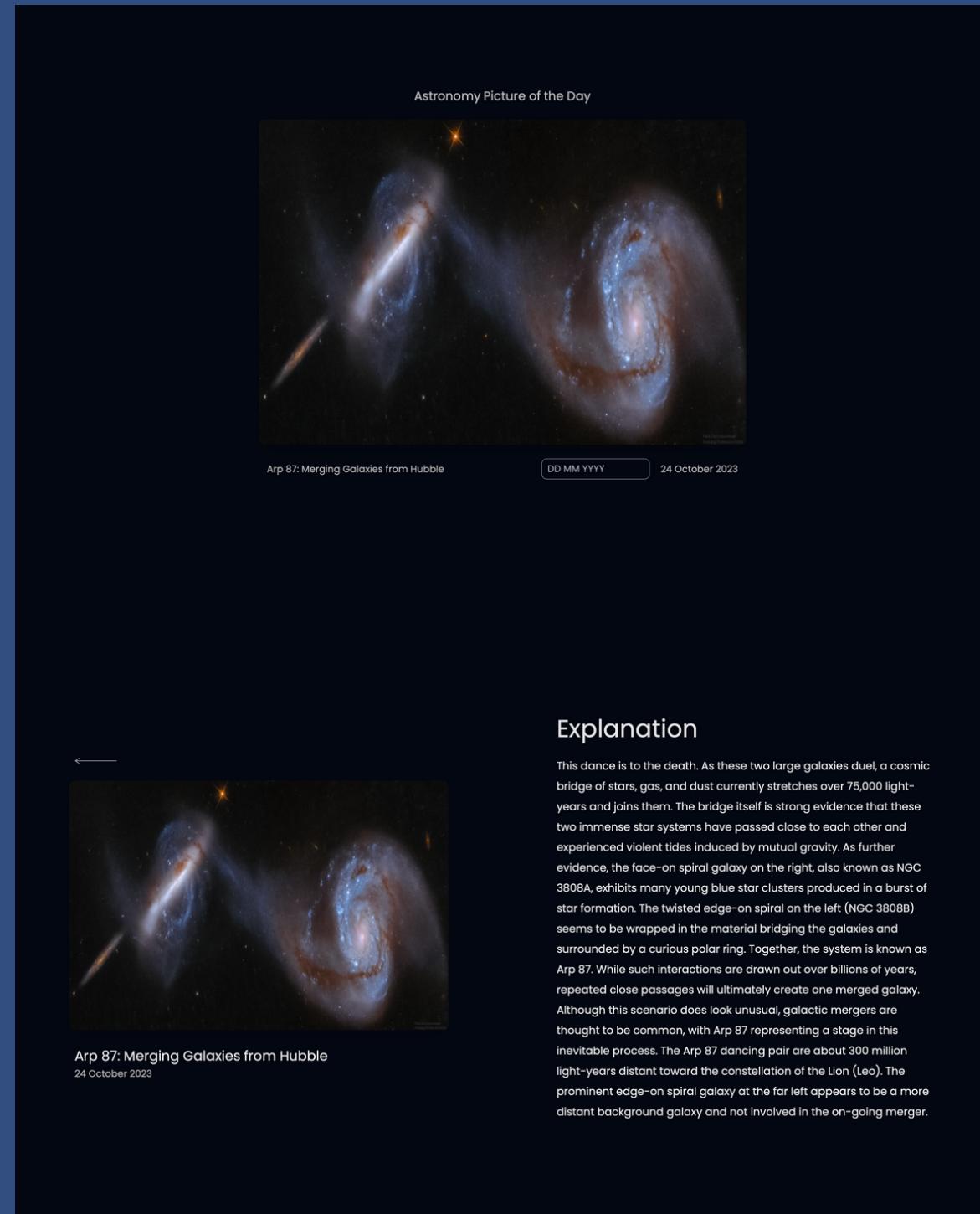
- It provides a deeper understanding of the featured celestial object and its significance.

# Context API

- The Context API provides a centralized store for our application's data.
- With the Context API, data is globally accessible within the app, making it readily available to any component that needs it.
- In my project, I've implemented a "Provider-Consumer" pattern, where the APIDataContext acts as the provider of data and various components consume that data where needed.
- Data flows efficiently from the provider to consumers, ensuring real-time updates and a responsive user experience.

# UI Interface

- The web app is built with React, a powerful JavaScript library, allowing us to create a dynamic and responsive user interface.
- Tailwind CSS, a utility-first CSS framework, was instrumental in crafting a clean and visually pleasing design.
- The design was carefully thought out to ensure readability, accessibility, and a captivating presentation of astronomy content.



# Challenges

## API Rate Limit and Data Storage

- Challenge: Dealing with the API rate limit when fetching Astronomy Picture of the Day (APoD) data.
- Solution: I stored the received API data in the local storage. If data for a specific date is already in local storage, the app uses it instead of making additional API requests.
- Result: Improved API usage efficiency and faster performance when navigating between pages.

## User Input and Data Override

- Challenge: Allowing users to search for images for specific dates and ensuring the correctness of date formats.
- Solution: Implemented an input box with a date format guide in the placeholder. Incorrect date formats trigger an error message.
- Result: Enhanced user experience and the ability to access APoD images for any desired date.

## Smooth Page Transitions

- Challenge: Ensuring smooth transitions between pages without full page reloads.
- Solution: Utilized React Router DOM for page navigation and incorporated the Link component for seamless transitions.
- Result: Enhanced user experience with smooth, non-disruptive page navigation and transitions.

# LESSONS LEARNED

- I learned how to effectively use TypeScript to our advantage, making our codebase more controlled, type-safe, and robust. Type definitions and interfaces played a crucial role in enhancing code quality and reducing potential runtime errors.
- I discovered the utility of AbortController for handling fetch requests. Implementing ‘AbortController’ within the cleanup function of useEffect allowed us to cancel requests when the component unmounts, improving overall resource management.
- I gained valuable experience in utilizing external APIs, such as the NASA Astronomy Picture of the Day (APoD) API. This provided insights into integrating real-world data into web projects, enhancing content richness and authenticity.
- The project reinforced the importance of structuring code in an organized and clean manner. The modular component architecture and project structure allow for scalability and easier maintenance.



Thank you  
very much!