

Brief Technical Description

Gaby Norris 241143

Open Window, Creative Technologies

Interactive Development 200

Tsungai Katsuro

22 May 2025

Table of Contents

Heading in Title Case.....	3
Key Features.....	3
Dropdown Title Selection.....	3
Interactive Doughnut Chart.....	3
Challenges and Limitations.....	3
Visualisation Used.....	4
UI Screenshots.....	4

Heading in Title Case

Disney+ Finder is a React-based web application that helps users easily explore and evaluate top-rated content available on Disney+. It uses the Disney Plus Top Movies and TV Shows API, which provides curated metadata for a selection of the most popular titles on the platform. The app is designed to make content discovery intuitive and data-driven. Instead of scrolling endlessly through categories, users can select a title from a dropdown list and immediately view key insights about that title, helping them choose the best possible TV shows and movies based on quality and popularity.

Key Features

Dropdown Title Selection

Instead of typing a search term, users can quickly select from a list of curated Disney+ titles via a dropdown menu, making the browsing experience fast and frustration-free.

Interactive Doughnut Chart

Each selected title generates a **Chart.js doughnut chart** that visually represents:

- IMDB score
- TMDB score
- IMDB popularity

Challenges and Limitations

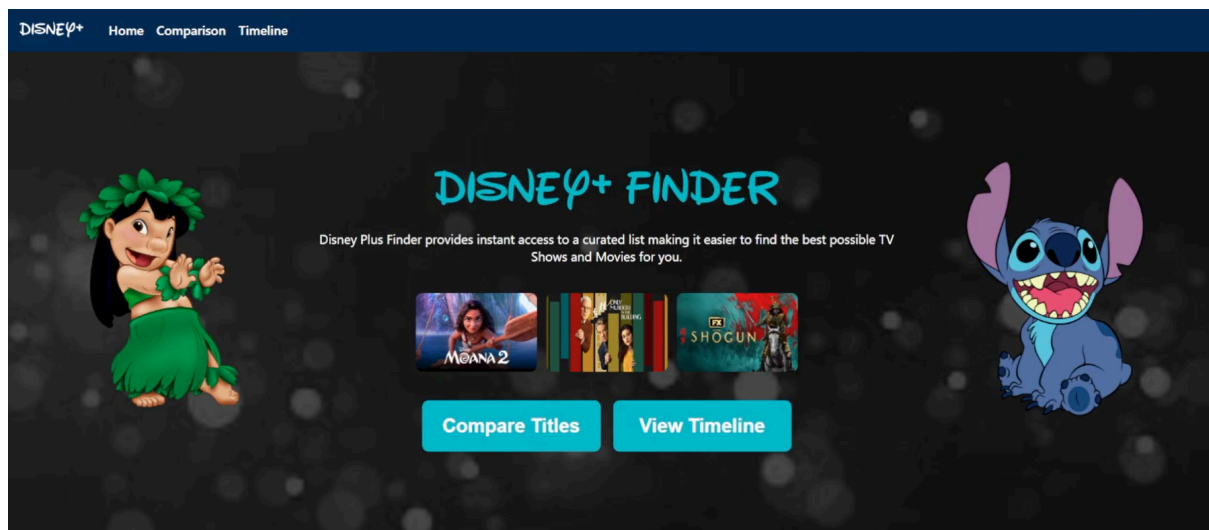
A challenge I came across was figuring out the Disney Plus Top Movies and TV Shows API because the endpoint provided by the creator is getting random, and the get list endpoint did not have descriptive element names for me to call from. So I just removed the get random in the fetch URL link however, the issue that still persists is that there are only 10 items in the array. I had a setback with the limited usage on the API I had to use 3 different emails, which means I called the API at least 202 times. This led to a delay in development as I reached out to the creator but did not get an answer.

I struggled with the visibility of the text in the dropdown, as when a title was selected, the font colour would change to black and against the background of the page, it was not legible. Another thing I battled was the alert box that pops up on the compare page, and how to create it. A minor challenge I experienced was what data to use that can be visibly distinct in the doughnut chart.

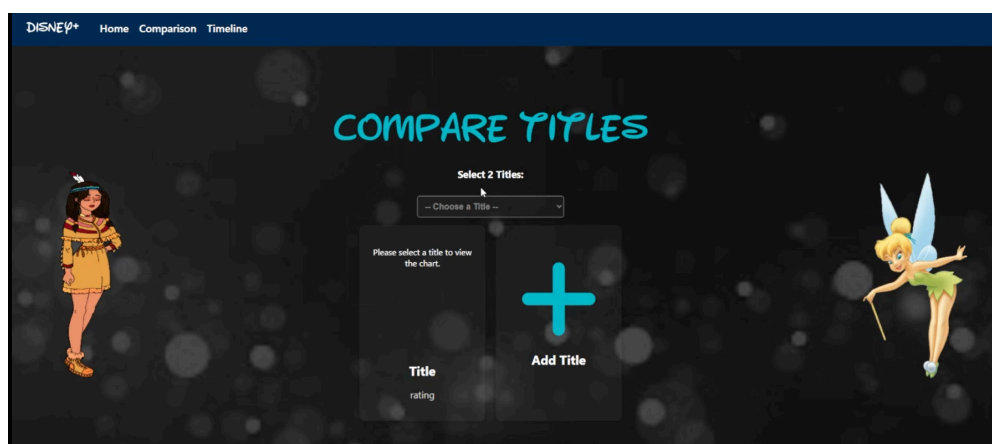
Visualisation Used

Using [Chart.js](#), the data is visualised in Doughnut Charts.

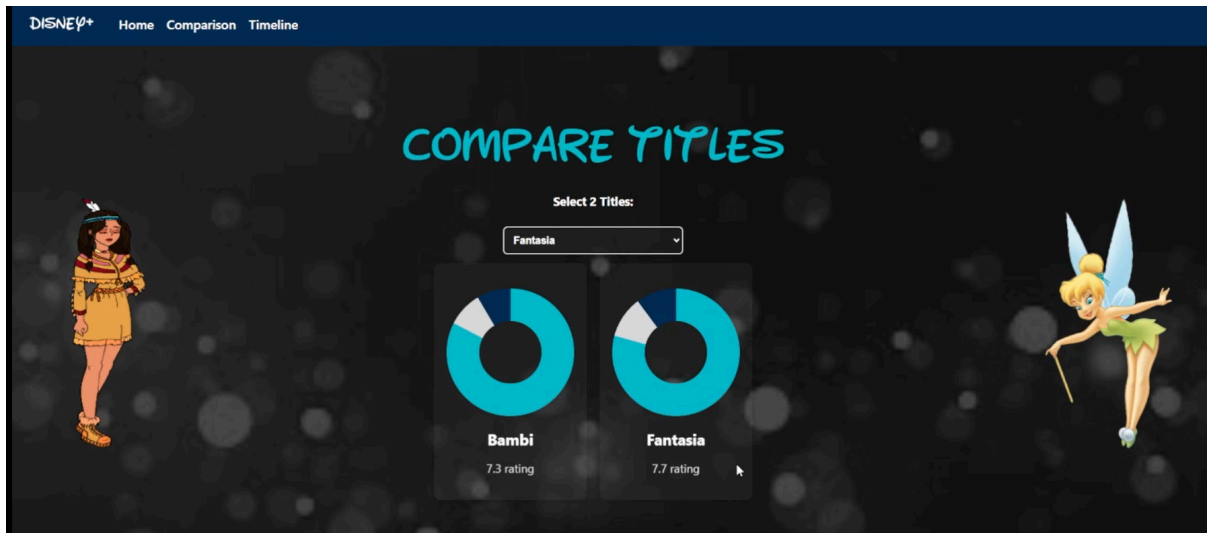
UI Screenshots



Home Page: Displays character images and thumbnails. It has navigation buttons that will take you to the other pages.



Comparison Page: Displays character images and cards that will contain the charts. There is a dropdown menu where the title will be selected for data representation.



Comparison Page: This is how the page will look when both Titles have been selected there will be two doughnut charts with the relevant data.

Timeline Page: This page is blank as I did not have enough time to do the necessary work to complete it.