Initial Technical Report

Introduction

About our site

The knowledge we have about outer space is always changing and it is very useful for researchers and space enthusiasts to be able to view up to date information of the expansive realm that is outer space. Our goal is to provide is simple way for people to explore the many aspects of objects in space and their relationships to others. Currently this information is held in many different places. This makes it difficult for users to find information about the models we have chosen.

How we accomplish our goal

We accomplished our goal by going out and scraping data from APIs that provide current information about our models. The names and descriptions of the APIs we used are located at http://spacecowboys.me/about. After scraping the data we stored them in json files so that we could read and render their data using python flask.

Design

Space and UML Associations Explained

Our models, although for the most part simple enough to understand on their own, still require a bit of astronomy 101 knowledge to fully grasp our setup. Planetoid body, for starters, refers to any substantial body in space that has characteristics similar to a planet. However, this would not only include the planets themselves, but also dwarf planets, moons, and asteroids. It is worth mentioning that stars do not fall under this category. As a result, planetoid bodies can have at most one star since they need something to orbit and it’s typically a star. This is true in the case of planets, comets and most asteroids. Still, you also have moons and some asteroids that orbit other planetoid bodies, hence the recursive association for the planetoid body model. A planetoid body will also usually be within a galaxy along with satellites and stars. However, there are instances when planets and stars get ejected from their respective galaxy and end up somewhere in between two different galaxies. These are typically referred to as stellar outcasts and rogue planets. This is why we represented these multiplicities as aggregation with an open diamond to galaxy as there can be one or no stars associated with these models. Another aspect of our models worth mentioning is that our model satellite refers to the man made kind. This should not be confused with natural ones like the moon which also orbit planets. Satellite is also the only model that has a many to many relation with stars, planetoids and galaxies. This makes sense because a star and planet can have several satellites in is orbit and of course a galaxy will contain all of those.

Frontend

For our frontend we used bootstrap, css, and jquery to make our site look beautiful.
Each page consists of a navbar used for navigation between pages and page specific data. For the splash page we implemented a simple carousel to welcome visitors and invite them to explore. For the each of the model pages we created a grid based off of the data provided in our json files. Each element in the grid also links to a page where one can learn more about that specific item. The about page displays information about each of the team members, the tools we used, the APIs that were scraped, and statistics about the development of the website.

Backend

Our backend uses python Flask to render dynamic templates.

Future features

In the future we hope to make our site and Rest API dynamic in that we do not have to manually scrape the data for new models found by researchers. We would also like to implement searching and sorting capabilities based on the attributes and relationships of our models. This can and will be done using Angular2 or React. The only reason that searching and sorting is not available yet is because Angular2 and React have a high learning curve and we decided that our time would be better spent finishing the overall design and look first.