

QUACKATHON REQUIREMENTS: Solar System

Deadline: 3/24 - 4/2

The website must be a SPA application. (A single-page application)

Planets:

- ☐ Each planet of the solar system **MUST** have its own profile page.
- ☐ Each page must include at a minimum (Name of the planet, size, distance from the Sun, three identifying facts, solid or gas coe, and a visual representation of the planet)
- ☐ **NOTE:** No images, assets, emojis, or SVGs are permitted.
- ☐ The visual representation of each planet **MUST** be made ENTIRELY out of **CSS**.
- ☐ If the planet has a moon, you **MUST** display the following information for every moon (Name, history behind name, and size).
- ☐ **Planet pages MUST** include at least two different component styles.
- ☐ If a planet has more than one moon, the moon information must be displayed in repeating elements.
- ☐ All planet and moon content **MUST** be stored in a database.

Backend:

- ☐ You **MUST** use foreign keys.
A foreign key is a set of attributes in a table that refers to the primary key of another table. The foreign key links these two tables.
- ☐ Planets and Moons must be stored separate from one another.
- ☐ You **MUST** incorporate one **ENUM** for the planet data.
Enum is a data type in programming languages that defines a set of constant values. It is a distinct value type (i.e) all values in an enum are unique. Using an enum makes the code readable and maintainable.

Homepage:

- ☐ The homepage should be an overview of the solar system.
- ☐ The data included will be up to each team.
- ☐ The homepage **MUST** include at least 2 components that are NOT used on any planet page.
- ☐ Homepage content does not need to be stored in a database and can be coded into your components.

REAME:

- ☐ Your README **MUST** include instructions for me to spin up the project locally, as well as an overview of the tech and/or libraries used.
- ☐ It should also include a list of the Quackathon requirements and how those were met or exceeded by your team.