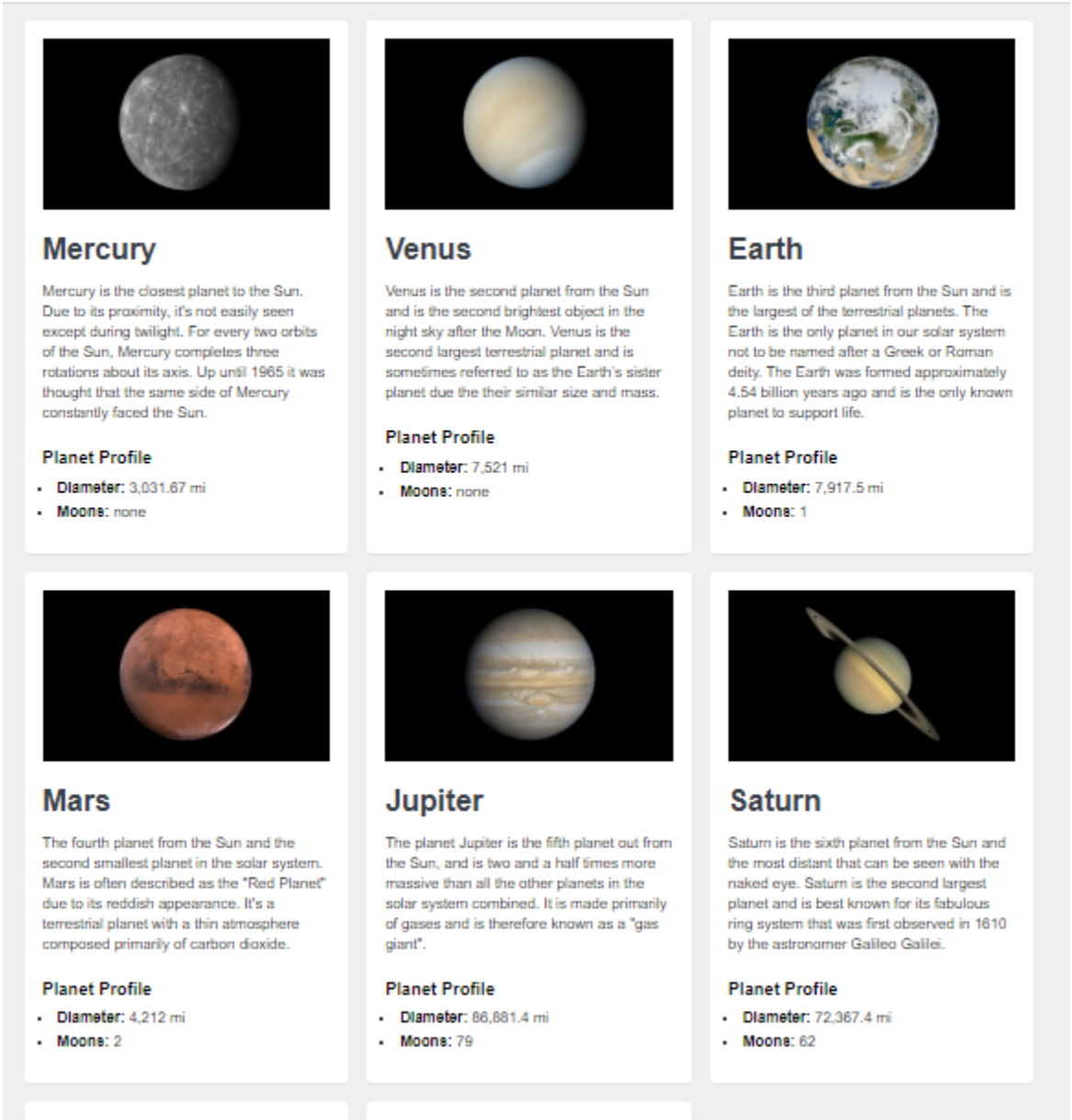
React Component Exercise:

In this Exercise you're going to practice JSX, creating and rendering components, passing down props, as well as iterating over data and more.

You are going to use React to build a Planet Card Site by breaking it up into smaller components. You must use CDN for your development.
For this lab, you are going to:

1- Open the project enclosed file,
2- Observe the app.js file
3- Check the instructions in the app.js
4- Build the React Program that renders the following image:



You need to use props and loops to build this page.

Your program should be functional and using React – 3
mark Using Props – 1 mark
Using loops – 1 mark
Adding delete function to each card - 1 mark

HTML, CSS, images and a JavaScript file for this exercise is in the folder. In index.html, links to React, and ReactDOM, and the Babel transpiler are provided at the bottom of the file.

You're going to write your React code inside the file app.js.

App.js currently contains an array of objects assigned to the constant planets. Each object has properties that describe a planet, like name, diameter, moons, description. And it has a URL property that points to an image located in the image folder. In index.html, in the comments you can see an example of the markup you'll need to use to create a planet card.

Just below the planet array in app.js, you will create two components, a Planet component that renders a planet card, and a container component, that iterates over the planet's array and renders a planet component for each object in the array.

You will need to pass the planet's data to the main container, then pass that data down to the planet card using props. You should use the commented markup in index.html as a reference for how to display the data.

The only text that should not use props is the h3 with the text, Planet Profile, and the text between the strong tags. Everything else needs to be displayed with props.

To display the planet cards, you will need to render the main container component to the DOM.

Finally, add a DELETE (or "X") Button to each card that removes the corresponding card if clicked. (1 mark)