Challenge 1

• A recapitulation of the topics from the lectures (100 - 200 words)

Lecture 1
The course started with a kick-off, this meant there was an introduction and explanation of the study guide. We had a test about the videos we had to watch through LinkedIn Learning. Chris explained the differences between HTML, CSS and Javascript and continued with the basics of JavaScript.

Lecture 2
The second lecture we started with a test again and discussed working with data, functions and objects. The entire lecture Chris showed many examples and how to use them. He showed how you can add features to objects and later find them back.

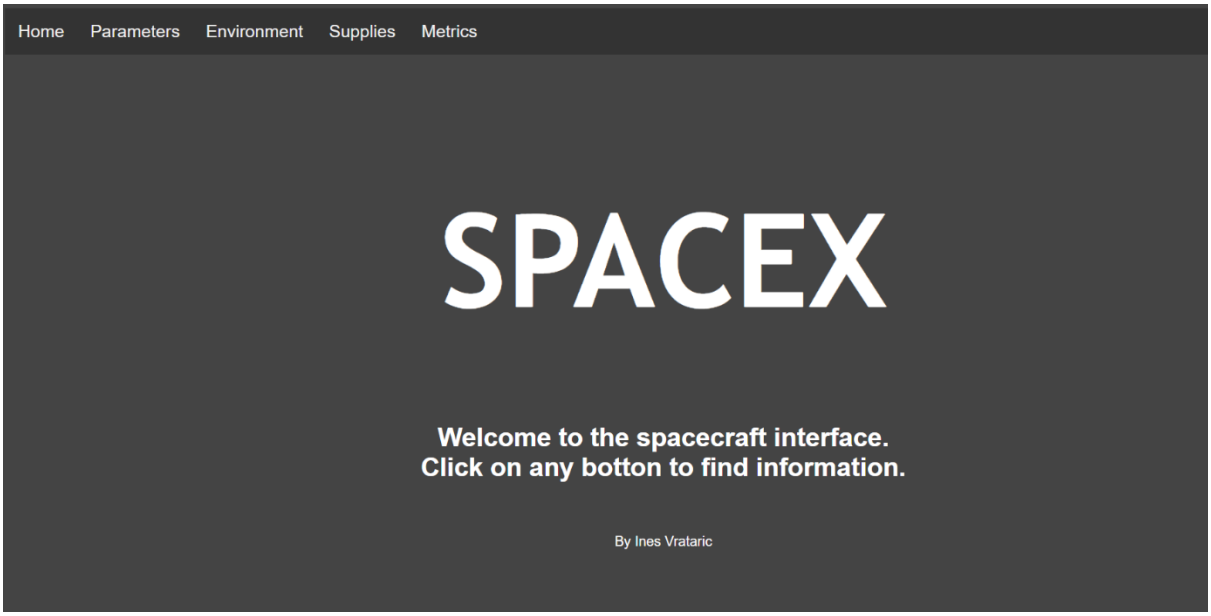
Lecture 3
The last week we once again started with a test. The topic this lecture was about functions and anonymous functions. This was followed by events and how to connect the events to objects. Chris showcased several examples and how to use them. Along the way of showing he explained common mistakes and error warning you could possibly encounter when programming yourself.

• A description of the assignment describing what the assignment was, what you tried to achieve and how you applied the topics. (100 – 200 words)
We need to code a website where potential Mars astronauts can find travel information. It will be an interface (ITSI) which is used around the entire spacecraft of SpaceX when it is send to Mars in 2030. The following features could be shown on the display: ships parameters, ships environment, supplies and metric. Within these topics, we could add these attributes: fuel, throttle, speed, acceleration (g-force), gravity, distance travelled, atmosphere, food, water, gravity converter and mars miles to earth miles. We have to keep our assignment to the following guidelines:

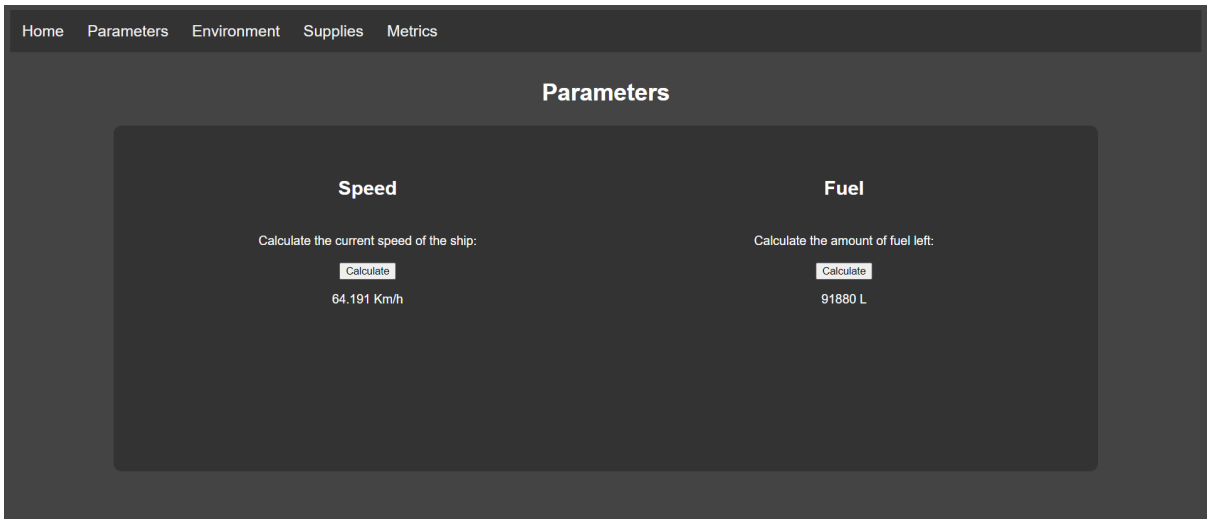
- Incorporate at least three working functionalities (static pieces of html-content are not functionalities)
- Use the topics we covered in class.
- Usage of frameworks and libraries are not allowed
- Write the page in validated HTML5
- This is an individual assignment.
- Elon Musk does not like Times New Roman on a white background, nor does he like comic sans. Please create a professional, consistent and modern look and feel.

It tried to achieve a simple to-the-point interface with clear and big buttons. I chose a grey background to correspond to the space-y aesthetic. I will make sure to user multiple JavaScript functions to make the interactive interface engaging. While doing this, I will use most of the covered topics during class like objects, elements, numbers and functions.

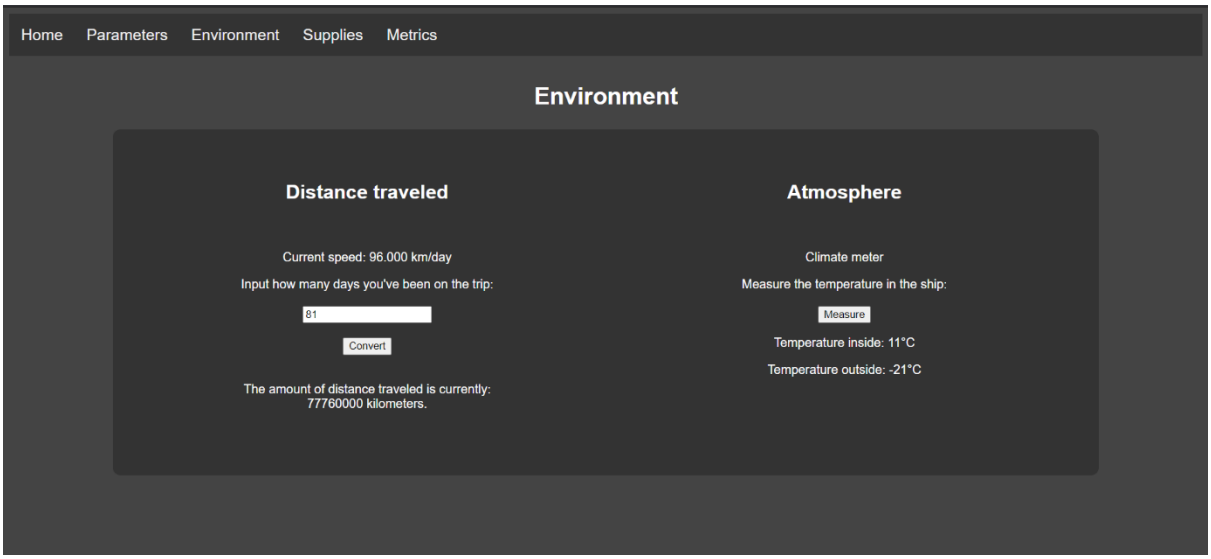
• The result of your assignment. Include screenshots/screen captures of your entire online challenge. Also include the URL.



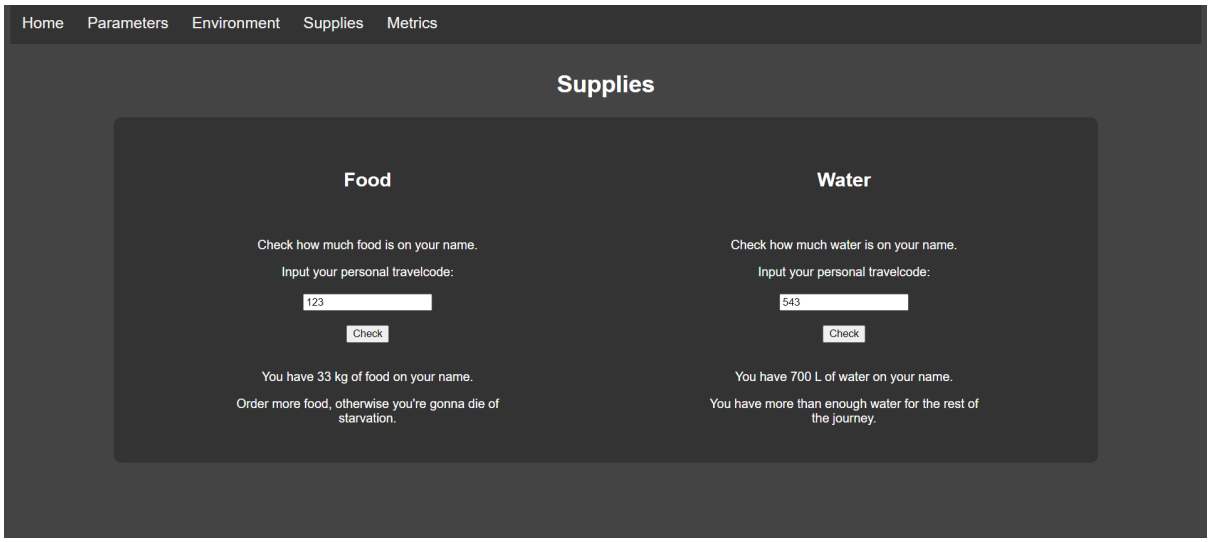
The user is met with a home screen with an navigation bar in the top left. The buttons display: parameters, environment, supplies and metrics respectively. When they press any of the buttons, they will be redirect to another page where more information is available. In linked all the different .html files in the index and the edited the lay out with the CSS.



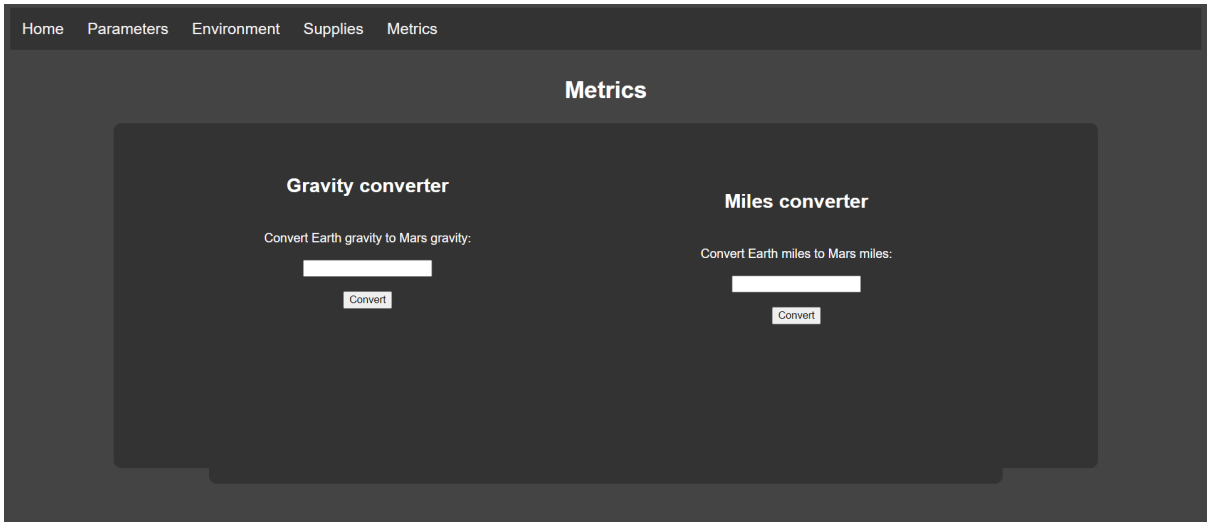
The second page is parameters and here the user can calculate at what current speed the spaceship is traveling. When the user presses the calculate button, they will see a randomly generated number which is the speed in km/h. Besides this function is the fuel meter where the astronaut can calculate how much liters of fuel is left on the ship. Once again, when the user presses the button they will get a random number which is the amount of fuel in liters.



The next page is environment and here the user can convert the amount of days they have been traveling into the total distance traveled so far. This is done by multiplying the current speed with the amount of days. On the right side the user can measure the current temperature inside and outside the spaceship. I did this by using Math.random() in the script.



The following page is supplies, this is the place the user can find how much food is dedicated to their name. The user puts in their personal travel code, which should be only numbers and then the system checks how much food is on their account. I made an if statement, so the user can get two different responses if they have enough food or not. The exact same function is used for the water check.



The last page shows the metrics, which are a gravity converter and a miles converter. The one on the left turns Earth gravity into Mars gravity with a function and the one on the right turns the Earth miles to Mars miles.

Link to website:

file:///Users/inesvrataric/Desktop/Programming%20Challenge%201/Challenge%201/index.html

Netfly:

https://ines-programming1.netlify.app/