Report Exam 1

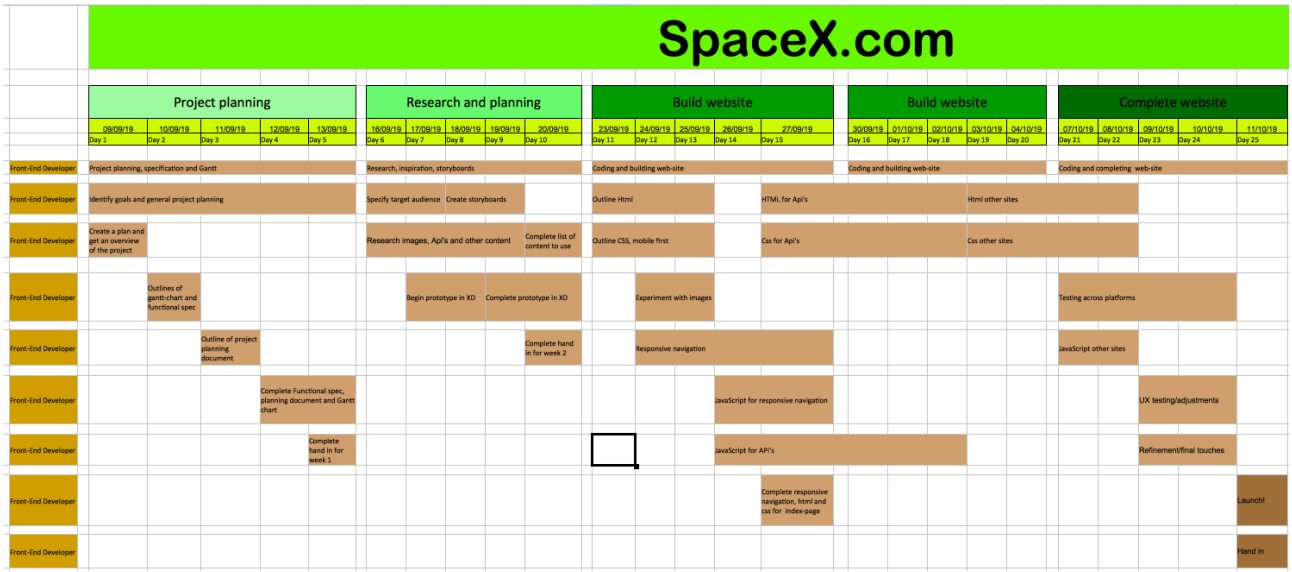
Link to website: http://mortenevensen.no/spacex
Link to Github repository: https://github.com/MortenEvensen/spaceX

Introduction

This exam was very interesting, and I was exited to create micro site. My experience is that micro sites are becoming more and more popular, and that these creative and exiting solutions that are used to share information. I think for instance that news sites like nrk.no and vg.no are becoming very good at using micro sites to create interesting cases. By exploring many micro sites, I got some pretty good ideas of how a microsite can be created, but a lot of them are using some very clever programming to achieve cool effects. My javaScript experience are limited, so I had to find some solutions that were possible to achieve with my knowledge and by "just" using vanilla script. I worked a lot on the scripts trying to figure out some nice ways of displaying information, and I had to try and fail a lot to get things right. I also enjoyed working with API's/Json and learn what is possible and not possible to do with that technology.

Body

I started the process by reading up on spaceX and Nasa activities, and researching what content my page could focus on. If I were to include "all" of the activities Nasa and spaceX are doing, the website would be way to big. Therefore I were looking for smaller areas of content that I could work with. As a part of the research, I looked at what content were available in the provided API's. I found that there were some pretty interesting information regarding upcoming launches, previous launches and history of spaceX, so I decided that the page would mainly focus on that. I originally thought that I would include NASA activities as well, but I eventually decided that spaceX activities were plenty enough to work with. As I researched the API's, I made the decision that most of the content of the page (if not all) should be generated through API's. The reason for that is that I wanted to use this opportunity to really get to experiment with and learn the possibilities and limitations of API's, as I haven't been able to use API's so much in the past.
Having the basic outline of what direction I would go planned, I went on to write a Gantt Chart. The Gantt chart ended up like this:



Gantt Chart

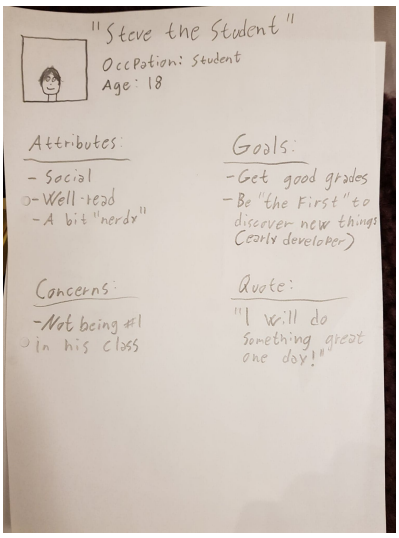
It contained all the work that I could imagine that needed to be done, and in an order that seemed doable. I made a similar design for a Gantt chart earlier in a previous module, where the feedback was good from the tutor. This time I got feedback that a Gantt chart should always go from top left to bottom right, so I've now learned the proper way of formatting a Gantt chart (this one is slightly wrong). From there I went on to start to write the functional specification.
I used the format for writing specifications that we have been taught earlier and I added all the elements that I found to be relevant for this assignment. The specification included milestones, list of functionalities, user cases, technical specification and more. In that process, I did come up with a few design ideas that I wanted to include in the final product, like a timer that is counting down until the next launch to space. I also included a mock-up of the page which looked like this:

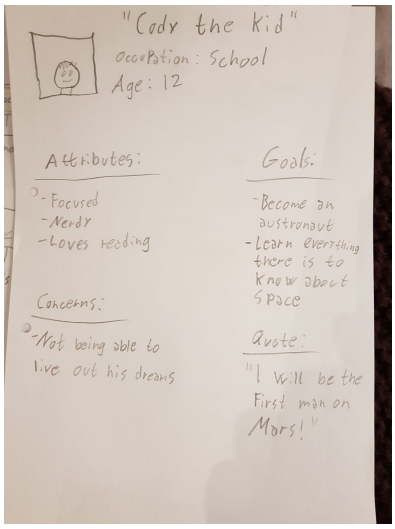
Webpage mock-up

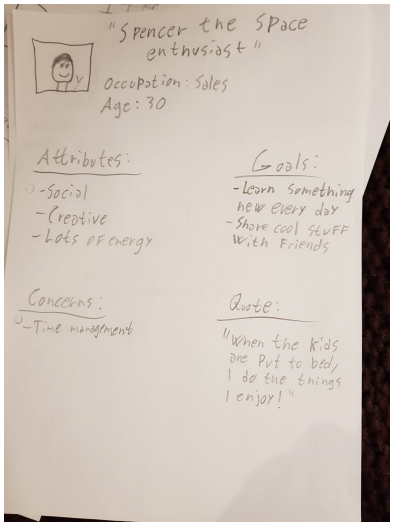
Phone mock-up

I made the mock-ups in Adobe XD for two reasons; 1) I'm not very good at drawing with pen and paper, and 2) I think Adobe XD is a really intuitive and fast way of testing out simple designs. It basically is the program that takes me the shortest amount of time to test out design ideas.
The mock-up included a simple navigation on top, some boxes for displaying upcoming launches and a timer that is counting down until the next launch to space. I knew I wanted those elements in the finished product, so that's why I focused on them in the early design. I also experimented with a few backgrounds and ended up with a pretty nice picture of earth seen from space.

The next step was decide on a target audience. As a wanted to display information in an interesting and interactive way, I quickly excluded “old people” in general as the target audience, since they have less experience with using the web (at least that's my experience). The target audience became young people/young adults who are curious or interested in space activity, as they would most likely be the audience that would appreciate an exiting microsite (compared to for instance a more Wikipedia-looking site). There is nothing wrong with Wikipedia, but it is a very straight forward and kind of boring way of finding information (at least in my opinion).
When I had decided on my target audience, I began creating personas. The personas were:





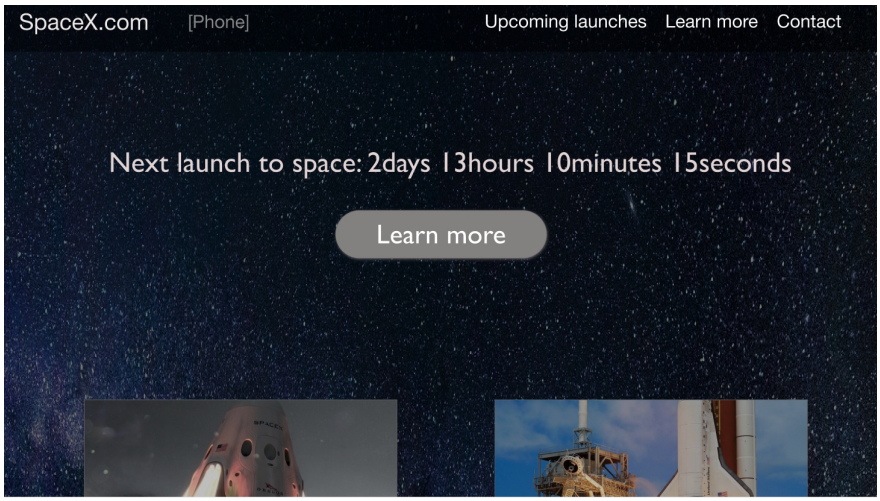


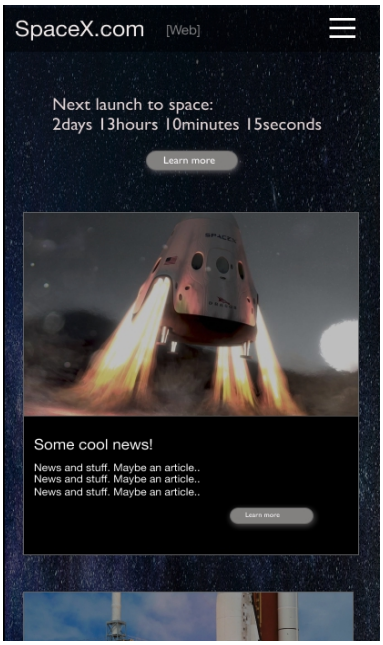
Steve the student *Cody the kid* *Spencer the space enthusiast*

They all got their separate user cases, which addressed potential issues that could occur.

I then went on to make a wireframe/prototype of the webpage in Adobe XD.
Link to prototype: https://xd.adobe.com/spec/d4389bbc-9de6-43aa-4ade-fbecafec272f-a209/

The prototype based the mock-up, but with much more consideration of what should actually go in to the page.





Prototype of index page - web

Prototype of index page on phone

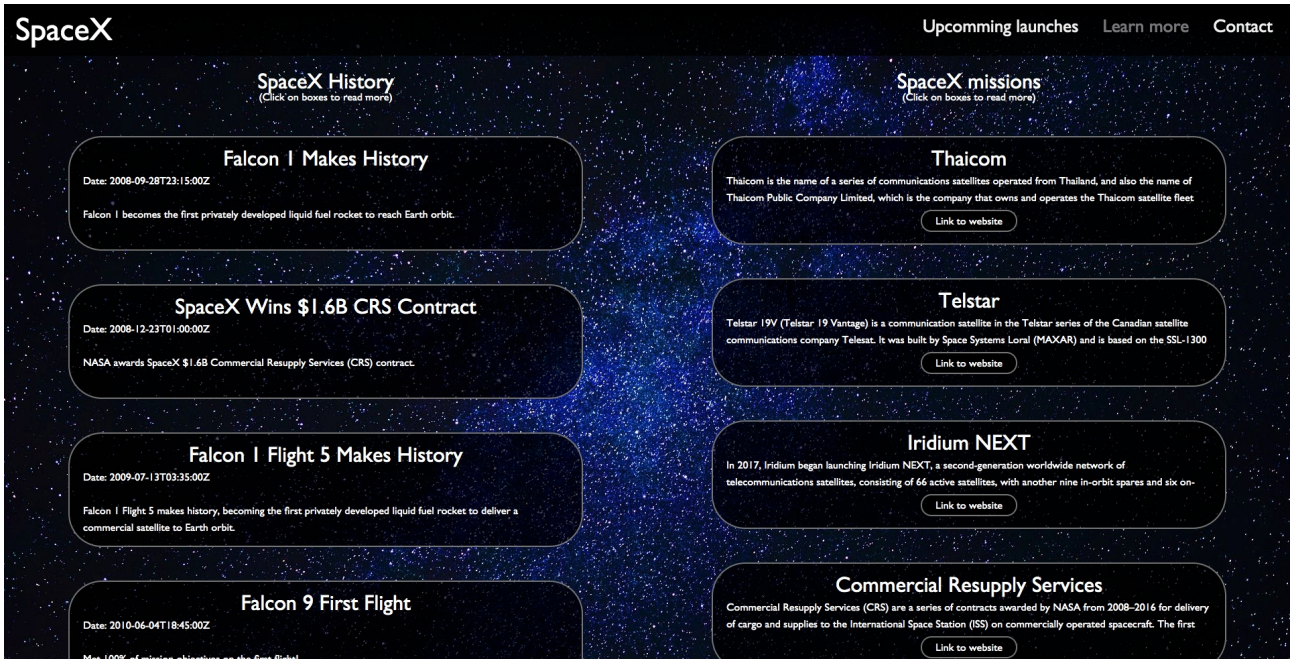
As I was making the prototype, I decided to drop the “about us” page, and replace it with a “learn more” page. The reason for that was that I didn't really have anything to write in the “about us”. I could have made something up, but that just seemed unnecessary for this assignment. I also felt like I could implement some of that information in other ways. A “learn more” page also opened up for other design and content possibilities.
I tried to use the background image from the mock-up, but I quickly realised that it didn't work. There was simply just to much going on in the image. I found another image that worked way better, just a “simple” image of stars and space. I chose dark colours for all backgrounds, and white colour for all text. Some borders etc are grey, but the main design are dark and light. I chose that to reflect the darkness of space in the background, and white is very readable on top of dark backgrounds.
I wanted a font that were easy to read, but not boring at the same time. I ended up with using the font “Gill Sans” which is a font I like a lot. I switch between font weights of the font at some places in the website, other than that it's all Gill Sans.

I then began creating the actual webpage. I started by creating a “wireframe” in html and css to make sure that I had functional and responsive solutions ready before adding any other content/text. The first thing I did was to create the navigation. I knew I wanted a responsive navigation bar that became a “hamburger” icon when the site were viewed on a phone or a tablet. I therefore made the navigation first, and added media queries in the css from the very beginning. The actual coding for achieving the responsive navigation is based on the teachings from w3school, but with custom styling and positioning.
(link to w3school: https://www.w3schools.com/howto/howto_js_topnav_responsive.asp).

Then I went on to create responsive “frames” for the rest of the content. I'm a big fan of using flex box as a way of creating responsive solutions, so a lot of the content are designed within flex boxes.

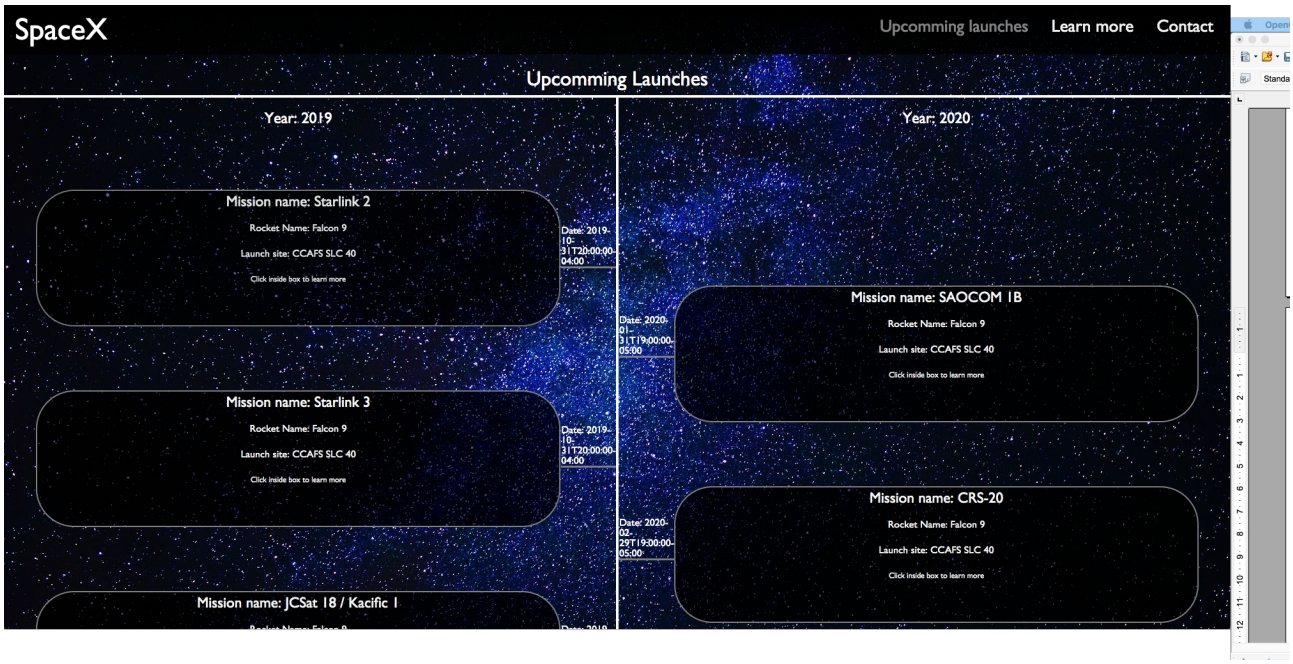
Both the page that are displaying upcoming launches and the learn more page are designed using flex box:

Learn More



Flex1 *Flex2*

Upcomming launches



Flex1 *Flex2 Flex3* *Flex4*

I like the way elements adjusts in flex box in a responsive way depending on the screen size, and that you can change the direction and the content to make content fit on smaller screens:

Learn more phone

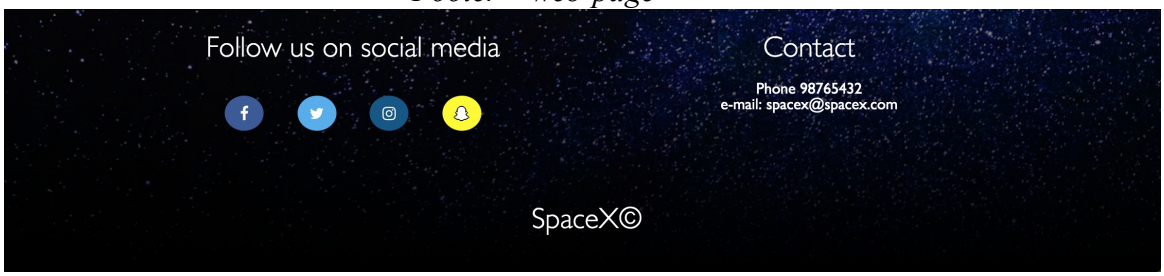
Flex-direction: column

Upcoming launches phone

Flex 2 and 3: hide

I did the same when creating the footer:

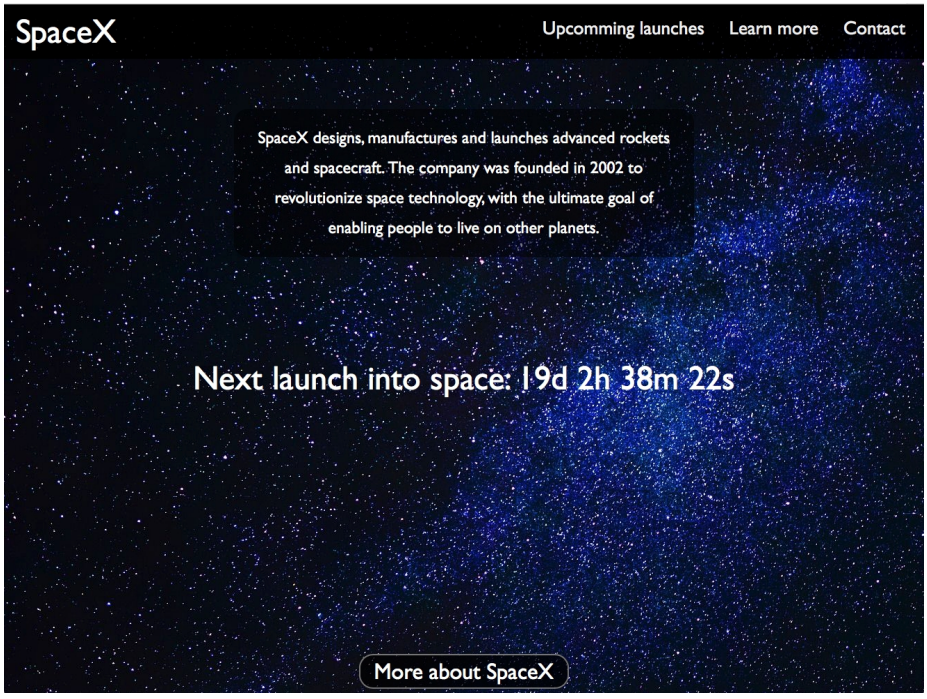
Footer – web-page



Footer - Phone

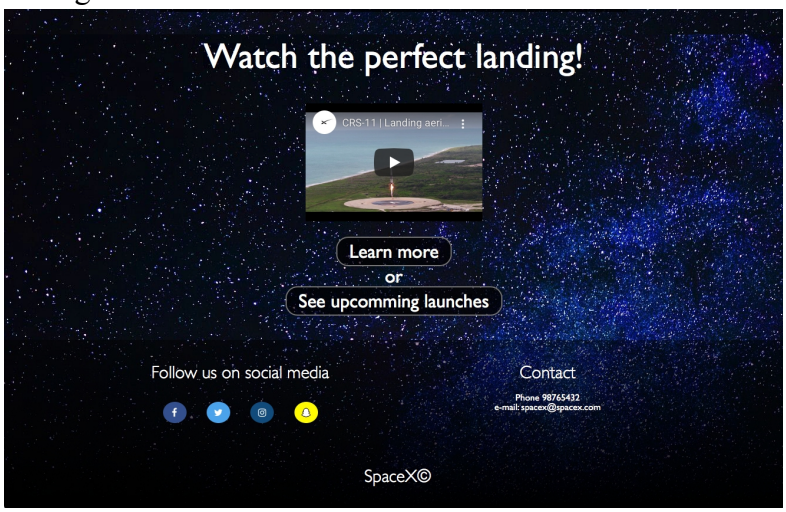
Flex-direction: column

I attempted to create the index page as I made it on the prototype, but I quickly realised that it needed some content that explained what the page was about. I therefore used some of the official text from spacex.com where they explain what spaceX is all about. I then added a placeholder countdown-timer (since I had no idea of how I were going to create a functional one yet) and a button that lets you move on to the next section of content. With all of that content in place, I felt that I had an index page that explained what it's about, how to navigate further and it had the look I wanted.



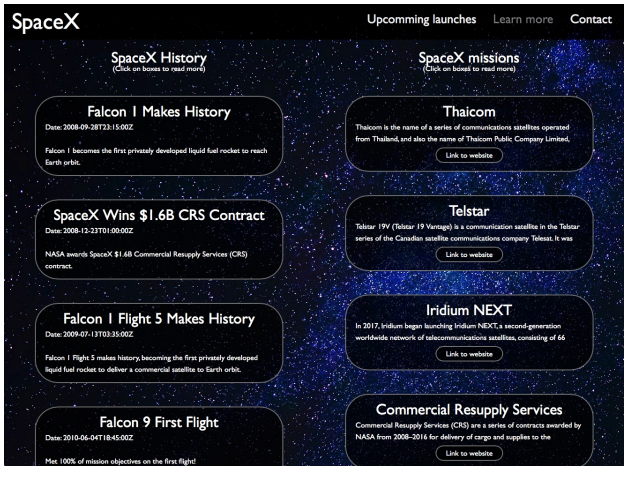
Index-page

I made the countdown-timer into a link, as I wanted people to be able to find information about the launch when they clicked on it. I didn't know what the link would link to, I just knew that I wanted that function to work at some point. The “more about spaceX” would take you further down the screen, but I wanted that transition to be “smooth”. Luckily I came across the css 'scroll-behavior: smooth option', which does exactly that (at least in most browsers). As you scroll down, you get to read a bit more about spaceX and there is a video of a perfect landing of one of their rockets. Finally you find a couple of buttons that can take you to “upcoming” launches or “learn more”. The idea was that you could navigate through the content on the different pages just by following the buttons and scrolling down the content.

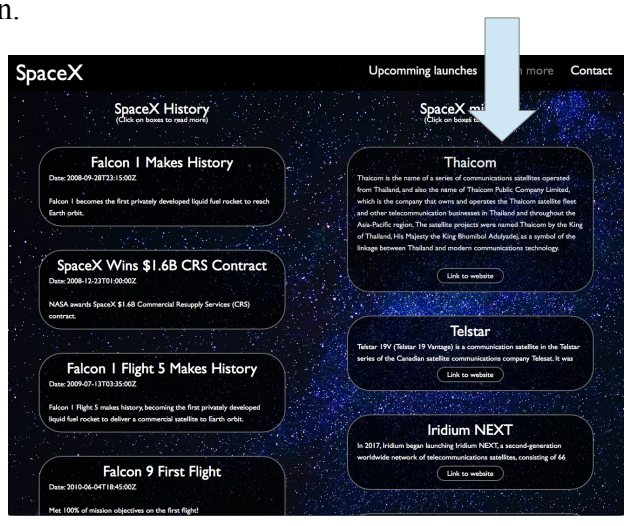


Bottom of index-page

For the “learn more” page, I made boxes that would contain information about spaceX historical moments, and their missions. The two categories were divided into two, and displayed on each of their side of the page. The content of the boxes were generated by an API call, and so the amount of content/text varied a lot. Therefore the size of the boxes varied a lot, and the whole design just looked messy. I then came up with the idea to set a fixed height of the boxes and use javaScript to enlarge the boxes to display all of the information if you simply click the box. That worked a lot better, and it gave the page a more consistent design.



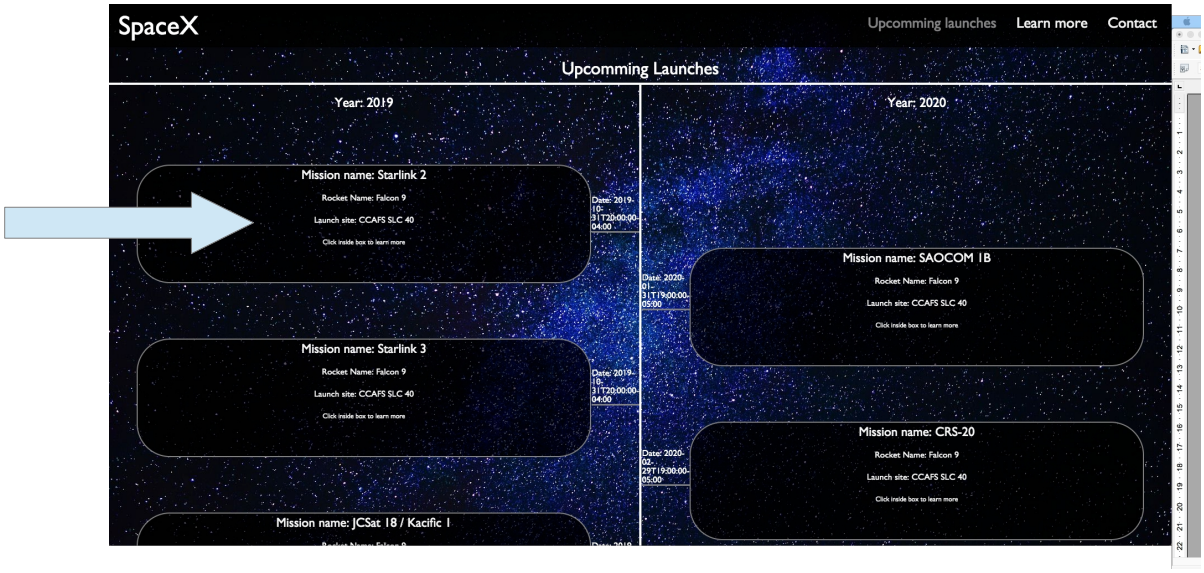
Learn more page



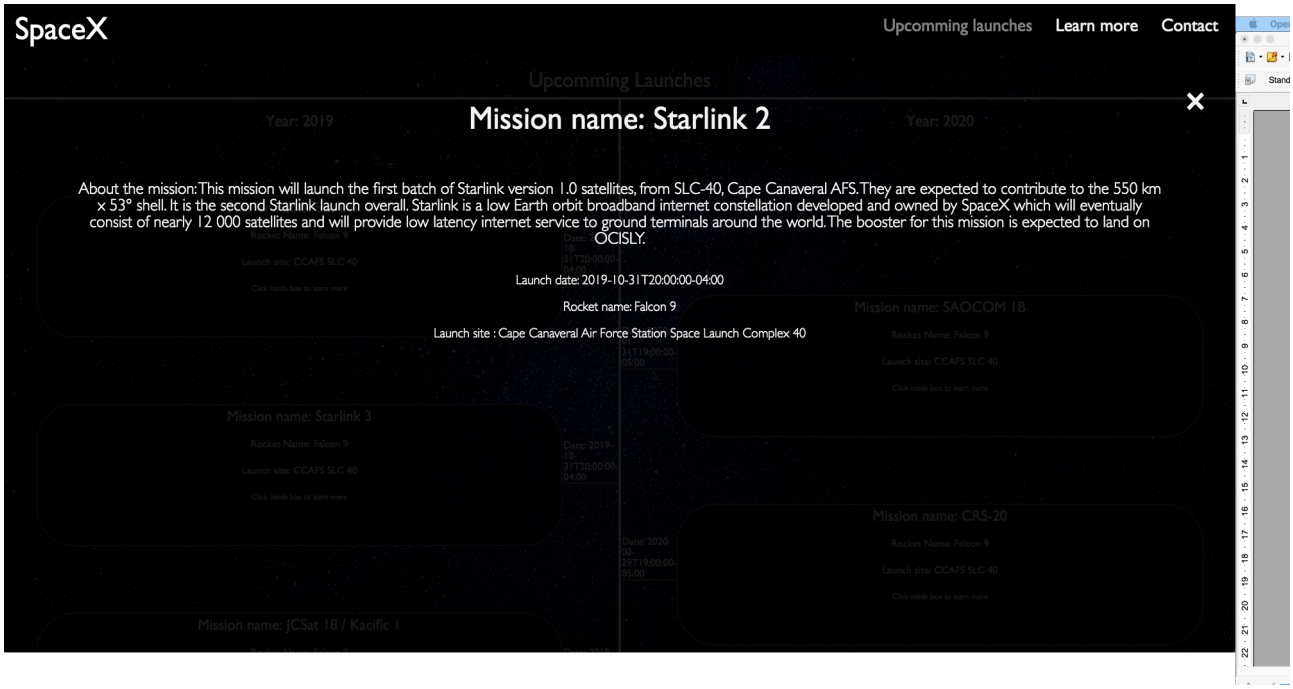
Clicked on a box

Some of the API's also contained a link where you could find more information, so I added that as a “link to website” button.

In the “upcoming launches” page, I had some similar issues with the content. All of the content is from an API, and the amount of content/text varied a lot. I also wanted to create a timeline for the upcoming launches, so I needed a consistent size of the boxes that were displayed. To achieve this, I wanted to attempt to be a bit more creative about how I would display all the information when you click on something, so I came up with another solution. The solution was based on a w3school tutorial on how to make on overlay navigation: (https://www.w3schools.com/howto/howto_js_fullscreen_overlay.asp). I used the same concept to create an overlay that displayed all the information when you clicked on a box of an upcoming launch.



Upcoming launch page where boxes can be pressed



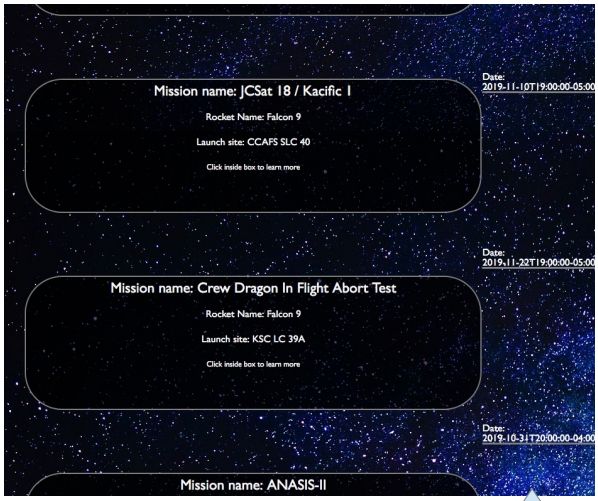
Content is displayed as an overlay when clicked

It took some time and effort to make that design work, but it's definitely one of the details I am most pleased about the web site. I just think I looks and feels nice when you can click a button, and information is displayed in a neat way! I think it works especially well on tablets as it is satisfying in a way to click on something and then there is some action happening.

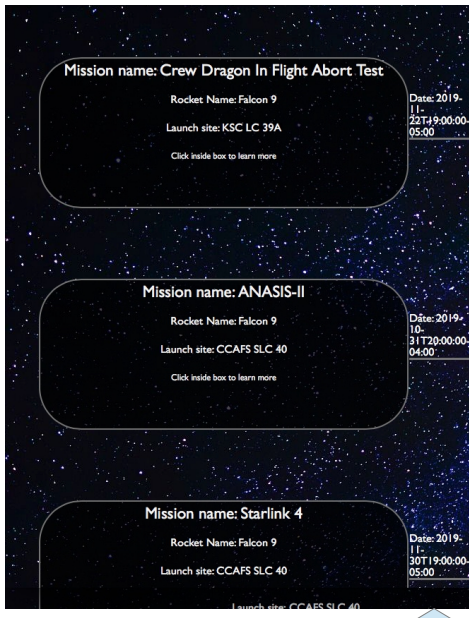
While creating these pages, I had to use, learn and experiment a lot with javaScript. I knew what I wanted, but I didn't really know exactly how to program it using javaScript. It took a lot of trying and failing by using API calls, running the result through loops, getting all the information in the right place in the html, making functions to make things behave in a certain way and a lot more. I started out with just one javascript file, but eventually as the amount of code grew, it became a lot to handle. I therefore decided to keep one script file that contained the javaScript that where used on two or more pages, and make a separate js-file for each of the four pages, containing just specific javaScript for the individual pages. That made everything a lot easier to keep track on. It might not be the best way of handling javaScript, but it seems to work quite well (I would love some feedback saying if thats a good or bad solution, and why).
The "upcoming launches" and "learn more" pages contained a lot of boxes/elements that needed javaScript to function, so I ended up writing simple function for each separate element. That ended up being a lot of code, but it works! There is probably a better way of getting the same result using fewer functions and less code, so I would appreciate some feedback on that as well.

The final touch to the design, I discovered pretty late in the process. I felt like it still was a bit challenging to navigate around (in general), and I didn't quite find the solution to how to solve it. I then let a friend of mine test the page, and eventually she said "I hate it when pages don't have the menu available at all times. It's so lame to scroll up all the time." Well, there was the answer I was looking for! I made the navigation to be fixed, so that it scrolls down on the page as the user does. It's a super simple solution that I probably should have used from the beginning, but as I had worked on the website for a few weeks, I suppose I eventually became "blind" to a simple and important solution like that.

I then used quite some time on adjustments and bug-fixes, but the website started to work well on all platforms. One bug I discovered late was a design flaw that makes the design of the "upcoming launches" break on Firefox. It turns out that Firefox displays some json objects differently than Chrome and Safari, which messed up the design:



In Firefox, dates are shown like this



In Chrome, dates are shown like this

That small difference makes the dates "hit" the boxes at the wrong place in Firefox, but it works perfectly in Chrome and other browsers. To prevent similar issues in the future, I think I will have to think of a completely different way of making the same kind of design.

Overall there are very few bugs that I can find, and the design and programming work pretty much as I want it to.

I used git and gitHub throughout the entire process, and used Terminal/git to update and save files as they were made. This is the first time I've used git throughout an entire project like this, and I am discovering what a great tool it is for backup, version control and more. If I were collaborating with someone, I would be a lot more precise with the text in the commits. In this case I made a system that worked for me, but later I will use more time to be super precise on what the changes actually are before committing a file.

Conclusion

This has been a big task with a lot of trying and failing, but I have learned a lot in the process. One thing I am continuously learning is what I can can achieve using javaScript. There is so much I still don't know, so in a way it's challenging to plan a project from the beginning. It's difficult to plan certain functionalities when I don't know how it's done or if it's possible at all, but I am definitely getting more confident as I am getting more experience using javaScript. For this assignment I have had for the most part a good idea of how programming problems could be solved, compared to earlier assignments where I often had no idea to where I should even begin to look. Being able to achieve the effects I wanted through programming has been a massive confident booster when it comes to programming. It has also been really useful to work a lot with API and Json, just to get more familiar with those technologies in general.

I am very pleased with a lot of the functionalities of the web page and how it is responsive across most platforms. Responsive design and javaScript manipulation of the DOM just gets more and more intuitive as I get more experience.

I am discovering that I really enjoy javaScript and programming a lot, and there are few things that feel as good as when I finally make a cool function work (often after hours of trying and failing).

It's been exiting to do a big project like this from start to finish, including all the steps from planning, prototyping, creating personas, designing, coding and testing.

References

Background image:
https://pixabay.com/photos/milky-way-starry-sky-night-sky-star-2675322/

Information about spaceX:
https://www.spacex.com/about

Link to w3school – responsive navigation:
https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

Link to w3school – overlay:
https://www.w3schools.com/howto/howto_js_fullscreen_overlay.asp

Link to w3school – countdown timer:
https://www.w3schools.com/howto/howto_js_countdown.asp

Json reader:
https://jsoneditoronline.org/

Image compressor:
https://compressor.io/compress

Html beautifier:
https://htmlformatter.com/

CSS beautifier:
https://www.freeformatter.com/css-beautifier.html

javaScript beautifier:
https://beautifier.io/