



Noroff

School of technology
and digital media

Technical Report

Project exam 1

Petter Iversen

Word count

Summary: 105 | Main text:3531



School of technology and digital media

Table of Contents

1. Summary	3
2. Body	4
2.1. Introduction	4
2.2. Main section of report	4
2.3. Conclusion	4
3. References	5
4. Acknowledgements	6
5. Appendices	7



1. Summary

The Project exam for the first year

Build a microsite for SpaceX or NASA. This should focus on space technology.

A microsite is a branded, self-contained site, usually on its own domain, with a single purpose and a limited number of pages. The purpose can be promotional or editorial, and may be linked to a specific event or period of time. A microsite is usually built as an addition to an existing brand website.

To tackle this project I'll create a plan, wireframes and prototypes. With me, I'll carry all the previous experience and also regrets so that I won't make the same mistakes again.

Link to github:

[GitHub - petterkoding/git_projectexam](https://github.com/petterkoding/git_projectexam)



2. Body

2.1. Introduction

I've chosen to build a microsite for SpaceX. The scope of the microsite is rocket and launch data. It will serve as an addition to the official website. The main focus will be to present rocket launch data from each launch, counting from 2006 until today.

The site will present launch data in a minimalistic manner and let users find rockets by searching or scrolling through a gallery.

Before any coding, I'm going through the REST API with Postman.

[GitHub - r-spacex/SpaceX-API: Open Source REST API for SpaceX launch, rocket, core, capsule, starlink, launchpad, and landing pad data.](#)

Endpoints of interest are "past", "upcoming", "latest" and "rockets".

Going through the API will let me get an overview of what's available and I can start thinking about how to group the categories together as in terms of app architecture for ease of use.

Target audience

My target audience will be rocket fans of age 12-99 years old. It's a broad audience, but the purpose of the microsite is also to let anyone collect launch data, simple and fast.

The goal is to present the data in an easy consumable way. The copy should be concise and easy to grasp. Personae/scenarios will be discussed along the way. As well as navigation and persuasion techniques.



Before going into the research I've written a functional spec for the website.

Functional specifications

Project scope

Goals: It's goal is to serve information about rocket launches, dates, images and highlight rocket data for rocket fans of any age.

Features: Rocket launches, rocket data and images

Deadline: 14th Feb 2021

Product overview

Rocket fans can go through a gallery of SpaceX launches from 2006 till today. Each rocket will display data about the rocket's details, such as engine type, costs, measurements, mass and images.

Homepage

The homepage should explain what the purpose of the site is.

Navigational links to the other pages, with direct links in the hero section. The most important parts are the "latest" and "upcoming" launches.

Some "nice-to-haves" are the current people in space and location of the ISS with links to read more.

Launches

Every rocket launch displayed in a gallery. With ease of use, the user can resort to searching by name instead of scrolling through.

Launches- details

To look closer at each launch information

Gallery (Photography)

Gallery of photography with description.

Each image must have a unique alt text and description.

Contact



Contact form
Send email or call information
Social media links

Use cases

example:

User A wants to look up specific rocket information for his school project.

User B wants to find recently launched rockets to learn more about space tech.

For more use cases/scenarios look at personas > scenario.

Requirements

What does the microsite do?

Present data about SpaceX launches, launch dates, rocket details in an easily consumable way. To cater to the wide audience, interested individuals in any age from 12 to 99 years old can read interesting facts about asteroids.

For individuals interested in astrophotography, the gallery page has a wide offer of detailed images.

Configuration

Creating the NASA microsite will require a text editor, such as Vscode, to create all the HTML, CSS and javascript functions. Web host to host the site on and also a FTP client to save all the files on.

Non-functional requirements

ISS station location and current number of people in space.

Error reporting:

API

In use cases where errors will occur during API calls, the error messages will be displayed as HTML on the site. Giving the user feedback that there's been an error.

Contact form



The contact form will handle input data and validate them through javascript functions and regEx expressions, like email validating.

Users will get feedback on invalid data input as a “red colored” message near the input area itself. For valid data, the feedback will be a “green” line underneath the input data.

2.2. Main section of report

Planning

The most important part of planning is to make sure that the architectural design matches the scope. If not, the app will be difficult to use and the target audience will abandon it.

I’ve made a gantt chart to lay out the work schedule for each block. Each block will represent a crucial part, such as planning, prototype, html, css and javascript.

Research

To reach such a broad audience, the design will be simple but most importantly, good hierarchy and easy navigation.

Most of the inspiration comes from the spacex official website. I followed some of the same principles such as, good use of whitespace on bigger screens. Minimalistic, clean edges. Mostly black, white and greyscale. But since this is a rocket site, I decided that the “brand” color for SpaceX Cowboys should be orange. As the fire from the engine has an “orange” tint too!

The homepage hero section matches the way the official website uses the full width and height to make it immersive. As the space is!

Graphic design

From design to coding

To make the design easier to code. I decided to use a two column layout for web screens and one column on mobile. I’ve mostly decided beforehand what the CSS should be for each column or row of HTML. That way it's easier to reuse a class for another section somewhere else.

What I’ve learned from studying the official spacex website is that it utilizes short lines of copy, large buttons, full bleed background images and good use of space between elements.

I’ve done full bleed on the hero section’s background image, but not on every section of the site. I wanted to use more whitespace instead of background images that can make the copy hard to read.



Typography

For the sake of being futuristic, I've used a sans-serif typeface. To make the font responsive I've decided to use "font-size: clamp(x, x, x)" which takes three size units for smallest, ideal and largest font size. A good line height of 1.6 to add space between sentences.

Accent color

Throughout the site, I've used a line of orange underneath text, headings or as border on images to grab attention, highlight new sections or important parts. It helps a lot on the hierarchy and helps the user know where to focus.

Hierarchy

H1 headings and subheadings

Most of the paragraphs are written in a slightly lighter dark-grey color. While the Headings have the darkest grey color. All the inputs have black and color.

I'm trying to have one main CTA per page. So the users won't get confused about what they should do next.

WCAG

The text is legible and responsive. Good text contrast across the site while maintaining "brand" identity. The animations are kept to a minimum to not disturb those with sensitive eyes.

Legibility

For accessibility and legibility, good contrasts against the background were checked in the chrome's dev tool. The hero text on the Homepage started out a bit in low contrast to the background-image. I fixed that by darkening the image in Lightroom with a radial mask and using a ::before on the image to cover it in a black, low opacity color. After the edit, the background image comes out more mysterious and adds to the feeling of being in space. Each image has alt text in case the user uses text-readers or the image hasn't loaded successfully.

Darkmode

The Launches page has a lot of white in the background. And I know from personal experience that it can be quite bothersome if you don't like the strong light it gives at night time. That's why I've implemented a "DARK MODE" button to invert light to dark colors.

Semantics

For the header I've used a NAV with UL list for the navigation. The navigation collapses into a burger menu at 960px. It's absolutely positioned outside the html document and slides in on user click.



The body has a main, section, aside and footer tag. Each element that is of the same theme or object is contained within a section. That section gets a classname for easier recognition and styling purposes. To work with the DOM, some of the classnames get a more “explaining” name rather than a confusing universal name.

And as I’ve explained before. Each section that is positioned as rows will have a wrapper of flex-direction row and column on mobile. That way it’s a reusable class and will need less CSS to maintain responsiveness.

CSS

I always try to use classes instead of repeating the same styles on the same elements. I’ve styled a couple of global elements such as headings, paragraphs, buttons and links at the top.

Other than that, I work my way from top to bottom. Meaning first NAV, HEADER, BODY, FOOTER and then MEDIA QUERIES. Some CSS at the bottom are used for the JS NAV animations.

I always group the CSS the same as the HTML, starting with the container and going down. The lines of codes ended up being a bit long this time, but as you can see. I have lots of hover states, ::before/::after elements and animation keyframes to cover. At the end of the project I gave myself time to add some design flair to the website. In my opinion, it helped the design stand out while the hierarchy was made even clearer.

On the main buttons, I did a similar hover state as the official site has. With transform on the ::before element to “fill” the button’s background until it’s fully covered.

The NAV Lis has a sliding underline to show that it’s being hovered. The active page has an orange underline, positioned at the same place.

The gallery

Images in the gallery and on the Launches site are wrapped inside a container that has almost a display grid with grid-template-columns: repeat(auto-fit, minmax(400px, 1fr)); It makes it almost not necessary to do any media query. I only adjusted the first value of minmax to a smaller one to fit inside the viewport width on mobile.

SEO

To optimize for search engine results, I’ve tried to make each H1 unique and explain what the page is about. The top paragraph should explain more in detail what’s to come.

Having a contact page increases the chance of local hits.



Personae/Scenario



Ed, 19 years old.

He's interested in games, space and astronauts.

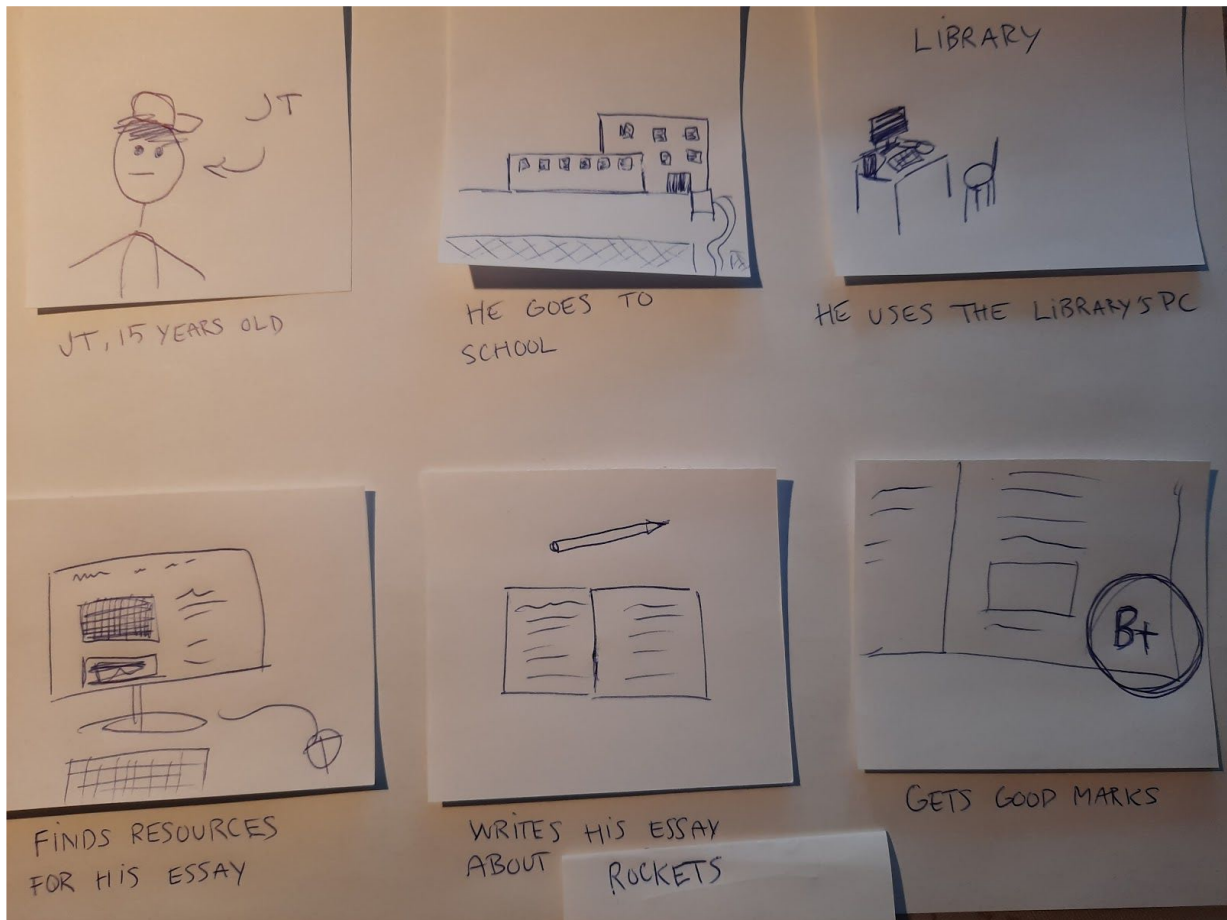
Goals: Learn more about rockets

Concerns: Difficult to find good resourceful websites that are easy to navigate.

In Ed's environment, being at home, he's casually browsing, means he doesn't want to visit websites that are cluttered or hard to use on mobile devices such as an iPad. It's important that every site is responsive and scales text and buttons accordingly.

Without being responsive, it will fail the majority of visitors that's on mobile.





JT, 15 years old student.

He's interested in aircrafts and space related photography.

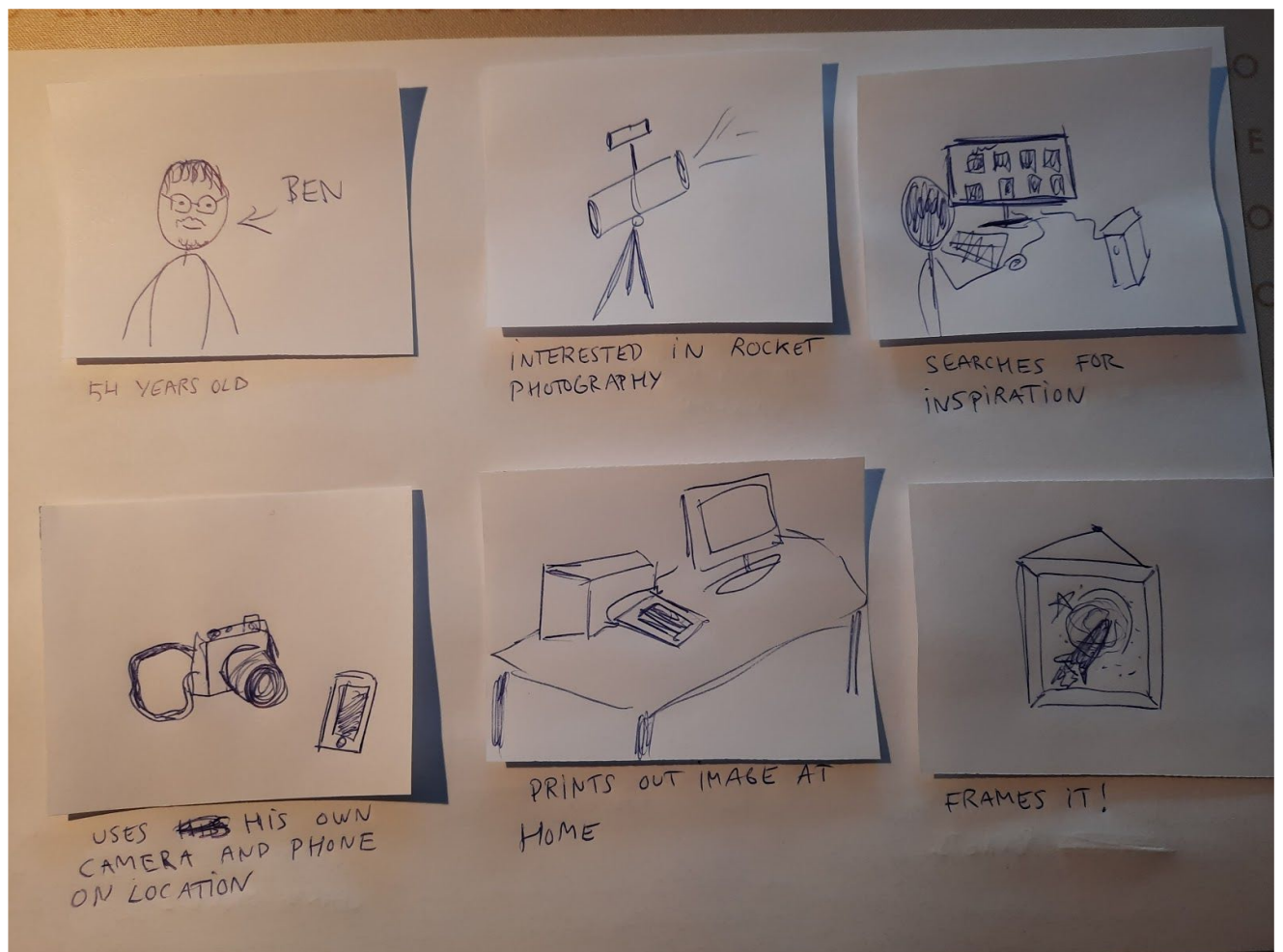
Goals: Get good grades on a written essay about rocket engines

Concerns: Not getting good marks on his essay

For a student, the site should be clear and tell visitors what to expect. Even for ages down to 10 years old, the site should be intuitive and easy to understand.

By having a cluttered navigation, difficult jargon and messy layout, the potential of scaring off the visitors will double. As the visitors scan the site in just 3-5 seconds. After that they get a feeling of "should I stay or leave".





Ben, 54 years old, retired firefighter.

He's interested in Tesla stocks, SpaceX and photography.

Goals: Frame and display his own photography in his living room

Concerns: Lacks motivation to invest time in his hobby.

For Ben, it's important that the text is big enough on web screens. He's looking for SpaceX rocket images. Having alt text will aid search engines find the page easier. And by having a well designed Gallery page. It will earn its trust as a place to return to.

Interface design

The prototype(s) can be viewed in 2 separate files. For web and mobile. It helped me understand how I was going to structure the website and how the navigational flow would be. I did a cardsorting



School of technology and digital media

Affordances

CTA

The buttons stand out with padding and border around the text. The hover state shows that something is changing. The cursor changes to a pointer that makes user's expect something is clickable. Although it's a hidden affordance, user's will expect it to be a button because of the styling.

The animated chevron pointing down tells there's more content.

At recent launches, there's 3 buttons and a slider beneath it. The slider is clickable and moves in sync with the images. It lets users know when they reach the end or the start.

The search bar stands out with different background color and changes on user click.

Persuasion

If the site looks badly designed, the visitor will leave within seconds. And secondly, if the content isn't clear about the purpose, the business will fail as well.

Credibility

The site looks professional, minimalistic and resembles a site without ads, pop ups and other annoying fields. The site should be purposeful and be clear.

Reciprocity

If you subscribe to the newsletter you'll get updates and notifications on next launch dates etc. In return the SpaceX Cowboys can store the email in their database.

If you participate in the APOD, by tweeting and following the account, you'' have a chance at winning a t-shirt and a mug.

Card sorting

To get an overview of the navigation and group each related element together. I've done a card sort before designing the prototype.

Design values

The site is designed to be used by a range of people. Under construction it has made sure to make no physical harm to users, such as epilepsy. The contrast follows the WCAG guidelines to avoid eye strain and not being legible.

The site is purposeful and designed as a rocket launch data where users can find rockets by going to the dedicated "Launches" page or at the Homepage.



Home

Main nav
Hero section
Latest launch
Upcoming launch
Timeline info (recent)
Other sections linking to internal pages

Launches

List of past launches
Show name, date, image etc.
A link to details

Details

Images
Description
Rocket details
Ext. links

Gallery

Images
APOD (nasa api)

Contact

Contact details
Send email
Contact form
Phone
Social Media

Wireframe

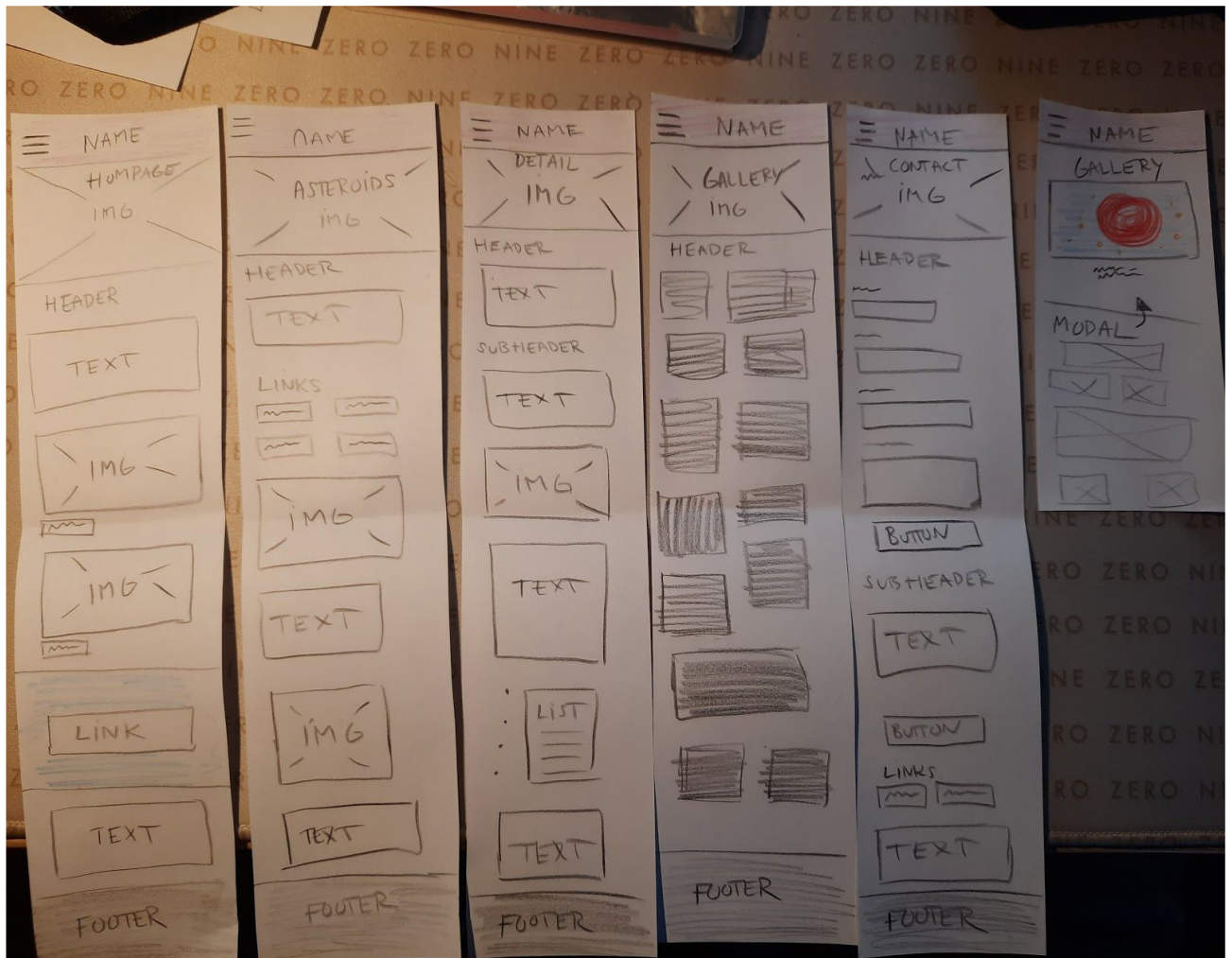
This wireframe shows the flow of the website. The only difference is that the web version will have two columns instead of one. I first decided on doing an asteroid microsite but changed to SpaceX because it felt more interesting and design wise it was more challenging. My initial idea was to have a hero section for each page with a full covering background image. I ended up only doing that on the Homepage to make it more unique. And make that page stand out as the Homepage.

See the prototypes made in XD in the files. Link to prototype

<https://xd.adobe.com/view/480c5fae-0c41-4e25-acb0-a2c363927c68-1d5f/>

<https://xd.adobe.com/view/ef011441-9434-4bed-bcbf-0812a2e7198d-ba52/>





Javascript

To make the report easier to follow, I'll explain the unique JS of each page by section.

REST API CALL

Not much to say here other than giving each function CALL a meaningful name and handling all the errors with a `displayError` that is scripted before the CALL.

For each call, I've used the Postman to make sure I've got the correct endpoints or object names. Because there are multiple CALLS on some sites, there's more than 1 loader to indicate that content is coming and to advocate good UX design. Because there's a few of the loaders, I made them simple and with only CSS for least demand on performance.



Homepage

Recent Launches image slider

It wasn't planned at first but ended up trying to do it right. I couldn't make the slider touch responsive (as it should be). But I think that's outside of my expertise for now.

The function is easy and must work with the grid. The grid gives each image a certain width. To let the user know what there's more content, I'd let the 4th image peek out. By giving the chevrons and the dot an EVENT LISTENER, that changes the container's position. The position is declared as a CSS property of `translateX(someNumber%)`; The `translateX` moves the container from left to right. With a transition duration, it makes it slide smoother.

Right below the images is a slider indicator that tells when you reach the end of the images. Also the orange border at the right (beside the last img) should also give that feeling.

Launches

Searchbar

This functions well. It takes the value of the input, then compares it with the names of the rockets in the array. If the input matches the a rocket name, then populate the UL with a LI element created in the JS. If nothing matches then empty the UL and hide it.

Each LI element has an anchor tag with an id matching the rocket, linking the user to the details page.

Load more

I've given the launch cards (total of 116 atm) a class of "items" and added a property of `display: "none"` to all the launch cards. Except for the first 8 cards. Those are visible. Therefore, the `currentItems` equals 8.

To be able to load more. I've added an EVENT LISTENER that loops through the `currentItems` and adds `display: "block"` to the next 8 "items" on every click.

The if statement checks if the `currentItems` are higher than the `items.length`, When it is, hide the LOAD MORE button and update the counter for how many rockets that are visible.



Script

Navlist (burger)

The burger menu has an EVENT LISTENER that listens to a click. It toggles the burger icons to be positioned back inside the “frame” of the viewport.

It then loops through each NAV LI and checks if they have animation applied. If it has, remove it. If it doesn't, then set the LI's animation style as such and give the property an animation delay, increment the delay for each LI by dividing the index of the LI with 10 then adding an offset of + 0.3s.

Animation name + duration + easing function + iteration + delay

The animation keyframes are specified in the CSS with transform translateX and opacity properties.

Contact form

The form on the contact page has input validation and regEx on the email input. If the input passes validation, the red underline is removed and valid styling is added.

Each input has an error message which explains the reason why it's not valid. Each message is positioned close to the input it belongs to.

Each input value is checked, and must be at least a few characters long to pass. That way users can't submit empty values.

Smaller JS functions

Across the JS files you'll find if statements and comparators that check if something is of the same value, object or src. Let's say it checks if a certain rocket has an article. If it doesn't then show a link to a wikipedia page.

Rollout

During the last process I used the console to check for errors in the JS functions and liveserver on both the monitor and the mobile at the same time. That way I can work more efficiently. The most important part was checking the media queries and responsiveness.



I was careful with the paddings on mobile, as they can screw up the flow and make things float out of the screen. Because I had an absolute positioned burger menu, the html and body got an overflow-x: hidden.

I selected the breakpoints and scaled things up or down to make it legible. Since almost all of the sections used the same class names. Doing the media queries becomes a piece of cake. But because I had a lot of specific styling to the child elements and all sections had different child element styling, the CSS became longer because of that.

By checking all the containers, widths and margins. And making sure the javascript was working. The site was ready to go live.

2.3. Conclusion

Self evaluation

After working on this website from scratch over the course of 5 weeks. I've taken my own evaluations from previous assignments and used them to not fall into the same traps again. I think the CSS file is a bit long, but there's also a ton of styling to each page. Especially to elements such as ::before and animations.

I think I did pretty good at this exam. To me, the site looks well designed, has clarity and uses consistent design on every page. My CSS class names are reused on different pages. That was a main goal to follow the DRY principle.

I've become more familiar with javascript and used a self exploring method to find what's working and what's not. I wanted to use the least amount of copy written in the html document. Rather use dynamically created html from the REST API calls. I had an interesting time exploring the objects in the response to find what's usable to my site.

I can still have difficulties with knowing which size unit to use for images. Therefore it's a good idea to get better in that area. But I'm glad the grid minmax handles most of the work at the gallery page.



3. References

For inspiration and information

[SpaceX](#)

[Official SpaceX Photos | Flickr](#)

[SpaceX - Wikipedia](#)

REST APIs:

Main API

[GitHub - r-spacex/SpaceX-API: Open Source REST API for SpaceX launch, rocket, core, capsule, starlink, launchpad, and landing pad data.](#)

Secondary, for the APOD:

[NASA Open APIs](#)

Current people in space and ISS location:

[Open Notify \(open-notify.org\)](#)

MDN:

-Code and syntax

[MDN Web Docs \(mozilla.org\)](#)

Noroff FEU:

Lessons and scrimba videos

Linkedin videos from Lessons:

Personae, Affordances, Persuasion

<https://www.linkedin.com/>



4. Acknowledgements

Start writing here

5. Appendices

Start writing here

