Problem Statement (BRIEF)

I am going to create a multi-page games website that promotes the game: Sunnyland.The website is to tell the end users others why they would want to play it and inform them in general about the game. The website must have external CSS, optimised images, multiple pages, easy to read and navigate and finally use relevant web conventions. The target audience for the website is people who enjoy gaming and are interested in platformer games, who are aged around 8-16 years old.

Implications

A copy of the screenshots are in Docs/Screenshots/Implications/

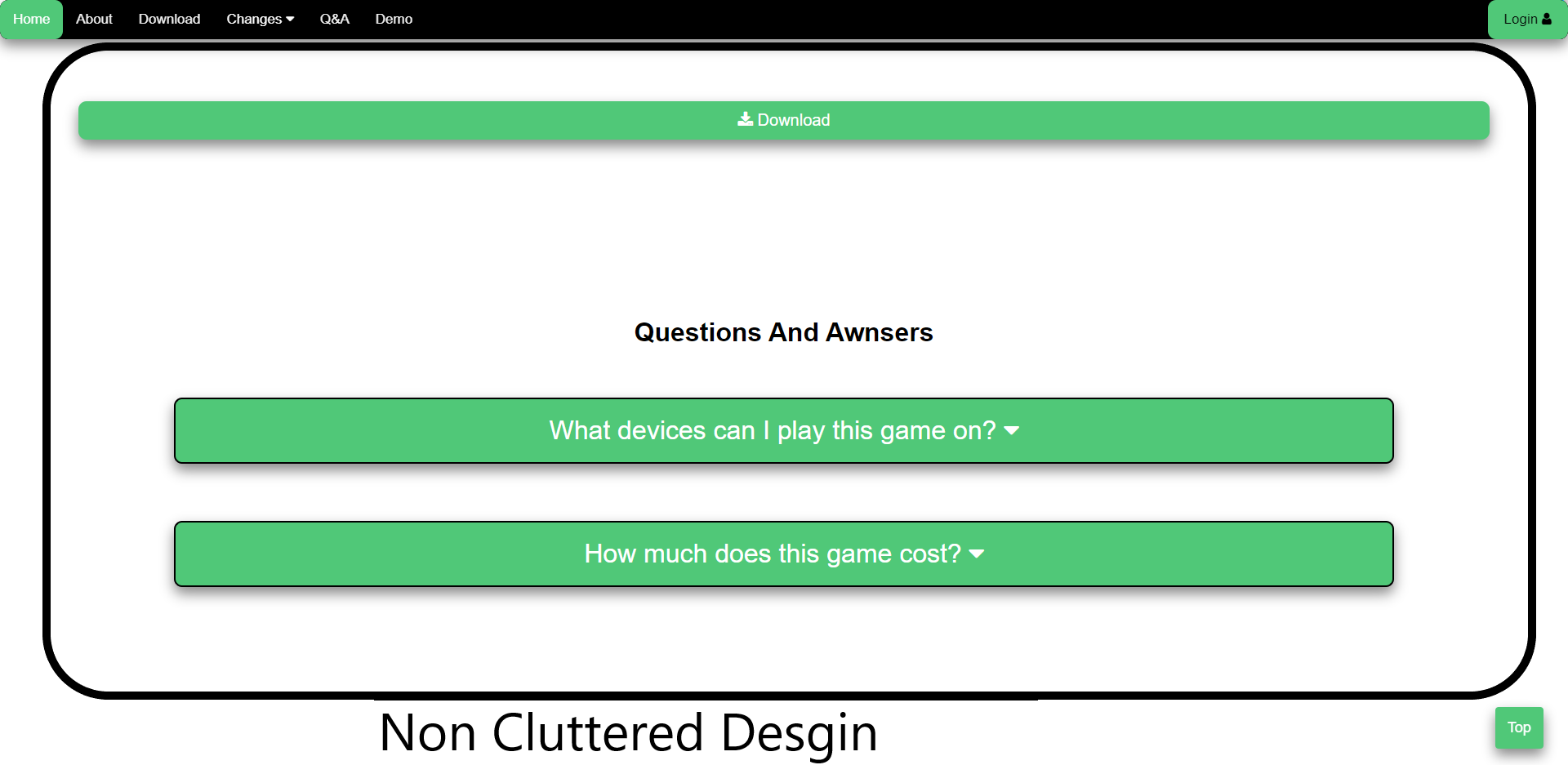
## Aesthetics

**Definition**:

Aesthetics is used to talk about beauty or art, and people's appreciation of beautiful things. products are chosen for their aesthetic appeal as well as their durability and quality.

**Use**: The website has a simple, modern and minimalist design and its layout is non-cluttered and easy to navigate. Transitions and animations are used to make the user experience smoother and less sharp and snappy. The website keeps the same font, theming and colour palette across each and every page so that when navigating between the pages of the website the user doesn't feel like they're in a completely different website.

**Images**:



## 

## 

## 

## Functionality

**Definition**: The quality of being suited to serve a purpose well; practicality.

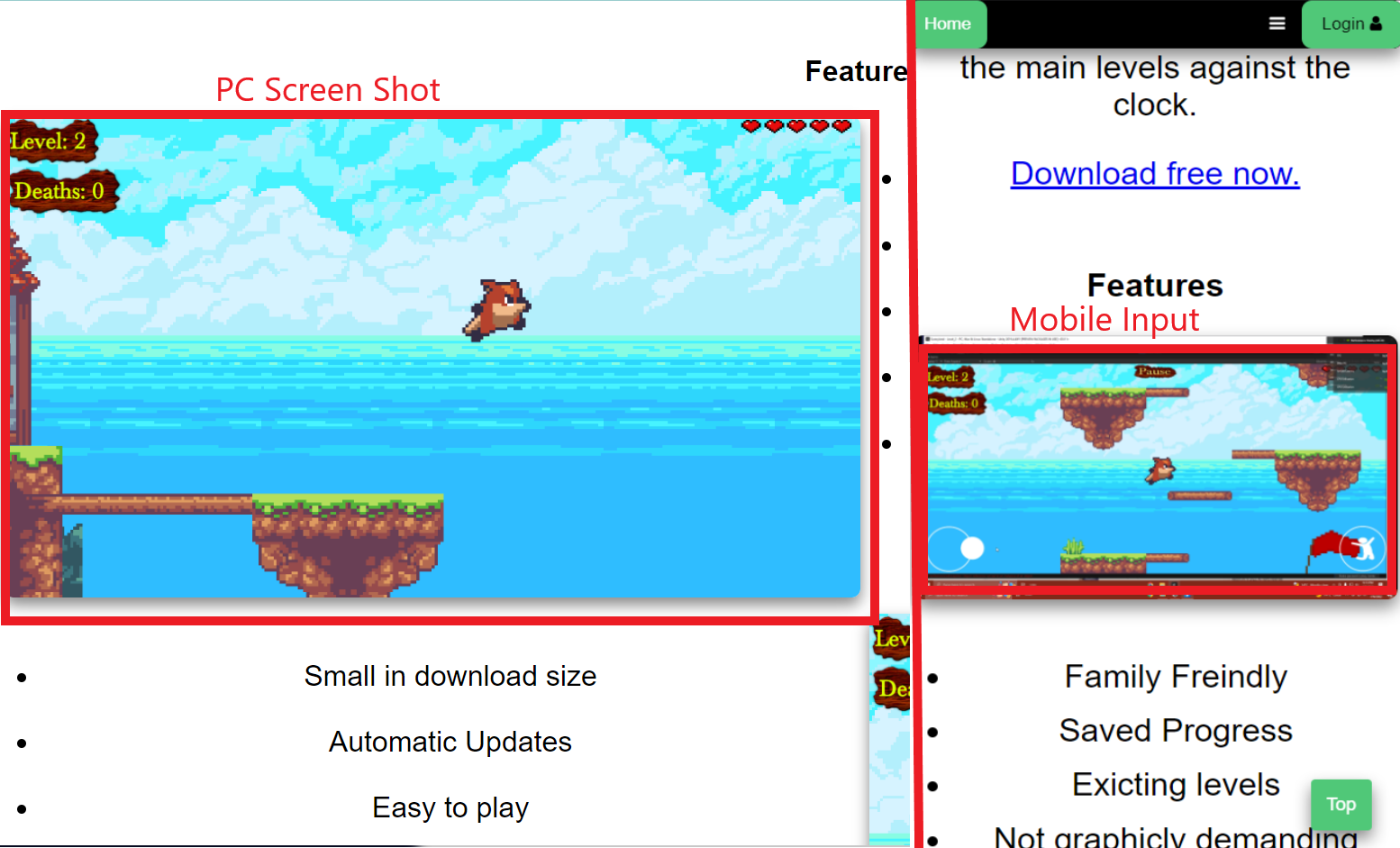
**Use**: The website is functional as JavaScript is used behind the scenes to fulfil the purpose of informing the end user about the game. It is used in multiple places such as the changelog to automatically generate the HTML from the JSON file. Another place where the website functions is that the download buttons change to launch the game buttons if the launcher is installed.

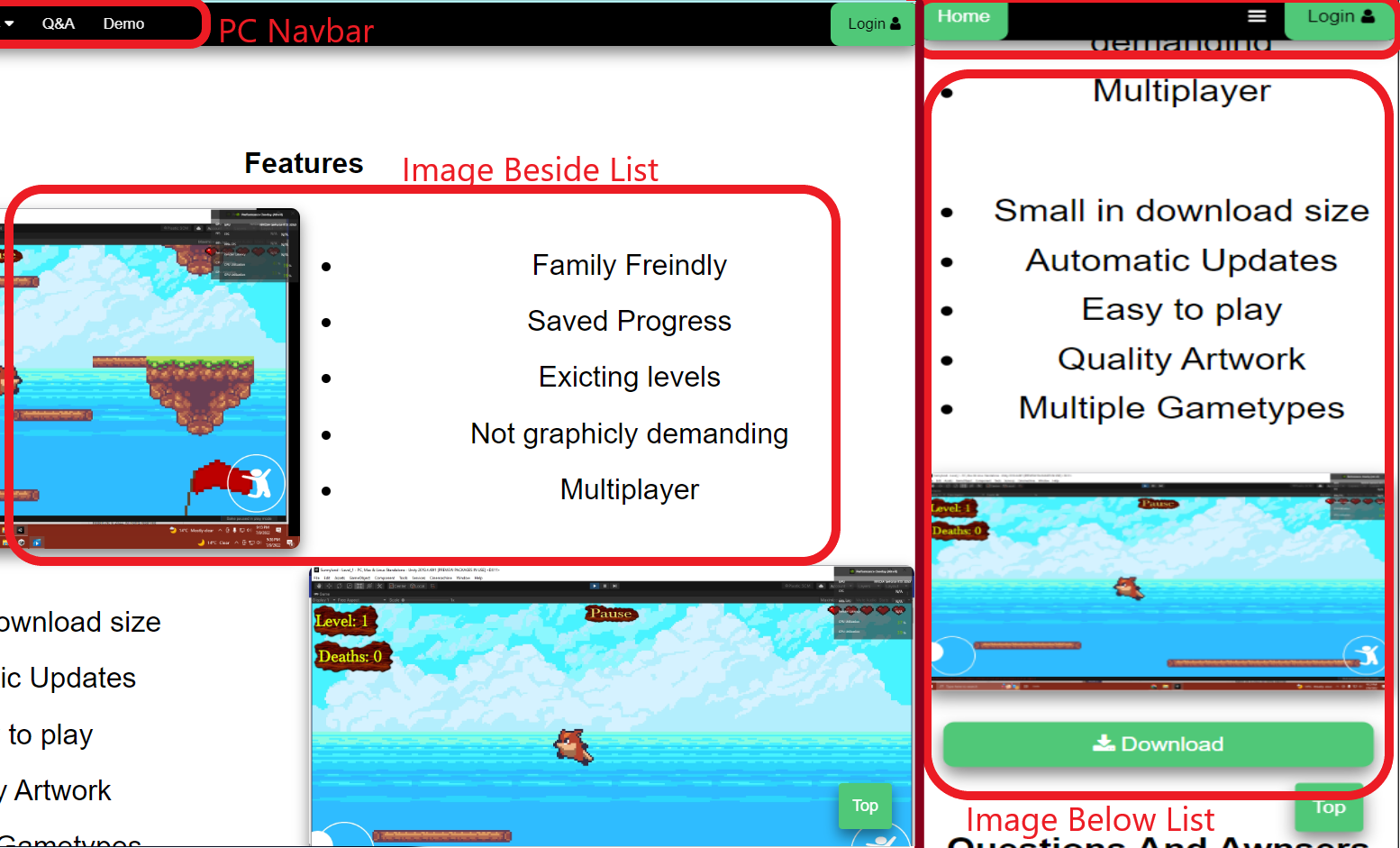
**Images: **

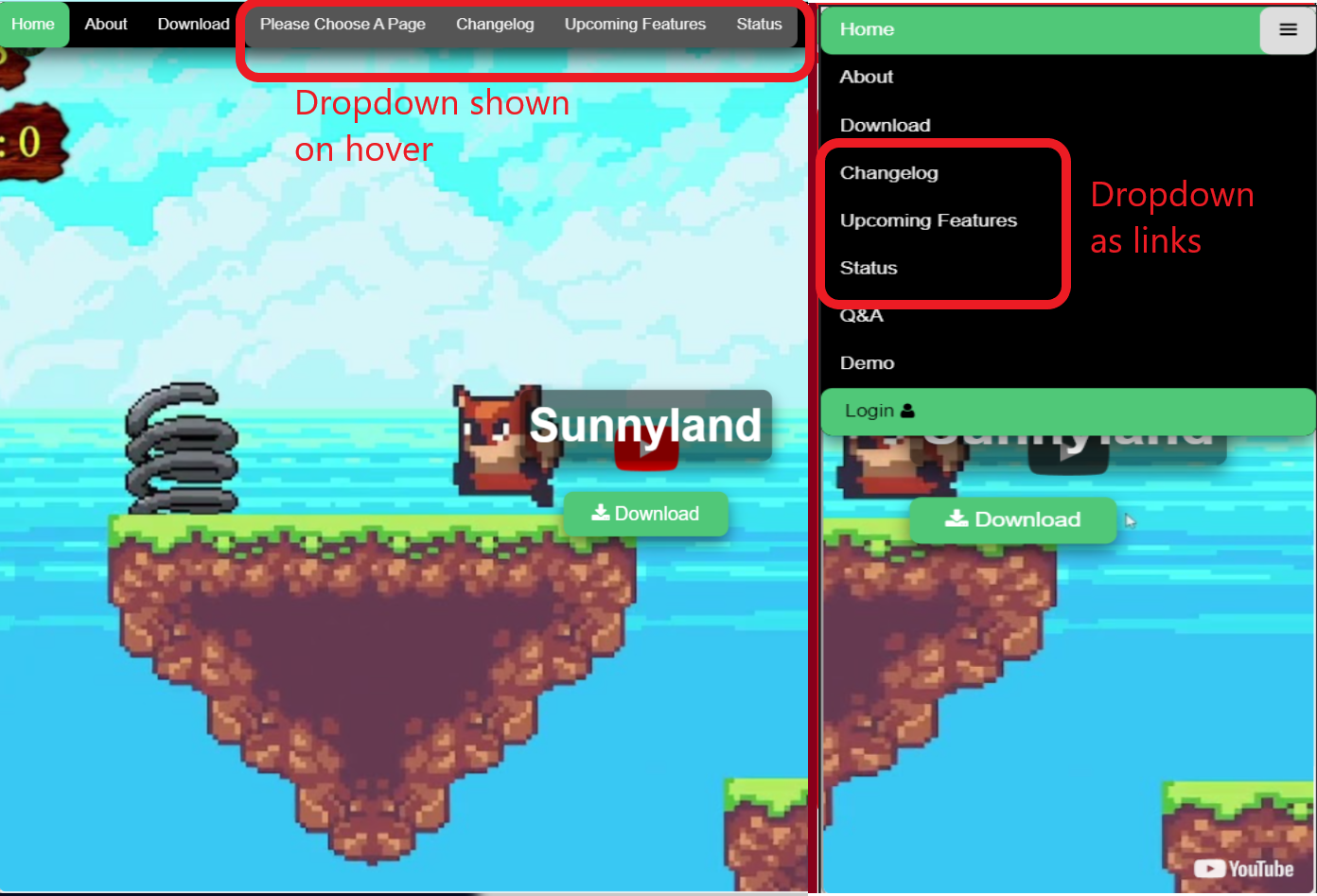
## Usability

**Definition**: Usability refers to the quality of a user's experience when interacting with products or systems, including websites, software, devices, or applications.

**Use**: The website is accessible and responsive to mobile devices and pc devices and it changes for the platform, for example on mobile devices the images are lower quality for a smaller screen and taken from the mobile version of the game or another example is the layout changes so that it fits on the screen. The download buttons also change and lead to the correct download site for each platform. The website also caters for mobile input as hovering doesn't work as well as it does with a mouse so the drop down menu is changed to be linked.

**Images**: 

****

****

Logs (summarised)

* Everything is logged in<https://github.com/maxtyson123/12_DGT_PYTHON/commits/main>
* If needed I can try to find video history of some of the earlier versions.
* Screenshots are labelled and located in Docs/Screenshots/Versions/
* Website versions are not the same as commit number, Versions aren't page descending either.

### Setup Docs

**Website Version 0:**

In this version I first set up a pdf brief using the template I found online using elementor. I then designed the plan for the index in Paint, using different colours to describe different features. After that I copied the header from the previous paint workspace into a new one labelled ABOUT and filled out the relevant information. After that I made a template page so it was easier to design the plans. Using this template I then filled out a plan for each and every individual page.

### Index.html

**Website Version 1: Create and work on Index.html:**

First I created the index.html file and all the other pages files for the site. I then made a colour palette to help with the site theming.First I made the top nav bar menu to navigate between the pages, this uses javascript to make it responsive by showing a button to expand it on mobile. After that I added the dropdown menu as shown in the plan.

**Website Version 2:** **Making the menu sticky:**

First I centred the Menu bar's links and then made the menu sticky. First I had problems where the drop down content is hidden behind the page so I had to fix this by adding it to the js: dropdown.classList.add("sticky"). After that the css made the width 100% so I had to make a second css class called sticky-drop down so that the drop down content doesn't take up the entire page width when shown. My next problem was with the mobile view. I tried adding in the js but that didn't work so next I changed the js to toggle the responsive class and that worked!

**Website Version 3: Youtube hero video:**

I managed to get the video done with no errors but then I came across the fault that the menu bar is hidden by the video until you scroll downwards to the content which is not wanted. I then added to the main css property (.Menu) z-index: 2; which fixed it on the desktop view. I quickly tested if it works on mobile but to my surprise the button that lets you view the menu is hidden so I had to fix this. While attempting to fix this I realised that no my little z index would not fix this as it was the sticky nav that was causing this. So I read up about it and then fixed it by ad position absolute. But we still had the error of the mobile button not showing up so after a bit of trial and error i fixed it by removing some css code:

**Website Version 4: Bug Hunting:**

Some bugs I found after testing:

MOBILE:

-hover doesn't work

- that also affects the drop down

-login button doesn't show up

DESKTOP:

None

**Website Version 5: Fixes**:

To fix most of the mobile bugs surprisingly all I had to do was remove the css part that hid the login button on smaller screens. This CSS element was going to be used later but I forgot about it and it turns out it was the reason for multiple errors in my code. I then made the video loop by adding “&loop=1&playlist=KAzCsr4Ew70”. After this I found another bug where the changelog text in the dropdown would show on mobile views when it should be hidden until button press. I fixed this by removing the video container class from the media view specifications and that seemed to fix it for some reason. After that I cleaned up my css and html code. Now I noticed that the video was not found so I fixed it by removing the accidental “-VC '' at the end of the looping parameter for the iframe video. Now the changelog dropdown isn't working on load but will work after it becomes sticky. So I had to fix this by making the dropdown content constantly dropdown-content sticky-dropdown. Just after I thought I was finished I realised the Q&A link was directing to the upcoming features page so I modified that to be the correct a href value.

**Website Version 6: Buttons:**

First I added a transition duration to make the site look better, then I tweaked it so it would work with js using -webkit-transition: all 0.5s ease-in-out;. After that I made a scroll to the top button and moved the JS code back into main.js (earlier it was in the html file because it was being buggy and wasn't detecting the window scrolling from the external file). Next I made a download button for windows and then if your device goes to mobile view the download button becomes links to android.

**Website Version 7: Brief description:**

First I had to fix the scroll to top button not showing up by moving all the js that I had previously moved back to the html file because it was being buggy and wouldn't work externally. I'm guessing that was just my browser loading weirdly. After that I added custom scroll bars to better fit the theming. Next I began work on the brief description after adding on hover tips to various buttons and links using the title tag. For the brief description I just centred a <p> element and wrote directly into it which means I can finally remove the placeholder text that I was previously using to test stickiness. Next I went and worked on the Images and bullet point list for the features. This is just 2 columns and 2 rows made using css.

[STARTED VIDEO LOG]

**Website Version 8: Mid page Download Button and Q and A:**

First I copied the download button from the header to the empty space down after the section I just made earlier. I also changed the buttons CSS so that it was the relevant orange theme and it was wider. I then put it inside a div so that I could locate and customise it more easily. For the Q and A section I copied the dropdown from the nav menu and similarly to the button I changed the css so that it was wider and had a background of the orange from the theme. Next made the site look a tiny bit nicer by adding a box shadow to buttons, images and some other elements. Then I went and replaced the placeholder content with actual answers to the questions. After that I wrote a small paragraph directing users to the q and a page if they happen to have more questions.

**Website Version 9: Why Recon Ops Section & Images**

First I made the "Why Recon Ops?" section following the plan with placeholder text, this section was just a duplicate of the Brief Description. Then I decided to add an image In-between this section and the previous one so that the user could have a break from reading and wouldn't get bored. Next I filled in the content with a paragraph and a list of reasons, I gave little explanations using the html title tag for each list item. After that I added another image below.

**Website Version 10: Footer:**

For the footer I decided to break off the plan and change it up to be a bit different. First I started by making a grey background and then I added a marquee like the plan specified and some social links. After that I grabbed some Icons from font awesome and added them around the site in places like the download button and social links to make the website look better. When I was testing out if the footer was working in a mobile view I noticed that the sticky class got removed off the menu so I added it back and changed the js so that it was fixed in there as well. I also noticed that when trying to open the changelog dropdown menu it would take you to that page because mobile hovering is a bit broken. The error was happening because you can't hover on mobile so to show the menu the user would have to click on the button which is also a link. My fix was to add an empty 'a href' and that did the trick. I also renamed the dropdown menu to "Changes''. A few finishing touches were to change the transition speed so that it was faster and less time was wasted.

### Template.html

**Website Version 11: Creation:**

I am going to make a template file so that creating other pages won't be as hard. This template file is going to be based on index.html and most elements are going to be copied from there. I did this by first copying the entire document of code from index.html across to template.html and updated the href to have proper folder path “../Pages/”. Then I replaced the text with filler text ready for copy and pasting. This template is going to be quite useful as it will speed up the development as I won't have to re-code everything, it will also help me keep everything organised and to the same theme across pages.

### About.html

**Website Version 12: Creation:**

To make about.html I just copied the medium sized paragraph and image from the helpful template page and replaced the filler content with actual information. After that I opened the game itself and took some screenshots for images that were going to be used in the website. Also I changed the CSS classes' names to be more relevant to what they do/contain.

### ChangeLog.html

**Website Version 13: Creation:**

First I made changelog.html and then copied the base code from template.html, then I noticed that the scroll to top button wasn't working so I had to fix that. The problem was that when I was messing around with the js earlier I accidently deleted the function to display the button. Afte I fixed that I tried to make use of the changelog from the patcher but it doesn't seem to work so I had to slowly, row by row, copy the items into a html list.

**Website Version 26: Update changelog to work with JSON**

I then updated the changelog page with auto-generated HTML from an external JSON file I pulled from the game servers. This way I don't have to add more bullet points every time there is an update, instead I just upload the new auto-generated changelog.

### Upcoming Features.html

**Website Version 14: Creation:**

First I copied the elements that I needed from template.html such as a download button then I decided to make a table and then fill in the filler content from the template with actual information. This table is just a simple HTML that was customised with CSS and made respon on mobile by adding vertical overflow.

### FeaturesID.html

**Website Version 15: Creation:**

These are small pages further explaining in detail the features from upcoming.html. All they contain is the images beside bullet points similar to index.html.

**Website Version 25: Removal:**

In this version I removed these pages as they were not needed as they were too small and didn't have enough information to be useful.

Notice.Html

**Website Version 16: Creation:**

First I copied the code from the Changelog page across to this current one and then rewrote the page contents to match. After that I coded the notice page, which is a basic JS slideshow. I managed to do this quite fast and the small issue I had was forgetting to add a plus like this (showSlide(slideIndex += n);)

**Website Version 23: Removal:**

When changing from Recon Ops to Sunnyland I had to remove the notice page as SunnyLand didn't have an in-game notice panel to pull from.

### Status.html

**Website Version 17: Creation:**

First I (as always) copied the contents I needed from tmeplate.html. Then I made a js service check which all that it really does is try to load an Image from a page. And if that image is not found then it changes the css class off the button and alerts the user with a dialog. Then I decided to implement cookies, with cookies the function won't have to be run every time the page loads and only run each day (cookie expiration). This helps lessen page load time and helps the user experience so that the alert box won't annoy them every reload.

### QandA.html

**Website Version 18: Creation:**

For the Q and A page I just copied the Q and A div from the index page. I sectioned off the Q and As into subsections like “Account” or “Device”.

**Website Version 22: Animation:**

Whilst adding animation to the other pages I noticed that the Q and A page was pretty boring, to fix this I added alternating left and right slides in transitions. I also added an animation so that when hovering over the question the answer would be displayed as an animation rather then just instantly showing.

### Login.html

**Website Version 19: Creation:**

To make login . html I had to do a lot, like learn hashing and php. First a I used a basic tutorial to make a login system. Then I made all the files local so that I could edit the page's css. I then designed it to be a modern mix between the plan and the current site. After that I spent AGES trying to figure out how to make an encryption system for the password. I did this for two reasons 1 being that it makes the password more secure and 2 being that the login plugin i use for the game is encrypted so they have to match for it to work. I spent a lot of time doing this and then realised that the system was chasing instead of encrypting. I then spent the rest of the day learning about bcrypt and other hashing. After that I had to implement the external page into the site. First I tried a full page iframe but I THINK that it can't store the session that way so I just redirected the user to the login site.

### More Bug Fixing

**Website Version 20: Bugs:**

PC:

Bug 1: Outdated link to reddit page, updated it and mirrored across all the pages

Bug 2: Favicon not showing up on feature pages, changed path across select pages

Bug 3: I frame on notice page scrollable, make a js function that auto resizes this so that I could add the scroll bar hidden function.

Bug 4: Duplicate Image on Q and A page, img28.png was used twice so I changed one of the duplicates to img29.png

MOBILE:

Bug 1: Dropdown is hidden on mobile, I fixed this by adding a new set of links that are only shown on mobile.

Bug 2: New nav links don't work, when implementing the new nav links i forgot to change the directories so i updated them to work

Bug 3: table isn't responsive, to fix this I added a scroll bar which then I had to do some changing for the table to align properly and orderly.

**Website Version 21: Bug Fixing Continued:**

Then i carried on bug fixing:

1. I fixed the active tag so that the homepage wasn't active on every single page

2. Changed the mobile active classes so that it wouldn't persistently show the link and it would only show when the menu was active

**Website Version 27: More Bug Fixing**

First I fixed a lot of the JS, then I went through and loaded a loader to every page except for the status.html page because if a site was down then the page would be loading forever. Next, I edited the q and a page to have Sunnyland contents.

### Improvements

**Website Version 21: Improvements:**

First I started by designing the website in mweb to test out the program. Then to make the site look better i started on a few changes:

1. I added keyframe animations to items like images and buttons

2. I made a JS function to add the animation tag when the element scrolled into view so that the animation was visible

3. Simple js function to title the page based on its file name.

4. Added animation to the q and a answers so that it looked more smooth

**Website Version 22: Animation and game launching:**

First I sent through and added animations to the about page. Then I had a random thought of including a button to open the launcher from the site. I did this by first including code in the installer to add the launcher to the registry, then I added a centred button that opens the prospect. After that i went through the other pages animating them where needed.

**Website Version 23: Change from reconops to sunnyland:**

After a bit of thought I changed the game the website was on from recon ops to sunnyland, this was because ReconOps may be classed as violent or unacceptable for marking. First I went through the index page and changed the text content to better describe the new game, and I also changed the colour theme to match the game.

**Website Version 24: Adding launch/download support for js:**

I then went through and added some JS to check if a cookie was loaded and if so then set the launcher classes for buttons to be active. This cookie is set ingame using the unity web handler

**Website Version 25: Added Loader**

I first went through and cleaned up the JS because my skills in the area have improved since earlier, whilst doing this i also fixed scrolling, fixed scroll to top button and also made it appear further down the page. I then added a loader which is displayed whilst the pages content is loading, this is on every page. Finally I fixed a bug with the loader mouse locking, which usually stops the user from scrolling to unloaded content whilst the page is loading.

**Website Version 28: Transitions, Demo**

I first added a demo page that has an embedded copy of the WebGL demo hosted on the Github page. Finally I added a transition between pages that is similar to the one in-game between levels. This transition was made using a css clipath animation and positioning JS. The problem I had whilst implementing the transition was that the animation would play at the centre of the page element rather than the centre of to users screen, this was fixed using a bit of css variables and JS maths (pageHeight = ocument.documentElement.scrollTop + (screen.height/2);)

**Website Version 29: Finished adding transitions, changed status so that there are less cookies**

First I finished adding transitions between the pages and made sure it worked on every page, then I changed the code from transitions being sent to the receiving page in the url (“index.html?transitionFrom=about\_html”) to being via a cookie. Whilst on the subject of cookies I also changed the cookies for the status page so that it was in a single cookie that held the data in JSON. This was because one cookie per status is a waste of cookie space as there can only be 20 cookies per site.

### Validity

**Website Version 30: Remove un-need ids, ad index alts, and comments , removed old html centre tags.**

First I went through and removed all the ids for the q and a elements as previously I was going to use them for scripting with JS but I never got around to using them as I decided to go with CSS animation instead. I then went through and added alts to every image as sir said I should. I then also replaced all my older <centre> tags with <div class=”centre”> that now use text align instead. I also began the tedious process of commenting.

**Website Version 31: Added more comments, remove overused centre divs**

Whilst continuing commententing I also began removing the centre divs as they were overused and tags were not closed.

**Website Version 32: Add Mobile Images, Add comments**

As I had finished pretesting the mobile version of the game I could add the mobile sized images that were displayed on viewports with a width less than 500px. I also continued adding individual comments to parts of the pages that were unique.

**Website Version 33: Validation Checking**

Using an online tool that pointed out my errors, I went through and fixed all the invalid pieces of code for my HTML and for my JS. I ordered the CSS document and then ran it through a linter and a validator making changes where needed.

Other Files:

Testing,

ScreenShots