

Database Project Plan

Github

<https://github.com/cozar1/Resource-Hub-Part-2>

Overview & Purpose

I noticed that when I was trying to download free assets for my projects such as games I was making, I found that most websites required a convoluted login process with an intuitive download process. I wanted to create a website that fixed these issues by providing completely free assets where anyone could download files creating an account. Originally my plan was to include Models, Textures, Sprites and audio files but I have decided to focus on only texture so as to make for an easier to navigate website where the user knows what to expect and make it easier for the user to search for what they are looking for. The purpose of this website is to provide a place for developers to quickly download files and create a fast workflow.

My website is going to have a basic UI with a minimalistic aesthetic that points the users to where they want to go rather than filling the website with unnecessary information. The user can choose to create an account to upload files. The users uploaded files can be viewed through their dashboard page and they can delete / upload files from there. Users who wish to download files will not need to login.

Stakeholders & End-Users

The end-users of my website will be people/developers looking for free and easily accessible assets for their game or project. The intended audience for my website are people who don't like how frustrating it is to create a login to hundreds of different websites just to have a chance to have access and download a texture.

Relevant Implications

Database

Privacy	functionality
Privacy is a critical implication because it makes sure that bad people don't have access to critical information such as passwords, locations and private information. A private database helps with reliability. My database makes sure that privacy is kept between	In a database design, functionality describes how well the database structure and features actually support the tasks it is meant to perform. It's about matching what the database can do to what the users and applications need. This can be achieved through efficient queries, manipulating data reliably and table

<p>its users by not storing critical data such as passwords, locations or other exposing information. By employing the privacy relevant implication I am able to build trust between my users so that they feel safe using my database.</p>	<p>structures such as many-to-many.. Etc.</p> <p>I use functionality in my database to make sure that my database functions correctly. I use different table structures such as one-to-many, one-to-one and many-to-many to create an efficient environment where the database is reliable to use. It is important for me to consider functionality because it allows for my data to be correct and complete as well as increase the performance of my overall program.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Website

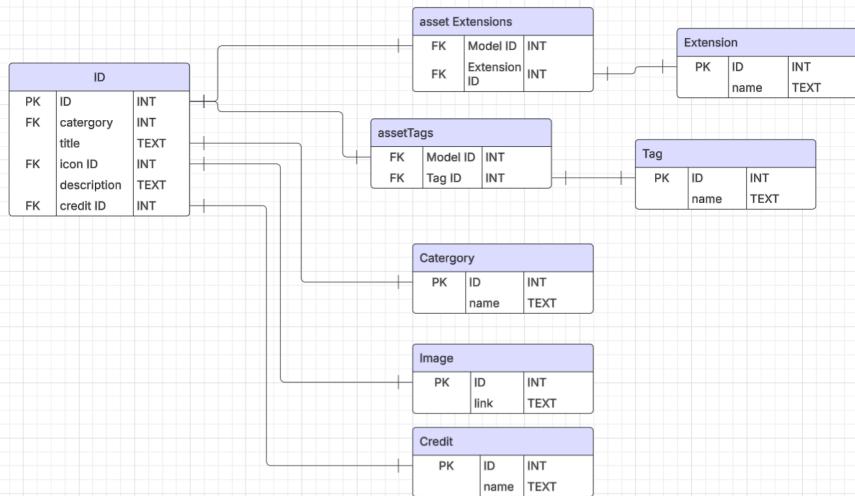
aesthetics	usability
<p>Aesthetics refers to how a website looks visually. Aesthetics influence how a website is perceived. Users will be drawn to a more modern website which creates user engagement. By using balanced layouts, appealing color schemes, and readable typography, a website is able to be easily accessible to people and make them have a more enjoyable experience.</p> <p>I use aesthetics in my website to create a much nicer user experience. Throughout my development process, I have gathered feedback to ensure that the aesthetics of my website are optimal. One of the ways I did this was through the font I used. Originally it was a pixelated font but I chose to switch it for a more readable one which heightened the user experience.</p>	<p>Usability is about how easy, efficient and satisfying it is for people to use your website and achieve what they need to do. Usability allows for faster navigation through the website which creates a greeted user experience because the user takes less steps to achieve their goals. Usability also allows for a more intuitive interface where new users are easily able to identify what to do, this can be done through recognition rather than recall and other usability heuristics.</p> <p>In my website I use icons to increase the experience for the user because they are easily able to recognise the purpose of elements based off of the icon. I have also made sure to reduce the amount of clicks it takes for the user to reach their intended destination by contemplating what the best way to create my layouts and interfaces would be.</p>

Sprint 1

In this sprint I am going to be designing all of the pages with their layouts, colors, fonts, routes and queries. This will provide a foundation for me to build off of and improve from feedback and testing. This sprint does not contain any tests because it is purely design. From this sprint on, all changes will be specified in said sprint, Queries from previous sprints will not be included in future sprints on this document but accumulate in my website.

Entity Relationship Diagram

I have created a main item table that controls the individual items that can be uploaded to the website that vary different types of asset files. I have added an asset-extension many-to-many table which will identify which items contain which tags. There is also another many-to-many relationship found in the asset tags, this identifies which items have what tags.



Visual Layout & Functionality

Font

Jersey 10
<https://fonts.googleapis.com/specimen/Jersey+10>
<link href="https://fonts.googleapis.com/css2?family=Jersey+10 display=swap" rel="stylesheet">

Colors

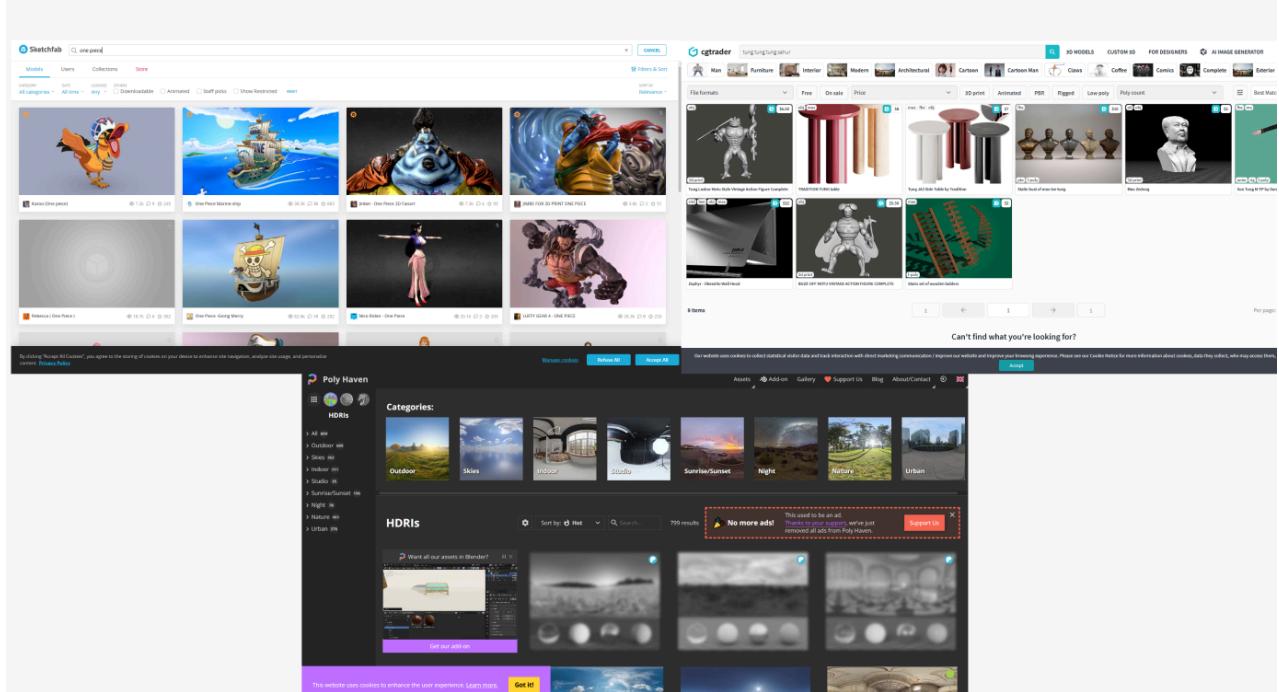
#C0A694
#1B2021
#612C2B
#30343F

Home Page @app.route('/')

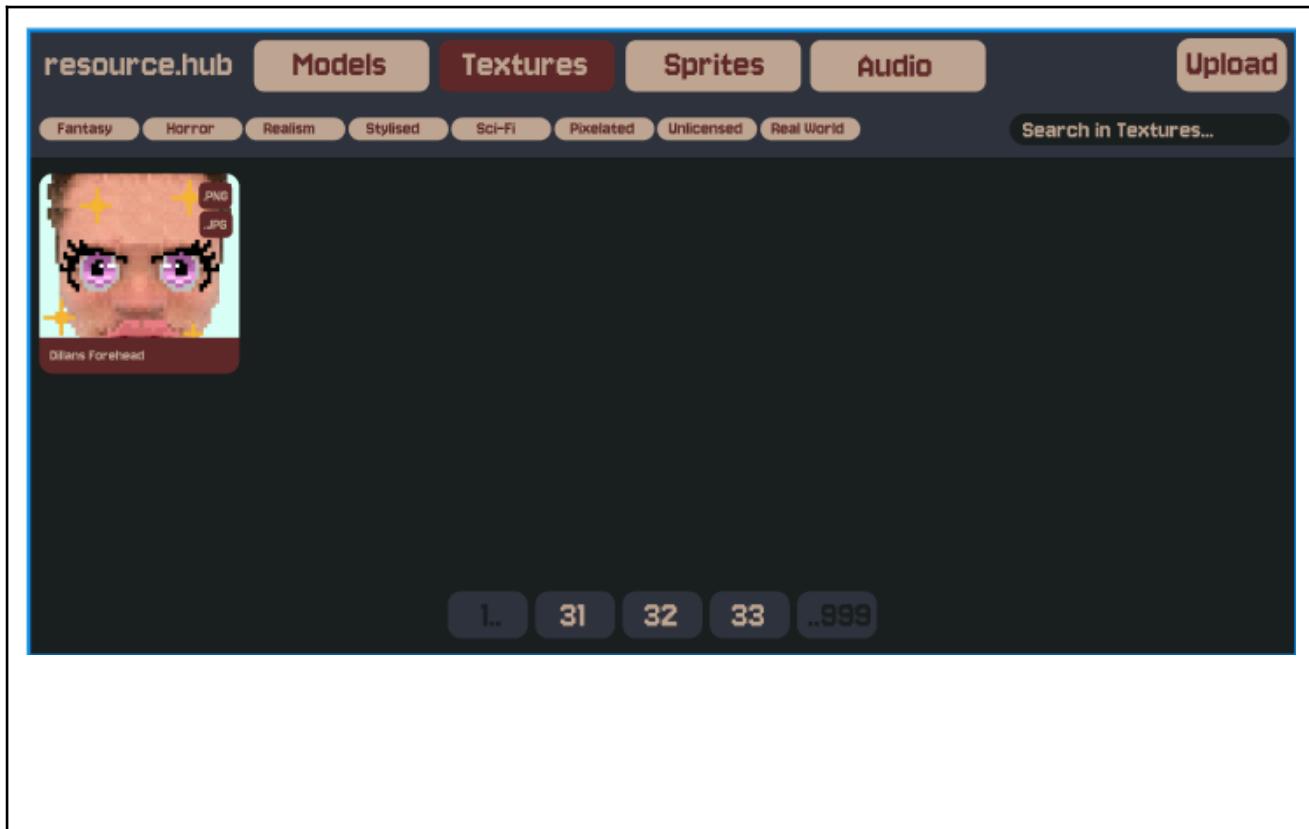
Mockup

I looked at lots of different websites that fit similar purposes as mine and identified a few key details I wanted to use in my own website. These include

- Header with title in the top left
- Tags in the navbar
- Grid of items with previews and details
- Consistent simple color schemes



For my home page, I wanted to try my best to create a similar design as my researched websites. I added a similar header with tabs in the navbar. I also added an upload button and the search bar on the right side to balance the elements on the screen. The assets will be displayed as the image with a few details such as the file type and name.



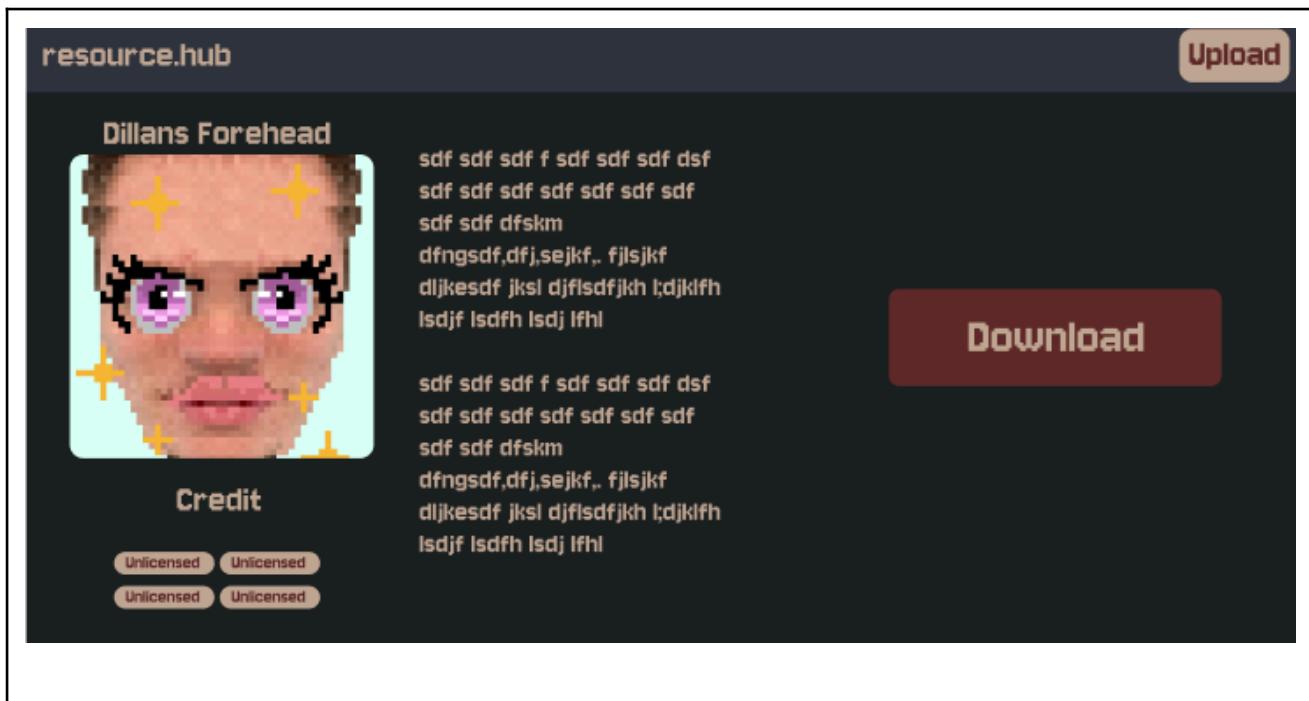
Queries

- "SELECT name,icon FROM Tag"
- "SELECT id, title FROM asset"
- "SELECT name,icon FROM Tag WHERE ID IN (SELECT Tag_ID FROM assetTags WHERE Model_ID = ?)", (asset[0],)

Asset Page @app.route('/asset')

Mockup

Following a similar color palette and style I make sure to display all the information clearly while making it obvious where to download.



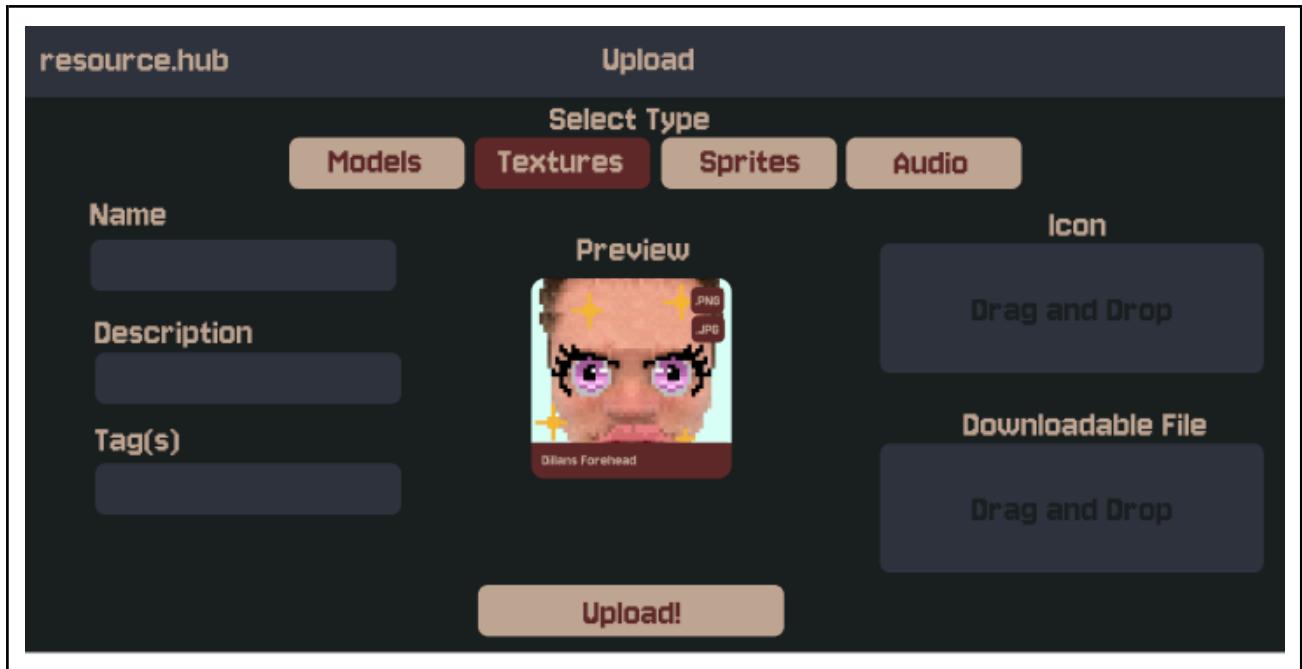
Queries

- "SELECT name,icon FROM Tag"
- "SELECT id, title FROM asset"
- "SELECT name,icon FROM Tag WHERE ID IN (SELECT Tag_ID FROM assetTags WHERE Model_ID = ?)", (asset[0],)

Upload Page @app.route('/upload')

Mockup

For the upload page, I wanted to focus on making an intuitive UI so that the user can easily identify the steps they needed to take in order to upload an image. I also wanted to make it clear to the user what kind of file they were uploaded by making those options big and contrasting at the top of the page



Queries

- cursor.execute("INSERT INTO asset(name, description) VALUES(?, ?)", (name, description))

Feedback

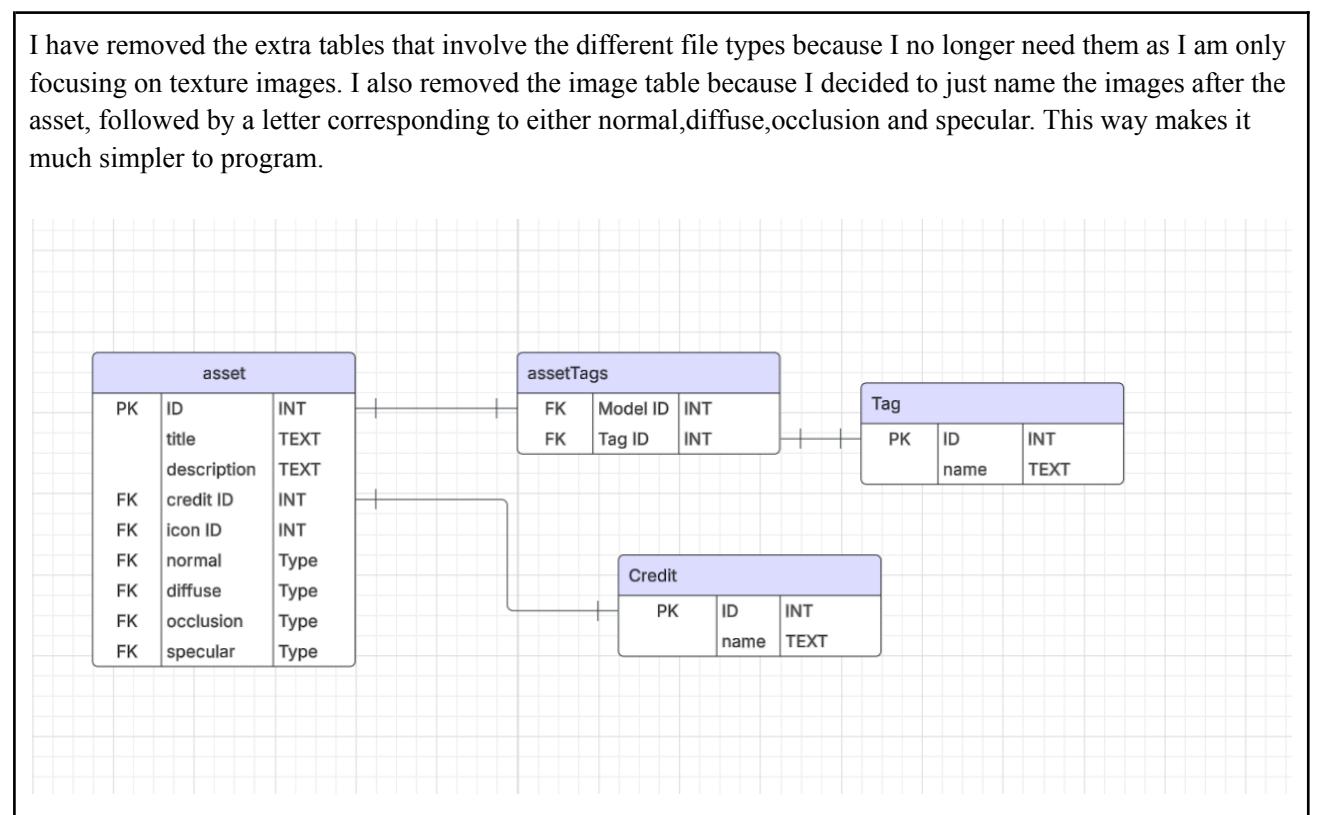
Feedback	Implications
Isaac "Make the title in call caps because it looks bad at the moment. I can't read the text very well when they have the drop shadow"	<ul style="list-style-type: none"> • Title I made the title in all caps to make it stand out more.
Olly "The pages would be very hard to implement so I would recommend ditching them"	<ul style="list-style-type: none"> • Symbols I used Font-Awesome to add unique symbols to the tags and upload button to address the heuristic "Recognition rather than recall" and add some artistic flair.
Dylan "I don't like how the file extension thingies cover the preview"	<ul style="list-style-type: none"> • Colors I inverted the color order to make them easier and look better with the drop shadow.
Ollie "The colors are very dark which could impact the visibility during the daytime so you should add a light mode."	<ul style="list-style-type: none"> • Asset Block I moved the file extensions under the image so that it would be easier to write the css because layering elements gets fiddly when dealing with dynamic scaling pages. I also added the tags under each asset to include more information.
Ollie "You should add symbols to make things more clear"	<ul style="list-style-type: none"> • Pages
Nathan "I like the layout but I don't like the colors its very simple and self explanatory"	

	<p>I didn't add the pages at the bottom of the screen because It would have been tricky to add and not directly necessary.</p> <ul style="list-style-type: none"> • Shadows I added drops shadows to most of the text to make them pop out
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sprint 2

In this sprint I will begin to implement the features I designed in sprint 1. This sprint will include tests and feedback that I will use to improve my design in sprint 3. This sprint will use designs, improved from feedback in sprint 1. I also decided during this sprint that I wanted to just focus on texture files so I wasn't overwhelmed by features that I needed to add.

Entity Relationship Diagram



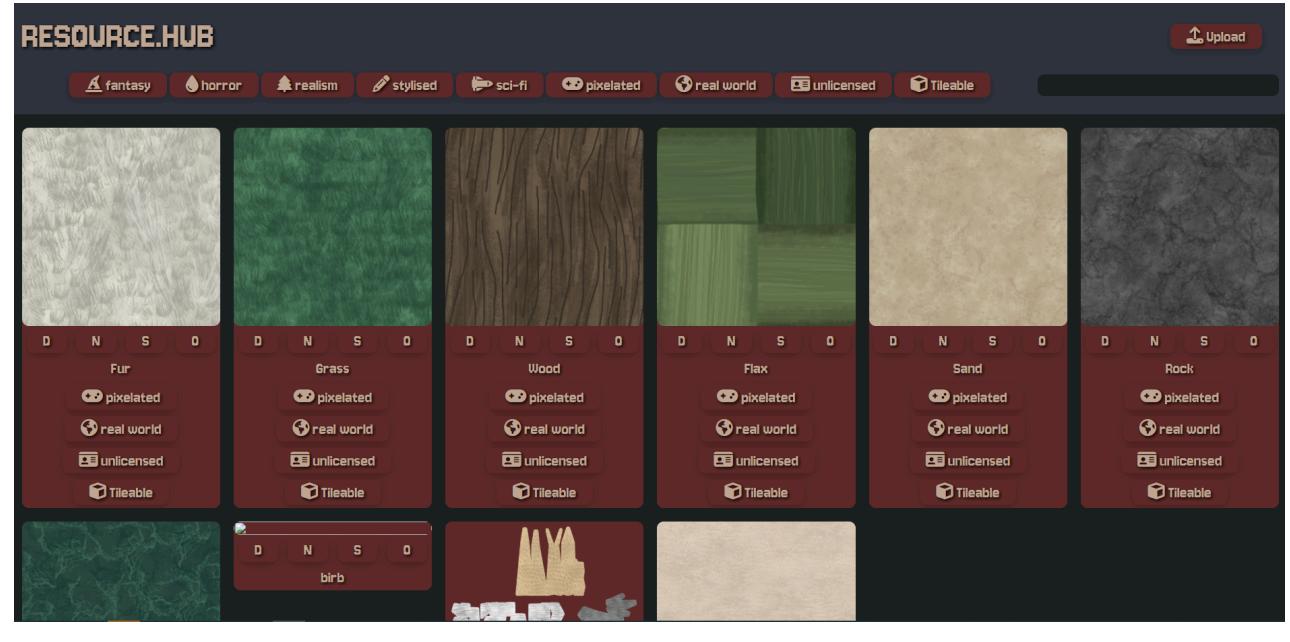
Visual Layout & Functionality

I kept the fonts and colors because there was no feedback to suggest that it should be changed or improved so I have opted to include and keep my original color scheme and fonts.

Home Page @app.route('/')

Design

I added the font-awesome library which allows me to include lots of different symbols to help the recognition rather than recall which is one of the usability heuristics



Queries

- "SELECT name,icon FROM Tag"
- "SELECT id, title FROM asset"
- "SELECT name,icon FROM Tag WHERE ID IN (SELECT Tag_ID FROM assetTags WHERE Model_ID = ?)", (asset[0],)

Asset Page @app.route('/asset/<int:id>')

Design

This page kept relatively simple except I moved the tags to the bottom of description so the image had more room on the screen and made it feel more balanced.

RESOURCE.HUB

Grass	Description	Downloads
	Tileable Grass Texture <input checked="" type="checkbox"/> pixelated <input type="checkbox"/> real world <input type="checkbox"/> unlicensed <input checked="" type="checkbox"/> Tileable	Normal Map Diffuse Map Specular Map Occlusion Map

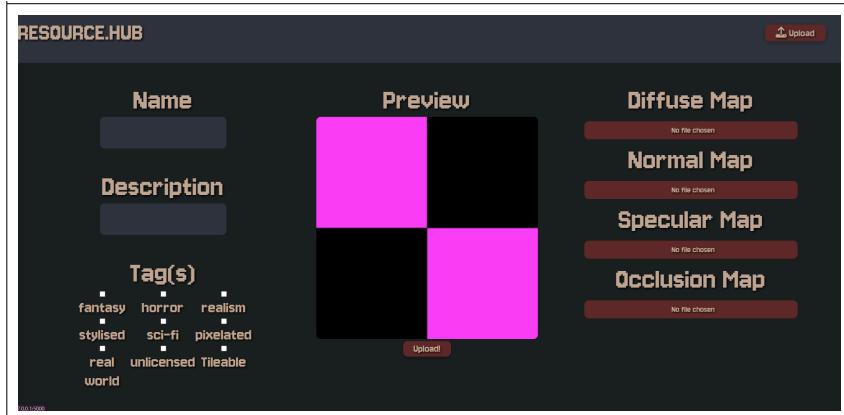
Queries

- "SELECT id,name,description FROM asset WHERE ID = ?", (id,)
- "SELECT name,icon FROM Tag WHERE ID IN (SELECT Tag_ID FROM assetTags WHERE Model_ID = ?)", (id,)

Upload Page @app.route('/upload')

Design

This page also kept similar apart from there now being 4 uploads, one for each texture type. I also decided to make the tags checkboxes which was easier than the system I wanted to use before where I didn't know how I was going to do it.



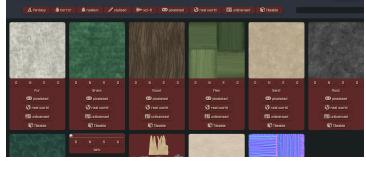
Queries

Get:	<ul style="list-style-type: none"> • "SELECT id, name FROM Tag"
Post:	<ul style="list-style-type: none"> • "INSERT INTO asset(name, description) VALUES(?, ?)", (name, description), commit=True • "INSERT INTO assetTags(Model_ID, Tag_ID) VALUES(?, ?)", (asset_id, tag_id), commit=True

Testing

Page	What am i testing	How am i testing it	What do i expect	What did i get	Pass or fail
Home	I am testing the hover effect	I am going to put my mouse over the asset divs to test if they correctly scale.	I expect for the asset to grow in size slightly when I put my mouse over it and reduce back to normal size when I remove my mouse	When I put my mouse over the asset it grew which was expected and returned to size shortly after.	Pass
Home	I am testing the search bar	I am going to search text and see if it filter my results and correctly reformats my layout	I expect for it to correctly filter results and reformat	The search worked and filtered my results but didn't correctly reformat	The test failed because it didn't reformat the page but I don't feel like fixing it so it's not a bug, it's a feature.
Home	Clicking on the asset pages	I will click on the page to see if it pulls up the page for the individual assets	For it to pull up the page and give me my results	It successfully pulled up the page with name, image, download buttons, description and creator.	pass

Data Integrity Testing

Link	Url that it links has	The data in the database	Where the link takes you																																	
	127.0.0.1:5000/asset/18	<table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>18</td> <td>Grass</td> <td>Tileable Grass Texture</td> </tr> <tr> <td>19</td> <td>Wood</td> <td>Tileable Wood Texture</td> </tr> <tr> <td>20</td> <td>Fax</td> <td>Tileable Fax Texture</td> </tr> <tr> <td>21</td> <td>Sand</td> <td>Tileable Sand Texture</td> </tr> <tr> <td>22</td> <td>Rock</td> <td>Tileable Rock Texture</td> </tr> <tr> <td>23</td> <td>Water</td> <td>Tileable Water Texture</td> </tr> <tr> <td>24</td> <td>birb</td> <td></td> </tr> <tr> <td>25</td> <td>birb</td> <td></td> </tr> <tr> <td>26</td> <td>Skin</td> <td></td> </tr> <tr> <td>27</td> <td>sdfdsdf</td> <td></td> </tr> </tbody> </table>	ID	Name	Description	18	Grass	Tileable Grass Texture	19	Wood	Tileable Wood Texture	20	Fax	Tileable Fax Texture	21	Sand	Tileable Sand Texture	22	Rock	Tileable Rock Texture	23	Water	Tileable Water Texture	24	birb		25	birb		26	Skin		27	sdfdsdf		
ID	Name	Description																																		
18	Grass	Tileable Grass Texture																																		
19	Wood	Tileable Wood Texture																																		
20	Fax	Tileable Fax Texture																																		
21	Sand	Tileable Sand Texture																																		
22	Rock	Tileable Rock Texture																																		
23	Water	Tileable Water Texture																																		
24	birb																																			
25	birb																																			
26	Skin																																			
27	sdfdsdf																																			
RESOURCE.HUB	127.0.0.1:5000	<table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Icon</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1 fant...</td> <td>fa-solid...</td> </tr> <tr> <td>2</td> <td>2 horr...</td> <td>fa-solid...</td> </tr> <tr> <td>3</td> <td>3 real...</td> <td>fa-solid...</td> </tr> <tr> <td>4</td> <td>4 styl...</td> <td>fa-solid...</td> </tr> <tr> <td>5</td> <td>5 sci-fi</td> <td>fa-solid...</td> </tr> <tr> <td>6</td> <td>6 pixe...</td> <td>fa-solid...</td> </tr> <tr> <td>7</td> <td>7 real ...</td> <td>fa-solid...</td> </tr> <tr> <td>8</td> <td>8 unli...</td> <td>fa-solid...</td> </tr> <tr> <td>9</td> <td>9 Tile...</td> <td>fa-solid...</td> </tr> </tbody> </table>	ID	Name	Icon	1	1 fant...	fa-solid...	2	2 horr...	fa-solid...	3	3 real...	fa-solid...	4	4 styl...	fa-solid...	5	5 sci-fi	fa-solid...	6	6 pixe...	fa-solid...	7	7 real ...	fa-solid...	8	8 unli...	fa-solid...	9	9 Tile...	fa-solid...				
ID	Name	Icon																																		
1	1 fant...	fa-solid...																																		
2	2 horr...	fa-solid...																																		
3	3 real...	fa-solid...																																		
4	4 styl...	fa-solid...																																		
5	5 sci-fi	fa-solid...																																		
6	6 pixe...	fa-solid...																																		
7	7 real ...	fa-solid...																																		
8	8 unli...	fa-solid...																																		
9	9 Tile...	fa-solid...																																		

Feedback

Feedback	Implications
<p>Ollie : "I like the use of the slightly brighter colors which have better contrast. I still think light mode would be cool. Some of the elements are a bit too small"</p> <p>Olly "I think that the download buttons could be bigger, because there is a lot of empty space around it, and because there will always be a fixed amount of download buttons, so keeping space for extras isn't needed.</p>	<ul style="list-style-type: none"> I am going to think about adding light mode so allow for a more inclusive experience from people with different preferences I am going to adjust the css to make the button sizes more impactful and obvious. I am going to try and fill the void space with better layouts

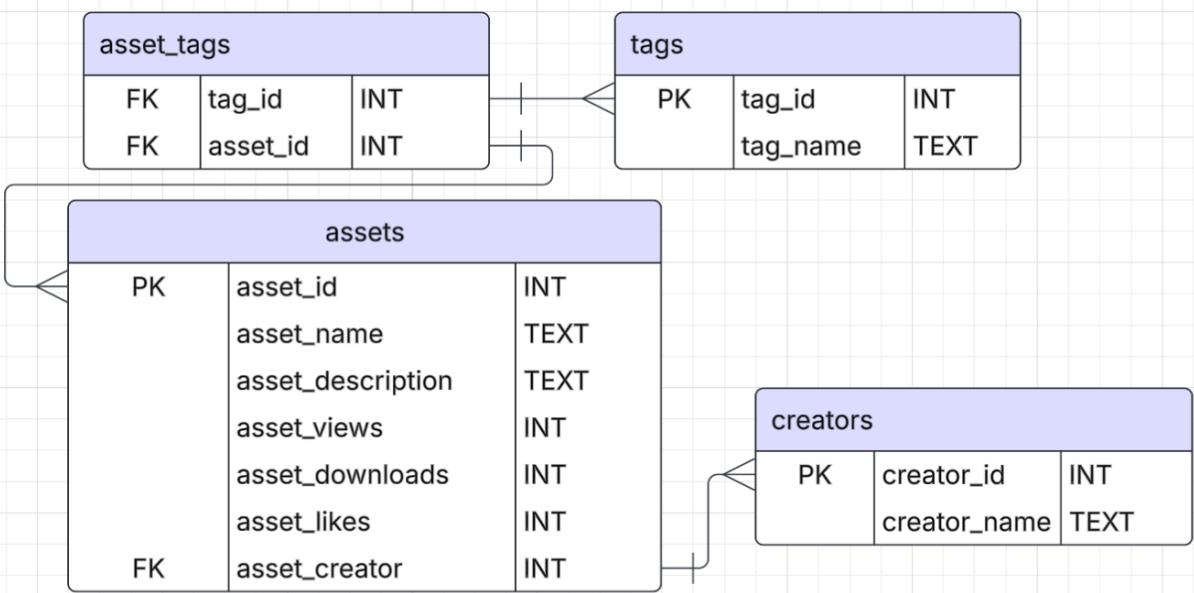
The header feels empty, so you should see if there is anything you can add like ads.”

Olliver : “Very resourceful and a good use of all the spaces with very nice features included. Filtering by tag is a bonus.”

Sprint 3

In this sprint I am going to implement the feedback given in sprint 2 and make sure my website doesn't have any bugs or issues to finish this project. I am also going to refine my naming and other aspects relative to the correct conventions.

Entity Relationship Diagram



changed:

- Use underscore_names instead of CamelCase
- Table names should be plural
- Spell out id fields (`item_id` instead of `id`)
- Don't use ambiguous column names
- When possible, name foreign key columns the same as the columns they refer to

I also added `asset_views`, `asset_downloads` and `asset_likes`.

Visual Layout & Functionality

Based on feedback I received and my own insights, I redesigned the basic colors and slight changes to aspects like border radii. I also changed the font to give off a more professional appearance. I made the colors more saturated because before they were a little dull. I also added light mode from feedback.

I added variables to my css so that all of the colors, margins.. Etc were consistent.

```
:root{  
    --primary-color: #18191F;  
    --secondary-color: #2C2E37;  
    --accent-color: #a21234;  
    --primary-font-color: #F2F8FB;  
    --secondary-font-color: #18191F;  
    --shadow-color: black;  
    --font-family: 'inter', sans-serif;  
  
    --border-radius: 8px;  
    --margin: 5px;  
    --padding: 5px;  
    --small-font-size: 16px;  
    --medium-font-size: 20px;  
    --large-font-size: 55px;  
}
```

I made consistent box shadow and font shadows to all of the elements so that everything popped and didn't feel flat like my previous website. I had to make sure the icons had a fainter drop shadow because they were too dark from the icons being smaller than text and revealing more shadow.

Colors:

- 18191F
- 2C2E37
- AD2E54
- F2F8FB

I am now using the [Inter](#) font.

Home Page @app.route('/')

I have done a massive overhaul this sprint and one of the biggest is to the home page. Here is a compilation of all the changes that were made to the home page:

- **Colors**

From feedback and personal experience, I have noticed that my old website had colors with very similar brightness values which made the experience feel dull and foggy. The colors were also very desaturated which only added to the problem. I reopened figma and started

tweaking the colors and decided on a darker background that had a linear gradient to spice it up, I did the same for the head elements except being slightly brighter.

Implementing this in css was a unique experience. I used this [Website](#) to input my colors and receive an output background css.

- **Fonts**

While in figma, I wanted a new font because my current one didn't feel very serious and I didn't want that so I stumbled upon the Inter font which felt perfect so I applied to the website.

- **Alignment**

There are changes to the alignment of the different items so that it feels more uniform. This includes consistent margins, paddings and properly aligned header items. The assets also now align correctly when filtering.

- **Asset layout**

I needed a new layout to the assets because with too many tags, the asset became huge and obtrusive. I removed displaying tags because they didn't matter, especially when you can just filter them. I also laid out the asset horizontally to increase room. I added a vignette to make the asset pop.

- **Views,downloads and likes**

I needed some sort of engagement for my website and a way for people to judge the quality of textures so I added views, downloads and likes. I calculate likes by simply adding to the database whenever the asset page is loaded. Downloads were a little trickier with requiring me to rewrite how I handle downloads so I could create a new route that handled the download system. When you download an image, it takes you to the route where it adds a download to the asset from the id.

Adds views to database whenever the asset route is loaded

```
query("UPDATE asset SET views = COALESCE(views, 0) + 1 WHERE ID = ?", (id,), commit=True)
```

Download buttons

```
<a class="button" href="/download/{{asset[0][0]}}/n">Normal Map <i class="fa-solid fa-download"></i></a>
<a class="button" href="/download/{{asset[0][0]}}/d">Diffuse Map <i class="fa-solid fa-download"></i></a>
<a class="button" href="/download/{{asset[0][0]}}/s">Specular Map <i class="fa-solid fa-download"></i></a>
<a class="button" href="/download/{{asset[0][0]}}/o">Occlusion Map <i class="fa-solid fa-download"></i></a>
```

Download route

```
@app.route('/download/<int:asset_id>/<string:image_type>')
def download_image(asset_id, image_type):
```

Adding downloads in download route

```
query("UPDATE asset SET downloads = COALESCE(downloads, 0) + 1 WHERE ID = ?", (asset_id,), commit=True)
```

- **Assets only display the option to view image types if that image type exists**

What this means is that if an asset only has a diffuse texture for example then it wont display all of the options (d,n,s,o). I did this by passing through a new list at the end of the assets tuple that then iterates through in the html to only display the buttons that are in that list.

Checking for available types in the home route

```
available_types = []
for suffix in ['d', 'n', 's', 'o']:
    file_path = os.path.join(app.config["UPLOAD_FOLDER"], f"{asset[0]}.{suffix}.png")
    if os.path.exists(file_path):
        available_types.append(suffix)
asset.append(available_types)
```

Iterating through the passed list to display the icons in the home.html.

```
{% for type in asset[6] %}
<div class="button img-type-btn" data-type="{{type}}" data-asset-id="{{asset[0]}}>{{type.upper()}}</div>
{% endfor %}
```

During this I ran into a bug where It would display the image types even when I didn't upload any. The problem came from when uploading, It didnt check if the file type was uploaded to it and created empty image textures. I fixed this by checking if the image existed and didn't have an empty name.

- **Resolution**

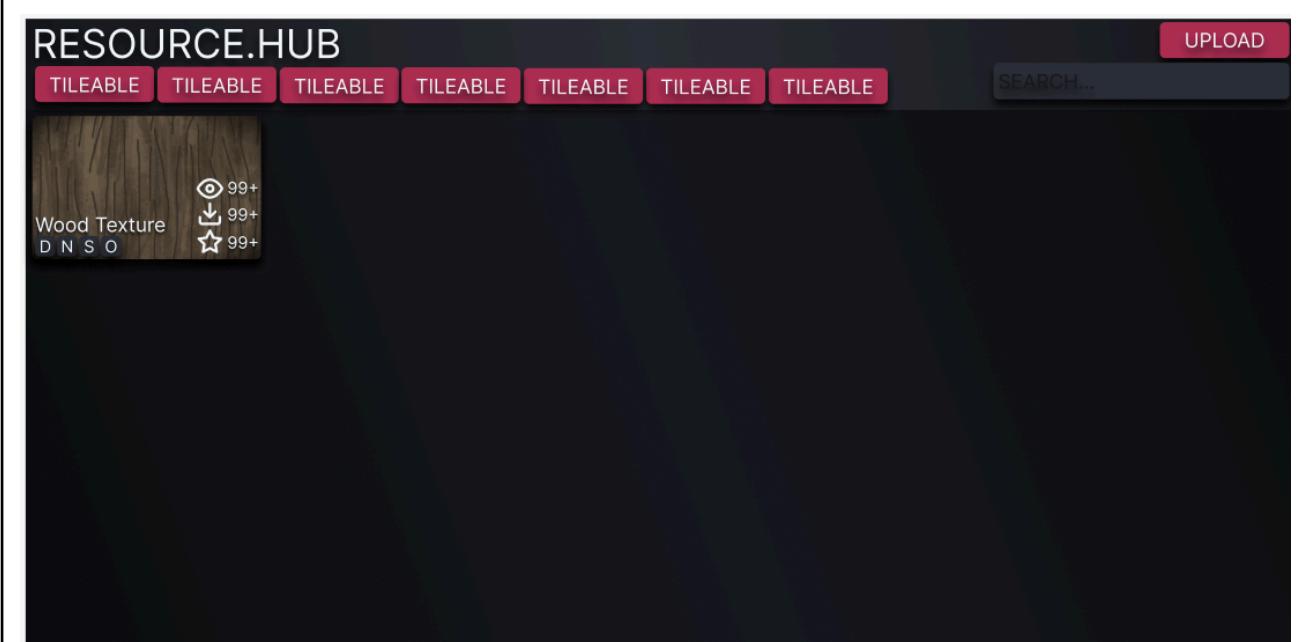
I added a resolution to the asset info to give people a better idea of what they are downloading.

- **Hiding invalid file types**

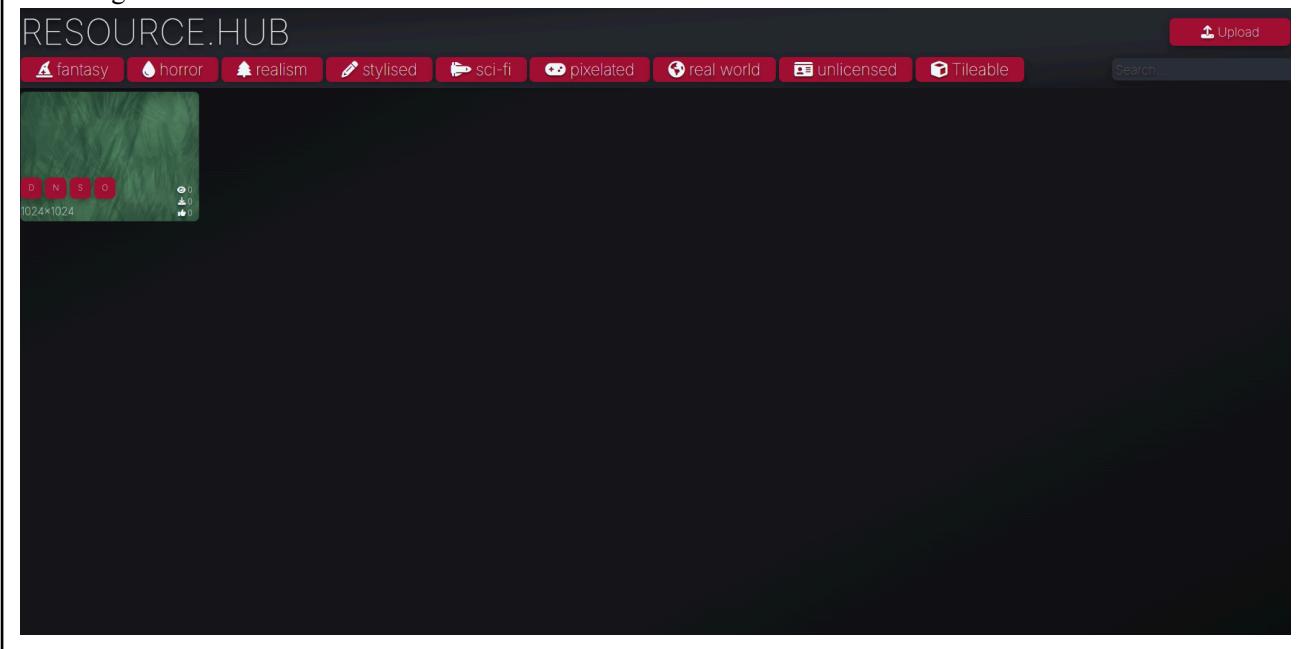
I made it so that if an asset doesn't have some of the image types then it won't display the option to preview it. Similarly, if an asset only has a diffuse texture there will be no option at all.

Design

Mockup



Final Page



Queries

- assets_data = query("SELECT id,name,COALESCE(views,0),COALESCE(downloads,0),COALESCE(likes,0),COALESCE(resolution,'Unknown') FROM asset")

Asset Page @app.route('/asset')

Changes

- **Image type previews**

I have added image type previews similar to how you can preview them in the home page

```
<div class="img-types">
    {% for suffix in available_downloads %}
        <div class="button img-type-btn" data-type="{{suffix}}" data-asset-id="{{asset[0][0]}}>{{suffix.upper()}}</div>
    {% endfor %}
</div>
```

- **Stats**

I have added a section underneath the downloads that allow the user to see the stats of the asset in the page instead of just the home page

```
<h1>Stats</h1>
<p>Views: {{asset[0][2]}}</p>
<p>Downloads: {{asset[0][3]}}</p>
<p>Likes: {{asset[0][4]}}</p>
```

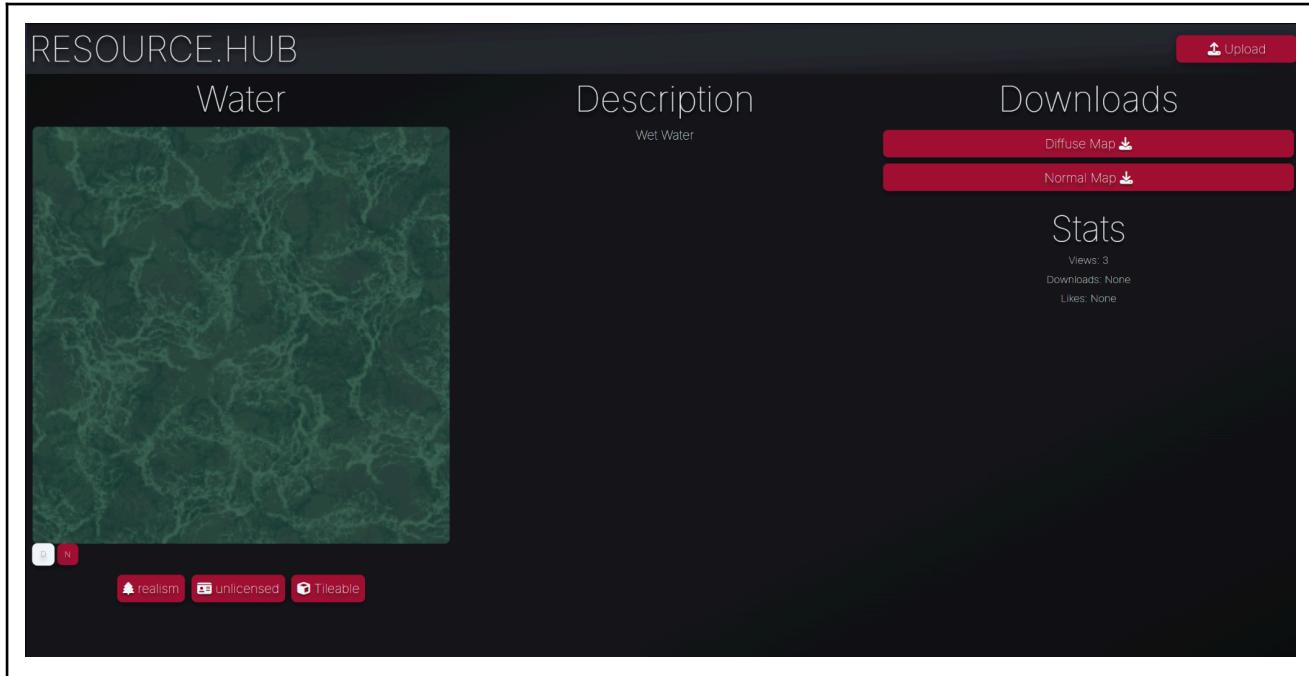
- **Restricted Download**

You can only download images that the asset has because earlier you just downloaded null files.

```
available_downloads = []
for suffix in ['d', 'n', 's', 'o']:
    file_path = os.path.join(app.config["UPLOAD_FOLDER"], f"{id}{suffix}.png")
    if os.path.exists(file_path):
        available_downloads.append(suffix)

<div class="vertical-align">
    {% if available_downloads %}
        {% for suffix in available_downloads %}
            {% if suffix == 'd' %}
                <a class="button" href="/download/{{asset[0][0]}}/d">Diffuse Map <i class="fa-solid fa-download"></i></a>
            {% elif suffix == 'n' %}
                <a class="button" href="/download/{{asset[0][0]}}/n">Normal Map <i class="fa-solid fa-download"></i></a>
            {% elif suffix == 's' %}
                <a class="button" href="/download/{{asset[0][0]}}/s">Specular Map <i class="fa-solid fa-download"></i></a>
            {% elif suffix == 'o' %}
                <a class="button" href="/download/{{asset[0][0]}}/o">Occlusion Map <i class="fa-solid fa-download"></i></a>
            {% endif %}
        {% endfor %}
    {% else %}
        <p>No downloadable files available.</p>
    {% endif %}
```

Design



Queries

- query("UPDATE asset SET views = COALESCE(views, 0) + 1 WHERE ID = ?", (id,), commit=True)
- query("SELECT id, name, views, downloads, likes, description FROM asset WHERE ID = ?", (id,))
- query("SELECT name, icon FROM Tag WHERE ID IN (SELECT Tag_ID FROM assetTags WHERE Model_ID = ?)", (id,))

Upload Page @app.route('/upload', methods=["GET", "POST"])

Changes

- **Vertical Layout**

I decided to make the layout vertical so that when uploading, the user has a clear view of the process in which to upload an asset. This layout will also help them reach the upload button at the bottom instead of pressing the one at the top which resets the page.

- **Cleaner tags**

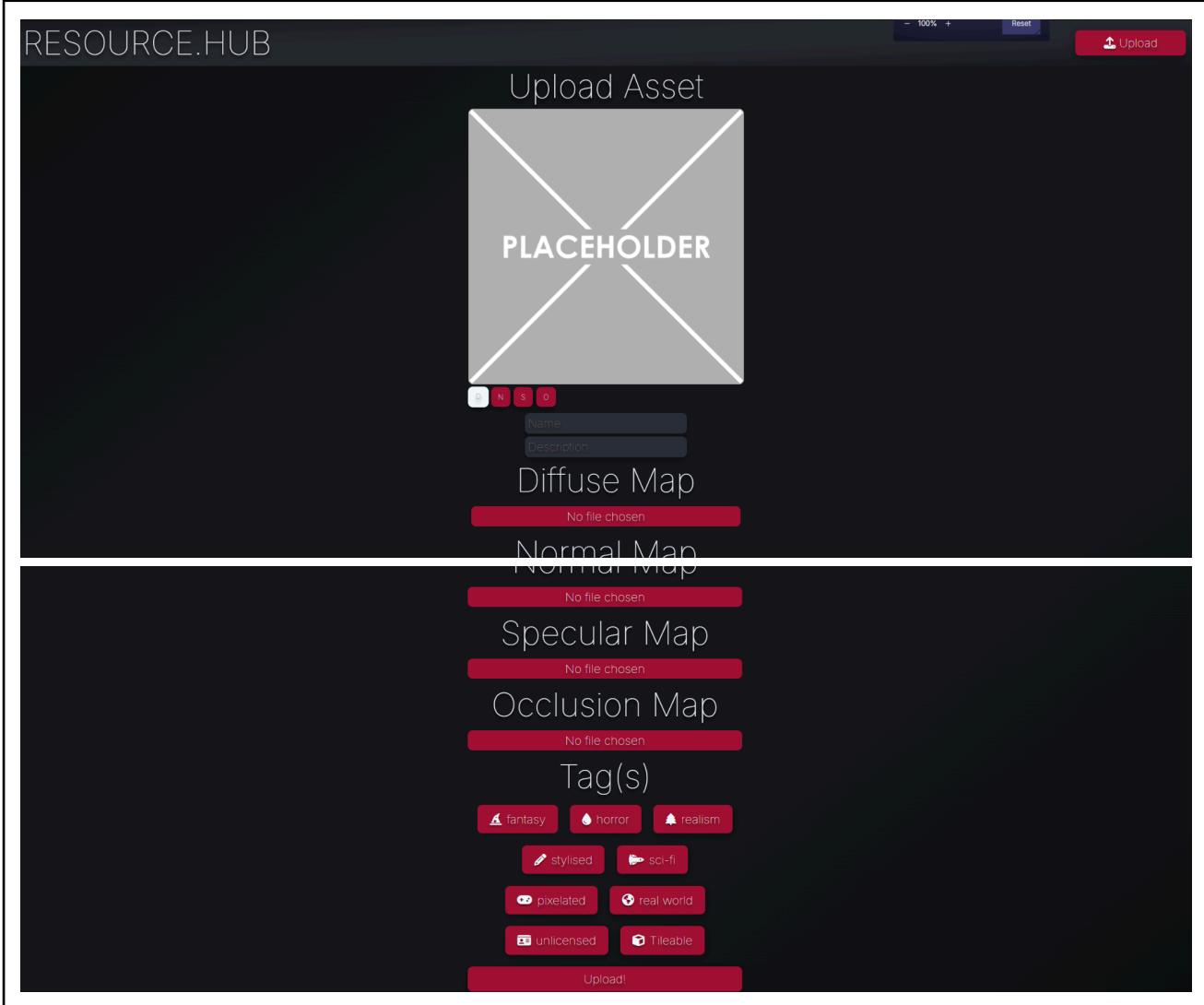
I styled the tags so that it fit the style better than last sprints ones.

- **Image previews**

Similar to the home page and the asset page, you can preview the images you upload using the buttons below the image.

```
<div class="img-types">
    <div class="button img-type-btn active" data-type="d" data-preview-id="preview-img">D</div>
    <div class="button img-type-btn" data-type="n" data-preview-id="preview-img">N</div>
    <div class="button img-type-btn" data-type="s" data-preview-id="preview-img">S</div>
    <div class="button img-type-btn" data-type="o" data-preview-id="preview-img">O</div>
</div>
```

Design



Queries

Get:

```
query("SELECT id, name, icon FROM Tag")
```

Post:

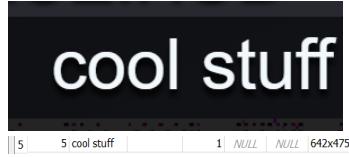
- query("INSERT INTO asset(name, description, resolution) VALUES(?, ?, ?)", (name, description, resolution), commit=True)
- query("INSERT INTO assetTags(Model_ID, Tag_ID) VALUES(?, ?)", (asset_id, tag_id), commit=True)

Testing

Page	What am I testing	How am i testing it	What do i expect	What did i get	Pass or fail (and how I fixed it)
Home	Assets will reorder when resizing the page to fill the space	I will resize the page	I expect the elements to flow in a conventional way, keeping the same basic structure and intended flow.	The website kept a conventional flow and remained usable	pass
Home	Hovering over image preview will preview the image	Hovering over the image type preview	The different available image previews will be shown when i hover over the icon	The different image previews appeared when I hovered over the icon	pass
Home	Only image types the asset has will appear as an option to preview	Hovering over the image type preview and seeing if they have an image type to preview	I expect for only available image types to be shown	The asset shows all images not just the ones it has	Fail. I updated the python code to accommodate for updated variable names.
Home	The search bar correctly filters results	I am going to search for a result I know exists and see if it appears	I expect for the result to appear	The asset I searched appeared	pass
Home	The upload button	I am going to press the upload button	I expect it to take me to the upload page	It took me to the upload page	pass
Home	The home button	I am going to press the home button	I expect it to take me to the home page	It took me to the home page	pass
Home	Assets button	I am going to press a random assortment of assets to see if it takes me to the appropriate page	I expect for the asset to take me to the respective asset page	It took me to the respective asset page	pass
Home	Selecting tags will filter results	I am going to select tags and see what results appear and then check that those assets that appeared have the correct tags	I expect for the tags to filter results and for those results to have the correct tag combinations	The filtered results had the right combinations of tags	pass
Asset	Clicking on an asset will give it a view	I am going to click on an asset and see if it gives it a view	I expect for the asset to get a view when I click on it	Each time I clicked on an asset, It gave it a view	pass
Asset	Downloading any image will give the asset a download	I am going to download an image and see if it gives it a view	I expect for the asset to gain a download when I download any image	For every available download, a download was added to the asset	pass
Asset	Only valid downloads will be displayed	I am going to upload an asset with on diffuse and normal and see if only those two appear	I expect for only a diffuse and normal texture appear to be downloaded	Only diffuse and normal were displayed	pass
Asset	All of an assets tags will be displayed	I am going to upload an image with an assortment of tags	I expect for those selected tags to be displayed when I look at the asset page	Only the tags selected when uploading were displayed	pass
Asset	You will be able to preview image types from available image types	I am going to upload an asset with a diffuse and normal texture	For those two images to be previewable and no other	Only the uploaded textures were displayed	pass
Asset	An assets stats will be displayed	I am going to check if the stats are displayed and correspond to the database	I expect for the correct stats to be displayed	The correct stats were displayed	pass
upload	You can select image type previews	I am going to select a normal and	To be able to preview both image	I could preview both image types	pass

	for valid image types	diffuse and see if I can preview them	types		
upload	Names are uploaded to the database	I am going to upload an asset with a name and see if it goes into the database	The name to be uploaded to the database	The name was uploaded to the database	pass
upload	Descriptions are uploaded to the database	I am going to upload an asset with a description	The description to be in the database	The description was uploaded to the database	pass
upload	Correct tags are uploaded to the database	I am going to upload an asset with an assortment of tags to the database	The correct tags to be in the database	The correct tags were in the database	pass
upload	The correct images are previewed	I am going to upload images and see if they match what I uploaded vs what I am previewing	I expect for the right images to be previewed	The correct images were previewed	pass
upload	Upload wrong file types returns an error	I am going to upload an invalid file type for each combination of files	I expect for an error to be returned	An error was returned	pass
upload	Uploading no diffuse returns an error	I am going to upload without a diffuse image	For an error to be returned	An error was returned	pass
upload	Inputting no name returns an error	I am going to upload without a name	For an error to be returned	No error was returned and it was added to the database	Fail. I didn't have an expectation in my code so I added one
Asset	Manually entering an incorrect asset id	I am going to enter an invalid asset id into the link area. Both negative and out of bounds.	For an error to be returned	A flask error	Fail. The program crashed. I fixed it by making sure that the route took you to a valid page or returned an error.
Upload	Entering a very long name / description	I am going to enter a very long name	For an error to be returned	Nothing happened and there is a very long name which is bad	Fail. There should be a character limit on the name. I fixed this by adding a character limit and if it exceeds, returns an error.

Data Integrity Testing

What im testing	Evidence
The correct asset name is loaded	
The correct asset description is loaded	

Description

Wet Water

The correct image for the assets is loaded



127.0.0.1:5000/asset/5

5 5 cc

5d.png

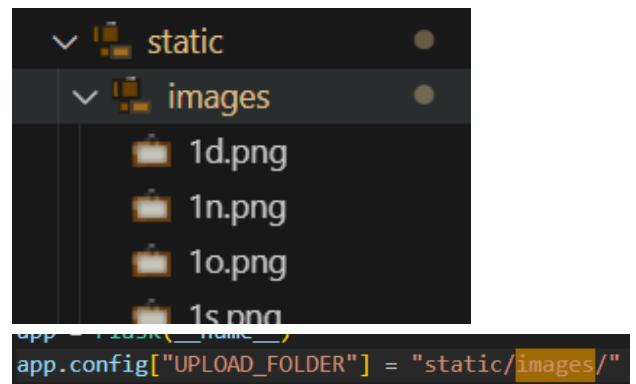


The name of the image matches the asset id

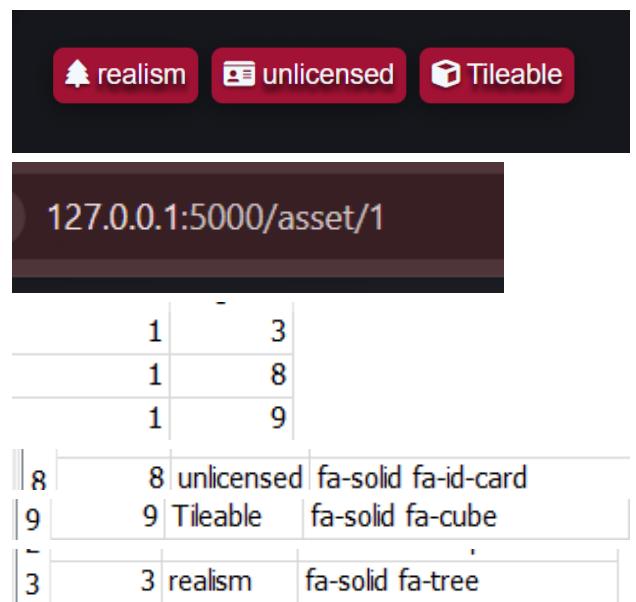
5 5 cc 5d.png



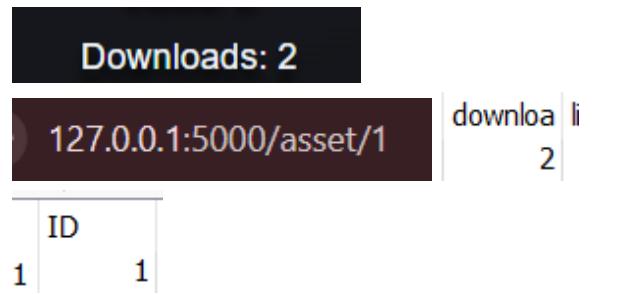
The images are saved in the correct path



The assets tags are consistent with the many to many relationship



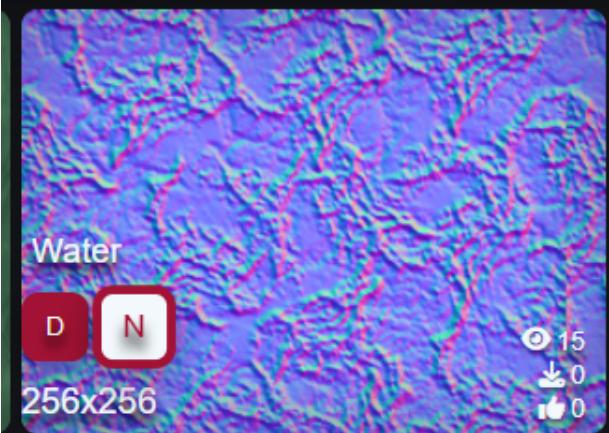
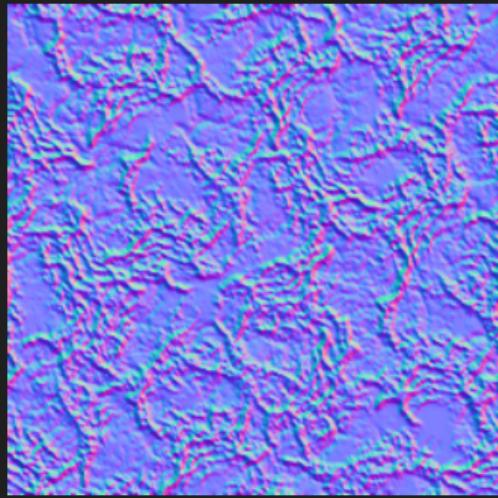
The correct number of downloads are displayed



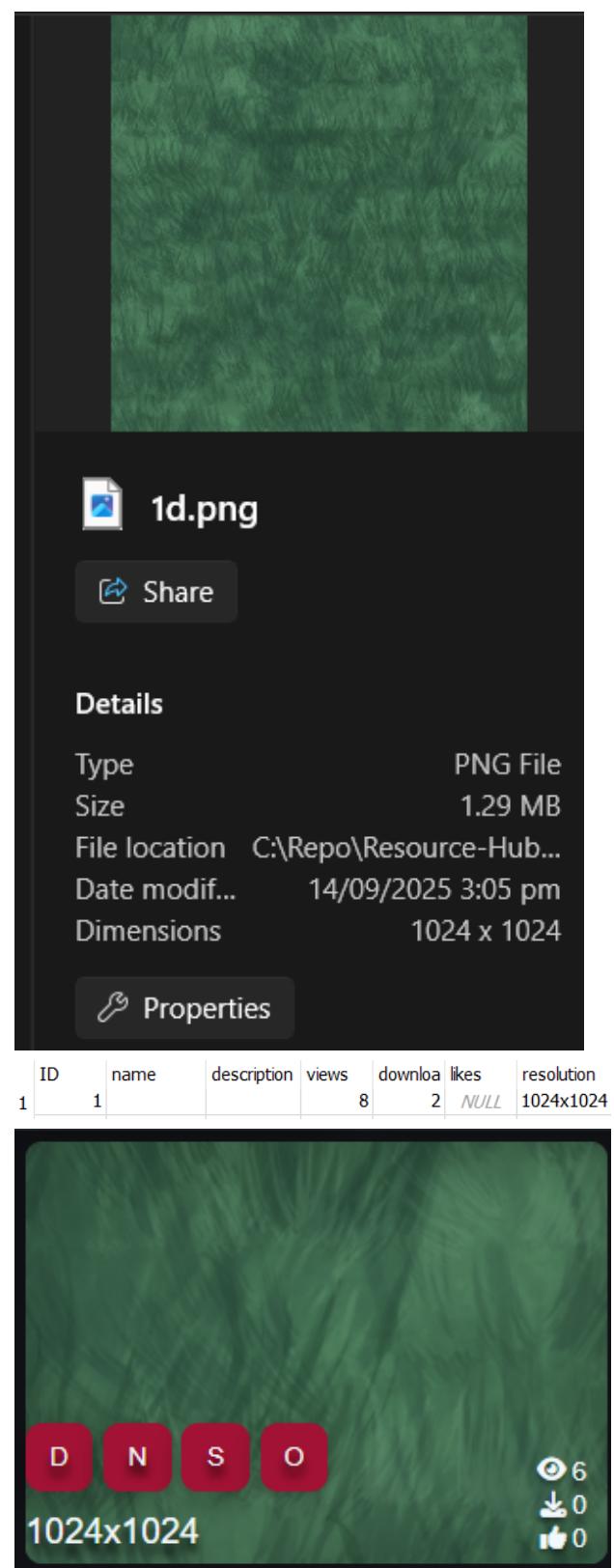
Clicking on an asset takes you to the correct assets page



Hovering over the image type button displays the correct image types image	<p>D N S O</p> <p>realism unlicensed Tileable</p> <p>i 127.0.0.1:5000/asset/1</p> <p>2n.png</p> <p>127.0.0.1:5000/asset/2</p>



The correct resolution is inserted into the database



The correct images are downloaded

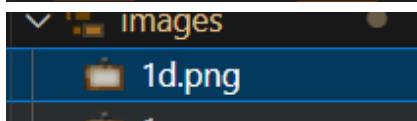
127.0.0.1:5000/download/1/d

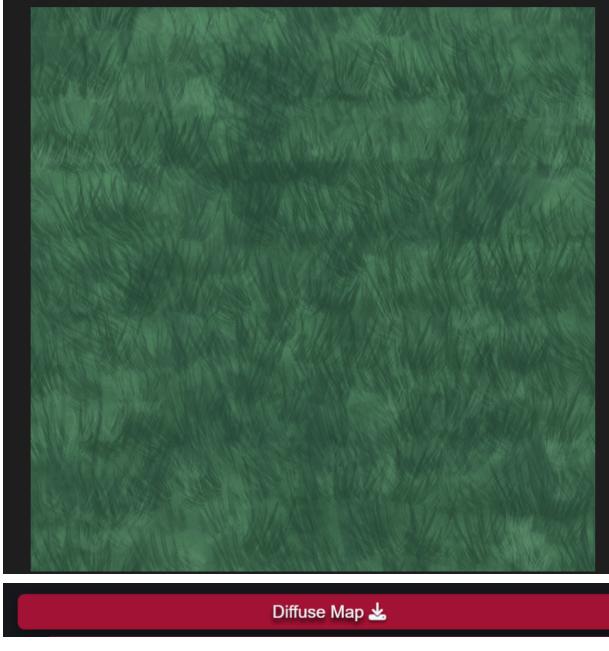
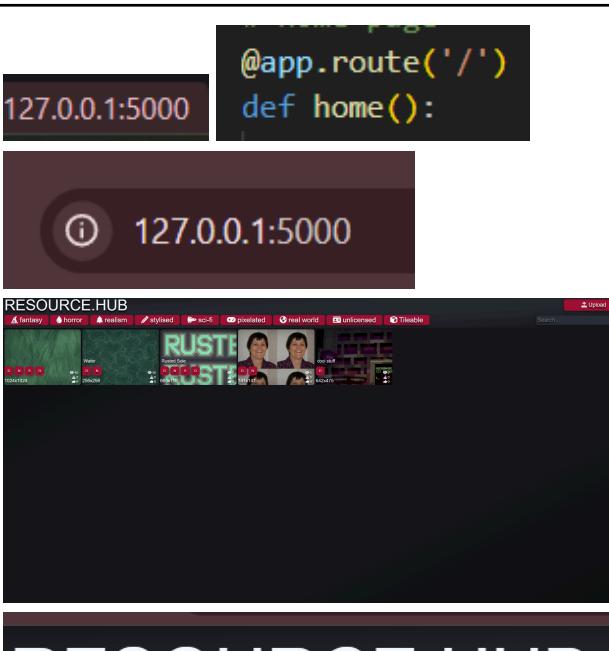
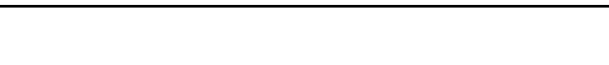


Details

Type	PNG File
Size	1.29 MB
File location	C:\Users\cohen\Downl...
Date modif...	17/09/2025 8:58 pm
Dimensions	1024 x 1024

```
@app.route('/download/<int:asset_id>/<string:image_type>')
def download_image(asset_id, image_type):
```



	
Clicking the home button takes you to the home page	 <pre>@app.route('/') def home():</pre>  <p>RESOURCES.HUB</p>
Clicking on the upload button takes you to the upload page	 <pre>127.0.0.1:5000/upload</pre>  <pre>@app.route('/upload', methods=['GET', 'POST']) def upload():</pre>

