

Sam's Zoo Project Document

Sammuel Farias

September 12, 2025

1 Team Members

Sammuel Farias

2 Project Description

So, this project is basically a simple static website for a made-up zoo called "Sam's Zoo". It's built with plain HTML. The main idea is a homepage that links out to four other pages, one for each animal exhibit: lions, giraffes, monkeys, and penguins. Each of those pages has a quick description and some pictures of the animals.

3 Requirements

3.1 Business Requirements (BR)

- **BR #1 - Homepage:** The user needs to be able to see a main homepage to get what the site is about and see the animal options.
- **BR #2 - Animal Info:** The user should be able to check out pages for the main animals (Lions, Penguins, etc.) to get more info on them.
- **BR #3 - See Pictures:** The user should be able to see photos of the animals on their pages.
- **BR #4 - Easy Navigation:** The user needs a simple way to get back to the homepage from any animal page.

3.2 Functional Requirements (FR)

- **FR #1 - Homepage Content:** The site has to show a homepage with the title "Welcome to Sam's Zoo" and a menu to get to the animal pages.
 - *BR Reference:* BR #1
- **FR #2 - Working Links:** The menu links on the homepage have to work, taking the user to the right animal page when clicked.
 - *BR Reference:* BR #2
- **FR #3 - Animal Page Content:** Every animal page needs a title, a short paragraph, and four pictures of that animal.
 - *BR Reference:* BR #2, BR #3
- **FR #4 - Back Link:** Every animal page has to have a "Back to Home" link that takes the user to the main index.html page.
 - *BR Reference:* BR #4

3.3 Non-Functional Requirements (NFR)

- **NFR #1 - Browser Support:** The site has to look right on new versions of Chrome, Firefox, and Safari.
- **NFR #2 - Usability:** All the links should be easy to find and work correctly on all pages.
- **NFR #3 - Reliability:** All the images need to load properly without any broken links.

4 Software Architecture and Diagram

The architecture is super straightforward—just a multi-page static site. The files are organized to keep things clean: the main `index.html` is separate from the other pages (in a `/pages` folder) and the images (in an `/images` folder). There's no backend or database stuff.

The structure is a simple hierarchy, like in the template doc:

- **Home:** `index.html`
 - **Category:** `pages/lions.html`
 - **Category:** `pages/giraffes.html`
 - **Category:** `pages/monkeys.html`
 - **Category:** `pages/penguins.html`

5 Epics

5.1 Epic #1 - Website Navigation

Requirement Reference: FR #1, FR #2, FR #4

Description: As a user, I want to be able to jump between the homepage and the animal pages easily so I can browse the site without getting stuck.

5.2 Epic #2 - Animal Content Display

Requirement Reference: FR #3

Description: As a user, I want the animal pages to actually show me info and pictures so I can learn something about them.

6 User Stories & Acceptance Criteria

6.1 User Story #1 - Check out the homepage

Epic Reference: Epic #1

Description: As a visitor, I want to see the main homepage so I can get the point of the site and see what animals are featured.

Acceptance Criteria:

- When I go to the site's main URL, the page loads.
- I see the main title "Welcome to Sam's Zoo".
- I see a list of links for "Lions," "Penguins," "Monkeys," and "Giraffes".

6.2 User Story #2 - Go to an animal page

Epic Reference: Epic #1, Epic #2

Description: As a visitor, I want to click an animal's link on the homepage and be taken to its page.

Acceptance Criteria:

- Given I'm on the homepage.
- When I click the "Lions" link.
- Then I land on the "Meet the Lions" page and can see the text and photos.

6.3 User Story #3 - Go back to the homepage

Epic Reference: Epic #1

Description: As a visitor on an animal page, I want a link to get back to the homepage so I can look at other stuff.

Acceptance Criteria:

- Given I'm on any animal page (like lions.html).
- When I click the "Back to Home" link.
- Then I am taken back to the main homepage.

7 Project Schedule

I can't make a real Gantt chart, but here's a basic schedule of how this project would go down.

7.1 Phase 1: Getting Started (1 Day)

- Figure out what the site needs to do.
- Set up the GitHub repo and the folders (/pages, /images).

7.2 Phase 2: Coding the Pages (2 Days)

- Code the main `index.html` page and its nav menu.
- Make a template for the animal pages.
- Make sure all the links between pages work.

7.3 Phase 3: Adding Content (1-2 Days)

- Find all the animal pictures.
- Write the little descriptions for each animal page.
- Put all the text and images on the right pages.

7.4 Phase 4: Finishing Up (1 Day)

- Click through the whole site to find broken links or images.
- Push it all to GitHub Pages to go live.
- Take one last look at the live site to make sure it's all good.