Software Requirements Specification

for

Portfolio Web Site

Version 1.0 approved

Prepared by Declan Dover

NMIT

25/09/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for the Static Portfolio Website. It will define the requirements of the site, content, functional, and non-functional, describe the way in which it will be hosted, and provide low fidelity wireframes as a base for the website's design.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

Potential employers, collaborators, or other students and aspiring developers viewing the page for inspiration or research.

1.4 Product Scope

The product is a portfolio website that will act as a professional showcase for Declan Dover, an IT student at NMIT, in order to fulfil the goal of building an online presence and finding employment. It will present fixed information about skills, projects, and contact details in a clear and structured way. As a static site, the portfolio will be built using only HTML and CSS, without dynamic back-end functionality, and will be hosted on GitHub Pages. End-users will be able to use the site to find out any relevant information about the portfolio owner in a professional sense.

1.5 References

Google SEO Guide

SEO Starter Guide: The Basics | Google Search Central | Documentation. (n.d.). Google for Developers. Retrieved August 31, 2025, from https://developers.google.com/search/docs/fundamentals/seo-starter-guide

The A11y Project

Checklist—The A11Y Project. (n.d.). Retrieved August 30, 2025, from https://a11yproject.com/checklist/

W3 and WAI ARIA

Initiative (WAI), W. W. A. (n.d.). *WAI-ARIA Overview*. Web Accessibility Initiative (WAI).

Retrieved August 30, 2025, from https://www.w3.org/WAI/standards-guidelines/aria/

2. Overall Description

2.1 Product Perspective

The product is a stand-alone static website consisting of multiple HTML pages styled with CSS. It will not depend on a database, back-end logic, or JavaScript frameworks.

2.2 Product Functions

Present “About Me” information about the portfolio owner.

List and describe the portfolio owners skills in an interesting way.

Display projects with images, video, and explanations.

Provide contact details for the project owner and links to their external profiles.

Provide a way to contact the project owner directly.

Allow the end-user to navigate through all the portfolio pages.

2.3 User Classes and Characteristics

IT Student and portfolio owner

User persona one is Declan Dover, an IT student at NMIT. He needs a portfolio website to provide some basic information about himself, describe his skills, provide a curated display of his past projects, and give visitors multiple ways to contact him. As his studies are ongoing, he needs to be able to add more projects and easily update information on his portfolio website. He wants this website to attract attention from fellow students and IT employers in order to build a greater online presence and ultimately find employment.

IT Employer and end-user

User persona two is James Smith, the hiring manager of a local IT company, James is often looking for potential new hires and seeing what kind of prospective employees are out there and what their skillsets look like. James values websites that can load quickly, display skills clearly, and present well-documented examples of work. He wants to assess technical ability, attention to detail, and design sensibility within a few minutes of visiting a portfolio. He also likes to read in depth explanations of projects with examples and images. James views websites on both desktop and mobile devices. He values the ability to be able to contact the owner of a portfolio page in multiple ways, and to see links to other sites so he can conduct more extensive research on a potential new hire.

2.4 Operating Environment

The portfolio will be hosted on GitHub Pages, which provides fast, globally distributed content delivery via a Content Delivery Network (CDN). This means that because of its large network of servers, content can be distributed from the closest server to a user, reducing latency and increasing site performance. Using GitHub Pages also ensures HTTPS is enforced by default, improving performance through HTTP/2 features like multiplexing. This reduces latency and improves overall page load speed. As the portfolio is static, github pages use of caching will also reduce server latency by storing the portfolio in the cache of whichever server the user is accessing for the duration of their browsing, reducing the need for repeated requests

2.5 Design and Implementation Constraints

The Website is to be completely static and contain only HTML and CSS. Each page will be stored as a separate HTML file and navigation will not contain the use of any jump links.

2.6 User Documentation

A readme will be created alongside the website with a description and explanation of how the website can be navigated, hosted, and used.

2.7 Assumptions and Dependencies

Users will have access to a modern browser.

The portfolio will be updated manually by editing HTML/CSS.

The site will be hosted on GitHub Pages.

3. External Interface Requirements

3.1 User Interfaces

The site will be designed reactively for all common devices and screen sizes that end-users will be using. The site will also be designed to be accessible for all modern browser types and accessibility devices.

3.2 Communications Interfaces

For security and performance reasons each page of the site will be served entirely over HTTPS and not HTTP. HTTPS uses end to end encryption to protect data in transit between a server and a client. Hosting the site on GitHub pages ensures that any traffic is over HTTPS by default.

4. Content Requirements

4.1 Home Page

4.2 Purpose

To provide an entry point for the portfolio and some basic information about the portfolio owner

4.3 Requirements

A brief description of the user in a mostly professional sense, such as name, education history and career aspirations.

Some non-professional information such as hobbies or interests.

A picture of the portfolio owner.

A distinct heading that includes the name of the portfolio owner.

Simple color scheme and a minimal design.

4.4 Skills Page

4.5 Purpose

To display the key IT skills of the portfolio owner to the end-user in an interesting way.

4.6 Requirements

List the main IT and design skills of the portfolio owner with a brief description of each.

The description needs to be succinct so that the skills can be read at a glance and quickly understood.

The skills will be split into three sections, front-end skills, back-end skill, and design skills.

Each skill will be displayed as a simple box, featuring the name of the skill, a relevant icon, and a very brief description of the skill.

The skills will also include hover effects that scale up the skill box and add interest through a subtle animation or color changing effect.

4.7 Projects Page

4.8 Purpose

To display a curated selection of the portfolio owners past projects and some explanations of the content.

4.9 Requirements

A diverse selection of three projects.

A thumbnail image to represent each project.

Each project needs a clear, descriptive title.

A short description of each project

A link to a dedicated page for each project containing detailed explanations and more images or other media.

4.10 Individual Project Page

4.11 Purpose

To display images, text and video as well as a further explanation of a project.

4.12 Requirements

A title explaining the project

Some background explaining the purpose of the project

A description of the steps taken when completing the project.

The outcome of the project and how it relates to a professional skill

A link to a dedicated page for each project containing detailed explanations and more images or other media.

Images of the project.

If necessary, a video of the project.

A link back to the main projects page.

4.13 Contact Page

4.14 Purpose

To provide ways for the end-users to contact the portfolio owner.

4.15 Requirements

The contact page needs to list multiple ways to contact the portfolio owner.

A section containing a phone number and email address.

Links to GitHub and social media.

A contact form for the end user to contact the portfolio owner using a text box, and providing their name and email address in order to be contacted in response.

4.16 Navigation Bar

4.17 Purpose

To provide a way for the end-user to move between sections of the portfolio.

4.18 Requirements

The navigation bar needs to have a clear, clickable label of each page of the portfolio website.

The bar needs to remain on the side of the page, occupying the same space on each page and moving with the display if the user scrolls down.

Hover and active effects on each navigation button.

Distinct colors for the button of each page.

5. System Features

5.1 Navigation Structure

5.2 Description and Priority

Navigation structure is essential. Each section of the site (About, Skills, Projects, Contact) will be a separate page, linked via a navigation bar.

5.3 Functional Requirements

Four links (Home, Skills, Projects, Contact)

No jump links.

The navigation bar must always be visible on the left side, and scroll with the page.

Projects must link to their respective pages with a see more button and return to the projects page with a back button.

5.4 Page Layouts

5.5 Description and Priority

It is important that each page follows a consistent structure and layout.

5.6 Functional Requirements

Each page must have header, left navigation, content, and footer.

Layouts must use components that can expand or adapt to added content.

Must be responsive without breaking design.

Must use consistent values for margins and padding.

5.7 Interactive Elements

5.8 Description and Priority

While the portfolio is static, certain interactive front-end features will be implemented purely with HTML/CSS.

5.9 Functional Requirements

Hover effects on links and buttons.

Active effects on links and buttons.

Video controls for video elements.

Hover effects on the project images to expand them. This function will be replaced with a lightbox in the future.

5.10 Contact Form

5.11 Description and Priority

Contact form is critical for providing a means for end users to contact the portfolio owner.

5.12 Functional Requirements

Must list email, phone, location, and links to GitHub/social media.

Must provide input fields: full name, email, message.

Must be easy to navigate in order to encourage contact.

5.13 Multimedia Displays

5.14 Description and Priority

The portfolio website will contain multimedia displays in the form of a video player and multiple images.

5.15 Functional Requirements

Images wherever necessary throughout the portfolio site.

A video on relevant project pages.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Having appropriate file sizes for any media hosted on the site is important to having good performance and fast loading times for pages. For example, if there are any images on the portfolio site that will be used as a thumbnail, the size of the image should be reduced to the largest size that the image will realistically be displayed at.

Using a vector format whenever possible for images will also help keep file sizes small, vector files such as .SVG are stored as a series of lines and directions instead of pixels and can be scaled up to any size without losing quality

The portfolio will be hosted on GitHub Pages, which provides fast, globally distributed content delivery via a Content Delivery Network (CDN). This means that because of its large network of servers, content can be distributed from the closest server to a user, reducing latency and increasing site performance.

As the portfolio is static, github pages use of caching will also reduce server latency by storing the portfolio in the cache of whichever server the user is accessing for the duration of their browsing, reducing the need for repeated requests.

Another important aspect of website performance is the use of browser caching. Since the portfolio is static, browser caching will be leveraged for assets such as images, fonts, and CSS files. This means that once these resources are loaded by the user's browser, they will be stored locally for the user and not need to be loaded repeatedly when navigating through different pages. The cache-control: HTTP header can be used to manually instruct browsers how long to keep static components in their cache

To test if the portfolio is meeting performance requirements Google Lighthouse will be used on each page of the site. This is a tool built into the chrome browser that can analyze a web page and give different scores and metrics based on the page’s performance. The portfolio site needs to achieve a score of at least 85 on the performance test.

6.2 Security Requirements

all parts of the web site are served over HTTPS and not HTTP. HTTPS uses end to end encryption to protect data in transit between a server and a client. Hosting the site on GitHub pages ensures that any traffic is over HTTPS by default.

The site contains limited personal information, only that which is necessary to contact the portfolio owner, and links only to business facing social media or other sites, such as LinkedIn and GitHub, and not more personal pages such as facebook or instagram. These links that are on the site should also be secure and only to addresses that are also served over HTTPS .

6.3 Usability and Access Requirements

Compliance with the ‘Web Content Accessibility Guidelines (WCAG)’.

Compliance with the A11y checklist.

Writing in plain readable language, having descriptive, intuitive text on buttons, links, and other interactive content, and having the correct text alignment direction.

The correct use of semantic HTML. This means appropriately using semantic HTML elements, meaning ones that describe their own meaning such as <header>, <footer>, or <nav>, wherever possible. These tags will help users with screen readers when navigating a page, by signposting and helping to move through the different sections without having to rely on the content of each element for context.

the use of ARIA attributes in some HTML elements of the portfolio site. WAI-ARIA, the Accessible Rich Internet Applications Suite, defines a way to make web content and web applications more accessible to people with disabilities. This is done through the inclusion of attributes such as aria-label that describe the role or state of an element in the context of an accessibility device. The aria-label attribute is used to label things like buttons or navigation controls that don’t have a description in their inner HTML. For example, if closing a popup on a page required clicking a small ‘x’ button, that button element would include the attribute aria-label=”close”, indicating its purpose to screen readers or other accessibility devices that a user might have. This is one of many kinds of ARIA attributes that should be included whenever necessary in a page’s HTML.

Correct fieldset labels on. Fieldset labels are used to associate an input with a specific label text. When a user with a screen reader focuses an input, the correct label text will be read out, allowing the user to effectively fill out forms or other inputs on a page.

Correct alt attributes. An alt attribute is used to describe a multimedia HTML element such as an image or video. Any such element that isn’t purely decorative requires an alt attribute in order for visually impaired users to understand an element's content.

Correct use of tab indexes. The tabindex is used to control how elements on a page will be focused by users navigating without the use of a mouse or other pointer. It can be used to set the order in which elements on a page are focused with the tab key, or to increase efficiency for those users by disabling the ability to focus non-interactive content

a system of visual feedback for user interactions with some elements. Users need clear feedback when they are focusing or hovering over an interactive element, either through borders, contrasting alternate colors, or mouse pointer styles. This feedback will be implemented through :focus, :hover, and :active CSS selectors for any buttons or links on the portfolio site. This will prevent users from clicking or selecting links or buttons that they did not intend to.

6.4 Responsive Design Requirements

In order for the portfolio site to be optimised and functional for the largest possible set of users, the website needs to be designed to be responsive.

The use of responsive CSS display types and units. Display types such as grid or flex box, combined with the use of units that can automatically adjust to a devices display, such as fractions or viewheight, allowing for layout structures to be resized without losing their shape and structure. In a flex box, items grow and shrink depending on the size of their container, the ‘flex’ CSS rule can also be used to define how much space an item will take up in its container.

Use of media queries to ensure responsiveness on different sized devices. Media queries are a technique to tweak CSS styles depending on the properties of a device's screen, such as height and width. These queries can be used to change the values of a CSS style, such as the size of an element's margins or padding, add new properties, such as maximum or minimum sizes for images, or to completely hide or reveal certain elements.

The site should be tested with a range of display sizes by resizing browser windows to any realistic size that a user might have. Chrome's developer tools can also be used as they provide the ability to simulate the viewports of most common mobile and tablet devices.

6.5 Search Engine Optimisation Requirements

The portfolio site requires good search engine optimization (SEO) in order to align with many of the main user goals such as building an online presence, being seen by potential employers, and allowing any other users to find their content.

Optimisation for indexing speed and crawlability. This requires an intuitive website structure and internal linking, and pages that are well optimised for performance. Because the portfolio site is static, it will benefit from fast indexing and avoid the risk of some content being hidden because of JavaScript.

The use of correct semantic HTML for SEO. Search engine indexes will place a higher importance on the inner HTML of some page elements. These are things like a page's headings and title. Because of this it is important that each page has a good relevant title and headings, and that each section of content is labelled by a clear heading that describes that section's content.

Correct use of meta tags. The meta tag can provide descriptions of a page that is not visible to end users but can be seen and used for search indexing. It can contain things like a description of the page, certain keywords relevant to the page's content, and information about the page's author

Correct use of alt attributes. Another HTML element that the site will use for SEO is the 'alt' attribute. This attribute is used to describe a multimedia element on a page so that search engines can understand that element's purpose and content.

Use of the 'canonical' tag. Another HTML tag that will improve the SEO of the site is the 'canonical' tag. This tag is a link in the head section of an HTML page that informs search engines which page should be prioritised in the case of pages with repeated content

The use of off page SEO. This refers to any SEO factors that come from outside the site, such as the frequency of links to the page on other sites, and on where these links appear. This can be done by posting links to the portfolio site on other pages, such as content the portfolio owner has written on another site, or an aggregation of links to pages on an article about portfolio pages

Appendix A: Glossary

HTML: The standard markup language used to structure content on the web.

CSS: A style sheet language that controls the presentation and layout of web pages.

JavaScript: A programming language that enables interactive and dynamic functionality in web applications.

SEO: Search Engine Optimization, the practice of improving a website's visibility in search engine results.

HTTP: Hypertext Transfer Protocol, the standard protocol for transferring data between a browser and a web server.

HTTPS: A secure version of HTTP that uses encryption (SSL/TLS) to protect data exchanged between a browser and a server.

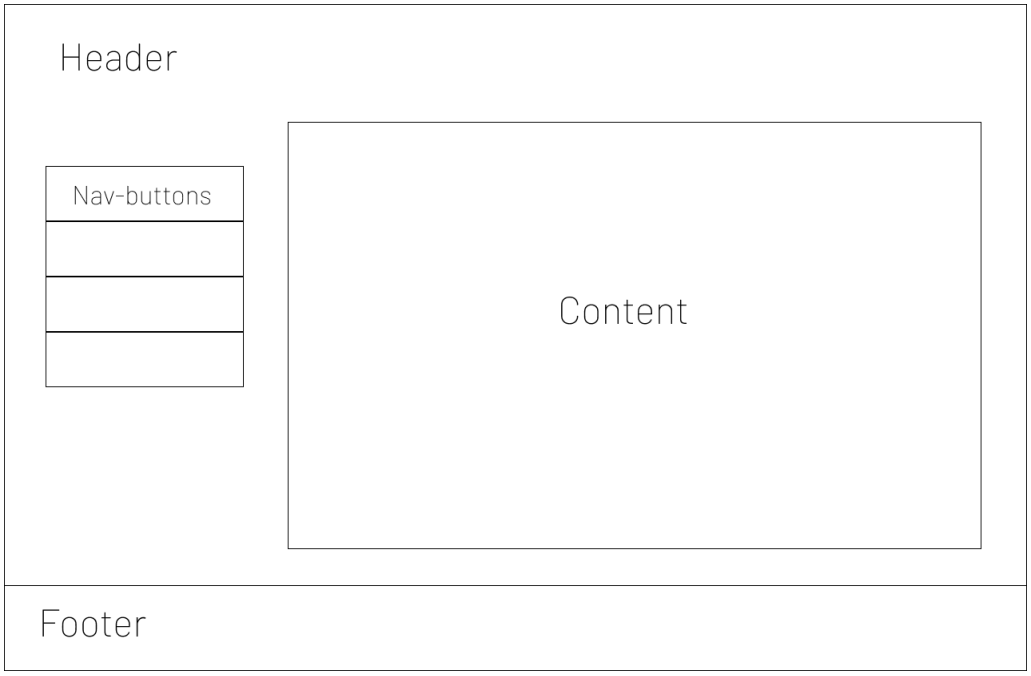
A11y: Short for accessibility, it refers to designing digital content so it can be used by people with disabilities.

WCAG: Web Content Accessibility Guidelines, an international set of standards for making web content more accessible.

Appendix B: Analysis Models

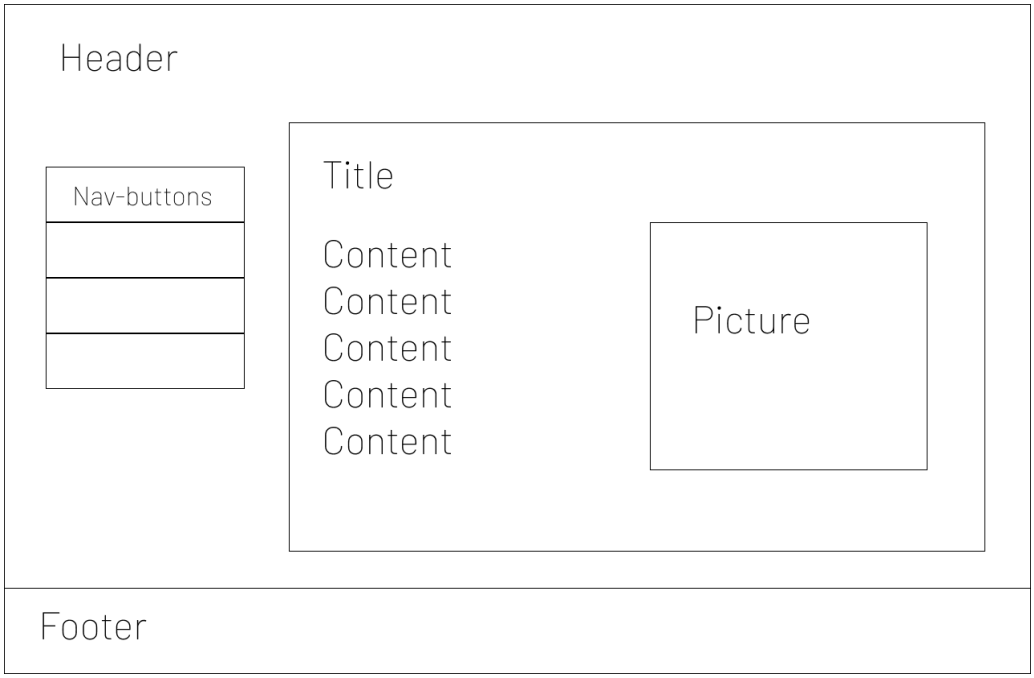
General page Layout

Wireframe

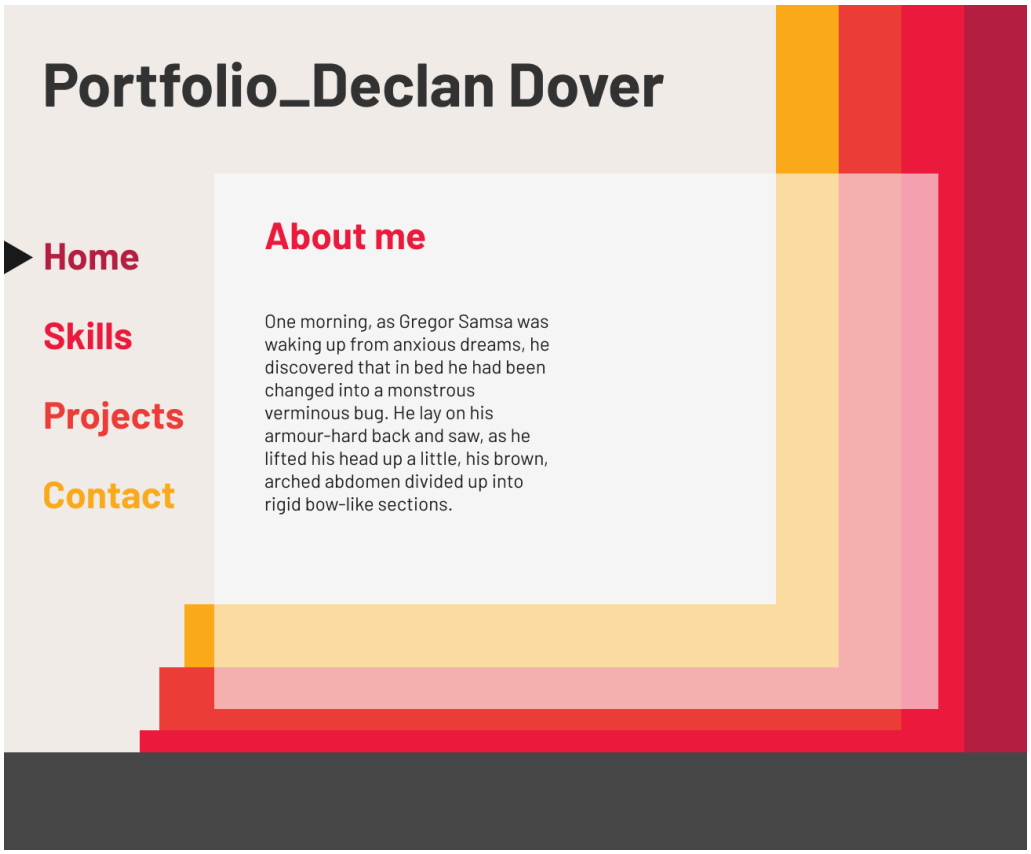


Home page

Wireframe

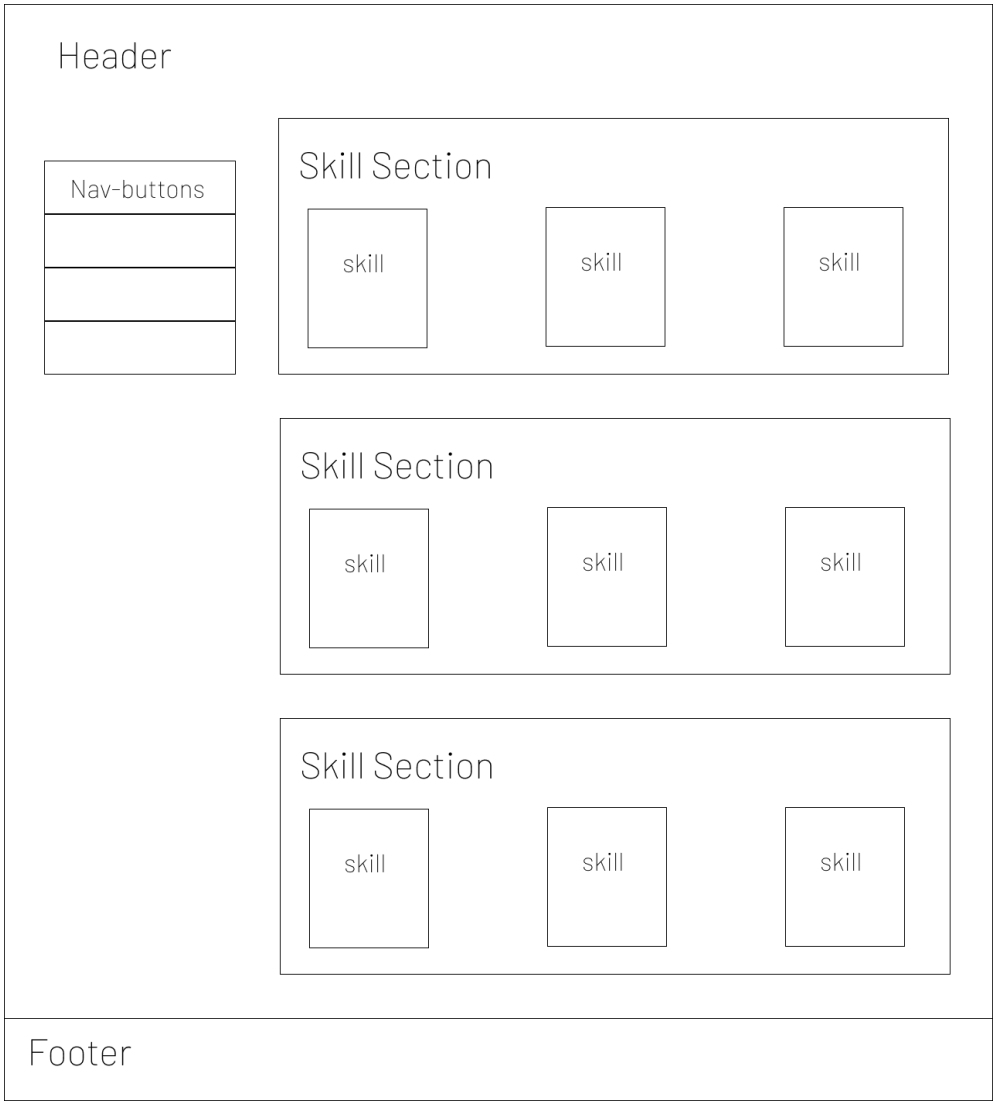

Header
Nav-buttons
Title
Content
Content
Content
Content
Content
Picture
Footer

Mockup

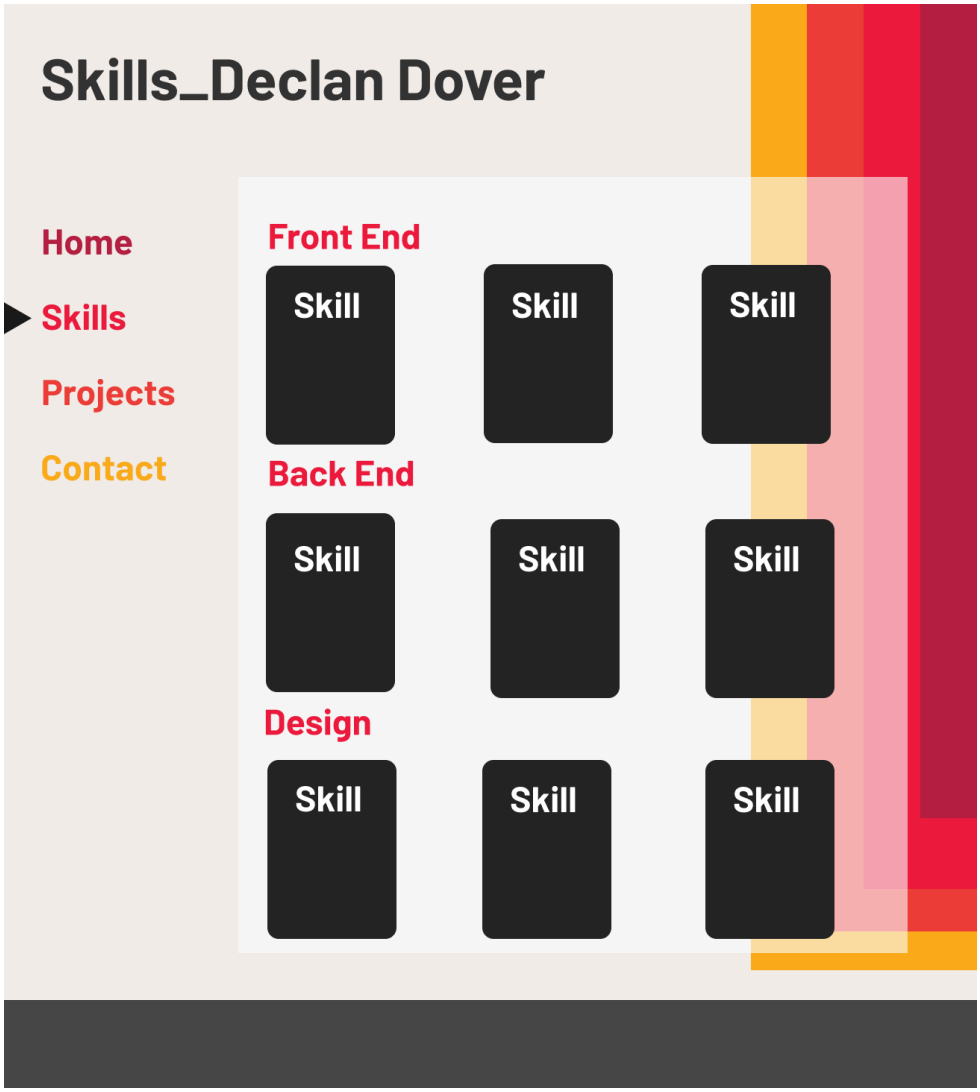

Portfolio_Declan Dover
Home
Skills
Projects
Contact
About me
One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections.

Skills Page

Wireframe

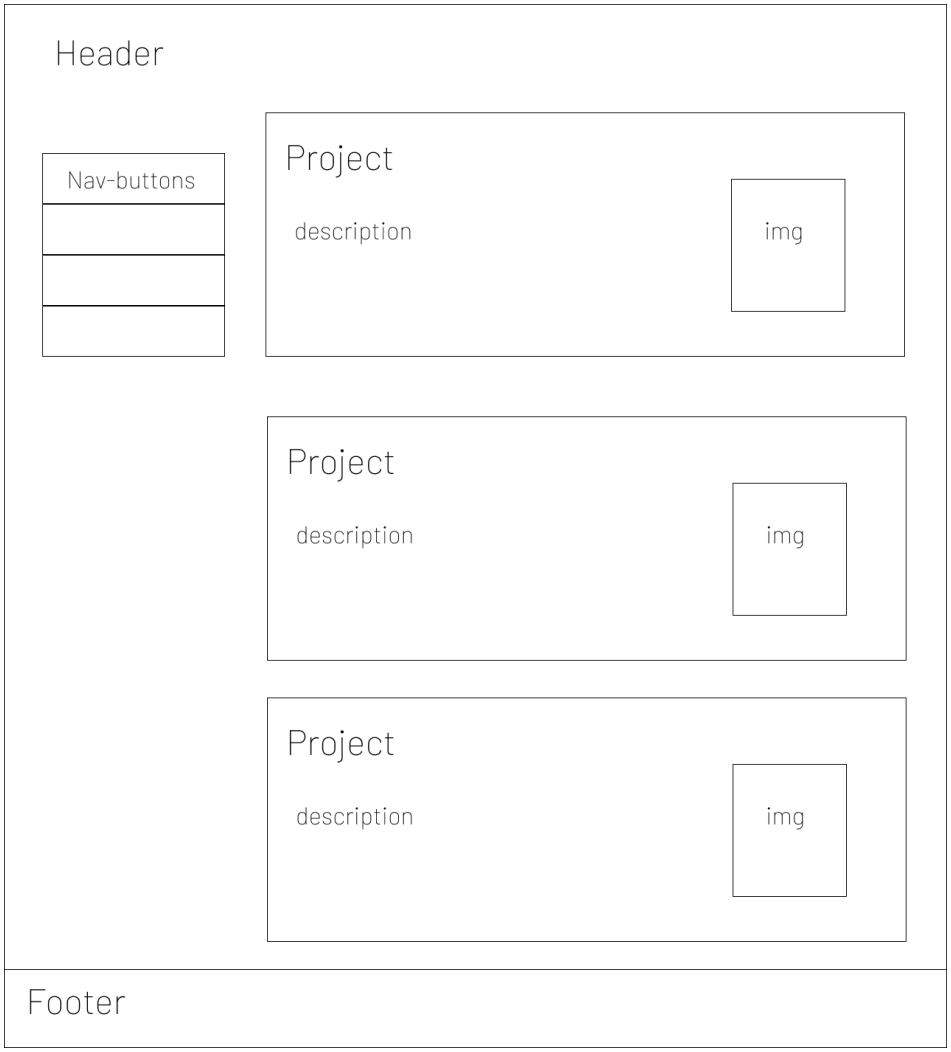


Mockup

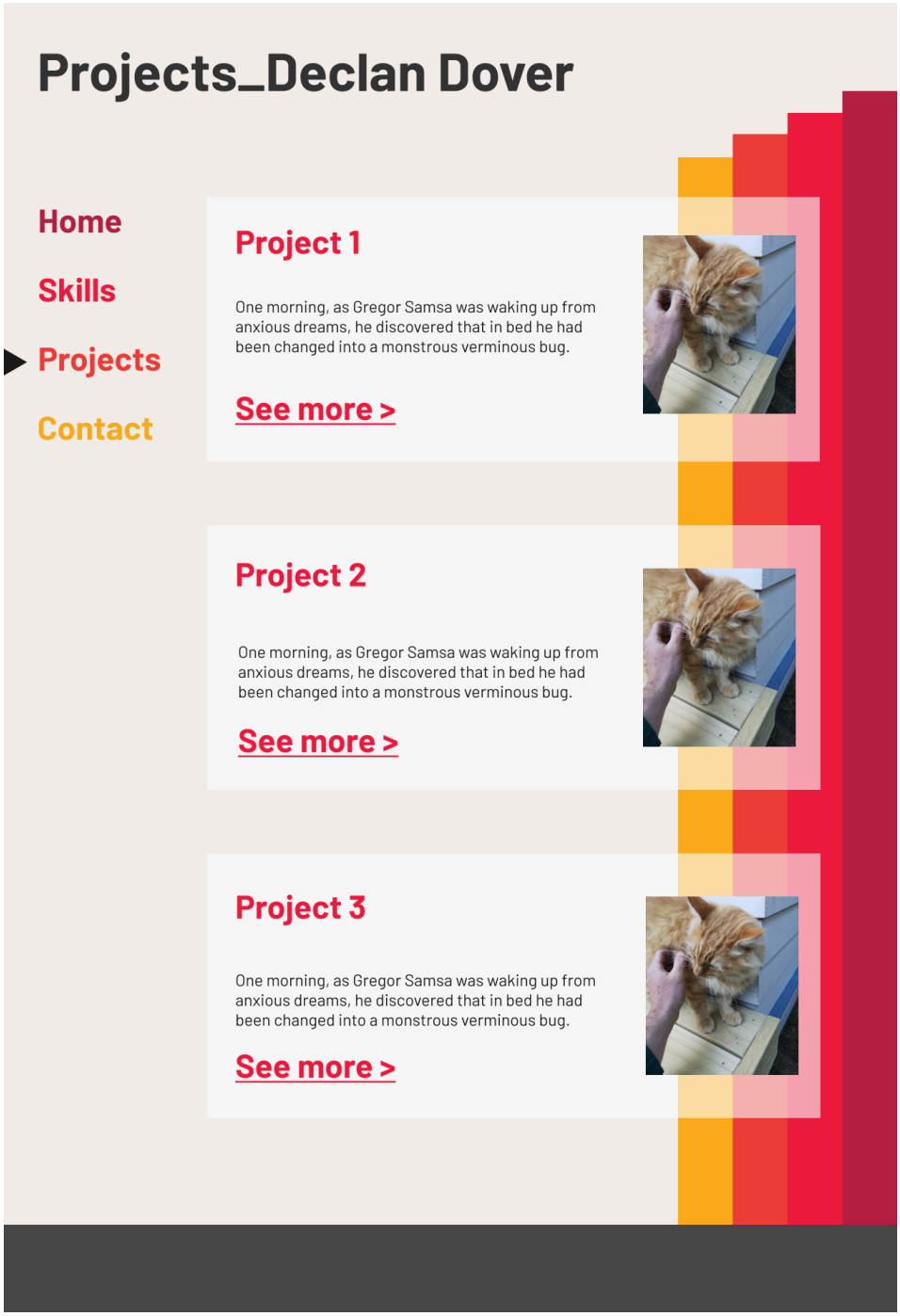


Projects page

Wireframe

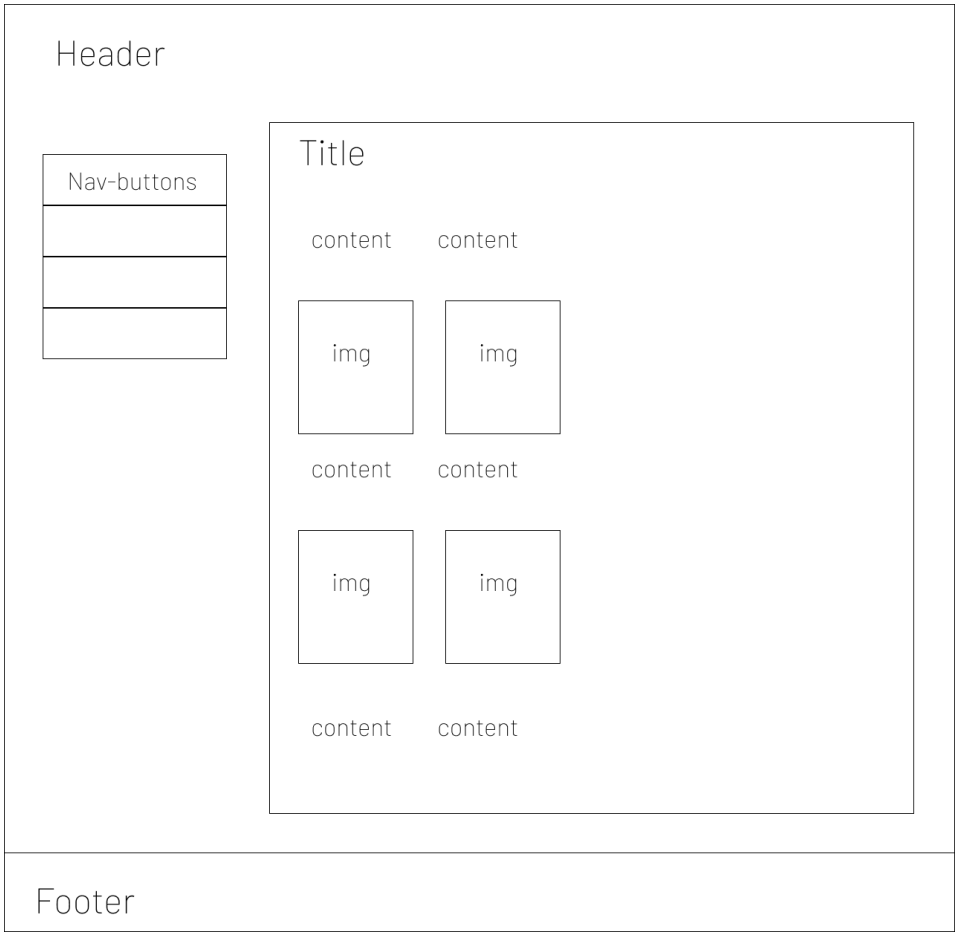

Header
Nav-buttons
Project
description
img
Project
description
img
Project
description
img
Footer

Mockup



Individual Project Page

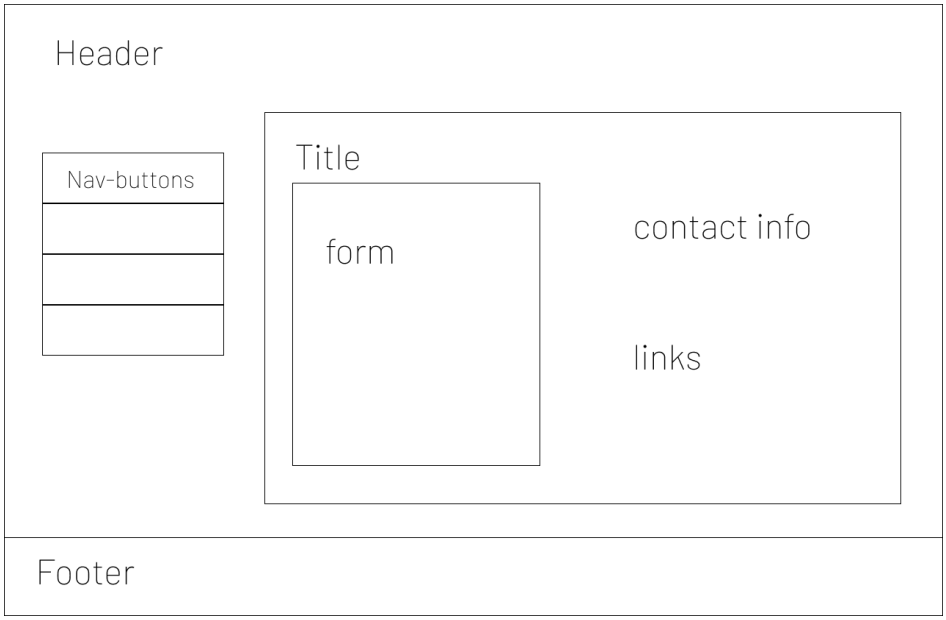
Wireframe


Header
Nav-buttons
Title
content
content
img
img
content
content
img
img
content
content
Footer

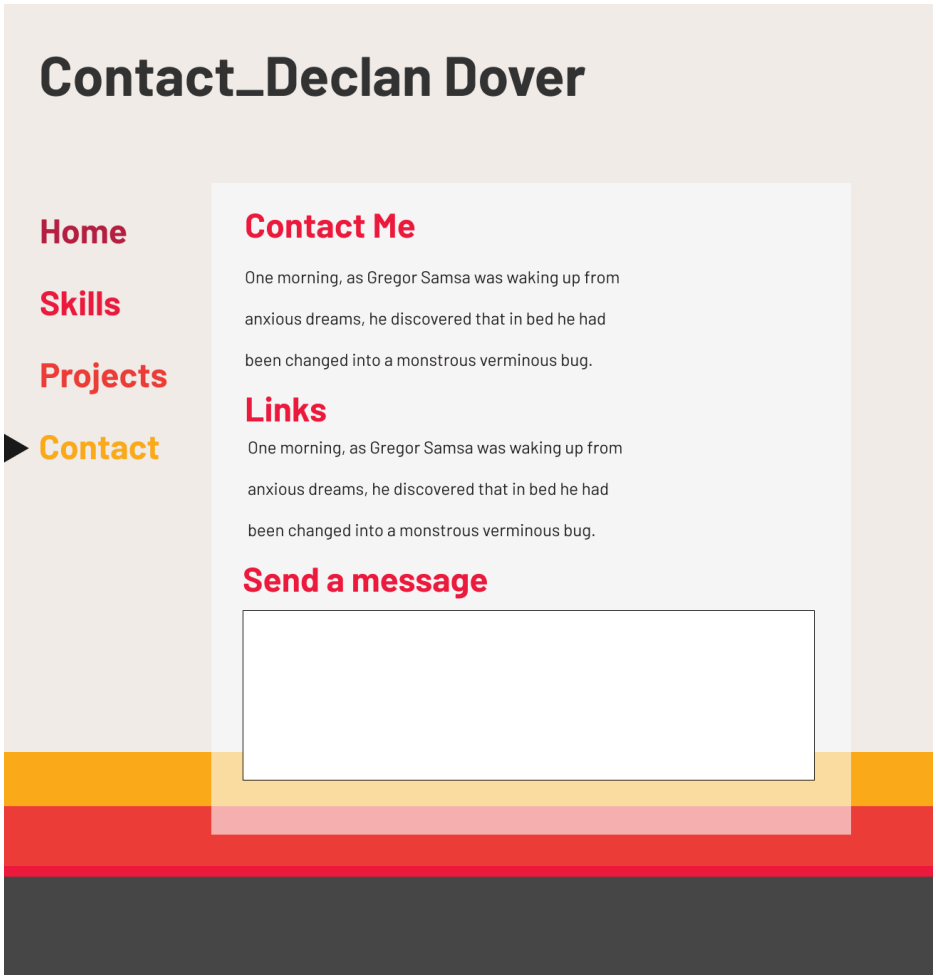
Mockup

Contact Page

Wireframe


Header
Nav-buttons
Title
form
contact info
links
Footer

Mockup


Contact_Declan Dover
Home
Skills
Projects
Contact
Contact Me
One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug.
Links
One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug.
Send a message