DGT :: Web Design Assessment Internal

AS91880

By James Malsher

Contents

[Overview 3](#_Toc15051382)

[Requirements 4](#_Toc15051383)

[Social 4](#_Toc15051384)

[Cultural 4](#_Toc15051385)

[Legal 4](#_Toc15051386)

[Ethical 4](#_Toc15051387)

[Intellectual property 4](#_Toc15051388)

[Privacy 4](#_Toc15051389)

[Accessibility 5](#_Toc15051390)

[Usability 5](#_Toc15051391)

[Functionality 5](#_Toc15051392)

[Aesthetics 5](#_Toc15051393)

[Sustainability and future proofing 5](#_Toc15051394)

[End-user considerations 5](#_Toc15051395)

[Health and safety 6](#_Toc15051396)

[My Design and Build Process 7](#_Toc15051397)

[New project 7](#_Toc15051398)

[Responsive design 7](#_Toc15051399)

[Framework 7](#_Toc15051400)

[Wireframe 7](#_Toc15051401)

[The beginning 9](#_Toc15051402)

[Grid Layout 10](#_Toc15051403)

[Building the wireframe 11](#_Toc15051404)

[Validation of code 20](#_Toc15051405)

[Different Browsers: 21](#_Toc15051406)

[Microsoft Edge: 21](#_Toc15051407)

[Google Chrome: 21](#_Toc15051408)

# Overview

I have built a website that I am happy with and I am using that as my test whilst I am building my final site.

This document is a timeline of the steps I have taken and the progress I have made to create my website.

# Requirements

I created a list of requirements to meet the excellence grade in the AS91880 achievement schedule.

I used a process called MoSCoW which stands for Must, Should, Could and Wont. My requirements were grouped based on the relevant implications in the standard.

## Social

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| The website must have some value to people and not just be about me and my hobbies | I want to make sure people will get value from my website either because it has useful information or because it helps them do something  I can test this by seeing if visitors will comeback to the website more than once | Must |

## Cultural

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Option for English or Maori | I want to make sure that language is not a barrier for people using the site | Should |

## Legal

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| A Page with Legal Statement | Terms and conditions | Must |
| Copywrite message on each page |  | Must |

## Ethical

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| I want to make sure people are happy using this site | It should be child friendly and the kind of page people are happy to share with friends | Must |

## Intellectual property

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| I want this to be free for anyone to use | All of the information, ideas, design and concept will have been learnt from other people and I am not charging for it, there for there is no intellectual property  I will ensure that I have not used someone else’s intellectual property | Must |

## Privacy

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| A Page with Legal Statement | Require page with legal statement | Must |

## Accessibility

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| I want people to use my sight even if they are visually impaired | On any device, all of the controls should be easy to use, and I will use the right programming to support devices that can read screens | Must |
| Colour choices | I will be careful to use colour combination that aren’t a problem for people who are colour blind or visually impaired | Should |

## Usability

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Responsive Design | The page must rerender when the window size is changed. This will also make sure it looks good on small screens like iPads and phones | Must |

## Functionality

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Demonstrate functional ideas | I will use some simple functionality to show what can be done without it being too complex | Must |

## Aesthetics

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Pleasant and easy to use | I will use a balance of text, images and colours that doesn’t distract from the content but creates a nice experience for the user | Should |

## Sustainability and future proofing

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Use Standards |  | Must |
| Commented Code |  | Must |
| Follow Examples | I will look for examples online so I can learn from professionals. | Must |

## End-user considerations

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| Low band width users | Image sizes will be suitable for mobile users | Must |
| Location of clickable areas | Easy to find, easy to click on any screen size | Must |
| Use familiar processes | I don’t want to user to guess how to use my site | Must |
| It should be able to use my site with either mouse, keyboard or touch |  | Must |

## Health and safety

|  |  |  |
| --- | --- | --- |
| Requirement | Notes | MoSCoW |
| The site should not put people in harm’s way | There will be not content that is disturbing or makes people question their own ability or will encourage people to do something that would fiscally harm themselves or others | Must |

# My Design and Build Process

## New project

I cloned the Repository given to me by my teacher into C:/ Dev/AS91880 using the GitHub Desktop Client so that I have a repeatable structure on my laptop for all of my assessments

The repository not related to this assessment; I chose not to remove these because they might be used for something else.

## Responsive design

The relevant implications of usability, accessibility, functionality, end-user considerations and aesthetics all require my website to look good and be easy to use on many different device sizes ad with people with limited visibility or usability.

There is a design method called responsive where a website makes several changes depending on where it is being rendered, sometimes this is just text and pictures changing size but it can also include changing the layout, minimizing the menus till they are required or hiding entire sections till they are needed.

The user should feel they have the same feel throughout my website regardless of the device they are using because of the responsive design will make it suitable for their needs.

## Framework

A responsive design requires the browser to respond either to the device type or the screen size. I tried doing this with HTML and DIVs but found that is requires CSS or Java Script to function.

There are many frameworks that do all the hard work for you, I was tempted to use these and I know they would have made a great webpage but I wouldn’t have learnt how to do it myself.

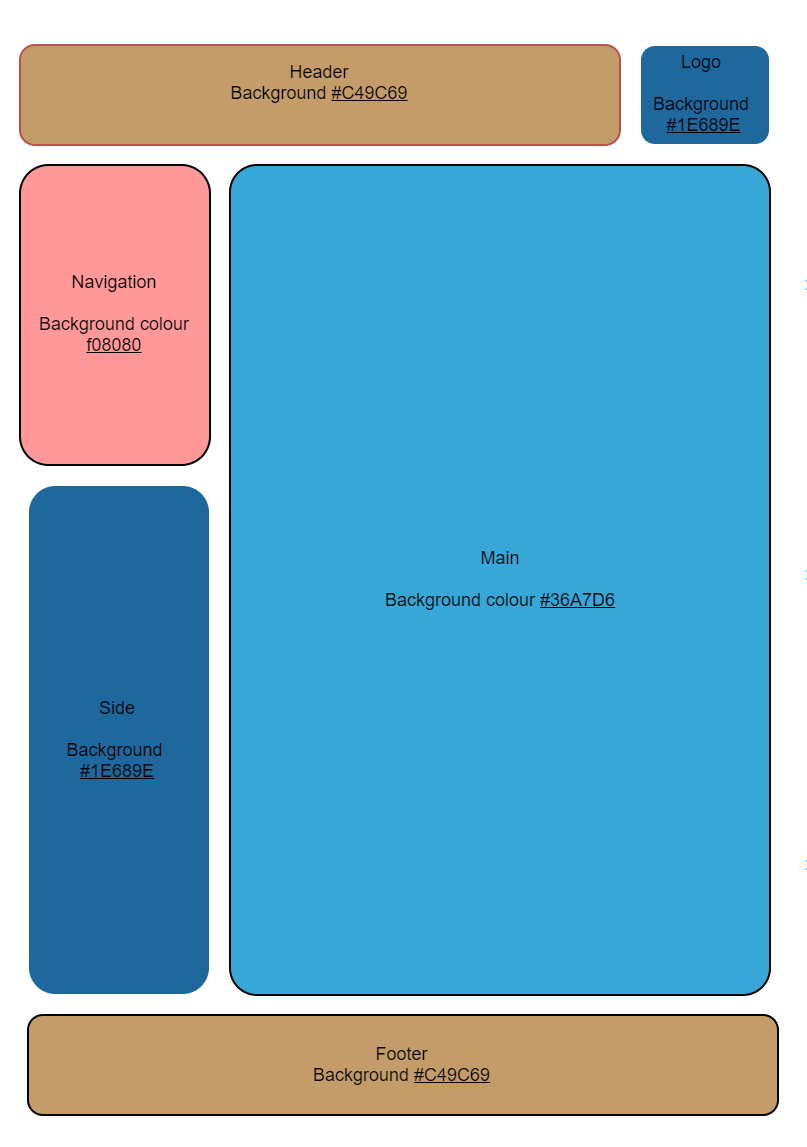
All the development of this page has been done by finding examples online, but I have taken small pieces at a time to learn how they work and fit together.

## Wireframe

We were asked to use colours from an image that we like. I didn’t realize at the time that the one that I chose was a little bit tricky. Here it is with the output of the extraction tool Canva found.



I have used these colours in my wireframe I am not sure they complement each other very well, so may look to change them when the website has more content in it.



I found a lot of examples online of website layouts that are responsive but weren’t quite the same as my wireframe. The easy option would e to change my wireframe, but I wanted to learn how to create this myself using whatever I could find in the examples. The biggest problem was having two DIVs on the left-hand side that needed to act or move together.

## The beginning

I started with a very simple index.html file as my main page. This is commit **59dc198** on Git Hub.

<!DOCKTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

</style>

</head>

<body>

This is my website.

</body>

</html>

This renders in the browser with just the title and a blank page. The important line here is the meta tag which sets the view port to the full width of the device as seen by the browser window. It also sets the initial scale as I don’t want to simply make things bigger or smaller based on the resolution of the screen. I found this online in a tutorial and it looked like a good place to start.

## Grid Layout

For my website I wanted a menu so that the user could go to the various pages that I have these include Legal and privacy as required by the various implication in the standard. I looked at menus that go along the top of websites as well as those that go down the sides and I preferred.

I started with a very simple grid layout that had all of the sections but without any of the positioning.

This is commit **f7765ba** on Git Hub.

DOCKTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

</style>

</head>

<body>

<div class="header">

This is the header.

</div>

<div class="logo">

This is the logo

</div>

<div class="nav">

nav

</div>

<div class="main">

Main

</div>

<div class="side">

side

</div>

<div class="footer">

footer

</div>

</body>

</html>

This produces a page in the browser that looks like this.

This is the header.

This is the logo

nav

Main

side

footer

There is no layout, the font is standard and you can’t see where the DIVs start and finish. However, it does prove that the content of the DIVs get rendered. I gave a class name to each of the DIVs so now I can start controlling what they look like and where they are displayed.

## Building the wireframe

Now that I have content from each of the DIVs being rendered, I will start to add some style so that the colours, positions and size are correct.

The width parameter of a DIVs style makes it posable for me to set the width how I want them. I have then used float so that DIVs sit side by side. For example if I set the header to 80% width and my logo to 20% width and both of them are float, they appear on the same line. This is the style section of my code.

<style>

.header {

background-color: #c49c69;

width: 80%;

float: left;

}

.logo {

background-color: #1e689e;

width: 20%;

float: left;

}

.nav {

background-color: #f08080;

width: 20%;

float: left;

}

.main {

background-color: #36a7d6;

width: 80%;

float: left;

}

.side {

background-color: #1e689e;

}

.footer {

background-color: #c49c69;

width: 100%;

float: left;

}

</style>

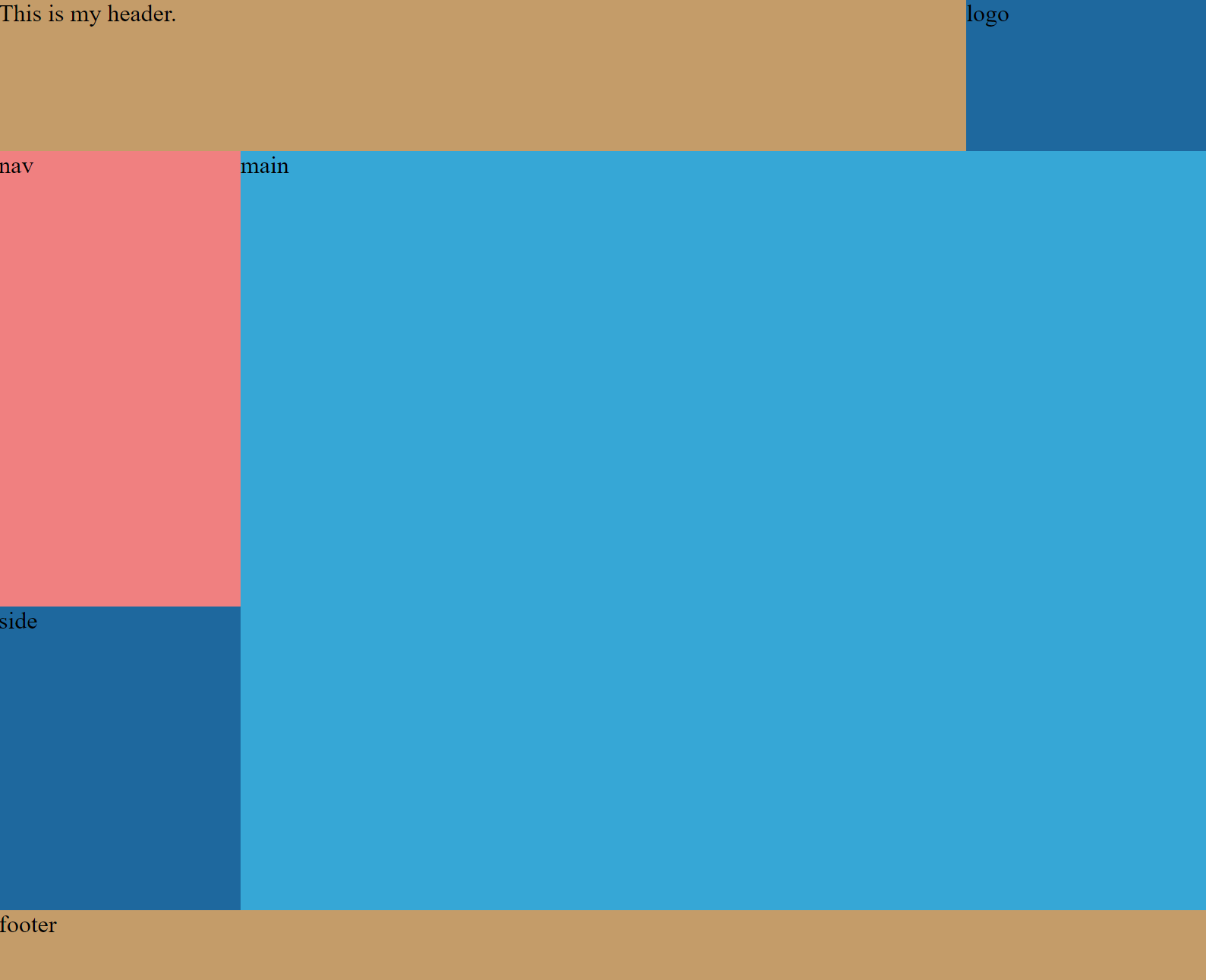
This is what it looks like when it is rendered. I have checked this into Git Hub. This is **d22589d.**



There are a lot of things wrong with this, but a lot of things that are good to. The side DIV should be under the Nav DIV and the Main DIV should be twice as high.

I think I need to have DIVs inside other DIVs. For example, Nav and Side could be in a DIV and that DIV could sit alongside the main DIV. I need to practice with this.

After trying lost of different options and discovering nested DIVs, I have been able to create this which matches my wireframe in the way it is laid out.



To achieve this, I created a DIV called row1 which contained the header DIV and the logo DIV.

I gave all of the DIVs a float value of left so that they would automaticly align with each other.

The row1 DIV was set to 100% width with the header set to 80% and the logo set to 20%. This made sure they always stayed on the top line together.

This is the section of the <body> that makes this work.

<div class="row1">

<div class="header">

This is my header.

</div>

<div class="logo">

logo

</div>

</div>

And this is the CSS.

.row1 {

width: 100%;

height: 100px;

float: left;

}

.header {

background-color: #c49c69;

width: 80%;

height: 100%;

float: left;

}

.logo {

background-color: #1e689e;

width: 20%;

height: 100%;

float: left;

}

The much harder part was having my NAV and SIDE DIVs on top of each other as I wanted these to be the same height as Main.

I used the nested DIVs again but this time created a second row and then two columns and then put the NAV and SIDE DIVs in the first column.

<div class="row2">

<div class="leftcolumn">

<div class="nav">

nav

</div>

<div class="side">

side

</div>

</div>

<div class="rightcolumn">

<div class="main">

main

</div>

</div>

</div>

And here is the CSS that I created.

.row2 {

width: 100%;

height: 500px;

float: left;

}

.leftcolumn {

width: 20%;

height: 100%;

float: left;

}

.rightcolmn {

width: 80%;

height: 100%;

float: left;

}

.nav {

background-color: #f08080;

width: 100%;

height: 60%;

float: left;

}

.side {

background-color: #1e689e;

width: 100%;

height: 40%;

float: left;

}

.main {

background-color: #36a7d6;

width: 80%;

height: 100%;

float: left;

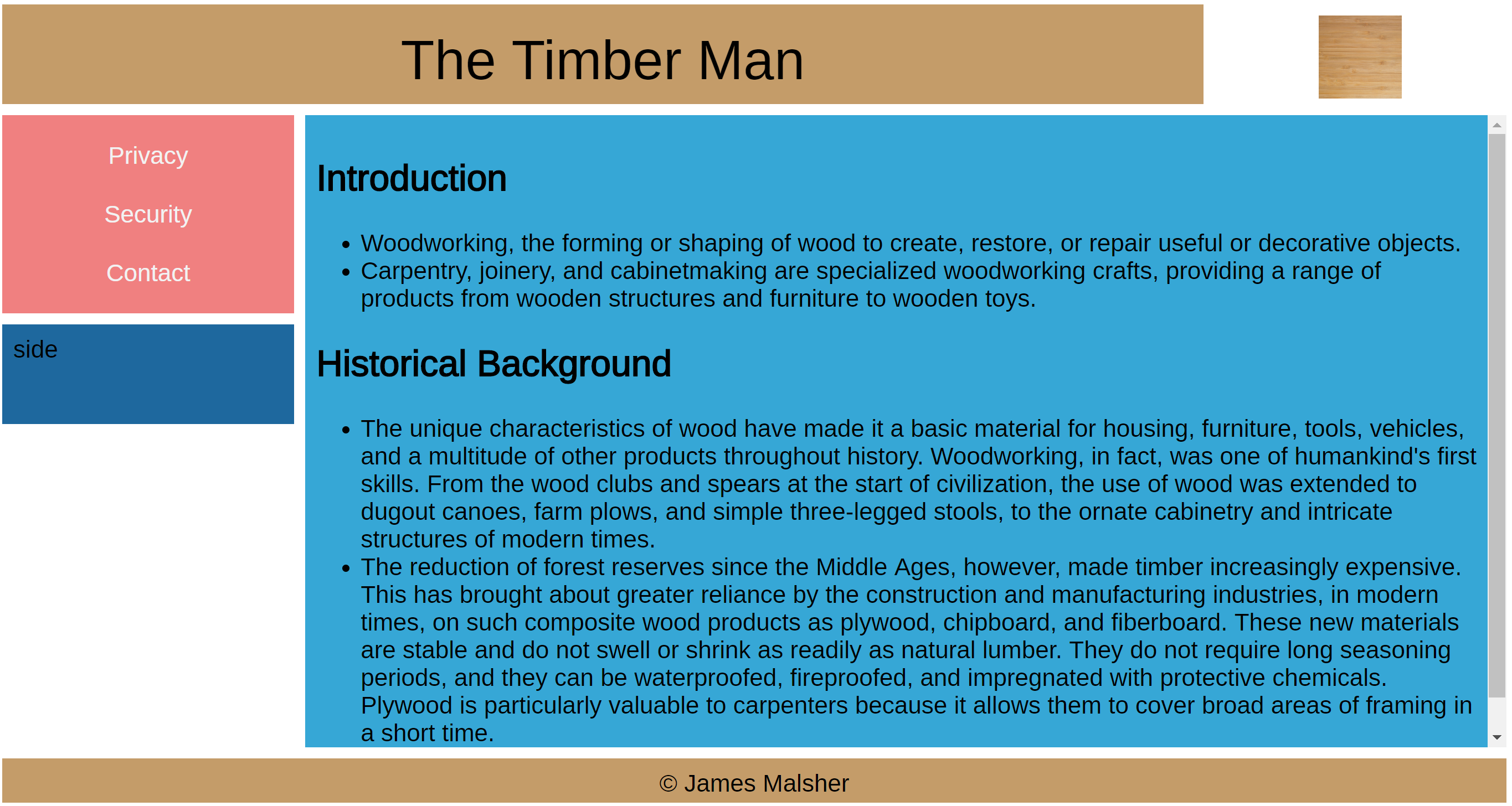
}

You can see from the CSS that I entered a height value for each of the DIVs, sometimes just for the outer DIV depending on what I wanted to size. This is just to help me see the layout because the size should be set by the content which I haven’t added yet. I have checked this into Git Hub as commit

The commit is much larger than my one index.html file because visual studio made changes to all files in the repository.

Next I wanted to work on the Navigation because this looked trickey.

I have fixed the Navigation. I have also improved the DIVs so that they stand out. I have made the main box scrollable. I have forced the footer to stay at the bottom.



This is the source code.

<!DOCKTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'>

<style>

body {

/\*I wanted to use the whole of the screen\*/

padding: 0px;

margin: 0px;

/\*I chose a nicer font\*/

font-family: 'Arimo';font-size: 22px;

}

.row1 {

/\*All rows are 100% width of the screen\*/

width: 100%;

/\*My header is 100px high\*/

height: 100px;

/\*All rows align with the left hand edge of the screen\*/

float: left;

}

.header {

background-color: #c49c69;

width: 80%;

height: 100%;

float: left;

/\*The page header text is in the middle of the DIV\*/

text-align: center;

font-size: 50px;

vertical-align: middle;

line-height: 100px;

/\*Each DIV has a white inner border to make them stand out \*/

box-sizing: border-box;

border: 5px solid white;

}

.logo {

width: 20%;

height: 100%;

float: left;

padding: 10px;

box-sizing: border-box;

border: 5px solid white;

/\*The logo is centered in the DIV\*/

text-align: center;

}

.row2 {

width: 100%;

float: left;

/\*row2 always starts just bellow row1\*/

position: absolute;

top: 100px;

/\*row2 fills the space down to the footer\*/

bottom: 50px;

}

.leftcolumn {

/\*leftcolumn and rightcolumn fill the width of row2\*/

width: 20%;

height: 100%;

float: left;

}

.rightcolumn {

width: 80%;

height: 100%;

float: left;

}

.nav {

background-color: #f08080;

width: 100%;

height: auto;

float: left;

/\*nav will always be the right size for links\*/

overflow: hidden;

padding: 10px;

box-sizing: border-box;

border: 5px solid white;

}

.nav a {

float: left;

/\*Each link is easy to use on a touch screen\*/

display: block;

color: #f2f2f2;

text-align: center;

/\*The link is easy to read\*/

padding: 14px 0px;

text-decoration: none;

width: 100%;

}

.nav a:hover {

/\*The link is easy to see when the mouse hovers over it\*/

background-color: #ddd;

color: black;

}

.side {

background-color: #1e689e;

width: 100%;

float: left;

/\*If I knew what to use this for, I could set the height\*/

height: 100px;

padding: 10px;

box-sizing: border-box;

border: 5px solid white;

}

.main {

background-color: #36a7d6;

width: 100%;

height: 100%;

float: left;

padding: 10px;

box-sizing: border-box;

border: 5px solid white;

/\*There is to much text for a small screen, so this is scrollable\*/

overflow-y: scroll;

}

.row3 {

height: 50px;

width: 100%;

float: left;

text-align: center;

position: absolute;

bottom: 0px;

}

.footer {

background-color: #c49c69;

height: 100%;

padding: 10px;

box-sizing: border-box;

border: 5px solid white;

}

/\*When the screen is less than 800px wide, row2 changes its layout\*/

/\*This is responsive design. It needs more work\*/

@media screen and (max-width: 800px) {

.leftcolumn, .rightcolumn {

width: 100%;

}

}

</style>

</head>

<body>

<!--The body is made up of three rows called row1, row2 and row3-->

<div class="row1"> <!--This row contains the Header and Logo-->

<div class="header"> <!--This DIV is nested inside the row1 DIV-->

The Timber Man

</div>

<div class="logo"> <!--I have used a simple image for the logo-->

<a href="Index.html"> <!--Clicking on the logo takes you to the main page-->

<img src="img/wood1.jpg" border="0" width="200" height="75"/>

</a>

</div>

</div>

<div class="row2"> <!--This was very trickey as there are multiple nested DIVs-->

<div class="leftcolumn"> <!--This has two DIVs stacked vertically-->

<div class="nav"> <!--This is where all links are held-->

<a href="Privacy.html">Privacy</a>

<a href="Security.html">Security</a>

<a href="Contact.html">Contact</a>

</div>

<div class="side">

side

</div>

</div>

<div class="rightcolumn"> <!--This Column contains the Main text of the page-->

<div class="main"> <!--I have used unorderd list items-->

<h2>Introduction</h2>

<ul>

<li>

Woodworking, the forming or shaping of wood to create, restore,

or repair useful or decorative objects.

</li>

<li>

Carpentry, joinery, and cabinetmaking are specialized woodworking crafts,

providing a range of products from wooden structures and furniture to wooden toys.

</li>

</ul>

<h2>Historical Background</h2>

<ul>

<li>

The unique characteristics of wood have made it a basic material

for housing, furniture, tools, vehicles, and a multitude of other

products throughout history. Woodworking, in fact, was one of humankind's

first skills. From the wood clubs and spears at the start of civilization,

the use of wood was extended to dugout canoes, farm plows, and simple three-legged

stools, to the ornate cabinetry and intricate structures of modern times.

</li>

<li>

The reduction of forest reserves since the Middle Ages, however, made

timber increasingly expensive. This has brought about greater reliance by

the construction and manufacturing industries, in modern times, on such composite

wood products as plywood, chipboard, and fiberboard. These new materials are stable

and do not swell or shrink as readily as natural lumber. They do not require long

seasoning periods, and they can be waterproofed, fireproofed, and impregnated with

protective chemicals. Plywood is particularly valuable to carpenters because it allows

them to cover broad areas of framing in a short time.

</li>

</ul>

</div>

</div>

</div>

<div class="row3"> <!--This row contains the footer-->

<div class="footer">

&copy; James Malsher

</div>

</div>

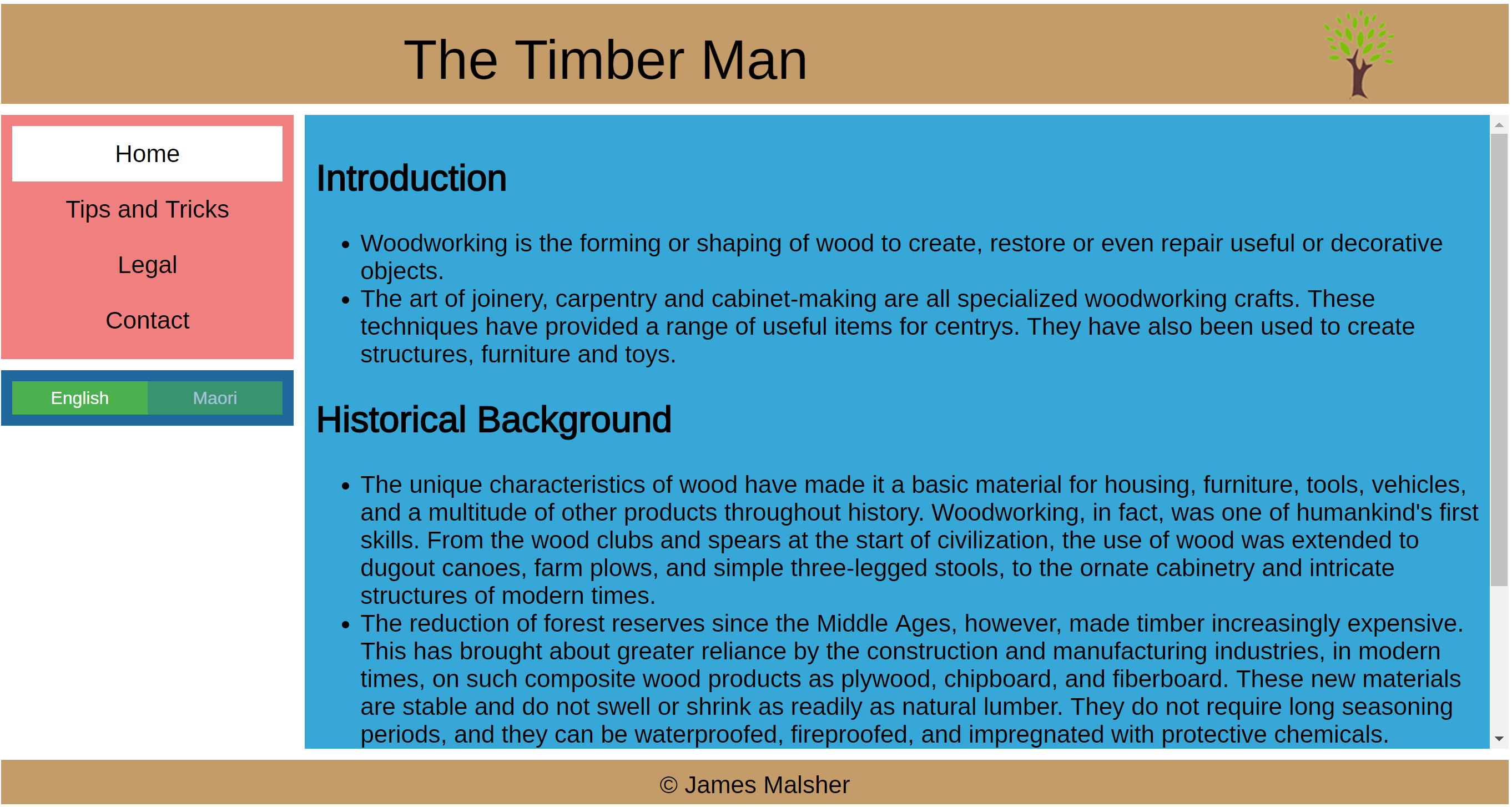
</body>

</html>

I have checked this into Git Hub. Commit code **3e0fd5e.**

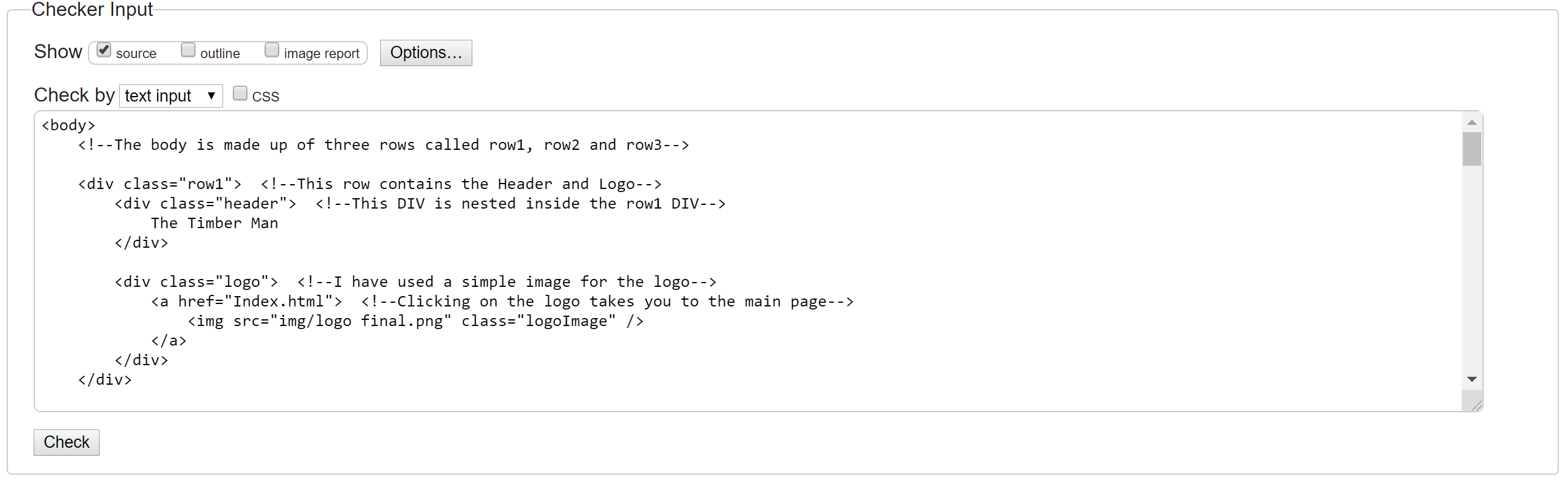
I have now got it to the point where I can hand the assessment in. I have met all the requirements of the assessment that are listed at the start of this document.

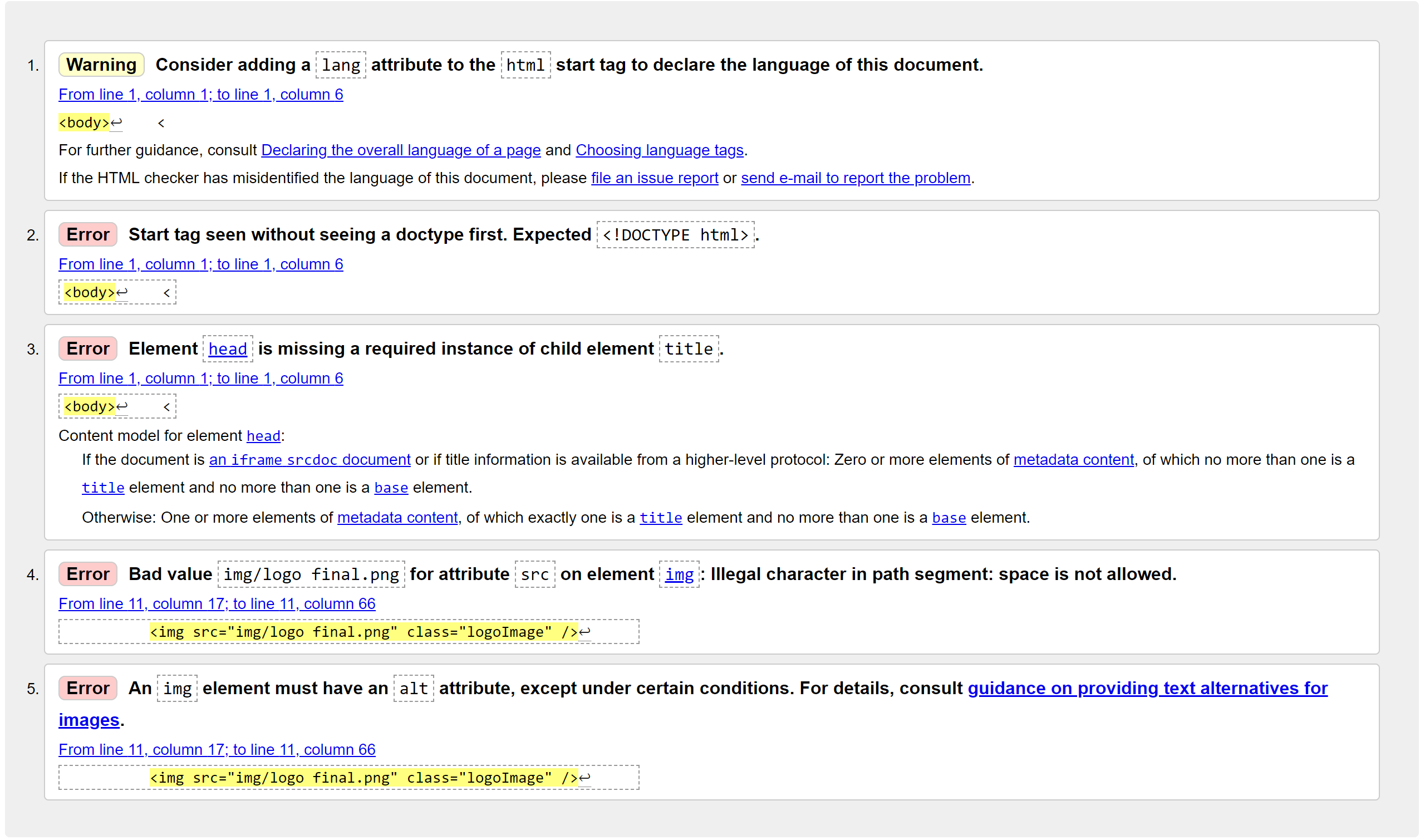
The final commit was **d5f8ae7** and looks like this.



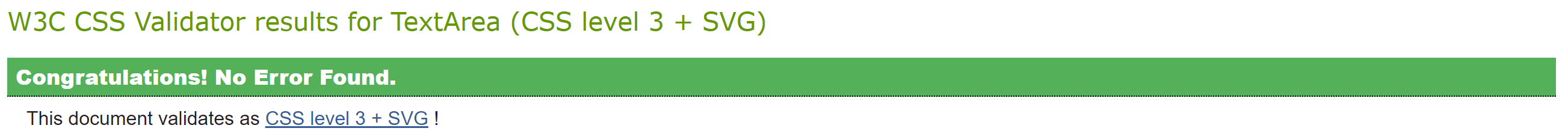
## Validation of code

HTML:



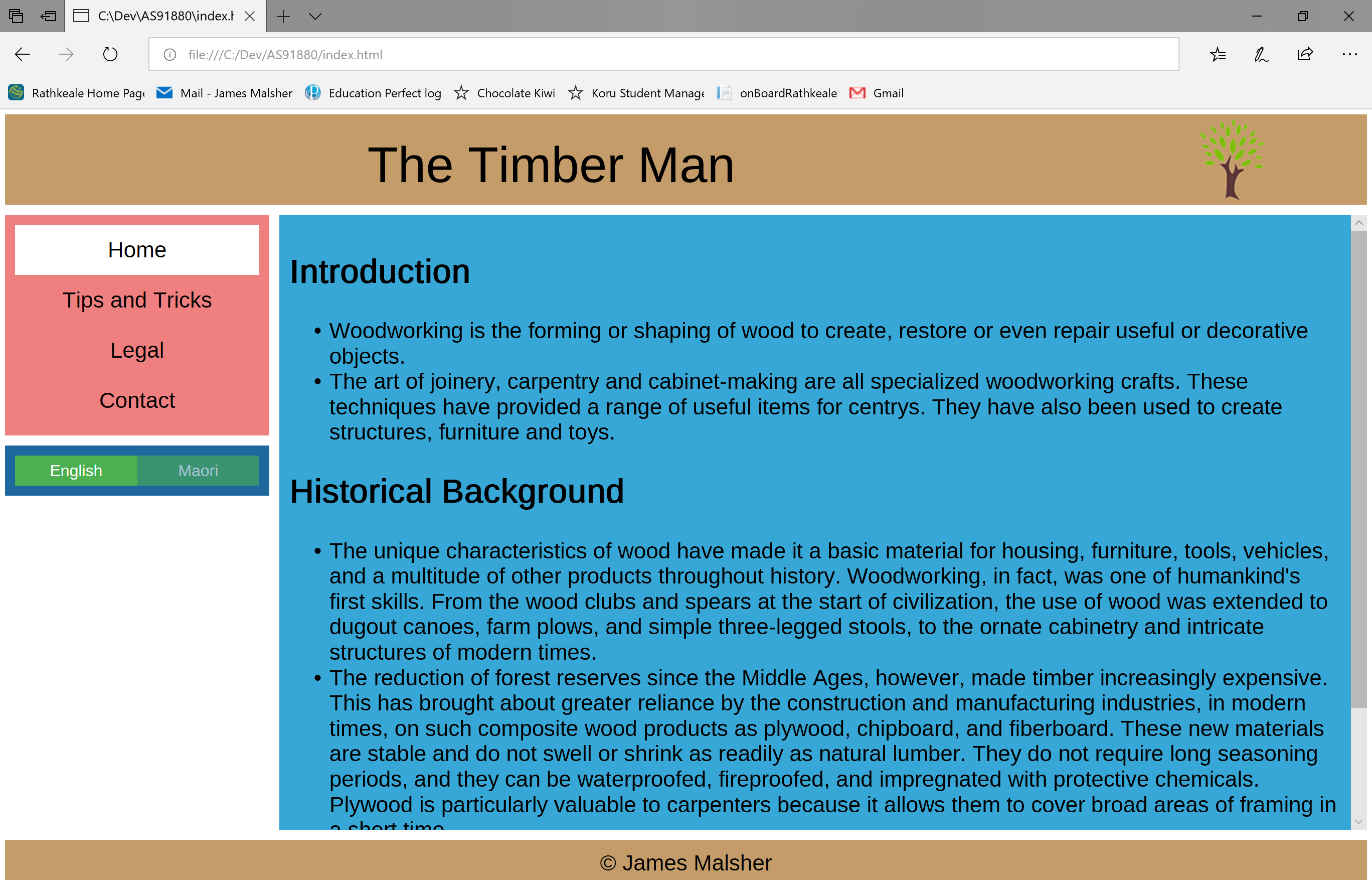


CSS:



# Different Browsers:

## Microsoft Edge:



## Google Chrome:

