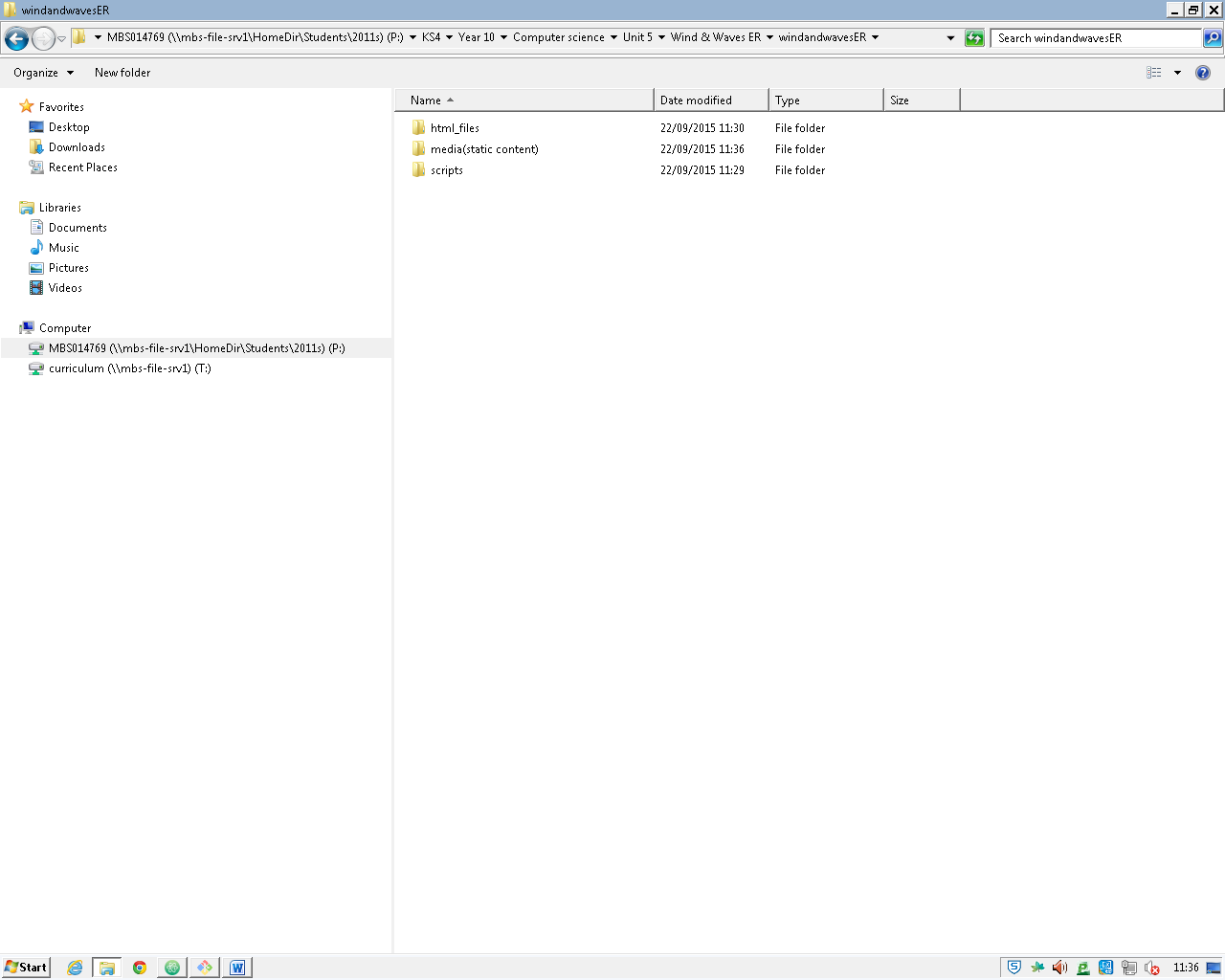
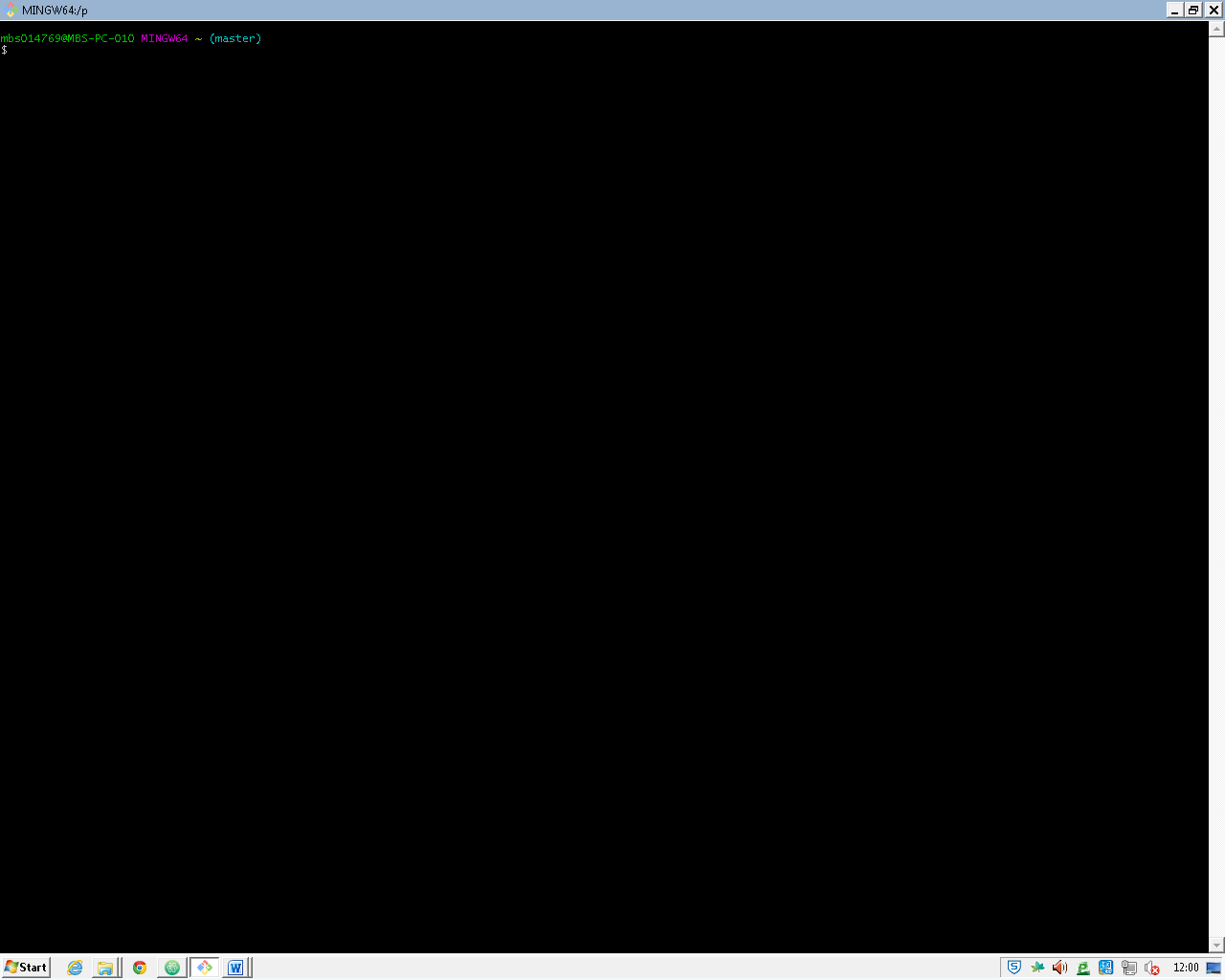
Setting up the wind and wave website



I have created a folder structure so that the websites links do not break when I move the root folder. Moreover I will add this to a git repository – a cloud based source control system this means that if I lose any work at all, or make a change that I did not work I can easily reverse.

I create this repository by firstly by bringing up a terminal program called git bash which will bring up a specialized terminal window for editing git repositories



I now enter in my git credentials(my username & email) via the command

git config --global user.email “[mbs014769@meole.co.uk](mailto:mbs014769@meole.co.uk)”

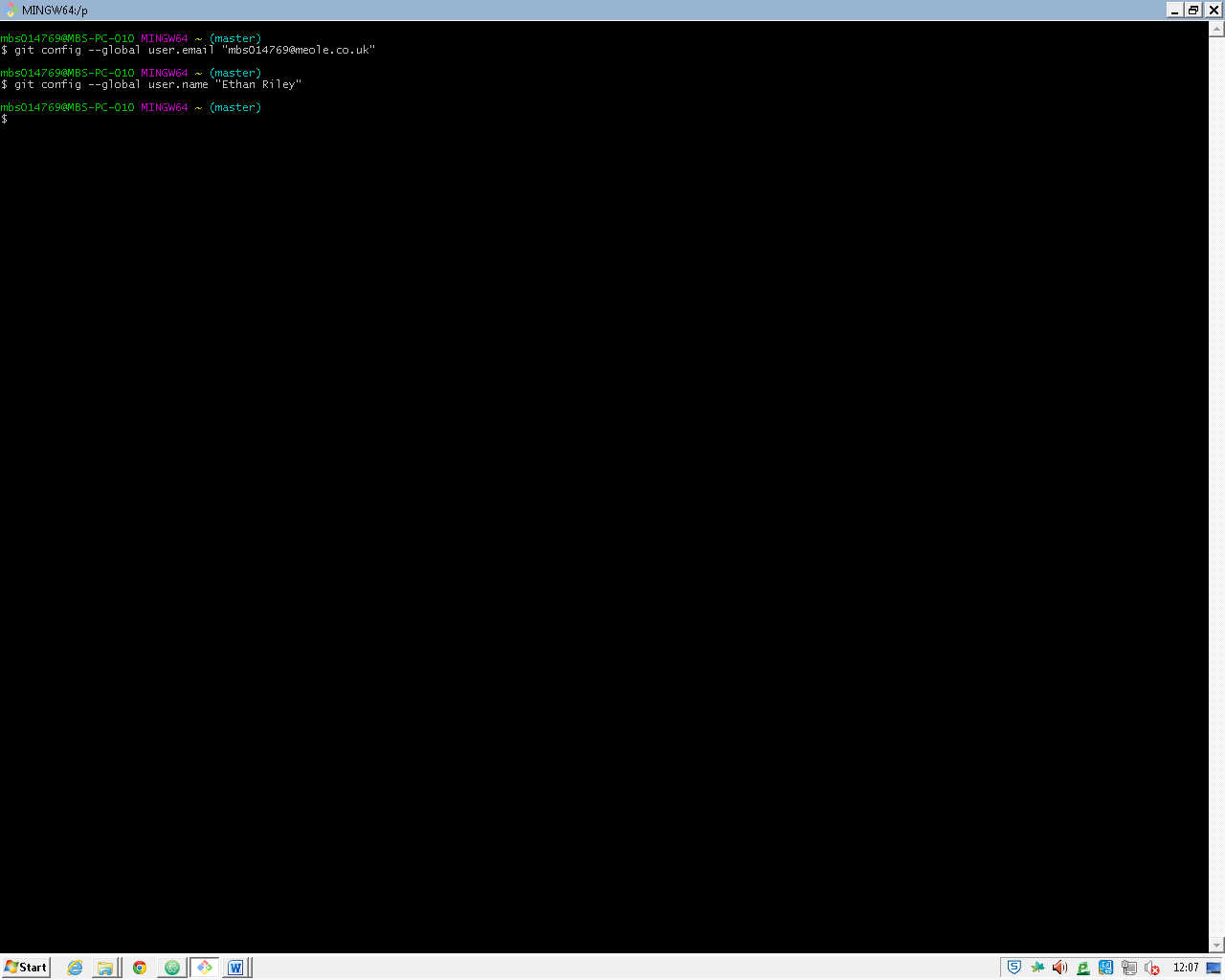
and my user name

git config --global user.name “Ethan Riley”

this command tells git that in all repositories that I create that my username is ‘Ethan Riley’

the config part of theese commands makes

and in the terminal it will look like this :



From Now on red text means that this will be executed in the git bash command window to reduce the amount of interruption in my explanations although I will add a screenshot of these commands at the end of the explanation.

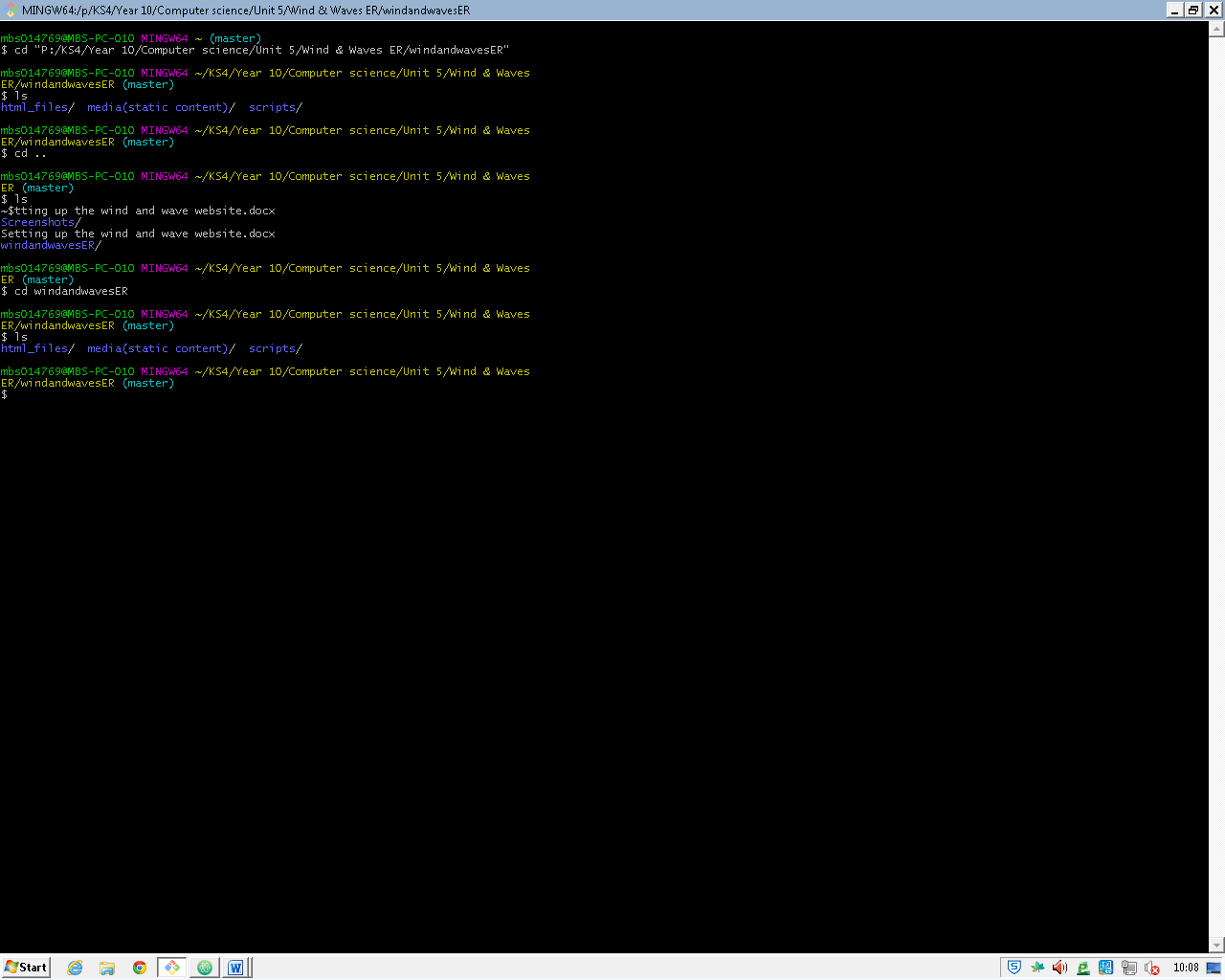
Next I will go to my root folder using the command cd(change directory)

cd “P:/KS4/Year 10/Computer science/Unit 5/Wind & Waves ER/windandwavesER”

note: you have use backslashes unlike in the normal windows terminal as bash is derived from linux terminals which uses the backslash

I then use ls to list my files and folders so I can make sure that this is the right path

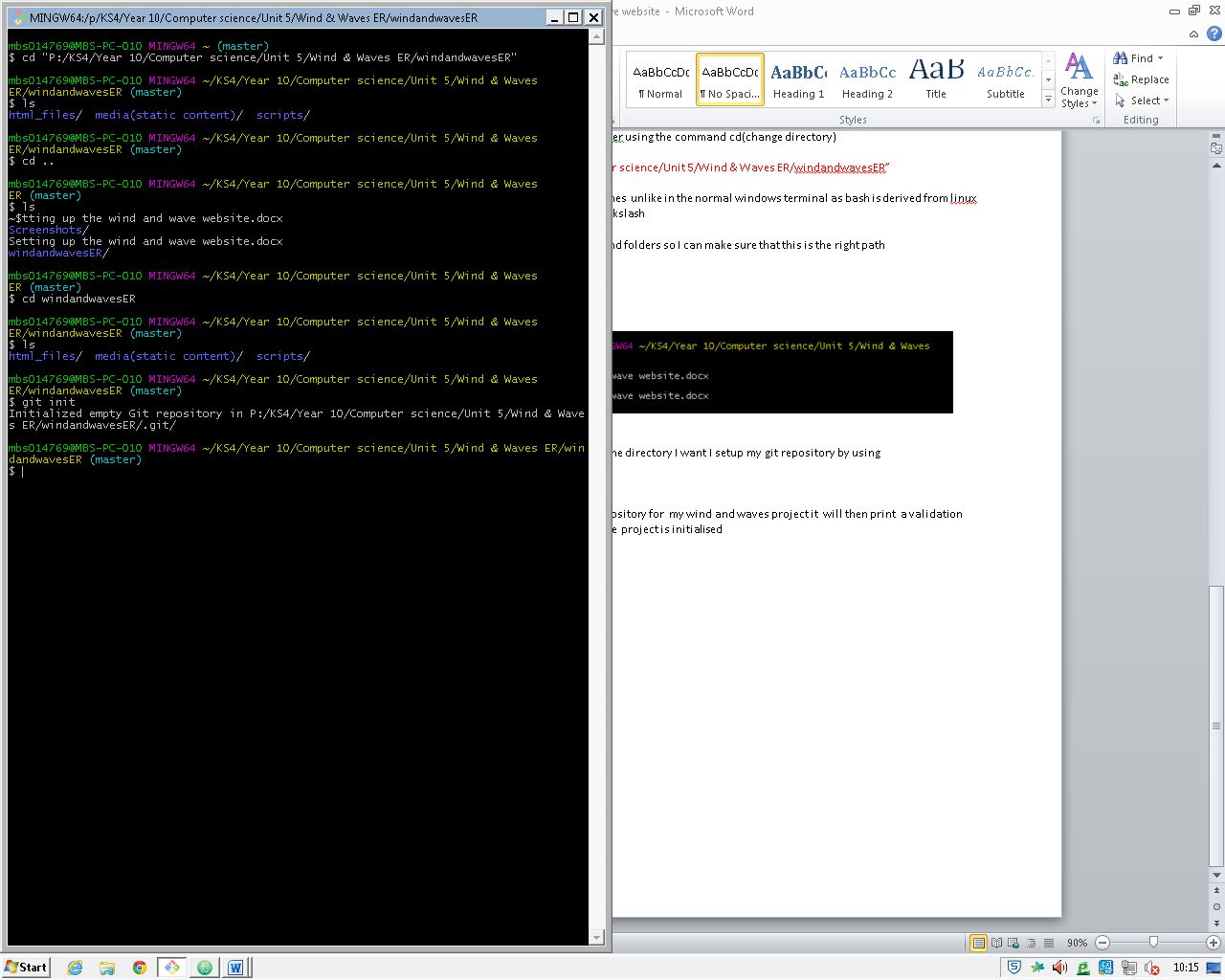
ls

this will output with:

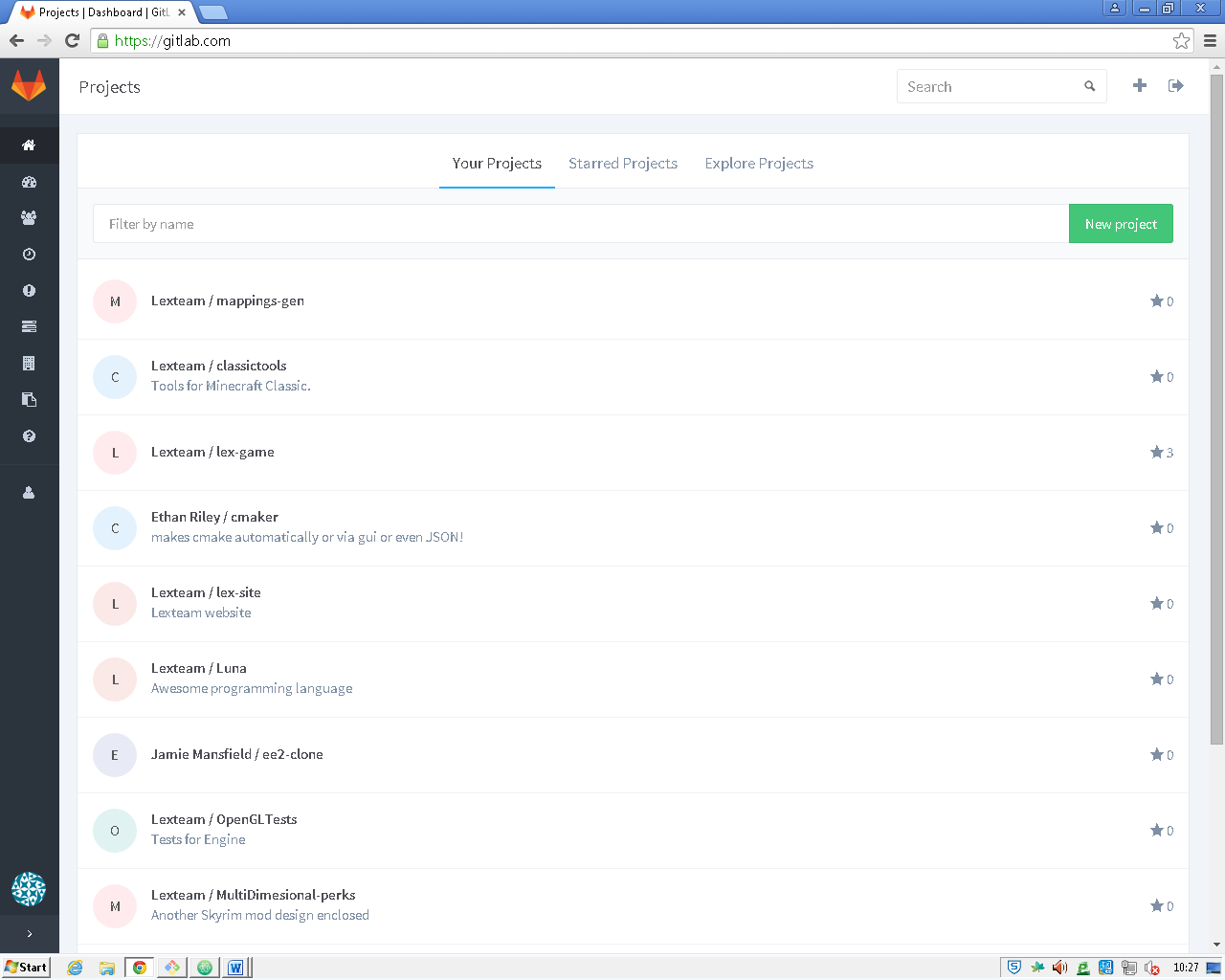
now that I know that this is the directory I want I setup my git repository by using

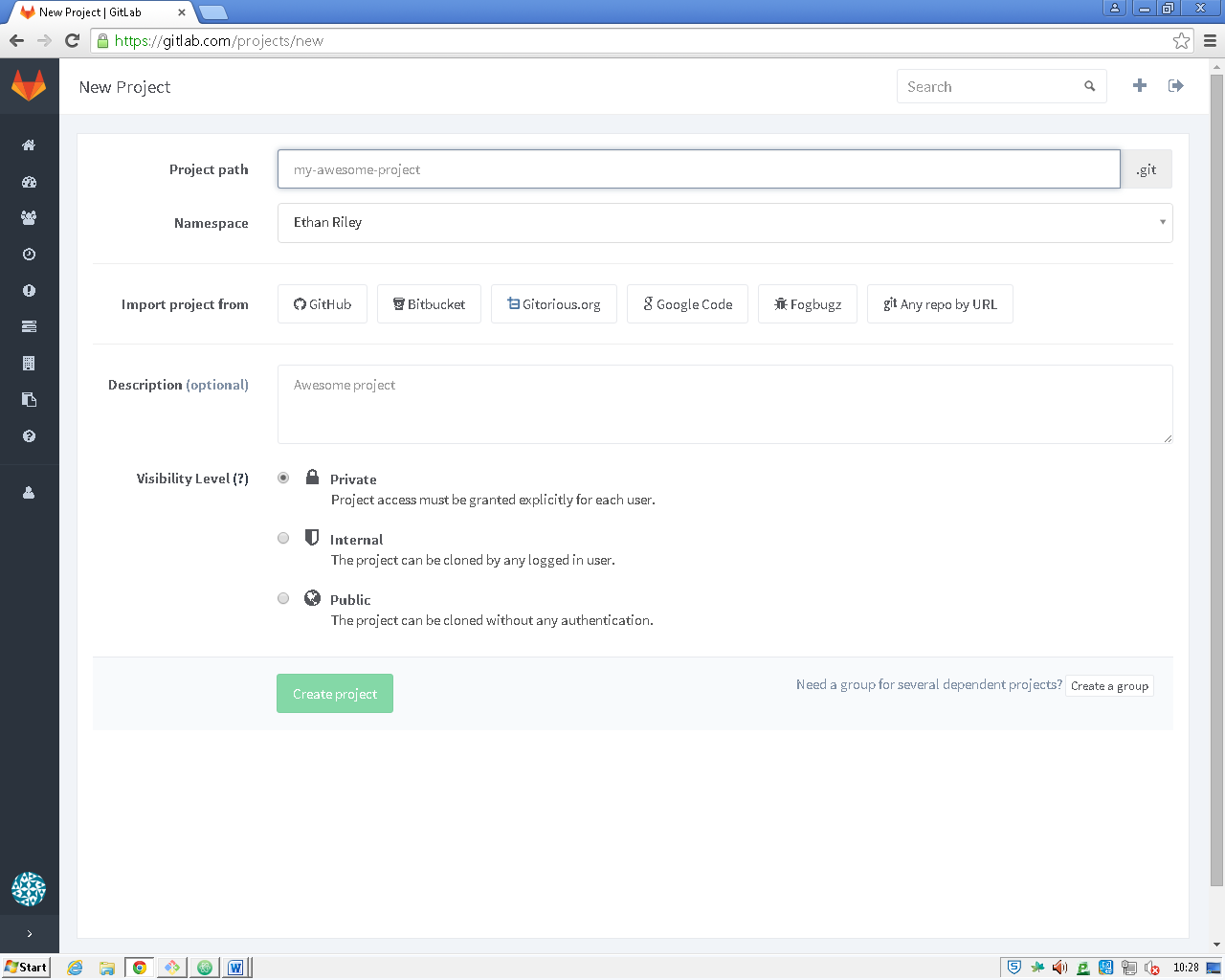
git init

this will create a local git repository for my wind and waves project it will then print a validation message that tells me that the project is initialised

I now add my remote repository which is on the cloud I will use the provider [gitlab](https://about.gitlab.com/).

Firstly I will sign into this provider using my existing account , once signed in I am presented to a window like so:





I click on the ‘new project’ pushbutton and fill in the form below

I keep the project access a private meaning that only I can see the code keeping it safe to other companies’ eyes.

Now the remote repository is ready to use, as the project page is created. I will link it to the local repository by running the command

git remote add master <https://gitlab.com/_UNKNOWN_/windandwavesWebsite.git>

Finally git is setup and I can start my project with safety and comfort of a git repository.

**Github pages**

After some time in development I will add this change git repository to github as it has a project running to provide free serveing and domain names( for files this will help with testing the project on different platforms i.e mobile, the new repository comes with it’s own code editor which I will be using to edit the files directly for bug fixes allowing me to change the website dynamically and give my clients an accurate preview of the site.

**Development Enviroments**

In this website I will be using a multitude of editors which will help me increase my productivity and workflow, although they do not have any effect in the output of the code.

**Jekyll**

I will be using a static generator called jekyll, it increases all aspects production by allowing advanced css preprocessors and templates(that you must create by yourself) that apply to markdown pages which keeps it to the DRY(don’t repeat yourself) standard.

**Jekylls folder structure**

Jekyll has a very strict folder structure which helps us set out our work neatly and in an organized manner reducing the chance of confusion and missing files/links while this is modifiable the benefits of this is massive in a productive environment

**CSS**

I will be using a preprocessor called scss which will compile to css using ruby. This preprocessor will allow me to create advanced stylesheets which can have class inheritance, mixins, more complex media tags, nesting to improve code readability. There are more benefits shown found on its [website](http://sass-lang.com/guide) too.

**Scss Compilation**

The scss compilation is handled by jekyll automatically allowing me to carry on without having to compile the scss every time

Markdown

Markdown is a simplified, lightweight version of html while allowing html tags when necessary it can also have simple notations for tags which act like shorthand for html, any other text that does not have this short hand is automatically put into a <p> tag, this allows a simple and effective method for adding pages which can be understood by non-programmers, allowing anyone with basic understanding of html to write their own pages.

These makes statically generated sites almost as powerful as dynamic sites but are infinitely cheaper and are simpler and quicker to make to a high quality standard.

**Template prototype**

Firstly I will use a templating system called Jekyll programmed in ruby which generates html files from a skeleton template and the required text. This can be done in realtime or just created statically whichever is the most convienent it is also the middle ground of an dynamic webpage and a static webpage allowing me to have the best of both worlds.

**html Includes**

Jekyll uses includes to add data that is used throughout the website, this can hold all the necessary tags to setup a html document, stopping us repeating ourselves and saving valuable time in the process.

<<screenshot>>

This is the include for the head tag which is used in all html files, it allows me to define outsourced files like jquery into my project and use partiton my stylesheets and javascript into seprate files, making things less cluttered and confusing here I load a font, my main stylesheet, jquery a requiement for my javascript file which I include on the next line.

Please see my glossary for the explaination of theese tags and what they do.

If I goto a more intresting tag like <title> you will notice that this is not vaild html, this is because it is not html but the liquid templating langauge which will eventually compile into html, the {% tells jekyll that this Is a start of the templating and %} is the end, like html’s <> and </> notations.

After this I need whether page.title exists this is done by checking that it is not nothing(null). Denoted by the not equal to operator- != , if the page exists we go onto the block of code inserting the page.title value, if not we goto the else block and insert the site.title value.

**Layouts**

In jekyll templates are seen as templates that the markdown/other pages conform to. Theese layouts have commen elements that all the pages include sppeding up development and decluttering pages.

<<screenshot>>

Above is my current default template at this time. As you can see it includes a the head include which I talked about in the last section this is done by {% include head.html %} which is interpreted by the templating engine (Jekyll) to include head.html code in the compiled file of this.

You can see also in this layout there is a for template tag(denoted by the {% %} tags).

This tag iterates though the value after in and puts the iteration into the value before.

For example:

{% for ipage in site.pages %}

<p>{{ipage.title}}</p> 🡨- content

{% endfor %}

Means that it repeats the content in the for loop for every page in the sites pages although the page variable is not used as it will clash with a predefined variable from Jekyll.