Improved Student Issue Reporting and Resolution

Project Report

Ulster University

Supervised by Dewar Finlay

Philip Colman - B00653176

BSc Hons Computer Systems PT

Date: Sunday 22nd April 2018

# Abstract

The inspiration for my project came about as a result of my time as a course representative for my class in University. I found that encouraging people to provide feedback in class was not very effective. We also used Facebook for informing the group of upcoming events like deadlines for assignments. I discovered that the class believed a few issues where not being addressed when it was reported to the studies advisor.

Several of our modules where on website development, therefore I decided that I could use the skills I had learnt and the communication capabilities of the internet to create a solution, that would allow students to easier come forward with any issues and that a process to determine if a solution has been met and only have issues signed off on when that happens.

I wanted broaden the scope of the technologies that I could use while doing this to determine if there was a better way to create the website. I discovered Ruby on Rails. It is a framework using the Ruby Language that handles a lot of the integration of the website allowing for simplification and greater scope with gems.

I however had no experience with the language and would have to perform a broad review of it in comparison to technologies I had already used to get an operational understanding of the framework.

There is little that is truly unique about the website, most if not all of the features can be found elsewhere but the aim is to package them into a product that can serve specific audience in this case students but it could be repurposed for an internal communications system in a working environment or simply a group of friends.

I followed the waterfall lifecycle model for the development of the project as it closely matched the order that the task would be completed in.

# Acknowledgements

I will take this opportunity to thank everyone that has encouraged and supported and helped me to not only complete my degree but to have the opportunity to do so.

I will start by thanking Dewar Finlay for being my supervisor and for providing me with guidance and encouragement towards completing the reports and the project on time.

Thank you to all the lectures I have had throughout my time at Ulster University, for providing me with the education for skills required to complete my degree.

I would like to thank Sarah, Christopher and James, for the support they give me throughout all 5 years.

I would like to thank my family for being

# Table of Contents

Chapter 1: Introduction…………………………………………………....1

* 1. Background……………………………………………………………1
  2. Project Aim…………………………………………………………….2
  3. Project Objectives…………………………………………………….2
  4. Outline Project dissertation structure……………………………….3

Chapter 2………………………………………………………………………….

Listing (title and pagination) of Figures presented.

Listing (title and pagination) of Tables presented.

# Chapter 1 Introduction

## 1.1 Background

The project came about as a result of a desire to create a better way for students to report issues to the course representative. It was determined that there was a using what was learnt throughout my course to create a dedicated platform for students to communicate and report issues that they are having.

At present when there is an issue there is the conventional methods of reporting or resolving it.

* Talking to the professor, another student in class, the course representative, studies advice or going directly to the course director
* Email the professor, course representative, studies advisor or course director
* Use Facebook group to ask the question and see if anyone gets back to you.

The problem with talking and email is that allot of students will only report something if it is really becoming a problem so you only get serious issues through this method. The Facebook group is a lot less formal and therefore is more approachable, however some students do not use Facebook and miss out on some things and not everyone actively participates in the group, a messenger group might be better but most of the class broke up into smaller groups with their friends making class wide communication through this means limited.

Then there is the Staff Student Consultation Committee (SSCC) form that it would be productive to collect problems or concerns that students may have. I find that asking in class about concerns or problems tends not to get much feedback, whether it is not being able to come up with anything at that moment that had been bothering them or not believing that the issues or concerns may not be important enough to raise. At which point the student representative will normally provide them with a time and date to provide any feedback and different means of getting the feedback to them.

## 1.2 Project Aim

The aim of the project is to create an all in one means of communication for the students to communicate, and report issues and concerns as they come up with them throughout the year so that the course representative and other members of the class may address them and when the time comes to complete a SSCC they can be compiled into a report with their title and description.

The student Representative could then edit the result to only include issues needing resolved and remove internal questions not related to the course or work. The students will be in a group with their class so they should all have the means to communicate with each other.

## 1.3 Project Objectives

1. A Question system should exist where students should be able to ask a question, and other students should be able to respond. The course representative will then be able to decide on the further action require for the Question
2. A one to one with the course representative, text or voice in app or on the web page with email available also.
3. Some way to communicate with the course tutors on the website or in app if necessary.
4. Some way to communicate with the course co-ordinator on the website or in app if necessary.
5. Application is secondary objective

Steps to complete objectives (needs update\*\*)

1. Define exact features for the webpage/app
2. Research programming language (PHP vs ruby)
3. Create database for users
4. Create basic interface
5. Basic messaging system
6. Research web sockets for live chat functionality
7. Storyboard website design
8. Research Twilio or alternative integration
9. Fully documented testing process

## 1.4 Outline Project dissertation structure

The following is an Overview of the Chapters that make up this report

Chapter 1 – This chapter provides the background objectives and aims of the project

Chapter 2 – An explanation of the technologies that will be used and a comparison to other technologies that could have been used as well as a justification for using the technologies that were used.

Chapter 3

# Chapter 2 Literature Review of Ruby on Rails

## 2.1 Introduction

This chapter is dedicated to the understanding of the Ruby on Rails frame work. What it is, how it works and how it compares to languages and servers I have made in the past. It should give an understanding of where to start when developing a rails server.

## 2.2 What is Ruby on Rails?

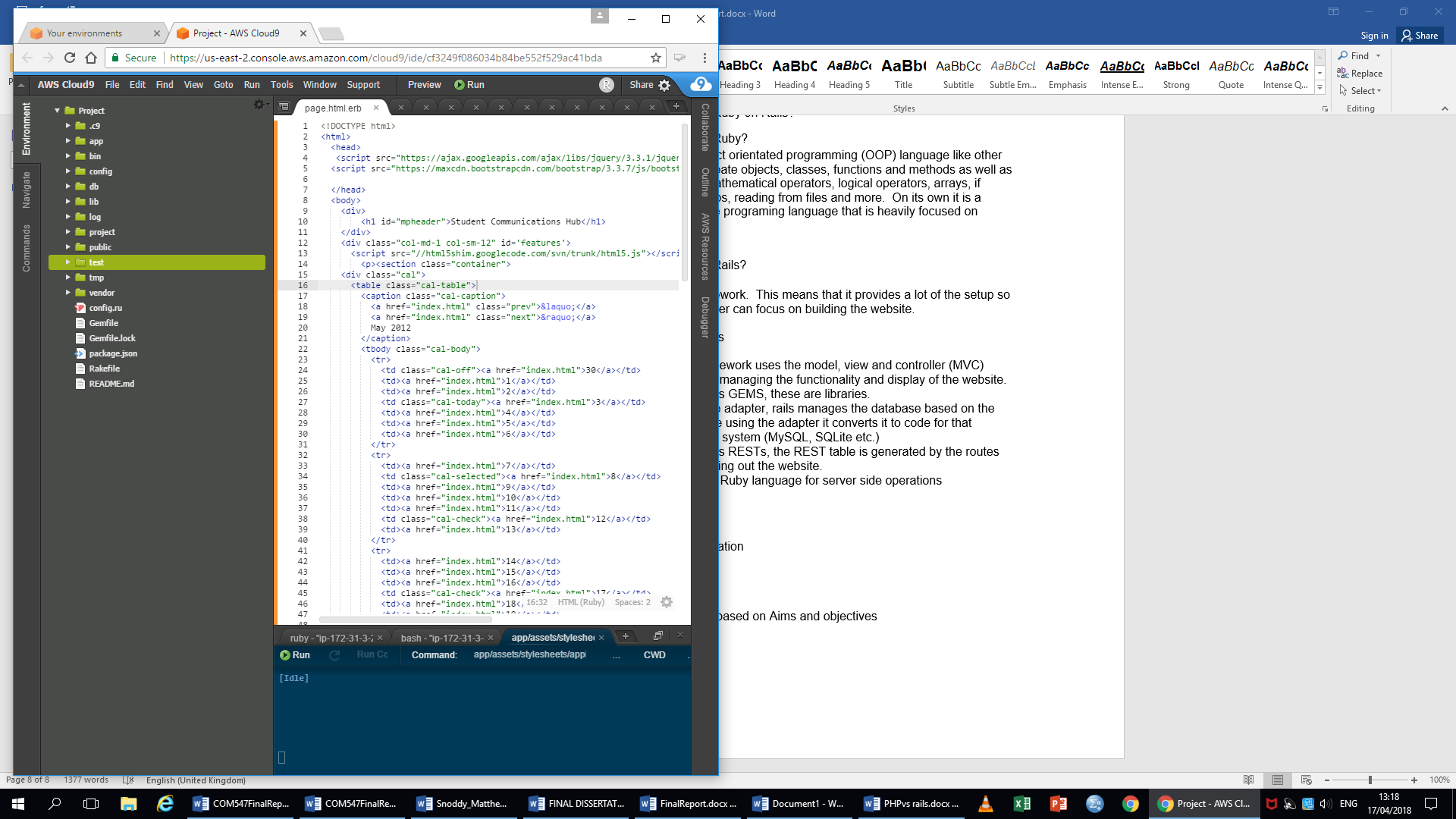
### 2.2.1 What is Ruby?

Ruby is an object orientated programming (OOP) language like other OOP’s it can create objects, classes, functions and methods as well as run code like mathematical operators, logical operators, arrays, if statements, loops, reading from files and more. On its own it is a general purpose programing language that is heavily focused on objects.

### 2.2.2 What is Rails?

Rails is a Framework. This means that it provides a lot of the setup so that the developer can focus on building the website.

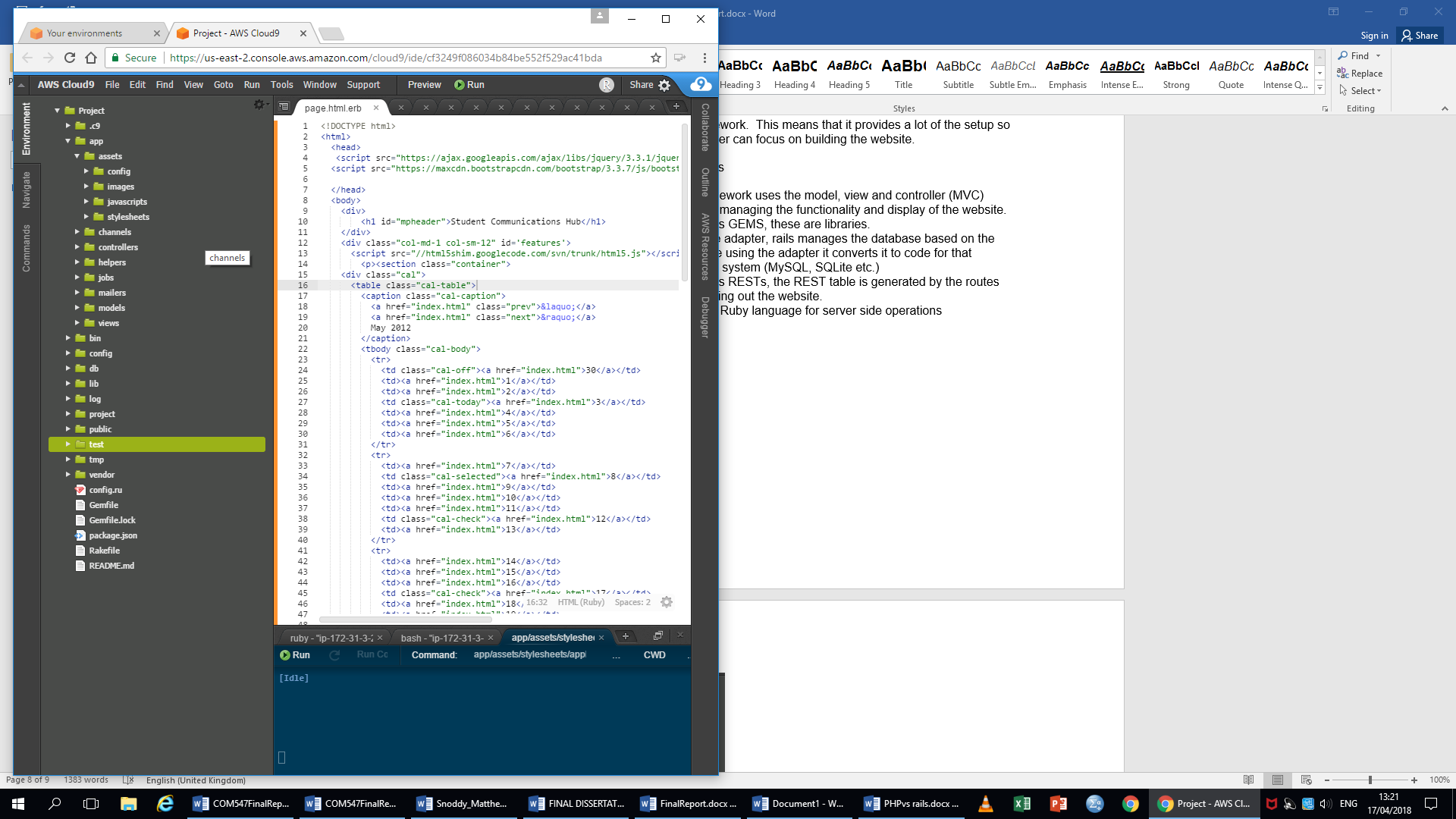
Rails directories



It sets up the directories, the directories will be able to see each other when in here.

(Image taken from my own environment)

The application

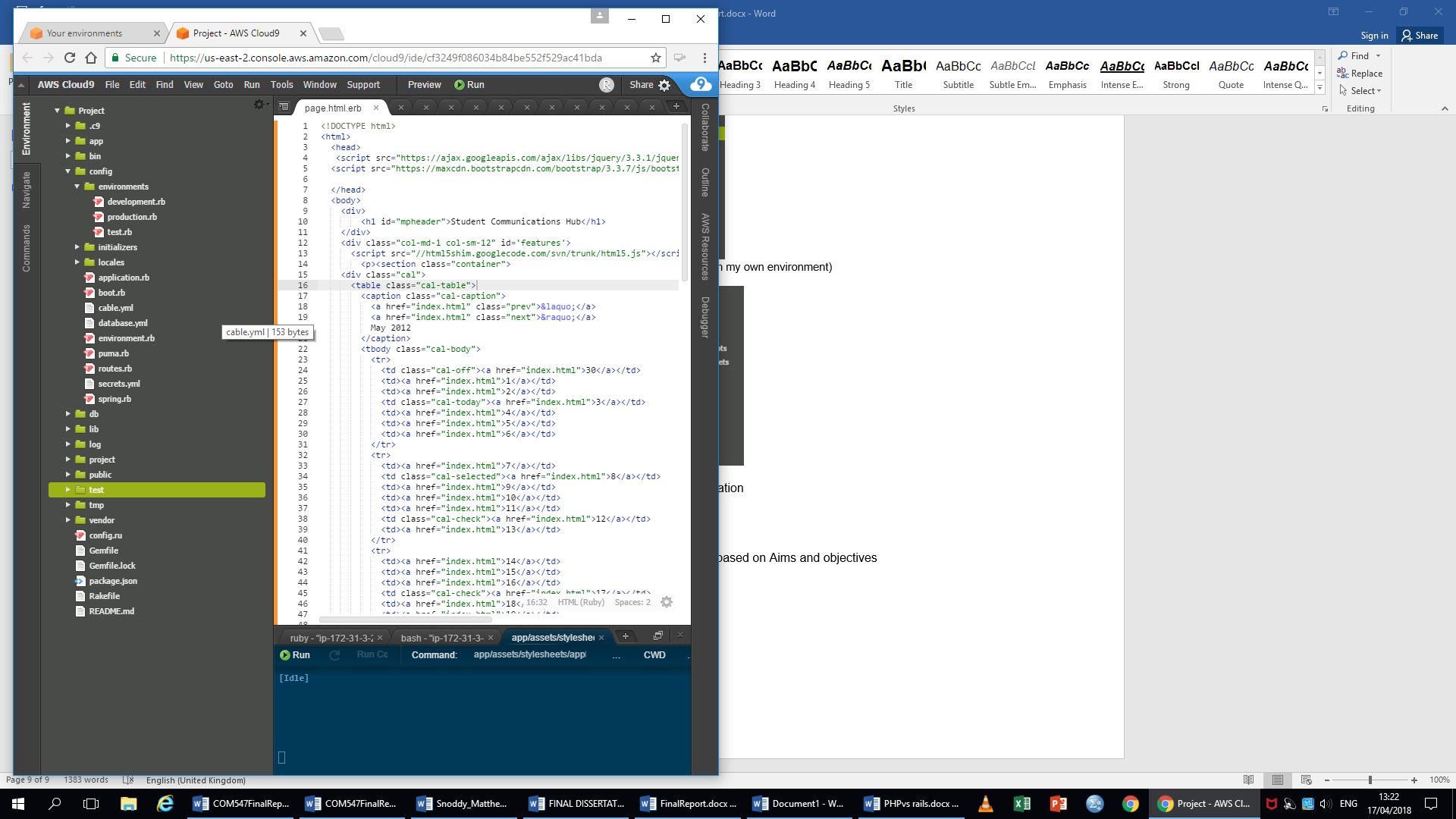


The app directory is where all the files and assets to do with the website are located. This is where the developer will develop the functionality and design of the website.

(Image taken from my own environment)

This framework uses the model, view and controller (MVC) setup for managing the functionality and display of the website. Views contain the HTML files that give the webpage its layout or forms, models are where most of the code logic based code will go, controllers are used for navigating and deciding what views will be displayed. They can each do some of each other’s work, the controller for example can output HTML code, and the view can run ruby, JSon, jQuery and other languages for logically operations. It is recommended however that you keep the code in each for the purpose they are designed for because it is tidier and easier to find problems that may occur.

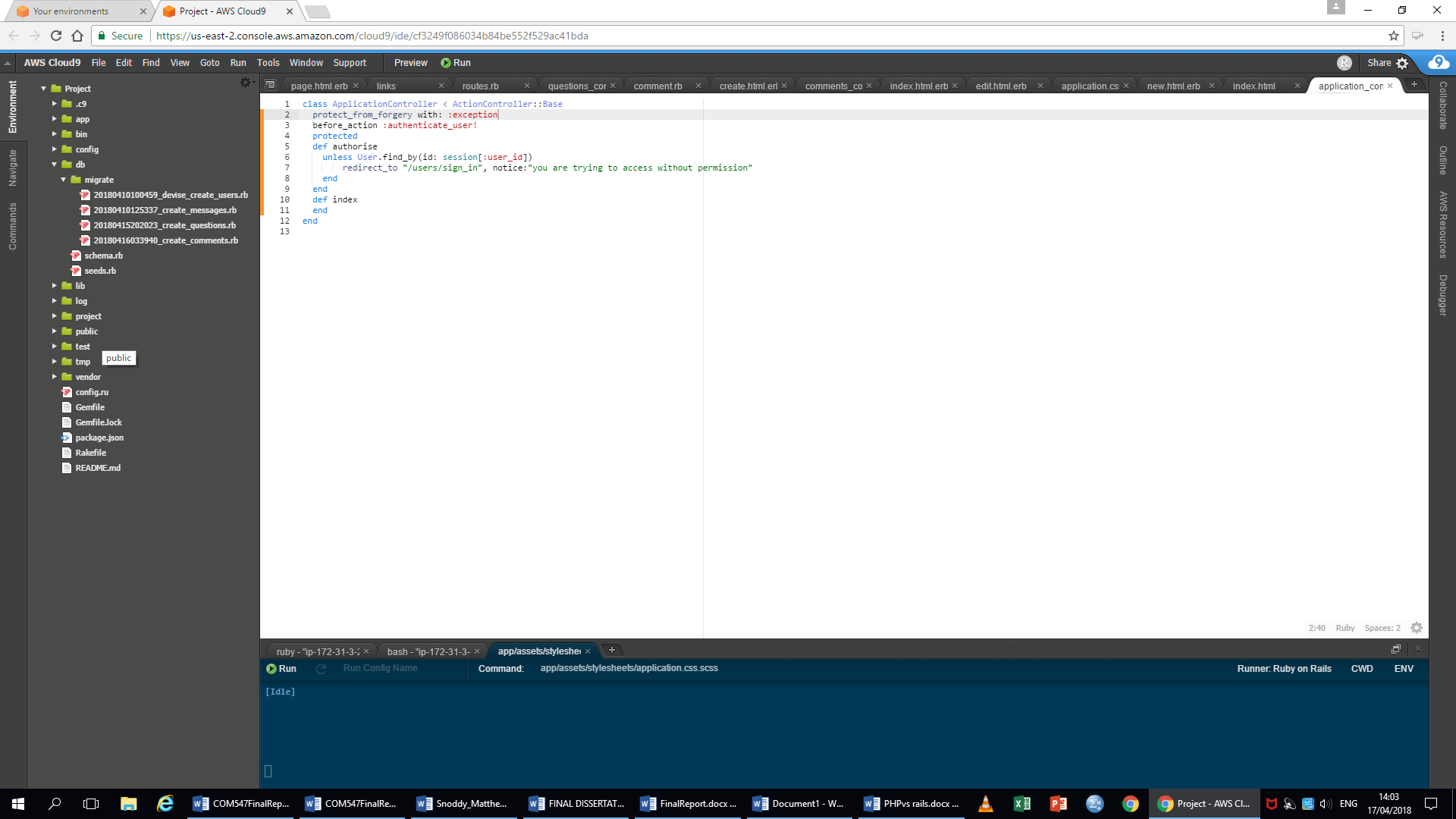
The configurations files

In the config directory at the project level there is files related to how the environment is setup. They all do important things but the ones that will be edited for the project are the cable.yml this is related to the action Cable used for chat functionality on website, the database.yml this is related to the database settings this is where the rails server can communicate with your databases and the routes.rb creates the map of the website all the address of the webpages, the cable is also mounted in here.

(Image taken from my own environment)

The database adapter in rails manages the database based on the ruby code using the adapter it converts it to code for that database system (MySQL, SQLite etc.). This means that the user does not have to work directly with the database.

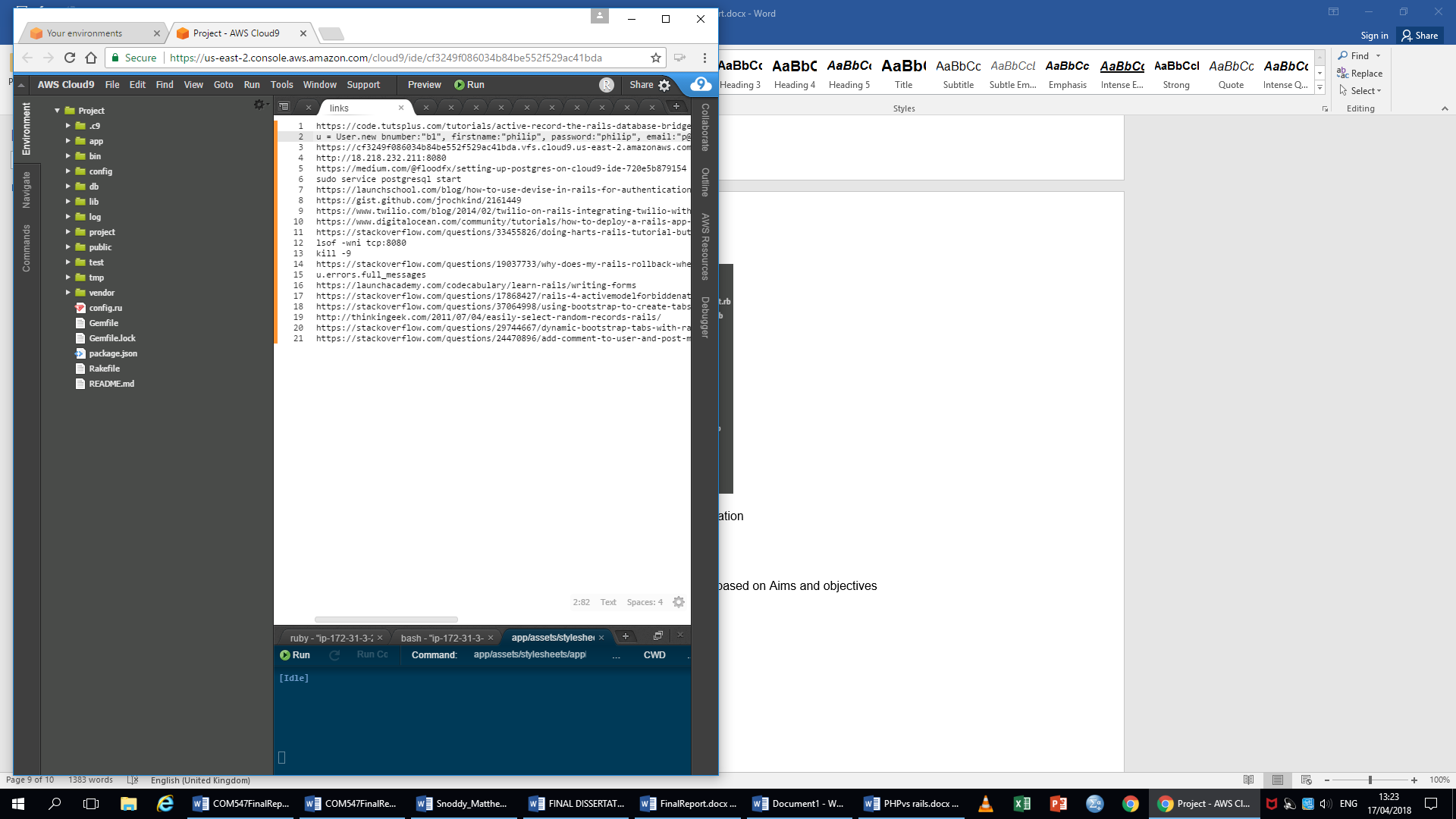
The database seeds

The Database seed files

The db directory is where the code for creating database is stored. When rake db:migrate is ran from command line the database(s) will be created.

(Image taken from my own environment)

Rails server files



At the bottom of the environment are the files that tell the rails server how to run the only file that a developer would change directly would be the Gemfile, if that want to add more libraries to work with.

(Image taken from my own environment)

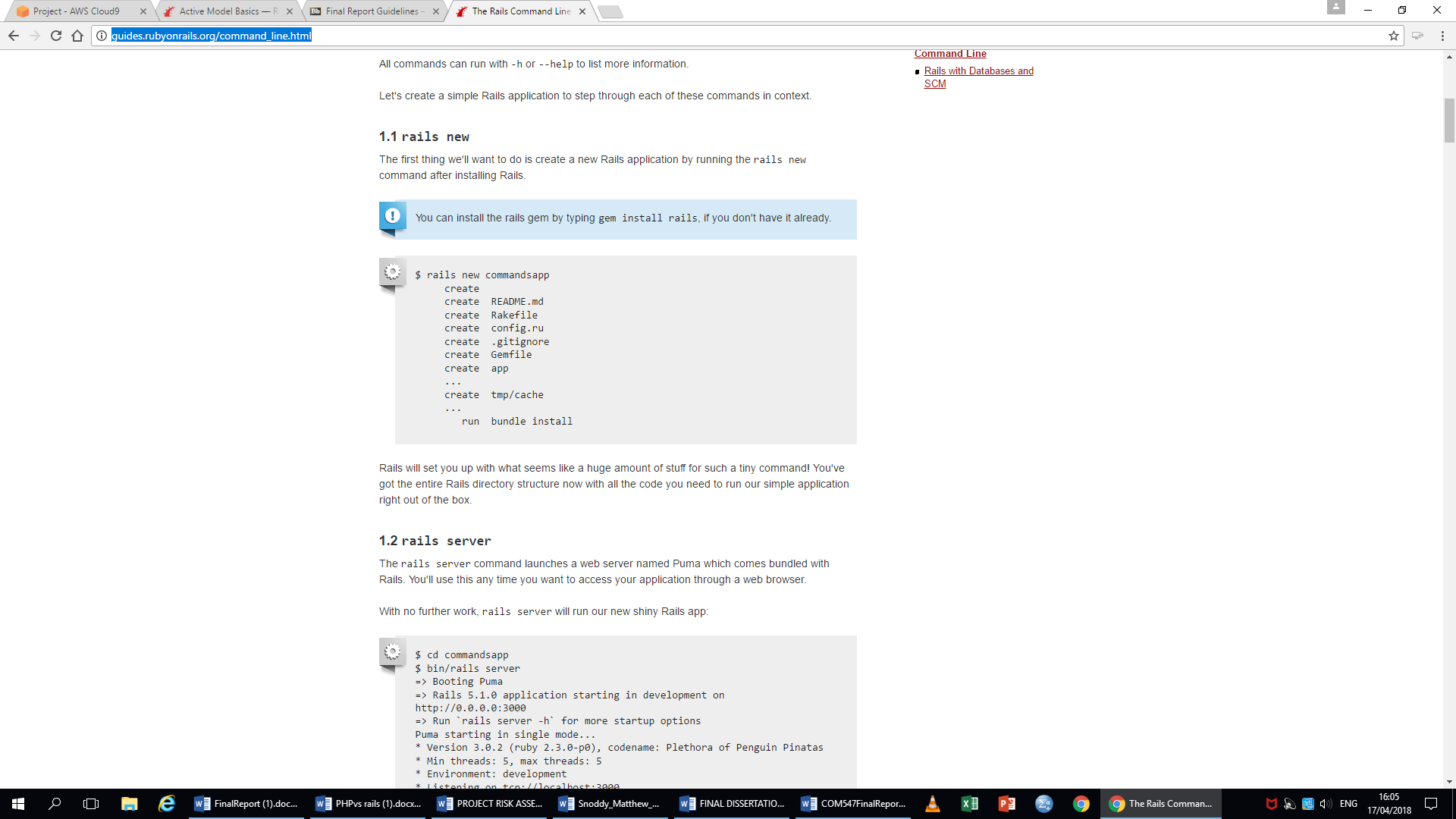
## 2.3 How Ruby on Rails works

### 2.3.1 How to set up a server

First you need to download and install Ruby on Rails. You can get the download from this link: <https://rubyinstaller.org/downloads/>

When it is downloaded open a command line interface that comes with your computer and run the command “gem install rails”. This will install the version of rails you downloaded.

Ruby on Rails creates all the files and directories and Rails files for the user when they run the command “rails new your\_app\_name”. Your\_app\_name being the name you choose for you rails app.

Creates all the directories and rails files

<http://guides.rubyonrails.org/command_line.html>

At this point you can run your Rails server by using the command “Rails server” and going to local host to see it.

### 2.3.2 Getting started with development

Ruby on rails provides templates for starting development of your website.

These can be created by running the commands “rails generate”. So we can generate the controller for a web page by using the command “rails generate controller controllerName”. This will create associated helpers, JavaScript’s, stylesheets and the views that you declare after the controllerName, “rails generate controller controllerName name”.

Another “rails generate” you can run is “rails generate model modelName”, this will generate the model, an .yml file and the file that tells rails what the database will be like. The database generates with a few default values like an Id, a timestamp and an updatetimestamp. If the developer wants to add fields to the database they could go in and edit it to include these fields or when running the generate command for the model he can include the fields, “rails generate modelName name: string”.

You will want to setup the database in the database.yml file, you tell the adapter what database software you are using and the username and password to access the database. To get the database to be created the command “rake db:migrate” is used.

There are many more features that Ruby on Rails has that will be required for creating a website, but what is explained above has built up all the files and directories that will be needed to start a basic website. All of which is done with a few lines in a command line interface.

## 2.4 How Ruby on Rails compares to PHP and a WAMP server

### 2.4.1 Setup

In chapter 2.3 above the setup of a Ruby on Rails server was explained. To make a website the with a PHP and WAMP setup would require the developer to create all of the files and folders that where generated themselves which they can do in Ruby on Rails as well but the website created in Ruby on Rails is already integrated so that a it should be able to access the other directories without further setup like the line of code “<link rel="stylesheet" href="/html/styles.css">”, in rails the stylesheet for that page is already setup and does not need linked to. Leaving the developer to decide on how the directory is setup will depend on the experience of the developer in organising there directories and tidying there code.

### 2.4.2 Security

In Ruby on Rails a tag <Cross-Site Request Forgery> will prevent a hacker from attacking an active user and changing their data or unknown to the user redirecting them to another site, where as in PHP you will need to create a logical check for the session. That could would look like this:

if (!isset($\_SESSION['token'])) {

$token = md5(uniqid(rand(), TRUE));

$\_SESSION['token'] = $token;

$\_SESSION['token\_time'] = time();

}

else

{

$token = $\_SESSION['token'];

}

<https://stackoverflow.com/questions/6287903/how-to-properly-add-csrf-token-using-php>

Ruby on Rails requires strong parameters in the newer versions for passed from a table. This means you have to define what variable is being passed and whitelist the input, so only this information is collected from the database. This makes it harder to post data that should not be posted.

### 2.4.3 Database setup

To create a database with PHP and WAMP would require the developer to run the database server themselves like MySqlAdmin. They would then use the interface on the language to create the database, as discussed in chapter 2.3 above in Ruby on Rails this is all handled by the Rails framework following the Ruby code.

### 2.4.4 Running the server

A Ruby on Rails is a server itself and will run itself. It comes with the Gem Puma for testing the code, this can be changed to another Gem like Passenger which will run the code faster for a released version of the code. PHP requires a server to run it like WAMP in this comparison. WAMP is the Windows version, there is also MAMP the Mac version and XAMP the cross platform version.

There are other services that can run local host as well, many of them are online and will host your website. If the website is going public having it hosted online is a good idea rather than running it on a computer all the time hosting it on a server of a large company like Amazon will greatly increase the likelihood that you website will remain online as they will have the resource for backing up their services.

## 2.5 Conclusion based on Aims and objectives

Ruby on Rails has many features that make putting together a website simpler by generating templates that will help direct me towards the purpose of my project. All the setup can be done with fewer lines of code as well as the creation of the database. The basics, HTML, CSS, JavaScript, JSon, JQuery, MediaQuery and others have not changed but PHP has been dropped for Ruby a more powerful language that can do much of what the other languages can do as well with a lot less code. I believe that the Ruby on Rails frame work will have a more positive impact on my website.

# Chapter 3 Literature review of other technologies used

3.1 Introduction

DNS elastic IP

Ports, HTTP:80, HTTPSecurity:443- automatic SSL, Telnet:23

TCP/IP?do I mention

Databases, MySql, SQLite and postgreSql

Routing, controller-view, get

Model-view-controller

HTML, CSS, SASS, JavaScript, JSon, JQuery, MediaQuery, AJAX, bootstrap, coffee,

<http://guides.rubyonrails.org/getting_started.html>

<http://guides.rubyonrails.org/command_line.html>

<https://rubyinstaller.org/downloads/>

<https://en.wikibooks.org/wiki/Ruby_on_Rails/Built-In_Rails_Tools/Generators>

<http://php.net/manual/en/intro-whatis.php>

<https://www.ruby-lang.org/en/about/>

<https://www.w3schools.com/html/html_css.asp>

<http://www.xyzpub.com/en/ruby-on-rails/3.2/scaffold_anlegen.html>

<http://www.cgisecurity.com/csrf-faq.html>

<https://stackoverflow.com/questions/6287903/how-to-properly-add-csrf-token-using-php>

<https://stackoverflow.com/questions/3009023/how-to-sanitize-sql-fragment-in-rails/35608331>

<http://guides.rubyonrails.org/command_line.html>