

StudentMania

Discord for students

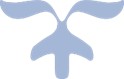
by

Jack Haugh G00359747

Stephen Gallagher G00314689

Kevin Flanagan G00228079

Mark Campbell Supervisor



May 5, 2021

Final year project

Department of Computer Science and Applied Physics

Galway-Mayo Institute of Technology (GMIT)

Table of Contents

[StudentMania 3](#_Toc70431141)

[1. Introduction 3](#_Toc70431142)

[Communication Tools 3](#_Toc70431143)

[GitHub Repo 3](#_Toc70431144)

[Ideas 3](#_Toc70431145)

[Motivation of our Project 4](#_Toc70431146)

[Literary Review on Communication platforms 4](#_Toc70431147)

[Abstract 4](#_Toc70431148)

[Introduction 4](#_Toc70431149)

[BODY 5](#_Toc70431150)

[Conclusion 6](#_Toc70431151)

[Methodology 7](#_Toc70431152)

[Research 7](#_Toc70431153)

[Frontend 7](#_Toc70431154)

[Linking aws amplify with custom domain – 9](#_Toc70431155)

[Amplify Table 10](#_Toc70431156)

[Font awesome Website: 11](#_Toc70431157)

[Linktree website: 11](#_Toc70431158)

[Linktree Table 11](#_Toc70431159)

[Publish.twitter.com - website: 12](#_Toc70431161)

[Technology Review 14](#_Toc70431163)

[Key Features 15](#_Toc70431164)

[Survey 15](#_Toc70431165)

[Databases 18](#_Toc70431166)

[Database/elastic beanstalk issues 19](#_Toc70431167)

[Week 11 & 12 25](#_Toc70431168)

[Conclusion 26](#_Toc70431169)

[Reference 27](#_Toc70431170)

[Bibliography 27](#_Toc70431171)

StudentMania – Communication is the key right now. With the state of the world and what’s happening right now it is more important than ever for students that are working on their education to be able to communicate with one another and with lectures or staff. Taken this into account we decided to create a website where all the students would be able to share notes and be able to communicate and ask questions to do with their workloads. When deciding on our project we were told to try and solve a problem, as students this hit close to home for the three of us.

When students are working from home a lot of different websites are needed (Teams, Discord and Moodle) are just to name a few. It can be difficult for non-computer students to keep track of all these sites. Trying to solve this problem for students and try to help them organize it into one website that could cover all of a student’s needs without them having to navigate and deal with too many sites. Our site will let the user login, ask questions, and have access to chat boxes and chat rooms for students to meet and discuss course work in real time.

# Introduction

### Communication Tools

Given the task to join as a group and come up with a project between us, after meeting with our supervisor Mark Campbell we decided to set up a team’s meeting every week to discuss/plan and work on or project for the coming months.

Before any plans or ideas were decided on were created a mind map for what our project would be, for the goals we wanted to reach and for each other’s parts within the project. We set up meeting with our supervisor and weekly meetings for the group also where would work on the project together so we would all understand what we were creating.

### GitHub Repo

Next up was the creation of a git hub repo where we would all be contributors to the project and work on it individually also as needed. We will also add mark to this repo as our supervisor.

GitHub Repo for the project: <https://github.com/GallagherStephen/4thYearProject-Dissertation>

[PUT GRAPHS FROM GITHUB HERE then remove this!!!]

The main factors we decided on for our project, were “what would be some way to help” and “what do people need”. With this in mind we decided to choose a project that would be a benefit to students as we are currently students ourselves and figured it could be an issue that affects students and their way of learning or living as a student.

### Ideas

Some of the ideas were a “delivery services” for food or other items and a “Lift Sharing” for students that don’t live near the college and need to travel, public transport may not be available to the student.

Car apps and rugby apps were some other ideas we had but decided to final build a student learning app that would store timetables, lab secludes, class notes and have a chat services to let the students communicate with their classmates.

### Motivation of our Project

A discord/teams/learn online type website/app would be a great benefit to students and help them keep track of their classes and notes. This app would be designed to be a one stop shop for all a student’s needs online connection with their college and classmates. Communication and friendly interface will be the key to this app/website being useful and successful.

A questionnaire would be something worth thinking about for this project as it would give us some great feedback on what other students outside of the software development course a chance for their voices to be heard when it comes to the design of the app. This app will cater to a wide range of students not only the students that are involved with technically or computer courses.

With the three of us completing this project we will all have to work together as much as possible and communicate on the work that is being completed. With then situation that the word is in now video calls will be our main mode of communication.

Another bit of software we have decided to use was Trello, which is an online managing and note tracking website that all three of us can use to track our progress with designing and building our project. This will let us all work and keep track of our work online that any of the three of us will be able to add tasks or move them to the completed section.

It was our supervisor Mark Campbell that suggested to use Trello as there will be so much going on in the coming weeks that it will be difficult to keep track of all the tasks, Trello allows us to track information and tasking in real time.

## Literary Review on Communication platforms

### Abstract

The goal of this platform was to make communication easier for students. According to our research findings, students tend to use multiple platforms to cover their workload from classes and lectures. When we were building our platform, we wanted to help students eliminate the need to use multiple platforms and reduce that need to a single platform that could meet all of a student's needs.

Our main finding from our research was that students required multiple platforms to complete their daily workloads. If a student could complete all of these tasks on a single platform, it would be easier, less stressful, and save time overall.

### Introduction

Adapting to new online learning methods is one of the most significant changes for students during this pandemic. In this day and age, communication is the most important factor in education. Online learning and communication between students and lecturers are at the forefront of planning and discussion, thanks to platforms such as **Microsoft Teams, Learn Online,** and **Discord,** to name a few.

Even in these difficult times, students must be able to continue their education. People can stay in touch and continue their education thanks to technologies that connect to the internet. Students are currently communicating with one another through the platforms mentioned above.

In this literary review, we will look at creating our own communications platform for students as well as discussing other platforms from which we drew inspiration to create our own more specialized platform. We will discuss the platforms that students are currently using to learn online, as well as the benefits and drawbacks of using these platforms, as well as what students need from these platforms and how they help students.

### Primary Research

**Online Learning**

Prior to the pandemic, online learning was used whenever a student was unable to attend a class or lecture for any number of reasons; however, it was only since the pandemic outbreak that online learning has taken on a more prominent role within the education system. One of the main disadvantages of online learning is the requirement for a stable internet connection and the devices that allow students to work remotely.

The **Discord** platform was created to allow gamers to stream their games to their fans and communicate with them directly via video, voice, and text boxes. Users can set up their own servers for their own use and invite people to join their servers in order to build a fan base on private servers. This allows the streamer to control the server and control who and where they join, which can cut out some of the toxic content and attitude some players may display.

**Microsoft Teams** is designed primarily for use by lecturers to share content with their students. Microsoft Teams is a popular workplace application for video conferencing, chat boxes, and file storage. Teams has some great features that Discord does not have, such as a calendar that tracks meetings and classes. Teams also has a feature that allows control to be passed to another user from their computer remotely. This is a great feature when a student is stuck on a subject, the student can allow their lecture to control their computer to solve their issue.

From researching different papers online such as: <https://link.springer.com/article/10.1186/s12909-019-1701-0#citeas> – CITE HERE

When compared to students who were given the same quiz after an in-class lecture, education taught online using a gaming style teaching practice, such as “Kahoots,” showed a higher percentage of students participated and successfully completed the quiz. This demonstrates how difficult it can be to keep a student focused on their work and projects during lectures. Moving all lectures, labs, and classes to an online platform can be challenging, not to mention the hardware required to get students and lecturers set up to handle online teaching. When compared to students who completed in-class one-time quizzes, these gamification scenarios produced very high results. This may be due to the format of the questions used in the online quizzes, multiple choice and true or false, but students reported that using this format was easier to learn.

<http://journal.umg.ac.id/index.php/jetlal/article/view/2528> - CITE HERE

This article discusses how well Discord could be used in an online learning environment, how students feel about its use, and whether Discord could be used more practically with its features.

The interface, interactivity, feedback, and interaction sections in Discord were some of the most popular features among students. Students expressed dissatisfaction with the unstable internet connection and the device capability, which could be on the low side at times. According to the findings of this study, the benefits of using Discord outweigh the drawbacks of using this platform.

<https://ojs.library.queensu.ca/index.php/PCEEA/article/view/13882> - CITE HERE

When it comes to online learning and using the platform to communicate with fellow students and keep in touch with lecturers, most students have found Microsoft teams to be extremely helpful. Teams also has some fantastic features, such as a calendar, chat and share files. Emails that will be used to schedule Team meetings can be used to update the calendar. Students can use this to organize their workload and manage their time on projects and assignments.

<https://www.porto.ucp.pt/open/curso/modulos/doc/Definition%20of%20Terms.pdf> – CITE HERE

There are numerous terms that can be used to describe online learning, and one of them is virtual education. When the pandemic hit, lectures and classes went virtual. Virtual environments are intended to bring people together and allow them to interact virtually with their peers as well as their teachers/lecturers. E-learning does not appear to be the best way to describe online learning because it appears to be too narrow a term and does not explain enough about online learning and what is involved, such as hardware, software, and stable internet connections.

Upon completion of this review, we have all reached the same conclusion that these online platforms share many characteristics such as chat boxes, live chat, and file sharing, which is everything a student would need to help with their workload. However, there are some differences. Discord can have many different servers or groups to loin and files are kept private, whereas Teams is a collaboration, and everything posted is public. When it comes to file sharing, Discord has limitations due to a maximum upload size. Teams is also set up to interact with more productive apps to enhance a user's experience.

### Conclusion of Lit Review

This study has revealed what students want and require in a communication platform. With the pandemic, it was critical for students to continue their studies as normally as possible. When it comes to a student using a communication platform, we want to cover all of their bases, and if we can provide all of their needs in one platform, it will be a huge help and time saver for students.

## Methodology

Having decided to create a student app for tracking classes and helping the student keep notes and stay in communication with their fellow students in these trying times. Student-mania will provide a platform for students to track and complete their workloads while also helping with time management on projects and lab assignments.

As a fellow student one of the biggest tasks is to keep on top of my workload that I receive every week from lectures, being able to connect with students on discord and lectures on Moodle and having live lab sessions on teams there is a lot of different applications on the go at the same time, it would be far more efficient to perform all this tasks within the same program or application. This would be a hub for students and lectures to use and to display notes from classes, to form a bond with your fellow students and to help one another as you would if you were working in the real world. Student-mania would perform all these tasks cutting down on the overall use of the different applications and have everything a student needs within this one application.

### Research

The student will login and have access to their notes and timetable for there week ahead. All their classes and lab assignments will be displayed, and a forum will be added to allow students to ask their fellow students about any inquires or other questions they may have.

Video conference would be a great feature to have within our application, the ability to talk and see your fellow students goes a long way in helping each other out when it comes to their workloads and will help in the social side of getting to know your peers. Part of college life is interacting with your fellow students and I think a feature allowing video calls would be a big step towards helping students in communicating. At the very least having a chat box that students can use to communicate among themselves.

### Frontend

Research into html and aws amazon.

#### 1st approach (long way):

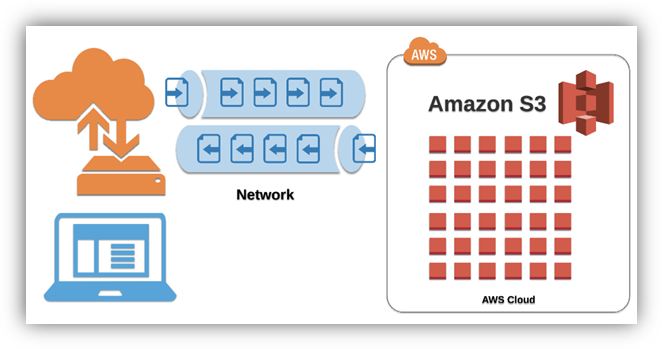
HTML & CSS files to aws amazon:

When setting up the application to be hosted by aws amazon we had no prior knowledge behind the hosting service. This which had us to doing extensive research into the hosting providers.

We Had to research the following before uploading anything: S3, route53, Ec2. Which we narrowed it down to after looking up what we would need from amazon’s web services.

To begin we researched S3 buckets. After a few posts and videos, we became aware that it is amazon’s way of storing files. Our understanding at the beginning was limited to seeing just files but after doing some more work on it we could see that you could restrict access to these files. We learned that this is used to protect the access of the files which can be retrieved from anywhere on any device through the web if needed.

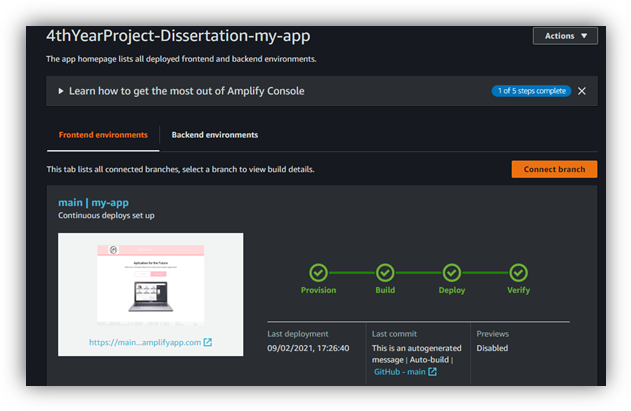
Below Showing a Diagram which we believe describes the process.



Following this research, we began looking at Route 53, we started to understand that this would be needed for directing the files that would be uploaded. Route 53 being amazons widely available and scalable cloud domain name system (DNS) service. From this we learned how it helps host your files to the web for the public to see on any browser worldwide.

***\*\*\* INSERT [1st way issues section here ]\*\*\* overleaf order***

#### 2nd approach (short way):



HTML & CSS & JavaScript & react to amazon aws:

## 

Uploading to amplify we discovered that we can only host the application to amplify if we were the admin holder of the repository of the project. We soon uploaded the project to amplify but came across some issues. Here where we wanted to be able to contact customer support but due to them removing the live chat functionality unless you pay for a plan, we were unable to access. We then had to research why we were getting the error “page cannot be found error 404”. We knew to be a linking issue as it was there but couldn’t be located. After extensive research we didn’t find the solution, we found that this was not covered enough within online forums or any of our other research, which produced no results.

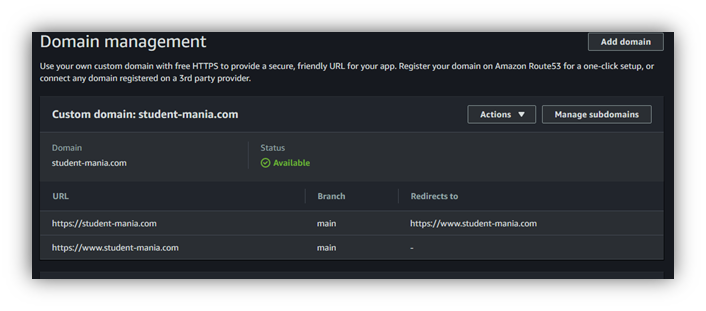
The solution we found was that when you are uploading to aws amplify and linking your repository to it you must click on the small box at the end that allows the admin to specify the exact folder of the application within your repository folder. This is not covered or explain to be “needed”. When we were going through the process, we thought this was an extra feature, but between trial and error we figured that this feature needs to be ticked to declare the primary “my-App”. We also put two and two together as when we were researching, we came across that when you are uploading the repo to amplify that when amplify discovers your app it looks for your “package. JSON” file. This then tells amplify all the commands that are needed to build the application.

Once we got round the issue of error 404, we had the application hosted by aws amplify as seen running through the deploying stage of amplify.

During the 2nd way that we were trying it was unfortunate that amazon started to set up plans for users to have to pay for help off customer service if you wanted help in other ways than live chat. This feature which we used at the end of the first way of going about uploading the application website to aws amazon. As after we uploaded everything, we were still unable to see the website but on live chat we were helped out within minutes. Being explained to us that it could take up to 48hrs to fully upload and be visible on the web with the URL they provide.

### Linking aws amplify with custom domain –

In our learning and researching of linking your domain to amplify we learned that when you setup a route53 it provides you with the four name servers and SOA .But when you go to your “Domain management” on your aws amplify and select the route53 you setup to link the two it generates the “CNAME” and “ALIAS(A)” for you. This which had to be done manually when not using aws amplify.

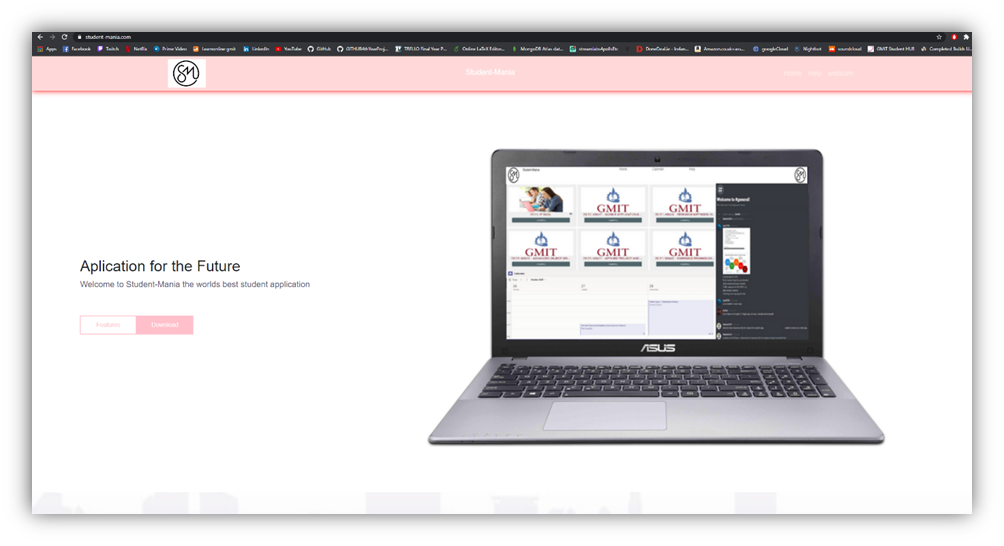


Since we have done this both ways, we know what amplify does for its users, so they don’t have to setup everything themselves which can make setup very convenient for users.

In the process of adding our custom servers to our GoDaddy domain we ran into an error when we clicked the save button. This error which was “an unexpected error”. When we looked up this error it came back to be a common error. This suggested to access it on an incognito window or clear the user’s cache. This which did not work for us and we ended up having to contact customer service via live chat.

We were extremely happy to hear they had live chat but unfortunately the customer service person did not provide us with a reason behind the error. The only solution was that he entered the name servers into the account for us and save it then. We found this to be very strange but decided to try the same thing again but under a different name but got the same results and got the same error. After all the issues by the end of the day we were happy to say we got the website up and running and searchable through google for the second time using a different approach.

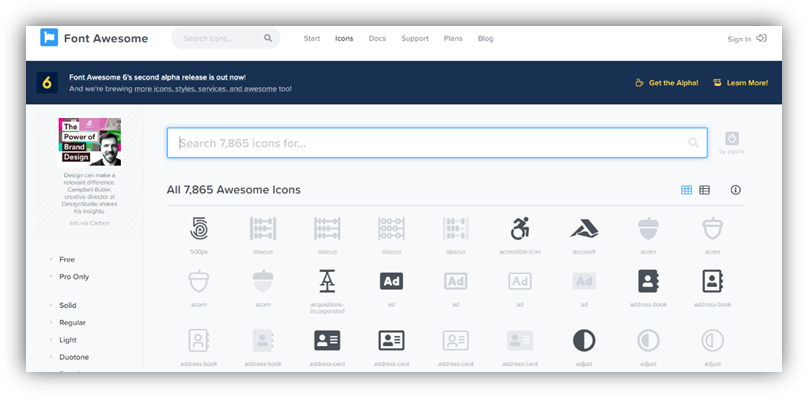
Amplify is very useful when making changes to your application as you don’t need to reupload your files to your s3 bucket to update your online application. It simply automatically rebuilds your application when you make a change/commit to your linked GitHub repository. This which saves time and helps in faster development of applications.



|  |  |
| --- | --- |
| **Pros:** | **Cons:** |
| Updates online App when you commit to GitHub | If Errors in code wont build. |
| Don’t have to do the linking between files and servers, does it for you |  |
| Makes you app public by default |  |
| Can add tests to application when uploading if you wished for in the build process |  |

## Amplify Table

### Font awesome Website:



We used Font Awesome website to get icons that we could use to link our social media accounts to our website for advertising and easy access. This makes navigation between the social media accounts easier for the users.

To use these icons on our website we had to make an account that provided us with html code that we had to have embedded in our code to allow the icons to appear.

This after having imbedded in the body of my code allowed us to search for icons on the website. There we selected the one we wanted, and it then provided us with more html that we simply had to paste into our application where we wanted it to be displayed. Here where we embedded this html code from the website into a href tag that allowed us to reference a website when a user clicks on the icon. This we repeated for each social media account that we wanted to reference.

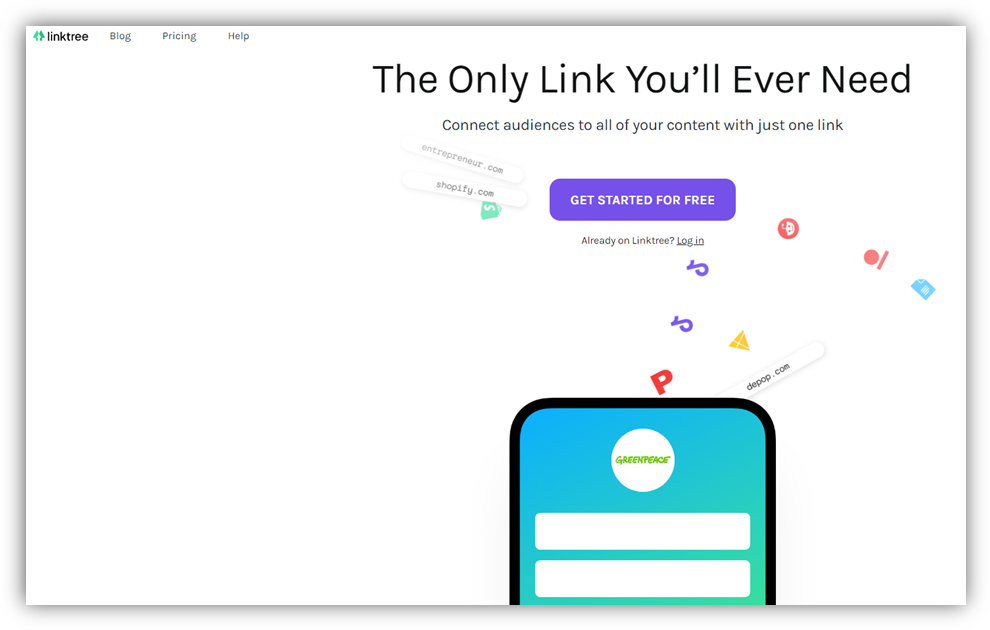
### Linktree website:

This website I signed up to for it to allow me to link multiple websites to the one account. This which makes it easier to list to the users the accounts all in one place neatly. I had to sign up using my college email and verify the email. Here I created the account under the username of the application called Student mania. This which I thought was a good idea for later expansion when multiple websites will be created for marketing purposes. Added this to the website to show its purpose and its workings.

### Linktree Table

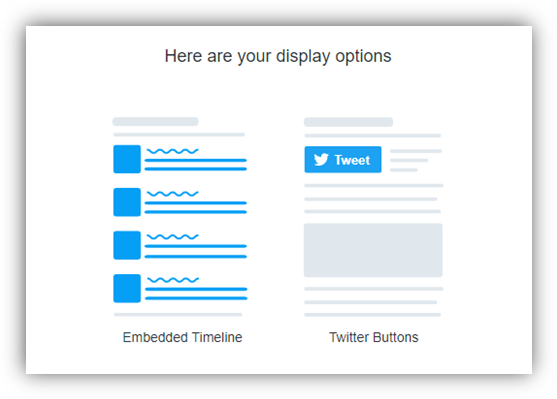
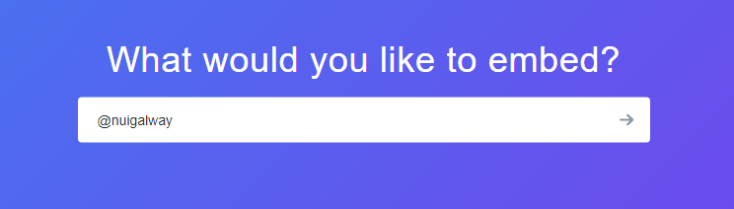
|  |  |
| --- | --- |
| **Pros:** | **Cons:** |
| Allows expansion | Have to setup another account |
| Neat layout |  |
| Less icons on website to be displayed |  |

### 



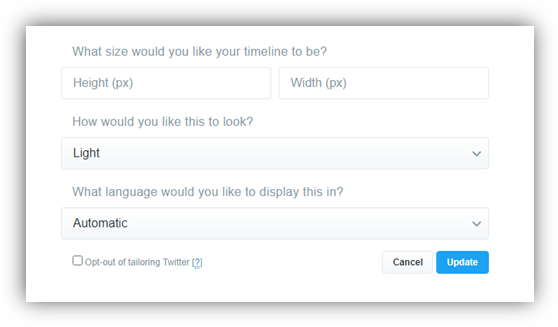
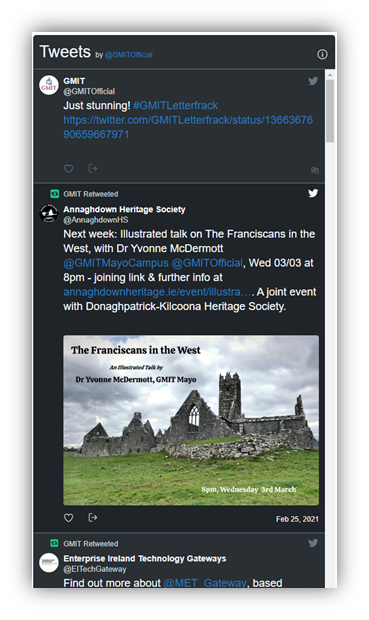
### Publish.twitter.com - website:

This website which allows you to enter the twitter account of your choosing and be provided with the embedded html code to paste into your website. This code that displays the twitter feed from the twitter account chosen.



After you select the account you can chose which way you would like to display the account feed on your website with two options. **(1)** Embedded Timeline or **(2)** twitter buttons.

Then you can customize the feed to your liking. This which can be size, width or even colour.



Here you can see the outcome after some custom inputs were chosen.

This which wouldn’t display correctly when added to the html as it was a react application. This which required the following.

1. To install the dependencies for the twitter feed to display in a react app

npm install --save react-twitter-embed

1. To import these dependencies to the html

**import** { TwitterTimelineEmbed, TwitterShareButton, TwitterFollowButton, TwitterHashtagButton, TwitterMentionButton, TwitterTweetEmbed, TwitterMomentShare, TwitterDMButton, TwitterVideoEmbed, TwitterOnAirButton } **from** 'react-twitter-embed';

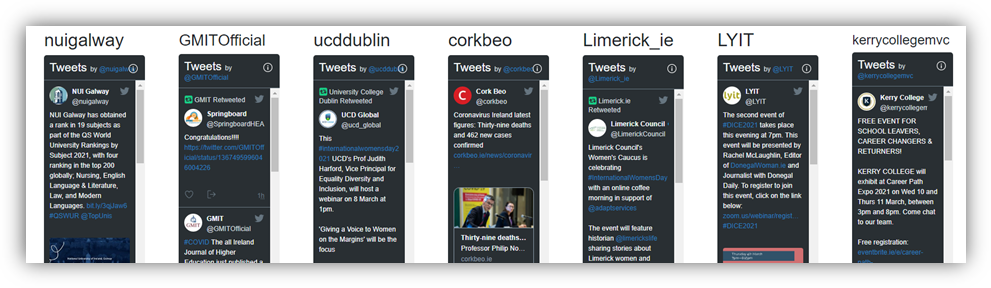
1. To add the following tag to display the feed on the page

<TwitterTimelineEmbed/>

This which allowed me to display multiple twitter feeds on the application.

## 

## 



## Technology Review

Having decided on our project we have had a few meetings at this stage with our supervisor and have decided on breaking up the project between the three of us:

Jack – WEBRTC the back end of the project and will use real time media communications. E.g. HTML5

Kevin – MongoDB setting up a database to allow information to be stored. E.g. Login details

Stephen – Frontend the user interface that will be displayed on our site. E.g. CSS files

***WEBRTC*** will enable us to use audio, video and share data communication between browsers without having to use any plugins. This will allow us to simplify communications on our website and will also improve the users experience while they are login into our website. After becoming available in 2011 WebRTC has only grown in popularity with an estimated 2 billion browsers that are enabled to use WebRTC. WebRTC is an open source project that can be customised as the developer needs it is also completely free to use and is constantly evolving and improving over the years.

***ZOOM*** and ***NETFLIX*** are only two of some of the biggest names in the industry that use WebRTC to help improve their products.

***MongoDB*** is the leading NoSQL database that is currently available on the market now and uses a document-oriented database. Large or small datasets are supported by rich query’s with fast response times and enables agile development using dynamic schemas and flexibility for requirement changes as the developer sees fit.

Using ***MongoDB*** gives us scalability, performance, and high availability that other databases would not be able to handle. MongoDB is already a proven solution for all business requirements to companies on a global scale. MongoDB uses a collection of documents, each of these documents include key/value attributes, a document is seen as a row in a table, with each key used as a column name and each keys value is like the rows value. The different is that a document is not constrained to be a certain schema or a column within the table. These documents may share similar elements such as ID fields and having different elements within the document. An example if this would be a garage and a shop, they both could have ID fields but then differ afterwards.

### Key Features

**Flexibility** - MongoDB was designed to work with cloud platforms and commodity hardware, data is then localized for queries that will ensure performance is at its peak no matter the deployment size of the project.

**Scale-Out –** MongoDB has been designed to be scaled across server clusters, when the data starts to grow, the user can add more nodes to the clusters and MongoDB will balance the data among the clusters evenly and automatically in the background.

**Dynamic Schemas** – This lets developers make changes that are required in the system without affecting the existing data already stored in the dataset, not incurring any downtime. With other schemas they must be defined before data is inserted. With MongoDB the schema is dynamic this allows it to be scaled as needed and supports fluent polymorphism.

**Rich Querying** – MongoDB uses a full query language; a primary and secondary indexing is also used and a Google like text search.

MongoDB is the perfect database to use within our project, with its flexibility, scalability and high performance using dynamic schemas this is perfect for our project. MongoDB is a document-oriented database that will allow us to store JSON documents with a dynamic schema that we will use to store our records/logins and will scale up as we need and also allow us to add or remove field and types as needed to our project. No other NoSQL or relational database software makes it as easy to use and to develop databases.

# Primary Research

### Survey

As a group we have decided to produce two different surveys to get feedback on the project and get an idea for what would be needed to keep our platform up to date, useful and the right tool for students to use to communicate with their fellow students. “*Survey Monkey”* was our first choice but after researching into it, we discovered that there was a limit to the amount of responses that we could receive before paying a fee for this we decided to migrate over to Google forms. With no limits we were free to conduct as many surveys as we needed to use as we saw fit.

For our surveys we researched different survey forms and example type questions that we could modify to get the best survey responses from our potential future users.

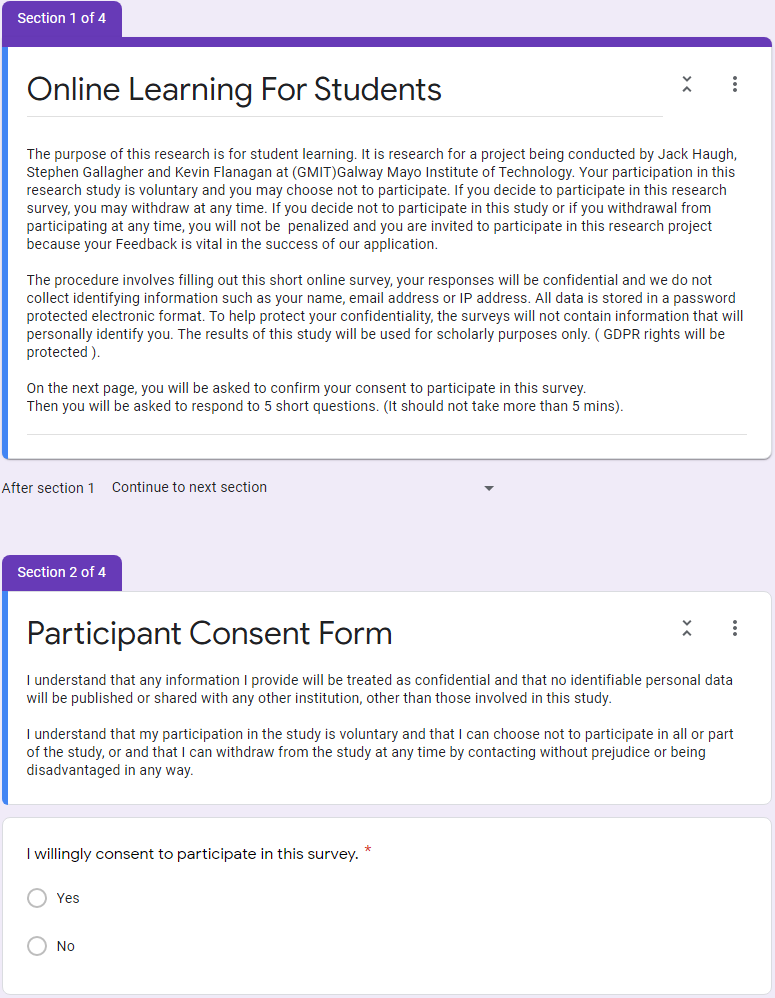
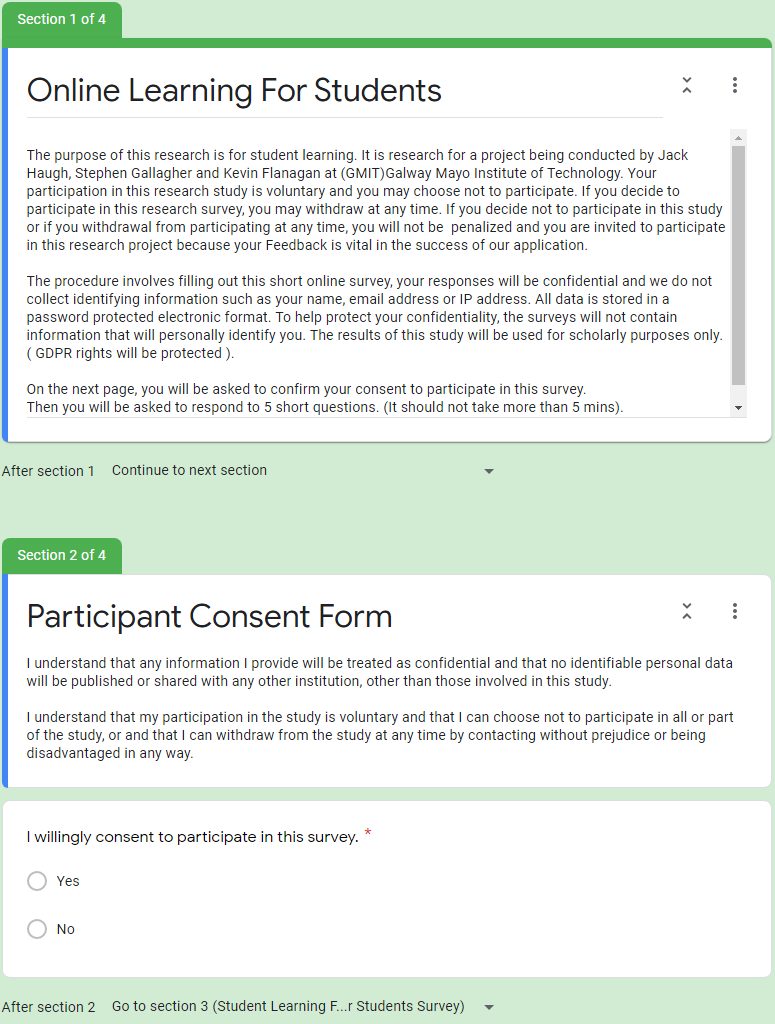
This consisted on different type questions like the following:

• Open-ended questions • Likert scale questions

• Multiple choice questions • Demographic questions

• Picture choice questions • Closed-ended questions

• Rating questions

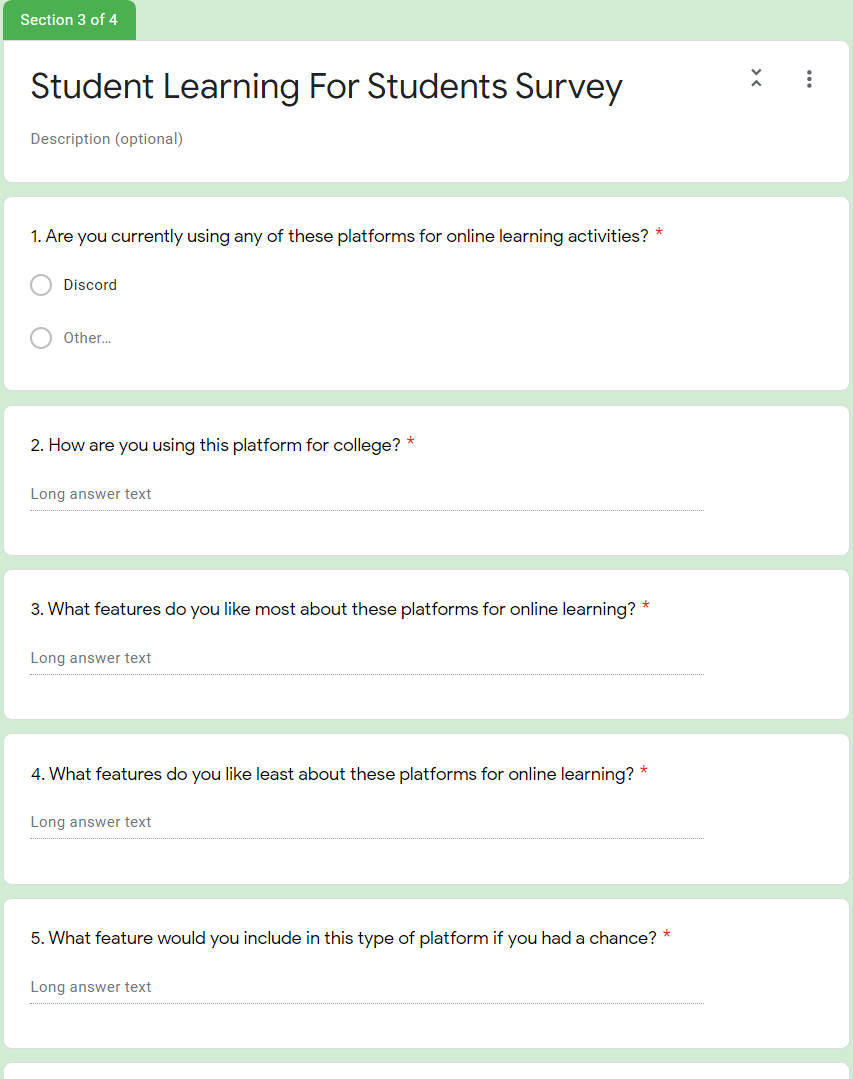
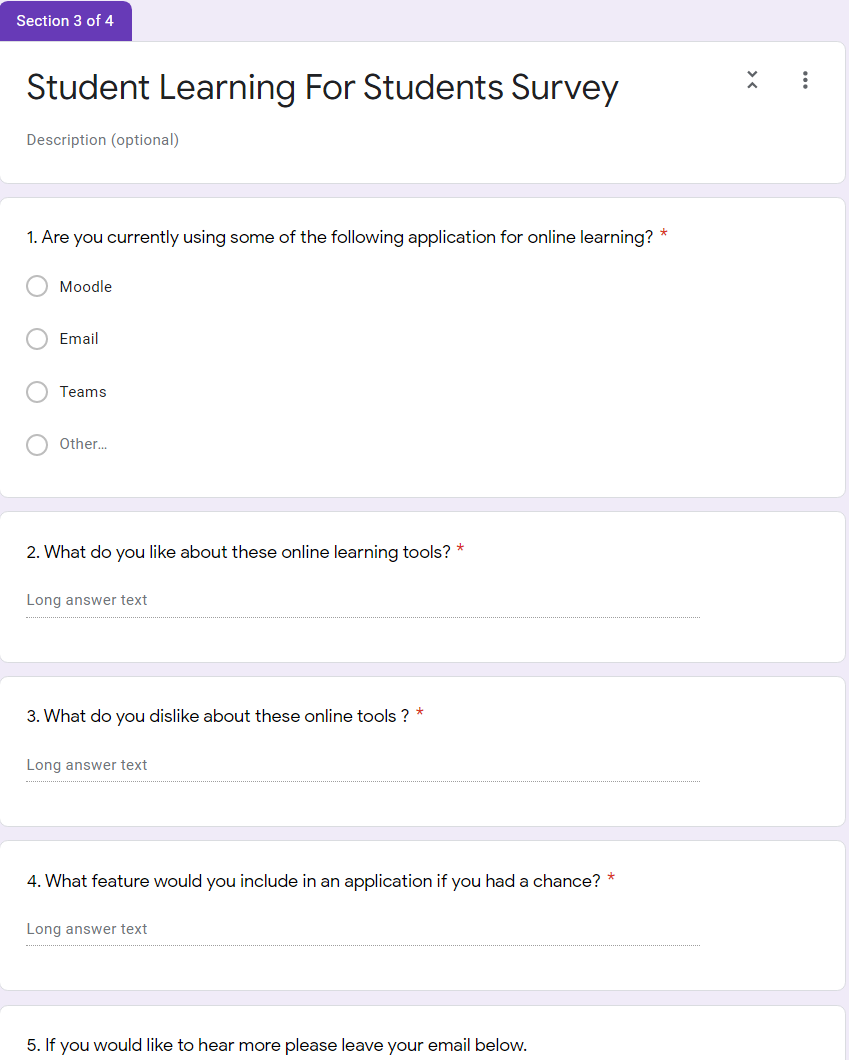


Working on Google forms we will create these surveys and send them out to our fellow peers and see what feedback we can get. This feedback will hopefully lead us down paths that we wouldn’t have thought of ourselves and give us some insight into what people would like to have implemented into our website.

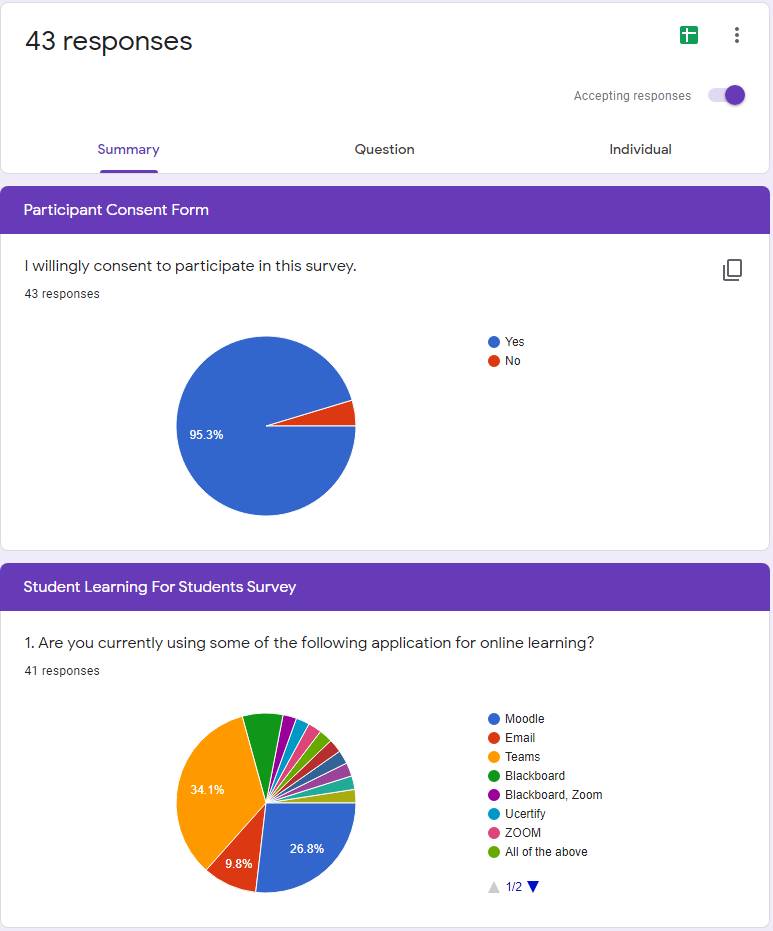
After deciding what we were going to ask (Questions below) students what features and what applications that they would like to see included and what would be helpful to their learning. Based on feedback we will be deciding on what we should be creating in our platform.

We set up the two surveys on Google Forms, where we created a quick survey to gather information from students and other users with about 10 questions and some terms and conditions for the user to agree too. With this information we plan on adding or deleting different functions within our website and will give us an idea of what other students are looking for. The goal of this first survey is to hopefully come across an idea or function that we failed to see ourselves and if possible, to implement said function into our own site. The first survey we decided to email it to our peers and students in other courses to see what their responses would be to our survey.

Example of questions we used in our two surveys:



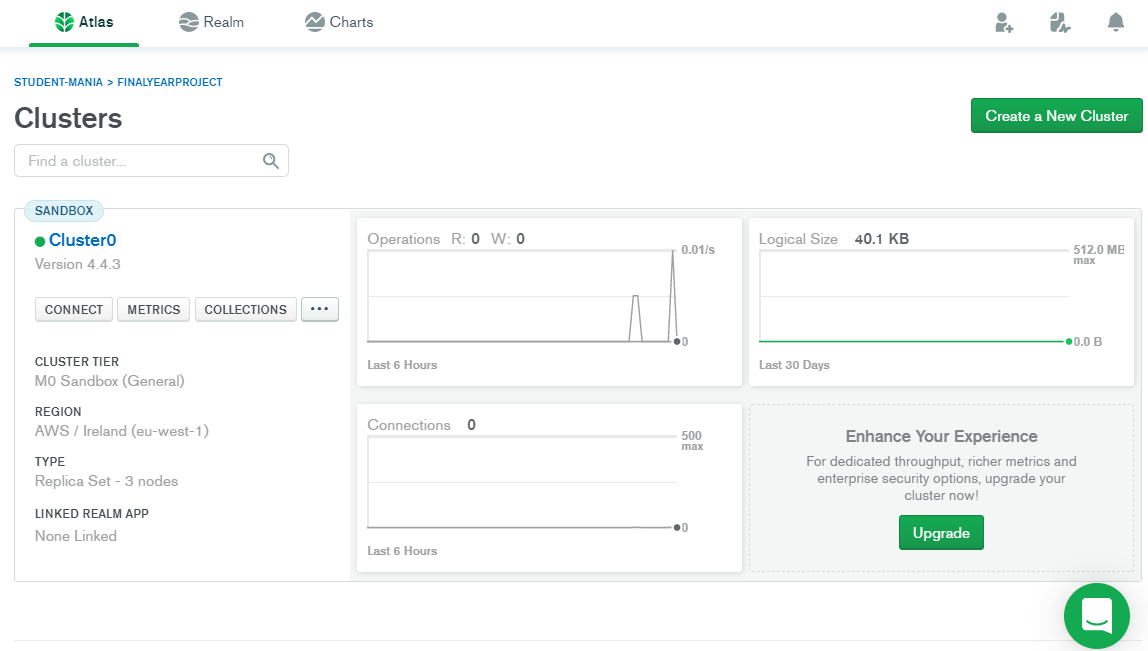
The second survey we decided to be more specific in our questions, this will let us narrow down the possible results and answers that we receive back from the participants of the survey. Again, we are hoping to get that one bit of information or idea that we never thought of ourselves. For this survey we only needed 5 questions and they were aimed towards Discord which is a VoIP, instant messaging and a digital distribution website/app, from our first survey we discovered that this seemed to be the main way that students were communicating outside of college websites such as “Moodle”. This information let us narrow down the questions (5) we asked in our second survey and got a better view of what students wanted from this type of website/app.



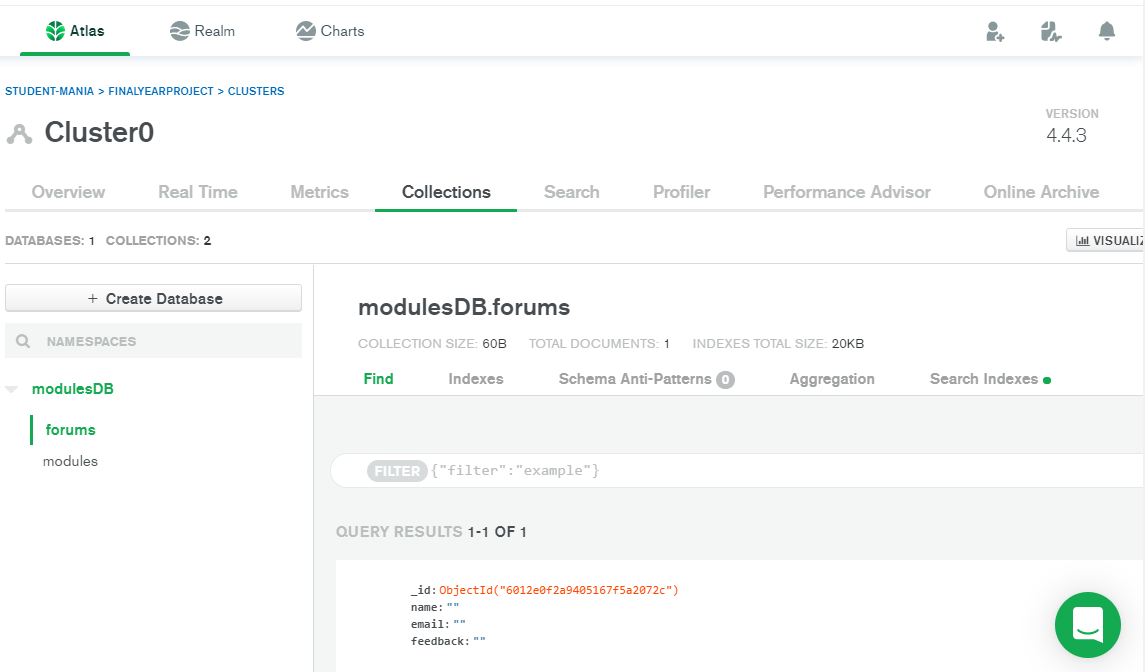
With some great data gathered from the surveys from Google Forms we have a good idea what is expected in this type of website/app. From this information we hope to achieve a website/app that our fellow students would not only be happy to use be would also be helpful in their college workloads.

### Databases

The front-end has been created with a great display and very user friendly. We plan on having our website running global and have our databases and servers running with a GoDaddy domain letting it be accessed from anywhere. The next biggest step is to set-up MongoDB to be our database for the website to store details and secure logins. This will allow students to share documents and notes from lectures and classes, forums to ask questions, timetable to track classes and labs.

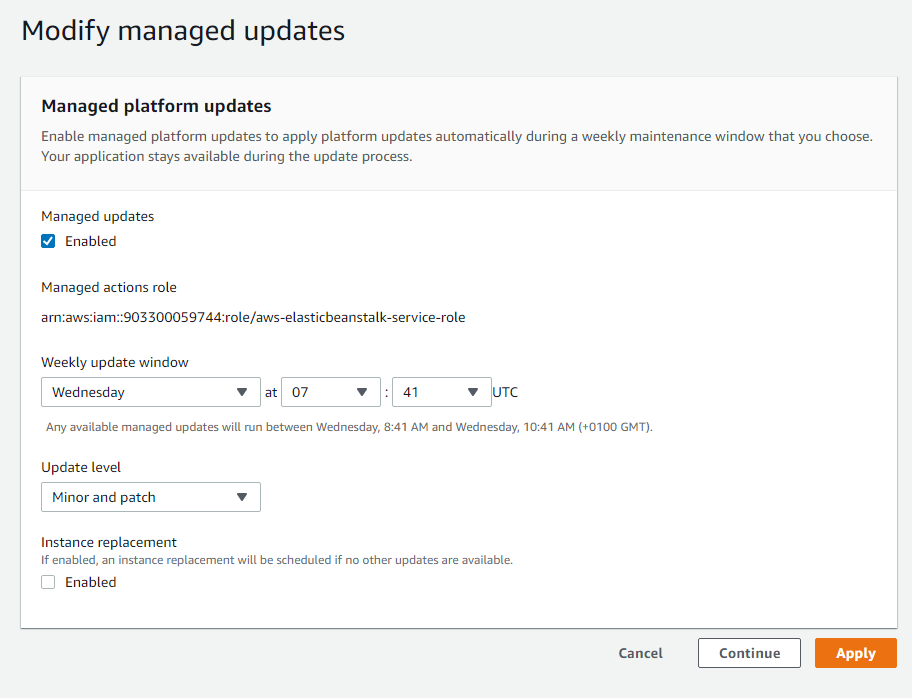


MongoDB is an object-oriented, dynamic and a scalable NoSQL database that stores information in data objects. These data objects are then stored as separate documents inside of a collection. These collections are created by the user and then a cluster is created within the collection.

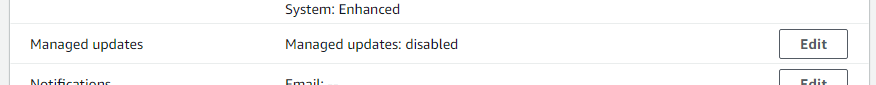


From the image above we had to connect the cluster to our application which involved a line of code, that would allow our website/app to connect to the database to write and share information while also saving it to the database to be recalled when server is running. To allow mongoDB to receive information from the website/app while also returning the information when called upon in the future we created a database called modulesDB which had collections that will store information that is desired for the different sections of our website/app. We then created different classes within our directory that “talked” to each other to send and receive information. This information is needed firstly to receive the information from the user and then secondly to return that information to be displayed for all users to be seen on our website/app.

### Database/elastic beanstalk issues



On elastic beanstalk you can enable auto updates. This option which allows you to pick a time and day each week that automatically updates your platform/application.

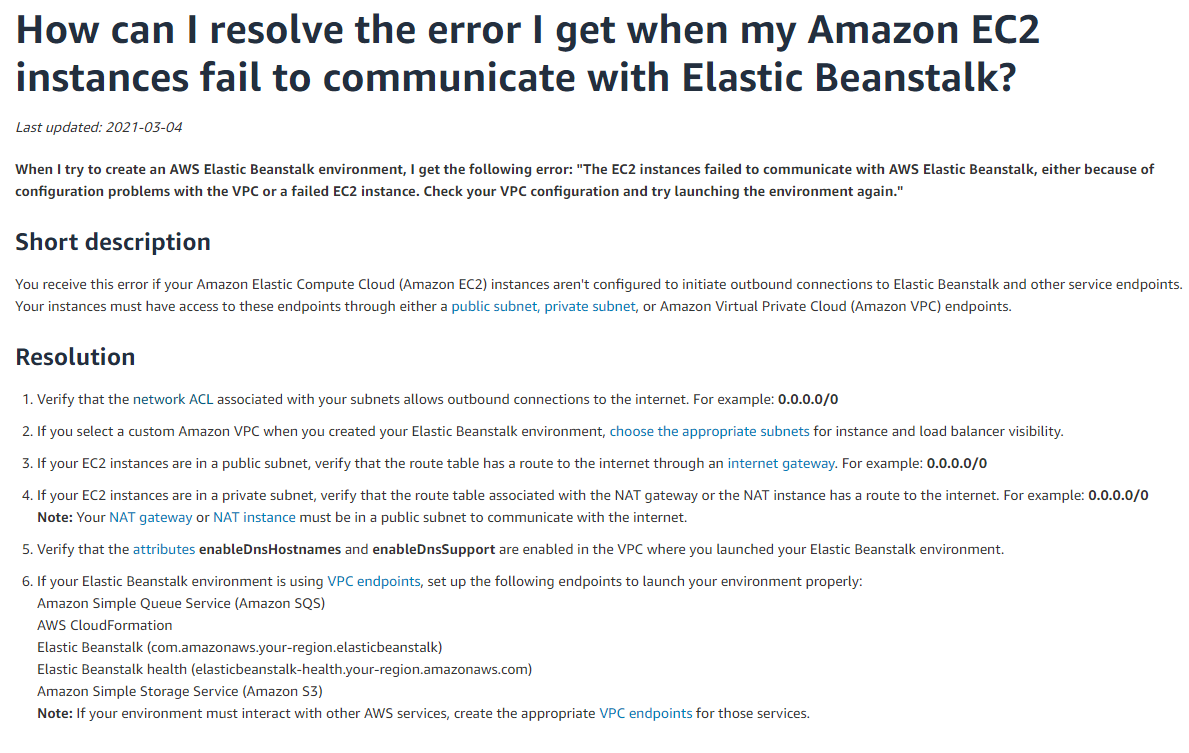


This can be found under configuration at the bottom few options.

**ERROR WITH INSTANCES:**

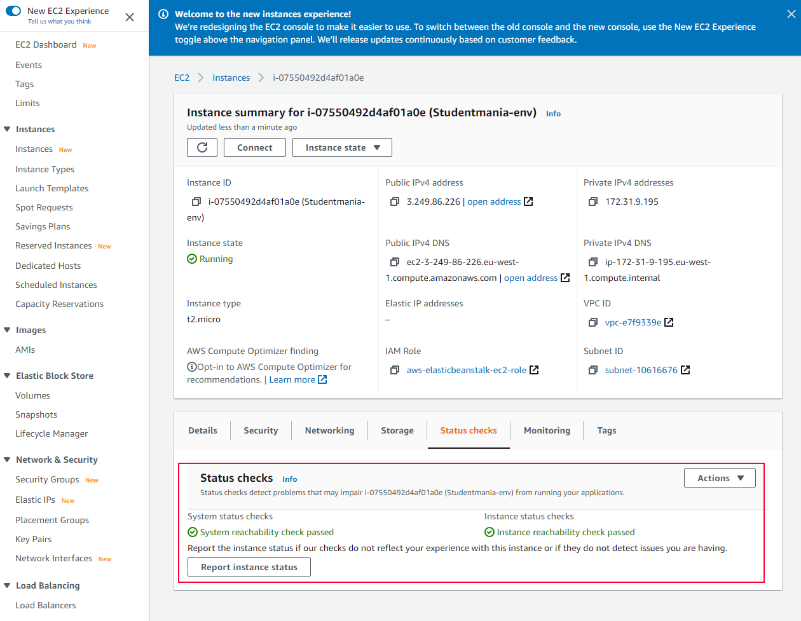
Looking into issue below:

Site : <https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-instance-failure/>

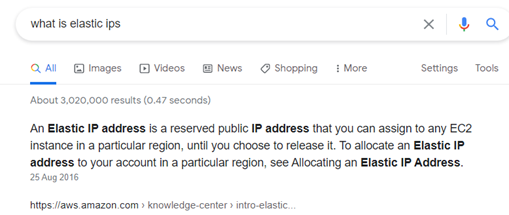
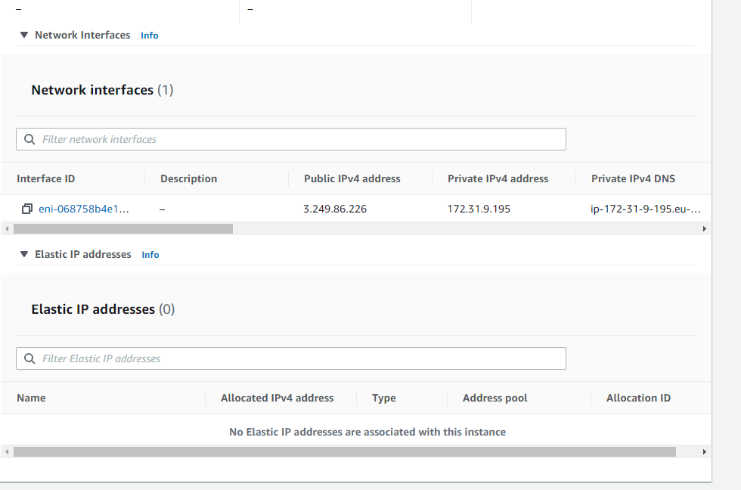


Based on the list above I tried going through each section and trying to narrow down the issue.

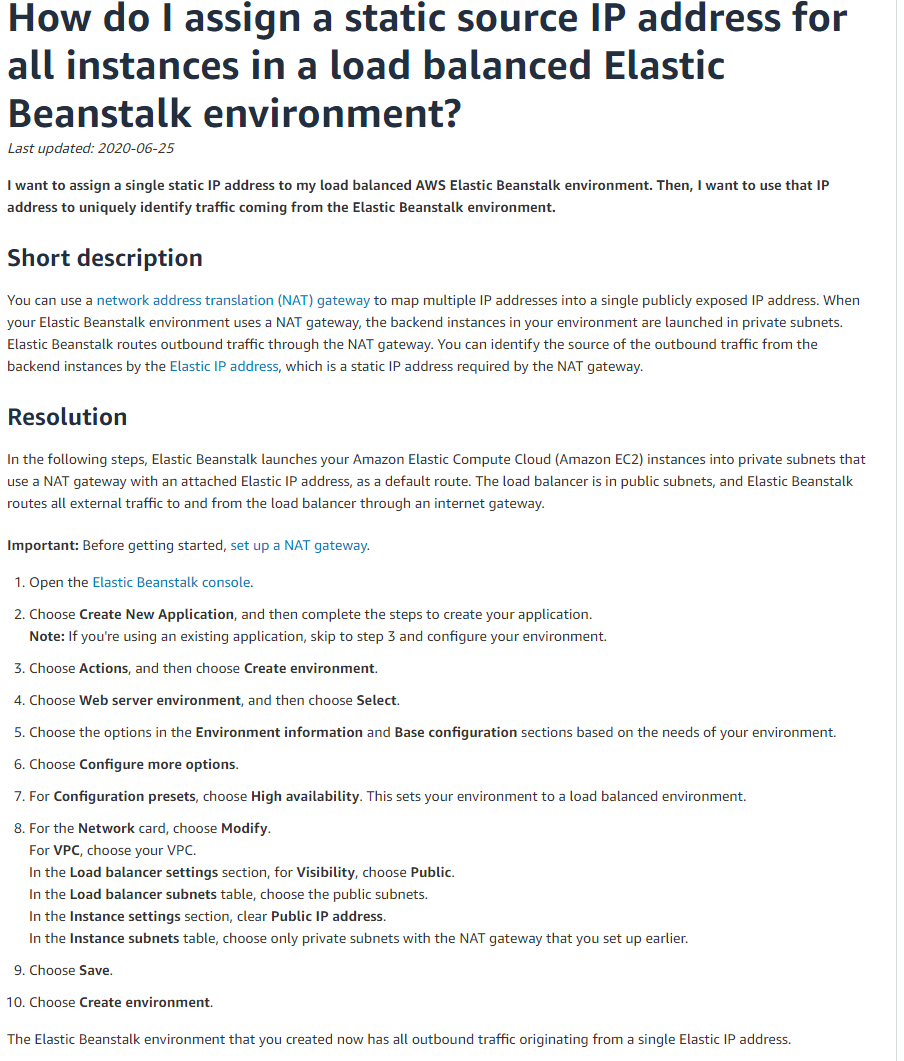
Started with checking is the instance running ok in general.



Noticed there was no Elastic Ip addresses assigned which made me look into it, to see if it was required for the traffic.



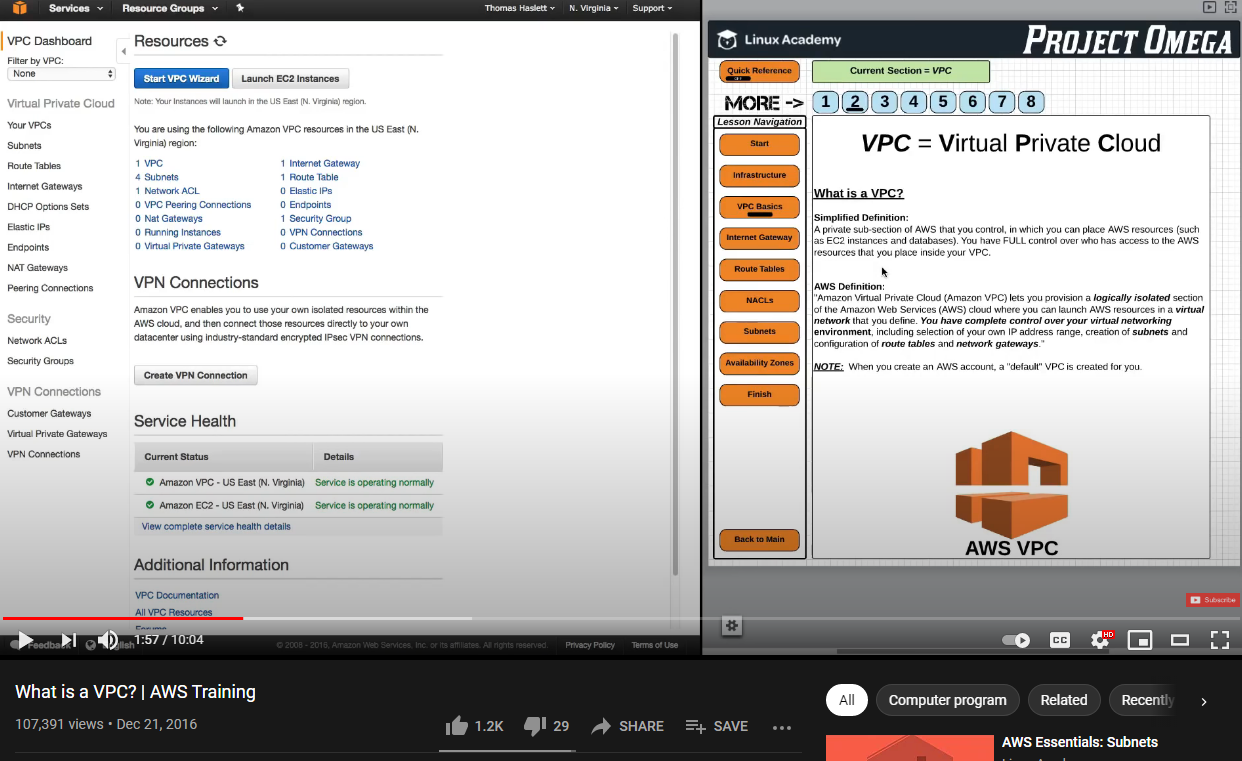
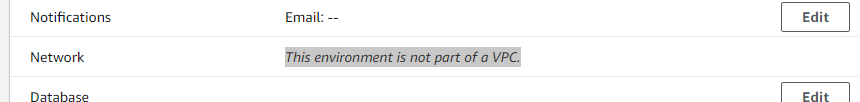
Started looking into VPC which is a Virtual Private cloud that allows you to control who has access to your AWS resources, because I noticed on the following website:



<https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-static-IP-address/> when setting up the environment on AWS you have the option to select VPC or load balancer which can balance the load for you and allow you to add extra instances but you can make the visibility public. As seen in image below

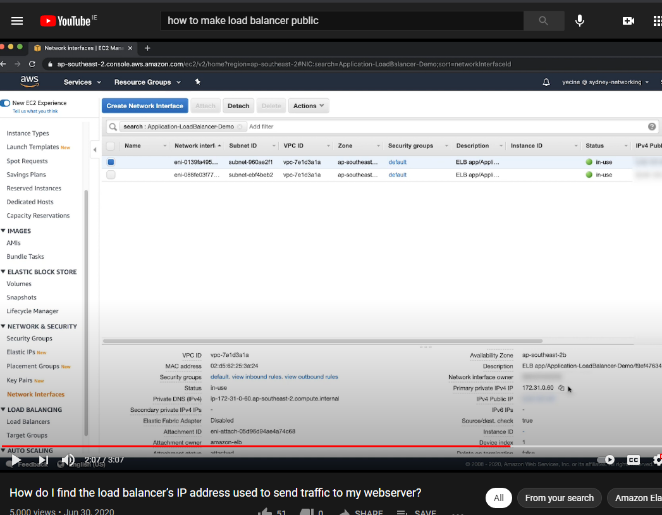
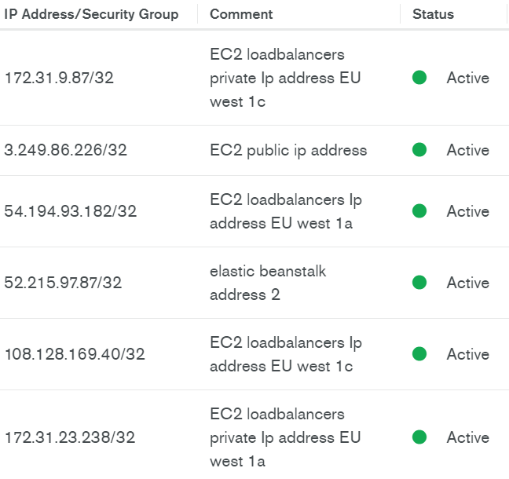
I also took to YouTube for more of an insight to VPC on AWS.

Also took note that our Network on our environment was not part of a VPC



Looked online for videos etc and came across the way in which to find the load balancers IP addresses. These addresses I added to MongoDB to see was it an access problem.

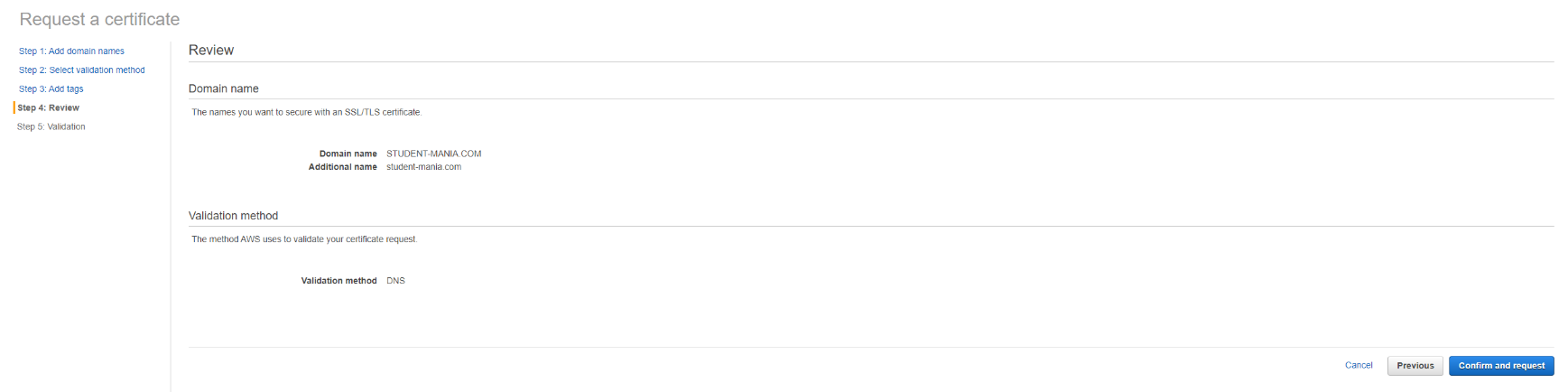
<https://www.youtube.com/watch?v=PKjbuxnispM&ab_channel=Academind>



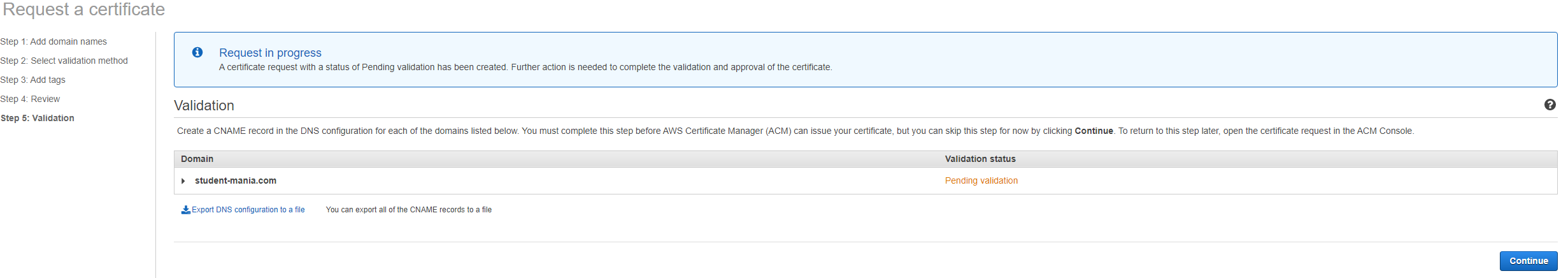
Looked up a video and it started to walk me through elastic beanstalk and linking the domain with the database through HTTPS. Which I believe to be for security reasons.

Video link : <https://www.youtube.com/watch?v=BeOKTpFsuvk&ab_channel=WornOffKeys>

Requesting certificate for domain:



request in progress:



looked up how long it will take:



**Firebase** is a Backend-as-a-Service provider (Baas). It offers a range of tools and resources to developers to help them create high-quality applications, expand their user base, and benefit. It is based on the infrastructure of Google. Firebase is a NoSQL storage application that stores data in JSON-like formats. Firebase is Google's mobile platform that allows you to create high-quality apps easily and grow your business.

## Web RTC

## Conclusion

## Communication is critical. Our primary goal for this project was to assist with the resolution of a crisis. As students, we determined that the most pressing issue for students affected by the pandemic at the moment is connectivity and managing their workload. As a result, we agreed to develop a communications platform that could grow into something more in the future. After some deliberation, we settled on this concept before considering any apps or programs that could benefit students. Some of our earlier concepts included a delivery app and a lift-sharing service. These were all promising thoughts, but communications seemed to be a student’s main problem during the pandemic.

## Once we had an idea, we began exploring various features and programs that we might develop to assist students in all classes, not just computer courses. Few of our functionality were beyond our abilities, so if we could step back in time and build or even contribute to this platform, we could have modified a few things. Live contact between two students using chat boxes was one of the features we developed for our platform. MongoDB is used to store questions and requests that students can ask their peers. This provides the student with two points of communication from the other students: (1) when a student poses a question, they may leave their email address and receive a private reply, or (2) the answer can be left in the chat area for other students to use and even comment on. Some other features are a help page and a social page with twitter feeds from different colleges and educational institutions from around the country.

## Amplify AWS is the technology we used to launch our website's frontend. Amplify AWS uses scalable full stack applications with SDKs (Software Development Kits), which are libraries, tools, and documents that will assist in the development of an application. We used Google's Firebase for security purposes, and firebase authentication was how we set up our user login and registration. Firebase is a backend tool used by some of the world's largest businesses, including Instacart, Stack and Twitch.

For our video calls we used WebRTC (Web Real-Time Communication), which allows real time video communication between peers.

We also included a software page where a student might update or find links to some software that could be used in the subsequent months of their course, as well as some general software such as word that any student would need at any point; this section can be customized to fit the course that was attended. We have added a QR code to our home page for students to share with their peers in order for them to easily access our application.

## Reference

<https://www.typeform.com/surveys/question-types/> Survey Research carried out

<https://ie.godaddy.com/> GoDaddy

<https://www.mongodb.com/> MongoDB

<https://www.google.com/forms/about/> Google Forms

<https://aws.amazon.com/amplify/> AWS Amplify

<https://aws.amazon.com/elasticbeanstalk/> Elastic Beanstalk

<https://learnonline.gmit.ie/> Learn Online “Moodle”

<https://discord.com/> Discord

<https://trello.com/> Trello

<https://www.surveymonkey.com/> Survey Monkey

<https://linktr.ee/> Linktree

<https://www.youtube.com/> YouTube

<https://www.instagram.com/student.mania/> Instagram

<https://www.facebook.com/Student-mania-108833497918996> Facebook

[https://publish.twitter.com/#](https://publish.twitter.com/) Twitter

<https://fontawesome.com/> Font Awesome Icons

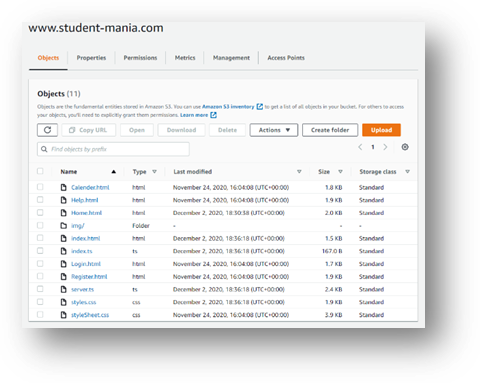
<https://reports.internic.net/cgi/whois?whois_nic=student-mania.com&type=domain> Name Domain Server Check

<https://www.overleaf.com/> Overleaf

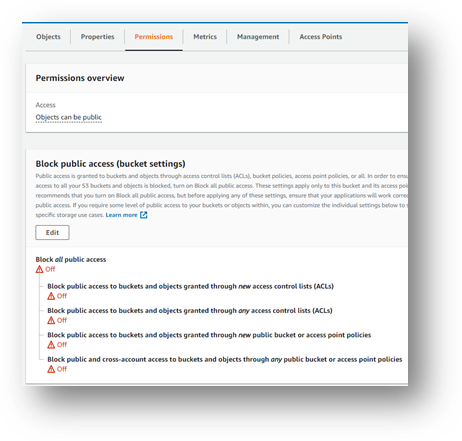
## Bibliography

***\*\*\*1st way issues section to put at start in first way:\*\*\****

When beginning the implementation of our research we came into some difficulties and roadblocks. With the s3 buckets. Some hidden settings that must be turned on. The two issues we had with Amazon's S3 buckets were that we didn't realize you had to allow permissions on them when we uploaded all the files for the application. This allowed users to have public access to these files. To be able to see the files on the web, we had to make them static while also selecting to unblock public access

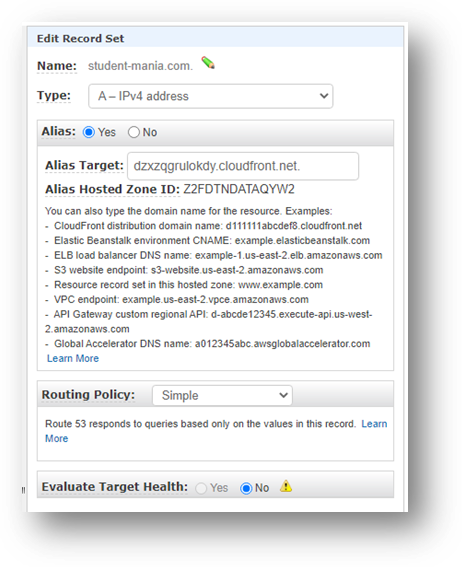


S3 bucket contents:



S3 bucket public access permission:

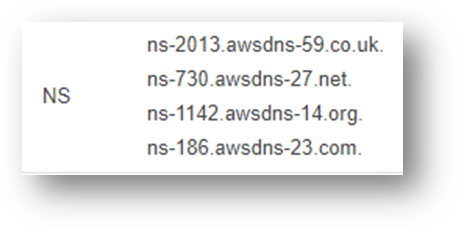
Route53 appeared to be for routing at first glance, but we had no idea what was required. After conducting extensive research, this became clear. After watching videos and participating in forums, we realized what we needed.



This included a "Record A," which maps the URL to one or more IP addresses when the IP are known and stable. The [A record](https://support.dnsimple.com/articles/a-record) points a name to a specific IP. For our application we used the alias target that amazon provided us with. This which saved us from looking up the IP address of the domain. It is by default that amazon give you a hosted URL

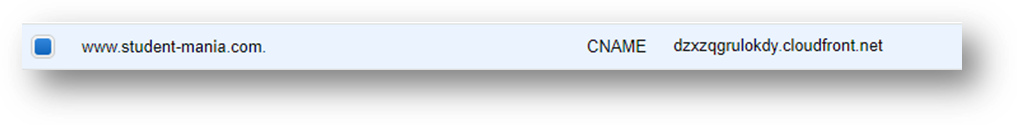
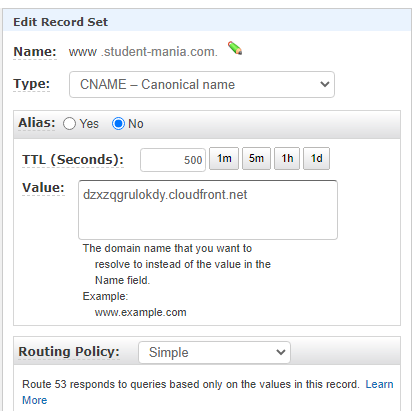


A "record SOA," which is the authority record that identifies the domain's DNS information, was also required. It keeps track of important details about a domain or zone, such as the administrator's email address, the last time the domain was updated, and how long the server should wait between refreshes. We discovered that in order to comply with IETF standards, all DNS zones require a SOA record.



What was also needed was what Amazon provided by default. The name server (NS) record. Once you've created a hosted zone, the name server record is created. It contains a list of the four authoritative name servers for your hosted zone. They advise against making any changes.to these.

A “CNAME” record was the final requirement that we needed. This record was required for mapping our Amazon default URL domain to www.Student-mania.com, which we purchased on GoDaddy.com.

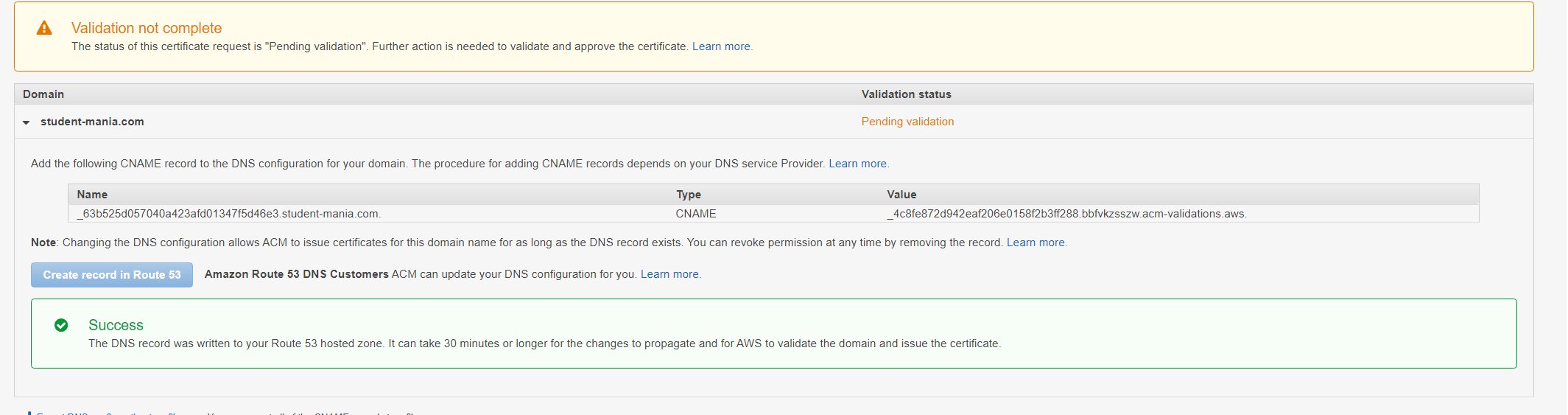


\*\*\*FIXING HTTPS ELASTIC BEANSTALK ISSUE \*\*

Above list the conversation with images with the goDaddy Team.

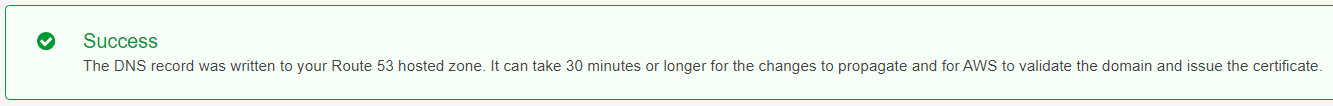
Helpful video : <https://www.youtube.com/watch?v=MVBAnucCwrg&ab_channel=CodeWithNate>

Step (1) Adding route 53 CNAME for HTTPS certificate

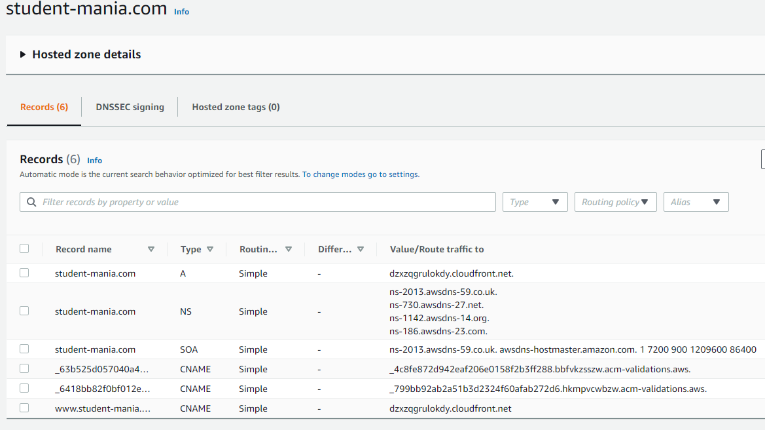


Step (2)

Ass you can see we added to route and now it is saying it can take up to 30 minutes for the change to propagate and for aws to validate the domain we have setup

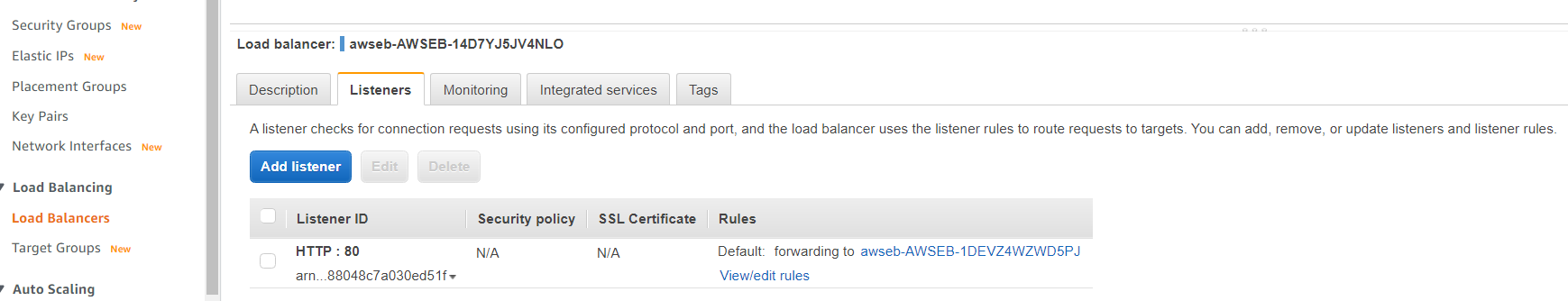


(2.1) after some time you can see it had been added



Step (3)

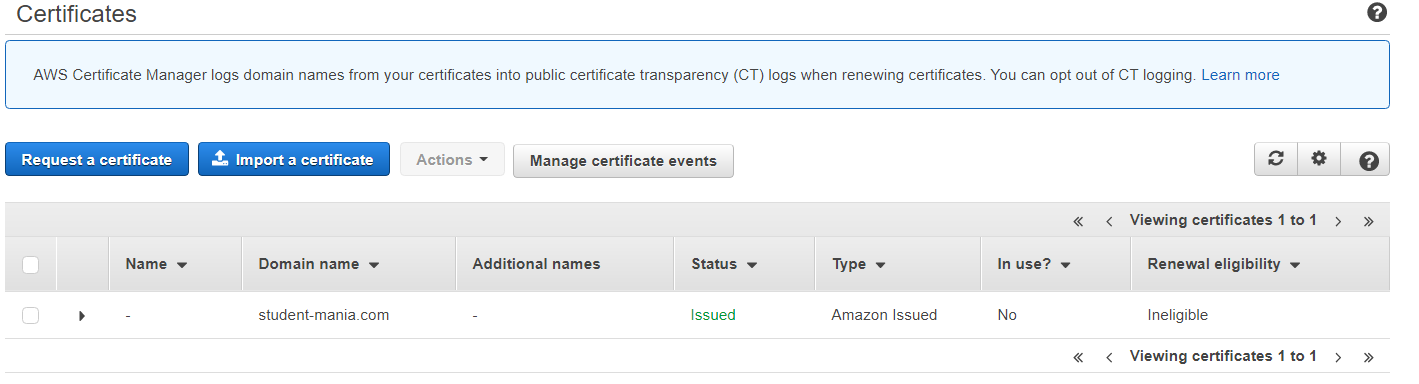
After this I went and checked what listeners I had on the load balancer. By default you can see that you have HTTP: on port 80 . As you can see below.



Step(5)

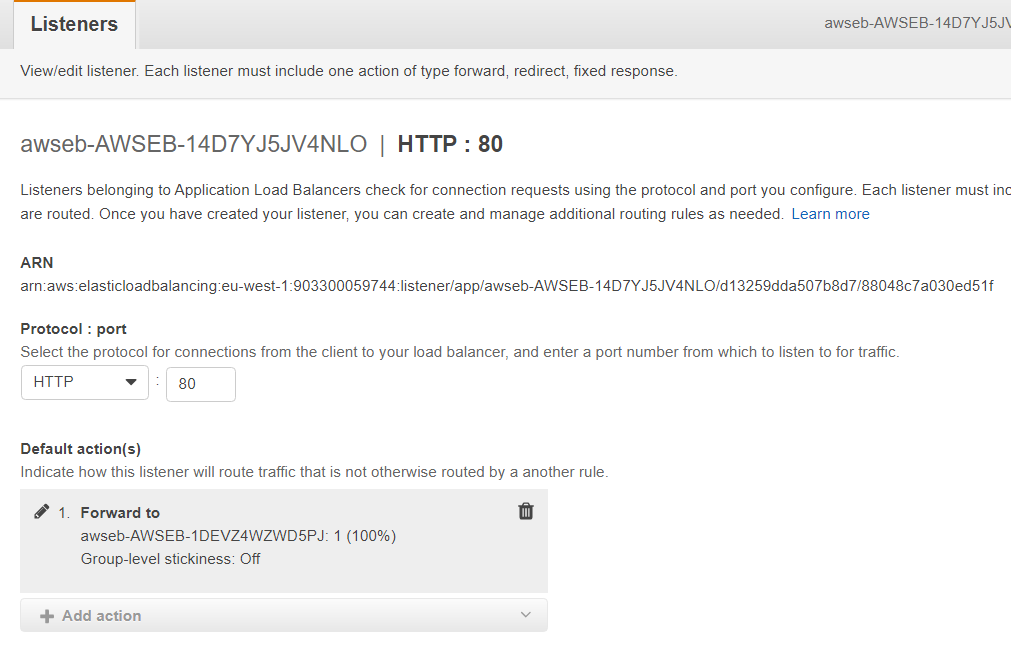
But we want to change it to HTTPS listener on port 443. As we don’t want unsupported traffic running to our servers

But first must check if it was issued yet , and as u can see it was

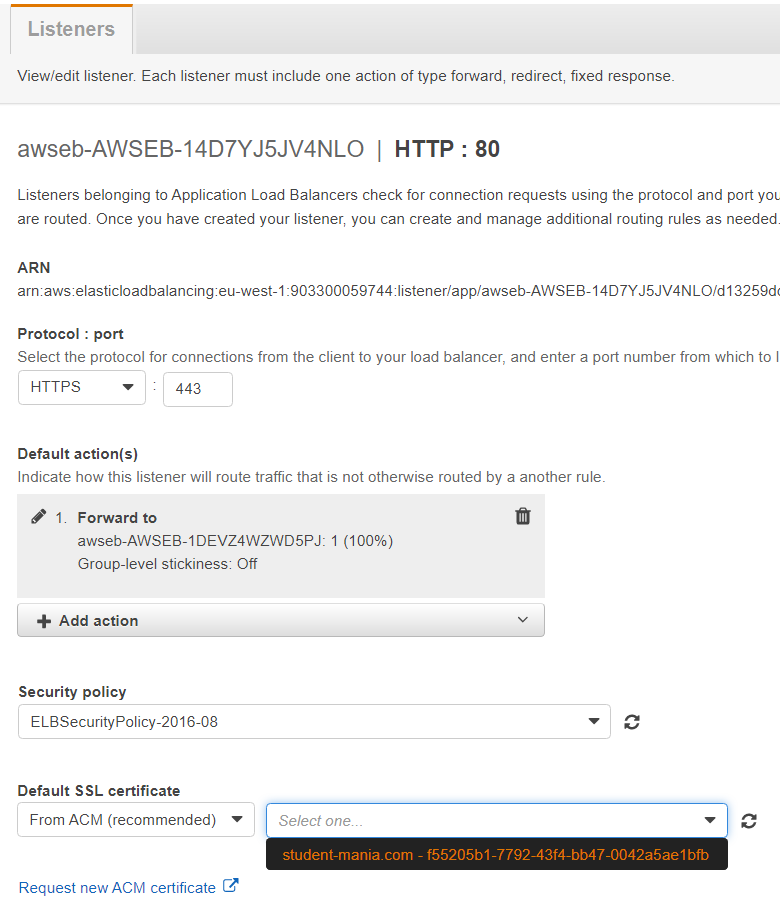


Step (6)

Change the http to Https: as seen below



From:



To: as you can see when the certificate is issued you will then be able to select the https as a listener at the bottom of the new 443 port.

* *This which will only appear when you have the CNAME added to the Route53 and certificate issued!*

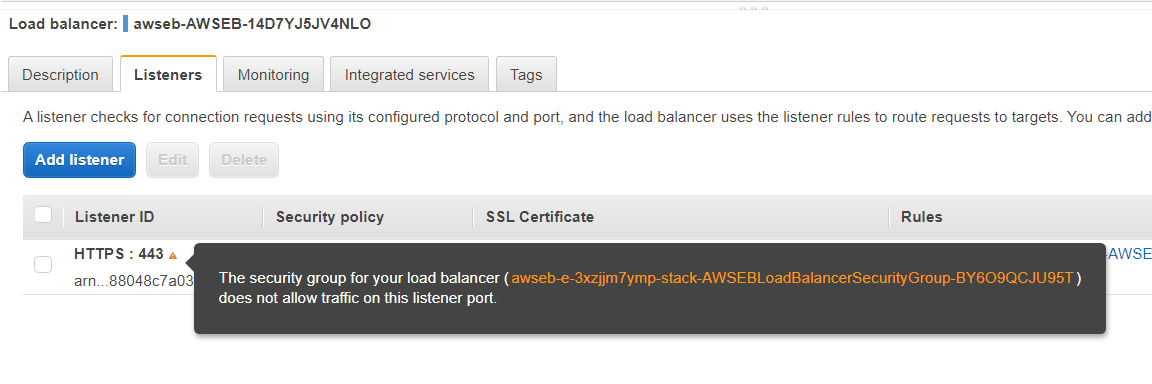
Step (7)

When you click “update” you will get the following response to confirm your changes.



Step(8)

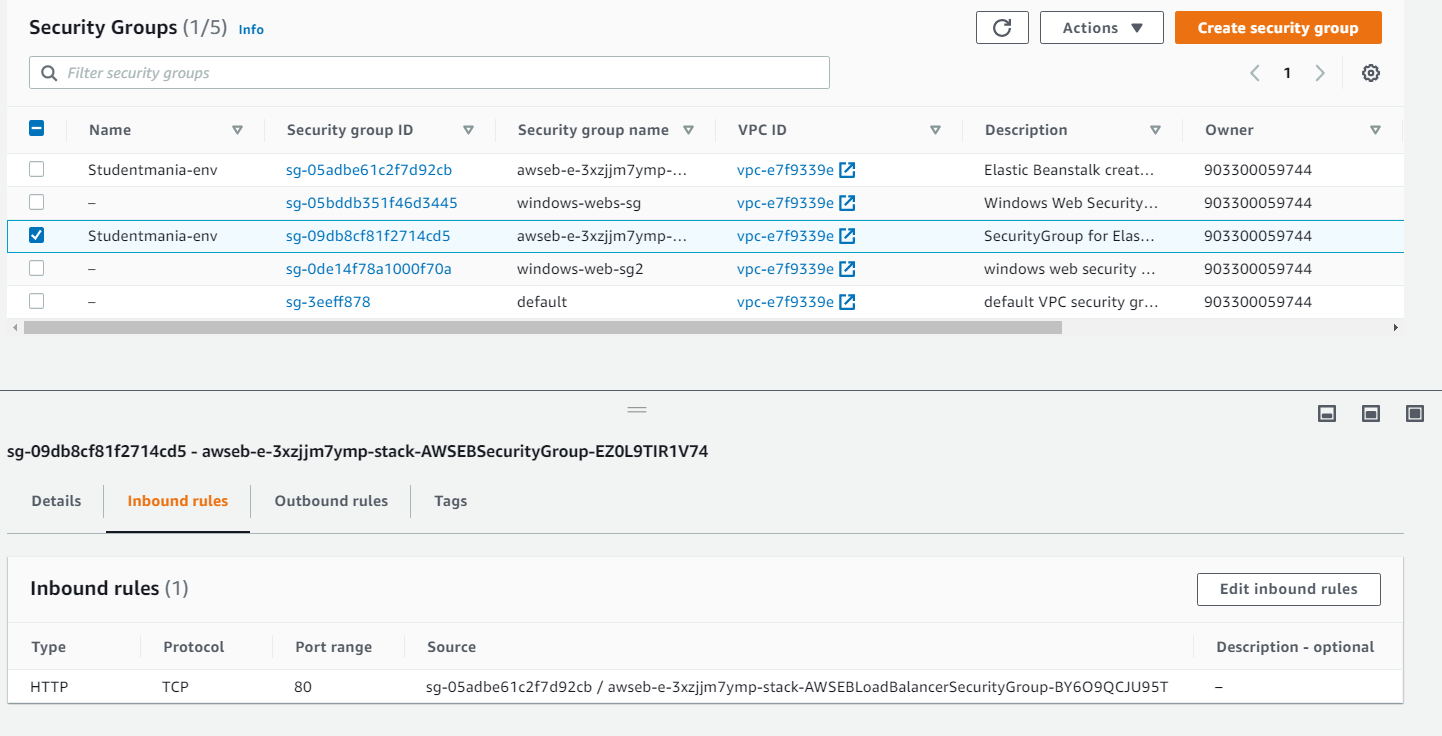
Was issued a warning after this update which told me that the port *was not* receiving traffic!.



Step (9)

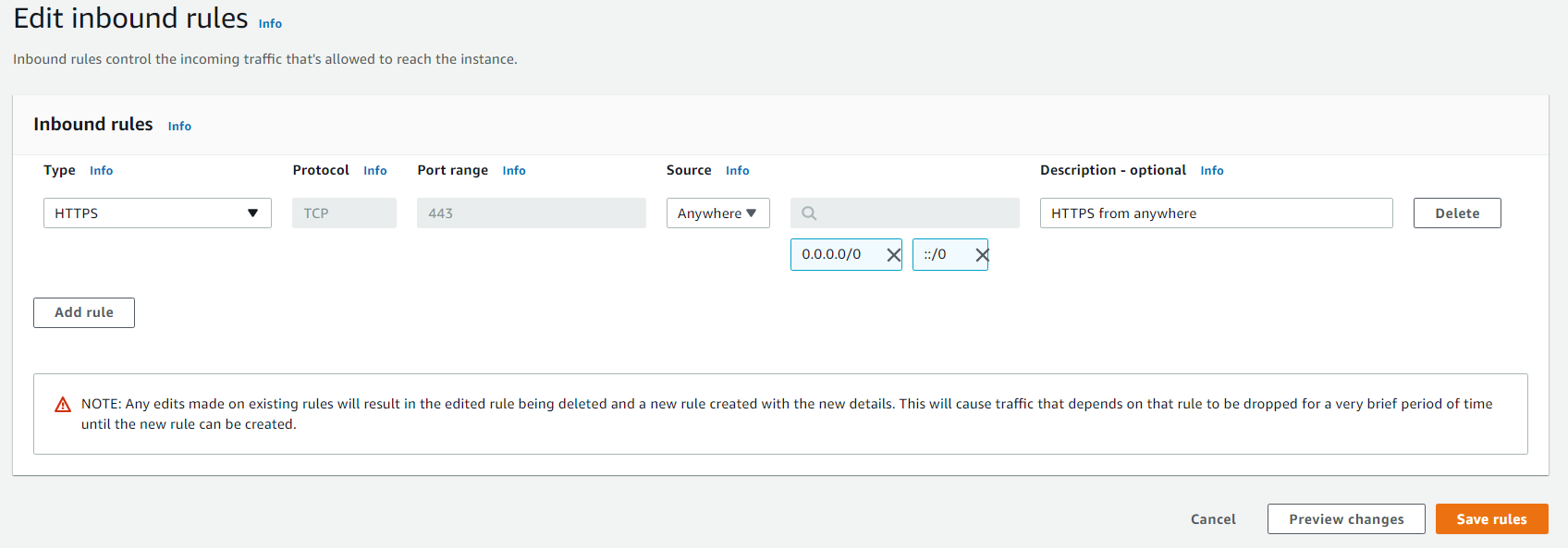
This which I had to check in the security group to see what ports it was using for access.

As you can see only port 80 was listed at the bottom of the image!



Step (10)

Changed inbound rules as seen below!

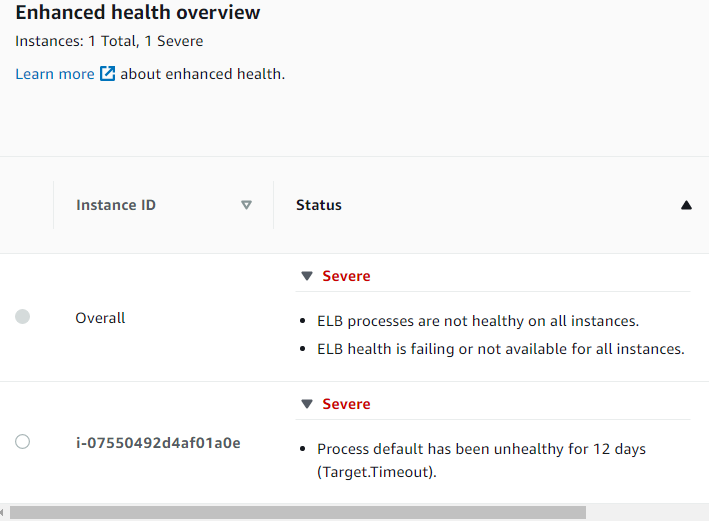


Step ( 11 )



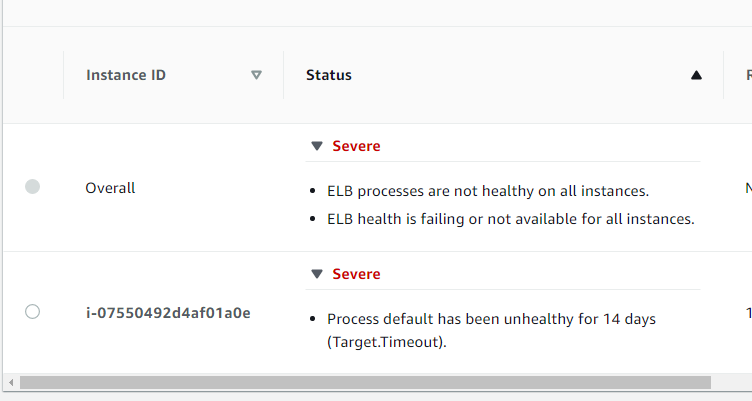
Confirmation of change and should work for website now (fixed our HTTPS Issue)

Next just to fix the following issue we have left with our database



Fixing ELB processes failing on all instances:

# Tackling the Elastic Beanstalk failed on all instance’s issue.



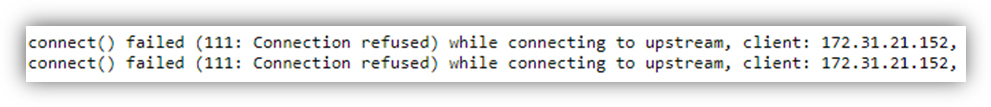
(Step1)

requested the last 100 lines of the logs so that I could see where the error is coming from



(Step 2)

Noticed it was a 111-connection upstream error.



(Step 3)

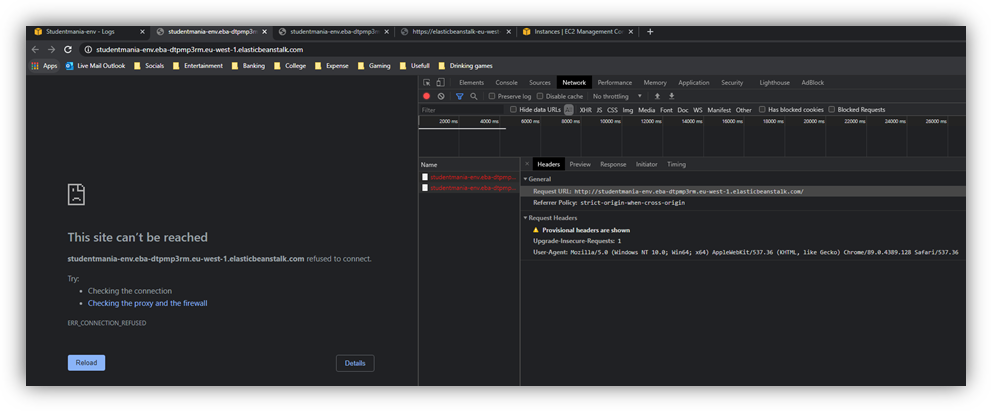
Researched the error on google and multiple forums

* Came across some people saying it was a dependency issue with their app and aws(which was a different type of application)
* Others were saying it was a connection issue between aws and the application.

Step 4)

Double checking what happens when run the environment for aws.

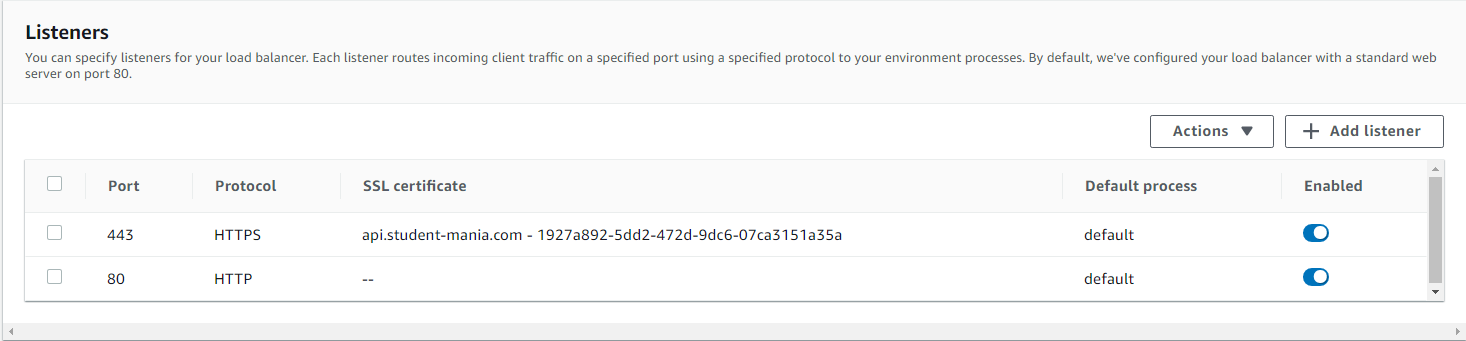
SOLUTION:



The only solution to fixing the severe health issue with our elastic beanstalk was to “Rebuild” the environment. This we tried to avoid doing as much as we could but when it came down to running out of time trying to get our database working globally to meet our deadlines with other areas of our project.

After “rebuilding” the environment and making the necessary changes with the following

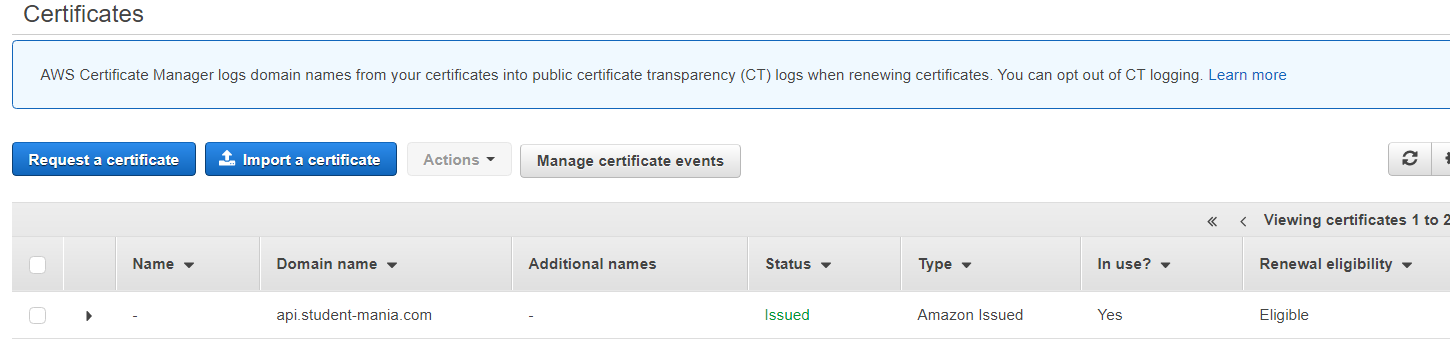
The load balancer



, route53



, and applying for the https certificate



to get it up and running again, we were delighted to present that the rebuild had worked and that elastic beanstalk had gotten confused somewhere along the way.

