|  |  |  |
| --- | --- | --- |
| Version | Author | Comment |
| 0.1 | Hayley-Belle | Created and added researched solutions, web-hosting, why amazon and why cloudflare. |
| 0.2 | Hayley-Belle | Added the already developed solutions and cost estimates, + various edits. |
| 0.3 | Hayley-Belle | Added “what is the best choice”. |
| 0.4 | Hayley-Belle | Added the versioning I should have done before. |

We have reached the conclusion that there are two different approaches we can take for a cloud-based solution for this project.

The first is a series of static web pages for all necessary components of the system. A small database, consisting of one or two tables will store data. A user will access a web page which will send a request to the server, which pulls data from the database and displays it on the webpage. The judges will also be using a web page, but instead they will be sending requests for the server to transfer data into the database.

This solution requires that data about the competition is collected, such as what each question is worth in points, the teams and students who are participating. It may also be necessary to generate login credentials for judges to keep the database secure.

Technologies used for this solution are expected to be as follows:

* + Amazon S3: Web hosting.
  + Cloudflare: Web security and optimisation.
  + Amazon RDS: Database Engine
  + Languages Used: PostgreSQL, HTML and JavaScript OR Java (run on a Tomcat servlet).

The second solution would be to develop an application that displays on the web, this application would need to have the same capabilities of the first, but it would also include ways for administrators to set up competitions. This application will take up significantly more data on the cloud server, but it will also be a dynamic way to set up competitions, as the front end will provide controls to create custom leader boards.

Technologies used for this solution are expected to be as follows:

* Amazon EC2(T2): Web hosting
* Cloudflare: Web security and optimisation.
* Amazon RDS: Database Engine
* Languages Used: PostgreSQL, HTML and Java (run on a Tomcat servlet).

**What is web hosting?**  
Web hosting can be thought of as having a high-tech computer, with a huge amount of storage space and processing power, all available through an internet browser. In reality, the web host already has all the necessary (and very powerful) hardware, they’re just letting you use it for a small price, and providing an interface for you to control it.

A web application is stored with the web host, and set up with a domain so that users can access the system. From there the web host handles all the data, requests and runs the software as it was designed. This solution is reliable, easily accessible and low-maintenance.

**Why Amazon?**

Amazon is a powerful and cost effective solution to web-hosting. It provides more computing power than any other online service and is completely free to join.

Amazon only charges for what you use, so setting up a small website will only cost cents per year.

S3 stands for Simple Storage Service. It provides an object storage to host cloud-based applications, websites, repositories and much more. It is designed to be fully scalable, boasts durability, a 99.99% uptime, easy to manage and fully integrated with a wide range of Amazon’s other web services.

EC2 provides a very similar service, but acts as a remote computer that can run software. EC2 is geared towards running applications, so the processing happens on Amazon’s side, rather than the user’s internet connection.

RDS is Amazon’s leading relational database system, it is a simple way to implement a database into an existing Amazon service and utilise it. A small database is free!

Amazon’s web services are unique because they are distributed worldwide, so users around the globe are able to access sites with improved latency. They are also unique due to their dynamic scaling – as soon as more storage or processing power is used, that is what you are charged for. Rather than most web-hosting services, which require you purchase the amount you need, and if you need more you must buy a set amount, and if you don’t use it all, well that’s just too bad because you’ve already paid for it.

All this information and more can be located at https://aws.amazon.com/

**Why Cloudflare?**

Cloudflare is like a virtual router for websites. It is a proxy server that filters malicious visitors, saves bandwidth and accelerates user connection to the website. Best of all, for a small website with low needs, Cloudflare is totally free!

Cloudflare increases the speed of a user’s connection by routing based on the user’s location, connecting them to the nearest datacentre in their location. Cloudflare also caches data on your website (temporary storage) so that when a user is loading up a page with the same images and code scripts, the data is all waiting in Cloudflare, rather than needing to be requested from the web host once again.

Cloudflare also reads a visitor’s IP to determine whether they are a threat to your website. Any detect threats are screened from the site and have no access to the site’s bandwidth.

All this information and more can be located at <https://www.cloudflare.com/>

**Is there already a solution out there?**

Absolutely, yes. There has been a demand for custom leaderboards, though not a very high one. After some research we found two suitable pre-made solutions.

AirScoreboard: <http://www.airscoreboard.com/>

An iOS exclusive app, for an affordable $0.99US. Allows users to create leaderboards, manage and share them to various social media sites. It has a few additional features like locations, languages, posters, and it has a way for ‘athletes’ to register themselves. A lot of these features are more than what is needed for the MATHEX scoreboard, and the iOS only limitation is a large barrier.

Rise: <https://www.rise.global/pages/simple>

This site offers a large variety of leaderboards, and what appears to be a commercial standard. It appears to be visually polished and is available on web, mobile or on-screen (presumably one with smart capabilities). While this is a great solution, there are no metrics to measure its reliability or security, it is not clear on the site whether they are using another web hosting service or have their own. There is almost a significantly larger cost involved, at 14.99GPB (up to 100 teams) or 29.99GBP (up to 300 teams) per month. Though it would only need to be up and running one month out of the year.

**What are the estimated costs?**

Assumptions:

* 400 audience users, each making about 50 GET requests from the database over the competition duration.
* 100 judges, each making 100 PUT/POST requests to the database over the competition duration.
* Total storage of 500Mb or less.
* Data transfer in/out at 100MB/1GB per month at most.

|  |  |  |
| --- | --- | --- |
| Solution | Monthly Cost\* | Yearly/Total Cost\* |
| Solution 1 | $0.21 | $0.37 (per year) |
| Solution 2 | $3.33 | $12.41 (per year) |
| AirScoreboard | $0.00 | $140.89 (total – for 100 devices) |
| Rise | $27.65 | $27.65 (yearly, by cancelling after 1 month) |

\*Currency converted to NZD 25/05/17 – this does not include conversion charges.

**So, what is the best choice?**

The most feasible choice here is Solution 1 – a simple set of static web pages hosted on Amazon S3. Not only is this a cost effective solution, that allows AUT to be branded alongside the AMA to encourage learning in New Zealand, it is highly achievable.

A simple site will not take an extended period to get up and running and ready for testing. This is ideal to ensure that the client gets to see an early prototype and have input on design decisions. The prototype will have plenty of time to be user-tested and presented to stakeholders part of the AMA.

It will also be very easy to build upon a simple site, to a full application in future, as the design decisions will have already been made.

Next up is between Rise and Solution 2. Both are supplying the same kind of functionality. However, Rise robs our group of the opportunity to develop anything, which we’re very keen to do. Solution 2, the Java application, will take some time to build, and may not have much time for testing. It is likely the prototype will be very basic and lacking a few features.

Solution 2 will be ideal to explore after the prototype of Solution 1 has been fully tested, it is unlikely that this will be undertaken by our group, unless outside of the Research and Development paper – the source code will always be available for anybody to build upon should they wish to.

Lastly, AirScoreboard is barely worth mentioning. It has a lot of ‘fun’ features, that are not necessary for commissioned system, and it being a paid app exclusive to iOS really puts limitations on it. The judges would each need to have an iOS device to log scores, which is a bit ridiculous.