# Task 2 Review

Throughout this project, my focus has been on developing a functional prototype that meets what the minimum viable product set out in my design documentation, while in many aspects I have succeeded in doing so, in other areas I have failed. The initial functional and non-functional requirements are as follows.

## Functional and non-functional requirements

Functional and non-functional requirements are key when considering how to approach the construction of a digital solution. In this case, GibJohn tutoring are requesting a digital presence that can be used by all its customers and staff. It is due to this that accessibility is an incredibly important consideration. The target market for this application will be learners, which contains a very large scope of potential clients in age, geographical location and ability. Functional requirements for this application must therefore include:

* Translation services.
* Functionality for text size and font alterations (for those with sight loss or conditions similar to dyslexia), with the default font being sans serif.
* Suitable alt text applied to images so screen readers can communicate content to the user.
* Simple navigation, with multiple methods of reaching desired location (to ensure inexperienced users cannot get lost on the application).
* Default colour option are to be high contrast and readable without any alteration should a colour-blind person intend to use the site.

Other such functional requirements stem from the necessities of the user needs, such as:

* Requiring multiple layouts, which can change dependent on the type of user currently logged in.
* A log in and sign-up system so users can maintain their privacy and progress. This should also include multi-factor authentication as an option for users.
* Access to teaching and learning resources for a range of subjects.
* Storage and maintenance of historic user data
* A bookings system to identify when collaborative learning events are taking place.

Non-functional requirements are more difficult to ascertain, although they, like functional requirements, can be obtained by studying the user needs. Some such requirements are:

* An intuitive layout and design (such features as drop-down menus should be preferrable to waffles, for example)
* An enjoyable user experience
* Suitable data security
* Fast and efficient code (as a necessary requirement for the gamified learning aspect of the program)
* The program should be scalable, should more courses be added and the user base grow.
* Maintenance should be simple and human interaction with the program code minimal, to prevent potentially destructive code being added.
* The system should be accessible on a wide array of mediums, such as tablets, phones, laptops and PCs, and across multiple platforms (including MacOS/iOS Windows and Linux based systems)

## What Was Delivered?

Functional requirements delivered:

* A site capable of being translated using a translator extension.
* The default font being sans serif, with the possibility of additional preferences in future iterations of the system.
* Simple navigation, with a consistent single method of navigating the site (this supports the concept of Schneiderman’s 8 golden rule, specifically the user locus of control).
* Default colours are high contrast and readable without any alteration should a colour-blind person intend to use the site.
* Multiple layouts, which can change dependent on the type of user currently logged in.
* A log in and sign-up system with the option to utilise cookies and sessions in future iterations of the system.
* Learning resources for a range of subjects found in the database (with the opportunity to include them in future iterations of the system)
* Storage and maintenance of historic user data

Non-functional requirements delivered:

* An intuitive layout and design
* An enjoyable user experience
* Fast and efficient code
* The program is scalable, more courses can be added, and the user base can grow.
* Maintenance is simple and human interaction with the program code minimal, although all code is written in a self-documenting fashion, should any alterations be necessary.
* The system should be accessible on any device with a large enough screen to prevent content overflow, with potential future iterations including failsafe to ensure device compatibility.

Graphical user interface, website

Description automatically generated

A descriptive and aesthetically pleasing home page, with suitable space for content introducing users to the system.

Text

Description automatically generatedCode that utilises the Laravel Blade templating system to efficiently produce consistent and efficient web page designs.