

A specialist engineering project

# Implement project management processes to develop a solution for an engineering project

---

Written by:

Name: Mohammed Ghalib Al-Hosni

Number: 1615

Class: 8A



### Report brief

The following report showcases the overall forethought for the entire production as well preproduction stages of the development process for a web development project involving the creation of a student-teacher grade analysis dashboard. The report is split into four different stages, with them being planning, technical specification, designing, and the written logbook.

Throughout the planning section, the report discusses the team members and their respective roles in the project, the necessary tools to begin production, any potential contingencies and constraints that need to be taken into account, and finally, the overall time scale of the project. From there, the report goes through the technical specifications of the project including the major features of the project, the hosting model and method for the website, how the website might be maintained over the span of a year, any security threats that would have to be taken into account and dealt with, while finally also discussing a potential testing plan and route that could be taken to ensure the quality of the final product.

From there, the report goes on to showcase the design concept that could be implemented into the final project while also documenting the thought behind each design and how it would affect in the user's overall experience. Finally, the report concludes with a logbook of the team's meeting minutes, which presents the overall thought process throughout the held meetings in which the team discussed varying aspects of the project.

## Table of contents

<b>Planning .....</b>	<b>3</b>
Project resources .....	4
Team members .....	4
Programming method .....	4
Required software .....	7
References .....	9
Expenditures .....	10
Potential project issues .....	11
Constraints .....	11
Contingencies .....	12
Major solutions .....	13
Allocation .....	14
Task allocation .....	14
Time allocation .....	14
<b>Features and overall scope .....</b>	<b>17</b>
Features and overall scope .....	18
Project idea .....	18
Target clients .....	18
Project objectives .....	18

Full feature list .....	19
Hosting model.....	21
Blazor hosting models .....	21
Features and limitations .....	22
Hosting model of choice .....	23
Sustainability and maintenance.....	24
Industry-known standards .....	24
Maintenance cycle .....	24
Cybersecurity .....	26
Malware attacks.....	26
Access control .....	27
Additional protection.....	27
Testing plan.....	29
Functionality testing .....	29
Security testing .....	30
Designing.....	31
Design concept .....	32
Main dashboard .....	32
Subject list.....	32
Subject specific dashboard .....	33
Design justification and thought process.....	33
Main dashboard justifications.....	33
Subject list justifications .....	34
Subject specific dashboard justifications .....	34
Logbook.....	35
Team organizations and planning meetings .....	36
(Meeting 1) Programming method and learning path.....	36
(Meeting 2) Web hosting method.....	36
(Meeting 3) Allocation of members and timescale.....	37
Website functionality meetings .....	37
(Meeting 4) Ranking system and general functionalities .....	37
(Meeting 5) Hosting model, expenditures, and maintenance routine.....	38
Risk assessment meetings .....	38
(Meeting 6) Constraints and contingencies .....	38
(Meeting 7) Cybersecurity.....	39
Design meetings .....	39
(Meeting 8) Full design meeting .....	39



# Planning



## Project resources

### Team members:

The team that has been put together in order to work on the project consists of three members, with them being Ziyad Al-Saleh, Mohammed Al-Farsi, and myself, Mohammed Al-Hosni. All of them/us having varying skills and experiences that would help in the different stages during the development process of the project.

Relative strengths & weaknesses	
Members	Strengths
Al-Hosni	Has a relatively extensive background in C# programming and has already completed on the courses to be studied. Is a fast learner and would work overtime on the project. He also has a brief background in graphical design through making posters.
Al-Saleh	Al-Saleh has a decent background in art and design and therefore would be a major help in the graphical/web design aspect of the project. He is also a fast learner and a hard worker that would work overtime to ensure the success of the project.
Al-Farsi	Al-Farsi has a decent background in drawing and design, he is also known for being composed and calm even under stress.
Members	Weaknesses
Al-Hosni	Has no experience in web development and is known for overcomplicating matters. Can sometimes get overly stressed under tight deadlines.
Al-Saleh	Has only a basic background in C# development and decent background in general programming.
Al-Farsi	Has very minimal background in C# development and programming in general.

### Programming method:

#### Dominant web development methodology:

Ever since the World Wide Web was invented back in 1989, the web development industry has seen a significant rise especially in 2004 as the first social media applications and websites started to appear as stated by (Hufford, 2022). During such period, a number of development trends and standards have risen to ensure that whatever website is being created, it will be compatible with ongoing web development browsers and technologies. Said standards come in the form of markup and programming languages that are used for creating the infrastructure of any given website, with them being the following:

**Content layer:** HTML (Hypertext Markup Language), is the code used to write and create the overall structure and layout of the page and is the core of any given website. The layer of code written using HTML is known as the content layer as it is where the barebone content of the website would be displayed, however, it would be done so in an unorganized manner without any styling to ensure that the website actually looks decent.



Figure 1 - HTML

**Presentation layer:** CSS (Cascading Stylesheet), is the styling language used in order create the presentation layer of the website as it allows for full freedom and customizability when it comes to styling the core content layer. Using CSS, a developer would be able to give any objects and items placed within the content layer styles and generally be able to alter them to ensure that they look pleasing overall by allowing developers to add animation into the content layer, change the font, color, size, position, and overall layout of anything written in the HTML layer, therefore allowing developers to fully customize the styling of their page in order to meet the client's and website's requirements.



Figure 2 - CSS

**Behavioral layer:** The final – and most crucial – layer of web development is the behavioral layer. Said layer is used to actually allow the user to interact with the website by providing functions to each of the given buttons and features within the website. That is the case since HTML and CSS combined together are only able to produce static web pages, in that the pages created are merely for displaying information and for aesthetic purposes and interactive features using HTML and CSS are rather limited. This is where the behavioral layer comes in, unlike HTML and CSS, the behavioral layer is written using a programming language and not a markup language as said layer might require a given algorithm or program to be executed while HTML and CSS – as mentioned – are simply used for displaying purposes.

The main programming language used for web development is JavaScript (JS), which is the cause due to its significant dominance in web development, which is the case due to it being the only programming language that can directly run on web browsers, in that if any other language were to be used instead, it would initially have to be converted using a special tool into JavaScript code, and only then would it be able to run on a given browser, with the reason to why JavaScript was chosen simply stemming from it being the first language to be implemented into web development. With that being said, although JavaScript is the most dominant language, there are nonetheless language alternatives for web development applications.



Figure 3 - JavaScript

**Server-side:** The server-side language or layer is the part of the website that strictly interacts with features of the website that are entirely hidden from the user. Such features might include the website's interaction with a given database in order to get the data that the website displays or making calculations or operations that might not be seen by the user such as automatically sending emails. As of right now, the most popular server-side programming language used for the backend is PHP due to it being strictly dedicated for backend programming, however, its dominance is not as major as JavaScript, meaning other than languages are being used a significant amount for backend development purposes such as Python, Java, C#, Ruby, GO, and more.



Figure 4 - PHP

## Main alternatives:

**C-sharp with Blazor:** C# is a Microsoft made programming language and is one of the more popular languages currently being used. However, as mentioned earlier, JS is the only language directly integrated into browsers, therefore simply utilizing C# without a dedicated compiler is not plausible, which is where Blazor comes in. Blazor is a C# web development framework created by Microsoft that gives developers to run and execute C# code for web development application, basically converting C# code into executable code that can be run by browsers. One of the main advantages about Blazor is that it is able to use C# code for both client side and server-side development, meaning code can be easily shared between the two development processes, this includes any libraries that might be used or simply of code would have to be copy pasted between the two ends, therefore making the overall development process much more convenient. Additionally, since Blazor is a first-part Microsoft application, it can make use of all the libraries and components that Microsoft have created for the C# programming language in C#'s original framework known as .NET and ASP.NET, which includes dedicate syntax and web development components and templates known as Razor components. On top of that, Blazorise is another component library that adds on top of Razor, with some of the components that it contains being related to charting data and data visualization, which would significantly assist in the development process of the project. Additionally, it should be noted that Blazor allows developers to write HTML and



Figure 5 - C# & Blazor



Blazor/C# code on the same file using a special syntax called “Razor” which allows programmers to alternate between HTML and Razor by adding an “@” as a prefix when using Razor.

**Python with Django:** One of the most utilized programming language in the market at the moment is Python, which is due to its easy to understand and read syntax and its relative ease to learn. Although python is generally utilized for data science and artificial intelligence applications, due to its popularity and syntax, dedicated web development frameworks have been created for the language, with the most popular of which being Django. The reason why someone might want to use python for web development other than the easier to syntax is due to the massive amount of libraries that the language has amassed for data science and machine learning, which would be especially useful in this specific usage due to data visualization being a major portion of data science, meaning dedicated libraries would be available for said purposes that would assist in easing the overall data visualization process.



**WordPress:** WordPress is an opensource application and is a method of creating websites that requires no coding whatsoever known as a CMS (Content Management System). The way that WordPress works is that the user/developer login into the WordPress website, select a given website template to follow or start from scratch, and then simply utilize the given tools in the WordPress website to write on the website, add images, style them, and basically create the entire layout of the website, with the closest comparison that could potentially be made is MS Word but for web applications as, with WordPress development being a “drag and drop” process in which components are simply added by dragging them onto the webpage. This means that the overall development process is significantly easier and much simpler in comparison to learning and having to use a programming language for the development process, and any changes that would later on be made to the WordPress website would automatically be edited in the code of the actual website. Moreover, WordPress has additional plugins that could be downloaded onto the website that provide additional functionalities such as payment gateways if the purpose of the application were to be a store, or a contact plugin that allows clients of the website to easily contact the creators, and most importantly, the charts and graphs plugins the allows developers to create interactive charts that can be easily inserted into the website.



Figure 7 - WordPress

### Method of choice:

---

Overall, the technology and programming methodology that would be used would be utilizing HTML and CSS for the frontend of the website and **C# and Blazor** for both the behavioral layer as well as the server-sided code. That is the case since both HTML and CSS are necessary aspects of the web development process that cannot be substituted, however, the reason that C# was chosen as the programming language of choice with Blazor as its framework is the following:

**Why C# and not JS or Python:** The main reason that C# was chosen over either JS and Python is simply due to the fact that most of the team members working on the project already have prior experience working with C#, and although most of them also have worked with Python, their overall experience and time spent working with the language is not to the same extent as that of C#. As such, utilizing the language for developing the website would be much easier in comparison to having to learn the syntax and error conventions of either JS or Python even though they might provide the same functionality as C# would. Moreover, if JS were to be used, two separate frameworks would have to be used, one for the frontend and one for the backend, or use JS for the frontend only and have to learn another server-sided language such as PHP, which would complicate the overall process compared to simply using a language that the team members are already familiar with. Finally, even though JS and Python and the community behind both of them might larger than that of C# and Blazor, the overall documentations written for the language and framework are both done by Microsoft, which have provided a significant amount of documentations explaining the language in its entirety and the feature as well as some of the components that Blazor has.

**Why C# and not WordPress:** Although WordPress on paper certainly seems like a much easier choice, it does nonetheless have certain disadvantages that need to be addressed. The major issue with WordPress is that the application significantly restricts developers as the majority of the development process using WordPress consists of utilizing plugins to achieve a given goal, however, in the case of an entire website, the number of plugins that would be required are of a great deal and therefore developers would fully have to rely on said plugins, and in the circumstance that a given feature is unavailable or has not been made into a plugin, the developer would simply be unable to implement the feature into their website. Moreover, as mentioned by (Mahmood, 2021), WordPress websites are susceptible to cyber threats and attacks since many of the plugins available on WordPress are open source, leaving them easy to exploit by potential attackers especially if a niche and unmonitored plugin were to be used. Additionally, due to the excessive number of plugins that would have to be used, it is very likely that the overall speed of the website would drop significantly, this could stem from either unoptimized or outdated plugins, with some of the issues simply being out of the developer's control. As such, it C# would be utilized in lieu of WordPress, however, if ever needed, the option of switching to WordPress would be considered.

## Required software:

### Necessary development software:

**Integrated Development Environment:** The Integrated Development Environment, also known as IDE, is the application or platform in which developers actually write their code in. Said IDE provides developers with multiple tools that assist them during programming processes such as a dedicated debugger for testing purposes, extension that can be installed from within the IDE to make the overall programming experience more convenient. Moreover, IDEs come with their own compiler for the programming language of choice that allows the developer to actually run their code and have it interacted with the hardware components of the computer system.

The two main competitors for a fully-fledged C# IDE are JetBrains's Rider and Visual Studio. Although Rider on paper seems better, and people using the IDE say that switching to it after initially working with Visual Studio is much better due to it compiling code faster, feeling smoother with less keyboard lag, and having less bugs. However, the Rider IDE is a paid service that costs 15 USD a month and 150 a year, and as such, Visual Studio would be the one used, which in itself is not a bad IDE, it is just the programming experience in Rider feels smoother than it does on Visual Studio. It should also be noted that the Visual Studio version that would be downloaded is Visual Studio Community, which contains fewer features compared to VS professional and VS enterprise such as the lack of a wide array of testing and debugging tools, however, the Visual Studio Community edition comes entirely free of charge.



Figure 8 - Visual Studio

**Blazor:** Blazor is – as mentioned – the framework that would be utilized for the libraries and components needed to make the project using the C# programming language. Downloading the framework is simply a process of launching the Visual Studio installer and downloading the Blazor app workload. Blazor was explained earlier in depth in the programming methodology section, and as such it would not be talked about here again as to avoid repetition.

**Web server:** A web server is a necessary requirement for deploying any given website/app and hosting it onto the World Wide Web for anyone to access. Although there are a variety of different domain and web hosting service providers such as HostGator, GoDaddy, DigitalOcean, and Znetlive. However, the one that is specifically going to be used is Microsoft Azure's App Service hosting. Azure is a web hosting platform developed by Microsoft is known for being one of the best hosting platforms on the market, with its great reliability and seamless connectivity between varying services such as hosting, varying SQL databases, storages, analytical tools, and more. Moreover, due to Azure being a Microsoft developed product, it already has dedicated documentation to hosting the Blazor WASM application on Azure App Service.



Figure 9 - Azure



However, a major problem with Azure is that although their service was considered a premium, it also costed a premium, with their App Service hosting costing around \$60 a month for a nearby UAE server. With that being said, Azure introduced a free plan, in which for the first 12 months of a person's subscription, they are able to use any of their services for free as long as they do not go over a given limit (i.e: do not utilize over 1gb of storage in their SQL database). Additionally, some services are given for free indefinitely, with one of those services being Azure App Service, the hosting service required by the team to host their application. As such, Azure App Service would be the hosting service to be used.

**MSSQL and its management studio:** SQL – also called Structured Query Language – is a type of database language that allows developers to handle information within any given database. Said language would be necessary for the project as a database would be needed in order to extract student names, grades, class, teachers, and more. However, SQL itself is the language used for working with databases and is not the database itself. The main two database services out there are MySQL and MSSQL (Microsoft SQL), however, both MSSQL and C# are made by Microsoft, and as such MSSQL would be utilized due to more specific documentations and greater compatibility.



Figure 10 - MSSQL

However, an MSSQL database would need to initially be rented or purchased, although a solution to this is to utilize Azure's SQL database. As mentioned earlier, Azure is offering their services for free for the first year of the subscription, which would be perfect in this situation to initially test out the service. Not to mention that since both Azure and MSSQL have been developed by Microsoft, a large number of documentations have been written on how to utilize the service exactly and the setup process of the database in accordance to the web application which would be hosted on Azure App Service. As such, the MSSQL database server would be hosted through Microsoft Azure.

**Domain:** A problem with Azure is that as of 2022, it does not offer domain registration, meaning that in order to actually provide a name to the website, a domain would have to be rented from a third-party provider. There are numerous domain register service providers, however, the biggest names in the market are Namecheap, Google Domains, GoDaddy, Bluehost, and HostGator. The domain registration process is rather a simple one and – although essential – is not something that requires extreme care when it comes to the provider, and – to certain extent – the choice could be based on arbitrary reasons. As such, the provider that was chosen was Namecheap, mainly due to it having better reviews than some of its competition while also providing domain registration at a competitive price point.

### Additional software:

**Project manager (Notion):** Using a project management application software is an essential aspect in ensuring that the overall development process is organized, steady, and that every team member is on board with the ongoing task. Said management application allows for user to interact with one another and edit a large database together which contains the team's notes, development timeline, client requirements, database details, learning resources and reference, and more. The application that would be specifically utilized would be Notion, it is a free management application that allows multiple users to join a single group called a "Teamspace", in which, users can create blocks in which notes can be written that can be altered and changed by authorized members of the team in order to ensure that the information is consistently being updated and kept up to date. Moreover, Notion provides easy to use components that organize the overall look of the documented notes while also providing additional components that significantly assist in ensuring that a single management environment is sufficient for everything by having a built-in Wiki and also supporting embedded links to GitHub, Figma, and other software development software applications that might be utilized. Overall, Notion provides the team with a space in which they can submit and take notes of all their tasks over an organized multi-view system which includes integrated file embeds to additional development and management applications.



Figure 11 - Notion

**Development local server:** Being able to test the website as it is running is necessary in ensuring that the code being written is working exactly as intended. As such, a local development server would have to be utilized in order to locally test the website. The specific development server that would be used is called the “Live Server” extension found in VS Code. Said extension allows for the developers to easily run their websites on a local server created within their computer through the click of a button, allowing them check in real time as the code is being edited whether or not the web pages being designed look as required.

**Web browser with element inspect:** Element inspect is an integrated feature found in certain browsers that allows the developer to check the frontend code. Said feature allows developers to inspect objects within any given web page such as images, text blocks, forms, lists, and so on. This feature is necessary for any given web developers as it allows them to check their own web application and ensure that if an error were to occur, they can directly check the elements on the web page and check what causes a given error. Additionally, since element inspect could be used on any web page, this means that if developers were to find a feature on another website that they would want to implement on their own, they would be able to inspect the frontend code of that web page to try to and implement something similar. Although most modern browsers nowadays have an element inspect feature, the browser that would be used is Opera simply due to personal preference.

## References:

---

### People:

---

**Mr. Waleed Al-Harthi:** Mr. Waleed Al-Harthi is the COO of Rihal, a web development company. He is one of the people would occasionally assist the team working on the project from time to time whenever it is needed. Due to his extensive experience in the software development field, his assistance is of great help and can be trusted when it comes to any form of assistance and consultation that he might provide.

**Mr. Abdulaziz Al-Jabri:** Mr. Abdulaziz Al-Jabri is the person supervising the entire project. His experience in the industry might not be very extensive, however, he certainly has more experience and knowledge about the subject than the team working on the project, and as such whenever needed, he can be relied on as source for information when issues occur while working on the project.

**Mr. Ahmed Al-Saleh:** Mr. Ahmed Al-Saleh is the head of IT within RGOTC and has a great experience within the field as well as the database that the school utilizes that stores student data and information including their names, candidate number, grades, and so on. As such, he would be of great assistance if the team ever needed information related to the school's database or even working with databases in general as well general information about how the team could incorporate some cybersecurity techniques in order to ensure that the website is safe overall.

### Internet resources:

---

**Appropriate subreddits:** Reddit is a platform in which anyone could submit posts and various questions into communities called “subreddits” that are dedicated towards one subject. This is extremely helpful when looking for answers or being unable to find any having to rely on asking someone on the internet. The way that reddit works is that the user would post their question within a given subreddit, and people could see that question and potentially answer it if they have relevant information to the posted question. Some subreddits that might be appropriate for use would be [r/learnprogramming](#) – [r/webdev](#) – [r/webdesign](#) – [r/csharp](#) – [r/blazor](#)

**Appropriate discord servers:** Discord is a messaging platform in which individuals could join groups dedicated towards a given topic, with said groups being known as “servers”. Multiple people within said groups potentially having some background and knowledge about said topic and as such could be asked when any assistance is needed. Some servers that could be of use to join are “C#” and “Web dev and web design”.

**Stack Overflow:** Stack overflow is very similar to reddit in that users post questions on a community billboard and anyone with relevant experience could reply to the post with an answer to the posted question. However, unlike reddit, stack overflow is entirely dedicated towards programming, and as such the replies found within it are generally more sophisticated and in depth compared to the ones found in reddit. Additionally, the programming community within stack overflow is also generally much larger than reddit and people with much greater experience in the field use the application. As such, it would be a great application to keep in mind when researching or asking for a given problem. Not to mention that solutions can also be found from prior posts that have been answered with similar issues, which would be a much faster process than creating a new post and waiting for someone to answer, which would generally be done only when no solution is found.



Figure 12 - StackOverflow

**W3School:** W3School is one of the world's leading programming language learning platforms that provides developers with interactive codes about various solutions on almost all programming languages. In that if a certain problem was faced related to the syntax of the language, W3School provides explanation as well as example codes with an online compiler than could be used to test and exactly understand how the code works, which can be of great assistance during the learning process or even when a team members forgets a particular aspect of the code or language.

**Microsoft documentations:** Since C# and Blazor are both going to be utilized for the project, both of which being Microsoft products, as such, the Microsoft documentations are an extremely valuable resource that would provide in depth explanation on how both the C# programming language as well as how Blazor components and Razor syntax works.

## Courses:

**Programming in Blazor by Filipe Gavilàn:** The programming in Blazor course by Filipe Gavilàn is the course that would teach the team the actual web development aspect that would be heavily necessary for developing the actual dashboard. The course goes through the Razor syntax and how to use Blazor in order to create dynamic single page applications with the use of C# and the Blazor framework, the course is 16.5 hours in length, and has a rating of 4.6 stars averages over 2000 reviews. However, in order to actually make use of the course, some prerequisites would have to initially be met so that the contents of the course actually make sense to the viewer, with said prerequisites being decent/intermediate knowledge of the C# programming language and knowing the basics in both HTML and CSS.

**C# Masterclass by Denis Panjuta:** The C# masterclass course hosted by Denis Panjuta goes through both beginner and intermediate level C# topics that include the language's basic syntax to classes, polymorphism, delegates, and additional topics such as game development and application UI creation using Windows Presentation Foundation (WPF). Overall, the course should get the team up to speed with C# in order to follow the Blazor course. The course is has 4.5 star rating over 22,000 reviews and is around 37 hours in total, however, a lot of that is from the WPF and Game development section of the course which can be ignored for the purposes of the project.

**The Ultimate HTML5 & CSS3 Series by Mosh Hamedani:** The HTML5 and CSS3 course created Mosh Hamedani is would be able to provide the team with basic, advanced, and intermediate knowledge of HTML and CSS, which would not only assist in understanding the Blazor course, but also help in creating better user interface by combining Blazor/C# with HTML and CSS.

## Expenditures:

Capital expenditures	
Item/service	Cost
Programming with Blazor by Filipe Gavilàn	\$12.99 + \$0.95 VAT (purchased while on sale)
C# masterclass by Denis Panjuta	\$19.99 + \$0.95 VAT (purchased while on sale)
HTML/CSS course by Mosh Hamedani	\$29 (purchased while on sale)
Domain name (1 year)	\$9.58
<b>Total</b>	<b>\$73.46 – 28.28OMR</b>

## Potential contingencies

### Constraints:

#### Scope is too difficult:

**Risk assessment:** One the main problems with the scope of the project is that it could potentially be overly ambitious, meaning that the overall features required for the product are significant and somewhat difficult. Another major problem that raises the point of the previous problem is that the work that is required to be done by the team members is entirely new to them, meaning a lot of ideas and concepts would have to be learnt from scratch, overall, this issue could possibly lead to an incomplete or subpar – but operational – website. With that being said, the team members should be able to learn any new topics with relative ease considering they have a decent background in programming in general, meaning that although web development itself might be a new topic, the overall programming and software development workflow is going to be somewhat familiar, which would certainly ease the learning process as the basics could be skipped.

Probability vs impact table				
Expected impact on the project	Very likely	Medium	High	Extreme
	Likely	Low	Medium	High
	Unlikely	Low	Low	Medium
		Minor	Moderate	Major
Expected impact on the project				

**Mitigation plan:** One possible solution that could be utilized to ensure that the problem is not even faced in the first place is to prioritize the tasks and features to be implement in a planned and thought-out manner as to ensure that the team is entirely aware of the scope and in which order they should be working on. However, in the circumstance that the issue if faced and the scope of the project seems too difficult to complete, a decrease in the number of features should be done in based on the aforementioned periodization plan, with the ones with least priority being removed or at least substituted ones that might be easier to implement and leaving the original ideas as ones that could be implemented further down in the project's life cycle as convenient updates.

#### Budget:

**Risk assessment:** Most of the resources that are going to be utilized have already been planned as seen in previous sections, and all of which have been within budget. Additionally, due to the fact that the planning done was thorough, it is unlikely that something necessary to the development process would be needed further down the line. However, if the budget were to be exceeded, some sacrifices to the overall quality of the project could potentially need to made in.

Probability vs impact table				
Expected impact on the project	Very likely	Medium	High	Extreme
	Likely	Low	Medium	High
	Unlikely	Low	Low	Medium
		Minor	Moderate	Major
Expected impact on the project				

**Mitigation plan:** To avoid ever getting into a situation where budget might be an issue, the planning done for the resources and expenditures was thorough as stated earlier. Meaning that a lot of possibilities were considered before concluding the final capital expenditure. However, if a circumstance where budget becomes an issue does occur, then

possible methods of mitigating or at least decreasing the damage caused is to change the domain's subscription plan to one that is cheaper or trying to refund the courses bought if possible.

### Time limit:

**Risk assessment:** The given time limit is possibly the biggest risk and constraint in the entire project. That is the case due to the team needing to learn a lot of new concepts and topics, which – although relatively easy – could alone take a significant amount of time. Moreover, the overall list of functions that need to be implemented is rather big, meaning in addition to having to learn a lot of new concepts, a lot of new features would have to be implemented which could potentially come with their own issues that would have to be researched and solved, taking a lot of time during the process. Finally, in the circumstance that the project was not completed, the website would simply be useless, and would not even be functional, compared to the scope constraints risk, in which although the website might be subpar, it still would be functional.

Probability vs impact table				
Expected impact on the project	Very likely	Medium	High	Extreme
	Likely	Low	Medium	High
	Unlikely	Low	Low	Medium
		Minor	Moderate	Major
Expected impact on the project				

**Mitigation plan:** One of the ways in which this issue could be mitigated is to plan/predict the entire timeline of the project using realistic times with appropriate tolerances in order to have an idea of whether or not it is possible to finish the project in the given amount of time. Additionally, as mentioned earlier, prioritization of tasks has to be done as to know which tasks to try and finish as to make sure that the site is at least functional even if not fully finished. Moreover, although not ideal, the team should be prepared to work overtime on the project and to reschedule deadlines to a closer date to make sure the website is functional and releasable by the final due date.

## Contingencies:

### Power outages:

**Risk assessment:** Power outages are a rare yet extremely dangerous occurrence. The reason to which is that during a sudden power outage, any storage drive such as SSDs and HDDs being utilized within databases could potentially be damaged, potentially leading to them losing all the data found within them. With that being said, as mentioned earlier, the database that is going to be utilized is the one hosted by the domain provider, meaning the database being used are very likely to have backup power sources in the cases of a power outage which would continue to supply power to the database in such circumstances, therefore preventing any damage from occurring to the data stored within them.

Probability vs impact table				
Expected impact on the project	Very likely	Medium	High	Extreme
	Likely	Low	Medium	High
	Unlikely	Low	Low	Medium
		Minor	Moderate	Major
Expected impact on the project				

**Mitigation plan:** Due to the already in place security, there are not many additions that could be implemented to significantly increase security from power outages, however, one thing that could be done is to simply have the data on the database backed up somewhere else in the case that the backup generators do not work and the data is actually lost from the data, in which case, having a backup would be extremely important.

### Loss of a team member:

**Risk assessment:** Losing a team member is can occur due to a variety of ways, with the most likely reason being illness which would prevent them from being able to work on the project, other reasons could also cause this such as family related issues or other more important matters that they might have to attend to before the project. Overall, if this were to happen, it could cause extreme problems to the overall workflow of the project, considering the team is already rather small and consists of only 3 members, so losing one of them would be a huge inconvenience and could potentially cause major hinderances and delays during the production process, consequently amplifying the fact that the time limit is already rather tight. Although it cannot be predicted if a team member would have to take a leave of absence, it is nonetheless something that is not very likely to occur.

Probability vs impact table				
Expected impact on the project	Very likely	Medium	High	Extreme
	Likely	Low	Medium	High
	Unlikely	Low	Low	Medium
		Minor	Moderate	Major
Expected impact on the project				

**Mitigation plan:** A way that the team could be prepared for such unexpected circumstances is to have a tolerances in the timeline of the project, in that project tasks do not start immediately right after one has been completed, but rather a given tolerance is put in place in order to ensure that if any unexpected circumstance were to occur that cause delays, deadlines would not be exceeded easily due to the aforementioned tolerances.

### Major solutions:

The following are some of the major solutions that would have to be implemented in the circumstance that the website could not be created whatsoever due to certain reasons such as the aforementioned ones.

**Using non-visualized data:** Utilizing non-visualized data would be a much simpler process that would be easier and faster to develop than visualized data, but it would in turn also remove the main premise of the website which is to create easily readable data to be analyzed. However, if major problems were to occur during the development process, taking this route be better than not finishing the project whatsoever as visualized data could still be added as an update to the website later on during the life cycle of the website.

**Using WordPress:** As mentioned earlier, WordPress allows developers to add features onto their webpage using a simple and rather convenient drag and drop method. Said method would decrease the overall time that would have to be spent in making the webpage and also in learning the actual programming process, which would consequently make the overall development process much easier and less time restrictive. With that being said, as mentioned earlier, WordPress is much more restrictive and does not allow for fully fledged customizability while also sacrificing security, which would be out of the developer's control due to the opensource nature of the plugins utilized by WordPress. Not to mention the fact that using a large number of plugins would lead to the overall website being rather slow. As such, using WordPress is not an ideal solutions compared to the C# and Blazor method, however, it would be much easier and fast to develop the website using WordPress in the circumstance that the initial method was too difficult given the available constraints.



# Allocation

## Task allocation:

---

### Al-Hosni:

---

Al-Hosni (myself) would be focusing and working on most parts of the project, starting with the frontend and also developing the backend of the website. The frontend tasks are going to revolve around the login page and the dashboard, while the backend would be specifically the graphing algorithm as well as extracting from the database displaying any general information such as name, class, number, and also categorizing students into their classes for teachers to be able to see the averages for the classes they teach, this would be lead by myself with Al-Saleh also working on it due to its complexity. Additionally, if possible, within the given time limit, also work on linking the database and SQL server, moreover, I would be working under but alongside Al-Saleh during the designing and wireframe drawing stage of the project. That is the case since I have the most experience in programming in general, and as I should be allocated with parts of the development process that require a lot of programming, additionally, I would be assisting Al-Saleh in the designing stage due to my brief experience in graphical design.

### Al-Saleh:

---

Al-Saleh has been allocated with working on the frontend as well as the backend. In terms of the frontend, he will be working on the main page of the website as well any additional pages or addons that would have to be made, while on the other hand, his work on the backend focusing on the SQL connection and creating the ranking system, additionally – as stated – he would also be working categorizing students by their classes in order for teachers to be able to view the data related to the classes they teach. Moreover, his frontend work not only involves creating the user interface, but also leading the wireframe designing stage of the project, as such his overall involvement in the project will be most influential on the frontend. Said tasks have been assigned to Al-Saleh due to his decent programming skills as well as great creative ideas and designs.

### Al-Farsi:

---

Al-Farsi's programming related tasks would be concentrated on the frontend, with specifically working on the creation of the student and teacher's classes list, while potentially slightly working on the backend in creating the search feature for the subjects or classes. Additionally, Al-Farsi's work would also revolve around data gathering and entry, with him leading the meeting and data entry stages of the project, in that he would be the one heading any meeting that would have to be conducted with the supervisor, Head of IT, or head of student affairs, while also making sure that a sufficient amount of data is entered into the database before the website is deployed.

## Time allocation:

---

### Critical path analysis:

---

**What is the critical path analysis:** The critical path refers to the sequence of tasks that would have to be completed in a given project which form the estimated total duration until the final product has been fully developed and completed. The purpose of analyzing said critical path is to know exactly if finishing the project in the given amount of time is feasible or not. This would in turn assist in any reorganizations of the plan and path that would have to be made in order to ensure that the project would indeed be completed within the given time limit while also taking into account possible circumstances that could lead to potential delays, and from there additionally adding tolerances to ensure that if something were to be delayed, the total duration of the project is not moved forward closer towards the deadline.

**Learning:** Before starting to work on the project, the first thing that would have to be done is for the team to actually learn how they are going to make the website. This includes learning HTML/CSS, C#, and Blazor development. HTML/CSS are relatively easy languages to learn, and therefore only seven days would be required for them, with a tolerance of three days in the case that additional practice was required. From there, due to the extend of the C# course, around three weeks might be required, with a

tolerance of three days if something was unclear during development. Finally, even though the Blazor course is shorter than the C# one, more practice would have to be done and therefore a minimum of three weeks are given with a tolerance of an entire week due to the unpredictability of the learning process.

**Meetings:** The meetings process involves talking with the people that could provide some consultation or idea that might be great to implement before the starting to actually work on the project. This would be done at the final days of the learning stage as to make sure that the team can go into the development stage with everything they need. The task allocation meeting is simply between the team, which would involve talking about how exactly the team would start working on the project, since this is between the team members, it is very unlikely that it would be delayed, as such only a single meeting would be needed. However, the other three meetings do have the potential of extending to two different meetings over two separate days as such a tolerance of one day is given.

**Frontend:** The front end is the major and possibly more difficult part of the project, with it consisting of the largest workload that the team would have to finish. The wireframe design would have to go through multiple iterations until both the team and supervisor are fully satisfied with the design of all the pages to be made, due to the multiple iterations, it might take anywhere from three to five days as the drawing process itself is not necessarily hard nor time consuming, but the process of generating different ideas is. From there, the first part of the website that would be made is the login page due to the relative ease of making one, and so it would only take a single day and a maximum of two days in case something went wrong as it will be the team's first time practically working on the real project. The main page on the other hand is a major aspect of the website, and it could potentially include some advanced CSS that would have to be implemented including stuff such as scroll animations, as such, around a week might be required to make it with a tolerance of two days just to be on the safe side.

After the main page, the major aspect of the project would start, which is working on the student dashboard, due to this being the critical part of the project, it would need to be perfected, meaning the possibility of multiple iterations and redos is possible, not to mention that it will likely contain a lot of feature which would also increase the amount of time required for it. As such, around three weeks might be required with a tolerance of an entire week just to be safe. From there, the class dashboard would be made, generally, the class dashboard would most likely be very similar to the student dashboard with only slight changes to include the students, however, it would still probably take a decent amount of time to reshape the student dashboard page effectively without breaking anything else, therefore it would potentially take three days but a tolerance of two days has been added in case something does not pan out as expected.

Finally, the last two sections are simply the student list and class list, with the student list displaying all the students available on the database, and the class list displaying all classes with their respective teachers and the students within said classes. This process is relatively simple and involves basic SQL to extract the relevant data, as such, it should not take a lot of time, so around two to four days for each should be reasonable.

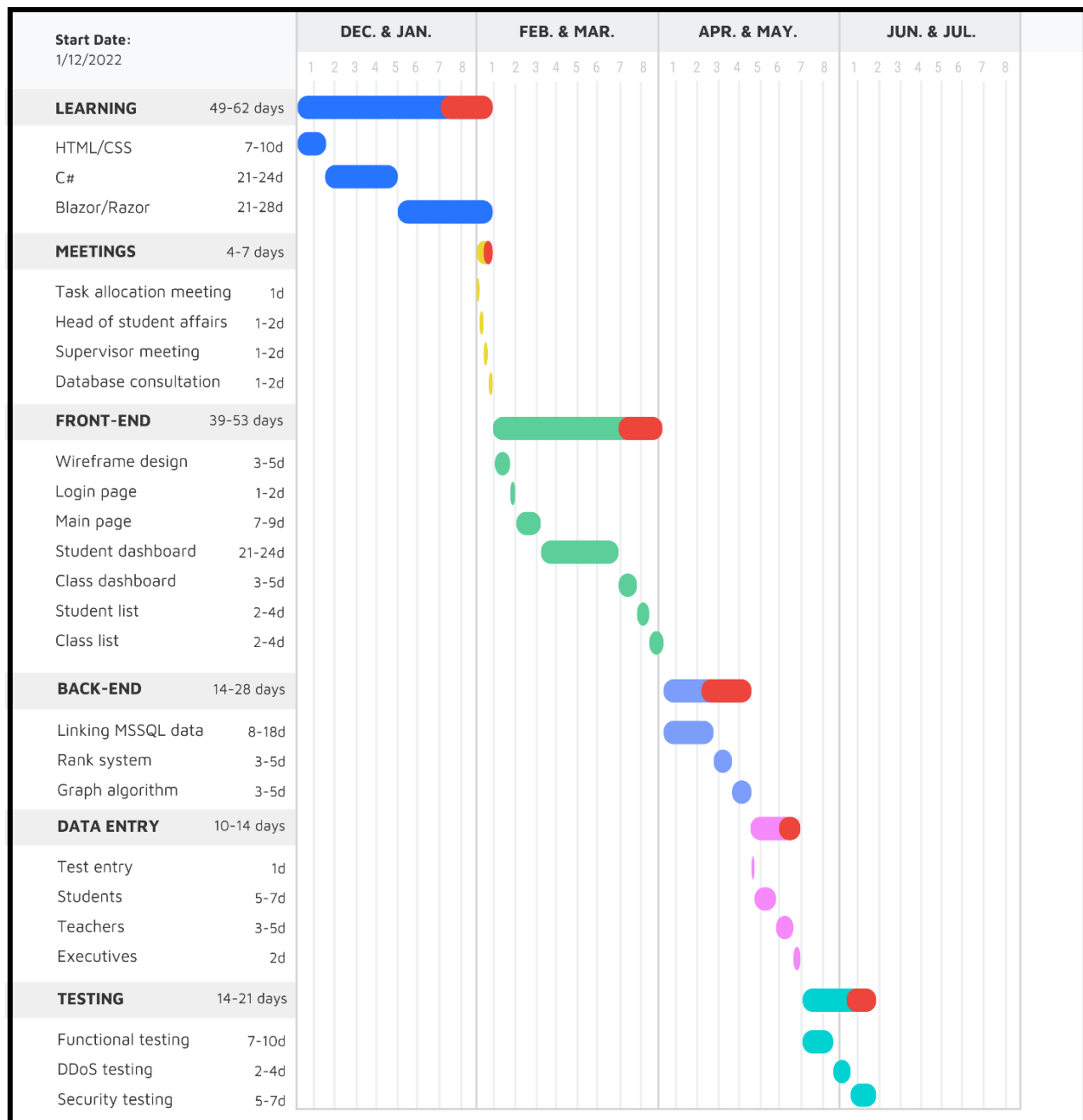
**Backend:** The main stage of the backend is linking the MSSQL database with the actual frontend in order to extract information from the DB so that they are effectively displayed to the user. This would be using SQL commands, which are relatively easy, however, the main issue stems from the fact that this process would have to be done for a large number of pages, in addition to the fact that a small error could lead to major issues in the displayed information, as such several verifications would have to be done to ensure that the information displayed is indeed correct. As such, around two to four weeks might be needed for this stage. From there, adding the ranking system to separate student, teacher, and admin accounts would have to be made, this stage is rather difficult as it could potentially lead to changes in the frontend as well, which would take additional amount of time, however, the major part of the creating this system is linking the database, and since linking the DB is done before this, the ranking system might not a lot of time and would only be around three to five days. Finally, adding the graphing algorithm is a necessary stage to visualize the data, this process would be relatively straightforward as the team would only need to use certain blazor components to make it work, however, it nonetheless does require some effort to implement appropriately, as such it would also take around three days to complete with two days of tolerance to ensure that the components being utilized is understood effectively before the team starts using it.

**Data entry:** The data entry process involves simply inputting data into the database table. The first that would be done is a simply test entry that involves a couple of students, teachers, and executives, this process is rather simple and would not take more than a single day. From there, inputting student, teacher, and admin account data would have to be done. The number of students are larger and each one has a greater amount of information that would have to be inputted such as grades, and therefore potentially an entire week might be required for this. Teachers have less information but are still more than admins,

and as such around two to four days might be needed. Finally, the number of admin accounts is very little, and consequently only two days might be needed to input the data for them.

**Testing:** Last path for the project is the testing stage, this involves initially testing the functionality of the website, and during this, it is very unlikely that the website would be entirely meaning multiple fixes would have to be made, therefore this could potentially take up to ten days to complete. From there, additional destructive and non-destructive tests would have to be made to ensure that the website is safe and sustainable, with said tests including an experimental DDoS attack to check how the hosting service would react, this might take around two to four days to setup due to issues in actually finding someone that would get this service and then try to implement a fix for the issue if the routed data required for an effective attack seemed to be too little. From there, a security check for the measures put in place would have to be done in order to ensure that said measures are working accordingly, this could around five to seven days to do especially if issues with security were to be noticed.

### Ganntt chart: (red = tolerance)





# Technical specifications



## Features and overall scope

### Project idea:

---

The general idea behind the project is to have a running website dashboard that could be accessed by any permitted individual. Said website would allow said individuals to easily look up and analyze varying types of data through an organized graphical interface by using data visualization methods for analysis, in that the data would be displayed using graphs and charts that would assist in making the overall analysis process easier for the user. While in addition also providing further information about the given dataset that would be automatically calculated such as varying averages and recommendations for potential improvement.

The website would handle data by using a database that would have the necessary data and through that, that it would then plot data into a visualized format for analysis purposes.

### Target clients:

---

The overall idea could be implemented throughout a number of business and companies, however, the project would be specifically designed and made to accommodate for the lack of grade analysis throughout schools, and would be specifically implemented in the Royal Guard of Oman's Technical College (RGOTC). The main people kept in mind when designing and creating the website would be the following:

**Students:** Students are the reason that the idea exists in the first place, as the data to be displayed and analyzed using the dashboard would hold student grades. Using said grades, students should be able to easily understand their weak points and where they should focus their effort in order to improve their overall grade. This could also act as an easy way for students to set goals and motivate themselves by setting challenges to improve their graphs by a given percentage.

**Teachers:** Teachers are a major part of the school and overall teaching system and therefore the website should be dedicated towards them as much as it is towards students. The way this would be done is that teachers would have admin privileges and would be able to all the students that they teach and have dedicated grade statistics per class that they teach, allowing them to not only evaluate their students but also their own teaching methods and processes.

**Student affairs executives:** In addition to students and teachers, student affair executives should have specialized privileges within the app. And the reason to which is that they are the individuals that have direct contact with the student's parents, and so in the occurrence that a parent access for the data on their students, student affair employees would be able to provide such information. Additionally, they might also want to filter out students based on certain subjects and skills as to choose students gifted in a particular field for a given competition or event that the school might want to join or host.

### Project objectives:

---

**Compact and visualized data:** Dashboards in general are created with compact designs in mind, as their main objective is to be able to provide a general indication on potential improvement or deterioration at a glance. As such, all the information should be seamlessly displayed in a compact manner that should allow for data to be fully displayed in a single page cleanly on most devices and aspect ratios. This also means that the data being displayed in each given graph or chart should be compact and contain a lot of information at once that would immediately assist the user in identifying potential performance indicators.

**Design should be comfortable to the eyes:** One of the most important web design etiquettes that should be followed during the designing process is to ensure that the overall layout of the page including all the graphs and charts should be displayed in an appropriate format and layout relative to their sizes and the type of graphs to be displayed. Moreover, the background color should not be too distracting, and the objects placed within a given page should be spaced out evenly to ensure that the design does not seem cluttered. This would in turn ensure that the website looks friendly and unthreatening, which would in turn encourage students and teachers to utilize the website and potentially even ease the overall analysis process by making the data easy to see.

**Establish and suggest potential targets:** The website should be able to assist students and teachers alike in establishing potential targets for them they should follow. This would be done in the form of suggestions that would assist whoever may be using the website in immediately getting a quick summary or idea of what they should try to work on in order to improve. For students, this would be subjects they are weak in and that they should work on, while for teachers this could potential weak students that should receive additional support.

## Full feature list:

---

### General features:

---

**Main page:** The main page of the website is a feature that all users – even ones without an account – should have access to. This page could include features such as an opening animation to attract users, an “About us” section which will include the history and general information of the school, and a sign in button that would prompt users to sign in whether they are teachers, students, or executives, or admins.

**Arabic and English options:** A feature where the user can switch between Arabic and English as the language for the website would be extremely necessary considering the fact that potentially many parents or younger students might not understand English as well as they do Arabic, and having Arabic as the language for the website would be more comfortable. This is not a necessary feature, and as such it might not be implemented before the website is deployed, however, it is a feature that should be added nonetheless.

**Light and dark mode options:** A switch allowing users to switch between light mode and dark mode is not something necessary to the website whatsoever, however, it is such a major convenience point, that not only gives the user an added option to make the website more comfortable to the eyes, but it also makes it look significantly more professional due to the addition being such a beloved feature. However, as stated, it is not necessary and as such it could be ignored for the time being and simply added further down the life cycle of the website.



Figure 13 - Light & Dark modes

### For students:

---

**Average grades dashboard:** Students should be able to see their overall average grade of all their subjects, which is going to be the core feature and implemented as the “profile” page for students. This would act as a general profile for the students showcasing their name, number, and class, while also displaying the visualized data.

**Subject based dashboard:** From the average grade dashboard page, students would be able to select specific subjects, to further analyze them specifically into more detail, this would open an almost identical page to the average grades dashboard, however, this one would showcase all the data for that one selected subject.



**Interactive visualized data:** Although unnecessary, neat feature to add would be having interactive visualized data. An example of which is to have a tooltip showing the exact grade pop up hovering over a part of the graph. This could be a feature to be added further on down the development life cycle after the initial website's deployment.

### For teachers:

---

**Full list of their classes:** Teachers would have a list displaying all the classes that they teach, which they can then click on in order to access the average class dashboard. An additional feature of this list could be to have a search bar or filter feature that filters by year that would assist in finding the specific class that the teacher is looking, which would be extremely helpful for teachers supervising a large number of classes.

**Average grades class dashboard:** After selecting a specific class from the classes list, teachers would be displayed with a dashboard similar to that of the student-average dashboard, however, this would be dedicated towards the average grade for the entire class they have selected across the subject that they teach. This would help teachers in analyzing if all their classes are on the same standard and if they should focus more on a given class.

**Full list of their students:** From within the class-average dashboard, a list with all the students they teach within that class would be displayed, which can then be clicked to look into that students specific dashboard for the subject.

**Student dashboard for the subject they teach:** After clicking on a given student from the student list, teachers would be able to select specific students to see their dashboards for the subject that they teach only. This would allow teachers to focus more on specific students within their classes and try and work with them further to help them improve their results.

**Ability to edit student dashboard for the subject they teach:** If any inaccuracies have been found within the student's dashboard, teachers should have the authority and ability to edit a given students dashboard but only for the subjects that they teach that student in.

### For student affair executives:

---

**Full list of the classes with their teachers:** Student affair executives should be able to have a list of all the teachers as well as the classes they teach, this would help in identifying which students are in which class and what teacher supervise them.

**Ability to view all student dashboards:** Student affair executives should be able to have access to every students average and subject-based dashboard. This would help if executives ever needed to find out the grades to a specific student such as a parent/guardian asking to see their child's grades.

**Filter feature for students great in specific skillsets:** A major beneficial feature for student affair executives is to have the ability to filter out students based on grades and which ones might potentially be great at given skillsets. The idea of this feature is that student executives could filter out students-based ones with the highest/lowest grades in a specific subject, but they could also filter out students based a new metric that would be called something along the lines of "skill" or "potential". Said metric would be calculate using a combination of two things, the student's grade, which would make up 25% of the metric, and the other the teacher's recommendation, which would make up the remaining 75%. This is extremely helpful as a student with low grades in a given subject does not necessarily mean that they are bad at it from a practical standpoint. An example of which is a student that has low grades in electronics, however, they skillsets in the subject and practical ability is extremely high. This would be very helpful in the event that executives were looking for students for specific projects or events.

### Admin account:

---

The admin account is an account that has all the aforementioned features in addition to being able to edit every aspect of the dashboards as well add/remove students, teachers, and even executive accounts. This rank of account would only be given to the people managing the system such as the head of IT as well the school's principal.

# Hosting model

## Blazor hosting models:

### Blazor WASM:

WebAssembly (WAM) is a form of “code to binary” converter that takes code from any given programming languages and compiles it directly into binary format which in turn is then executed in the browser in order to run the a given web page. When it comes to Blazor, WASM is what allows for Blazor’s C# code to be executed on the browser rather than the default programming language that is JavaScript. According to the (Microsoft documentations, 10), the way that this process works practically is that the web page’s static files are uploaded onto a web server, which are then extracted and compiled by WASM on the browser. This means that the user would have to initially download two things, the .NET runtime as well as any libraries that the web application is using, which would be necessary in order to enable WASM on the client’s browser to run the code and any libraries were utilized to run the site’s components. Meaning that the entirety of the DOM (Document Object Models) for the frontend would be taken locally from the client’s browser rather than sending HTTP requests to a web server.

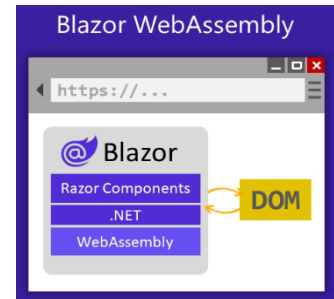


Figure 14 - Blazor WASM data transfer process illustration

This hosting model could be done in two different ways, the first of which being hosting the website using WASM and .NET for the frontend only, in that WASM would only be utilized to execute the web page’s static files and to run the UI of the website, with the backend being controlled by an external method such as GO or Node.js, this method of Blazor WASM hosting is known as the standalone method. The other method is to utilize Blazor for both the frontend and backend by hosting the Blazor WASM application on an ASP.NET core server, this allows for both the frontend and backend to be written in C# and controlled by Blazor, allowing developers to share the code between the two and utilize a single programming language for both, with this method hosting the application being the hosted method. For the purposes of the project, the hosted method would be utilized as to use only a single programming language which would simplify the overall process, especially when considering that heavy time constraints are placed upon the development process.

### Blazor Server Side:

Unlike Blazor WASM, the Blazor Server Side application works strictly on a web server and takes all the information it requires such as the UI components and updates to them, any events, and requests are done via a connection between the browser and said external web server is where the aforementioned information such as images, texts, forms, buttons, and anything else required for the website is contained. This process is done through a special connection known as a signalR connection, a specially developed connection method between the client (browser) and the server. In traditional communication in which the client sends an HTTP request signal and then the server replies by sending back something and is done in a unidirectional manner. However, using signalR, said connection is bidirectional as both the client and server can send a given signal without having to wait for a response or request from either side, consequently making the connection faster than the traditional connection method, also known as REST. It should also be noted that – as stated in the (Microsoft documentations, 10) – the stated signalR connections connecting the web server with the client’s device is formally known as a “circuit”.

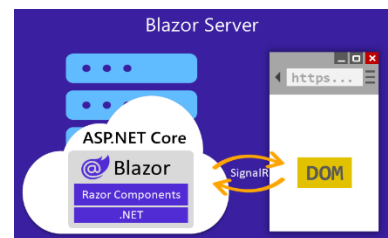


Figure 15 - Blazor Server side data transfer process illustration

It should also be noted that the way that the Blazor Server Side architecture has been setup, opening multiple tabs of the same exact web page creates a new connection or “circuit” for each individual tab, meaning that separate connections would have to be formed for each one, therefore demanding additional resources from the both the computer and the server as the new signalR connection would be sending and receiving signals from both the user’s device and web server.

## Features and limitations:

---

### Offline capability:

---

**Blazor WASM (pro):** Since the files are downloaded onto the user's browser, as long as the tab containing the website is not closed and neither is the browser itself, said files would still remain downloaded onto the browser, meaning that after the compile time, the website is not dependent on the internet for its core functionalities. Although the website would still be unable to gather or upload new information regarding the database, the entire website would be nonetheless functional, which would be extremely convenient for the user in the circumstance the user is unable to connect to the internet, as they would still be able to use the website offline, not to mention that students living in the barracks are likely to experience connection issues due to the heavy internet traffic that the modem would be receiving as potentially over 60 students would be using it simultaneously.

**Blazor Server Side (con):** As Blazor Server Side necessitates a server for all of its functionalities, anything being done without an internet connection will lead to an error message being sent by the website as no connection can be established between the browser and the web server in which every component required for the website to run on the browser is available.

### Scalability:

---

**Blazor WASM (pro):** Due to the fact that the workload is offloaded from the server and only runs on the client, the traffic on the server is rather limited, meaning that connectivity issues related to heavy traffic on the web server is a nonexistent issue except for requesting database information, which is significantly lighter on its than requesting both database information as well as the web page's components. This means that the website can be scaled significantly better than Blazor server side due to the ability to mitigate traffic-related connection and speed issues.

**Blazor Server Side (con):** Unlike Blazor WASM, the server side hosting model does necessitate the use of a dedicated server for requesting any type of information. Meaning that if the website were to accumulate a large number of concurrent users, depending on the web server, there is a chance that the overall user experience will feel extremely "laggy" due to the server being unable to handle the amount of traffic it is receiving. Although this issue is only prominent in servers with thousands of concurrent users, it could be an issue if the school decides to allow students to keep their accounts after they have finished their studies at the school.

### Runtime speeds:

---

**Blazor WASM (pro):** Runtime refers to the process of executing code after the application has been initialized and is already running on the browser. On Blazor WASM, due to the entire application being downloaded onto the user's personal browser, and as such, any HTTP requests needed are simply extracted from the code, without having the network's connection strength or server traffic potentially effecting the speed. As such, the overall runtime speeds of the Blazor WASM application are almost instantaneous and are rather extremely quick.

**Blazor Server Side (con):** Just like the issue mentioned about the scalability, due to the fact that a signal would have to be sent received from the web server each time any action is called rather than having it all done in one place means that the website would be much slower simply due to the send and receive process, not to mention other possible factors that could decrease the speed which are not found in Blazor WASM such as internet connectivity and web server traffic.

### Dependency on ASP.NET:

---

**Blazor WASM (pro):** The files uploaded for the Blazor WASM app are merely static files which are then sent to the client's browser to be launched, with the stated sending operation being done using any given hosting method.

**Blazor Server Side (con):** Unlike Blazor WASM, the connection established between the server and the client in the Blazor Server Side hosting model requires for a signalR connection to be established, a connection specialized for ASP.NET applications, meaning the hosting service that would be utilized needs to be able to support ASP.NET, which limits the number of potential hosting services to be used and also probably increases the price of the subscription plan to be chosen.

## Browser compatibility:

**Blazor WASM (con):** Not every browser is able to support the WebAssembly architecture of compiling into binary and running the web application on the browser independent from a server, which is especially the case on older version of some browser and is entirely unsupported on Internet Explorer. With that being said, the number of people using said browsers is extremely niche and probably accounts to 0% of the target clients, however, it still is a disadvantage of using Blazor WASM.

**Blazor Server Side (pro):** Since Blazor Server Side is very similar to the traditional method of web development, all older and modern browser support application hosting using this hosting model.

## Compile speeds:

**Blazor WASM (con):** Since the files required to actually run the application have to initially be downloaded onto the client's browser before that can be compiled, with said files potentially being up to two megabytes. As such, downloads plus the initial compilation time when launching the app for all components and libraries can take a significant amount of time.

**Blazor Server Side (pro):** The Blazor Server Side hosting model does not require any downloads that are equivalent to that of Blazor WASM, with the only thing that would be compiled simply being the data being sent to it by the server which would occur for the page currently opened, and not for the entirety of the website.

## Dependency on hardware capabilities:

**Blazor WASM (con):** Due to the fact that Blazor WASM runs all the compilation and even the website itself locally, this also means that local resources would have to be spent in order to run the web application. This could be a major issue with devices that have weak hardware capabilities, as the website could significantly weaken the system or simply not run at all due to a memory requirement exceeding what is currently available on the system.

**Blazor Server Side (pro):** Blazor Server Side has very minimal dependencies on the hardware of the system that is running the web application. That is the case since most of the components are being utilized by application would be taken from the sever, and as such the workload would be mostly put on the server rather than the client's device. With that being said, as mention earlier, the way that the Blazer Server Side architecture has been setup makes it so opening the application in multiple tabs creates a new circuit/connection between the client's system and the server, meaning additional load being applied to the device, however, launching two or three tabs would nonetheless take up less resources than the Blazor WASM hosting model.

## Hosting model of choice:

To conclude, the hosting model of choice to be utilized for the project is going to be **Blazor WASM**. The reason to which being that the latency issue during runtime for Blazor server side are much more severe in comparison to any of the given drawbacks seen in Blazor WASM. This is the case since students – the majority of the targeted user base – are more likely to have poor internet connection than have devices with weak hardware, especially when most 100% made in the last 10 year should be able to run WebAssembly without any issues. Moreover, as mentioned earlier, students in the barracks and the ones living in rural areas could have frequent issues relating to their network connection, meaning having a website that can run without any lag after it has been initially compiled is going to be extremely convenient to them. Moreover, as mentioned earlier, it would be very unlikely that anyone is utilizing browsers that do not support WebAssembly, especially when considering said browsers are extremely old and almost no one uses them, meaning issues relating to the compatibility of student devices should not be a significant issue.

Strengths & weaknesses		
Feature	Blazor WASM	Blazor Server side
Scalability	✓	✗
Offline capabilities	✓	✗
Fast runtime	✓	✗
Fast compile time	✗	✓
Compatible with all browsers	✗	✓
Independent from ASP.NET	✓	✗
Reliability on hardware	✗	✓

Figure 16 - List of all strengths and weaknesses

# Sustainability and maintenance

## Industry-known standards:

---

### Code structure and validation:

---

One of the main aspects of the project as a whole is the codebase in which it has been built on. As such, said codebase is the most important part when it comes to maintaining and ensuring the sustainability of the code as new updates roll out in order to improve the website. However, if it were to be written in an unorganized and random manner with varying inconsistencies throughout all of the project's files, then maintaining the code would be extremely difficult, and finding the locations in which errors might occur would consequently also be a tough process, especially if the person doing said maintenance is not apart of the original team that developed the project and is unfamiliar with the thought process and the structure of standards – if any – that was used.

Some potential ways in which this issue could be mitigated is to use a standardized programming structure. The leading organization trusted by most software developers in making said code structures is the World Wide Web Consortium, or W3C for short. As such, W3C's standards and online code validators are going to be utilized in order to validate the formatting of the team's HTML and CSS code. Since C# is not the leading web development language, W3C does not have a working validator for C#, as such, the formatting structure to be used for C# would be the coding conventions for the C# programming language written in the Microsoft documentations.

### Comments and documentations:

---

Even if the codebase written for the project is extremely organized and utilized standard coding structures that most programmers might be familiar with, if the code had no form of documentations whatsoever, it would nonetheless be especially confusing for whoever is to work on the project after the initial team. That is the case since web development and programming in general is a significantly large field that consists of varying libraries and methods that are utilized to perform the same solution, meaning it cannot be expected of someone – even a senior developer – to understand every single line of code, which is where documentation of the code comes in. However, in the circumstance that there are no comments, written nor have there been documentations provided for the code, it would be almost impossible for someone to work on the code in order to either fix it or update the website.

However, although comments are necessary additions to ensure that the code written is easily comprehensible by other developers, they should also not be overused, but rather used carefully and only where necessary, even then, comments should try to be as short as possible so that the code written is still the main focus within the project file.

## Maintenance cycle:

---

### Monthly:

---

**Test for general website functionality:** Simply going through all the account ranks and testing various functionalities in the website to make sure that everything is working accordingly and fixing it if not.

**Check and note user interactions:** Check how users interact with the website through the use of google analytics. This includes checking popular pages, most clicked buttons, what devices the user's are mostly using, etc. During this stage, only noting down any interesting interactions and statistics would be done, nothing would be acted upon from one month's notes yet.



**Check for component updates:** Check if any of the components or libraries utilized have a newer version available and updating them if needed. This is done to ensure that potential vulnerabilities in older versions of said components cannot be exploited by attackers.

### Quarterly:

---

**Analyze user interactions and update accordingly:** Analyze the four months' worth of client interactions with the website and make changes to the most popular pages and look for potential enhancements that could be made to the most utilized aspects of the web pages according to the given data as well as potential optimization of the website based on the client's most used device.

**Run a full security test:** Testing the entire security infrastructure put in place is necessary every four months or so to ensure that no exploitations have been missed during earlier development. This maintenance step also includes researching on potential vulnerabilities that have been found in the utilized components and libraries as well as any recently discovered zero-day attacks and the solutions to them.

**Analyze loading speeds and update accordingly:** Check the website's overall loading speed, which in this specific case would be the rate WebAssembly file downloads. If the data seems to be much slower than previous ones, then analysis on the reason should be done, and from there potential updates should be rolled out to decrease file download times. Example of potential updates could be to remove unnecessary files or decrease image quality through compression in order to reduce the file sizes that would have to be downloaded upon launch.

### Yearly:

---

**Design overhaul considerations:** Consideration and analysis of the entire design should be made as to make sure that any flaws with the design that have been noted earlier should be edited and improved upon now. Additionally, it should actually be encouraged to change the UI, even if the new UI does not cause a great improvement, simply to provide users with a fresh experience and to show that the website is being consistently updated, further improving the reputation of the school as it shows care is being put into a product made specifically to assist their students.

**Full review of codebase:** A full review of the entire codebase should be done even if no UI changes were to be made. This would assist in making sure that everything has been checked for potential vulnerabilities that can be exploited by attackers. Moreover, this could also help in cleaning the code further and placing it in a better and more readable format than what might have been used initially, this should of course be done while still being within standardized formatting so that any developers that works on the project next is familiar with the layout of the codebase.

**Renewing and analyzing subscription plans:** Finally, any subscriptions and annual expenditures that have been used should be renewed by the end of the year, which would most likely only consist of the domain and hosting service, although there is the possibility of additional subscription fees being added due to implementations of further features that required a paid subscription.

In addition to simply renewing subscription, with all the data and information that has been gathered throughout the past year, an analysis of the available subscription plans provided by the hosting service should be done in order to accommodate for any issues that might have been faced that have been caused by the chosen plan. Ideally, the plan would stay the same, however, the analysis could also lead to either a downgrade or upgrade in the chosen plan, which would be caused by an initial under or overestimation in the website's traffic.



# Cybersecurity

## Malware attacks:

---

### Using a Web Application firewall:

---

A Web Application Firewall (WAF) is a necessary method used in protecting website and any device connected on a given network from potential malware attacks. A firewall program is a utility software containing an immense library of preconfigured codes, packets, IP addresses, and data that could potentially contain data harmful to the website. Using that library, the firewall scans incoming traffic that attempts to go into the website and compares it with the codes available for it in its library. In the circumstance that a file trying to go into the website matches with anything on the library, the firewall would prevent that packet/data from going passing through. Basically, a firewall could be depicted as an actual gate that scans the IDs for all incoming traffic, and if an ID (the packet) matches a blacklisted ID (the codes from the library), then it would prevent it from passing through and going into the website. The firewall that would be specifically utilized would be Azure's Web Application Firewall due to an easier integration process as Azure already by the hosting service provider for the website. Said firewall would be acquired through Azure as they provide their own WAF if the team were to utilize them as their hosting provider.

### Restricting entry fields:

---

One of the main – and relatively easy – ways that attackers could install malware read by the website or database is by simply uploading said malware into the website using an unrestricted upload box. Meaning that one of the many potential forms on the website has an upload box which allows for users to enter any type of file, which may include .php files or a .xml file that contains an SQL command which can be used to gather information from the database. As such, one of the main and probably simplest ways in which this issue could be resolved is by restricting the type of files that could be inserted into the website's forms. In HTML, this could be done by adding this code snippet into the form's codebase: `<input type="file" accept=".png, .jpeg.">`

### Check for component vulnerabilities:

---

One of the leading ways in which attackers can exploit web applications is by searching for some of the opensource components that have been used by the application in order to find potential vulnerabilities that could be exploited through them such as the component having permissions higher than it should have, showing or extracting vital information about the database, and more. This is done because the original codebase of the website is most likely not going to be opensource and therefore they would be unable to look for weaknesses in the original codebase.

As mentioned by (Kiprin, 2021) solutions could be implemented as to mitigate this issue, the first of which is to actually know which part of the codebase is opensource, and to do so, a Software Composition Analysis (SCA) tool should be utilized, said tool scans the entirety of the codebase and outputs a report containing all opensource code that could be found on the internet. From there, security tools should be used to find the vulnerabilities themselves. Some examples of these tools include Crashtest's Security Suite, an application that scans the codebase for potential vulnerabilities and provides a report on them, with this being done in an automatic and continuous manner, it should also be noted that this is a paid service, however, there is a 14 day free trial which would be utilized during the testing stage of the development process. Additionally, Metasploit is another important application utilized in enhancing a website's overall security, it is used as an exploitation area in which an attacker could attempt to breach the vulnerabilities on the website, and if successful, this allows developers to know whether or not their security additions are sufficient or not.

## Access control:

### User authentication:

A potential method that an attacker could use in order to gain access to vital information about the website is to gain access to an account with high-level authorizations. This could occur through phishing attacks or somehow finding out the user's username/email and password. A method that could be used to prevent this issue is to implement two-factor-authentication (2FA). 2FA is a method in which users are sent a message to their personal phone or email, with said message consisting mainly of a verification number/code that would be inputted in the login box. This allows the website to verify that the person trying to enter the account is in fact the real user. Of course this method is not foolproof, and if the attacker has access to the user's email account, there is very little that could be done. Additionally, it should be noted that 2FA should not be forced on users, but rather given as an option that can be turned off in the settings, as it is a rather tedious process for the user if they had to do every single time they login, however, having 2FA on would be the default setting in order to encourage more users to have the setting turned on.

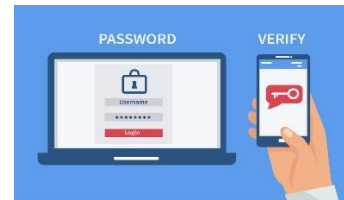


Figure 17 - 2FA process

### Teach clients about phishing attacks:

One of the most prominent ways in which attackers could get access to user's information, with financial information and passwords being the main target, is through the use of phishing attacks. Phishing is the process of exploiting user's and trying to manipulate them into thinking the attacker is a trusted source, leading to user's deliberately giving attacker's their personal information without them knowing that the person receiving the information is not a trusted source, but rather a malicious attacker. This is done mostly through email addresses or text messages in which the attacker pretends to be someone's superior asking for the user's password for one reason or another.

One of the main ways used in preventing potential phishing attacks is to teach the userbase about them, allowing them to differentiate between real requests and messages from superiors and coworkers and phishing attacks. In general, this would be done by displaying a message in various parts of the website saying that "Neither the website nor superiors would ever ask about your account's password", which would help in making any message that asks about the password immediately suspicious. This would be done, however, in addition to that, since the team has easy access to the website's client base considering it's a school, an assembly would be held discussing phishing attacks specifically, therefore ensuring that the message has been received by the entirety of the userbase.

## Additional protection:

### Using a public key certificate:

A public key or Security Socket Layer (SSL) Certificate is a secure DNS connection between the website's web server and any device it connects to. This is done by using a cryptography software to encrypt any HTTP GET/POST requests. This is extremely helpful in protecting users, since if an attacker were to connect to a given network, then – with the right software and knowledge – could see the traffic going into and out of said network, which includes networks that the user could be logged into. This danger in this revolves around potential Man-in-the-Middle (MitM) attacks, an attack were the



Figure 18 - SSL protection

attacker places himself in between the user and website's web server, therefore seeing all the traffic that goes between the two and in turn also being able to change the traffic and files into whatever they want, therefore potentially changing the web server's respond into malicious malware and files. However, if the connection were to be encrypted, the attacker would be unable to understand what requests are being sent through nor tamper with them since they would need the decode key, consequently preventing them from initiating MitM attacks.

The provider from which the SSL certificates would be acquired is Azure, as they offer free certificates as long as the team utilizes MS Azure as their web server, with said certificates being created by DigiCert but provided through Azure as mentioned by (Philip, 2021). Once the SSL certificate has been setup, it can be realized whether or not it is activated through looking at the website's URL, as SSL certified website starts with "https" rather than "http".

### Backing up the website:

In the case that an attack was successful, and said attack caused major damage to the entire website and caused it to go down, in which case, having a backup file is extremely necessary. Said backup should mainly include the most recent database version, since information on the database could occasionally change if something were to be added or removed from it by the teacher/admin account. That is important since certain types of attacks – such as SQL injections – could manipulate and affect the database. Backing up the files is a rather easy process, if Azure were to be the team's hosting provider of choice, through the Azure services portal, in the database section, developers would find a backup option in which they could download the latest version of the database in a .xml file format.

### DDoS protection:

A DDoS attack – an abbreviation for Distributed denial of service – is a form of attack launched by a malicious attacker in which they request data from the web server in a simultaneous and rapid manner, which consequently overloads that network, causing it to stop functioning appropriately. This could be done in a variety of manners, but one of the most common ways is for the attacker to utilize a large number of different computers or even household utilities that have internet

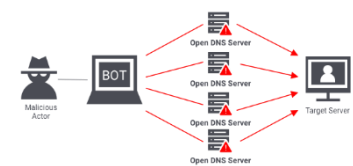


Figure 19 - DDoS attack illustration

connectivity such as microwaves or fridges (IoT devices) in order to send a number of similar or varying requests to the network such as data request messages like Echo/Ping requests, such request are rather common and are used by a device to request a given item from the web server such as an image or a text, from which the web server then responds and – generally – provides what has been requests. If this is done in a very rapid and overwhelming manner, the server would be unable to respond to everything at once, therefore overloading it and causing it shut down.

One of the simplest ways and potentially most effective ones that could be used it to enable DDoS protection in the Azure App Service portal. Said protection offers DDoS protection through the use of a machine-learning algorithm that is able to detect whenever traffic going through and into the web server is abnormal and is not apart of normal user traffic, but rather is a malicious attempt of overloading the web server through continuous and repetitive requests. Another method is to use a WAF, which – as has been discussed earlier – prevents potential malware from going into the web server, said malware could include DDoS-related viruses, therefore utilizing a firewall could in fact prevent more advanced DDoS attempts utilizing sophisticated malware attacks. Although the aforementioned processes should be able to help significantly with preventing DDoS attacks, another method to at least decrease the damage that the attack might cause is to create a rate limiter that limits that maximum number of requests that can be sent for each given second. As mentioned in the (Microsoft, 2022) documentations, a method of doing this is to use advanced request throttling and utilizing a "rate-limit-by-key" function that can be simply implemented in the codebase, which prevents any given IP address from sending the same requests several times per a given time, therefore preventing brute force DDoS attempts. Example: `<rate-limit-by-key calls="50" renewal-period="60" counter-key="@context.Request.IpAddress"/>` → the shown code snippet would prevent a single IP address from sending the same request 50 within a single minute.

# Testing plan

## Functionality testing:

---

### Account rank functionality and testing:

---

The account rank system is one of the core functions of the website, and as such testing that it's working it would be the first thing to be carried out during the testing stage. This test would involve logging into all account ranks and making sure that the type of information being displayed in each is accurate and does not accidentally display information that is outside of the rank's intended authorizations and limits. This would also include checking and making sure that each account and its ranks cannot use functions that are not intended for said rank, in that an executive account should not be able to edit student grades the same way that a teacher account could, nor should student accounts be able to see information related to average grade of other students, and so on. If something were to be displayed accidentally, or an account were to have extended or weaker than intended authorization, then the team would have to check codebase and SQL commands written and edit them accordingly.

### Graph algorithm testing:

---

This test would be carried out using a separately programmed algorithm that would be coded in python which would check that the data of the graph are showing are indeed accurate and there is no discrepancy with the data relative to the actual grades. The testing process would involve using a graphing algorithm that would depend on the graph shown (i.e: quadratic graphs would utilize their sperate method, same with cubic graphs, and so on). This would allow the team to know whether or not the data being shown on the website's graph matches the actual data and has very little (rounding inaccuracy) to no inaccuracies.

### Usability and convenience testing:

---

This test would revolve around testing the usability of the UI and the overall convenience of trying to get a piece of information. The way that this test would be carried out is by asking people that have never seen the website to try and find something specific, and depending on the time they take to do, the team would know whether the UI requires any enhancements or pointers to assists users more. This would assist in making sure that the website's UI is not frustrating, and users can find information quickly without any trouble. This test would also include analysis of all the pages and buttons found within the website and writing down a list of buttons that could potentially be removed due to them being unnecessary. This would result in buttons either being entirely removed or their functionality being combined with another buttons, consequently making the UI cleaner and more streamlined.

### Testing loading times:

---

An area that could be easily overlooked during the testing stage is the loading times for the website, as this is a major convenience point for the userbase, with long loading times being a major cause for frustration. This test would be done through initially testing out the loading speed of the website with an empty database, and then slowly increase the amount of information found within it until it is almost full, while testing and analyzing the loading speeds throughout the entire process, and if they end up significantly above what is deemed to be acceptable, then changes would have to be made in order to decrease the total compile time. Some changes that could be made include remove unnecessary files or decreasing image quality by compressing them further which in turn would decrease the size of the file that would have to be compiled to lead them.

## Security testing:

### Checking vulnerable components:

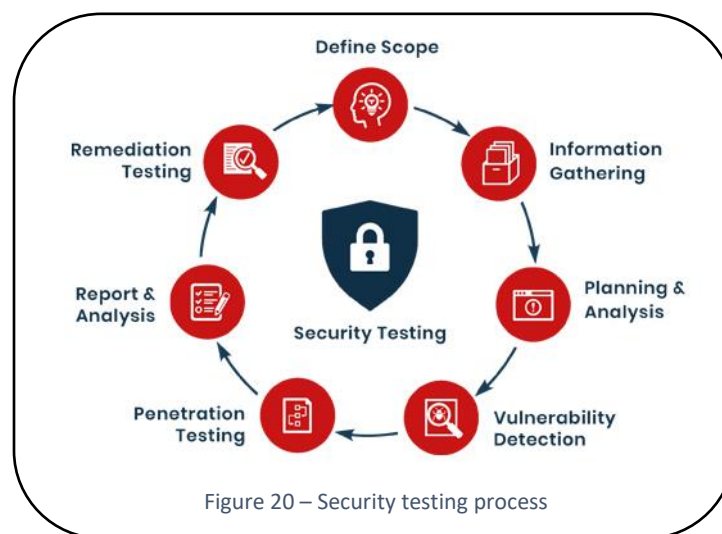
One of the first security related tests that would have to be carried out is testing for any potential vulnerabilities that could be exploited in any open-source components utilized, since although being open-source means that it can be easily integrated and utilized, it also means that malicious attackers could freely find any vulnerabilities in the code and attempt to exploit them. The testing process – as mentioned in the cybersecurity section – would initially involve using a Software Composition Analysis (SCA) tool in order to find out any opensource code found within the codebase, and from there a vulnerability scanning tool would be utilized, examples of which being Crashtest's security suite or Acunetix's vulnerability scanner. Any vulnerabilities found within the outputted report created by the scanner would then be further analyzed and potentially fixed.

### DDoS limit testing:

This test would involve attempting a deliberate DDoS attack as to find out whether all of the aforementioned DDoS protection methods would be sufficient enough to protect against a real DDoS attack. Some methods that could be used to initiate the DDoS attack is through the use of an Alpha Adapter, a data transmissions device capable of sending out request signals at extreme speeds if a big enough antenna were to be used with it, another method that could be used if using the Alpha Adapter was not possible is to try and utilize a legitimate bot server dedicated for DDoS attack testing. If the attack were to be successful, then the team's protection methods would have to be thought over and analyzed, and new methods would need to be implemented.

### Breach simulation test:

Lastly – and mostly importantly – is to launch a real breaching attack, in which an attacker would attempt to hack the website in order to simulate what a real attacking process would look like, and whether or not they would be able to breach the website's defenses or not. If the attack was successful, then the team would be able to know what exactly the attacker was able to exploit and fix the issue, and if not, then it would be almost sure of that the website is in fact extremely difficult to breach. Said attack would be carried out by a skilled breacher heavily involved in the red team/black hat side of the cybersecurity profession, who is also an associate of the team.



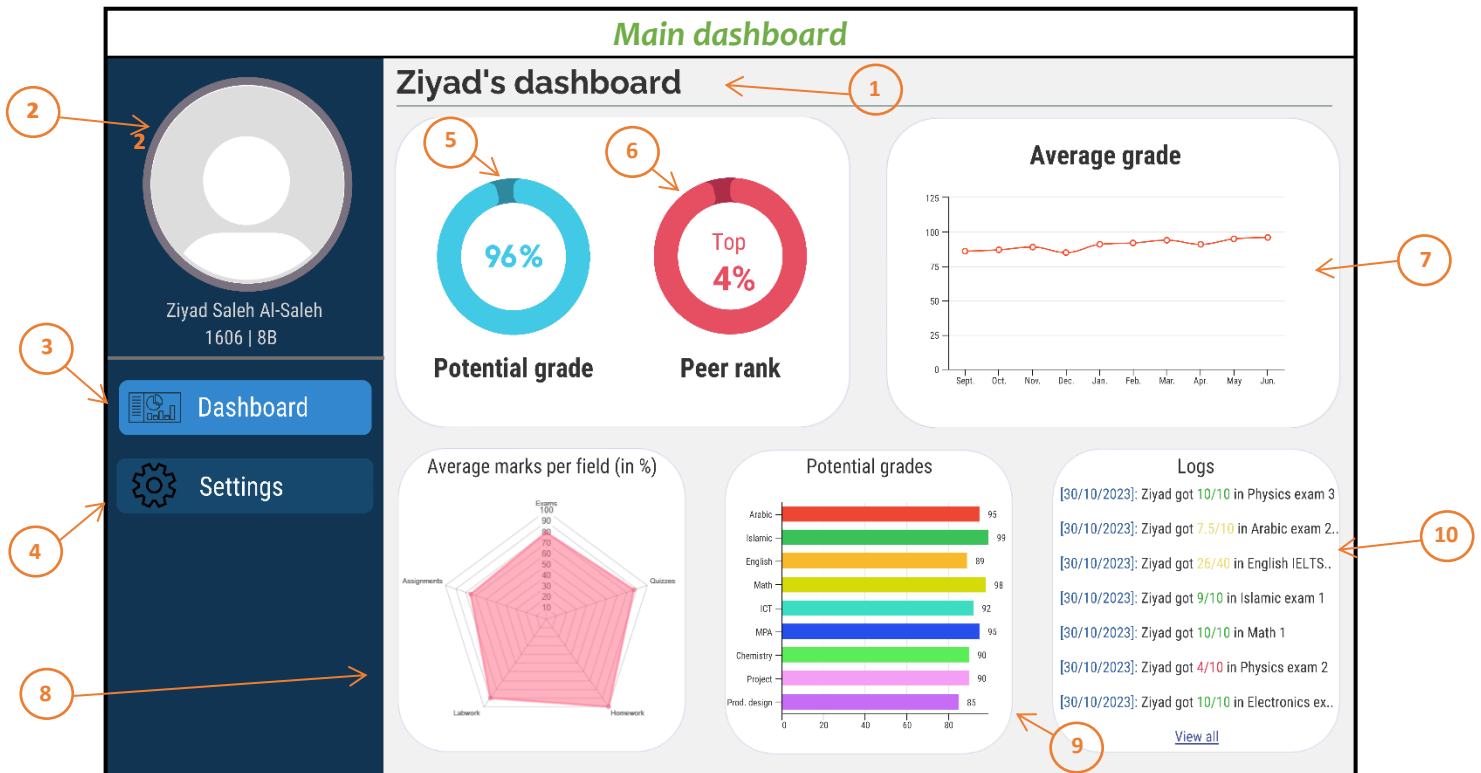


# Designing



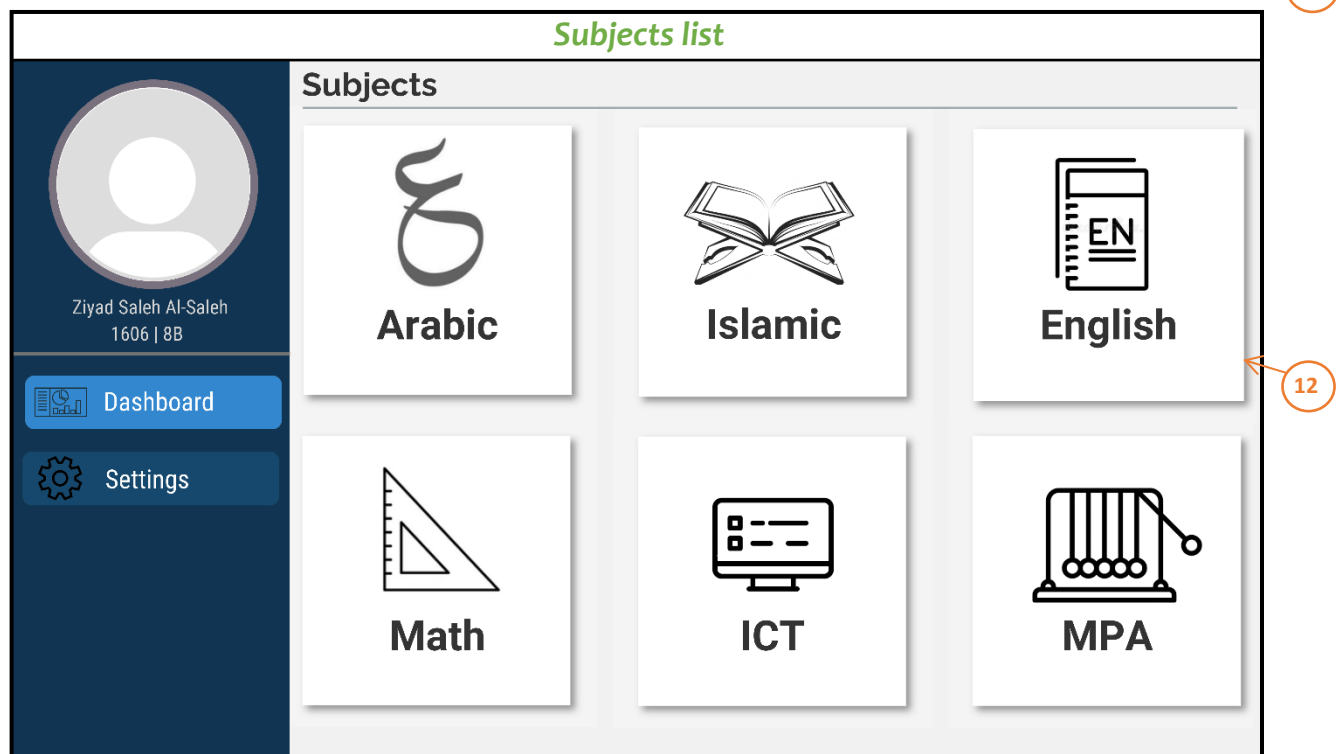


## Design concepts



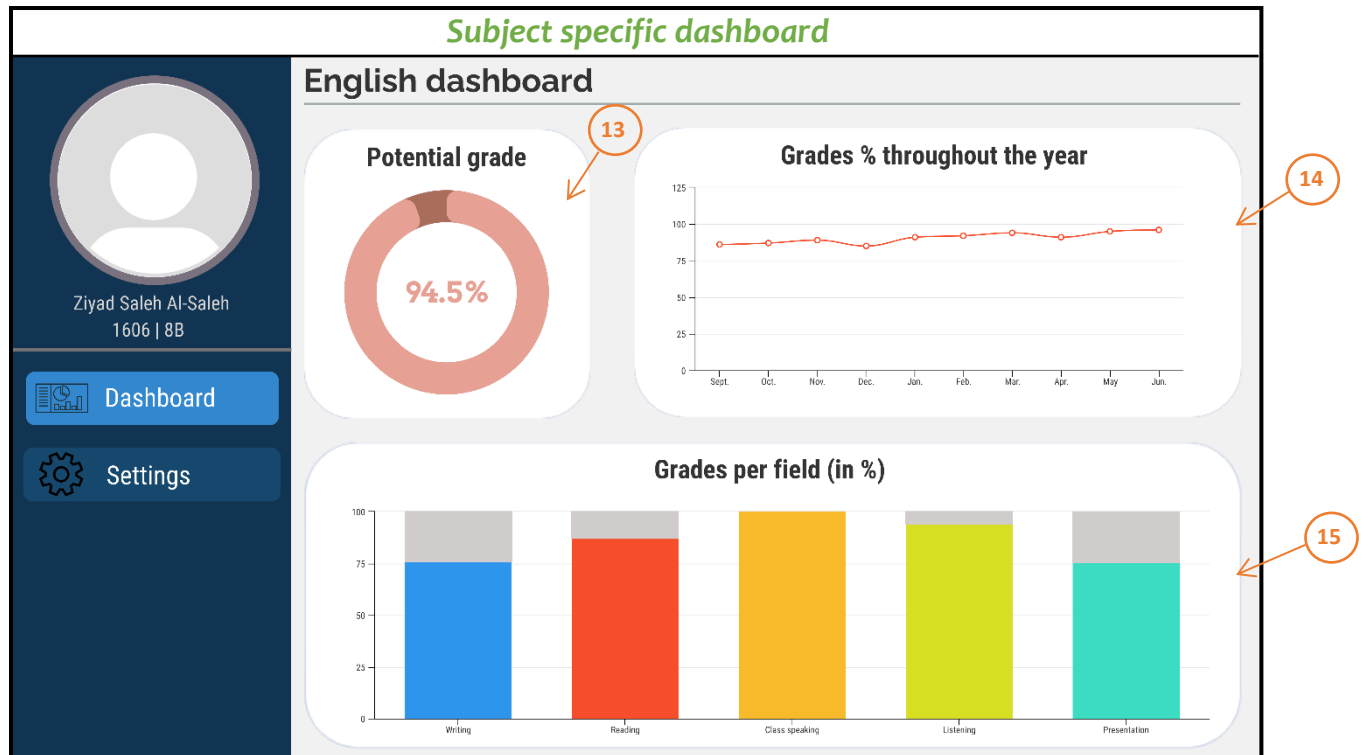
Scroll down

Fade-in CSS scroll animation



After clicking on subject

Immediate transition or CSS animation



## Design thought process

### Common elements:

(1) **Title:** An `<h1>` element that simply displays the title of the current page. This element is common across all pages and has a thematic break line `<hr>` simply for aesthetic purposes.

(2) **Profile:** An area displaying the student logged into the account, consists of three elements, one image `<img>`, and two small headings `<h2>`.

(3) **Dashboard button:** The dashboard button, consisting of an image element `<img>`, as well as a small bit of text which would be the value of the button `<button type="submit" value="Dashboard">`, said button would also have a `focus` and `visited` CSS attributes, which cause its color to change when it is clicked and when it is selected.

(4) **Settings button:** The settings button would have the same attributes as the dashboard button. This button would take the user to the settings menu. Items on the settings menu have not entirely been decided, but some things that could be expected would be an Arabic and English as well as dark and light theme options.

## Main dashboard:

---

**(5) Potential grade:** This pie chart displays the potential mark the student is able to achieve if they do not lose a single mark until the end of the year.

**(6) Rank among peers:** A pie chart displaying the student's current rank among their peers in the form of a percentage. Used to motivate students and encourage competitiveness.

**(7) Average grades timeline:** A smoothed lined graph displaying the percentage of marks the student has gotten throughout each month of the year. Said graph could also be changed to display the percentage per week. Helps students know whether they are improving or not.

**(8) Marks per study area:** A radar chart showing the student's grade in a percentage in five areas of their studies, exams, tests, homework, labwork, and assignments. The areas would change based on the year the student is in.

**(9) Potential grades for each subject:** This bar chart displays the potential grade for all the subjects that the student is taking.

**(10) Logs:** The logs tab/area shows the grade logs of the student, displaying the grade that they got, the maximum grade, the subject, and the test/exam's name. This tab is automatically updated whenever anything gets added to the student's grades.

## Subject list:

---

**(11) CSS scroll animation:** A CSS scroll animation would be implemented when the user scrolls down towards the subjects list. Said transition would be a simple fade-in animation, which would cause the boxes to fade and fade out as the user is scrolling towards and away from the subject list section. Although this would be a small feature, it makes the dashboard look significantly more professional.

**(12) Subject buttons:** A button for each subject would be created. The button would have both a **hover** and a **focus** CSS attributes, making the button look much cleaner and nicer when the user is hovering over it. The button would have a slight yet noticeable shadow effect and would contain both the name of the subject as well as a representative image as its value.

## Subject specific dashboard:

---

**(13) Potential grade:** Just like the potential grade pie chart in the main dashboard, this would display the same thing, however, it would do so on a smaller scale as it would be only for the subject that has been clicked on.

**(14) Grades timeline:** Same type of graph as the one shown in the main dashboard, however – just like the potential grades pie chart – this is done on a smaller scale and only shows the percentage of grades gotten for the subject that has been chosen.

**(15) Grades for each study area:** This part of the subject dashboard displays each area that makes up the total of grades of the subject and displays them on a bar chart displaying the grades in the form of their percentage and showcasing how much the student has lost through the greyed-out section of the bar.



# Logbook



## Initial team organization and planning meetings:

Date: 28/9/2022	Programming method and learning path (meeting 1)
Agendas:	Discussion
<ul style="list-style-type: none"> <li>• Dominant web dev methodology</li> <li>• Possibility of using C#</li> <li>• Learning resources</li> </ul>	<p>During the first meeting, the team discussed and discussed the dominant programming methodology and the potential of using C# as the programming language for developing the website, this was done due to C# being a language familiar to all members of the team. From there, the team started looking for learning resources that could be utilized in order to learn the programming languages necessary for creating the website, during this, the team looked at snippets of the programming courses and reviews about different courses and books. In addition, the team was also trying to figure out how long would it take the team in order to have a basic understanding of the concepts and be able to semi-independently work on the website.</p>
Action points for the next session:	Conclusions
<ul style="list-style-type: none"> <li>• Discuss web hosting method</li> <li>• Discuss hosting provider</li> <li>• Discuss databases</li> </ul>	<p>After the meeting, it was concluded that the programming methodology to be used would be C# as the language and Blazor was the framework of choice. From there, the resources that were chosen to be used for learning the web development were the "C# masterclass" by Denis Panjuta, a suggestion brought by Al-Hosni due to him utilizing this course to learn the C# language prior to starting the project. Al-Hosni additionally suggested Programming with Blazor by Filipe Gavilán as he was interested in the course prior to the project, and now he had the chance to actually buy it. Finally, the HTML/CSS course by Mosh Hamedani was chosen due to the great presentation and teaching method that was presented during the course snippet found on YouTube.</p>

Date: 30/9/2022	Web hosting method (meeting 2)
Agendas:	Discussion
<ul style="list-style-type: none"> <li>• How to deploy the website</li> <li>• What database to use</li> <li>• What IDE to use</li> </ul>	<p>The main thing that was discussed during this meeting is the overall website hosting and deployment method, which was something none of the team was familiar with and so everyone had to do their research on it beforehand. From there, the meeting by reading reviews of the main hosting providers in the market including Azure, GoDaddy, AWS, and more. Additionally, the team discussed what databases might be used, and although Al-Hosni used MSSQL as part of the C# masterclass course, he is not familiar with other SQL databases, and so he was open for further discussion about the topic. Finally, although every member was already sure about their IDE choice, discussion on potential alternatives was nonetheless.</p>
Action points for the next session:	Conclusions
<ul style="list-style-type: none"> <li>• Allocation of tasks</li> <li>• Prioritizing tasks</li> </ul>	<p>After hours of discussion and researched, the team finally decided on choosing Azure as the IDE of choice due to its great compatibility with Blazor and C# as well having great documentations relating to hosting Blazor apps. Furthermore, since Azure was chosen, the team also decided on going with MSSQL as it would have greater documentations in relation to hosting an MSSQL database on Azure, furthermore, SQL server is a part of the C# masterclass course, meaning it would already be something more familiar than MySQL. Lastly, the team looked into JetBrains Rider as well as JetBrains's Fleet, while comparing them to Visual Studio, however, Rider was a paid service, and Fleet was extremely new and had very little documentations for it, not to mention that it was not necessarily going to be free in the long run, as such, the team simply stuck with Visual Studio as they have already had experience working with VS, it – just like most other things utilized by the team – is a Microsoft product, and both the C# and Blazor courses chose Visual Studio as their IDE of choice, meaning it would be easier to follow along than if Fleet or Rider were to be chosen.</p>

<b>Date: 5/10/2022</b>	<b>Allocations of members and timescale (meeting 3)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• Main milestones</li> <li>• Decide on task allocations</li> <li>• Predict timeline through Gannt chart</li> </ul>	<p>This meeting is probably the most crucial of all as it is when the team decided on allocating tasks and having an idea of how the timeline would approximately look like for the entire production process. The meeting first started with dotting the major milestones and functions that needed to be implemented into the website, from there, a rough Gannt chart was created in order to have an idea of how the entire process would take the team which would assist in at least having a baseline for what might be considered the “ideal” timeline. From there, the team allocated one another to certain tasks based on each person’s respective skills.</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<ul style="list-style-type: none"> <li>• Brainstorm the core features</li> <li>• Discuss features for each rank</li> <li>• Discuss the Blazor hosting mode</li> </ul>	<p>Thankfully, during the task allocation aspect of the meeting everyone agreed and was satisfied about the job that they had to do, with Al-Hosni being chosen for the algorithms in the backend and connecting SQL server, while working on the dashboard and its required components on the frontend, while Al-Saleh had the main page and ranking algorithm, and finally, Al-Farsi was allocated to meetings management, graphical design assistance on the frontend, and data entry on the backend. The team also mapped out an approximate Gannt chart with the only the core features, with the entire timeline coming out to approximately 130 days, which is under the time limit, however – as mentioned – this were simply the core features.</p>

### Website functionality meetings:

<b>Date: 7/10/2022</b>	<b>Ranking system and general functionalities (meeting 4)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• Core features</li> <li>• Features for each rank</li> </ul>	<p>The first that was done during this meeting is brainstorm some ideas that could be implemented into the website, with said ideas being core features that everyone could utilize from students, teachers, and executives, and would be more convenience related features. From there, the team started talking about the ranking feature of the website, with the first thing being considered is whether or not teachers and executives deserve separate tanks or not, and what authorizations should each rank have. The Blazor hosting model was supposed to be discussed during this meeting, however, Al-Hosni and Al-Farsi had to leave due to personal reasons.</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<ul style="list-style-type: none"> <li>• Discuss the Blazor hosting mode</li> <li>• Site maintenance</li> <li>• Total expenditures</li> </ul>	<p>The core features that thought of during the meeting were to implement were a toggle for Arabic and English languages as well as a toggle for light and dark mode, it was also decided that said features would be left at the end due to them not being a necessity, with the defaults being Arabic and light mode. Additionally, it was decided that the any visualized data should be interactive to a certain extent (i.e: students can hover over a part of the graph to display the exact value). It was then decided that the number of ranks that would be available would be as follows: Student &lt; Teacher &lt; Executive &lt; Admin. Student accounts are only able to view their grades, teachers can see the grades of the classes they teach as well edit their own student’s grades, executive accounts can see everything but are unable to edit anything, and admin accounts can both see and edit anything, and said admin account would most likely be only given to the school’s principle and people working at the IT department in the school.</p>



<b>Date: 9/10/2022</b>	<b>Hosting model, expenditures, and maintenance routine (meeting 5)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• <b>Hosting model</b></li> <li>• <b>Site maintenance</b></li> <li>• <b>Total expenditures</b></li> </ul>	<p>The meeting initiate with discussing the Blazor hosting model by comparing the pros and cons of the two available models, with most of the meeting being held through a live stream hosted by one member showing the Microsoft documentations discussing said pros and cons. From there, the team looked up how a website would be maintained in general and wrote down the maintenance process and routine that was relevant for the dashboard. Finally, a small excel table was created to showcase the capital and operating expenditures of the site.</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<ul style="list-style-type: none"> <li>• <b>Consideration of constraints</b></li> <li>• <b>Potential risks</b></li> </ul>	<p>The team decided on Blazor WebAssembly as the model of choice due to its fast runtime being a much greater advantage than Blazor server-side's fast compile time. A site maintenance plan was then mapped out, which included functionality testing and component updating on a monthly basis, user interaction analysis and security testing quarterly, and design overhaul analysis and extensive codebase review on a yearly basis. Finally, the expenses came out to around \$73.48/28.28OMR capital, which consisted of the courses and domain name, while the operational expenses consisted of the web and database hosting, which were both free for the first 12 months of use, which was the case due to Azure's first year free trial.</p>

### Risk assessment meetings:

<b>Date: 12/10/2022</b>	<b>Constraints and contingencies (meeting 6)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• <b>Project constraints analysis</b></li> <li>• <b>Contingencies</b></li> <li>• <b>Mitigation plan</b></li> </ul>	<p>The meeting started with what sort of contingencies would the project potentially face during the development process. After which, an analysis of the budget, time, and scope constraints was done. From there, a likelihood vs impact map/table was created for both the contingencies and constraints in order to showcase their risk in terms of the two major factors when it comes to risk assessment (risk = probability and influence).</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<ul style="list-style-type: none"> <li>• <b>Cybersecurity brainstorming</b></li> <li>• <b>Malware mitigation</b></li> <li>• <b>DDoS mitigation</b></li> <li>• <b>Account takeover fraud mitigation</b></li> <li>• <b>Testing plan</b></li> </ul>	<p>The contingencies that were concluded which were relevant to the project were losing a team member and power outages. Losing a team member was considered to inherit a medium risk due to it being unlikely but having a major impact, with its mitigation plan being the inclusion of tolerances in the timescale in expectance for such circumstances. While power outages had a low risk as it is not only unlikely that it would occur but also due to emergency generated protecting the databases, however, in order to further mitigate the issue, a simply process of backing up the data is the easiest and most effective plan of action that could be taken.</p> <p>When it comes to the constraints, time was given an extreme risk due to the rather stressful timescale and the impact being high if the project was not completed as it would be useless, mitigation for this was mainly prioritization and proper timescale planning. The scope was given a medium risk due to the number of new concepts that would have to be learnt, however, it is unlikely as the team members already have a background in programming so it should not be that big of an issue, mitigation of this was mainly a replacement plan in the case that something was too difficult to implement. The budget was decided to have a rather low risk as there is very little in terms of budget that needs to be bought, and going over budget slightly should not be a significant issue even if it did occur.</p> <p>Finally, major solutions were also chosen by the team in the case that the website could not be completed whatsoever, with said major solutions being utilized visualized data and using WordPress rather than C# and Blazor due to WordPress's simplicity but inherit disadvantages.</p>

<b>Date: 19/10/2022</b>	<b>Cybersecurity meeting (meeting 7)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• <i>Malware mitigation</i></li> <li>• <i>DDoS mitigation</i></li> <li>• <i>Account takeover fraud mitigation</i></li> <li>• <i>Testing plan</i></li> </ul>	<p>This meeting was rather long and basically the team went through all the cybersecurity that would be required to ensure that the website is safe from potential malicious attackers. For this, a friend and an associate of the team, Abdulrahman Al-Ghammari, was also invited to the meeting due to his background in cybersecurity, specifically in the red team of the field. The main topics that were discussed were methods of protecting the website from malware attacks, method that could be implemented to mitigate DDoS attacks, and finally protection methods from potential account takeover fraud were discussed. From there, a testing plan was then created to not only test the security of the website, but also its overall quality.</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<ul style="list-style-type: none"> <li>• <i>Design the layout of the dashboard</i></li> <li>• <i>Document the design</i></li> </ul>	<p>When it came to the malware mitigation, upon research, it was decided that the optimal way to try and mitigate potential attacks is to use a WAF, restrict entry fields, and check for any vulnerabilities that could be found in opensource components. In terms of DDoS mitigation, simply using Azure's DDoS protection method seemed sufficient enough, however, for extra protection, it was also decided to implement an IP request limit per a given time. Regarding the account takeover fraud, simply using two factor authentication should be sufficient in protecting ensuring user authenticity.</p> <p>Finally, the testing plan created revolves around initially testing the rank feature functionality, the graph algorithm, the overall loading times, and the general usability of the website, with the usability testing involving several people that have never seen the website trying to extract certain information from it, and depending on the time they take to do so, the team would be able to know if they should improve the UI or not. After this, security testing is going to be done, which includes testing for potential vulnerabilities in opensource components, testing the implemented DDoS mitigations using an Alpha Adapter, and finally, having a simulated breach attack to try and find out if there are any vulnerabilities remaining that need further investigation.</p>

## Design meetings:

<b>Date: 26/10/2022</b>	<b>Full design concept meeting (meeting 8)</b>
<b>Agendas:</b>	<b>Discussion</b>
<ul style="list-style-type: none"> <li>• <i>Design ideas brainstorming</i></li> <li>• <i>Choice of designing software</i></li> <li>• <i>Designing the dashboard</i></li> </ul>	<p>Although the team had a design concept in mind right from the beginning, it was not necessarily production ready, and as such the team had to brainstorm varying ideas and layouts for how the dashboard should look like. After which, the team started with discussing the software that would be utilized for the designing stage, which ranged from a couple of different platforms, some having been used before by members of the team while others haven't but are big competitors in the industry. Finally, the last part of the meeting was actually designing the dashboard together.</p>
<b>Action points for the next session:</b>	<b>Conclusions</b>
<b>N/A</b>	<p>The team went through four different layouts before coming to the final one chosen as the design concept, which includes two pie charts, one of which displaying the rank of the student among their peers while the other displaying the potential grade. A line graph was also included showcasing the student's average grade throughout the year, a bar chart displaying the grades of all the potential grades all subjects, a radar chart representing the student's grades over various aspects of their studies such as tests, lab work, homework, and son. Finally, another pie chart displaying the total amount of grades and where exactly coming from (i.e 5% from homework, 60% from exams, 30% from tests, and so on).</p> <p>After deciding on the layout, the team then initially used Figma to draw the layout, which was rather unsuccessful, from there, an attempt at using Canva was made but that also did not work out as intended, at the end, Venngage was chosen, and the design concept was finalized on it.</p>

*“Anyone who has lost track of time when using a computer knows the propensity to dream, the urge to make dreams come true and the tendency to miss lunch.”*

- Tim Berners-Lee

Inventor of the World Wide Web