Hayley Sharpe - Scholarship Programming Project.

Context

My dad owns an office building in Riverlea, which he rents out to tenants, both individuals and companies. Given he is not there all of the time, his tenants need keys, which they can use to access the building when they enter, leave, and after hours. Dad currently uses a spreadsheet to store all of the information about the keys he has, such as the person who currently has it, the colour of the key, and more, however he is finding that it is a hassle to use the spreadsheet format for this. This is his problem - he lacks a secure system he can use to store and search through all this information. Therefore, he wants a digital outcome that he can use to hold all the information about the keys, and keep track better of who has which key at any given time.

Initial contact with client - Dad

What purpose do you want the digital outcome to serve?

I am involved in handing out and tracking keys for tenants in a commercial building. It is important that the keys are controlled for security and cost purposes (in case keys are lost, stolen, etc). Would be good to know at any time, who has what key. At the moment, I use an excel spreadsheet to store it, but it is tedious - having automated software would make things a lot more simple.

How does key storage/distribution work within your building?

There are multiple tenants, each tenant needs a physical key to enter the building at the beginning/end of day or after hours. When tenants join or leave being a member of the building, I need to be able to provide a key or have the key returned. Whenever I hand out a key, I try to ensure I have the key number, colour (if not silver), company name, and date that the key was provided, and records on a Spreadsheet. When returned, I go back into the spreadsheet and black out that info for that person. This is hard to keep track of - a database would be more useful (what I assume), but ultimately just want a simple, accessible, up to date method to record key numbers and to record keys against tenants and their names.

Project constraints

Time management - Dad is very busy - he runs a company in the USA, is a board member of a large NZ company, and is regularly involved in multiple small business ventures, to just name a few things. Therefore, he does not have much time to fiddle around with figuring out software or exploring different alternatives to help him solve his problem. He also does not have time to, if he found an alternative, transfer all of the data he has over. Therefore, any software that is set up for him would ideally already have the data in it, and/or make any new additions a quick and easy process.

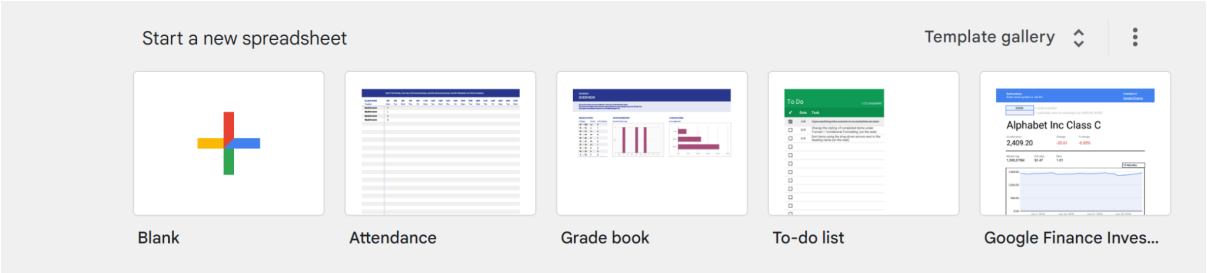
Mobility - With Dad's intercontinental business endeavours he needs software he can use anywhere and any time. He uses multiple devices for work - a computer, desktop, iPhone and iPad - therefore the software needs to be versatile enough to run on multiple devices with various screen sizes and operating systems.

Investigating existing products

Excel

Excel is currently what Dad uses to store all of the data he has. He likes using Excel as all of the information is visible to him. He uses Excel frequently and is very familiar with the software, so has no problems with it whatsoever. The largest issues he has with Excel are how rigid a spreadsheet is - meaning if he has any exceptions to rules, it cannot cope. He also often has to go through and delete data from one person and add to another rather than just replacing information on one person - meaning he is spending a lot of time double handing data. There is also no efficient way to search through all of the data he has inside the spreadsheet.

Google Sheets



Menu to open a new spreadsheet.

C9 *fx*

	A	B	C	D
1	People	Keys	Key colour	Company
2	Joe	Key 2	Purple	Apple
3	John	Key 7	Pink	Facebook
4	Phil	Key 1	Yellow	Instagram
5	Susan	Key 3	Silver	Google
6	Matthew	Key 4	Silver	Google
7				
8				

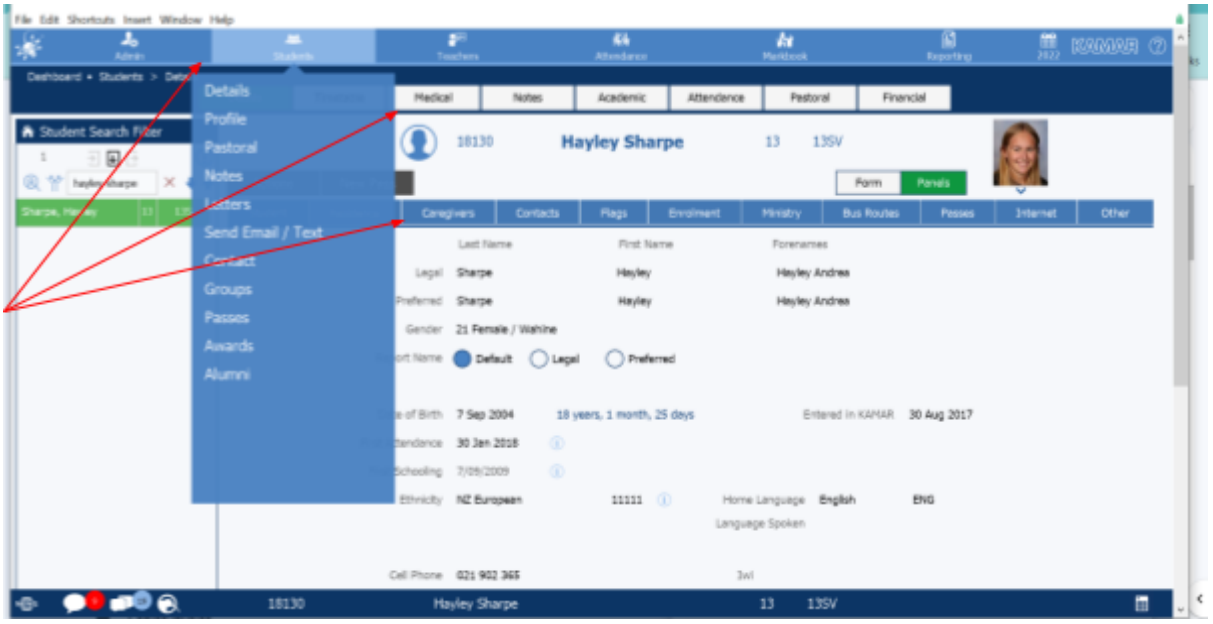
Mock up spreadsheet for Dad's office.

Google Sheets is similar to Excel, as it is also a spreadsheet tool. However, this can be accessed from any device, as the Google Sheets app can be installed on his Apple products using the app store. There are still drawbacks to Google Sheets. One is that Dad's primary email is an old company email which was not set up through Gmail. This means that the software is not entirely compatible with his devices, and so he would have to use a separate email, or deal with some of the issues that can arise from using a non-Google account for Google products. The other drawback is

that it is still a disorganised system in a similar manner to what he complained about with Excel, which was annoying for him and very time consuming.

Kamar

Kamar is the database system that most NZ schools use to store and view student information. Though this database is very complex, probably to a much higher degree than what I will be creating, it is still relevant to my intended project as it is a database that is used to track relational information about students.

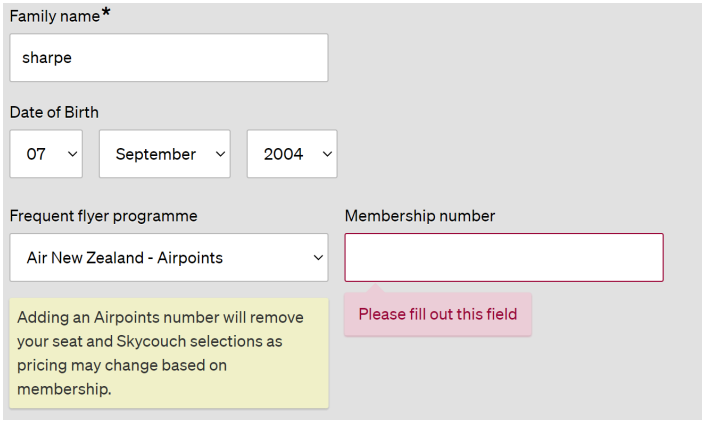


Above is a screenshot of my student profile on KAMAR which teachers have access to. Looking at this you can see the visual clutter that occurs. There are three rows of options available when looking at a student's details, and within each of these top rows, there are drop down options. The amount of information being displayed to the user is so visually overwhelming and with such minimal captions on some of them, it is quite unclear what will happen when they are clicked on. Even when navigating to this page we found that KAMAR was slow to register clicks and took a long time to load. Though there are massive amounts of capability with this software, its very low levels of usability would make it not a suitable tool for my Dad and his busy schedule.

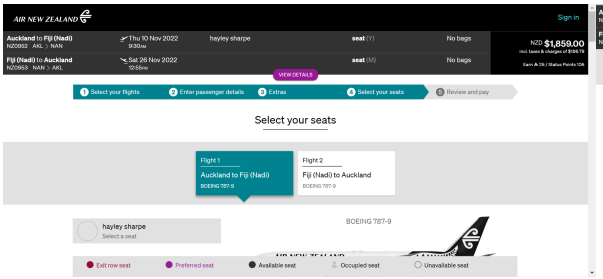
AirNZ

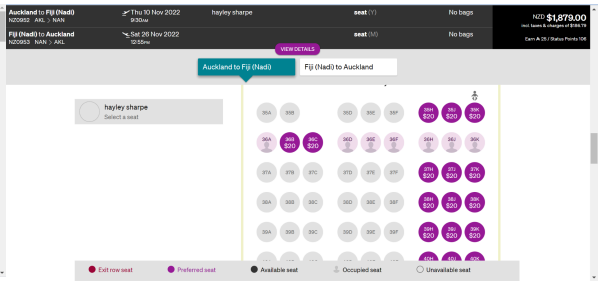
I decided to test the AirNZ software by using it directly, as if I were to book some flights. The website has a nice layout, where it is not too visually overwhelming. However, I found myself coming across some issues while in the process of booking a flight.

Error message while attempting to book.



The first issue was during the booking process. I selected being a part of the frequent flyer programme however didn't give a membership number, as it wasn't shown that it was a required field (as it did with my last name). However, it gave me the above error message and stopped me from submitting the form. The error messages were not particularly helpful in diagnosing the problem.





The next problem I had was when I was trying to book seats. The content does not fit adequately on the page, as can be seen in the left image. The right image contains the view I had after scrolling about halfway down the page. Not having everything on the screen when it is opened adds to the time taken to use the software and decreases the usability. Though the visual components were nice, this would not be suitable for Dad as it does not meet his needs for a fast paced and simple mechanism which would help him diagnose and recover from errors.

What products could I potentially create?

- Database. This will most likely be necessary, as the outcome I am creating requires long term storage of information, and therefore it will need to be kept securely. He currently uses a spreadsheet - which is kind of like a database. However, a standalone database does not serve much purpose without some sort of interface for a user to interact with - this is because they can be hard to navigate meaning some potential end users may not be able to understand the technology. This would prevent it from being fit for purpose in the broadest sense.
- Desktop application (using C#) - this would be good as it is what I have used before - I am most familiar and therefore would have the most capacity for creating a complex outcome with this software. The issue with desktop applications is that they are confined to desktops - so he would be unable to access the program from his ipad, which he uses a lot for on the go.

- Website - websites are versatile in that they can be accessed anywhere, meaning he can use the program whether he is at home with his computer, or away with a laptop or ipad. Websites are something I have not worked with before though, and therefore it would be much harder to complete such a difficult project in the confined time frame I have. There would be a massive time investment required for me to understand how to create a website - which I am willing to do, however I may be limited in that my time is not infinite, and therefore could be limited in how far I am able to progress with the project in this limited time frame.

Other applications

Apartment landlords, or business owners, or even other building owners like Dad could use a similar system for their own management of keys and where they are currently. Even people like school caretakers could use a system like this for tracking the keys on loan across the school, or teachers at school who loan out equipment, such as my DVC teacher, who loans out drawing supplies, or even on a much larger scale, the librarians who have many books on loan. I have also found that I myself could use a database management system for my own school leavers gear committee. Being a head student at school means that I am in charge of managing and distributing leavers gear orders to my year level. A project like this could really help me to streamline the process and safely store all of the information of our students so that no people are left out or forgotten. Similar processes could be applied later in the year for awarding tickets to students at the annual Leavers Dinner.

Reflection

I have decided that it will be best to create a **website** which uses some sort of inbuilt data storage/database. Despite how easy it would be for me to create a desktop app, this will not allow him to access his data on the go, making the project not fit for purpose in the broadest sense. I also cannot create an application as this cannot necessarily be used on a computer. Despite my lack of prior knowledge around what is required for building websites, I hope that I can find some way to use the knowledge I already have in C# or the experience I have gained with some of my prior projects to help me build a successful outcome. Creating a website may require finding a compromise between complexity of outcome vs time spent on development, meaning in some aspects, it will be very fit for purpose, whereas others it may not be quite so suitable. Therefore, I hope to prioritise the most important aspects of my outcome, based on the specifications I create and the needs of my stakeholder(s).

Brief One

Conceptual Statement

I will be developing a website for my Dad (and potentially any other clients) to use to store and manipulate data about keys. It will be situated on my personal computer when being developed, and then when finalised, would be able to be used on any device of Dad or another client's including a phone, tablet, or computer.

My idea should be developed to fulfil Dad's needs more so than an excel sheet which is incredibly disorganised and hard to follow. Building this app should allow him to have a much more streamlined process when adding or updating data and also when looking through it to find the things he needs. It should also allow him to access the data regardless of where he is and what operating system he is on.

Specifications

Uses permanent data storage.

- Pre-populate the data for Dad before he begins using it
- Allow functionality to update, delete or add records

Versatile.

- Needs to be functional on multiple devices, and on the go.

High usability.

- Streamlined and efficient process for quick access and updating
- Organised to suit his needs
- Well tested and efficient

Stakeholders

Dad - primary client.
Murray - school caretaker.
Mrs Stonestreet - DVC Teacher.
Mrs Langman - programming teacher.

How will I maintain contact with my stakeholders?

Dad is at home, so easily contactable. I'll have conversations with him when I'm home, and ask him as I go. The other stakeholders I may end up having, such as the school caretaker, I will email for initial contact, and then aim to meet with them where I can for further feedback and review. This should mean that at all stages of producing my outcome, I have ongoing contact with at least one of my clients.

How feasible is the above outcome?

The outcome doesn't seem that far removed from things I have done in the past, such as the nutrient plan I made earlier this year as an internal for 3DTP, or the mental health tracking app I made last year as my major project in 2DTP. That being said, I have certainly never created anything that uses permanent data storage, so this is definitely a skill I will need to learn. I may also need to look into data security if this site will be accessible to people other than Dad (website, etc) - as he should be the only one who is able to edit the keys and their owners. If tracking when people

have owned the keys in the past is also important, some sort of calendar or timing system I will also have to learn.

The outcome seems to be scalable to the timeframe of my project. Additional functionality could likely be added, but the outcome could also be quite basic, as at its core, it is matching keys to people. Therefore, the size of the outcome will be dependent on how much time I have to complete it, and how I choose to invest this time.

This also means that should the outcome contain enough complexity, it could potentially serve a greater field of people than just my dad.

Implications

Where will my project be developed and used? (social and physical environment)

The outcome will be used by Dad, however this could extend well beyond just him. Many people would want to have some sort of system to manage keys, or even other objects. This could extend to the school caretakers, who manage lockers for students and keys for teachers, or myself, for leavers gear. Therefore, there are other users who could be able to benefit from a project like mine; this means the scope of my project may need to encompass more than just the niche area that is office buildings, specifically Dad's.

Depending on the digital outcome I choose to use to develop my project, I may be very limited in where I can develop my project. Desktop applications (developed in C#) I would be quite limited, as I only have a Chromebook, therefore I can really only do it at school. A database however I am unsure what the scope of this may be. I imagine it is likely that I'll have to use multiple platforms for the completeness of my project.

It will be used by Dad at his office, home, or abroad. Mr Burton would likely use it at school, and I would use it at school or at home, depending where I see fit. Given it is a website, it will need to also consider differently sized browsers and screens.

Resources

I am unlucky that I have a chromebook and therefore most likely will not have access to the programs I will need all the time. Therefore, I will likely have to work at school, or at home if I am able to access what I need off of Dad's computer, to complete the task. This produces an issue as with time being such a restraining factor in my project's success, I will have to ensure I am working hard at school when I have access to the hardware I need, to produce an outcome as successful as possible.

With Dad wanting to use the outcome on multiple devices and OS I may run into the issues where my project is not easily accessible from many devices. I will have to ensure that regardless of the screen size and OS, my program can still be used easily with no struggles.

I have software requirements that I can use tools that I understand and will be able to construct an entire project on, while my stakeholders will require software with enough complexity to fully satisfy their needs. This may produce another problem as often the easier software to use lacks the complexity I may need; therefore finding software that provides a good balance may be an issue.

What management tools will I use?

I have chosen to investigate various project management tools to help me to manage what may end up being quite a large task. There are many options for which I could manage my project. These include waterfall, the V model, spiral, iterative/incremental, RAD, and Agile models. All of them have various pros and cons.

The agile models I've explored are scrum and Kanban. Kanban is a very physical approach - people use sticky notes and whiteboards to visually depict how the components of their product could be laid out. This is very positive for keeping a larger team motivated on a main goal - but not so ideal for long term projects where the timeline is not easy to stick to with so many small tasks.

There is also the scrum method, where from a large selection of tasks, 'sprints' are set up. The sprint contains critical tasks to be completed in, say, a week, where as the task is completed, it is moved forward into a 'done' pile. The only struggle with this is that as stakeholders provide more requests for functionality, there can be too many tasks and the final deadline can be overlooked. This I know would not be an issue with me as I don't have a choice on the final deadline - I would continue creating my outcome, making sure to put the most important things in my print before starting to add additional functionality. Therefore, I will use the Agile scrum model of project management to continue to manage and monitor my project. This is also the most relevant to me as I have used scrum-type project management before, so I will use minimal time learning how to navigate this type of project management.

The software I plan to use for my project management is Trello. Trello is often advertised as a Kanban alternative, but having used it for agile project management before, I have managed to use it as a scrum-type alternative. It is modelled to Kanban in that you can move tasks like sticky notes along a board, however I can make an extra column (something like 'this week'), which would allow me to create a sprint like in Scrum, where I just move tasks from a backlog into a to do column.

What are the constraints I have in completing my outcome?

A large constraint will obviously be **time**. This could largely be a constraint as the amount of the project I am able to get done is very dependent on how long I have to work on it before the end of the year.

Access will also be a constraint due to my limited access at home. I have only a chromebook and so cannot code at home, therefore will rely on the school

computers to do so. Therefore I will need to use the school hours I have effectively and potentially come in outside of class to get work done as I see fit.

Limitations surrounding my **knowledge** and confidence to complete this outcome may come into play as I decide what software to use to complete the project. I can already see that this project is outside the scope of anything I have done previously or would know how to do with where I am at. I have fears that I will lose confidence in my ability to get it done in time, which will in turn make me less motivated to even try, and also that my lack of prior knowledge will render me unable to complete it altogether. At this point I hope to lean on the feedback I gain from others to ensure that I stay motivated, and also to call on experts should I need help, or supplement my knowledge with further research.

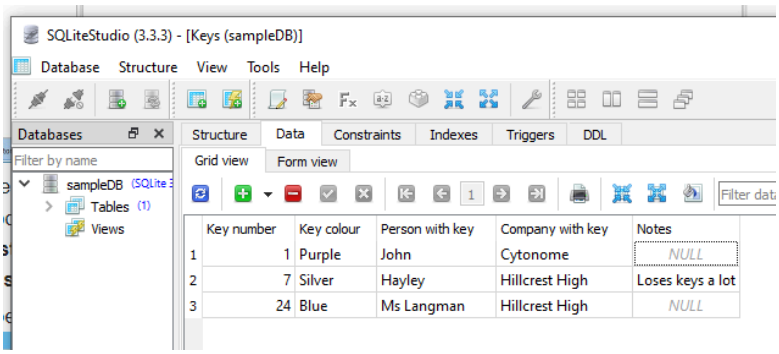
There is some rigidity with **various software**. For example, I know that creating desktop applications with C# means that I can have little to no visual components, whereas using MIT App Inventor, I can only create mobile phone applications. Therefore, I hope to find software that allows for use on any device, and some freedom in the interface itself.

Ongoing maintenance is another potential constraint in my outcome being complete. While I hope to leave it off in a finished state, there is always the possibility and therefore need for future maintenance, as if there is no way to maintain it, it won't stay in a completed state. I plan to use good naming conventions and comment on my code so that should anyone need to maintain it, the code itself makes sense and can be understood. I also want to use up to date software to ensure it

Initial Research into Software Platforms

In carrying out this project I knew I would have a lot of work to do, particularly surrounding finding out how to go about data storage and editing software, which I had no experience with programming. I decided to do some independent research, as well as reach out to some experts who I knew could help give me suggestions as to what I could do.

SQLite



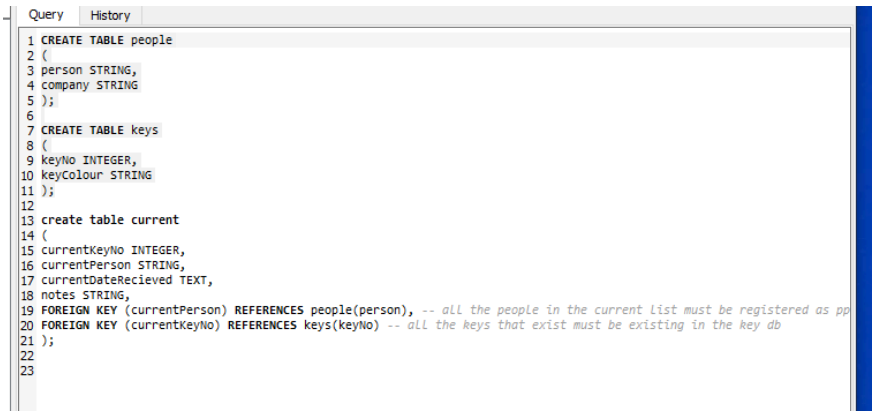
One of the possible tools I found that I could end up using was SQLite Studio, which is a database tool allowing me to create a database to use.
I decided to make a mock up of what a database for Dad could look like, using all of the information he had provided me, regarding the different things he wanted to store. I started by

completing a sample database which may end up looking like something that Dad would use. In creating this, I thought about how filling the database could be done, realistically speaking. Would the end user fill their database in this SQL format, which would be difficult for a non-programmer to execute yet definitely possible; or is there some way to automate this process for the user, so that they just have to add in the details themselves? This is probably a good thing to ask Dad - will you be accessing the program frequently enough to want to have it automated, or would you prefer your database is already set up, so that you don't have to make this yourself?

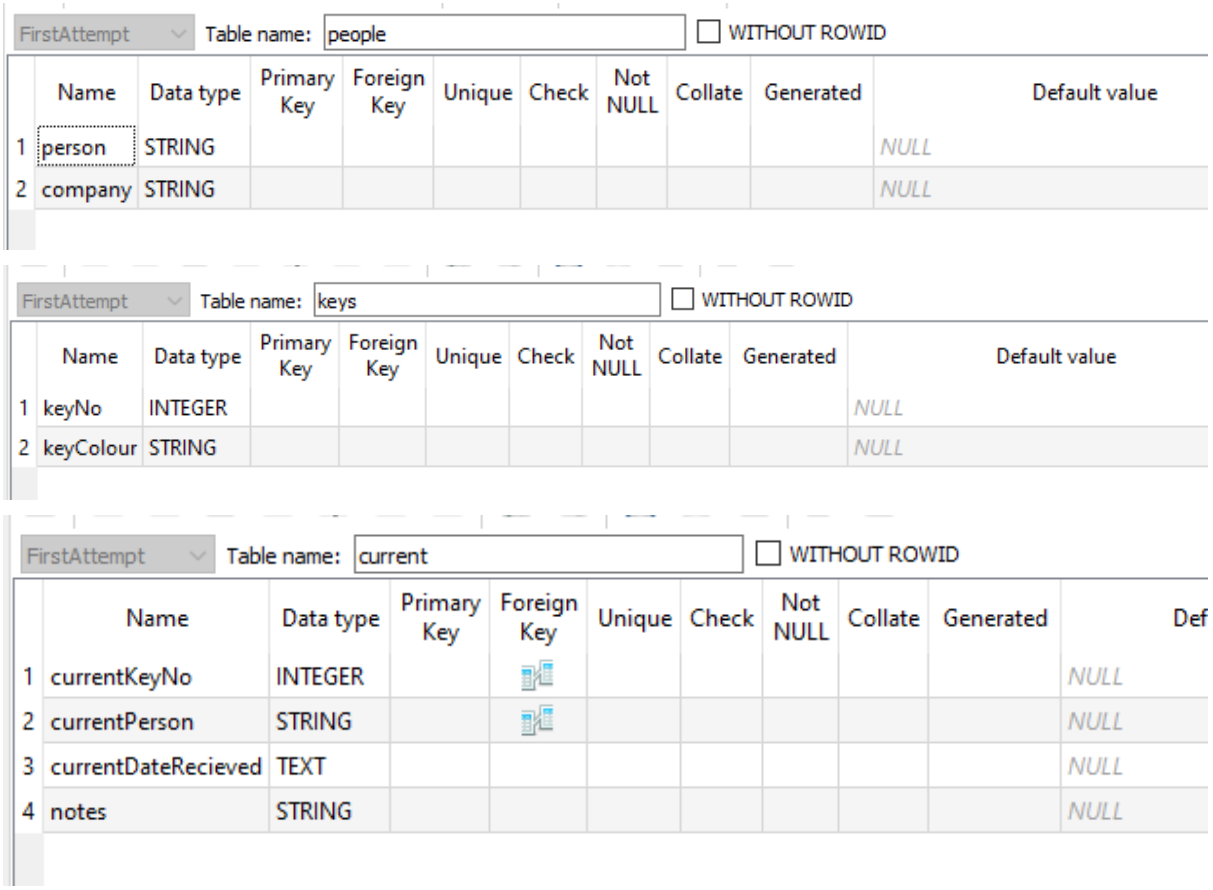
I asked him the following week what he would prefer, and he said that he likely wouldn't access the database frequently enough to want to change it heaps - so he'd prefer it was just set up, and that he could search an already filled database. Therefore, it seems it should be a priority first and foremost to create a database that can be easily viewed or searched, with the additional functionality of searching so that should he need to change anything, he has this ability, making it fit for purpose in the long run.

I also found this tutorial, which is useful for syntax. I then followed through the tutorial again, this time using my project as the focus rather than their data.

From here I began building what I would hope is the actual database that may get used if I do decide to go through with SQLite, alongside the tutorial I had used previously to help support this. I have to be careful in building this table as according to this website, it is not possible to edit constraints (for example, you cannot change not null constraints to nullable) after building and committing changes to the database.



Following the tutorial I had, I managed to mock up what I think is a pretty successful first solid attempt, and first trial. There are 3 tables, each of which storing unique things - info about the keys, info about the people, and information about the current links between these keys and people. I chose not to add in my fourth table for now as this was going to complicate things prematurely which was not necessary.



FirstAttempt Table name: people WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	person	STRING								NULL
2	company	STRING								NULL

FirstAttempt Table name: keys WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	keyNo	INTEGER								NULL
2	keyColour	STRING								NULL

FirstAttempt Table name: current WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Def
1	currentKeyNo	INTEGER								NULL
2	currentPerson	STRING								NULL
3	currentDateRecieved	TEXT								NULL
4	notes	STRING								NULL

Foreign keys are what links tables together, and I chose to use them in this way so that anyone who had a key must be registered in the people log, and any keys must be registered in the key log.

Reflection

Obviously with SQLite there are some drawbacks. The database is quite difficult to edit once things have been initialised, meaning that for someone who makes many mistakes, it is easy to stuff up small things. It also would mean that, given I don't yet know how to link this database to editing within the website, if Dad or Murray had any changes to their key system, they would have to go into SQLite to change things around (unless I managed to figure this out). This is also a large undertaking as it really does not simplify the process for either of them, as well as that it assumes they have some knowledge in database construction, which I could not assume they'd know how to do, or expect them to do - that's what my program should do.

There are also some positives however, being that SQLite is fairly easy and quick to set up. You can type directly into the tables, requiring minimal knowledge on my part to get the database up and running. This is helpful for the time constraints on my project. However, a quicker solution for myself is not able to justify a massive lack of convenience for my clients and therefore I don't think SQLite is necessarily the answer.

Mrs Langman

Within my research I knew I would need to contact some experts within the field for some guidance. The first expert I chose to communicate with was my programming teacher, Mrs Langman. Though she also had little experience with data storage and collection, she had contacts with other teachers from other schools, who she was able to contact and ask for resources. She forwarded me the following information:

if you're looking to build some kind of web app then I have made a few tutorials around using Python (with Flask) which is something our senor software engineering students do - here is a link to one I did not too long ago based on a possible level 2 project: tinyurl.com/24fdrrv4

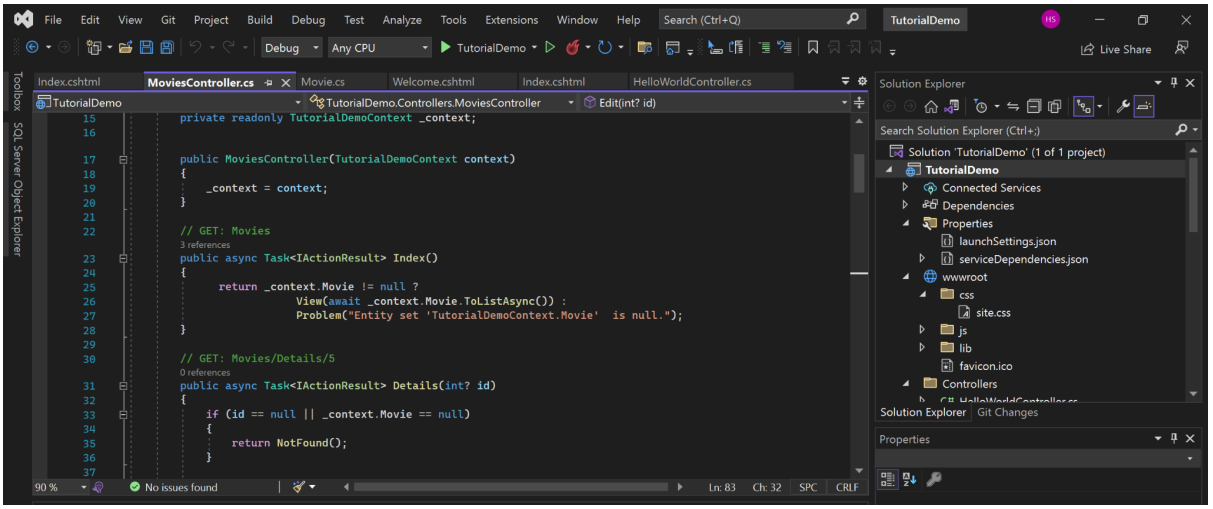
…students use SQLite3, Python and HTML/CSS to …design and build their own database …

I took a look at the tutorials I had been forwarded and discovered that they used Python, a programming language I had never used, and Flask, which is essentially a framework for web application development using the Python language. Looking at the documentation provided, I found that the reason that the teacher uses Flask is because “the main advantage of a Python-based framework is that most of us are already familiar with Python, and teaching it.” Despite this there was a very thorough set of documentation attached to the tutorials which could help me step through a tutorial. Though it was thorough, I found myself struggling to understand some of the jargon associated with website building as this was all very new to me.

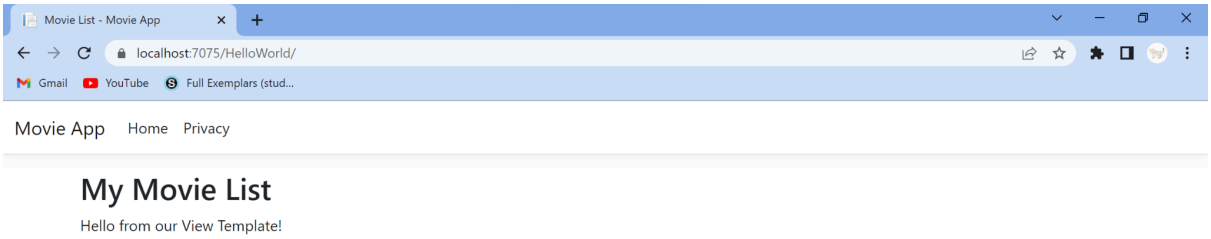
ASP.NET and MVC, and Blazor

After doing some googling I decided to begin investigating ASP.NET. This is a web application framework which uses Visual Studio. I know other students have used it in the past, therefore I trust it may be able to carry out the tasks I hope to execute. I have absolutely no experience so chose to look into some tutorials.
https://docs.microsoft.com/en-us/aspnet/overview
https://docs.microsoft.com/en-us/shows/asp-net-site-videos/making-websites-aspnet
The video tells me that WebPages are likely good for a beginner (like myself!). It also shows how to actually create the database within the video, as the man is able to fill out the data table himself.
Having watched the video I decided I needed more practice with the database software I am using. I therefore decided to find a tutorial that I could potentially build my project upon, as it seems to be similar in content to what Dad wants to do - search through a database for the values he wants.
The tutorials I decided to use:
Searching tutorial
Easy beginners tutorial

I also decided it would be necessary to familiarise myself with the whole thing that is the .NET system. I watched the following tutorials to get a grasp on the concept:
https://www.youtube.com/watch?v=lE8NdaX97m0&list=PLdo4fOcmZ0oW8nviYduHq7bmKode-p8Wy
https://www.youtube.com/watch?v=JfnTG955cuk&list=PLdo4fOcmZ0oW8nviYduHq7bmKode-p8Wy&index=3
https://www.youtube.com/watch?v=eIHKZfgddLM&list=PLdo4fOcmZ0oWoazjhXQzBKMrFuArxpW80&t=104s
This helped me understand that there are actually two distinct things here which I was mixing up - ASP.NET and .NET Core. .NET Core is a fairly recent addition to Visual Studio (2016), which I would tend to trust more than the older, clunkier ASP.NET. I am choosing to continue with .NET Core as this is where I am able to find the most tutorials and resources, which will help me effectively develop my project as soon as possible. .NET Core also seemed to have a lot more specified pre-made templates, which would enable me to have the website set up for my specific purpose.
An aspect of .NET Core that tended to stick out to me while researching for which template to use was .NET Core MVC. This is what a previous Hillcrest student, who I am family friends with, used for a fairly similar website to my own, and also what makes sense for a data storage site. This also meant that I could reach out to him should I continue with MVC if I needed help from an expert. MVC stands for Models, Views and Controllers. Models essentially represent databases inside Visual Studio, Views are the page that the user sees (the interface), and Controllers are what link these things together. This seemed to make sense as I wanted to be able to have user interaction with databases, which were stored inside Visual Studio. I therefore downloaded the 2022 version of Visual Studio and began to follow through this website's tutorial:
https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-6.0&tabs=visual-studio.



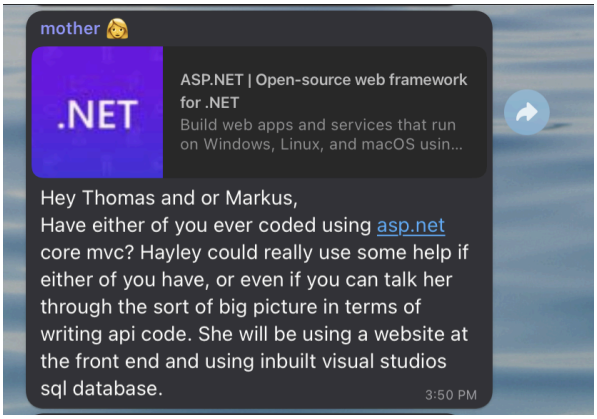
A snip of some of the code I worked on through the tutorial. This is for a controller.



A separate page within the site, as indicated by the /HelloWorld extension.
I soon realised the major confusions that came with MVC in having so many components to carry out one simple function. This was not the C# console apps I was used to, where code executed in order and when called; rather things were happening that I was not even aware were functions that were written into the app for me. I also found it confusing knowing where to put things and linking the three components together exactly how I wanted.

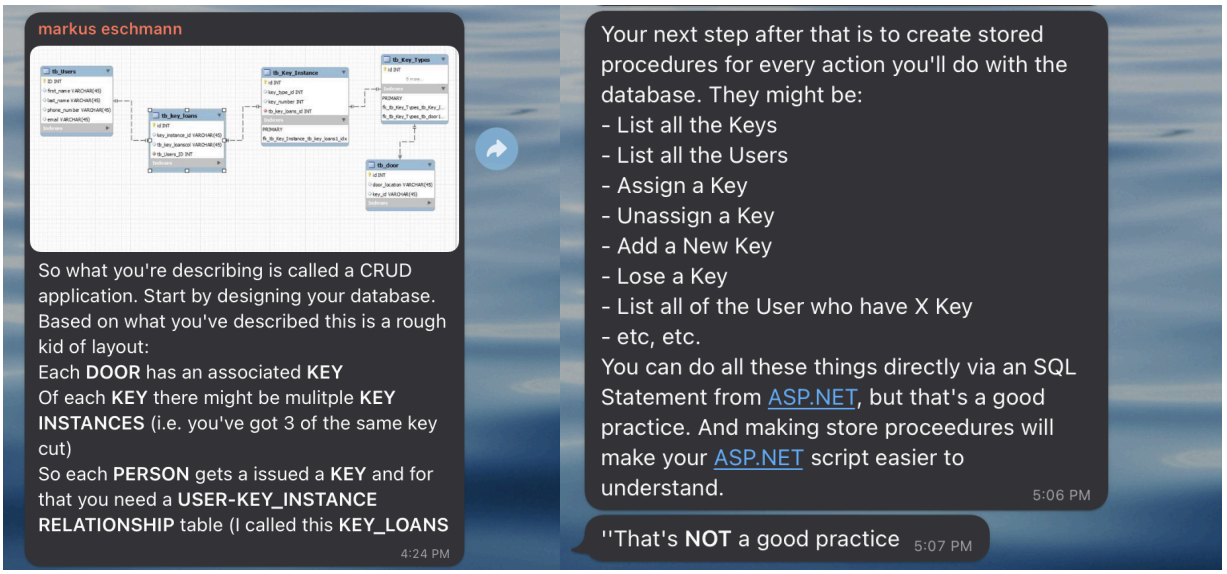
Uncle Markus & Thomas

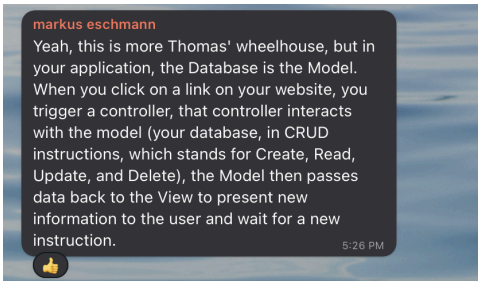
The next people I chose to contact were my two uncles. Both of them are involved in computer programming - one of them (Markus) started a computer science degree at university, and has had some small side hobbies including building websites similar to Facebook; the other (Thomas) builds hotel booking websites for a living; so I thought that either of the two could give me some insight into what I needed to accomplish. Below is the message we sent to the two of them:



They helpfully responded with some suggestions.

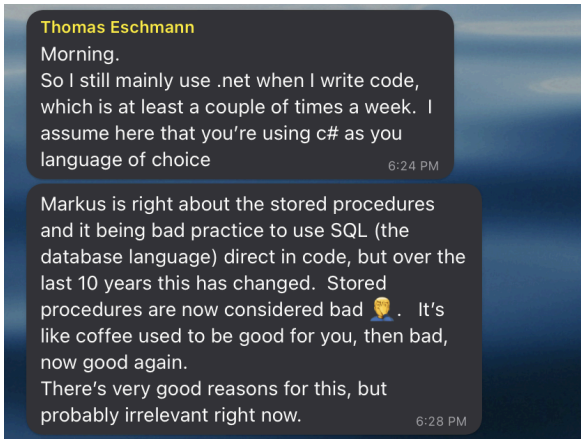
Markus:

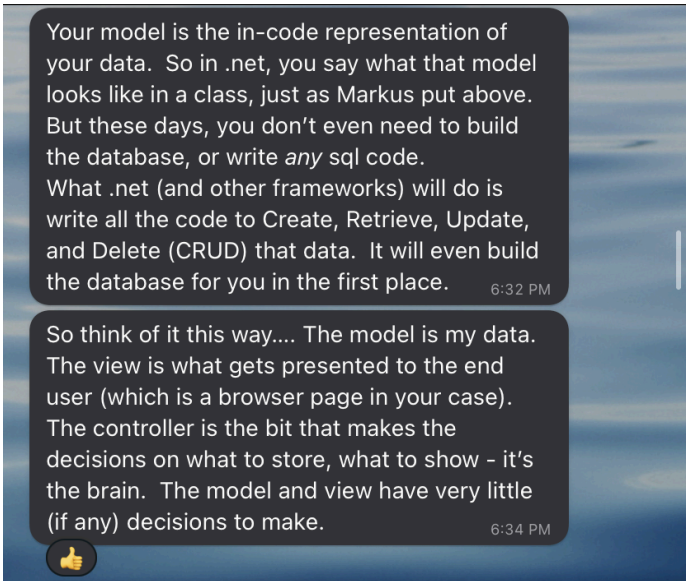




Uncle Markus was able to explain what the system I had in mind would look like in a MVC context. He also discussed the CRUD application format, which essentially are the four basic operations of permanent data storage.

Thomas:





Uncle Thomas essentially confirmed all that Uncle Markus had to say - however he was able to elaborate one step further, mentioning that I wouldn't need to write any SQL code to set up the database, and that in .NET, it sets this up for me.

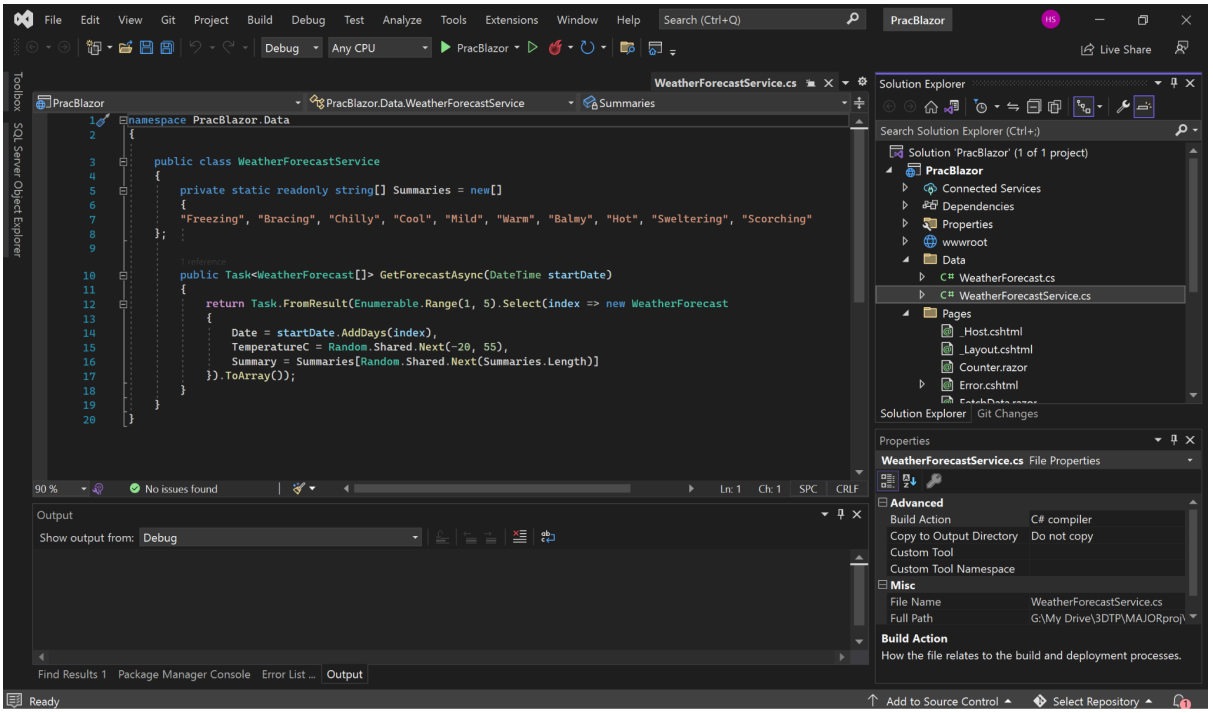
Both of my uncles were able to provide me with some clarity on MVC, and additionally, Uncle Thomas offered to jump on a video call with me to help me get set up with the beginnings of my project.

At this point looking at MVC, I felt like I was really stumped. I had invested a lot of time into understanding these confusing terms surrounding .NET Core and MVC and ended up just as confused if not more after following through the tutorial. Despite talking to my uncles, it seemed that neither of them had the most experience with MVC, and both had preferred software of choice, neither of which chose to use MVC.

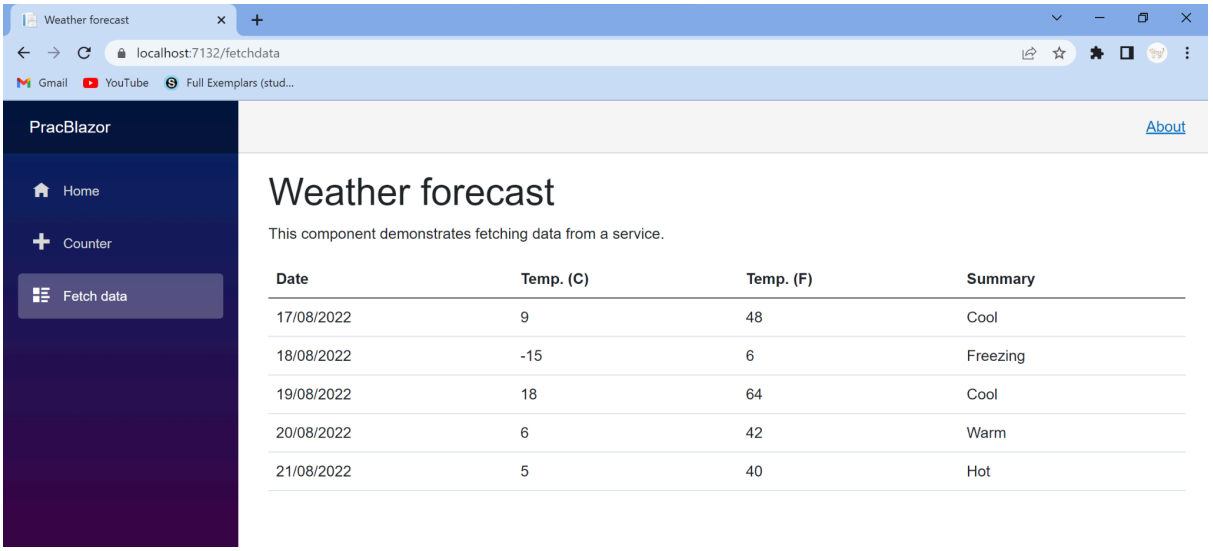
When I jumped on the call with Uncle Thomas, he suggested that I investigate Blazor, which is another web development tool. The difference here is that .NET Core MVC uses models, controllers and views - therefore 3 separate components - to set up pages within the website. It also uses a combination of Javascript, C#, HTML and CSS languages. In Blazor however each webpage is designed in a much more straightforward way (in my eyes), where I am not needing to combine things like models and views and controllers, and rather can put the content of a page directly into a page format, and program the background executions separately. It also uses only C# and a bit of HTML for UI, and so it will take a lot less time to configure.

Blazor

With the help of some youtube videos, I have completed a basic tutorial in Blazor server side. It appears to be much more straightforward than the MVC I had already trialled.



A snippet of the tutorial of Tim Corey's I followed. This sets up a basic web page (below).



The numbers and words are randomised, whereas the dates are accurate to real time.

Having watched the Tim Corey videos, I jumped on a second call with my uncle. I had thought I would build the databases and then add them into the project, but it actually turns out I can get Dad and/or Murray to set up the databases in the website, if I build it to have such functionality. This works as while things are added into the database, they need to be displayed also - this is just an added level of functionality which I can provide for either of them.

Reflection

The resource I had been provided by Mrs Langman was very thorough, and seemed to be appropriate to what I wanted to achieve. However, I knew that the time investment required to make my outcome work with this framework would be massive. Not only did I not know any Python, there was no one around me who knew it either, meaning that even when faced with bugs, which are inevitable, I would be on my own trying to solve them in an unfamiliar language. It also caused me concern to go fully in with Flask/Python as if I got to the point where I was fully stumped on what to do to fix a bug, or found I could not move on with the project, I would have invested so much time into it at that point that I'd find it very difficult to start over again with a new format.

Giving up on the MVC I have been working on is a difficult prospect, as I have invested a lot of time into researching MVC so far, which will only be useful to some extent for me. With Blazor, once I get the hang of things, I imagine I will be able to work a lot faster, especially with the experience of my uncle which I can call on if I get stuck. However, it will be essentially starting over with my project - despite having spent so much time already on understanding MVC and SQL.

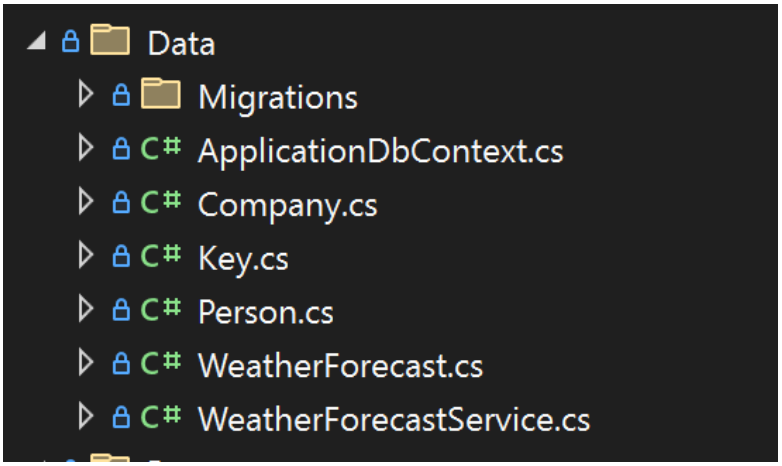
All the above being said, I think it makes a lot more sense for me to use **Blazor**. From the tutorial, which seemed to make a lot more sense than the MVC one, to the ability to put the data into the website itself and have Dad and Murray custom build their database, the possibilities I have with Blazor are much more fulfilling in such a

short timeframe. I also have much greater access to the help of experts (Uncle Thomas), meaning that I have much more confidence in my outcome being successful.

Inbuilt data storage

The other thing Uncle Thomas and I discussed on our initial phone call was about using the inbuilt Visual Studio database building software, where rather than setting up a database in advance, I could choose to insert the data I have into VS and they set up the database for me. This is obviously very efficient, however there are some major downfalls. One is that the tutorial I have been using surrounding Blazor does use an actual pre-built database, which is added into the project. The other is that for my project to be fit for purpose in the broadest sense, it cannot be built strictly on Dad's hard coded data since Murray is also a potential client of mine. Therefore, I would need to build the two databases separately, and then create a more flexible version of the program which will allow me to ensure that my project is fit for purpose in the broadest sense. Alternatively, I could create two separate instances of the project (with any small modifications that either of them may need), allowing each of them to have access to their own uniquely made project, making it better fit for their specific needs.

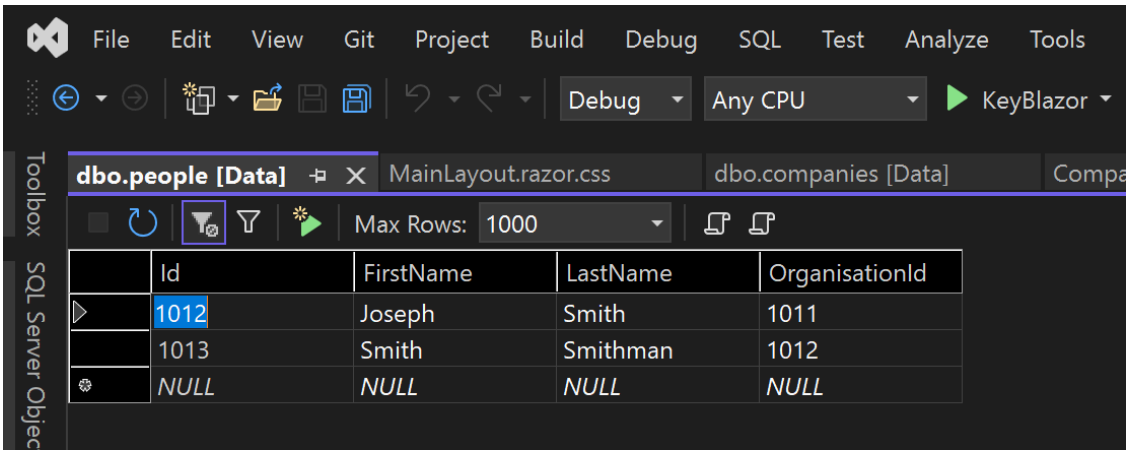
Creating the databases is not a difficult task. The databases are set up inside classes in the blazor project, within a data folder:



Each class essentially creates a separate table. For example, the people table:

```
namespace KeyBlazor.Data
{
    7 references
    public class Person
    {
        3 references
        public int Id { get; set; }
        7 references
        public string FirstName { get; set; }
        5 references
        public string? LastName { get; set; }
        2 references
        public Company Organisation { get; set; } = new();
    }
}
```

And then the actual data table itself (in contrast to those in SQLite):



This also makes it very easy to create relational databases as you can make it (as above, where Organisation is the person's company from the list of companies) so that the people's companies have to belong to the company database. Same goes with the key database, where the keys must belong to people within the people database.

Reflection

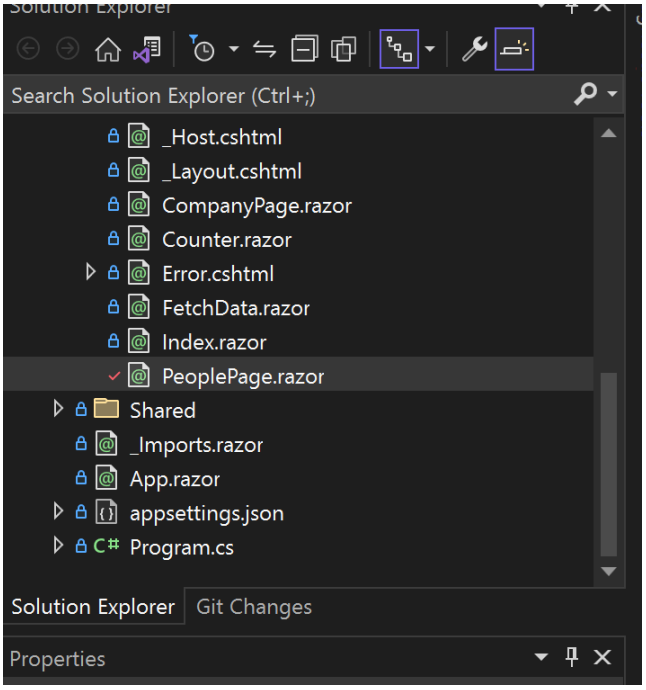
Having trialled the way each of SQL and inbuilt databases are built and thinking about how easy this would be to add into my project, I have **decided to use the Visual Studio database** software. Though it was very easy to go and set up the SQLite data tables, I struggled to figure out how to connect them to Visual Studio, and also found they lacked flexibility with changing my mind about things. Visual studio's software is all local to one place, and I had no problems trying to get it set up with the website. It also means that I will be able to get my program functional in a much shorter time frame, which is quite essential at this point given the time investments I have made in trialling the many different components above. I am also sure that I can figure out how to link everything up as my project begins to develop. Although there are limitations with the fitness for purpose of Visual Studio's inbuilt databases in that there is only one database, it is incredibly easy to create a separate instance of the project before setting it up for the specific client in need, therefore making it easy to adapt for Dad, Murray, myself or any other potential clients.

Managing version control

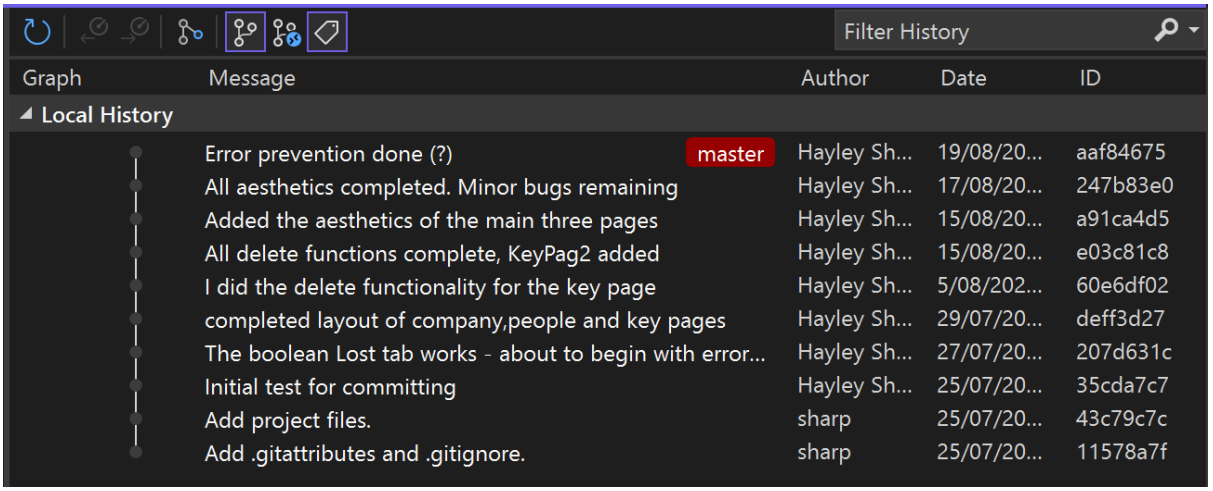
Managing version control will be very important with my project as I am fairly unfamiliar with a lot of the techniques I am using, and therefore I may not be able to quickly replicate any work I have created should I lose access to it.

I have decided to use GitChanges for version control (within the Visual Studio/behind the solution explorer), as it allows me to save things at multiple points throughout the project and go back to every instance where I have committed a change to the code. It also allows me to see (with the icons below) when I have edited each page, and

therefore remind me to save a new version regularly which will be important if I do lose code or find it no longer works.
This, alongside the trello board snips I have, will help me identify at what point I had what content inside my code, and therefore go back to older versions to find a working version of the section of my code that may not be working. If I entirely lose my project, this also helps as I can find the most recent version of it, and pick it up from a much later point than if I had to go back to the beginning.



For example, I have created a GitChanges version of my project, and then edited the PeoplePage. This edit that I did is shown as a red tick whereas all other pages are the same as previously, indicated by the blue locks. If I find multiple places within my project have red ticks, it will be an indication that it's a good time to commit the changes and therefore have a new backup.
I could have used GitHub however as I am working on this project solo, branching and re merging my project is unnecessary and adds more complexity to a project I am already figuring out so many new aspects of. Therefore, I think using GitChanges is a happy medium that does a lot of the work for me but is still safe and easy to use.



Above: Some of the version control history I have so far.

I have prioritised the following relevant implications:

Legal, Intellectual Property

In creating this outcome I have an obligation to ensure everything I use or create is my own. In any instance where I am not using code that I have created, or have researched and then taken knowledge or course of action from this research, I should credit this to the source in an accurate manner, explicitly making it obvious where this is the case. I cannot infringe upon others' work or intellectual property.

Privacy and ethics

In creating this project, which will eventually be used with real names and information, it is inevitable that at some point testing will occur with the real information, particularly when Dad goes to use it himself. There is therefore a need for the people whose data is being stored to be notified of this when it does happen and what exactly is being stored. On that note, it would be unethical to store information that was unnecessary to the scope of the project. Therefore, the only information I plan to program into the application to allow Dad to store and use about the actual people involved are their full names, the company they work for, and what key they hold (if any). I will not test with the actual information and will use only fake names or the names of myself and my stakeholders, who consent to their names being used, at all testing stages, to prevent sharing of these employees' actual names where not necessary.

Accessibility

Accessibility refers to how my program caters to the needs of all users, such as font size and colour choices for people with sight disabilities. Dad does wear glasses but not always, and so I will need to ensure that any font I use is large enough for him to read easily. Though he is not colourblind, for my project to be fit for purpose in the broadest sense, I would like to look into colour choices and other things that can be done for people with vision impairments to ensure my project is fully accessible.

Functionality and usability

At its core this is the most important business of the project, to serve its most basic purpose. I need to ensure that the project can actually be used for its intended purpose, to store information about keys at Dad's office business. It should be easy to use and any features that are included must all be fully functional.

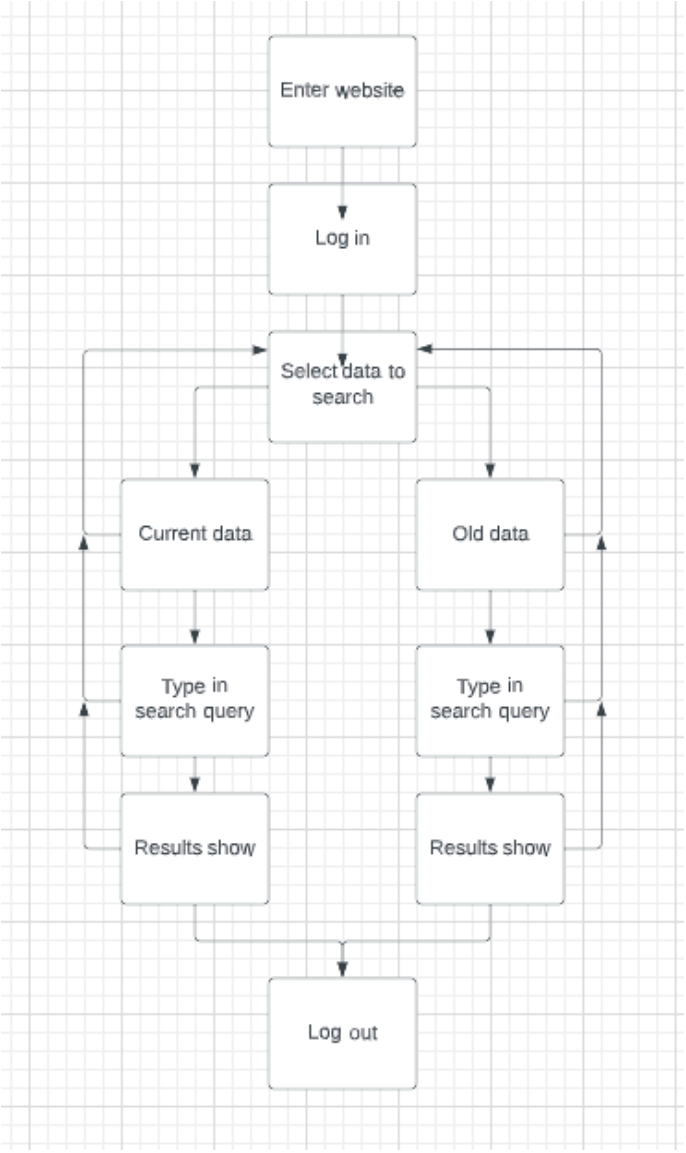
Sustainability and future proofing

To be sustainable and future proofed, the site should be debugged to the point that there are no obvious errors upon completion. Should there be any issues, these should be fixable. It should be made in a way that I am not jumping through loopholes to avoid more work in the future. The code should be commented so that if someone other than myself looks at what I have created, they are able to understand what I have done and hopefully fix it. The software I plan to use (Blazor) is very current, therefore there should be minimal issues in relation to software updates and

ageing of the software when compared to old software, which may be clunky and hard to fix in the future. Finally, Dad will have the ability to create, update and delete the records himself, meaning that unless there are bugs in the program, he has complete control over the data in the database, and I wouldn't need to be called on.

Site layout prototyping

Having now researched the alternatives I could use, their drawbacks and positives, I decided that it may be necessary to plan out the flow of how the website could be used, so that I can give an indication to Dad of how his software may appear.

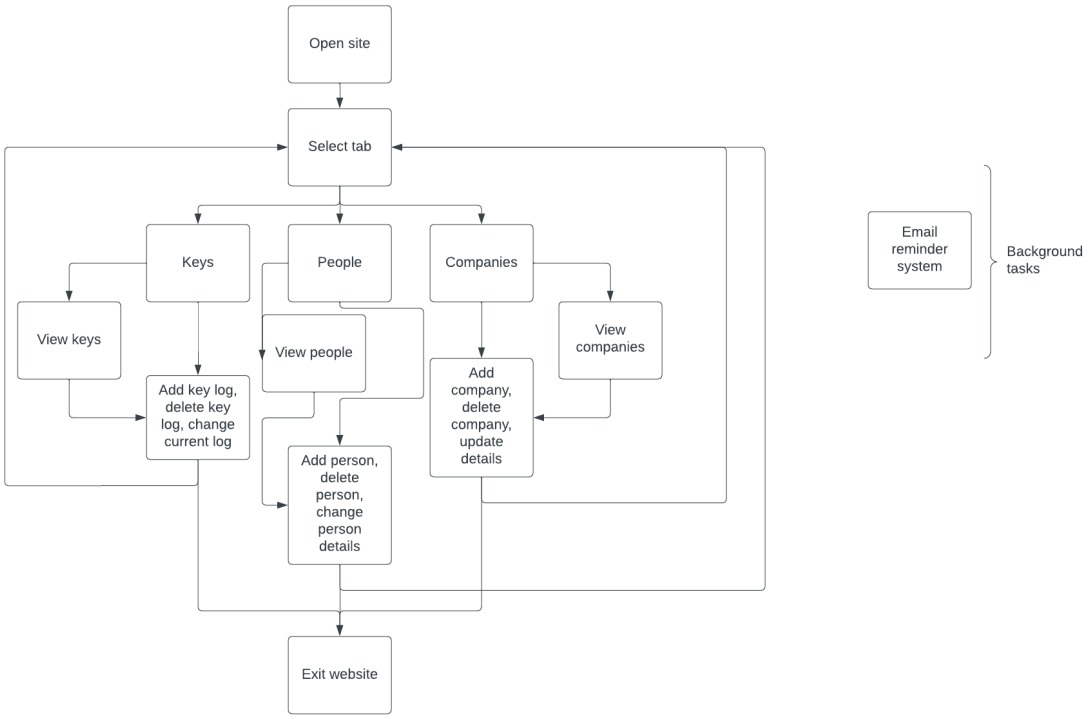


On the left shows a prototype of the workflow for Dad. He would log into the website, then select which of the sets of data they would like to investigate (data about keys currently in rotation, or data about past users). At any point through this process, he can go back and change which dataset he wants to search through. Once he is done they then log out of the system.

(01/06)

Having now spoken to Murray, there may be additional steps in this process that he would like to have as a feature of the website. One of these is an alert system. Maybe every time it logs in, it gives him an alert for the next key to be due back? Or externally to the website, he gets notifications and alerts?

From here on out the database will be personal to the user (i.e. Dad/me/murray/ other users), as now inputting people into the database would no longer make it fit for purpose in the broadest sense.



Above is a revised overview of the website. This gives a lot more user freedom, which makes more sense for users to have control over what they want to search, or view, or edit, etc. Dad also agrees that this is the case, and thinks the second flow makes a lot more sense - he mentioned that this way, he was able to decide what he wanted to search through (i.e. one individual category, rather than all results). This provides the functionality that Dad requires and means that it is suited to his requirements as an end user. The email reminder system was requested by Murray so that he could be reminded to request keys back, however this was not a request of Dad's.

Reflection

At this point now that my project is scalable and I am more aware of the undertaking, I think it makes some sense to split it up into parts to make it more manageable, and also to be able to track the development of my project to ensure that all sections are fit for purpose in the broadest sense.

Distribution of workload

Section one - Database - This is about the background functionality of my database, being storing and relating data. The database must be able to permanently store accurate information. There must be security in the database not being accessible to others. The data being stored must be intuitive, functional, and satisfy the needs of my end user.

Section two - Website - Stage two refers to the functionality of the website at a base level. This includes having functional and multiple tabs, one for each thing to be

searched through, working links, accurate links to the database, and the ability of CRUD (create, update, read and delete) for any stored information. This should also involve error prevention and testing for valid, boundary, and unexpected inputs.

Section Three - Aesthetics - In my mind, this refers to the usability of the website. Using intuitive and visually pleasing colours, logos, and having an intuitive layout will all be crucial to ensure I have suitably completed this section.

Potential Section Four - Outreach - This would involve actually getting the digital outcome to people beyond Dad. Creating personalised log in details, making the website available on a public domain, or other things along those lines. The process of turning a working prototype into a live, working, website, which multiple people can access and use.

Brief Two

After completing research and initial investigation, I reviewed my project specifications. I will be creating a permanent relational database within a website to allow Dad to store and sort information of the keys held by tenants of his building.

Specifications (from Brief One, with new additions)

Uses permanent data storage.

- Pre-populate the data for Dad before he begins using it
- Allow functionality to update, delete or add records
- Uses a relational database to link individual information
- Accurate and contain full records of the data that is necessary, where it has been input. There should be no 'holes' in this data.
- Data privacy. No one other than myself and Dad should have access to the data that he has input

Versatile.

- Needs to be functional on multiple devices, and on the go - done using Blazor

High usability.

- Streamlined and efficient process for quick access and updating
- Organised to suit his needs
- Well tested and efficient
- Users have the ability to create, update, view and delete any data within the table/s.
- Mistakes made during the CRUD process are fed back to the user with helpful error messages.

Predictability

- Time, effort and learning investments in my project are understandable and manageable. Components will not take longer amounts of time to complete than expected. It is known what components will be prioritised in the

development process and what components will be additional should time permit this.

Satisfaction

- Users should be satisfied using the website. Site has some complexity in having data groups separated by different tabs.
- Colours, icons, emblems all make sense to the context. This is tested using other people to ensure a wide range of individuals can agree on the useability.
- Layout makes sense and is intuitive. There is no unnecessary visual pollution. Layout could be altered keeping in mind Nielsen's Heuristics.

Updated stakeholders:

There are various stakeholders I could call on throughout the duration of my project. These include:

Dad - Dad is the main stakeholder in my project, as he is also the end user. I should be reflecting on many choices throughout the project with Dad, not so much in terms of technicality, but how he wants the website itself to function, and potentially also some of the aesthetic decisions I make.

Uncle Thomas - Uncle Thomas works in website development and so could be someone I call on for stakeholder feedback. He takes interest in my project due to general interest in website development and also his expert knowledge of Blazor and webpage design.

Uncle Markus - Uncle Markus also has experience with website design having created his own websites over the years. He does have limited experience with Blazor, but has some expert knowledge surrounding interface design.

Mrs Langman - Mrs Langman is my DTP teacher. She takes interest in the ins and outs of my project, particularly working towards submission. She could be a good stakeholder to talk to when reflecting on time boundaries or decision making surrounding time management and workload.

Murray - Murray is the caretaker at Hillcrest High and also has many keys to store and track. He could be another potential stakeholder as my project could serve him a similar purpose to Dad's. His thoughts towards what I am making could help me get a broader understanding of whether my project is fit for purpose in the broadest sense.

Other teachers - This could include Mrs Stonestreet, my DVC teacher, any of the technicians such as the science technician, or the librarian.

Various other students in 3DTP - I sit among other students taking DTP, who have some limited knowledge in Blazor, but also can provide opinions particularly towards aesthetic decisions. These people may come in handy at stage three when I am having to make decisions surrounding aesthetics where they can help provide a majority vote on what makes most sense. They have interest in my project as fellow classmates who I sit near, therefore they may often hear about what I am working on or have some curiosity to find out more about the new (to them) programming languages of HTML or CSS.

Beginning my project

Stage One - Database.

Traits of data, value types

Having trialled different potential options to use as a database, I settled on using inbuilt Visual Studio database storing capability, and began to build the database. For this, I set up three tables, one for keys, one for people, and one for companies. Each table had various features:

```
public class Key
{
    1 reference
    public int Id { get; set; }

    [Required(ErrorMessage = "Please enter a colour.")]
    5 references
    public string? Colour { get; set; }
    6 references
    public bool Lost  { get; set; }
    5 references
    public string? Notes   { get; set; }
    3 references
    public Person? User { get; set; }

}
```

```
public class Person
{
    3 references
    public int Id { get; set; }


    [Required(ErrorMessage = "Please enter a first name.")]
    10 references
    public string FirstName { get; set; }


    7 references
    public string? LastName { get; set; }

    [Required(ErrorMessage = "Please select a company.")]
    5 references
    public Company? Organisation { get; set; } = new();

    2 references
    public List<Key> Keys { get; set; } = new List<Key>();
}
```

```
11 references
public class Company
{
    2 references
    public int Id { get; set; }


    [Required(ErrorMessage = "Please enter a company name.")]
    11 references
    public string CompanyName { get; set; }


    2 references
    public List<Person> People { get; set; } = new List<Person>();
}
```

As can be seen above, the various databases were linked with one another. One characteristic of each key entry was a person who was installed in the people database. Each person also had a characteristic of a company. These were non-nullable (identified by the question mark). Though this does mean that the people needed to belong to a company, which is rigid, all of the people within the building are employed by a company, meaning that there should never be a problem. Even if the person was a freelancer, for example, the site could be set up with a company entry of “No company”, meaning that anyone who does not belong could still be assigned to a company of sorts.

Dad wanted this functionality as he had complained that often, the keys would be given to a person from a particular company, who would then pass the key on to others in the same company. Therefore, even if the key was no longer in that person’s hands, it was still somewhere within that company.

Other specifications of the databases included the required fields, such as the colour of the keys, the first name of a person, or the name of the company. These were fundamental to the basic functionality of the site. Without these things that really define the basic characteristics of the key, company, or person, there was no way to define who they are, which could cause some confusion or error later on.

Privacy

The good thing about using Visual Studio's data storage functionality is that since I am the only one with access to my code, I am the only one who has access to the data that Dad has input for testing. When the project is completed, I can create a copy of the entire project for him, so that he will be the only one with access to that particular copy. This could be the case for any potential stakeholders, where a separate instance can be made for the individual, maintaining privacy of the data.

Reflection

Section One has been completed. There are three tables within the database, for people, keys, and companies. I have discussed this system with Dad, my primary stakeholder, who appreciates the separated tables, as they will be ideal for searching through a specific category when the site is built. There is very little stakeholder feedback required as this is part of the functionality of the website, however it was helpful to know that Dad thought the way I had set the tables up for him was ideal.

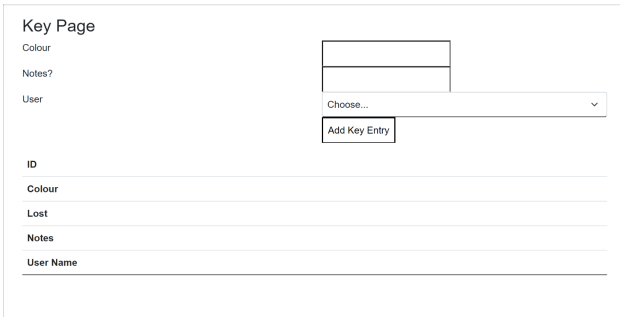
Stage Two - Site building

Site overall layout

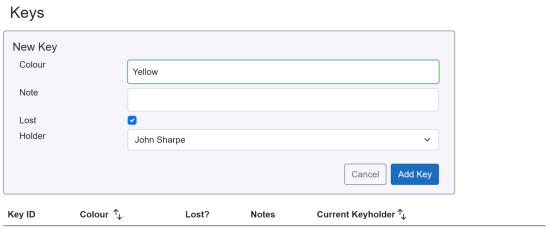
I had earlier asked Dad, with the support of the flow diagrams earlier in this report, how he wanted to be able to search or look through the data - whether he wanted to be able to search through all three categories at once, or to find only one category and search through it. He had mentioned that his preference would be to be able to look through each individually, so I chose to have separate tabs for each of the three categories. This way, his options are separated and always visible, and therefore he can easily switch over whenever necessary.

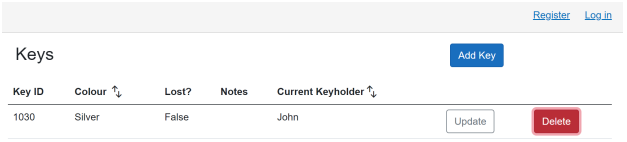
Initial layout setup

Building the site required basic knowledge of CSS and HTML. Thankfully with Blazor, most things are highly scaffolded and all that needs to be done is for the content of the page to be filled out. I decided to speak with Uncle Thomas about how best to set up the websites, and he recommended it was best to ensure that the data I planned to use within the page worked and was showing up where it needed to be, before worrying too much about how it's laid out. I started with the left layout which simply served the purpose of getting all of the information on the page, before actually properly moving on to a cohesive layout (as below). More detail on the layout decisions is provided in Section Three.



Page before consideration of formatting.





Pages after formatting.

Difficulties with data visibility

I had some issues with getting the correct data to show up within my tables. The first issue I encountered was that in my key page, I allowed for the user to check whether the key was lost, however regardless of whether they said yes or no, the table would always record a false entry.

green	False	none	hayley
purple	False		alisa
yellow	False		hayley

E.g. I said true for yellow, but it has come up as false.

I added this line in. The bind-Value command sets the value of the thing in quotations (newKeyEntry.Lost in this case) to the desired thing, in this case the current key entry's lost or not lost tag (it then does this in the ChangeLost function).

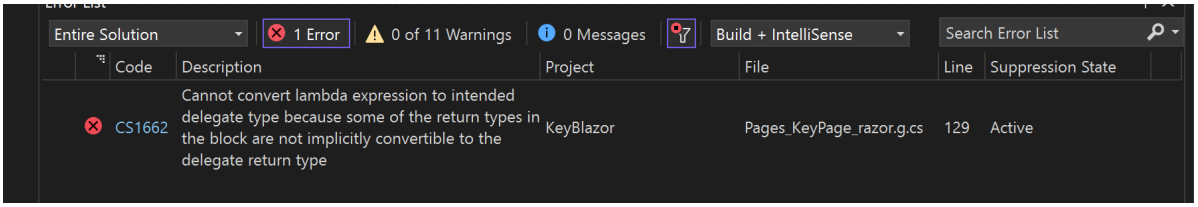
```
<div class= "col">
    <select @onchange = "ChangeLost" class="form-select">

        <option>Choose...</option>

        <option [ngValue]="False">False</option>
        <option [ngValue]="True">True</option>
        @bind-Value="newKeyEntry.Lost"

    </select>
</div>
```

This gave the following error, i commented all the other lines out to confirm this was the issue, which it was, as it did build when i commented that line out.



I noticed the bind-Value was not activating in the same way the others were (it should turn purple when used correctly). When dictating the values of other characteristics of the key, such as the colour or person assigned, I used InputText, so I thought using InputSelect rather than just select could help, which still did run. However, after selecting the option for true, it still spits out false.

Notes			
User Name			
green	False	none	hayley
purple	False		alisa
yellow	False		hayley
hello	False	hi	hayley

I realised that the bind-Value command is similar to the onchange command, which I know executes after there is a selection input. The stuff within the InputSelect executed in creating the options, and sits there otherwise, not actually executing any decisions. Therefore, I moved the bind-Value statement into this top line of InputSelect:

```
</div>
<div class= "col">
    <InputSelect @bind-Value="newKeyEntry.Lost" @onchange = "@ChangeLost" class="form-select">


        <option>Choose...</option>

        <option [ngValue]="False">False</option>
        <option [ngValue]="True">True</option>

    </InputSelect>
```

green	False	none	hayley
purple	False		alisa
yellow	False		hayley
hello	False	hi	hayley
hellp	True	ehfk	hayley

And it works.

I decided to add a company category to the keys page, so that the client can search and sort by company when looking at the keys, in case there are keys that are lost within a company.
This is the line of code I had, which would put a row in the table for the user's organisation.

```
<td>@c.User.FirstName</td>
<td>@c.User.Organisation</td>
<div class="col"><button @onclick="@(() => s
```

ID	First Name	Last Name	Organisation	
1006	Joe	Smith	Company 1	Delete
1007	susan	jones	HELLO	Delete

Above are the companies of the two people in the table below. The code I had used isn't working, as it should come up with those two organisations in the last column, and is rather inputting KeyBlazor.Data.Company (KeyBlazor is the name of my solution).

Key ID	Colour	Lost?	Notes	Current Keyholder	Company		
1006	Blue	False		Joe	KeyBlazor.Data.Company	Update	Delete
1007	purple	False	this should say hello	susan	KeyBlazor.Data.Company	Update	Delete

I had concerns that there were issues with the data, so decided to take a look at the company data table:

	Id	CompanyName
▶	1007	Company 1
	1008	HELLO
*	*NULL*	*NULL*

This is the data table of the two companies I had at this point of testing.

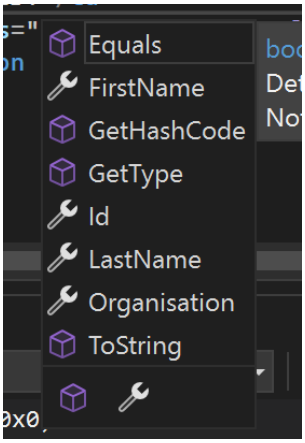
	Id	FirstName	LastName	OrganisationId
▶	1006	Joe	Smith	*1007*
	1007	susan	jones	1008
*	*NULL*	*NULL*	*NULL*	*NULL*

and this (above) is the table for the people. I thought I had them stored under their organisation, but turns out it is actually stored under their organisation ID, so I will need to figure out how to access the organisation from the company table, going through the organisationID in the people table.
I substituted the User.Organisation for the following:

```
<td>@c.Lost</td>
<td>@c.Notes</td>
<td>@c.User.FirstName</td>
<td>@c.User.Organisation.CompanyName</td>
<div class="col"><button @onclick="@(() => setEdit(c))">Update</button></
<td><button @onclick="(() => deleteKey(c))">Delete</button></td>
```

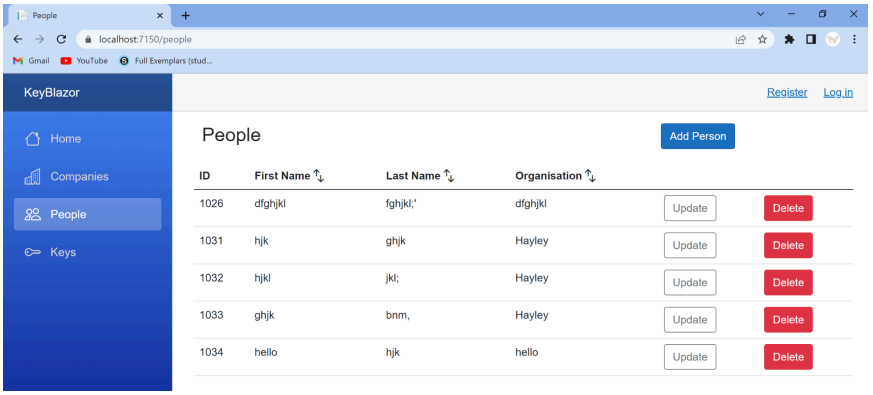
And now it is giving me blank spaces in the table.

Key ID	Colour	Lost?	Notes	Current Keyholder	Company		
1006	Blue	False		Joe		Update	Delete
1007	purple	False	this should say hello	susan		Update	Delete

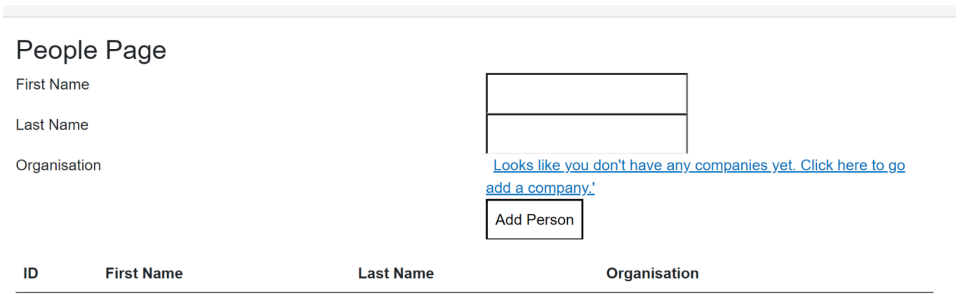


Interestingly it isn't autofilling for me the option for organisation ID when I go to write the above line of code into the program, which I thought it would've - it only autocorrects to Organisation. This leads me to think that I probably need to go through an alternative method. Since OrganisationID is a characteristic of the person, and not the company, I probably need to go via the characteristic of the company (being its name), since the database goes from companies to people to keys, rather than the other direction.

Changing the problematic line to c.CompanyName.Organisation worked, and I am now getting a proper display of the companies.

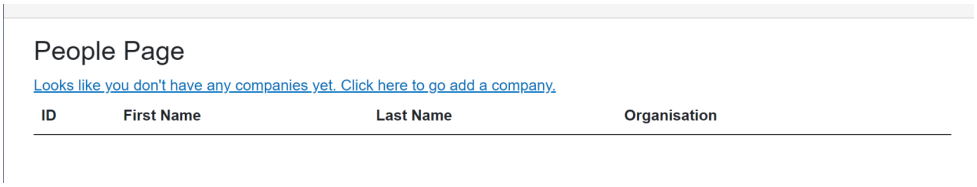


Error prevention



This is how my "people page" looks when I have no companies in my database.

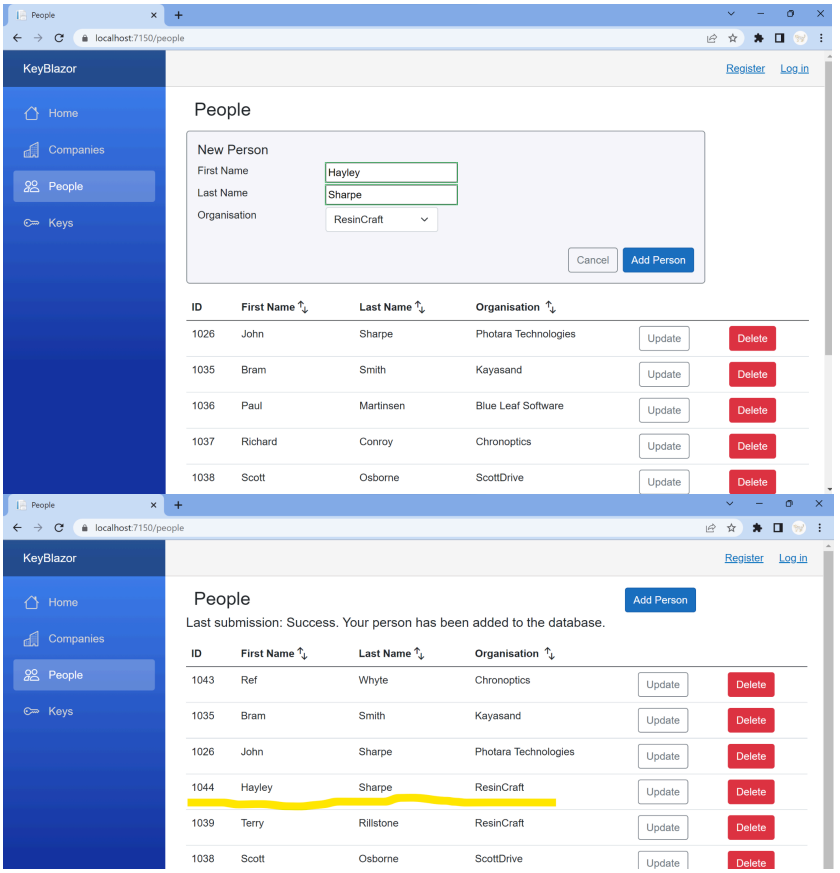
What I need to do is to ensure that the user cannot attempt to add someone into the database without assigning them to the company (not possible as the box would have disappeared). Currently, the link will take the user to the “company page”, which is correct in what I want, but clicking the add person button will cause the program to crash as there are no values in the non nullable fields. Therefore, I want to make it so that the user can only have the option to enter things into the fields when they have companies, and the same for keys (they’ll need people).
Where I had the code that would execute if there were no companies in the companies database, causing the link to pop up, I ensured that the add person functionality was not created, making it impossible for anyone to fill any information without at least one company existing:



This left only the table and link visible, giving the user no confusion as to what their next step should be, and no way to crash the program.
Note: this was later changed during aesthetics to only appear on button click.

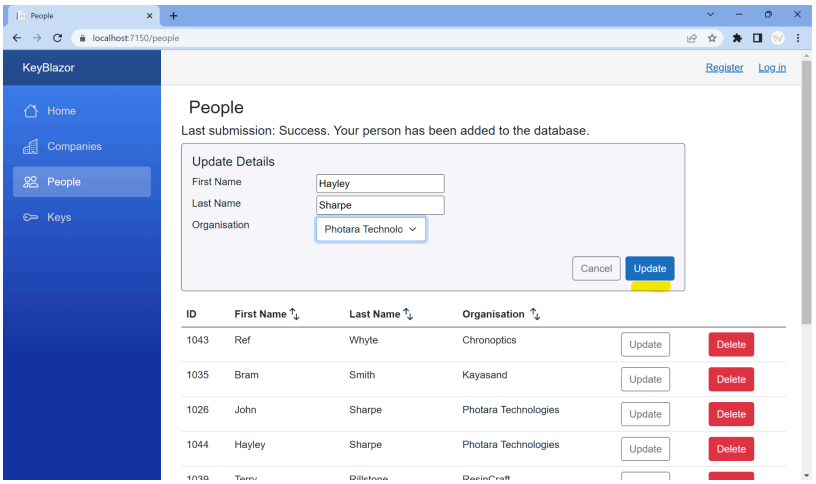
Create functionality

The user of the website should be able to create new entries. Though I am using the “people page” to demonstrate this, there is similar functionality across all three pages. The user can create a new person to be added to the database, having the traits of a first name, a last name, and belonging to a company. Upon successful addition there is a message of success saying that the person has been added to the database, and the table is updated with this person’s information, as on the right.



Update functionality

The user has the ability to update their information. As on the right, when editing my entry of myself in the database, I can change the organisation I belong to (or first or last name, should I wish to do so). This updates upon change as below in the table, the organisation I belong to has become Photara Technologies.



Read functionality

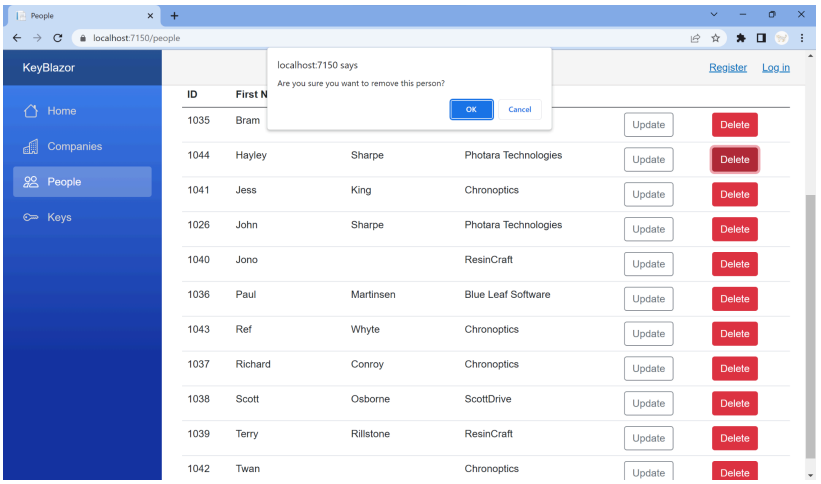
The user should be able to freely view the data at any point. Though it could be useful to have a physical search bar, I feel this may not be necessary, as Dad's office building only accommodates up to 20 individuals, all of whom he knows by full name. Additionally, it would mean another step in the process before Dad can see his data, as with a search bar, the data is generally hidden before it is looked through.This would prevent it from having high usability as it would decrease the efficiency of the website. Therefore, I am able to display all of the records there on the page, and also have the ability to alphabetise them, meaning that he can quickly get the names of the bottom users to the top of his screen. He could also alphabetise by organisation, to group all of those within the same organisation together, or by first or last name (done by first name on the right). This gives him some flexibility with the category he searches through, further streamlining the process for him based on what exactly he wants to find.

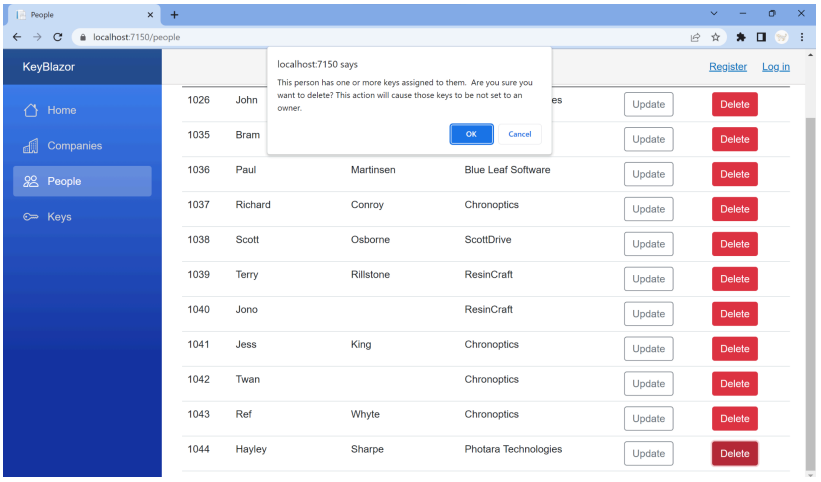
ID	First Name	Last Name	Organisation		
1042	Twan		Chronoptics	Update	Delete
1039	Terry	Rillstone	ResinCraft	Update	Delete
1038	Scott	Osborne	ScottDrive	Update	Delete
1037	Richard	Conroy	Chronoptics	Update	Delete
1043	Ref	Whyte	Chronoptics	Update	Delete
1036	Paul	Martinsen	Blue Leaf Software	Update	Delete
1040	Jono		ResinCraft	Update	Delete
1026	John	Sharpe	Photara Technologies	Update	Delete
1041	Jess	King	Chronoptics	Update	Delete
1044	Hayley	Sharpe	Photara Technologies	Update	Delete
1035	Bram	Smith	Kayasand	Update	Delete

ID	First Name	Last Name	Organisation		
1035	Bram	Smith	Kayasand	Update	Delete
1044	Hayley	Sharpe	Photara Technologies	Update	Delete
1041	Jess	King	Chronoptics	Update	Delete
1026	John	Sharpe	Photara Technologies	Update	Delete
1040	Jono		ResinCraft	Update	Delete
1036	Paul	Martinsen	Blue Leaf Software	Update	Delete
1043	Ref	Whyte	Chronoptics	Update	Delete
1037	Richard	Conroy	Chronoptics	Update	Delete
1038	Scott	Osborne	ScottDrive	Update	Delete
1039	Terry	Rillstone	ResinCraft	Update	Delete
1042	Twan		Chronoptics	Update	Delete

Delete

When going to delete I had a choice between searching up an entry and then deleting, therefore implementing a search function for this specific purpose, or scrolling through the list where each entry has a delete option available. The first option does add another step in the process of being able to delete an entry - which may not be best from an efficiency point of view - but it produces less visual clutter and decreases the chance of an accidental delete. The second option does increase visual clutter, however is one less step in the process, and means he can delete people as he comes across them, if he has not used the site in a while and finds a person he wants to delete.



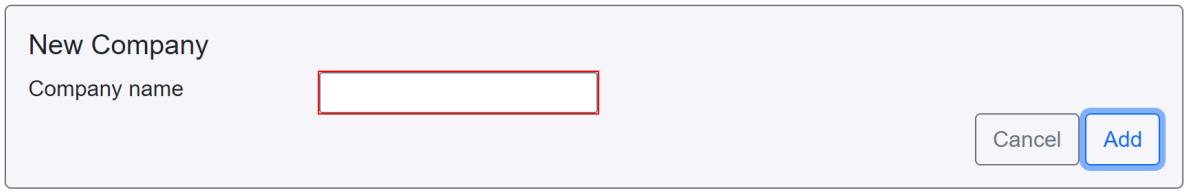


From a usability point of view, the second one makes more sense. However, there is cause for concern over accidental deletes, therefore I decided it would be best to add a confirmation of deletion so that in case he does make a mistake, he doesn't need to go through and re enter it. Even though this is another step in the process, to click two buttons is still faster than having to click and activate a search bar, type, and then click again to delete an entry. Upon talking to Dad, he agreed and said that he doesn't mind so much if the page is a bit more cluttered, so that he can quickly go through and delete what it is that needs to be deleted. More on this, particularly in terms of usability heuristics, is discussed in Section Three.

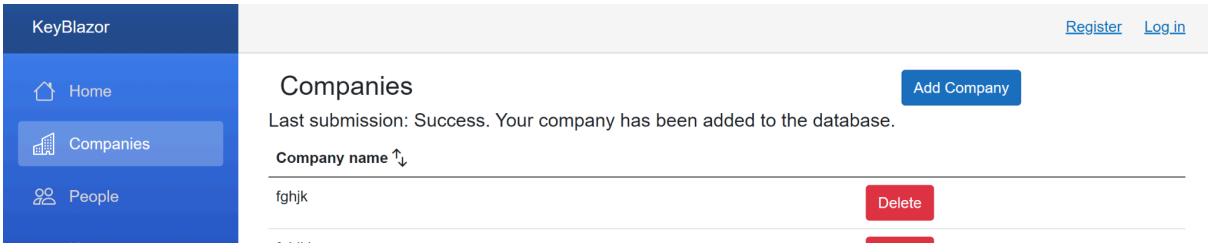
Error prevention and testing

Ensuring invalid inputs cannot be submitted, and appropriate error messages appear.

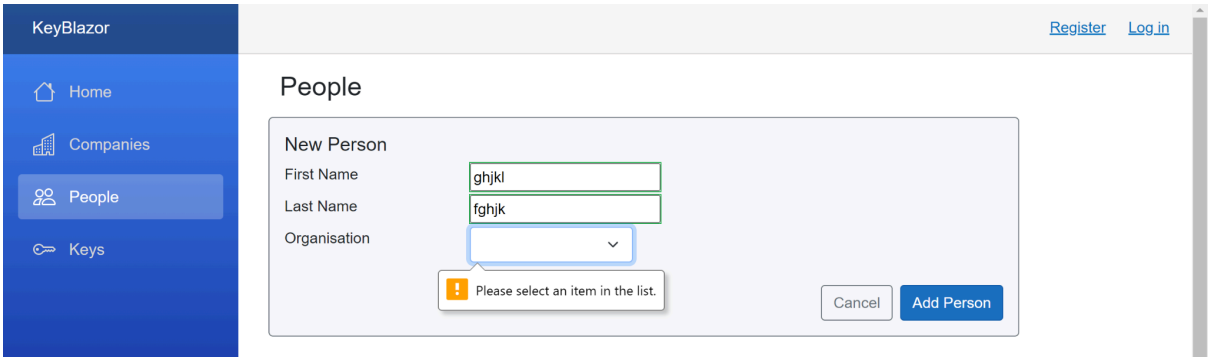
- Please enter a company name.



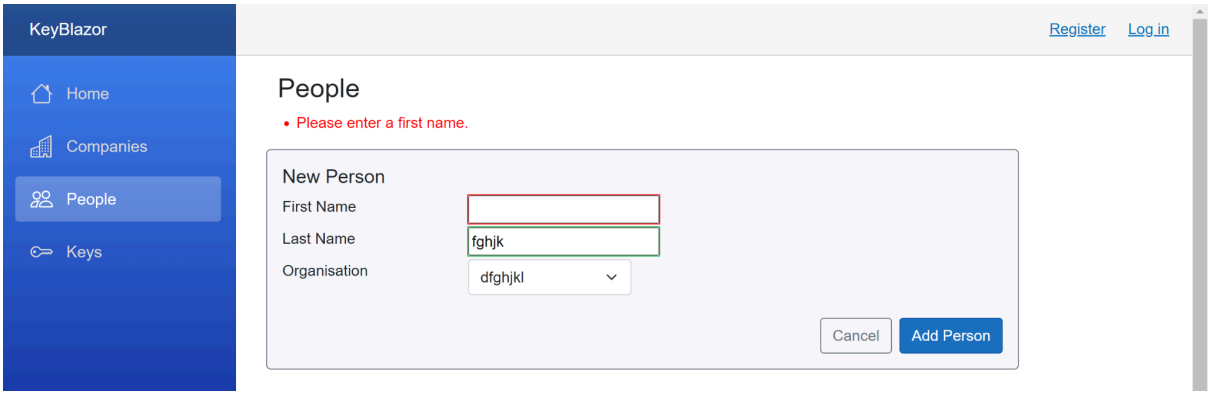
Above - prevents no name being entered for the company.



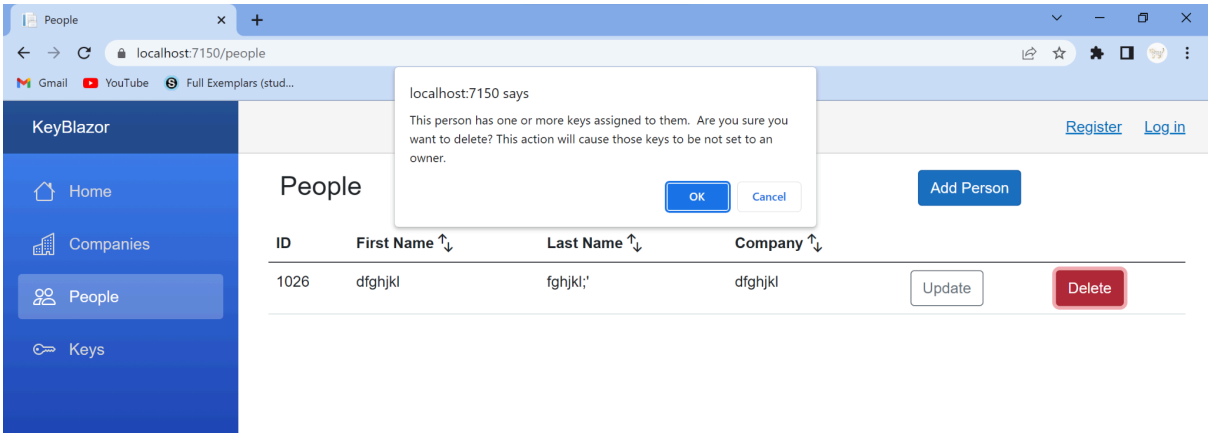
Shows status of submission.



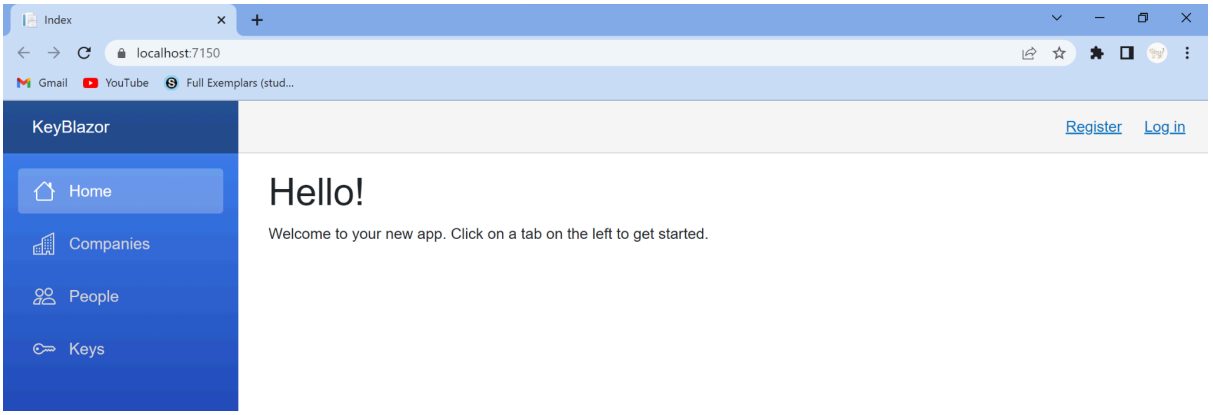
People must be assigned to a company.



People must have a first name when entered into the database.



There are warnings when going to delete items.



Home page - this is the page that the website opens to.

Reflection and stakeholder feedback

Section two was the largest undertaking in the project and I am pleased with how it worked out. This has turned the project from a database with no UI into an interactable website that can actually be used! I was also pleased to complete large amounts of error prevention as this is pivotal in ensuring my code is clean and maintainable.

I gave the application to Dad to have a go with and get his thoughts, as this point in the project was quite pivotal - most of the function was complete, yet very little of the aesthetics. This is what he said:

“It’s good to have software developed for the task I need. I couldn’t find anything that did it in a simple yet effective way, and this is what I envisioned.”

“I like the simplicity of it - I don’t actually think it needs much in the way of cleaning up for the purposes I have in mind. I like the minimal distractions. I can get onto the software, do what I need to do, and get on with my day.”

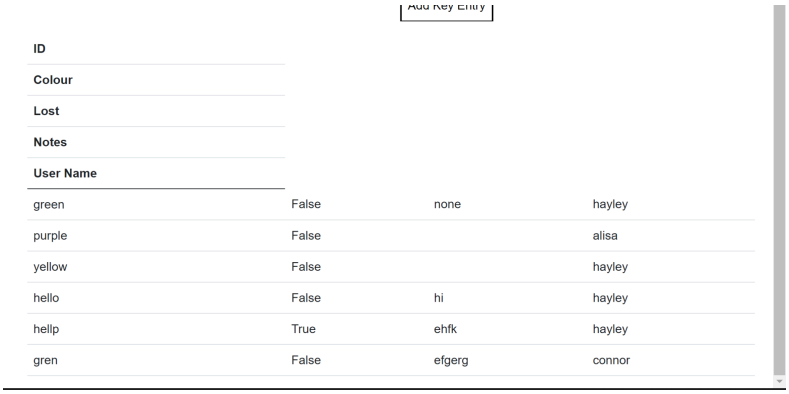
“I appreciate the checks and balances so that I don’t inadvertently destroy/delete the data I have entered. Cool feature.”

Given the overwhelmingly positive feedback I had been given by Dad, I am sure that I am on track to creating an effective website that caters towards my brief.

Section Three - Aesthetic choices

Layout

When beginning on aesthetics, the layout of my pages was not considered as my focus was on the function of the website rather than usability and aesthetics. That being said, I know that good layout can increase the functionality of the website as it provides for a more intuitive experience for the user. Clearly at this point there were issues with alignment:



ID			
Colour			
Lost			
Notes			
User Name			
green	False	none	hayley
purple	False		alisa
yellow	False		hayley
hello	False	hi	hayley
hellp	True	ehfk	hayley
gren	False	efgerg	connor

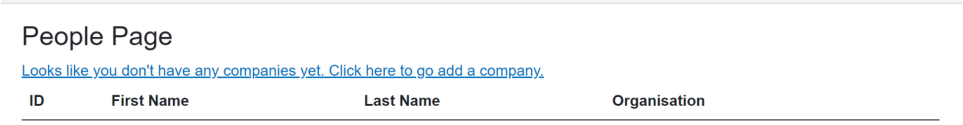
I was putting each of the headings in a different row, rather than a different column. They should line up, and also have the ID values being displayed, as I currently have 5 headers and 4 rows.

```
<table class="table">
    <thead>
        <tr>
            <th>ID</th>
        </tr>
        <tr>
            <th>First Name</th>
        </tr>
        <tr>
            <th>Last Name</th>
        </tr>
        <tr>
            <th>Organisation</th>
        </tr>
    </thead>
```

I was adding a new row per item, but am now adding them all onto the same row as table headers. This was done by putting all of the rows I had created into the <thead> brackets.

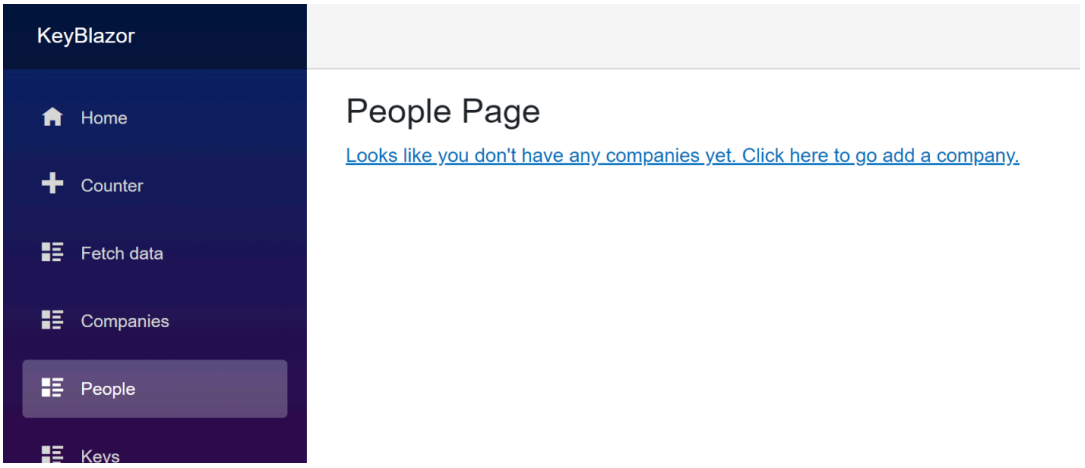
Key ID	Colour	Lost?	Notes	Current Keyholder
1031	yellow	False	sdfg	Hayley
1032	purple	True		John

Empty pages



This was immediately after I removed the fields for data entry from the form (see section one), making only the empty table and the link visible. However, I think it looks cluttered on the page. The table layout is unnecessary to a user who has no way of inputting information without being taken to the company page, and there is no point in overloading the user with information that they can't use. Therefore, I'll take the table away so that there is only the necessary information, so that the user knows exactly where they should be going.

From a usability, aesthetics and functionality point of view I think that this is a lot better (below) than what it was, where there is only one option for the user on this page, and that is to add companies (this is necessary for the people page as the people need to be assigned to a company). I followed through on the key page with the same procedure - except for people instead, as the people are necessary for key logs.



Icons

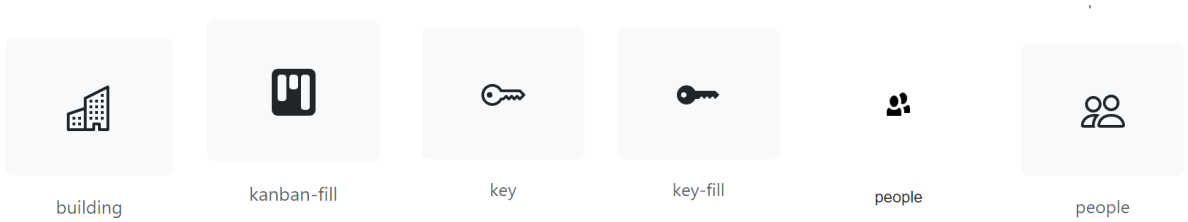
In seeing the above page and looking at the overall visuals, I have noticed that the icons on the left are also counterintuitive, and don't make sense in context with the operations of that page, so I am going to clean up the side bar.

Each of the 3 things I can choose an icon for. Thankfully I could find icons for keys and people, however there is no obvious company icon, so I went to some stakeholders for some advice. I started by googling a company icon, which came up with people in front of buildings. I also talked to Alisa and Eliza who both gave a couple of ideas for what a company icon could be - these included buildings, people, line graphs, handshake, and a briefcase. Looking at their ideas I went to Mrs Langman and asked which made the most sense of the ones I was able to find through the Bootstrap libraries for a company, and her idea was the set of buildings.

Opinions on the company icon from other students:

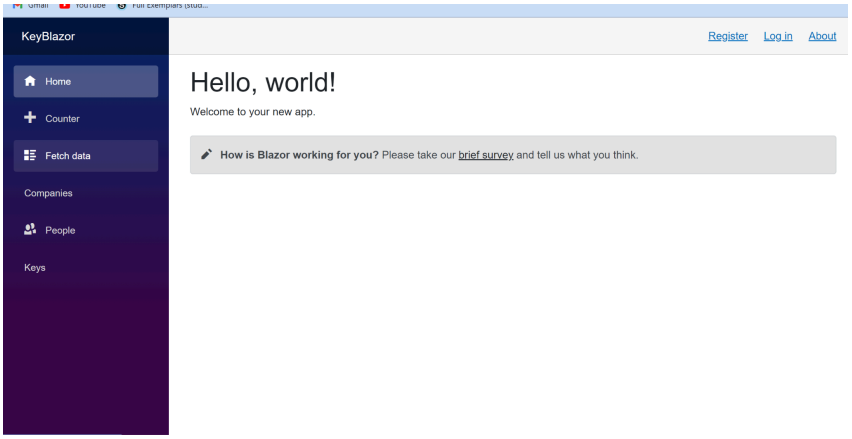
Alisa - people, Line graph, briefcase - consistency with full or non full icons

Eliza - a face, handshake, blank avatar



Above: Some of the icons I considered for the sidebar. I liked the hollowed out shapes more than the filled in ones, and wanted consistency (the kanban didn't

seem very intuitive for companies and there were no filled in buildings), however the filled in icons are much more visible.



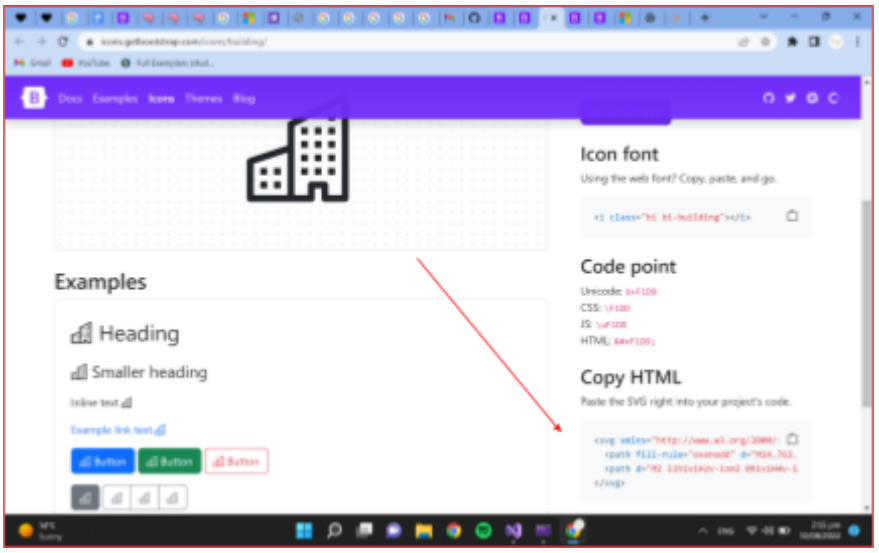
In trying to put the icons in I encountered some problems. Though I have used the same syntax as my original icons (left), my bootstrap icons for Companies and keys are not appearing.

The code for companies (bi) is the same as for people (oi)

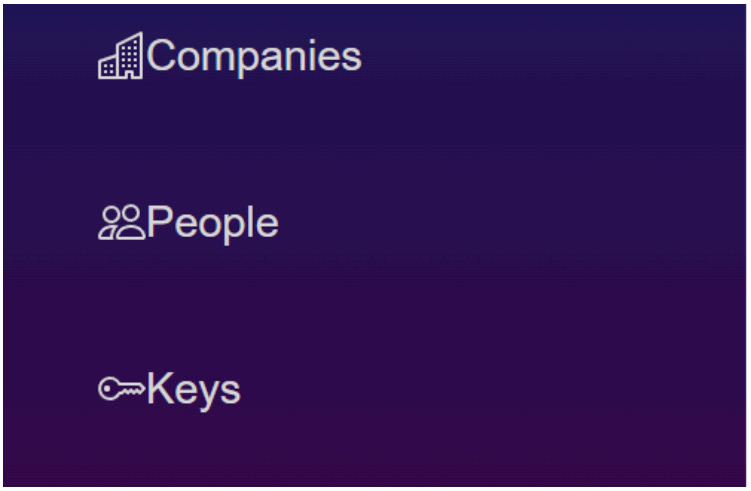
```
</div><div class="nav-item px-3">
    <NavLink class="nav-link" href="companies">
        <span class="bi bi-building" aria-hidden="true"></span> Companies
    </NavLink>
</div>
<div class="nav-item px-3">
    <NavLink class="nav-link" href="people">
        <span class="oi oi-people" aria-hidden="true"></span> People
    </NavLink>
</div>
<div class="nav-item px-3">
    <NavLink class="nav-link" href="keys">
        <i class="bi-key" aria-hidden="true"></i> Keys
    </NavLink>
</div>
av>
```

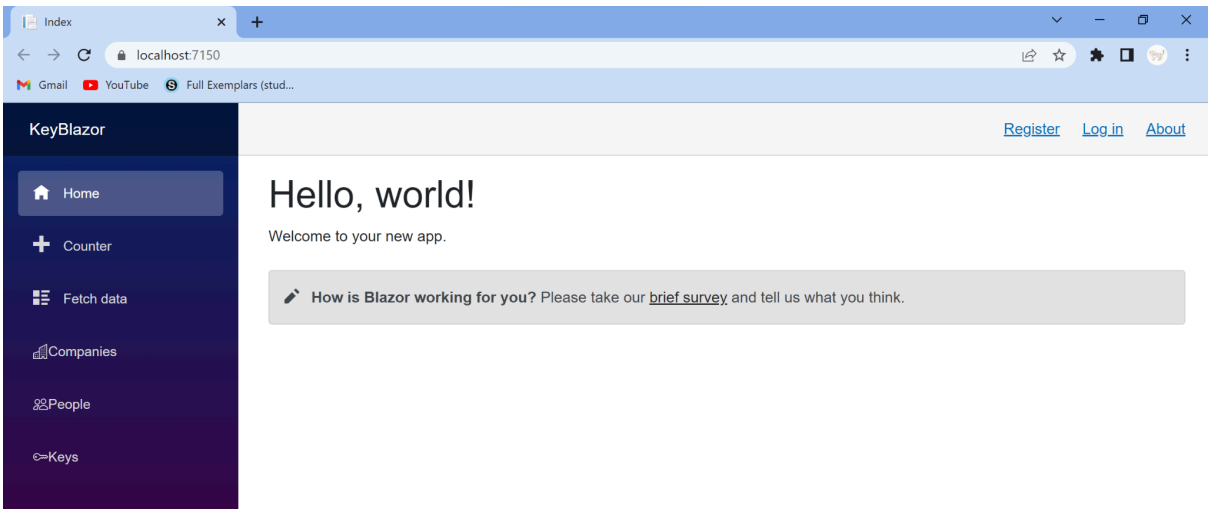
The [bootstrap source code] suggested the following, which I tried to implement with the bottom icon (key). Once again I had a blank spot where the icon should show up. I decided to look at other ways I could input the icons as listed on the bootstrap page. I found the following:

This way I could embed the icon image within the page. I decided to try this with the specific icons I wanted to use.

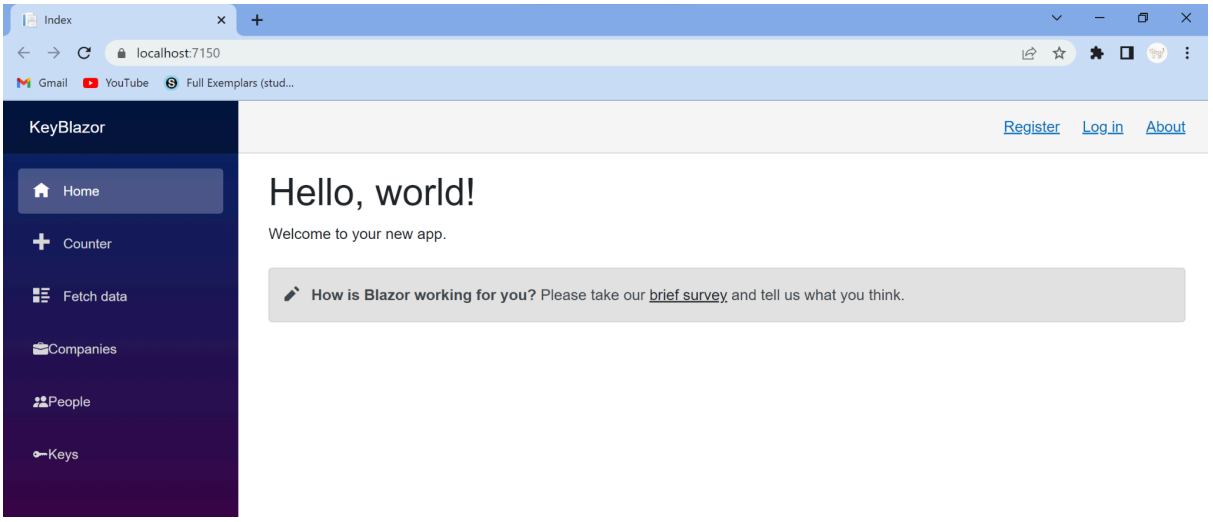


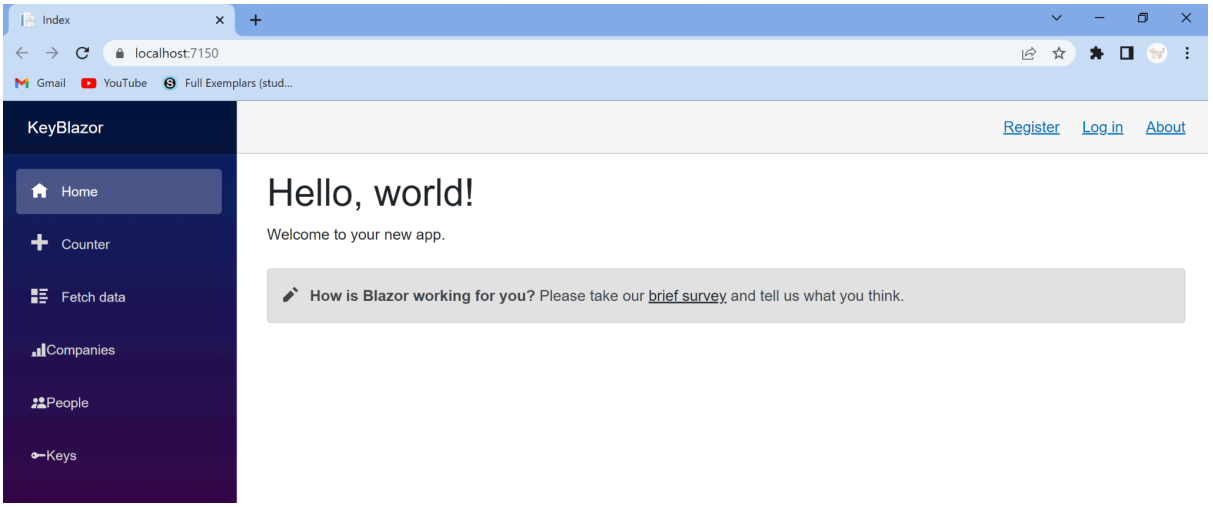
Pasting this HTML into the Nav Menu page gave me the following icons. I think these have increased the usability and aesthetics as not only are there now visuals, these can give a further indication to the user what the tabs are for, rather than just the words. The icons selected are also intuitive and make sense in context which is necessary for the usability of my website. This could speed up the process for Dad seeing icons rather than having to read the words, making his experience that much faster. However, looking at the icons on the whole web page, they are quite small and hard to see:



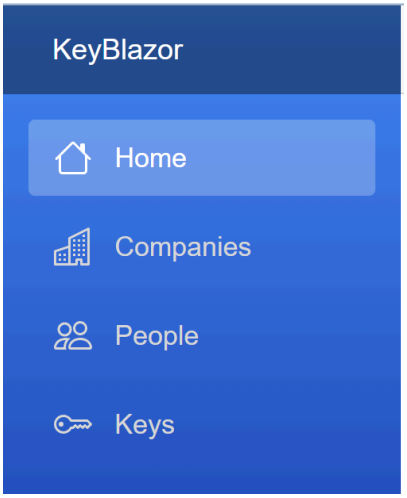


I want to trial some filled in icons to see whether this makes a difference, as the empty icons are very small and hard to distinguish. The only problem is that the buildings do not come as a filled version, so I have to use an alternative, either a briefcase or bar graph as Alisa suggested:



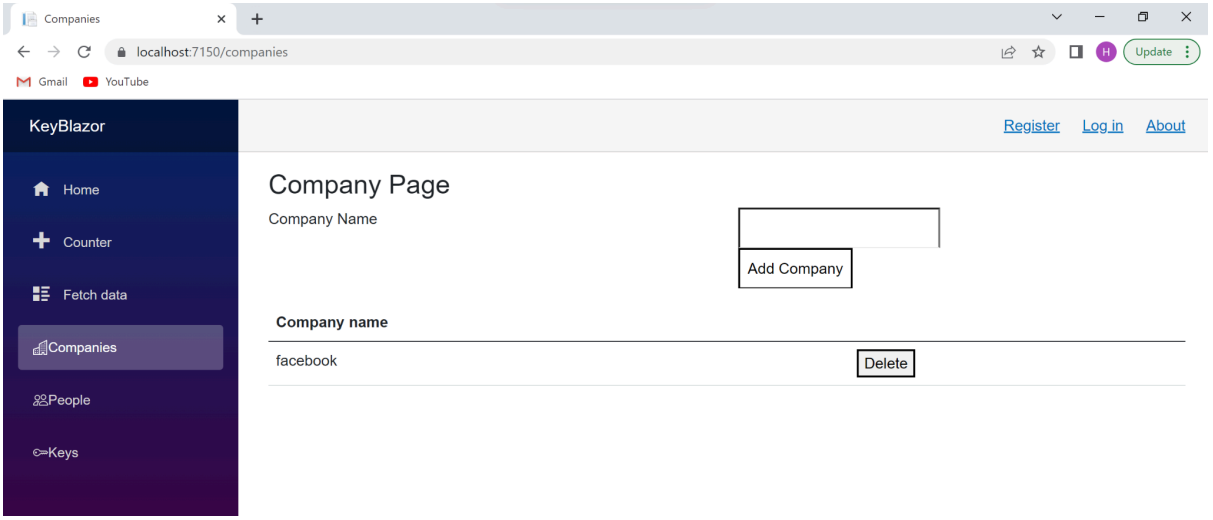


I think that the filled icons are more clearly visible, which is better for usability, but the lack of the building means that the company icon is not so clear. Since I was torn between visibility and aesthetics I asked Dad for feedback, and he decided that his preference was actually the empty icons.

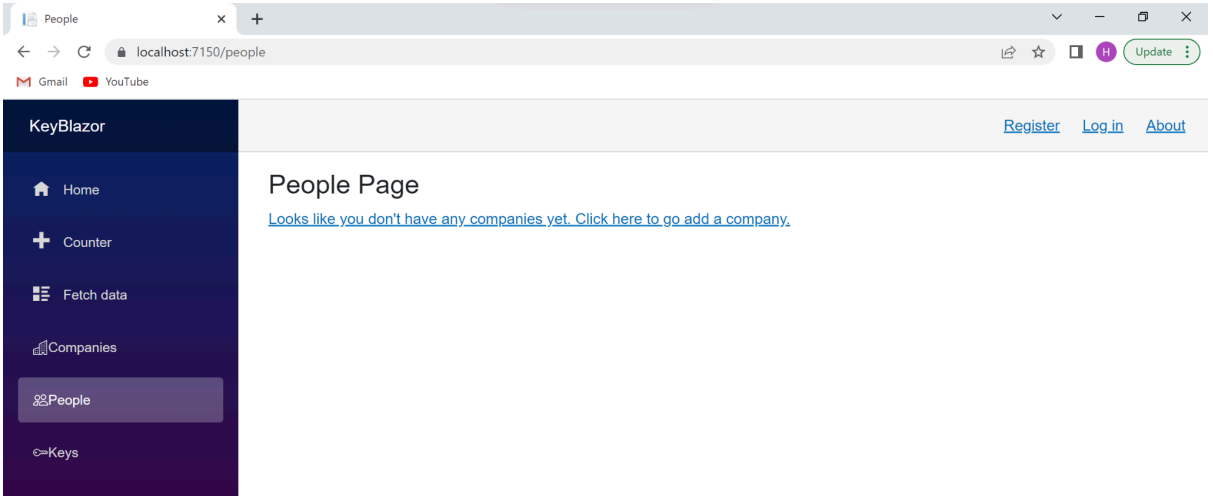


To maintain this on the site, I will try making all of the icons larger so that it is easier to read, and also adding a space between icons and text. On the left is how this ended up appearing.

Now for the formatting of my page. Currently, this is how one window looks:



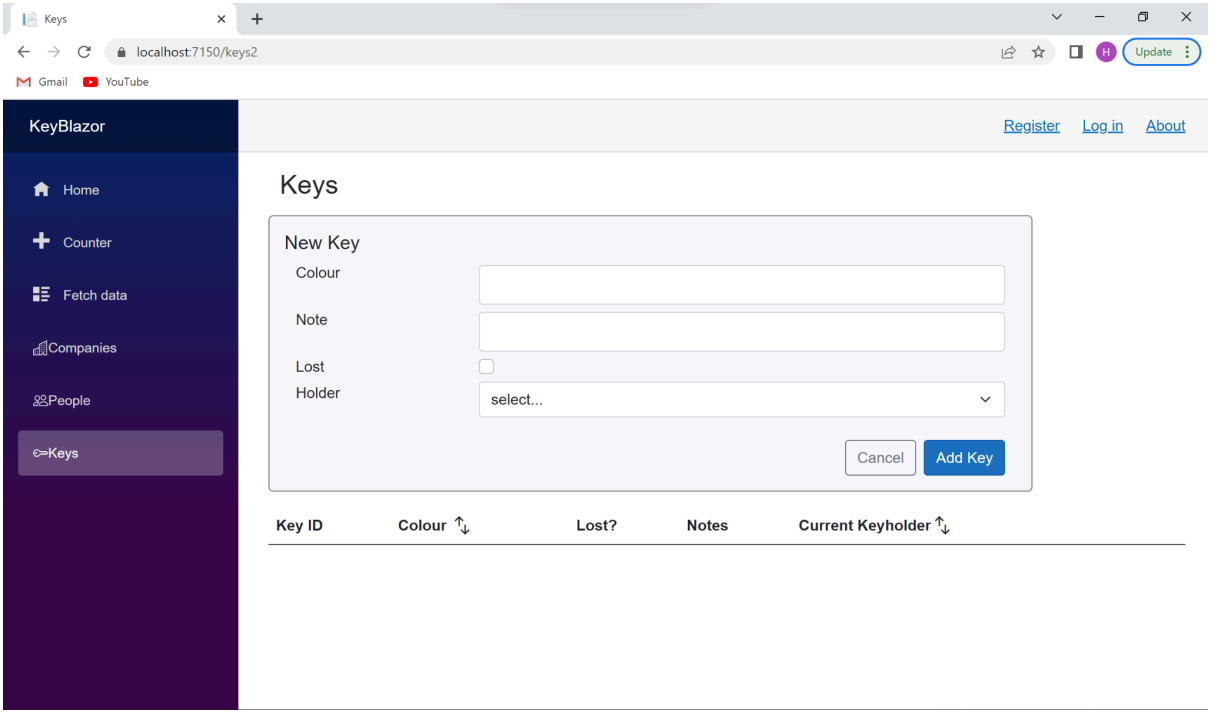
(with data)



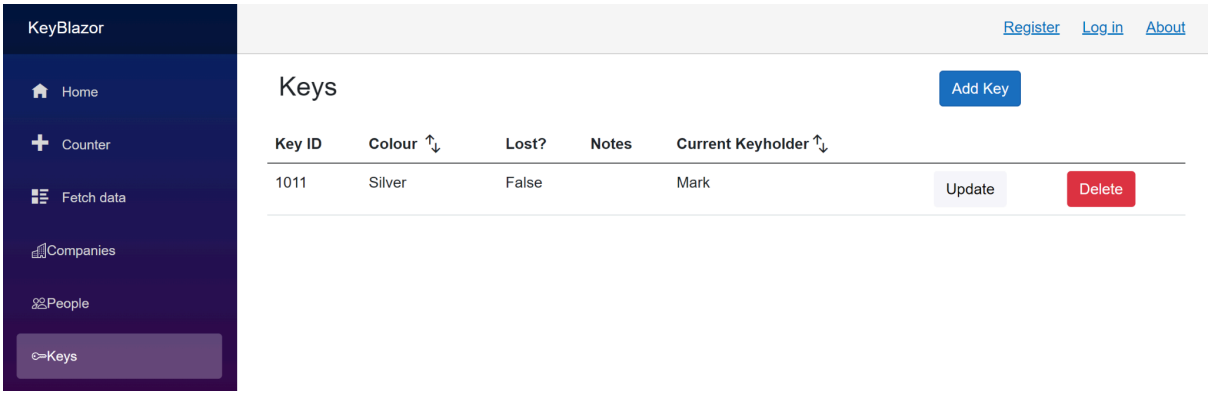
(without)

Though all the information is displayed on the page, there is very minimal formatting and overall it is not very usable or intuitively laid out. Therefore, I want to change the formatting of my pages so that they are all aesthetically pleasing and provide further functionality to my program.

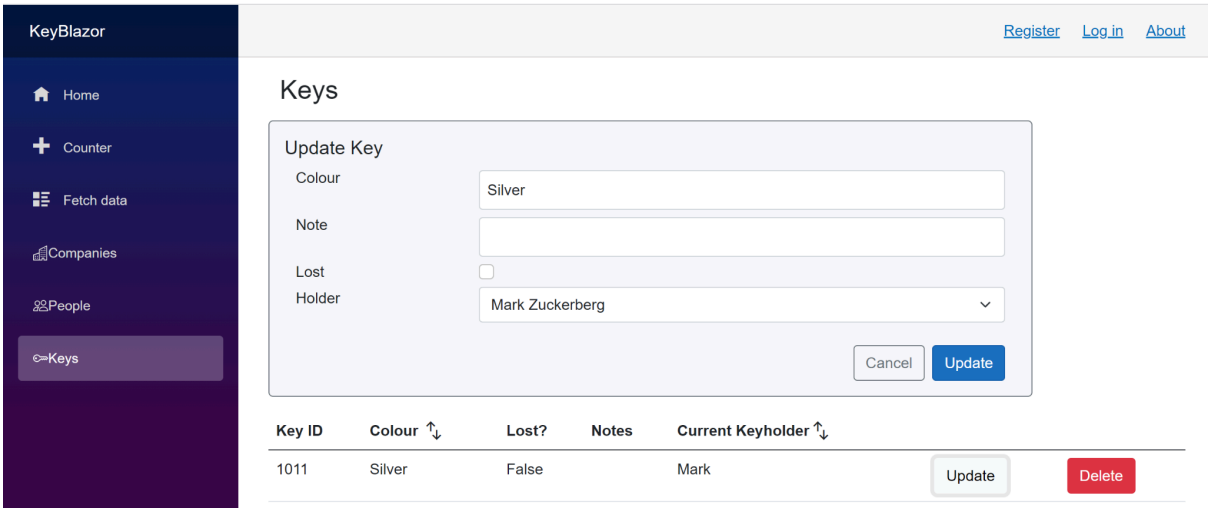
I went to my uncle and expressed to him my concerns, as I knew that he as an expert in web design could help provide me ideas on how to best address some of the issues I had with aesthetics. He suggested to me that I set up an imitation page that used my code, but had added features in terms of layout and visual components that made more sense in alignment with some of Nielsen's Heuristics:



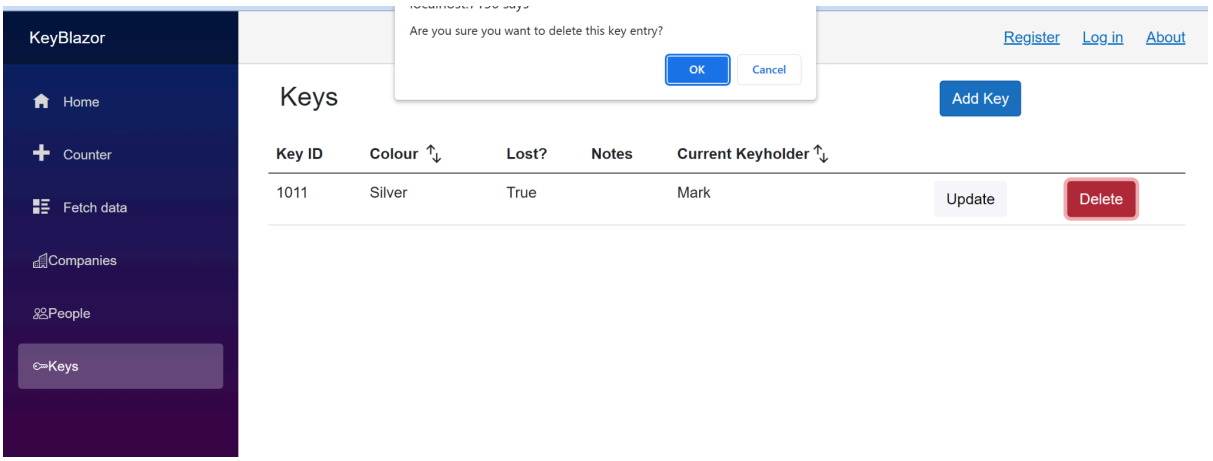
When you have nothing in the data table, the ‘New Key’ container is automatically present.



Adding an entry collapses the New Key section.



Updating a key is also possible and the menu reappears.



Deleting also allows a pop up box to appear as a confirmation.

Below are all of the usability heuristics I considered and what I did to ensure my website met with all of them: .

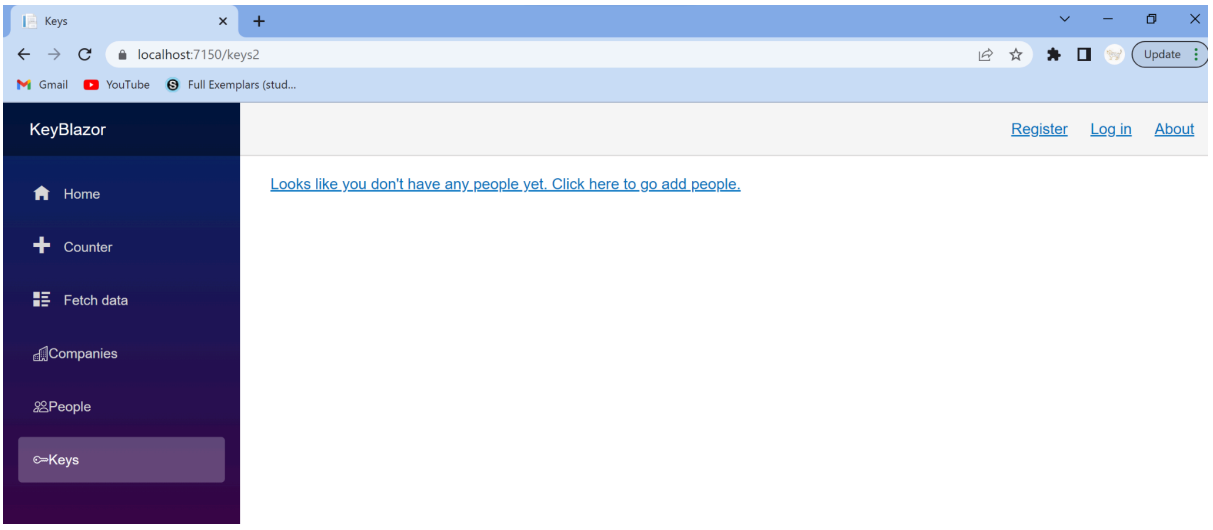
Visibility of system status. - It should be clear to the user that they are adding, deleting, etc, when doing that thing. - It should also be clear when a task has been completed/carried out that this has been done. - Users should be able to tell when they are hovering over something. - Users should know what tab they are on at any time	- When adding a key, the new key form is displayed, and when updating, the title is changed to update key. Taking action on these things requires an update button clicked or an add button clicked - so the action they are doing matches with the button they are clicking. When deleting, the confirmation box says are you sure you would like to *delete* this item. - Users are provided with a message as to whether their submission was successful. - Update and cancel buttons fill in; delete and add buttons darken on hover. - Blazor inbuilt shows a darkened region around the tab in the navbar you are active upon.
Match between system and the real world. - Colour should align with what is expected (eg red for delete button, green for go button).	- My delete button is red, which is intuitive as a caution/danger signal. The icons in the bar on the left align with the things that can be done on

	that page. The blue button for add is aligned with what users see in everyday life with other apps like google without being too visually overwhelming.
User control and freedom. - Users should be able to cancel, update, delete, and add at their own free will, except for in situations where this cannot be executed (i.e. things cannot be deleted). - Users should be able to delete or change things if they make mistakes.	- Users can add, update or delete at any time. Delete function is only available on each entry, i.e. won't be able to delete a non-existent entry. - Users have free will to update and delete any of the key entries they have recorded in the table.
Consistency and standards. - If there is nothing to be deleted, don't display a delete button. Make all delete buttons exist and appear the same way.	- The delete buttons only exist in line with the entries, so deleting just deletes that one entry. All update and delete operations work in the same way as one another.
Error prevention. - The program should prevent errors in the first instance, or warn users when making unchangeable edits.	- Program prevents problematic data from being entered. There are warning messages provided when people go to delete.
Recognition rather than recall. - Users should be provided options (ie when selecting people from a list), rather than having to go back and check, or try to remember themselves.	- Users are provided with a drop down list of people, and companies.
Flexibility and efficiency of use. - Things should be executed in a minimal number of steps whilst	- No unnecessary fluff - process is minimal and easy to follow

still being clear and fully functional.	
Aesthetic and minimalist design. - Page should not be visually cluttered, should be visually appealing and easy to navigate.	- Page is visually simple and contains entirely necessary information. Nothing that is unnecessary is displayed.
Help users recognize, diagnose, and recover from errors - Provide helpful error messages that explain what has gone wrong, and what to fix	- Error messages suggest what is missing.
Help and documentation	- My product is clear and simple enough to not grant this necessary.

Warning for having no people in list

I decided to have it so that none of the above is visible when there are no people in the database. This is because when there are no people, there is no possibility to add a key, and therefore having the functionality for adding a key available would mean that the user could cause the program to crash. Therefore, I decided to make it so that if there are no people in the database, only a link to the people page is displayed, otherwise the content of the key page is displayed instead.



Confirmation messages

I want to make it so that when a key has been added, the user gets some visual feedback. Though the table does expand, this may not be enough of a visual cue for

the user. The same applies for updates. It should appear above the table so that the user doesn't need to scroll to find it.

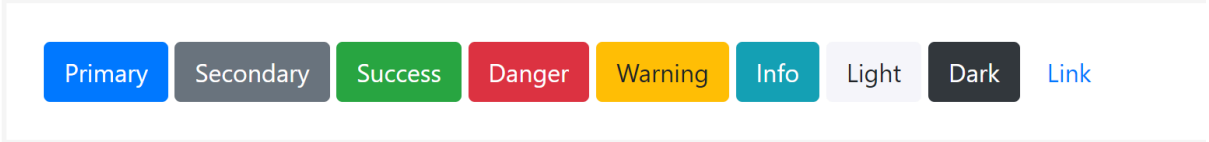
Again, this could be done in multiple ways -

- When the add or update button is clicked, a message appears.
- As things are being updated as they are typed or selected, the update message appears (while the update form is still open). The add message appears after the button is clicked.
- Visual cues are provided ie a checkmark for the add (though there is no obvious alternative for the update)

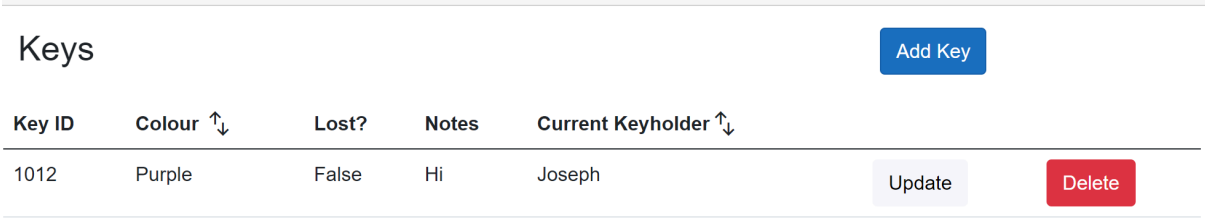
Consistency was one of the heuristics and so I think it is best that whatever I do to one I should also do to the other. I also think that encouraging the user to click the button for that confirmation message is a good idea as this is what is necessary for the add button and therefore it makes sense to have consistency between these two components. Therefore, I will make it so that a confirmation message appears once the button has been clicked, for both adding and updating. This has also been done for the delete function so that the user knows their delete has been successful.

Button colours

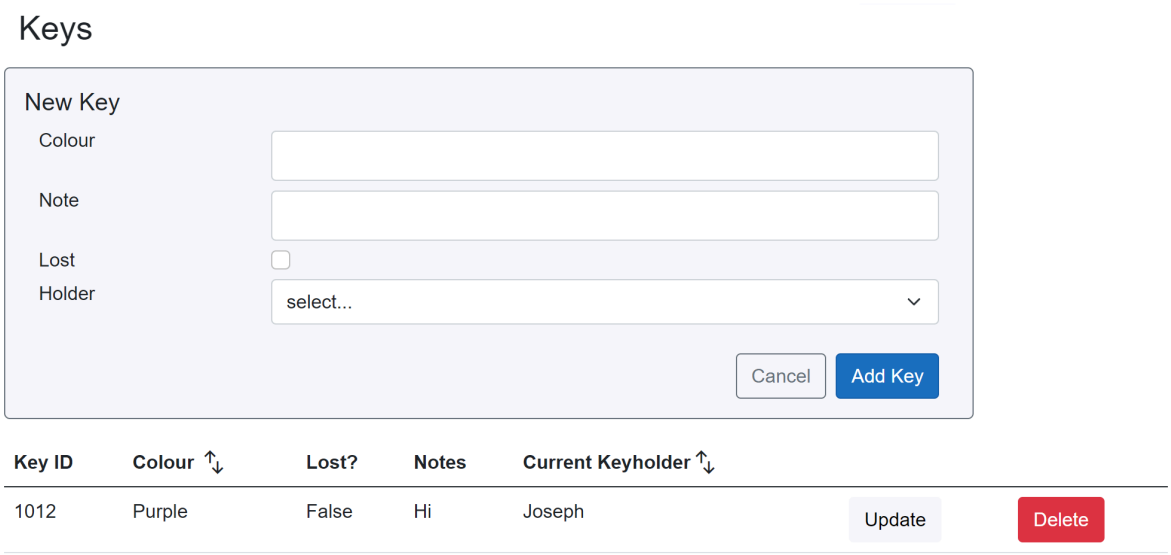
Below are some of the pre-programmed options for buttons in Blazor. There are also the same buttons available with exclusively outlines.



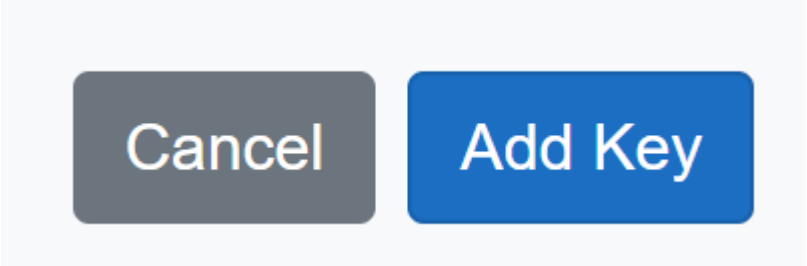
The current button colours on my form are:



(With the add menu closed)



(With add menu open)

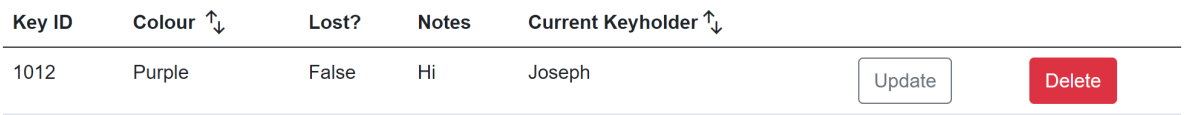


(Hovering over the cancel button).

I have my doubts about some of the colours of the buttons. It makes sense that the delete is red, as this is like a warning colour. However, the update button is a light grey which is hard to see, and the cancel is just an outline which activates once the mouse hovers over it (above). The add button is also blue where I feel it may be more intuitive to make this green, as this is an executive colour. Dad has no issues with colour blindness or visibility, and is flexible with colour/artistic choice, so in this regard I have complete freedom. However, I still want to ensure that for a client base wider than just Dad, I would be able to cater towards any needs, ensuring it is fit for purpose in the broadest sense. I took my concerns to some stakeholders (Alisa, Eliza, from my 3DTP Class, and Ashleigh, a design student), who agreed with some of my thoughts and suggested various changes I could make.

<u>Update button</u>

option 1: Outline, secondary colour



Option 1 with mouse hover:

Key ID	Colour	Lost?	Notes	Current Keyholder		
1012	Purple	False	Hi	Joseph	Update	Delete

Option 2: No outline, secondary colour

Key ID	Colour	Lost?	Notes	Current Keyholder		
1012	Purple	False	Hi	Joseph	Update	Delete

Option 3: Dark, no outline

Key ID	Colour	Lost?	Notes	Current Keyholder		
1012	Purple	False	Hi	Joseph	Update	Delete

Option 4: Dark, outline

Key ID	Colour	Lost?	Notes	Current Keyholder		
1012	Purple	False	Hi	Joseph	Update	Delete

Option 4 with mouse hover:

Key ID	Colour	Lost?	Notes	Current Keyholder		
1012	Purple	False	Hi	Joseph	Update	Delete

We all agree that the hovering change is better, and it was also pointed out that this visual indication is better for showing status and for colourblind people to be able to see a difference between the blank outline and the filled delete button. However, the drastic change from light to the 'dark' we were all unsure about as we thought that the dark update made it seem like a more warning task than the delete, and also that the massive contrast from being on and off the button would be too much. Therefore, I've decided to go with the secondary grey, with an outline (option 1).

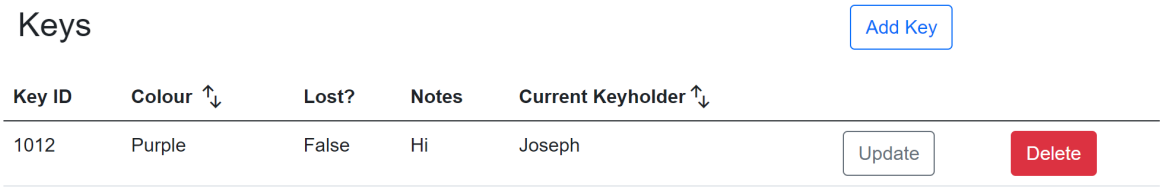
<u>Add button:</u>

Option 1: Primary colour

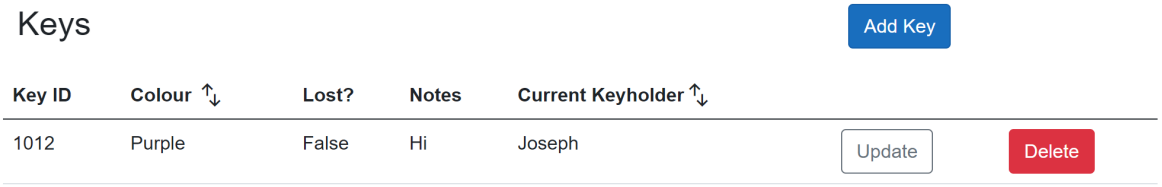
Option 2: Success colour

The general consensus from myself and Alisa, Ashleigh and Eliza is that blue is the way to go. The green is too full on and also seems like too much of a commitment, which could potentially cause anxiety with the client. The buttons I am familiar with,

such as those on Google, are all blue, so this helped me decide that the blue button is the better choice.



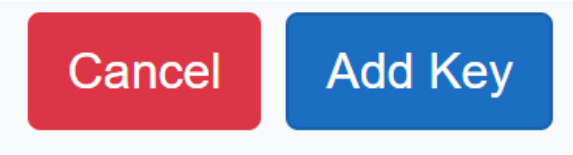
I could use an outline, however this makes it seem too insignificant and I want it to be the primary action on the page. Therefore, the filled in blue is the way to go.



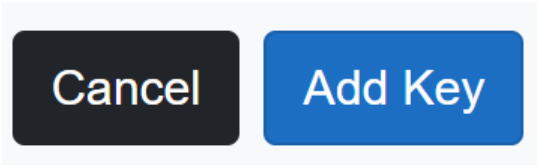
Final choice for the add button.

<u>Cancel button:</u>
Option 1: Red outline

Option 1 when hovered over:

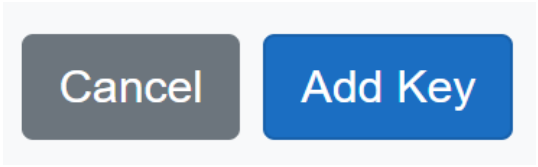


I think the red is too visually overwhelming, but I like the idea of the outline, which makes it seem like a primary task compared to adding the key.



Once again the black is too dark and competes with the add button.

Secondary colour:



Though this is the original colour, I do think it's better than any of the alternatives so I will use this for the button colour. It is less intense than the others and does not compete with the add button which is the primary action.

Side bar

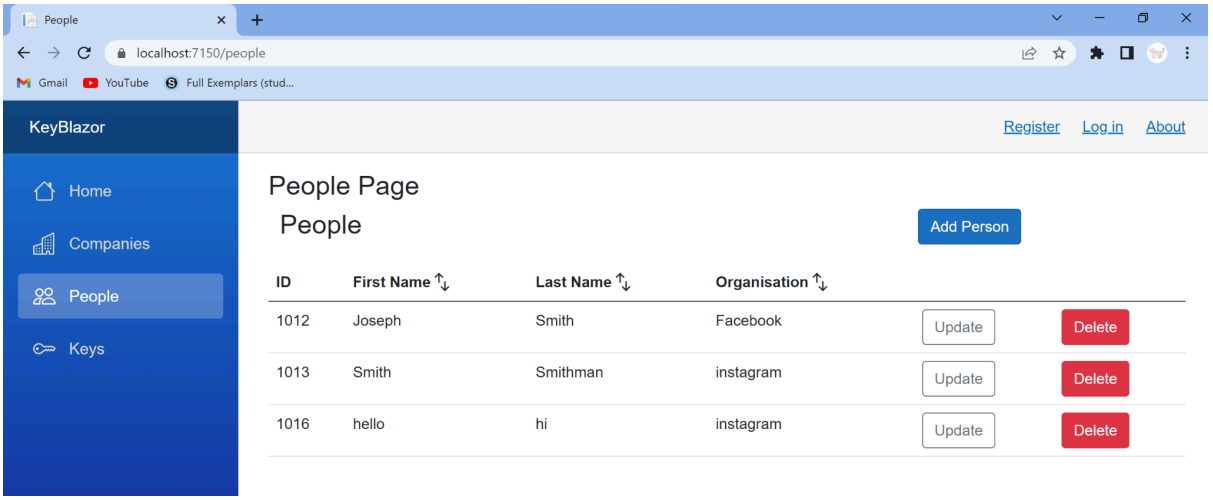
While looking at the aesthetics of my website, I can also look at the colours of the sidebar. I decided to do some research as to what colours represent what I want the website to get across, and I think the idea of using blue would be good as it represents professionalism, intelligence and trust. White could also be good due to its cleanliness, as well as the colours that neighbour blue (purple, green). Having a gender neutral sidebar is also important due to the project ideally being versatile for many people.

- red: caution, anger, love, negative (in finance), hot
- orange: warm, autumn
- yellow: happy, fun, young
- green: nature, calm, good luck
- blue: stability, professional, cold, trust, intelligence
- purple: wealth, mystical, decadent
- brown: rustic, practical, warm, vintage
- white: sterile, innocence, peace, truth, cleanliness
- black: sophistication, death, night, contemporary
- multicolour: international, all inclusive, multicultural

I also asked for Dad's thoughts and he liked the idea of blue, indigo, or green. I will trial a few combinations to see what is cohesive with the site.

From LTR: rgb(5, 39, 103) 0%, #3a0647 70% / all blue, gradient - #5263a1 0%, #1d2c61 70% / all blue, lighter blue, gradient - #5d7be3 0%, #1736a3 70% / Blue to green, gradient - #162570 0%, #2c6963 70% / Blue to purple gradient - #162570 0%, #4c318f 70%

I presented Dad with the above colour options and he thought that the best choice was #3, with the two lighter blues having a gradient. While I agreed in that this one was visually appealing, and less moody than some of the other options, I thought it would be even better if I matched the upper blue to that of the blue buttons on my form, so I found the hex code of the primary blue and added this in place of the first blue to get the below gradient:
#4287f5 0%, #1736a3 70% as below.

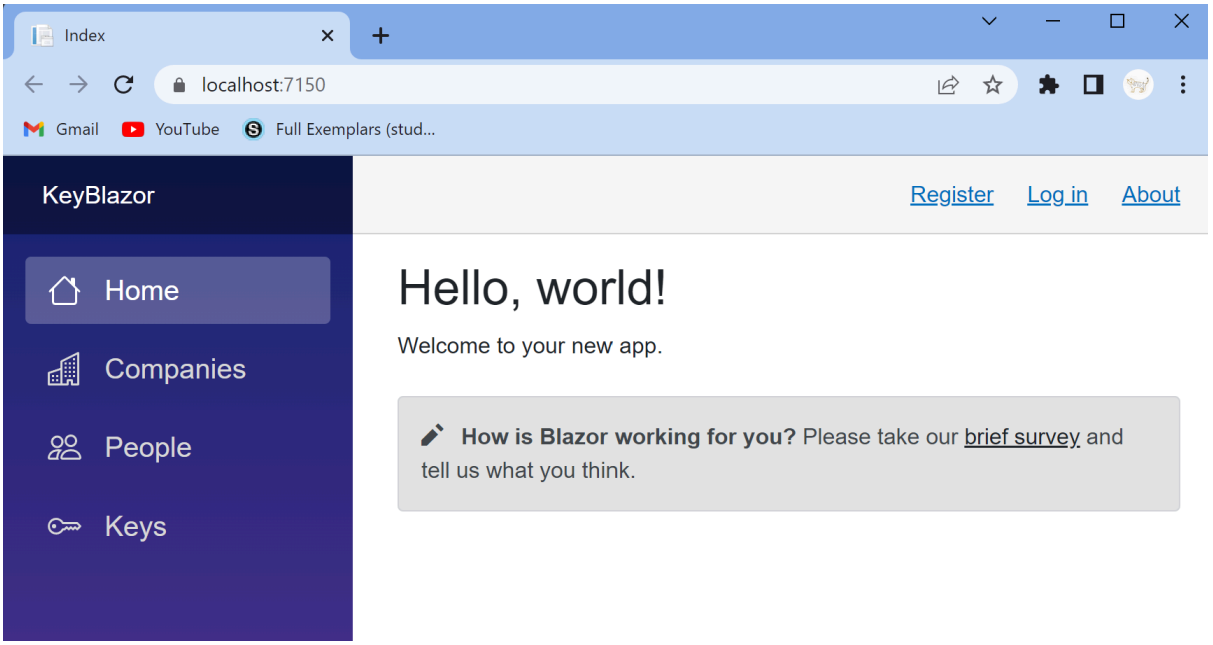


This provides a much more cohesive overall view which I think has improved the aesthetics overall. It being a bright blue also means that there are no obvious gender

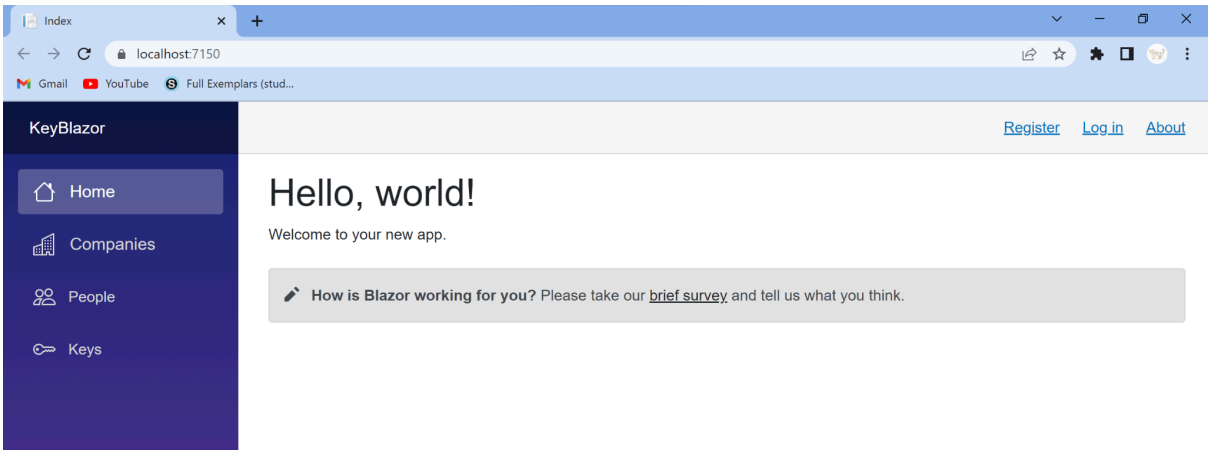
considerations, where the site seems too masculine or feminine. I think it hits the right balance and therefore is able to be used by any gender.

Text size

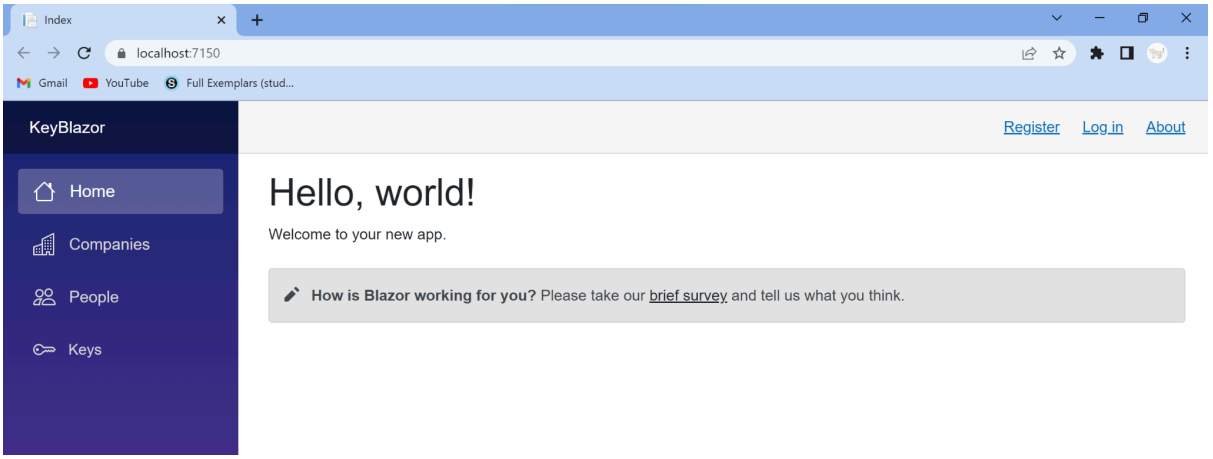
Next was to consider the size of text (Note I did some of this before hearing back from Dad hence the colour of background!)



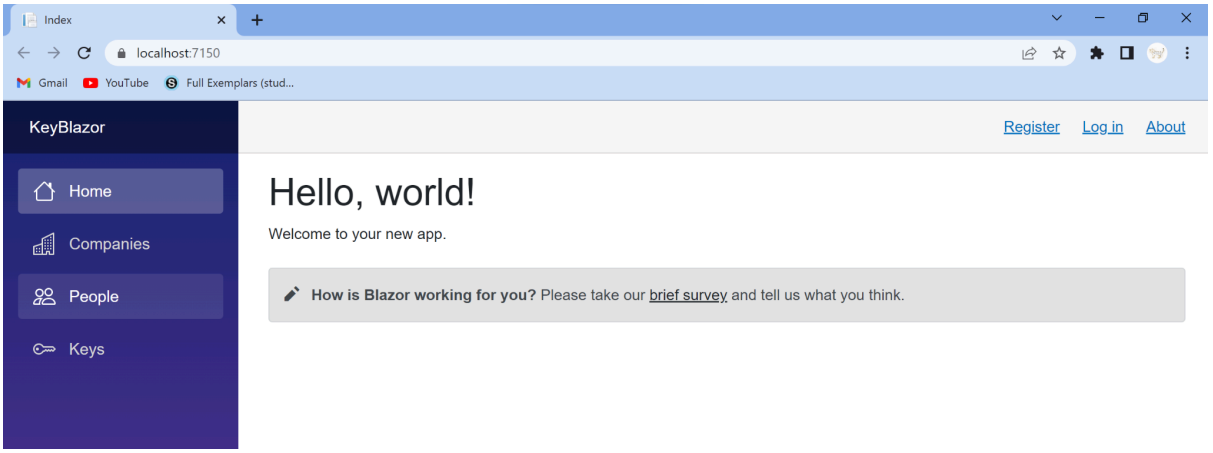
20px - I think is probably a bit big, especially as it is a lot bigger than the text on the page. It also potentially overwhelms the icons and the gap size between the icons and text, to where it seems visually cluttered.



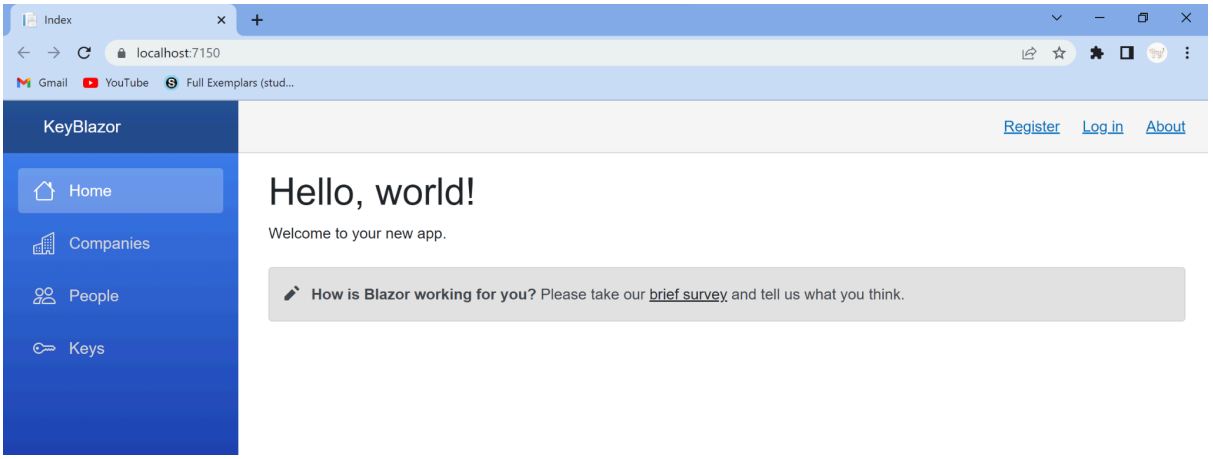
Top 2 are 18px, bottom are 16px - I think 16px is the better fit, though it now may be too small.



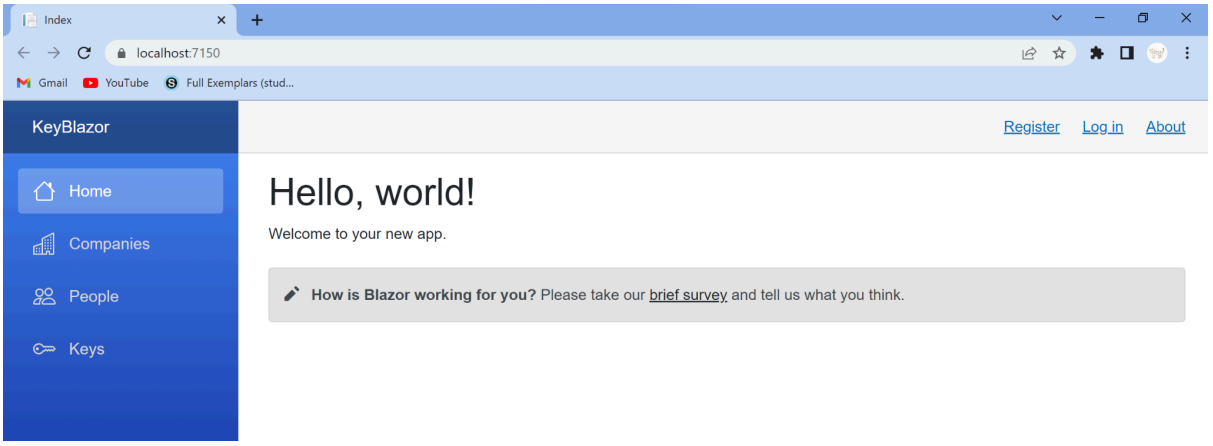
18, 17, 17, 16 in order. 17 is the best in my opinion - large enough to read and slightly larger than the primary text on the form, but not too large to overwhelm the icons.



Above is the form with size 17 text, which I think fits perfectly.



I increased the size of the KeyBlazor text just slightly so that it overheads the rest of the items in the navbar. However, I think it looks funny being out of line with the icons and text below.

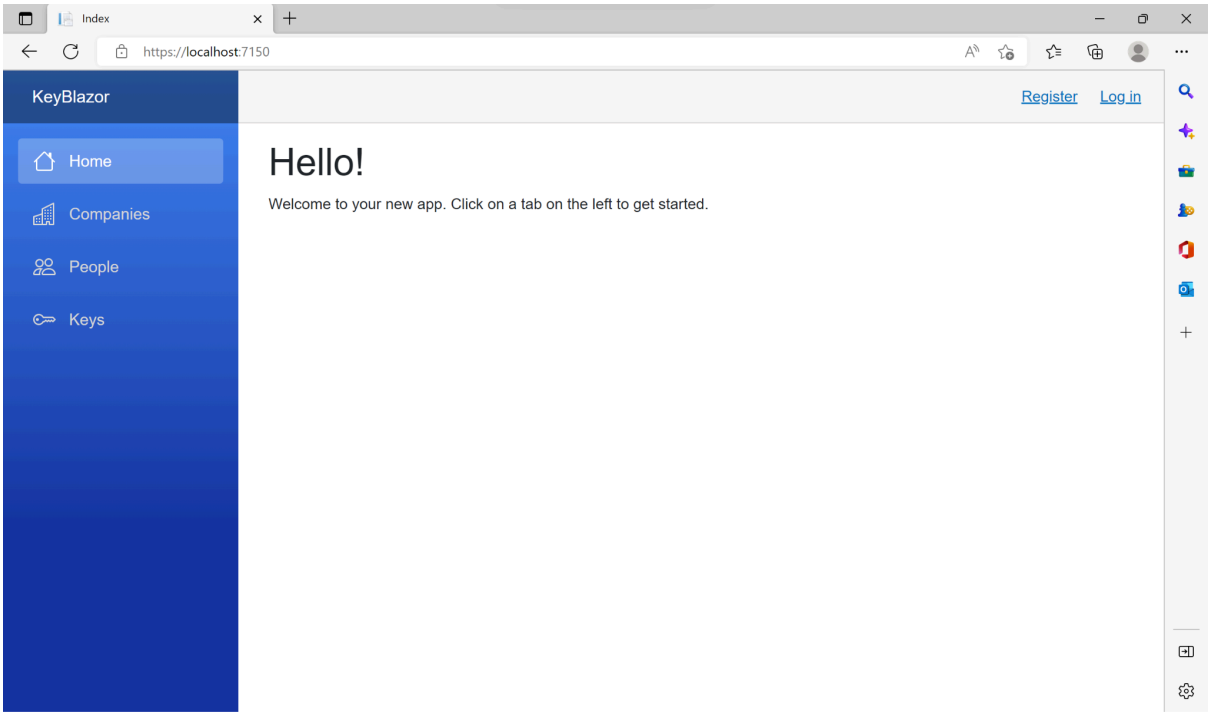


I reduced the size of the padding in the left hand side of the KeyBlazor text and think that this has fixed the misalignment issues for a very cohesive sidebar.

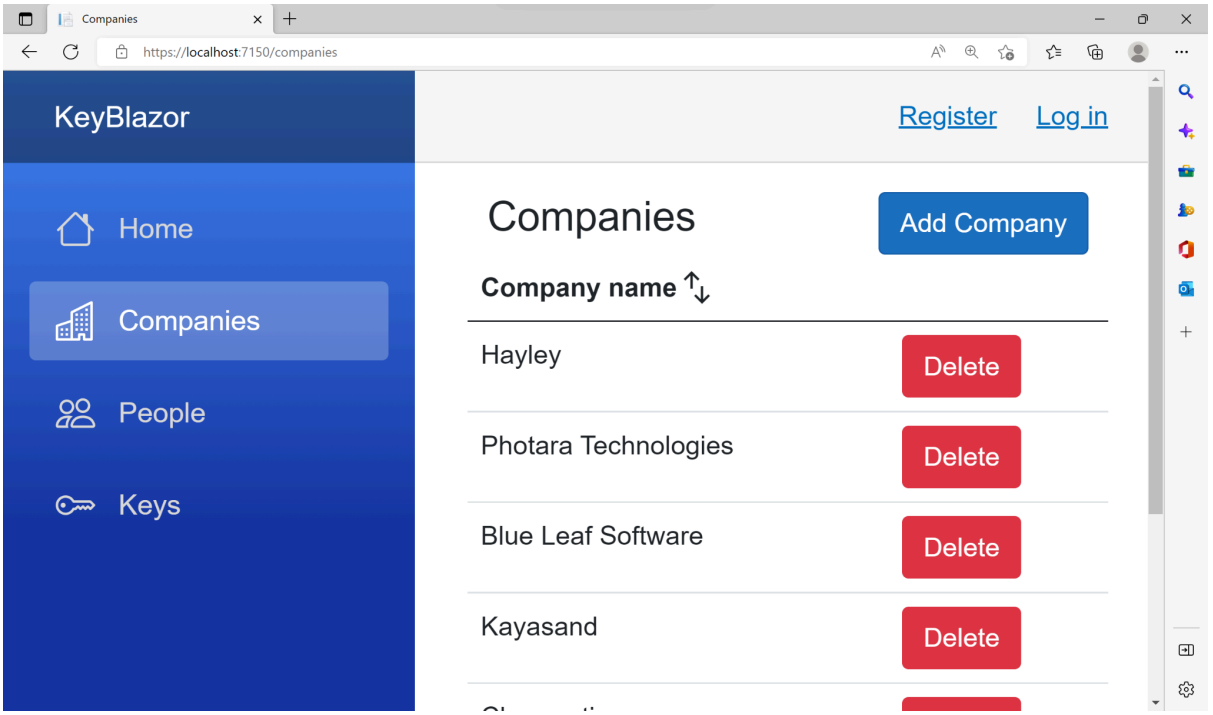
Reflection

Section 3 has helped me polish the look and feel of the website. Though many of the choices in this section didn't contribute to the function of the app, they have improved the professionalism and in some ways the usability, further meeting the specifications I set out in my brief.

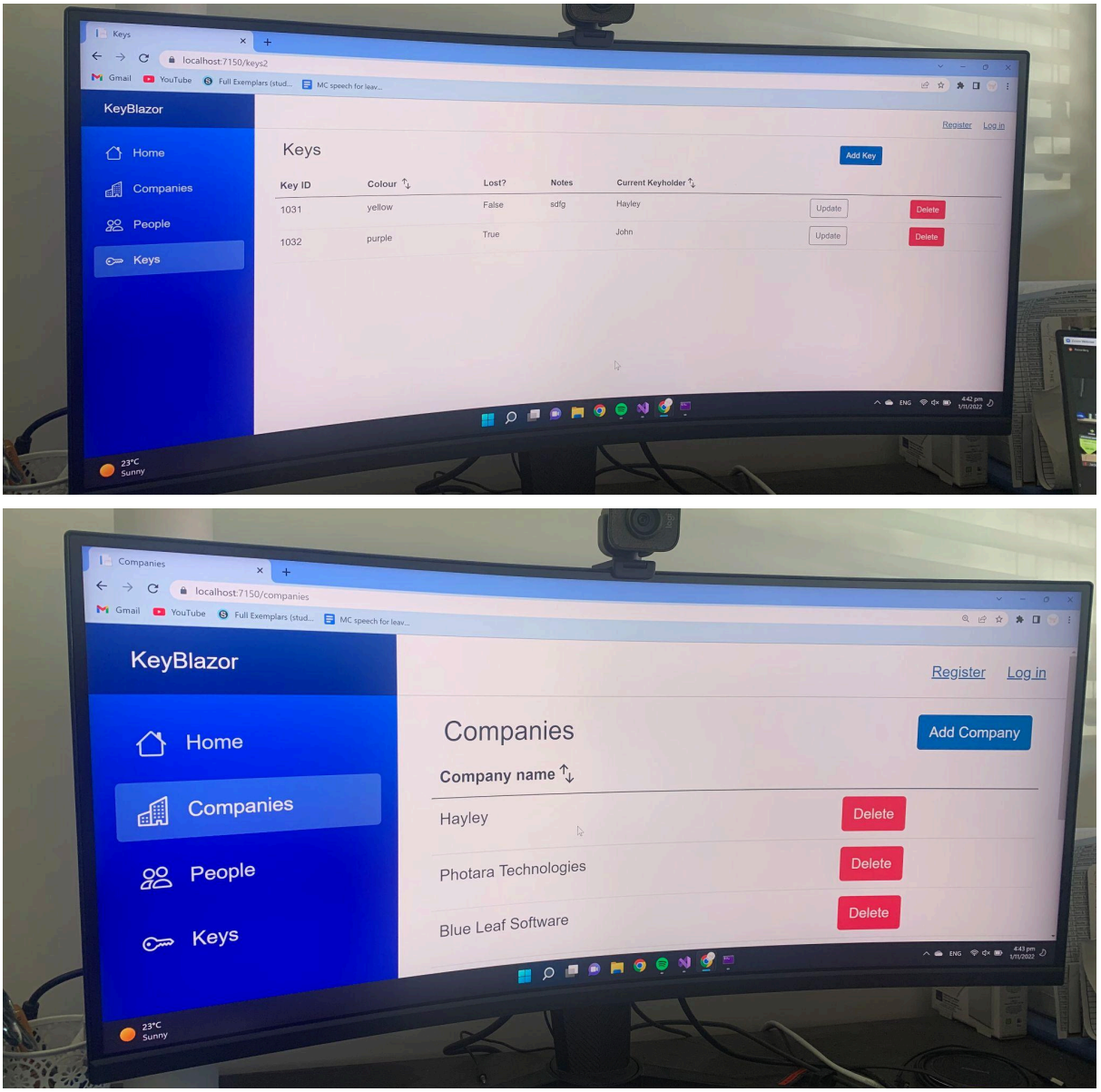
Testing versatility



Testing different browsers - Microsoft Edge.



Testing the zoom - this is zoomed to 175%. All components have resized accordingly.



Testing on Dad's large monitor with a very different aspect ratio to my computer (1) and zoomed to 180% (2).

User Testing

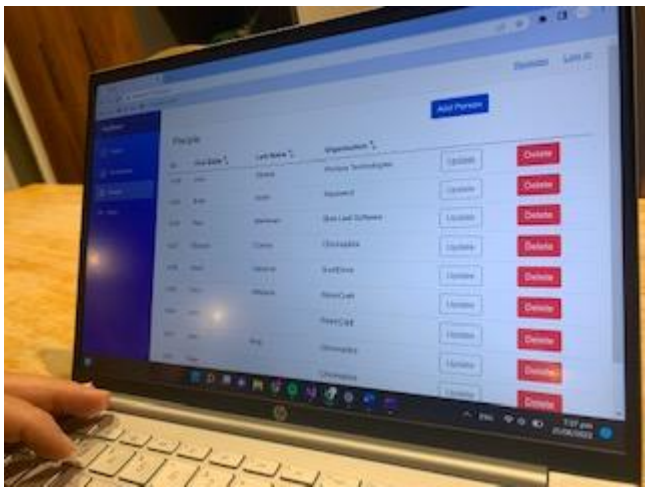
Snips below - Dad testing my project.

People

Add Person

ID	First Name	Last Name	Organisation		
1026	John	Sharpe	Photara Technologies	Update	Delete

Dad has put himself into the database.



Testing the application with the tenants and companies of the building.

Dad really liked the outcome I had created. He commented on how it was really useful how he was able to alphabetise the lists to order things by company or by whether they were lost or not. He also really liked the layout of the webpage, which he had contributed to the design of. He even gave some suggestions of things he hadn't recommended, but after seeing the finished product, realised he would find it beneficial (such as imagery on the home page). This gave me satisfaction as though he initially wanted a very simple outcome, upon seeing it, all of his initial requirements were satisfied, and the only things he had to suggest were additional features. Once he had seen the finished product, he could see the potential in having the app available when he travels, and this is when he began to think of some additional features. Overall, he was very pleased with the end result of my project in its current form.

It is worth noting that for a large amount of the time I spent developing my project, Dad was overseas travelling for work. This meant I had a high amount of independence in developing the outcome for him, as he was often unavailable, and couldn't provide ongoing feedback as much as would have been ideal. Had I been able to get his feedback on a more regular basis, we both could have found more features throughout the process of development which he would have liked to have added into the program.

This definitely helps me to see the benefit of the agile method for project management due to the iterative nature of development that can be carried out using it. Although the program is 'finished', knowing that Dad would want some additional features, I could quite easily put this into my Trello board and continue working through them.

Conclusion

I set out to create a permanent database within a website, for my Dad to use to keep track of the keys being held by tenants within his building. This has been achieved by using Blazor, a web framework, to create a thoroughly tested relational database, which can store visible data, and be edited with deletes, updates, and additions.

Project management tools and techniques allowed me to manage all stages of my website development. I used version control to ensure that at all stages I had a secure version of my project. Project management was done using Trello, which helped me to track the development of my project. As I moved onwards and received feedback from stakeholders I was easily able to add this into the early stages of my Trello board to be continually developed. I also could add new features at any stage and cancel features at any stage of the project as I had the flexibility to move things around as necessary. The use of project management was very helpful in ensuring that all aspects of my outcome were completed to a high standard, therefore helping it to meet all of the requirements of my brief.

Good programming conventions were used throughout my development process. These included appropriately named variables and correct syntax. Additionally, my code has been thoroughly reviewed and commented to explain every piece that is difficult to understand or any confusing terminology. Therefore, the program I have written is easy to follow for not only myself but also others. This should ensure it is able to be maintained in the future by myself or anyone else who has an interest in the upkeep of the project.

By following the technology process, I have gained information through research initially, when deciding what software to use. I identified and gathered information from stakeholders, including classmates, parents, teachers, and my uncle, who is an expert within the field. This allowed me to trial different types of software when comparing the research I did with the trials I had carried out and the things I was recommended. Trialling and testing also allowed me to find alternative ways of doing things and reevaluate what the best strategy would be. All of this in combination has led to the development of the website I have created. I have been able to synthesise the requirements of my brief, the trials, tests, and research I have carried out, ongoing interactions with stakeholders and with experts, into a final product which combines the best of all stages of creating this outcome.

Carrying out the execution of this project has been no easy task, however, I have managed to create a successful website which meets the needs of my Dad, the end user, and the specifications I set out with. The site I have created follows the needs of any database, where the user has CRUD functionality (create, read, update and delete). This has met the permanent data storage specification, where Dad is able to store all of his data on the website, as well as edit it himself. The input system as well as any actions he takes with updating or deleting have been thoroughly tested to ensure that Dad has a smooth and efficient experience using the software.

The website has high usability and a streamlined process to match Dad’s busy schedule. He is able to accurately input information, with no unnecessary information on the page. The website design takes into account Nielsen’s Heuristics, which I have followed to ensure that my website uses the best possible practices of layout, usability and aesthetics. All components have been thoroughly tested to ensure the best possible outcomes and prevent there being any issues for Dad further down the line.

By creating a website on Blazor, the website could be used on any device. I carried out thorough testing of the site with all the available resources I had possible. This involved using different browsers (Chrome and Microsoft Edge), changing the scale of the website (zoom), and using different monitors, including a desktop and a laptop, with different aspect ratios. This gives me confidence that the website would be scalable to Dad's smaller devices such as his iPad and iPhone should he want to run it there. This was all the testing that was able to be done prior to the site actually going live, and was done as thoroughly as possible.

A constraint I had in creating this outcome would have always been time. I was limited in the time I had from start to finish of this project, and furthermore the time I could spend actually coding, as a large portion of my time was invested into initial research. There are risks and struggles associated with getting the site live, particularly that there is difficulty with ensuring the safety and security of the website. Although unlikely, it is possible that, when putting the website onto the cloud, even with security in front of it, the site could be attacked. This would put the information of the people in the database at risk. I understand the need for safety of the data and data integrity with running this project, and so did not want my limited time to mean I did a negligent job of getting the website live and functional, potentially risking the security of the individuals who are stored within the website. The privacy of the people within Dad's building is far more important than doing a rushed job of getting the site live. Though my outcome is still a prototype, I would hope to, in the future, get the site running in the most secure way possible, so that it can eventually be used by Dad on any device.

Though the purpose of this project was to create an outcome for Dad, there are others who can benefit from my outcome. Murray can use the website to track the keys he gives out to various community groups who borrow the gym for sports activities. Miss Stonestreet, my DVC teacher, can use it to keep track of who she is lending out supplies to, to ensure they get returned at the end of the year. Although Dad is the primary client, the simplicity of my project and its ease to be duplicated means that the project could be used easily by all of my stakeholders.

Overall, I have enjoyed developing this project for my Dad to use. Knowing how busy he is, I hope my project can help him in leading a more efficient life.