**Contents**

**Table of Contents**

**1. Analysis................................................................................................................................2**

**1.1 Introduction ......................................................................................................................2**

**1.2 Investigation......................................................................................................................3**

**1.3 A general overview of my objectives ..............................................................................11**

**1.4 Specific overview of my objectives..................................................................................12**

**1.5 Prototyping......................................................................................................................13**

**2.Design.................................................................................................................................17**

**2.1 High level overview .........................................................................................................17**

**2.2 Descriptions of modular systems ....................................................................................22**

**2.3 Data dictionary.................................................................................................................23**

**2.4 Validation.........................................................................................................................30**

**2.5 Database Design ..............................................................................................................31**

**2.6 Planned SQL Queries .......................................................................................................34**

**2.7 UI Design .........................................................................................................................37**

**2.8 Security ............................................................................................................................46**

**3.Technical solution ..............................................................................................................47**

**4.Testing...............................................................................................................................123**

**4.1 Postman.........................................................................................................................123**

**4.2 Testing Strategy .............................................................................................................124**

**5.Evaluation .........................................................................................................................136**

**5.1 Objective Check .............................................................................................................136**

**5.2 Feedback from users .....................................................................................................138**

**5.3 Potential Improvements ................................................................................................141**

**Analysis – 1**

**Introduction – 1.1**

**Background**

For my NEA project I will be making a social networking website that will have profiles, groups, events and a calendar.

I will be coding in JavaScript and HTML, for the database I'll be using MySQL and using Docker to hold the table in a container.

**Problem**

A lot of the time when you’re working or in school studying, you’ll end up having a very busy schedule and you will find that you and your friends never really have time to get together. My aim is to find an easy way to discover whether you and your friends are free for a night out or not. By creating a calendar that allows for multiple people to input their appointments and schedules all you have to do is look into the calendar and find a day where none of your friends have something planned making meeting up with your buddies exceptionally easy.

**Limitations of Current Situation**

The current limitations of social networks such as snapchat and Instagram is that they tend to stay in their lane in terms of what they're good at. For instance, snapchat does not copy Instagram and have a profile with people's pictures from years ago. Why not have a social network application that can have everything the user wants? You could combine all of the greatest features of every social network out there.

**Target Audience**

My project will be aimed at all ages from teenager and above as everyone starts to use a calendar from roughly secondary school in order to stay organised and on top of work. It can be used by younger people to organize school group projects or meet ups with friends. It can also be used by older people in terms of meetings for work or lunch

**Investigation – 1.2**

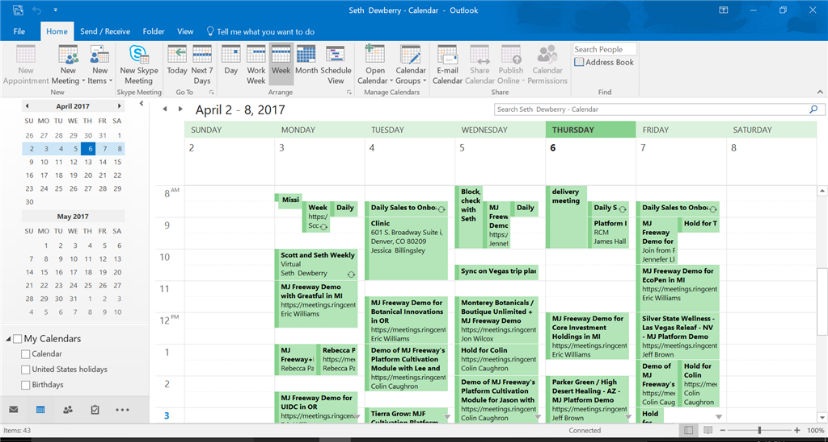
**Preparation**

At the start of year 12 we were learning to code in python which was a language I had a bit of knowledge of since I learned and used it quite a bit during GCSE so naturally when looking at possible ways to code this project, I tried to search up ways python can work with html. I didn’t find many possible options, one of the few was flask which is a web framework for python although I found it all to be rather clunky and in the end I decided to not carry on using this. I decided to look at other languages and quite quickly found that html very easily links up with JavaScript so I looked into using this for my project. Researching JavaScript showed me how easy it was for me to link databases then to my project by maybe using PHP or node js.

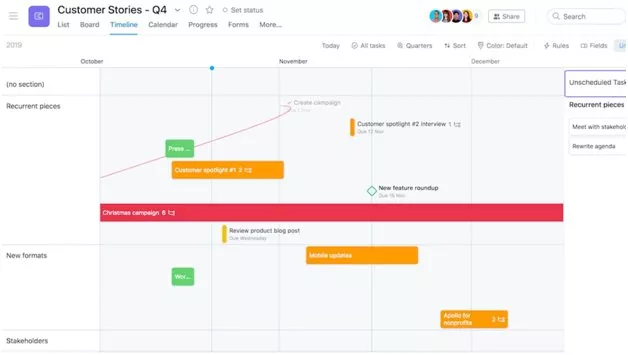
So I came to the conclusion that I was going to use JavaScript and HTML because it led very easily to database accessibility and the code was all very nicely laid out to be understood. There were also tones of learning resources on the internet which I needed because it was a new language to me.

**Inspiration**

**Outlook Calendar**



Outlook calendar is a very well-known calendar. Since a lot of people use outlook as their primary email the ease of having a calendar connected to your email appeals to a lot of people. It also has the ability to share your calendar with co-workers which is the start of becoming a group calendar but it’s not quite there. Another application is that you can create additional calendars for specific projects which influenced my idea of having separate calendar groups for things such as work, friends or school. A disadvantage of the Outlook calendar is that the GUI is close to identical with Googles Calendar showing a lack of thought in terms of the creative side of the calendar. Another disadvantage is that because the calendar for Outlook is almost like a side piece for the email it’s quite difficult to locate it from your email especially if you’re new to Outlook.

**Asana**

Asana puts emphasis on collaboration and productivity, the main feature of Asana is how flexible it can really be. You can use it to keep track of ongoing work and long term projects, you can also use it to coordinate with other teams and distribute workloads among your co-workers. Asana manages to put task management, communication and other helpful tools all within a single interface which shows how useful it really is.

The unique GUI (shown in the picture) is something I want to implement into my own calendar as I want it to stand out and not fall into the group of common calendars such as Outlook or Googles calendars. While by itself Asana doesn’t have a lot of useful features built in such as time tracking or live chat it achieves a lot of functionality through app integrations.

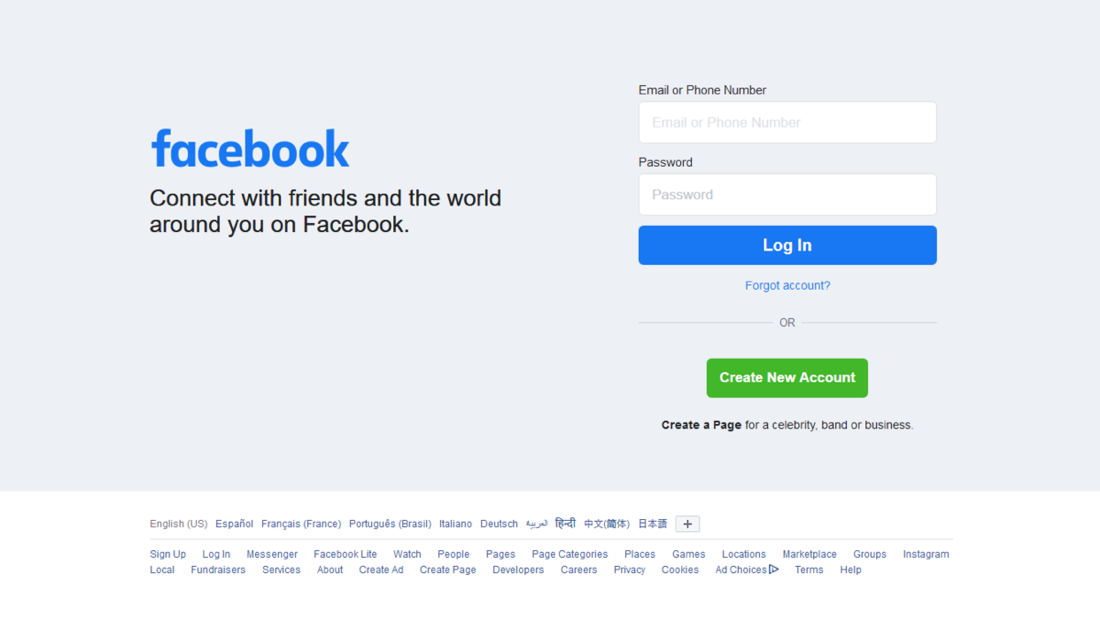
It works with slack and Microsoft teams for live chat and Harvest for time tracking. This could be seen as a disadvantage as it has to rely on other programs other than its own for all its features. This is something that I think I can improve on as these are things that I think should be implemented into a calendar by default.

A useful part of Asana that I plan on taking inspiration from is how easy it is to manage team projects as it has a timeline feature that shows every piece of a project, how it fits together and it helps you track the changes you or your group members make. This would be extremely useful for younger people in school who sometimes leave their projects to the last minute or struggle spreading the workload between the group evenly.

**Facebook**



While the other applications I’ve looked at have been solely calendars, Facebook differs from the trend. A huge part of my own project will be the social networking aspect and Facebook is a perfect example of this. Facebook uses profiles expertly by creating links between mutual friends and by having groups that people can join and chat in.



When looking at the Facebook website I found their login page to be simple and clean, with each of the buttons coloured and laid out to be seen easily. This was something I wanted to replicate on my own project as it’s user friendly and appeals to the eye.

Facebook also has tons of features that would take too long for me to create such as a feed which shows the user their friends’ recent posts and updates. While there are plenty of advantages to this website there are also plenty of disadvantages. They have a lot of privacy issues which when creating a social networking application is a major problem. Normally when creating an app like that a main point in creation would be cyber security but with the time constraints I have, this will be unable to be created but will still be mentioned on where it would normally be added.

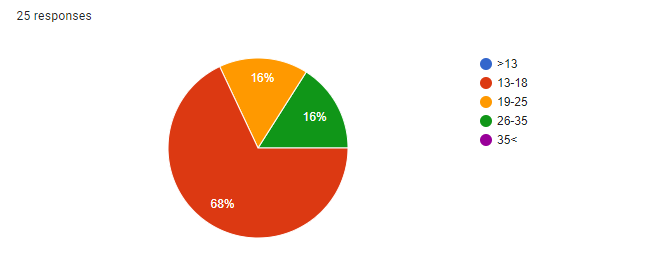
Social networking websites and apps are hugely popular, Facebook has 2.8 billion active users and is considered the biggest social network worldwide. This shows are useful applications like this are, there are multiple different websites and apps who are like Facebook such as Instagram or Snapchat and they’re also hugely popular despite being so similar to each other. This begs the question of what makes a social network website stand out?

* Target market penetration
* Better content display
* Active users

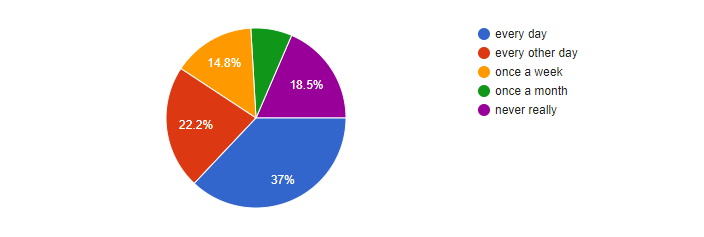
While these three points are very important when it comes to a social network, they're also extremely hard for me to implement but they’re interesting things to consider if one day I created a huge social networking application.

**Peer Survey**

I constructed a quick survey using Google Forms, which uses a simple interface to ask users questions, it also provides a useful analytic graphic of the results. Not all of the questions are mandatory as I felt some of them would require a bit more knowledge on calendars and in case they didn’t have the time to type long answers I left it as optional.

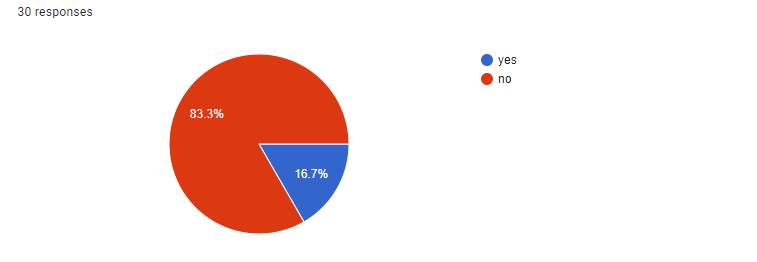
1. What age group are you in?

With most of my participants being teenagers or young adults I can assume they will all be used to new technology and to using calendars.

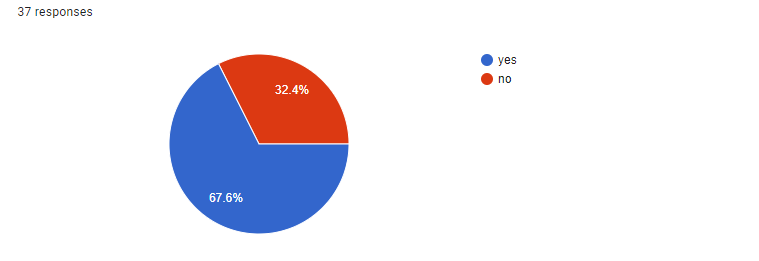
1. How often do you use a calendar?

This question provided a reasonably wide range of answers which the majority saying they check their calendar at least every other day and the next largest group saying they used it every day which shows how vital a calendar is to the ordinary person.

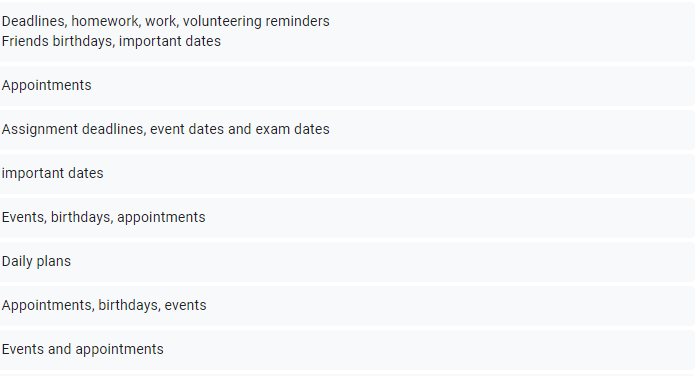
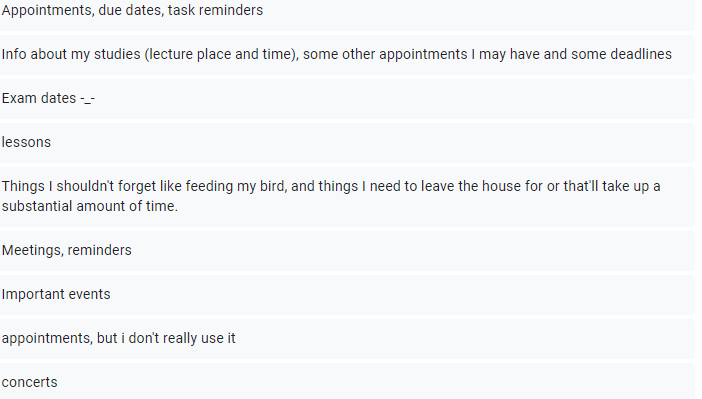
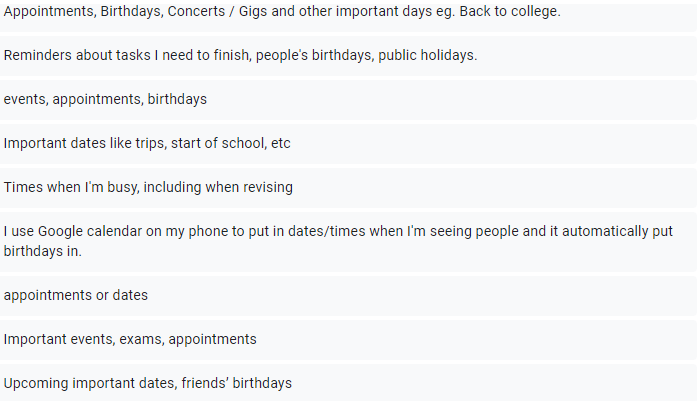
1. Do you have separate calendars for your work and social?

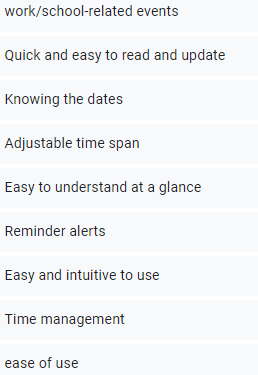
I found the results for this question to be quite interesting as I’d originally thought a lot of people would have separate calendars in order to avoid clutter in their calendars. However, I’ll still need to implement a way to have separate calendars because there could be an instance where the user is working with multiple different groups and needs to have different calendars for each of the groups they are with.

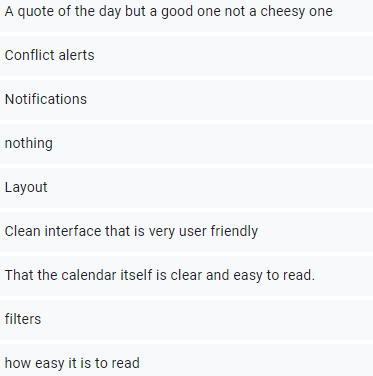
1. Would you benefit from everyone in your work group or social group adding their dates onto the same calendar so everyone could see when you're busy or not?

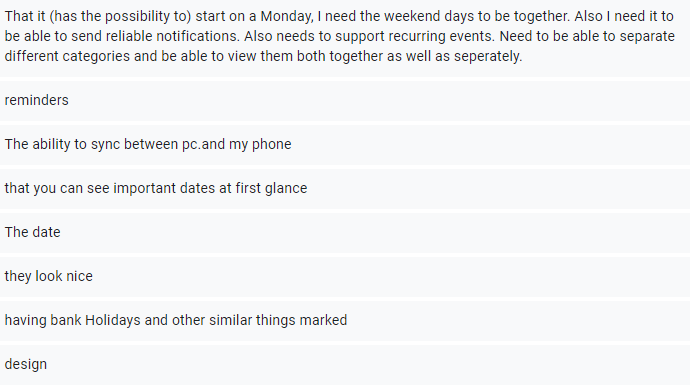
Two thirds of the participants said that they would benefit from having a group calendar in order to see when their work colleagues or friends were busy which shows that there is a market for this type of thing and a lack of a group calendar out there for public use.

5)what do you normally put in your calendar?

With this question the participants gave a large range of answers showing the importance of a calendar as it helps remind these people of appointments, birthdays, concerts and other things that they just can’t forget about.

1. What’s the most important thing for you when it comes to calendars?





“a quote of the day” – this is an interesting suggestion and can definitely be implemented but it’s not needed for the final project.

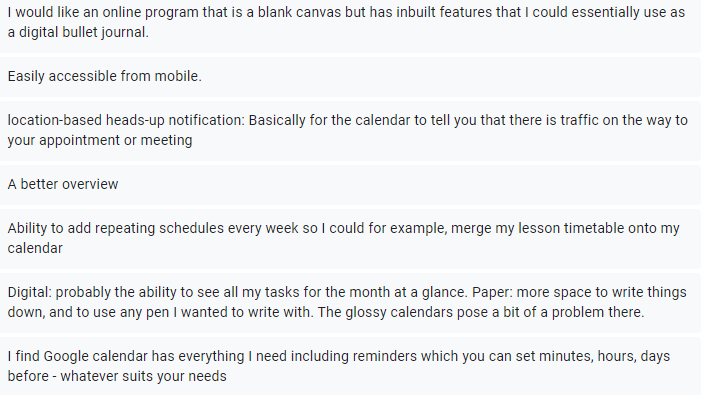
“layout”, “clean interface”, “user friendly” – it’s very clear that a lot of the participants felt that the user interface had to be clean and user friendly which is something I was hoping to implement as I feel that the interface is a main factor in what differentiates one calendar to another.

“customisable design, colour categorising for types of events” – giving the user a choice on the background and other things lets their calendar feel more personalised and they’re more likely to enjoy using it.

“conflict alerts”, “notifications” – this is something I planned on adding. I wanted to have a notification be sent to the user if they had been invited to a event or if one of their most recent changes to their calendar had created an overlap and was now conflicting with another event.

“that you can see important dates at first” – I plan on having the user set priorities on certain events that they add to calendar that will notify the user when it’s coming close to the date in intervals (a week, 3 days, a day, etc).

7) What is something you'd like to see added to make a calendar better or more efficient?



“location-based heads up notification” – this is a particularly good suggestion but could make the project become quite difficult and end up being more about this instead of the calendar.

“repeating schedules” – another good suggestion as to appeals to people who have the same timetable every week or maybe for a worker who has a meeting at the same time on the same day every week. I will definitely think about adding this as a potential feature.

“ability to see all my tasks for the month at a glance” – I plan on making the view of the calendar adjustable so you can view a day, a week, a month or a whole year if that’s your preference.

“reminders that you can set minutes, hours, days” – this is an easy feature that I’ll add which can provide a lot of use to the user.

**General Overview of my objectives – 1.3**

* Create a calendar
* Social networking aspect – profiles – groups
* Database for profiles, groups and events
* List all events and groups and have some sort of search for them
* Login and register page
* Be able to edit and save user, group and event information.

**Specific overview of my objectives – 1.4**

Create a calendar

* Have a working calendar
* Shows the current day

Social networking aspect – profiles – groups

* Each user should have an individual profile and each group will have a separate page.
* No username will be the same
* Save the user and group information in SQL database

Database for profiles, groups and events

* I will use docker to hold the database in a container in order to decrease deployment time
* I’ll also use MySQL to code the database
* Ajax calls and node JS will be used to store and display the data

List all events and groups and have some sort of search for them

* Let the user search an exact name for an event or a group and their searched input comes back with results
* Uses ajax calls and node API
* Search results refresh the table at the press of a button

Login and register page

* Allows user to register a new account
* Allows user to log into their account
* Checks to see if the username and password are correct according to database
* Hashes the password for safety

Be able to edit and save profile, group and event information

* Users can change profile, group and event info after creating each of them
* Saving this information makes a change in the database so when the page is loaded up next the new data will be shown
* Users can also delete their account, event or group which removes it from the database

**Prototyping – 1.5**

Initially when starting the project, I had planned to write in HTML, PHP, SQL and a small bit of JavaScript but after a few tries I found PHP to not be the most effective language to use and instead changed to using JavaScript as the main language. This proved to be much more successful and found my code to be laid out better and be able to be understood easier. Prior to this project I had no JavaScript, PHP or HTML knowledge so there was no loss of time swapping from PHP to JavaScript. Having no prior knowledge did mean a lot of my time was spent on learning the languages but this made the project a very useful learning experience and by the end of it I could confidently use JavaScript and HTML.

Making a website meant I had to host the website and I had trouble finding a suitable way of doing this. At the start I was using WAMP which had a useful SQL GUI that allowed me to create the database easily and also had a connection to PHP, so it was useful at the time. But then I swapped computers and started doing my project on a mac which meant WAMP was no longer available, there was a mac version, but it was not user friendly and couldn’t run on my computer.

At the start I just hosted locally on my computer which was fine as the files were on my computer and they weren't going to be moved but when I was implementing cookies I found issues to arise. This meant I had to find another way to host my website. Eventually I found a solution, by running this command in terminal (python -m SimpleHTTPServer 8000) I was able to create a very basic http server, allowing cookies to be created which are vital for my website. This was required to be done in order to make use of cookies are it couldn’t be done when serving the HTML files directly through the browser.

I then used docker to deploy my SQL database to the cloud. Doing this had several advantages, for example it increased the speed to which I could run queries. The docker containers decrease deployment time to seconds which is very impressive, another advantage of using docker was the mobility, they can run anywhere.

|  |  |
| --- | --- |
| <?php  include("../../includes/databaseconnection.php");    $email = $\_POST['emailInput'];  $password = $\_POST['passwordInput'];  $sql = "INSERT INTO `users` (`UserID`, `Username`, `Password`, `Timestamp`) VALUES (NULL, '$email', '$password', CURRENT\_TIMESTAMP)";    if ($conn->query($sql) === TRUE) {  echo "New record created successfully";  } else {  echo "Error: " . $sql . "<br>" . $conn->error;  }    $conn->close();  ?> | This was my original PHP code to create a user.  Even though I ended up not using PHP for my final project this helped me learn about connecting to databases and implementing SQL into my code. |
| <?php  include("../../includes/databaseconnection.php");    $newemail = $\_POST['newemailInput'];  $userID = $\_POST['userIDInput'];  $sql = "UPDATE `users` SET `Username` = '$newemail' WHERE `users`.`UserID` = '$userID'";    if ($conn->query($sql) === TRUE) {  echo "New record updated successfully";  } else {  echo "Error: " . $sql . "<br>" . $conn->error;  }    $conn->close();  ?> | This was the original PHP code to update a user's information.  I did not like how php laid out the code and when I had added the rest of the functions my code would have been extremely hard to navigate. |
| <?php  include("../../includes/databaseconnection.php");    $sql = "SELECT \* FROM `users` WHERE `UserID`=52";    $result = $conn->query($sql);    if ($result->num\_rows > 0) {  // output data of each row  while($row = $result->fetch\_assoc()) {    echo " - Name: " . $row["Username"];  }  } else {  echo "0 results";  }    $conn->close();  ?> | Prototype php code to get the username of the user. In this code the UserID is just a static number so this function will only work for the UserID 52  Later in my code after I'd researched different ways to pass the UserID dynamically I use cookies which is a much more efficient way.  But this was still a useful learning experience. |
| <?php  $servername = "localhost";  $username = "server\_access";  //this is a local database and a strong password would be used when it's remote  $password = "12345";  $dbname = "group\_calendar";    // Create connection  $conn = new mysqli($servername, $username, $password, $dbname);    // Check connection  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }    ?> | This prototype PHP code is for the connection to the database.  Recreating the connection in JavaScript was extremely easy as it was similar to this PHP way. |
| ocument.querySelector('form.form-signin').addEventListener('submit', function (e) {  var emailInput = document.getElementById('inputEmail');  var passwordInput = document.getElementById('inputPassword');  //prevent the normal submission of the form  e.preventDefault();    console.log(emailInput.value);  console.log(passwordInput.value);  $.ajax({  type: "POST",  url: "./api/create\_user.php",  data: { 'emailInput': emailInput.value, 'passwordInput': passwordInput.value },  //dataType: 'json', this breaks everything look into why  success: function () {  console.log("it worked");  },  error: function () {  console.log("it did not work");  },  });  }); | This is a prototype of ajax calls. This was my code at the start of my project when I was getting used to using ajax calls and learning how to use them effectively.  This was very useful as seen in my final code, ajax calls are a vital part of how my whole project functions. |
| function getCookie(cname) {  var name = cname + "=";  var decodedCookie = decodeURIComponent(document.cookie);  var ca = decodedCookie.split(';');  for(var i = 0; i <ca.length; i++) {  var c = ca[i];  while (c.charAt(0) == ' ') {  c = c.substring(1);  }  if (c.indexOf(name) == 0) {  return c.substring(name.length, c.length);  }  }  return "";  } function setCookie(cname, cvalue, exdays) {  var d = new Date();  d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));  var expires = "expires="+ d.toUTCString();  document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  }  $(document).ready(function(){  $("#create\_cookie").click(function(){  setCookie("UserID", 1, 2);  console.log(getCookie("UserID"))  });  });  $(document).ready(function(){  $("#create\_cookie2").click(function(){  setCookie("GroupID", 3, 2);  console.log(getCookie("GroupID"))  });  });  $(document).ready(function(){  $("#see\_cookie").click(function(){  console.log(getCookie("UserID"))  console.log("working")  var x = getCookie()  console.log(x)  });  });  $(document).ready(function(){  $("#delete\_cookie").click(function(){  console.log("this button works")  document.cookie = "UserID=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";  });  }); | When learning about cookies it took me a while to become well versed in using them. In order to practice them I created a test.js and test.html to help me learn.  In the html I created multiple buttons then in the JavaScript I made functions connected to the buttons. I also used console log throughout the code to see how the results came out which I then could see how to handle them properly.  I also practiced deleting cookies, there didn’t seem to be a delete cookie function already so I just simply set the expiry date to sometime in the past so the cookie no longer existed. |