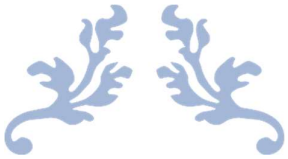


SOFTWARE DEVELOPMENT IN APPS FRAMEWORKS

Independent Learning Module ILM MOD002677

DISTANCE LEARNING UCP SEMESTER 2 02 FEBRUARY- 04TH MAY 2018
Module based on Software Development In Application Frameworks
with logbook

Contents **Page**

Introduction

This is a logbook which, represents information and research carried out to complete the five coursework's in a suitable chosen framework that has been chosen. In this case a web-development framework called ASP.NET has been used to create the coursework's, alongside with C# a programming language with combinations of extensions, libraries, APIs and web services.

This logbook contains designs of the webpage, complex code that has been used via research and working screenshots of coursework's and what went wrong and solutions used to overcome these problems.

All of the work has been uploaded to GitHub and the it can be accessed at the https://github.com/sdaf1718?tab=repositories. This weblink is private and can only be accessed by users like my lecturer who have been given access.

Reflective Commentary why ASP.NET was chosen

ASP.NET is an open source server-side web development application framework. Its' purpose is for the deployment of websites, web services and web applications which, has been created by Microsoft. It is widely used because it consists of various frameworks which can communicate within ASP.NET to create a fully dynamic website for various formats such as, Android, IOS, tablets and many more like Desktop PCs. This is because ASP.NET can work along with HTML, CSS stylesheets and JavaScript in with different libraries to create stunning websites. Whilst it also, has gives the flexibility to work with APIs and Web services which are used in industry like for casting the weather or for navigation from one place to another. This is because these use real-life technologies like Google and web socket services which, can communicate with ASP.Net for this to be allowed.

In todays society and industry ASP.NET is widely used alongside C sharp a programming language which, allows programmers to code and allow functionality to the user to navigate or to buy products on their website with ease. An example, of this could be like when a user logs in to buy a product they only see their own personal data this is because of programming techniques which can work along with ASP.NET that have been used.

There are many alternatives to this web framework like Haskell, PHP, Ruby, PHP and more which, can be used instead as they all similar to ASP.NET however, there are limitations in these frameworks because ASP.NET has more functionality, extensions and libraries which make it unique to use as there is more broader features and services which can be achieved in ASP.NET over the others as it is widely used and constantly been updated with more tools for users to use. ASP.Net also, has a large community base with forums where you can talk to other people to overcome bug fixes and errors. ASP.NET does allow plugins which allow PHP to work with it and all the mentioned above frameworks because in industry it is widely used therefore, it is constantly been updated with new features to keep it bug and error free.

ASP.NET has a feature which allows you to use the MVC model view controller which, is an architectural pattern mostly used to create a user-interface. This allows the division of the application to be split into three tiers. A model which deals with all the extensions, logic and manages the data. A View is the visual representation like a graph or a table. The controller is

the communication device between the model and view to convert the information. For example, if the user clicks on a button it is the job of the controller to respond to the users input and communicate this back between the model and the view. Overall, MVC is accepted by the user as of its natural interface and has now become very popular and is being used with other programming languages and frameworks like Java and Ruby.

My own opinion is ASP.Net is a all in one platform framework

In my own opinion ASP.Net is like an all in one platform which, allows a bundle of frameworks to communicate with its own framework to create easy, dynamic websites whilst using a variety of tools like J-query, Ajax, JavaScript and many more to create a good project. Whilst this gives the user like myself to learn new techniques and leagues and libraries which, can help to utilize the skills in other industrial languages. For Example, JavaScript can be learnt which, is helpful as it can be used in Java programming. This platform is a learning curve which, in my opinion a broad knowledge about different frameworks and programming languages giving new skills to use in other frameworks.

Some benefits of using ASP.NET are as follows and to sum up for this reason I have chosen ASP.NET because it has reduced time, energy and effort when it comes to creating a website.

- With the web development method, the framework technology allows reduction of coding which is helpful in large business where you do not have much time on projects and it can be very expensive.

- As this framework uses native optimisation, early binding caching services and just in time completion which is an advantage over other frameworks.

- ASP.Net is much securer than other frameworks due to its built-in windows authentication which, is another advantage if you feel at risk of security of personal data.

- Development has become easier with ASP.Net because there is no need to register components instead the web configuration is already built-in. As a result, it means less work.

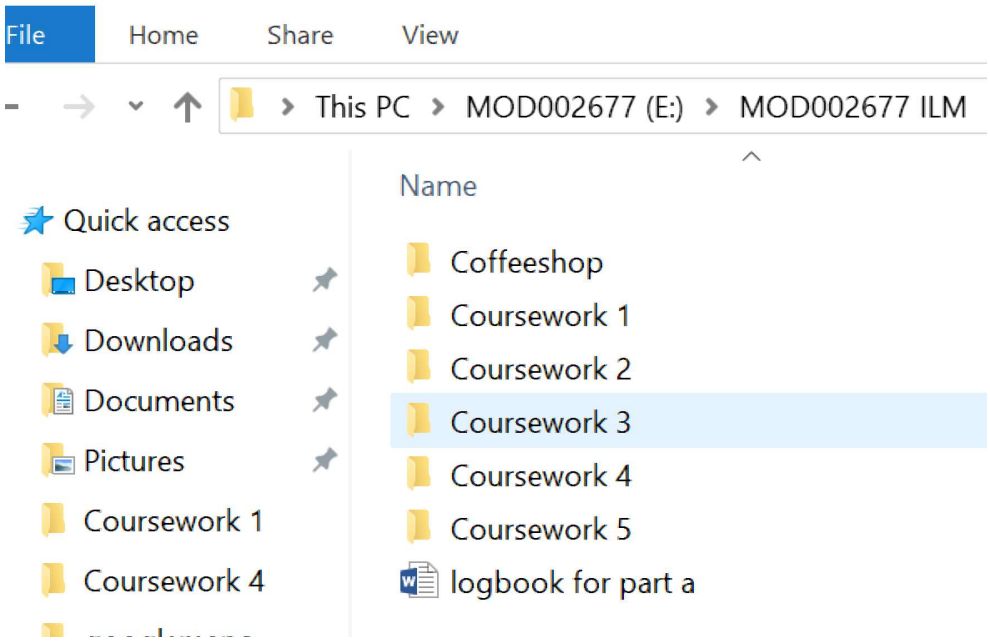
- Server-side coding which, helps as it is faster as the code is executed before it is sent to the browser for the user to test or see.

- Most important of all request monitor processing is always present in case a process dies or crashes a new one will be created. This is good as you do not waste time or have to restart the program or go back to code to re-write the process.

02/02/18 Setting up GitHub

The course module overview and criteria were outlined. The coursework criteria had part a and part b and each week there would be a worksheet to go through and then show to my lecturer. As this was a distance-learning intendent learning module I could work at my own pace and a lot of research would be ideal.

First of all, as this was a new module, best practices I have learnt is to set up a new folder and file structure as shown below on my USB which, will be handed in with part a and b projects which will show the work carried out by myself. The module will be an independent learning module code MOD002677. For this reason, I called my USB this and had separate folders for each coursework as there will be five coursework's for part a and a coffee shop assignment for part b.

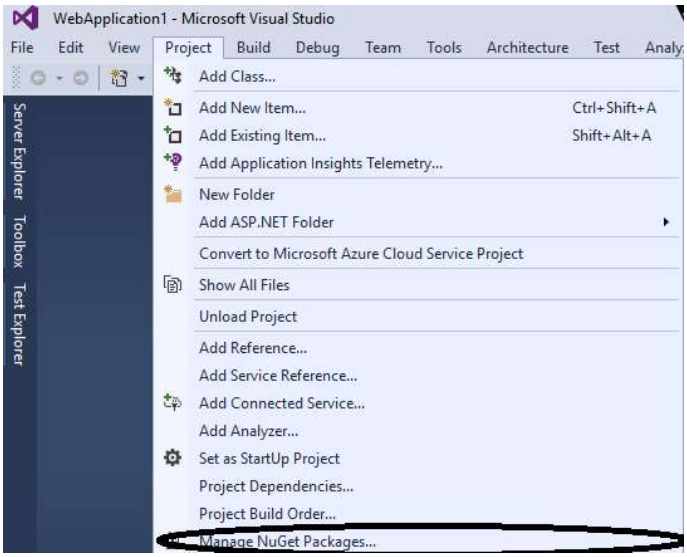


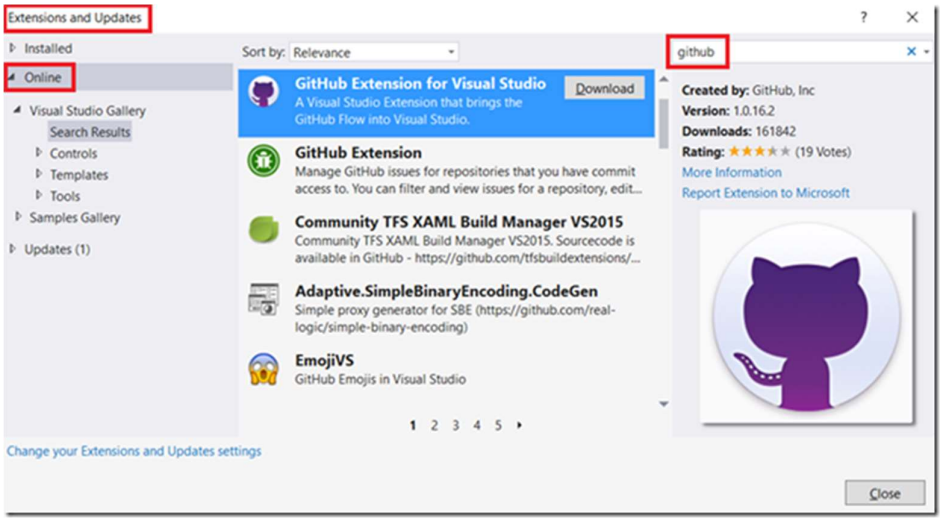
After reading he module guide and passements I found a worksheet that it would be ideal to set-up GitHub to share your work. After reading the worksheet and following its guidelines it appeared to be old and not for the current set-up for GitHub for Visual Studios 2015. Research was carried out to which, showed how to set up GitHub for VS[visual-studios]. By following this website, I was able to set up GitHub in Visual Studios 2015 Professional edition on windows.

Source taken from (Step by Step working with GitHub Repository and Visual Studio 2015 | Infragistics Blog, 2018).

GitHub Setup

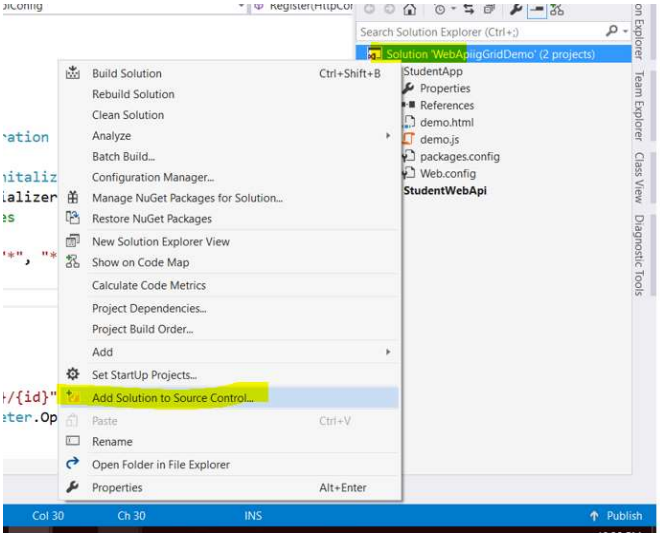
Step 1 was one was to open Visual studios and head to packet manager NuGet which, is a tool where extensions, libraries and updates can be downloaded which are computable to be used with Visual Studio. To enable GitHub, use the driver had to be downloaded the plugin so it could communicate with the website to allow communication to save and import work. To access NuGet click on view then manage NuGet packages. After this you can type on search bar the plugin you require for example, GitHub extension and a list will appear we select the first one and click download to install it.

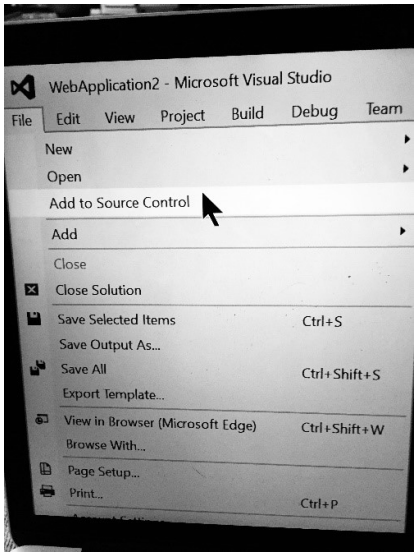




Step 2

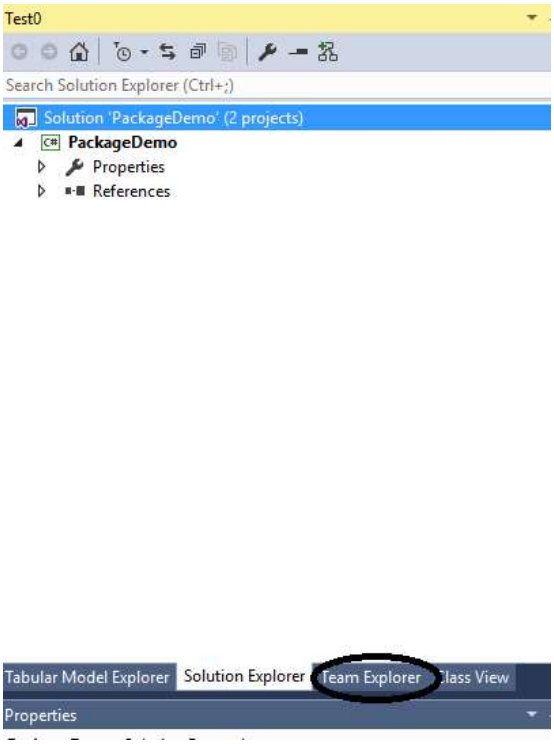
To add GitHub as it is now installed you can either click on file add to source control or right hand click on solution explorer of your project and add to source. My own preference is just click on file menu add to source control as I find this easier using a Mac instead of clicking the two-button finger mouse on the right-hand corner.





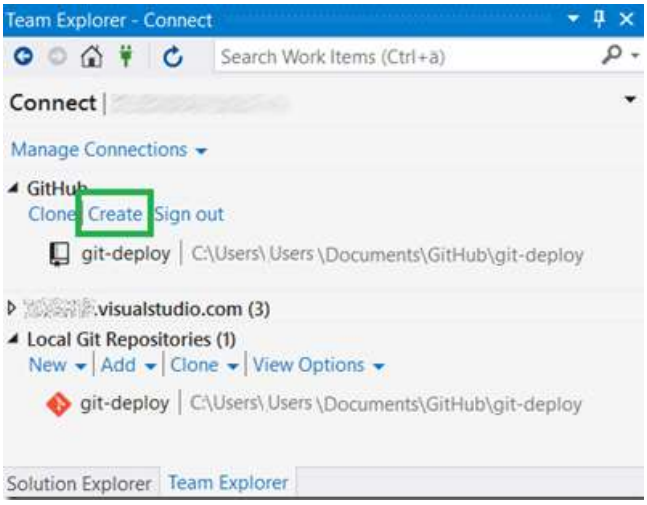
Step 3

In solution explorer at the bottom click on team explorer. Here you can create an account or login with your credentials to GitHub. As I have previously used GitHub I will enter my credentials and click on login.

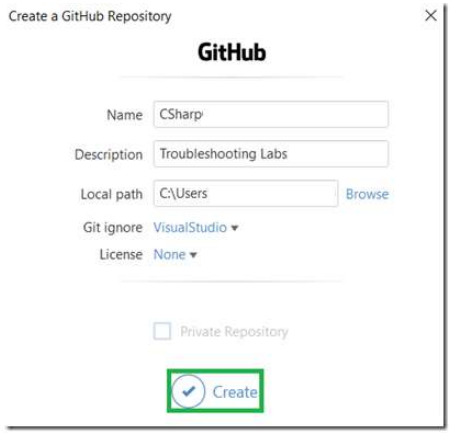


Step 4

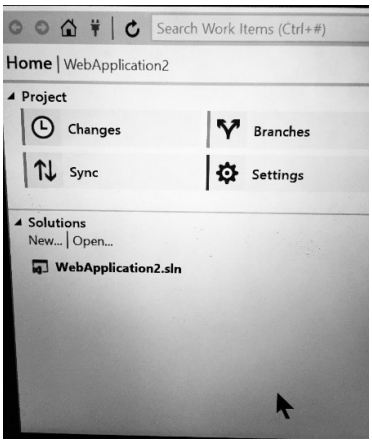
You will be asked to create a new repository which is like a new folder to keep your work in if you don't have any or you can click on the sync or change button to add work to an existing project created before. Here we will create a new repository



We can then enter a few details for example, the name of the repository, description and where it stored or you like to store it and then click create.



Then we finished and we can upload our project by clicking sync or we can modify it by clicking change.



Coursework 1

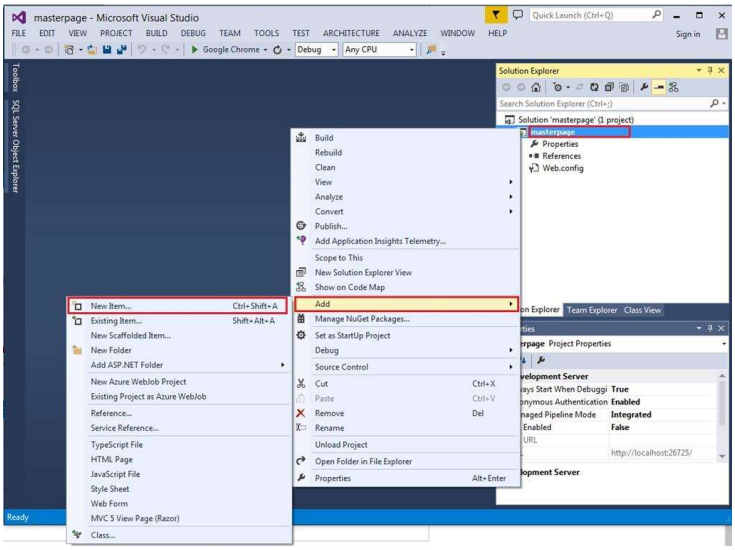
03/02/18 Coursework 1

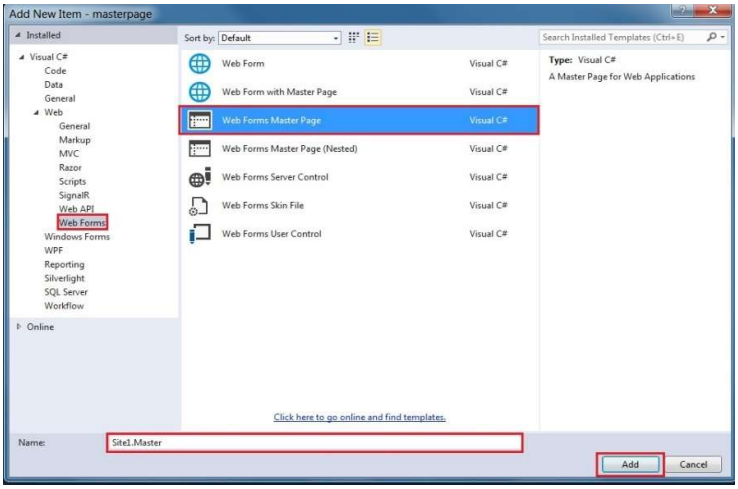
Looking at coursework 1 the first task was using my skills of what knowledge I had learnt to make a website using ASP.NET which had a consistent layout. It should allow users to login.

As used ASP.NET before it was clear to create a website using HTML and CSS mark-up language which, had been used by me in previous module and in my final project. Research, demonstrated that by designing a master-page it could be used on all pages which, would save time to re-design each page. This was a new learning curve for me. By following the link and making changes to my previous CSS that I had used in my coursework I started to create a master-page. The step-by step in depth guide helped me to understand what a master-page was and what it did. The link is (How To Create Master Page In ASP.NET, 2018). A basic website has a home page, about page and contact page which I created in my website.

Step 1

Right hand clicks on your project and click on add, new item and then select web form master page. Rename it if you like and click ok.





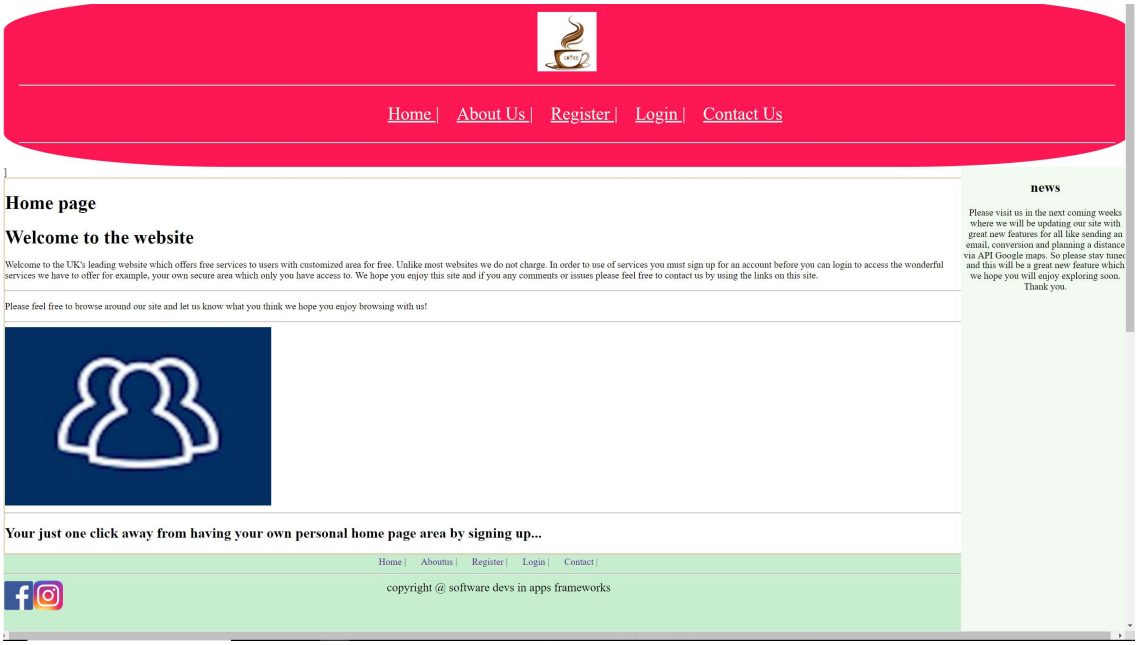
The web form will load with some HTML tags and here you can write your own HTML which I did. As what I had learnt from Internet Databases I used my previous knowledge to create a web page. In the same way you can add a CSS stylesheet page and to use the master-page you have to select from the list web form with master page which is important otherwise you will not be able to use the master-page created. The overview of this was just like creating a web page in HTML using a text-editor with all the elements of HTML like divs, titles and much more.

My home page which had to be called default.aspx as I renamed it to home but it kept crashing and after asking my lecturer and showing it to them they told me that the default.aspx file was missing. This is like in HTML where you need an index.html for it to work.

In my CSS stylesheet at the top I put in the following code to ensure when the user loads the website it will always appear to 100% for all medias as many platforms and devices have different resolutions scales.

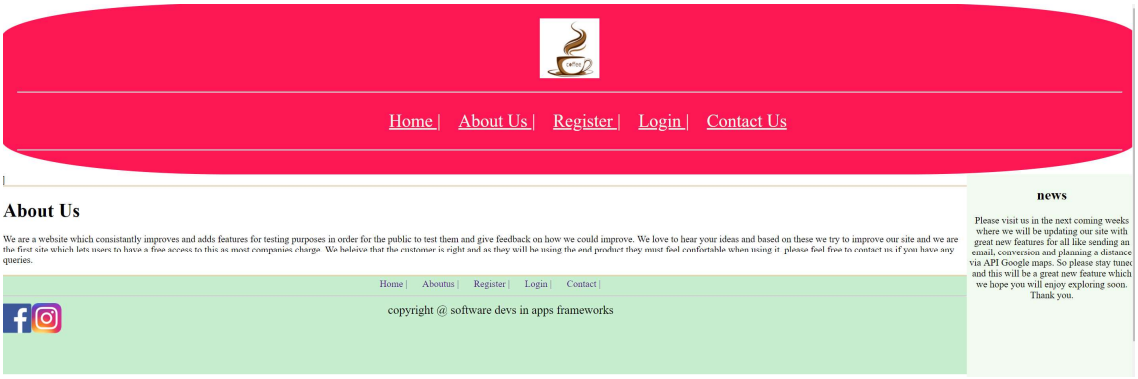
```
1  body{width:100%;}
2  #nav{
3        margin-bottom: 0;
4        border-radius: 0;
5      background-color:#FF1654;
6      text-align: center;
7      font-size: 30px;
8      padding: 25px;
```

My home page

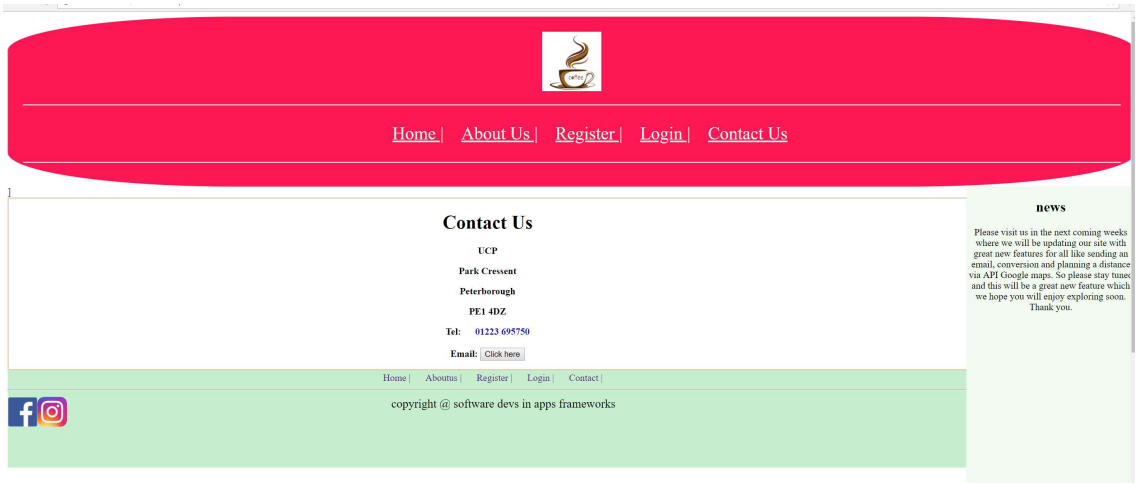


The choice for the menu was taken from W3c schools help and guidance. The CSS choice I adapted from my old projects created. I felt that as it was for a coffee shop the menu bar could be in shape of a circle however, I could not get it exactly how I wanted to but felt it looked nice so I used it. The logo was created by me in paint and photoshop.

About page



Contact page

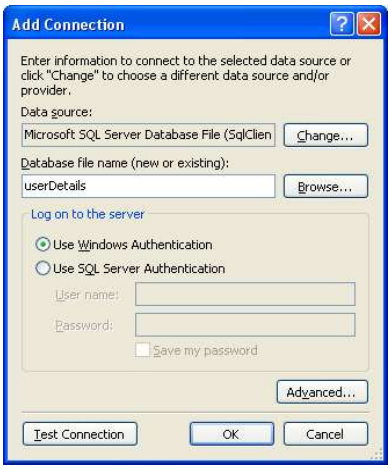


Creating a database connection for the registration and login page to store details

The next step was to create a registration page and a login page to allow users to login to the website. However, before creating this I felt it was necessary to create a database to store the details which is one of the coursework's criteria for 7-10 marks.

I had used Microsoft management studios before and workbench to create databases. However, I had never used the built-in database tool. This was a new learning curve for me and I decided to use it and connect to it by using the MYSQL data client library. This was done as follows:

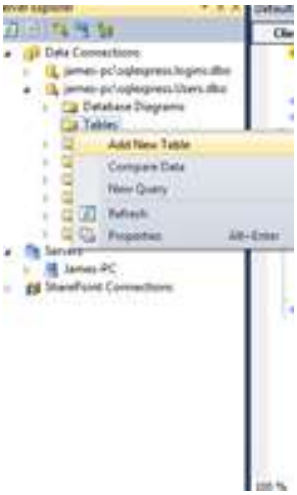
1. To click on tools and connect to database.



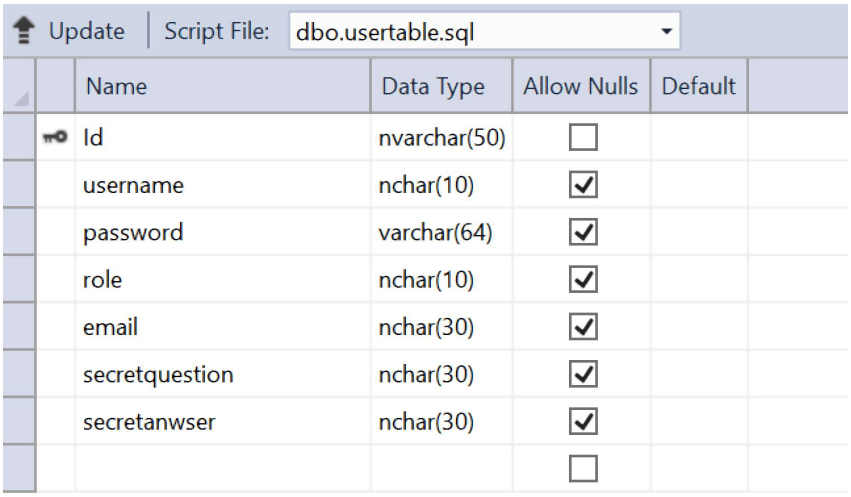
Above I can enter a database new and browse where to save it or I can browse to open a existing database. As this is a new database I cannot test its connection as it will not exist and will fail. Instead I click ok and a message will prompt me if I like to create this new database. We will click yes and the message will also, inform us where it will be saved.

The database will be created and we can view it in server explorer which will be on the left-hand side. If you do not see this please click on view server explorer and it will appear.

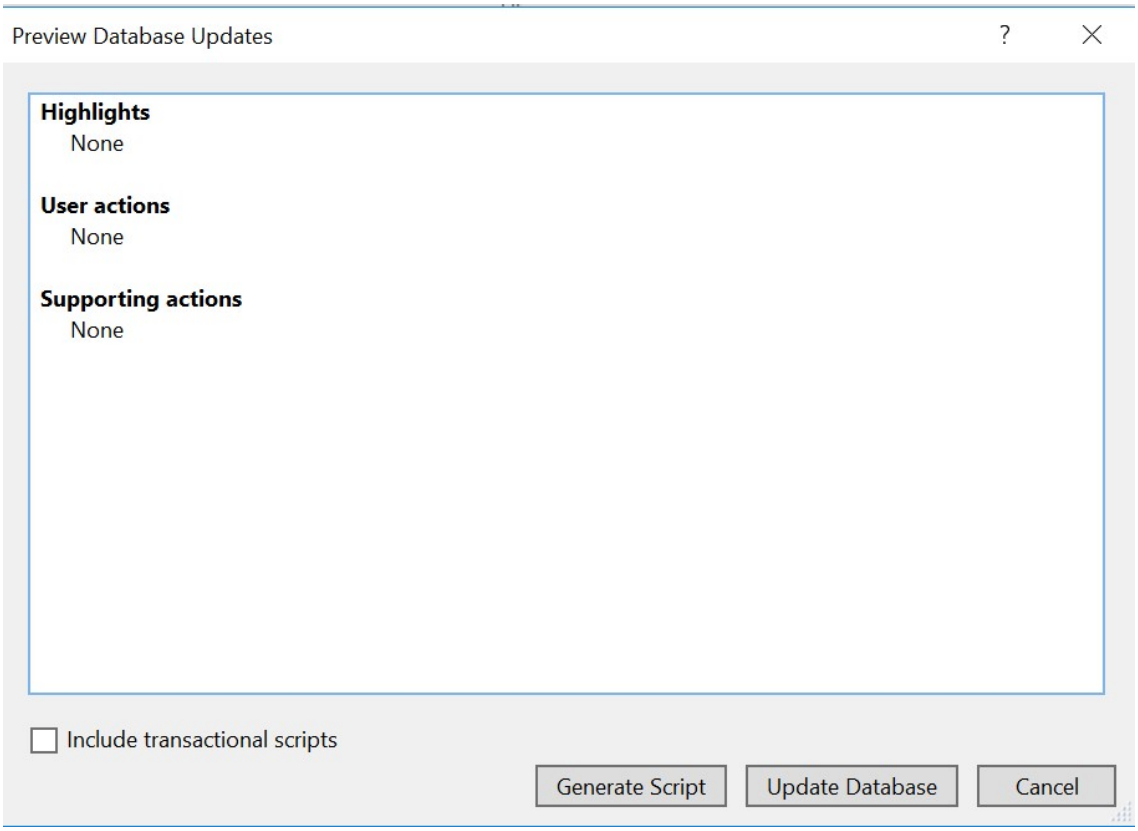
We can now click on the database and click create new table.



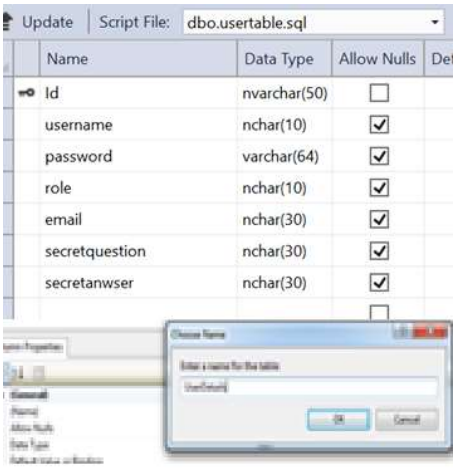
The table will open in design view just like in Microsoft Access and workbench and we can select our data-types and names for our fields. Before we can save it we must click update then save to save the table otherwise it will not be save correctly. Below is my database table design which, I have used.



Click update and it will ask to generate the script and then you can click save afterwards to save the table.



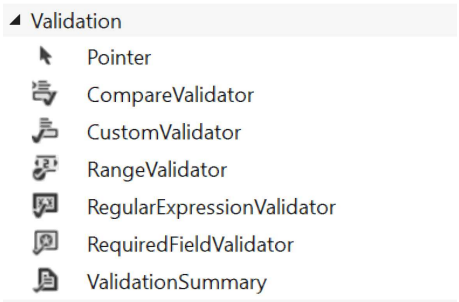
Save the table by giving it a sensible name like usertable not **user** as this is a reserved word in C# and in ASP.NET.



Now we can create our register page as we have created a database which, we can use later to store details of users in so they can login to the website. it is advised not to have data-types with too many characters like username I have left mine at 10 because I not come across a person with a name longer than 10 characters. This helps to improve memory and database storage optimisation.

05/02/18 Registration page

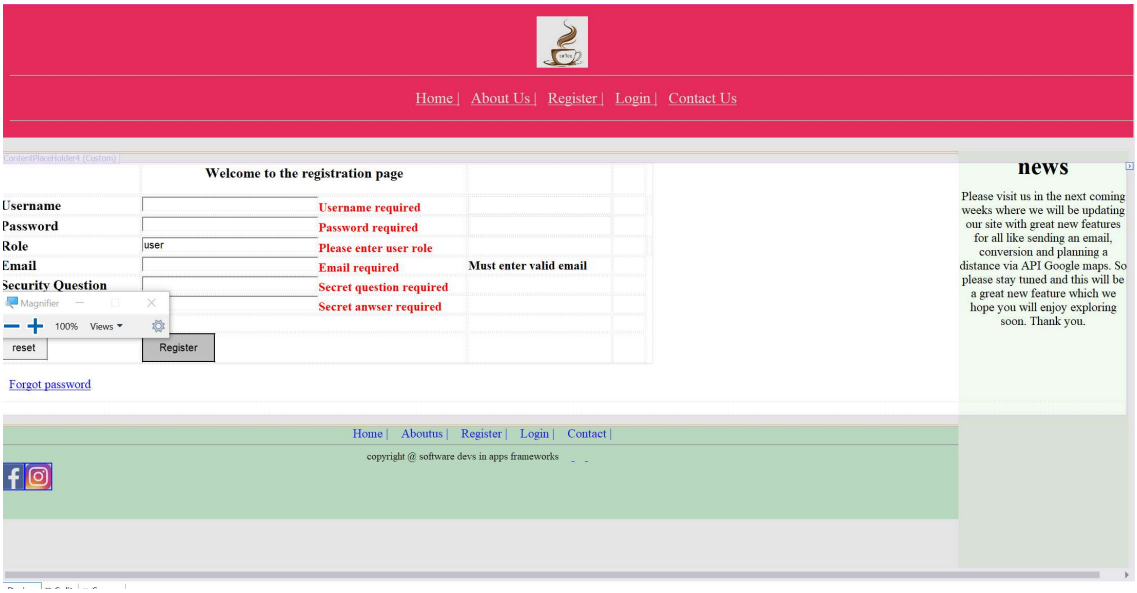
As I had created many registration pages with login pages it was hard to decide which path to take to create one as there are many ways some simple ways and some use complex functionality. Research, showed on YouTube a set of videos how to create a registration page with unique identifiers and login page with validation and connecting to the database. These videos I followed a link (ASP.NET Tutorial 6- Create a Login website - Login page & Validating User and Password in database, n.d.). From this, I learnt from these videos was that ASP.NET had validation controls which could be used from the toolbox under validation. These could be used to ensure a user does not leave a blank field which could corrupt the database created or create errors.



Also, what was learnt was GUiD which stands for Global Unique Identifier which is a built-in feature which generates a unique ID. It consists of 128-bit integer to identify resources normally used by developers like Microsoft.

By following the videos, I created a registration page first with the SQL code which can be seen on the USB which has been given with this logbook as the code that has been used is very long but has been commented to show what has been learnt and as most of it has come from previous knowledge that has been learnt from various modules under-taken.

Most of the controls that have been used have been taken from the toolbox such as, labels, validation controls and the textboxes. All of the creation has been done in design view.



As I had been following the videos very carefully the only error I got was when I tried entering data. This was because I tried to enter too much information into a field which, I had restricted to a smaller length, otherwise, there was no other errors that I had to overcome. The error is shown below.

```
error:System.Data.SqlClient.SqlException (0x80131904): String or binary data would be truncated. The statement has been terminated. at
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) at
System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction) at
System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean
asyncClose) at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream,
BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady) at
System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString, Boolean
isInternal, Boolean forDescribeParameterEncryption) at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior
cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task& task, Boolean asyncWrite, Boolean inRetry,
SqlDataReader ds, Boolean describeParameterEncryptionRequest) at
System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream,
String method, TaskCompletionSource`1 completion, Int32 timeout, Task& task, Boolean& usedCache, Boolean asyncWrite, Boolean inRetry) at
System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(TaskCompletionSource`1 completion, String methodName, Boolean sendToPipe,
Int32 timeout, Boolean& usedCache, Boolean asyncWrite, Boolean inRetry) at System.Data.SqlClient.SqlCommand.ExecuteNonQuery() at
ex1.Register.btnregister_Click(Object sender, EventArgs e) in E:\MOD002653 software devs\coursework1\coursework1-master\coursework1-
master\ex1\Register.aspx.cs:line 88 ClientConnectionId:30357885-467f-44b6-968d-db504234faef Error Number:8152,State:13,Class:16
```

From the video as well, I learnt to check the database if the user exists to prevent two users having the same name. This is important if the username is the primary key but in this case my ID is. This is important because it can cause clashes when trying to find information or when matching information, the following code was used.

```
public partial class Register : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if(IsPostBack)
            {
                // used set of tutuiorals with code for guidence to help with creating register,login, redirect pages for user and admin with
sessions
                // webiste is youtube video https://www.youtube.com/watch?v=KV1Xccl-XBA&list=PLS1QulWo1RIaM8-S7kTHgWd_pGNu-CyQS
                // below is a way built in using configuration manager to access the database via a connection string instead of usign a data
source which is better to use in some instances if accessing a external  database

                SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["logindetailsConnectionString"].ConnectionString);
                conn.Open(); // opens connection to database

                // string used to select from the table called user tables
                string checkuser = "select count(*) from usertable where username = '" + txtusername.Text + "'";

                // checking to see if the user already exisits in the database if so inform user to avoid duplicates

                SqlCommand com = new SqlCommand(checkuser, conn);
                int temp = Convert.ToInt32(com.ExecuteScalar().ToString());
                if (temp == 1)
                {
                    Response.Write("user already taken please chose another");
                }

                conn.Close();


            }

        }
```

06/02/18 register page and login page with scripts

After looking back at my work, I realized that I had used encryption in my database which, I had not here. Previously I had used BCRPT and MD5 however, these have recently been cracked.

Researching I found a link which used Sha 256 which interested me and by following several tutorials and guides I was able to create a script to encrypt my password using this and the Cryptography library which was one of the tasks in the lectures worksheets provided. I was pleased as it was something new that I had learnt and achieved to create. After reading several

guides I could have used salt and hashing which can be cracked but with Sha256 it is hard to crack and the user' password is not directly stored anywhere in the database so if the database is hacked the user information can be secure as the hacker will not be able to get their password. The main link used was (Hashing with SHA1 Algorithm in C#, n.d.).

For this to work I had to create the database as a class script library to declare the database. The disadvantage of sha256 was that the password hashed the user could not retrieve their own password instead a new one would have had to be either generated or given. This is one of the coursework criteria instead for this I have generated a new random password via ASCII using sha256 as if the user has forgotten their password they will not know it and best practice in this case would be is to give them a random password. Another way would be to use salt but again if the salt was stored in the database along with the hash it can be cracked and therefore, is not secure as what I have implemented. Below is the sha256 code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Security.Cryptography;
using System.Text;

namespace ex1.scripts
{
    public class PasswordHasher
    {
        // got this code from my old university lecturer Dr Ian sexton and adapted it.
        // also followed some videos for help on sha256 passwords https://www.youtube.com/watch?
v=oJpZ5ygg4qQ
        // also used this link as well for help https://www.youtube.com/watch?v=0dgTf9TUDHU
        // guidence from https://codeshare.co.uk/blog/sha-256-and-sha-512-hash-examples/


        public static string GetPasswordHash(string password)
        {
            SHA256 sha256 = SHA256Managed.Create(); // create the sha256 password you can have 512 and 128
but 256 is strong enough
            byte[] hash = sha256.ComputeHash(Encoding.ASCII.GetBytes(password)); // it encodes the hash
            string pwdHash = "";
            foreach (byte b in hash)
                pwdHash += b.ToString("X2");
            return pwdHash; // returns the paswwrod in a hash value
        }
    }
}
```

Part of the database library code for the sha256 to work

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.SqlClient;
using System.Configuration;

namespace ex1.scripts
{
	public class Database
	{
		public class UserInfo
		{
			public string Id;
			public string UserName;
			public string Password;
			public string Role;
			public string Email;
			public string SecretQuestion;
			public string SecretAnswer;
		}

		public static UserInfo GetUser(string name, string password)
		{
			SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["logindetailsConnectionString"].ConnectionString);
			conn.Open();


			string checkuser = "select * from usertable where username = '" + name + "'";
			SqlCommand com = new SqlCommand(checkuser, conn);

			UserInfo ui = null;
			SqlDataReader reader = com.ExecuteReader();
			if (reader.Read())
			{
				// 0 = id
				// 1 = username
				// 2 = password
				// 3 = role
				// 4 = email
				// 5 = secretquestion
				// 6 = secretanswer
				string pwd = PasswordHasher.GetPasswordHash(password);
				if (pwd == (string)reader[2])
				{
					ui = new UserInfo()
					{
						Id = ((string)reader[0]).Trim(),
						UserName = ((string)reader[1]).Trim(),
						Password = (string)reader[2],
						Role = ((string)reader[3]).Trim(),
						Email = ((string)reader[4]).Trim(),
						SecretQuestion = ((string)reader[5]).Trim(),
						SecretAnswer = ((string)reader[6]).Trim()
					};
				}
			}
			reader.Close();
			conn.Close();
			return ui;

		}

		public static void ChangePassword(UserInfo ui, string newPassword)
		{
			SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["logindetailsConnectionString"].ConnectionString);
```

Result if the user registers we can see the password is encrypted

Id	username	password	role	email	secretqu...	secretan...
68ef46bc72	linda	C179BBF0E38DD100C19A6F92F3C98F7C6EC9EE7CE84E3A6A0AD688B75A97A0A3	user	linda@lin...	linda ...	michell ...
14b7246...	pete	6CB912DE5B6EF87CB3ECF92C13052D80D02ED91771E6E640625D759CD444E941	user	pete@pe...	pet ...	pet ...

07/02/18 login page and change password

Coursework task was to create so users could login in and to Consider OO techniques with customer and manager logins. In my project users and managers can login via sessions and roles as this has been used to separate the logins using the same login page. Users have a different access page to the manager.

Another way of considering this was to create a store procedure and a sperate class for manager and user via this way but would be long way to do write it. This would still involve some kind of roles and sessions or cookies to exist to check who was logging in.

The best way I have implemented it as it saves time is to use user roles. The username is compared to the user role in the database and if the username registered is a user it will be re-directed as on its credentials and a session will be created unless the user logs out and it is destroyed. Code used for this is below whilst using the scripts I created for the password hash to compare the information.

```
ex1                                   ex1.login                                   Page_Load(object sender, EventArgs e)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;
using ex1.scripts;

namespace ex1
{
    1 reference
    public partial class login : System.Web.UI.Page
    {
        // helpufl notes taken from my Final project module created last year
        // using scripts for the hashed password abvoe
        // roles can be used for multiple logins for diferent users who have set permissions like a user, paid user or manager giving them access to certain permissions


        0 references
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        0 references
        protected void Button1_Click(object sender, EventArgs e)
        {
            Database.UserInfo ui = Database.GetUser(txtuser.Text, txtpassword.Text);
            if (ui != null)
            {
                Session["TheUser"] = ui; // session used to check if user logged or manager page logged in
                string redirect = ui.Role == "manager" ? "~/adminpage.aspx" : "~/userpage.aspx"; // redirects to the followign page via roles
                Response.Redirect(redirect, false);
                HttpContext.Current.ApplicationInstance.CompleteRequest();
            }
            else
            {
                Response.Write("Username or Password is incorrect"); // else if information correct display message
            }
        }
    }
}
```

Change password

The change password creation was used from previous knowledge and I had done several of these in previous modules and applying this knowledge I created this.

I also, felt it was necessary and best practices to create a session on form load so that no external user could get to this page via entering it in the URL and trying to change someone password without them knowing it. This made my page secure as without having an account and logging

into the website the password could not be changed. I used the scripts created before to compare the old password with the password hasher to change the password and to encrypt the new password for security. Code below.

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;
using System.Xml.Linq;
using ex1.scripts;

namespace ex1
{
    public partial class changepassword : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

                if (Session["TheUser"] != null) // checks to see if user is logged in if so display welcome message
                {
                    lblwelcome.Text += Session["TheUser"].ToString();
                }
                else
                {
                    Response.Redirect("login.aspx"); // else it redirects to the login page informing user not logged in

                }

            }



        protected void btnupdate_Click(object sender, EventArgs e)
        {
            // gets the user info and the password hash which is stored in the database to match to change
            Database.UserInfo ui = (Database.UserInfo)Session["TheUser"];
            if (ui != null)
            {
                if (ui.Password == PasswordHasher.GetPasswordHash(txtold.Text))
                {
                    Database.ChangePassword(ui, PasswordHasher.GetPasswordHash(txtnew.Text));
                    Response.Write("Password Changed!");
                    Response.Redirect("userpage.aspx");
                }
                else
                {
                    Response.Write("Old password doesn't match"); // message displayed to inform user that password is incorrect
                }
            }
```

<u>08/02/18 Add a way for users to retrieve their password if they have forgotten it.</u>

The best way for me to do this was I felt was to send an email out to the user sending their password out to them. This was because if any user decided to retrieve someone's password they could if a feature in the site existed where they could enter a few details and get the users password. Instead a safe way for verification was to send it via email as if the user forgets their password they can retrieve it by their own personal email as it unlikely another use would have access to someone else's email. Guidance, was taken from my lecturer notes which showed how to send an email to someone.

As I had encrypted the password there was no way of finding it the best way was if the user had forgotten their password it would be easier to generate a new one using ASCII method informing

them to change it next time they login. This was done via guidance from (Generate random password in C#, 2018).

This showed how to allow certain characters using ASCII and then I could randomly generate a password and encrypt it and send the encrypted password to the user via email. This is a safe way I felt because it uses two step verifications because anyone could fill in some details and pretend to be yourself but by entering your email address and then they would have to login to your email address to get the password which is unlikely unless you have given it to them.

The only error I cannot fix is when the email is sent it will end up in the user's junk folder. I feel this is due to the spam filters of the email servers like Google and Hotmail however, once if marked as not junk it will always appear in the inbox. Code used.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.Net.Mail;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using ex1.scripts;



namespace ex1
{
   public partial class forgotpassword : System.Web.UI.Page
   {
      // guidence from https://madskristensen.net/blog/generate-random-password-in-c/

      protected void Page_Load(object sender, EventArgs e)
```

```
    {
    }

    public string GenPwd()
    {
        string Passlen = "10"; // declares password length to ten
        string NewPwd = "";

        // allowed characters for password as it has been hashed we will generate a random password for the user
        // we will allow numbers, uppercase letters, lowercase letters and certain speical keys like a comma and speach marks
        string allowedChars = "";
        allowedChars = "1,2,3,4,5,6,7,8,9,0";
        allowedChars += "A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,";
        allowedChars += "a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,";

        // below we using ascii and the random function to generate a new password

        char[] sep = { ',' };
        string[] arr = allowedChars.Split(sep);


        string IDString = "";
        string temp = "";

        Random rand = new Random();

        for (int i = 0; i < Convert.ToInt32(Passlen); i++)
        {
            temp = arr[rand.Next(1, arr.Length)];
            IDString += temp;
            NewPwd = IDString;
```

```
            }
            return NewPwd;
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string strNewPassword = GenPwd().ToString();
            SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["logindetailsConnectionString"].Co
nnectionString); // connection string to connect to database
            conn.Open(); // open connection to database

            string str = string.Format("update usertable set Password='{0}' where
Username='{1}';", PasswordHasher.GetPasswordHash(strNewPassword), TextBox1.Text);
            SqlCommand com = new SqlCommand(str, conn);
            com.ExecuteNonQuery();
            conn.Close();

            // to send the random password in email
            // following taken from lecturer worksheet

            // smtp is used with creditentials to send an email

            MailMessage msg = new MailMessage();
            NetworkCredential cred = new NetworkCredential("SDIAF1415@gmail.com",
"Software1415");
            msg.From = new MailAddress("SDIAF1415@gmail.com");
            msg.To.Add(TextBox1.Text);
            msg.Subject = "Recover your Password";
            msg.Body = "Your password is: " + strNewPassword;
            msg.IsBodyHtml = true;
            SmtpClient smpt = new SmtpClient("smtp.gmail.com");
```

```
            smpt.UseDefaultCredentials = false;
            smpt.EnableSsl = true;
            smpt.Credentials = cred;
            smpt.Port = 587;

            smpt.Send(msg);
            Response.Write("you should receive an email shortly with your details");
            Response.Redirect("~/Login.aspx");
        }
    }
}
```

09/02/18 review date

At 2:30 I contacted my lecturer via skype to show him what I had done and he was pleased with the work shown as I had not much to show at this present moment in time. This was because I had errors and after speaking with my lecturer it was sorted. This was because I had renamed the default.aspx to home.aspx and therefore, the project would not run. After this fixture I was able to complete the project to show. There are better ways than what I have implemented to create this project nonetheless I am proud as I am have learnt new knowledge which, can be used in other industrial areas and programming languages.

Coursework 2

10/02/18 Coursework 2 Unit testing

This coursework concentrates on creating unit testing by creating suites, test classes and code. A unit testing is software development process which, are called units individually and independently studied. Tests can be done manually but normally are automatically operated (What is unit testing? - Definition from WhatIs.com, 2018).

Previously, I have created tests for my national diploma. I had used NUnit testing which, is open source framework designed for writing and testing for Visual Studio. The software design is paradigm known as Extreme Programming.

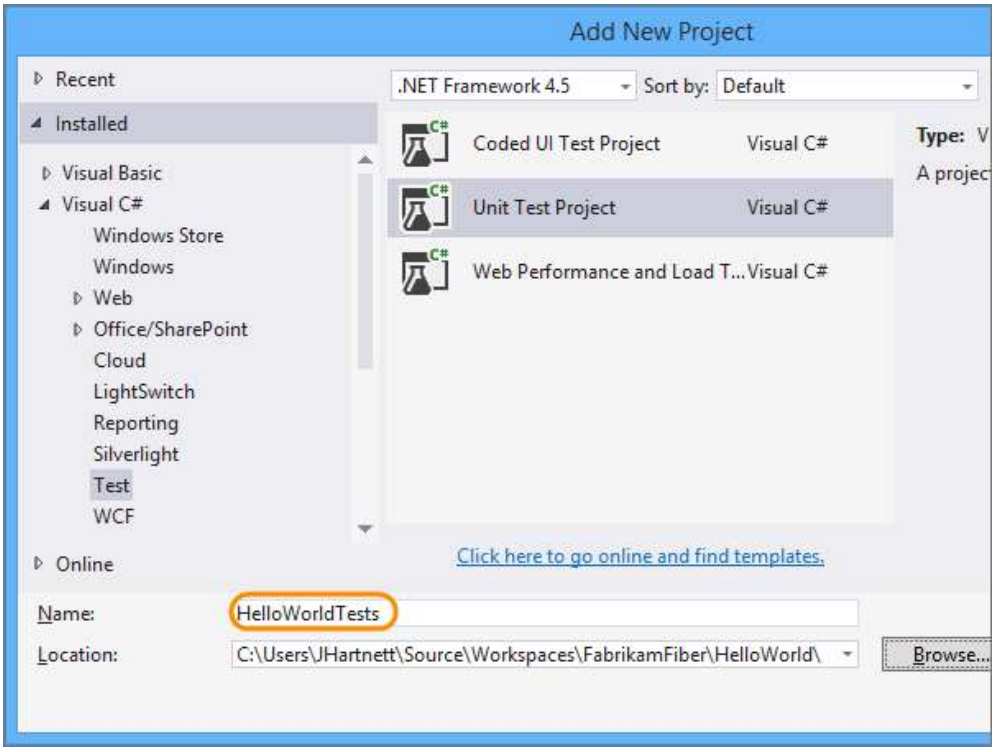
Problem 1

Problem one was to find the sum of all the multiples of 3 or 5 below a 1000. As if we list all the natural numbers below 10 that are multiples of 3 or 5 we get 23 from 3,5,6 and 9.

In my national diploma I had done a similar program to this and therefore, by looking back at my work I found a link that I had used. This help me to create the programs and to answer the problems. This is because the problems were in the research link used and I adapted them and created tests for them.

I researched and found out that Microsoft Test explorer which is part of Visual Studio and as I had never used it and this was a new learning curve for me which, I decided to use. To use this you have to declare the library using Microsoft.VisualStudio. TestTools.UnitTesting;.

First of all, we open Visual Studio and click new project and from the list we select unit test project.



Then we click ok and the screen is loaded where we can write our test.

HelloWorldTests.HomeControllerIndexTests

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web;
using System.Web.Mvc;
using HelloWorld;
using HelloWorld.Controllers;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace HelloWorldTests
{
    [TestClass]
    0 references
    public class HomeControllerIndexTests
    {
        [TestMethod]
        0 references
        public void HomeIndexTests()
        {
            // Arrange
            HomeController controller = new HomeController();

            // Act
            ViewResult result = controller.Index() as ViewResult;

            // Assert
            Assert.AreEqual("Hello, World!", result.ViewBag.Message);
        }
    }
}
```

- Arrange is section of the unit test which initialises the object and sets the value of the data which, is then passed to the method under test.
- Act is which, raises the method with the correct parameters.
- Assert that the method behaves which, is the outcome as expected.

Problem 1 code and outcome

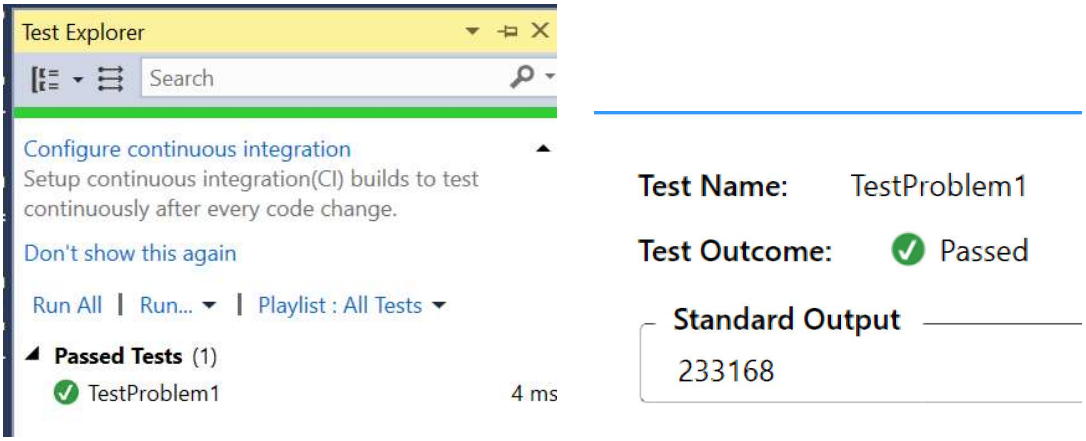
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace problem1 // problem 1 to list all the mutliples of 3 and 5 below 1000
{

   public class multiplesof3and5
   {


      static void Main(string[] args)
      {
         int res = Solution(1000);
         Console.WriteLine(res);



      }

      // guidence for code from
https://sites.google.com/site/eulerproblemsincsharp/home/problem-5
      // If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and
9. The sum of these multiples is 23.
```

```
// Find the sum of all the multiples of 3 or 5 below 1000.
public static int DoProb1()
{
   int sum = 8; // 3 + 5 // we start with the two numbers and add them together as these are the first two numbers we want the multiples from

   // repeat from the number immediately after our largest if (5) -> 1000
   for (int i = 6; i < 1000; i++)
   {
      // Check to see if this number is divisible by 3 and 5 without any remainders
      if ((i % 3) == 0 ||
         (i % 5) == 0)
         sum += i;   // number is evenly divisible by 3 and 5, so add in our sum
   }
   Console.WriteLine(sum);
   return sum; // Give the sum back
}


public static int Solution(int lim) // lim is for the limit
{
   return sum(lim, 3) + sum(lim, 5) - sum(lim, 15);
}


static int sum(int lim, int inc) // inc for increment
{
   int n = (lim - 1) / inc;
   return inc * n * (n + 1) / 2;

}
[TestClass]
public class Problem1Test
```

```
        {
            public void StmtEx()
            {
                int outcomeres = multiplesof3and5.Solution(10);
                Assert.AreEqual(23, outcomeres);
            }

            [TestMethod]
            public void TestProblem1()
            {
                // The sum of all the multiples of 3 or 5 below 1000 which is 233,168
                Assert.AreEqual(233168, DoProb1());
            }

        }
    }
}
```

Outcome the project must be build and then to test the project

Click test on the menu then click windows then text explorer to run all the tests that have been built.



12/02/18 Unit test 2 for Coursework 2

The second problem was what is the smallest positive number which, is evenly divisible by all of the numbers from 1 to 20. This is 2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without having a mod.

Research was done again and the same link was used to help me as it had some answers which I could use. As I am good at maths I knew that everything is divisible by 1 and to write the code in this way and with the help of the link I was able to write the code and then a test to show this.

Code used

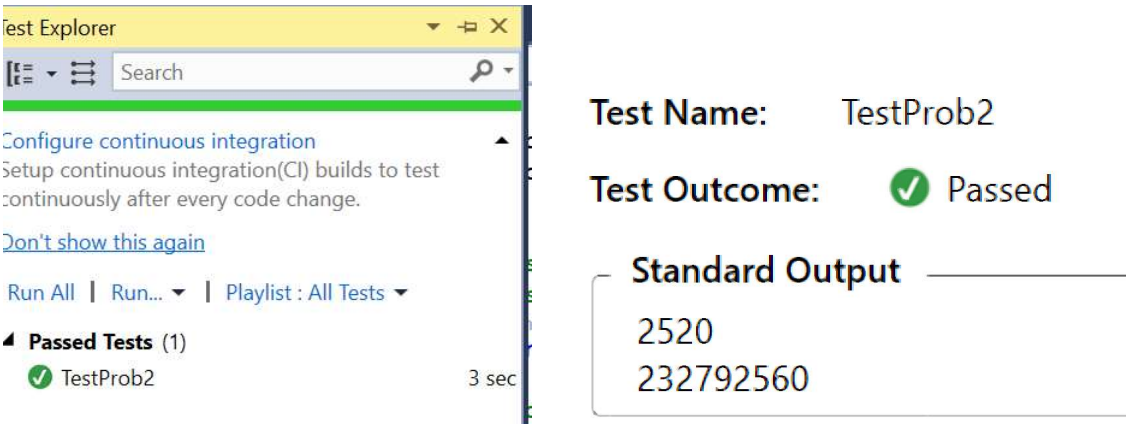
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace p2
{
   // guidence from https://sites.google.com/site/eulerproblemsincsharp/home/problem-5
   public class Class1
   {
      [TestClass]
      public class UnitTest1
      {
         private long Prob2(long n)
         {
            if (n < 2)
               throw new ArgumentException("n must be greater than 1");

            bool got;
            long res;
            for (int i = 1; i < Int32.MaxValue; i++)
            {
```

```
            // First Improvement
            res = n * i;
            got = true;

            // Second Improvement
            for (var si = n / 2; si <= n; ++si)
            {
               if (res % si != 0)
               {
                  got = false;
                  break;
               }
            }
            if (got)
               return res;
         }
         return -1;
      }

      public object Prob2_2()
      {
         var Outcome = Prob2(2);
         Assert.AreEqual(9009, Outcome);
         return Outcome;
      }

      // 2520 is the smallest number that can be divided by each of the numbers from 1 to 10
without any remainder.
      // What is the smallest positive number that is evenly divisible by all of the numbers
from 1 to 20? and 1 can go into everything and 2 only even prime number
      public static int prob2(int higher)
      {
         // repeat to the maximum integer to find the first number that meets our conditions
```

```
            for (int i = higher; i < Int32.MaxValue; i++)
            {
               bool noGood = false;        // Assume that this number will work for us

               // repeat through 1 -> higher
               for (int si = 1; si < higher; si++)
               {
                  // Mod the number we are testing with the digit here in our range (1 -> upper)

                  if (i % si != 0)
                  {
                     // The result of a MOD is the remainder of the division. So if the result here is
not 0, i is not evenly divisible.
                     noGood = true;
                     break;  // No need to check the rest of the range
                  }
               }
               if (!noGood)
               {
                  // This number was evenly divisible by our entire range, this meets the condition
                  Console.WriteLine(i);
                  return i;   // Return the number we found
               }
            }
            return -1;  // Return -1 if we didn't find a suitable number
         }

         [TestMethod]
         public void TestProb2()
         {
            // The example range is 1 -> 10, the result should be 2520
            Assert.AreEqual(2520, prob2(10));
```

```
            // The Question range 1 -> 20, the result is 232,792,560
            Assert.AreEqual(232792560, prob2(20));
        }
    }
  }
}
```

Result



13/02/18 Unit test 3

Unit test 3 again the link I found I used the answers from it and adapted the code after understanding it. Test 3 was find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum. As the following are examples,

The sum of the squares of the first ten natural numbers is,

$$1^2 + 2^2 + ... + 10^2 = 385$$

The square of the sum of the first ten natural numbers is,

$$(1 + 2 + ... + 10)^2 = 55^2 = 3025$$

Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is $3025 - 385 = 2640$.

The link helped me and comments have been added and the code has been changed to ensure I understand it.

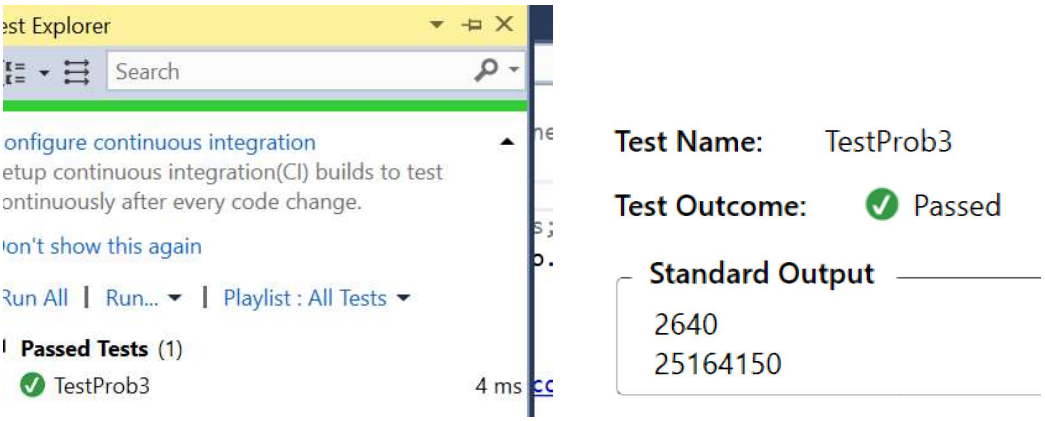
Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace p3
{
   //https://sites.google.com/site/eulerproblemsincsharp/home/problem-3
   [TestClass]
   public class unittest3
   {
      static void Main(string[] args)

      {

         Console.WriteLine((SumOfN(100) * SumOfN(100)) - SumSquareOf(100));

         Console.ReadLine();

      }


      static long SumSquareOf(int n)


      {
```

```
        return ((n * (n + 1) * (2 * n + 1) / 6));

    }

    static long SumOfN(int n)

    {

        return n * (n + 1) / 2;

    }

    // The sum of the squares of the first ten natural numbers is,
    //  12 + 22 + ... + 102 = 385
    // The square of the sum of the first ten natural numbers is,
    //  (1 + 2 + ... + 10)2 = 552 = 3025
    // Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is 3025 − 385 = 2640.
    // Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum.
    public static int Prob3(int limit)
    {
        // Start with 0
        int sum = 0;
        int squaresSum = 0;

        // start from 1 -> limit (inclusive)
        for (int i = 1; i <= limit; i++)
        {
            sum += i;                 // Add this number to our sum of numbers 1->limit
            squaresSum += (i * i);      // Square this number and add it to the sum of squares
        }
        int squaresq = sum * sum;           // Square the sum of the numbers
```

```
            int thedifference = squaresq - squaresSum;    // Get the difference between the sum of
squares and the square of the sum
            Console.WriteLine(thedifference);
            return thedifference;
        }

        [TestMethod]
        public void TestProb3()
        {
            // Make sure that Problem 3 1->10 is 2640 like the example
            Assert.AreEqual(2640, Prob3(10));

            // Problem 3 with 1 -> 100 is 25,164,150
            Assert.AreEqual(25164150, Prob3(100));
        }
    }
}
```

Result



16/02/18

I had another review with my lecturer and he was happy wit most of the tests. It was mentioned that could use the Microsoft test instead of NUnit which I did. However, test 3 was not working properly and building and running at runtime so it was advised to look at this which, I did. The

issue here was Visual Studio and my naming convention as I had spelt some code wrong. All this work so far was uploaded to GitHub as a draft so my lecturer could view and mark.

Coursework 3

17/02/18 Coursework 3

Coursework 3 was to send an email and let the user check their inbox whilst allowing them access to their address book and to make amendments such as, add, edit and delete an address book.

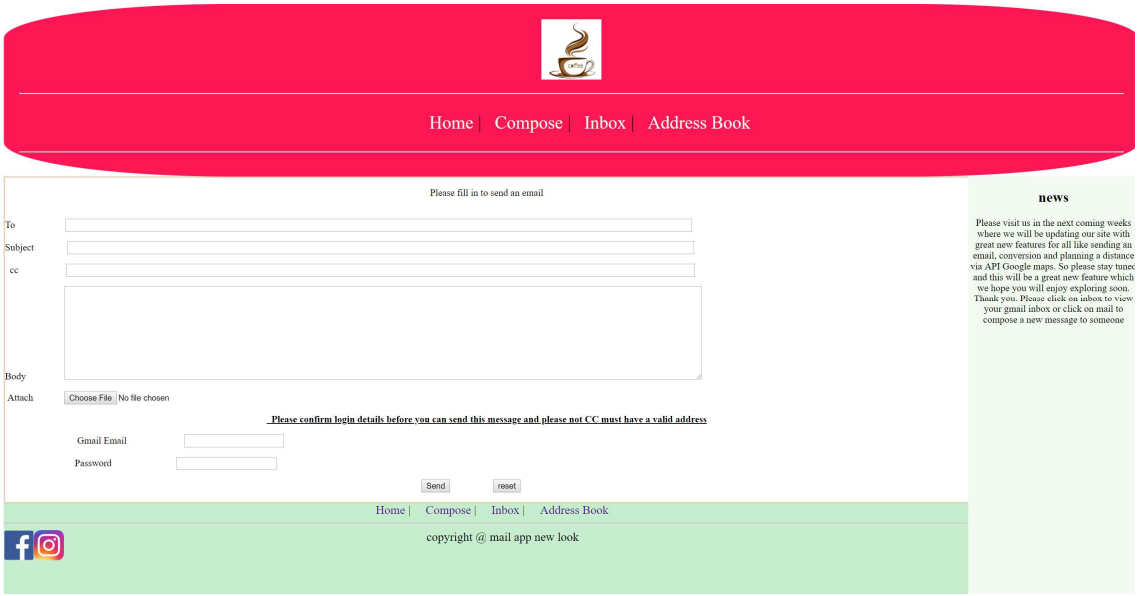
This task I had previously sending an email I had created in C# and I used my knowledge learnt from this to create a basic form and applied it. However, this coursework I learnt a lot as it required me to configure the SMTP, POP3 and use an API which I had not done before and by researching and finding various videos and step-by-step guides I feel I created an ideal solution. As when sending an email, you cannot send it from a spoof address as you must enter your username and password to send the email. The only down side due to the email servers of Google and Hotmail is that it will end up in the junk folder which I have no control over due to the filters.

The first step was to allow the user to send an email which, should allow the following which, the user can choose from:

- Recipient
- Title
- Message
- CC

I had help from the lecture's worksheet which showed me how to create a basic mail form and how to send an email which I created in coursework 1. A link I found helped me a lot to make the send email safer so the user would have to login before sending the email. The following link helps and shows how to create a email form step-by-step walkthrough which helped me and to understand the code as well (Khan, 2018).

Form design for send email



This is a simple layout and do feel improvements could have been made to make it look like a realistic email form by adding in images and a colourful top menu.

Code used to send an email

```
using System;
using System.IO;
using System.Net;
using System.Net.Mail;

namespace coursework4mailapp
{
   public partial class compose : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack) // asp.net which restores values
            txtTo.Text = Request["To"];
      }

      protected void SendEmailClick(object sender, EventArgs e)
      {
```

```
        using (MailMessage mailmessege = new MailMessage(txtEmail.Text, txtTo.Text))
        {
            // below is how to send an email whilst using smtp.

            // guidence from lecture notes, worksheets and
https://www.aspsnippets.com/Articles/Send-email-using-Gmail-SMTP-Mail-Server-in-
ASPNet.aspx

            // following code below declares the message contents for example to, cc, the main
body of the message and the subject accoridng to the form textboxes

            mailmessege.CC.Add(txtccemail.Text); // cc carbon copy to someone declares the
text box used
            mailmessege.Subject = txtSubject.Text;
            mailmessege.Body = txtBody.Text;
            if (fuAttachment.HasFile) // code for attach a file
            {
                string FileName = Path.GetFileName(fuAttachment.PostedFile.FileName);
                mailmessege.Attachments.Add(new
Attachment(fuAttachment.PostedFile.InputStream, FileName));
            }

            // mail code to send the mail with credentials needed we can only use google as this
is what has been set as default by me
            mailmessege.IsBodyHtml = false;
            SmtpClient smtp = new SmtpClient();
            smtp.Host = "smtp.gmail.com";
            smtp.EnableSsl = true;
            NetworkCredential netcr = new NetworkCredential(txtEmail.Text,
txtPassword.Text);
            smtp.UseDefaultCredentials = true;
            smtp.Credentials = netcr;
```

```
            smtp.Port = 587; // smtp port which is needed by the google server to send mail without won't work
            smtp.Send(mailmessege);
            ClientScript.RegisterStartupScript(GetType(), "pleasenote", "important('Email has been sent successfully you may close this page now.');", true);
        }
    }
  }
}
```

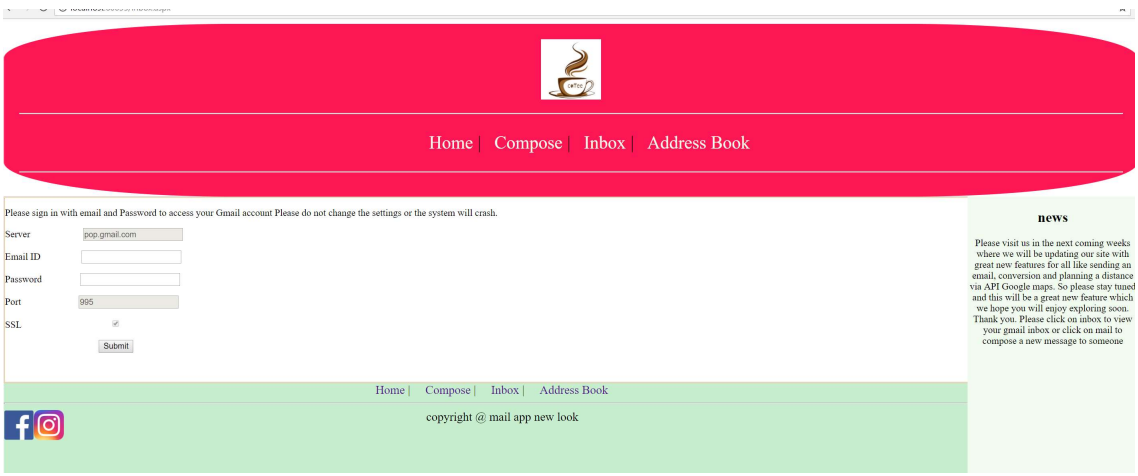
The notes helped from my lecturer's worksheet and the link helped me to create a send mail form with ease as most of the code I adapted from the link I followed and my lecture's notes.

19/02/18 creating the inbox

I had learnt in my previous module under-taken this can be done via POP3 which is new to POP which used to be used to read emails. This was shown to us in Network Services Engineering and from what I had learnt I applied my knowledge. As you could configure the ports and use POP3 to read an email or inbox.

I used two links which I found interesting which, helped me to step-by-step to create this form. Links were https://www.codeproject.com/Articles/188349/Read-Gmail-Inbox-Message-in-ASP-NET and https://www.aspsnippets.com/Articles/Fetch-and-read-email-messages-with-attachments-from-GMAIL-POP3-mail-server-in-ASPNet.aspx.

Design



Code

```
using System;
using System.Data;
using System.Web.UI;
using OpenPop.Pop3;

namespace coursework4mailapp
{
    // pop3 or imap is uesd to read mail learnt in NSE with Hugh Chadwick in great details with the INET D demons and configue files
    // from the NSE module network services enginering I took what I learnt and applied it to this
    // guidence from https://www.codeproject.com/Articles/188349/Read-Gmail-Inbox-Message-in-ASP-NET
    // guidence from https://www.aspsnippets.com/Articles/Fetch-and-read-email-messages-with-attachments-from-GMAIL-POP3-mail-server-in-ASPNet.aspx
    public partial class inbox : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void ReadEmails_Click(object sender, EventArgs e)
```

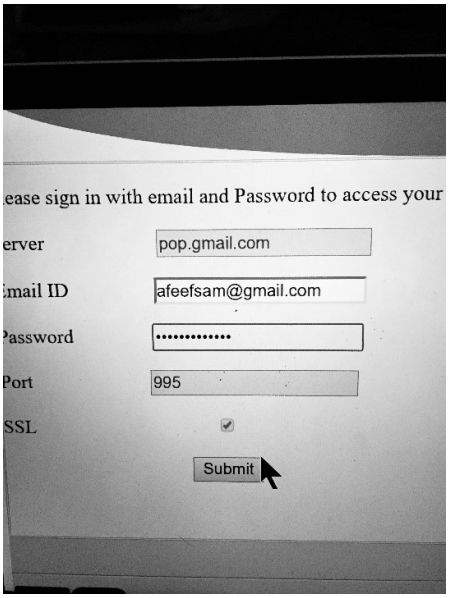
```
{
    Pop3Client pop3protocol; // using the POP3 client which is used in email you can use
IMAP as well or SMTP

    if (Session["Pop3Client"] == null) // session to check POP3 client server

    {
        pop3protocol = new Pop3Client();

        pop3protocol.Connect(txtServer.Text, int.Parse(txtPort.Text), chkSSL.Checked);

        pop3protocol.Authenticate(txtEmail.Text, txtpwd.Text);

        Session["Pop3Client"] = pop3protocol;
    }

    else

    {
        pop3protocol = (Pop3Client) Session["Pop3Client"];
    }

    var count = pop3protocol.GetMessageCount();

    var dtmsg = new DataTable();

    dtmsg.Columns.Add("Messageorder");

    dtmsg.Columns.Add("From");

    dtmsg.Columns.Add("Subject");

    dtmsg.Columns.Add("DateSent");
```

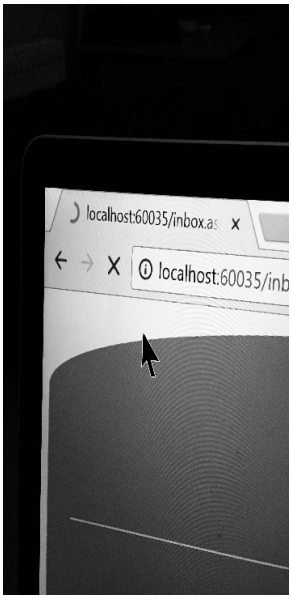
```
            dtmsg.Columns.Add("Body");

            var counter = 0;

            for (var i = count; i >= 1; i--)

            {
               try
               {
                  // gets the messages when clicked to display in window
                  var mgs = pop3protocol.GetMessage(i);

                  dtmsg.Rows.Add();

                  dtmsg.Rows[dtmsg.Rows.Count - 1]["Messageorder"] = i;

                  dtmsg.Rows[dtmsg.Rows.Count - 1]["From"] = mgs.Headers.From.Address;

                  dtmsg.Rows[dtmsg.Rows.Count - 1]["Subject"] = mgs.Headers.Subject;

                  dtmsg.Rows[dtmsg.Rows.Count - 1]["DateSent"] =
mgs.Headers.DateSent.ToLocalTime();

                  dtmsg.Rows[dtmsg.Rows.Count - 1]["Body"] =
mgs.MessagePart.GetBodyAsText();

                  counter++;

                  if (counter > 4)

                  {
                     break;
```

```
                }
            }
            catch (Exception ex)
            {

            }
        }
        GvEmails.DataSource = dtmsg;
        GvEmails.DataBind();
    }
  }
}
```

Test to show this working

First, we login, then we click the submit button and then we wait for a couple of minutes so the server can connect and find our information to display our messages in a grid view. Please note you cannot change the pre-filled boxes as they have been disabled to cause less error in case by accident someone does change them and says they cannot login. Also, you can only use this service if you have a google account only.



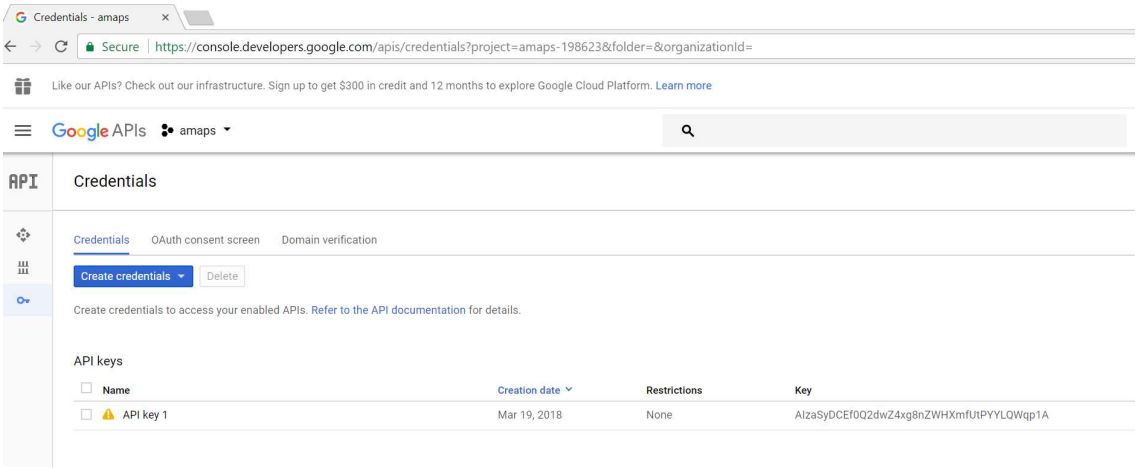


We login then we have to wait and then finally we can get our emails and we can click on them to view them as I created by using the guide via the link to view an email in the browser window by creating a hyperlink field. You can also, view the date received on.

20/02/18-21/02/18 Creating the address book

This had me confused and a lot of research and trial and error methods were carried out as I found a link with the library Aspron however, as I had used other libraries I came to know that it would not be compatible therefore, further research helped me with videos and guides to use OAuth2. I had never come across this however, the links helped me to understand it. It is google API which, you need a token for and can register for free. It is version 2 of the OAuth protocol also, called framework. This allows third-party applications to grant limited access to the HTTP services. Other research showed many different ways which, confused me but after watching a few links videos I felt comfortable using this and I decided to use this. With the help and some code snippets taken from the links I was able to consume this. This was a proud moment for me as I had learnt a new way which could be used in other industrial areas and other websites developers when creating a website.

This felt advanced and I was proud because I had learnt something challenging. I cannot say I know it all but I understood to create what was needed as it helped me to create a add, update and delete address book feature as well. You need to login and then sign into your Google account once re-directed but as the API key is registered and you put in the correct information about your localhost used then you are re-directed back on a successful login to view your address book on your website and make any necessary changes. The only disadvantage is you need to refresh and give some time like a few seconds for the update to happen which, is not in my control due to servers involved. However, research did show me that I could have created a database bind the data and refresh the server so it would update quicker but I felt this was a much easier way after understanding the procedure and principles of this. The Google Developers tools were used with the link in references page to create the token ID which, allowed the creation of the linking of the address book with Google so users, could get restricted access to modify it and edit it (OAuth2: Authenticate Users with Google - Google Chrome, 2018). Screenshot below shows the Google Developers interface which, was used.



Code

```
using System;
using System.Data;
using System.Threading;
using System.Web.UI;
using System.Web.UI.WebControls;
using Google.Contacts;
using Google.GData.Client;
using Google.GData.Contacts;
using Google.GData.Extensions;

namespace coursework4mailapp
{

    // guidence from https://www.youtube.com/watch?v=DYAwYxVs2TI
    // guidence from https://www.youtube.com/watch?v=HvQ6fbNFxO4
    // gudience from https://www.youtube.com/watch?v=WsRyvWvo4EI
    // guidence from https://www.youtube.com/watch?v=CtDE9gTwmyo

    // the following website helped me to understand Oauth2parameters and I have learnt something new which I can use later on in life website https://developers.google.com/identity/protocols/OAuth2WebServer
```

```
    // the videos have been a great step by step help in creating this there was the aspone method but after research i felt the gdata method was easier for me to understand
    public partial class addressbook : Page
    {
        private readonly OAuth2Parameters _oAuth2Parameters = new OAuth2Parameters
        {
            ClientId = "641646083232-is46s4sj4jr92resqrpk129h848hnh82.apps.googleusercontent.com",
            //acquire from google api app
            ClientSecret = "F8f1vfmcHgM3UgiKwweOVh85", //acquire from google api app it is the secret code needed for the program to access the contacts
            //normally it is hashed to prevent hackers
            RedirectUri = "http://localhost:60035/addressbook.aspx", //add to credentials in google api app
            Scope = "https://www.googleapis.com/auth/contacts", //allow through google api app
            AccessType = "offline",
            TokenType = "refresh"
        };

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                if (!String.IsNullOrWhiteSpace(Request.QueryString["code"]) && Session["token"] == null)
                {
                    _oAuth2Parameters.AccessCode = Request.QueryString["code"];

                    OAuthUtil.GetAccessToken(_oAuth2Parameters);

                    Session["token"] = _oAuth2Parameters;
```

```
            ListContacts();
        }
        else if (Session["token"] == null)
        {
            string url = OAuthUtil.CreateOAuth2AuthorizationUrl(_oAuth2Parameters);

            Response.Redirect(url);
        }

        ListContacts();
    }
}

private void ListContacts()
{
    var cr = new ContactsRequest(new RequestSettings("myapp", (OAuth2Parameters)
Session["token"]){PageSize = int.MaxValue, AutoPaging = false});
    Feed<Contact> f = cr.GetContacts();

    var table = new DataTable();
    table.Columns.Add("EmailID");

    foreach (Contact contact in f.Entries)
    {
        foreach (EMail email in contact.Emails)
        {
            DataRow row = table.NewRow();
            row["EmailID"] = email.Address;
            table.Rows.Add(row);
        }

        // following contacts will be shown in a data grid viewww with the number of contacts
the user has
```

```
            grdEmails.DataSource = table;
            grdEmails.DataBind();
            lblInfo.Text = "total contacts: " + table.Rows.Count;
        }
    }

    private void ListContacts(ContactsRequest cr)
    {
        Feed<Contact> f = cr.GetContacts();

        var table = new DataTable();
        table.Columns.Add("EmailID");

        foreach (Contact contact in f.Entries)
        {
            foreach (EMail email in contact.Emails)
            {
                DataRow row = table.NewRow();
                row["EmailID"] = email.Address;
                table.Rows.Add(row);
            }

            // following contacts will be shown in a data grid vieww with the number of contacts
the user has
            grdEmails.DataSource = table;
            grdEmails.DataBind();
            lblInfo.Text = "total contacts: " + table.Rows.Count;
        }
    }

    protected void grdEmails_OnRowEditing(object sender, GridViewEditEventArgs e)
    {
        grdEmails.EditIndex = e.NewEditIndex;
```

```
            ListContacts();
        }

        protected void grdEmails_OnRowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
        {
            grdEmails.EditIndex = -1;

        }

        protected void grdEmails_OnRowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            // first of we select the email contact and then delete it
            var selectedEmail = ((HiddenField)
grdEmails.Rows[e.RowIndex].FindControl("hdEmailID")).Value;

            var cr = new ContactsRequest(new RequestSettings("myapp",
(OAuth2Parameters)Session["token"]) { PageSize = int.MaxValue, AutoPaging = false });
            Feed<Contact> f = cr.GetContacts();
            foreach (Contact contact in f.Entries)
            {
                foreach (EMail email in contact.Emails)
                {
                    if (selectedEmail == email.Address)
                    {
                        cr.Delete(contact);
                        Response.Write(
                            "please wait for a few seconds then refresh for changes to take place due to
the google server thank you");

                        break;
                    }
```

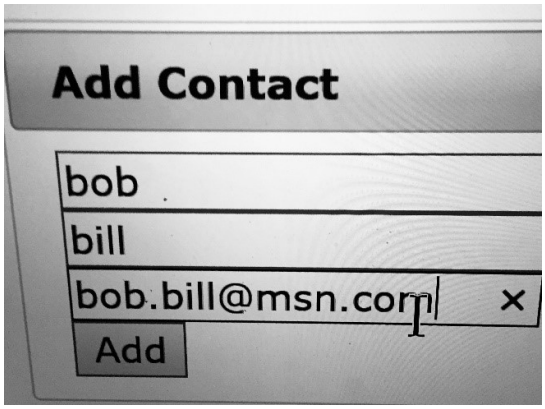
```
            }
        }
        ListContacts(cr);
    }

    protected void grdEmails_OnRowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        var selectedEmail = ((HiddenField)
grdEmails.Rows[e.RowIndex].FindControl("hdEmailID")).Value;

        var cr = new ContactsRequest(new RequestSettings("myapp",
(OAuth2Parameters)Session["token"]) { PageSize = int.MaxValue, AutoPaging = false });
        Feed<Contact> f = cr.GetContacts();
        foreach (Contact contact in f.Entries)
        {
            foreach (EMail email in contact.Emails)
            {
                if (selectedEmail == email.Address)
                {
                    email.Address = ((TextBox)
grdEmails.Rows[e.RowIndex].FindControl("txtEmailID")).Text;
                    cr.Update(contact);
                    Response.Write(
                        "please wait for a few seconds then refresh for changes to take place due to
the google server thank you");
                    break;
                }
            }
        }
        grdEmails.EditIndex = -1;
        ListContacts(cr);
    }
```

```
protected void btnAdd_OnClick(object sender, EventArgs e)
{
    var newEnt = new Contact
    {
        Name = new Name
        {
            FullName = txtFirst.Text + " " + txtLast.Text,
            GivenName = txtFirst.Text,
            FamilyName = txtLast.Text,
        }
    };
    // Set the contact's name.
    // Set the contact's e-mail addresses.
    newEnt.Emails.Add(new EMail
    {
        Primary = true,
        Rel = ContactsRelationships.IsHome,
        Address = txtEmail.Text
    });

    newEnt.IMs.Add(new IMAddress
    {
        Address = txtEmail.Text,
        Primary = true,
        Rel = ContactsRelationships.IsHome,
        Protocol = ContactsProtocols.IsGoogleTalk,
    });

    var cr = new ContactsRequest(new RequestSettings("myapp",
(OAuth2Parameters)Session["token"]) { PageSize = int.MaxValue, AutoPaging = false });

    cr.Insert(cr.GetContacts(), newEnt);
```

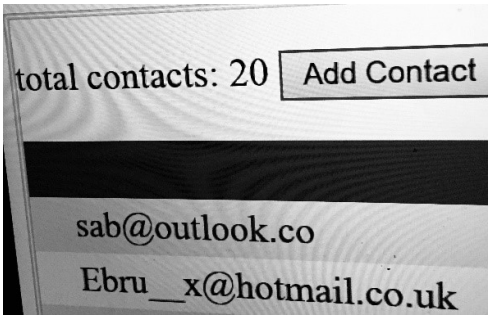
```
            ListContacts(cr);
            Response.Write("please refresh the page for the add to happen");
        }
    }
}
```

Outcome

As I logged in I can go to my address book we can see I got 19 contacts so if I click add contact I can add a contact by filling the details on the right. If you click on the user email it will appear in the to box on the compose message box allowing you to choose the email you want to send a message to from your contacts in your address book.

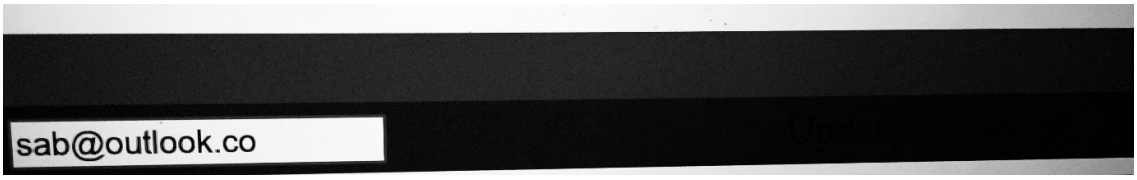


I click add and then refresh the browser and I should see 20 contacts and this email in my address book.

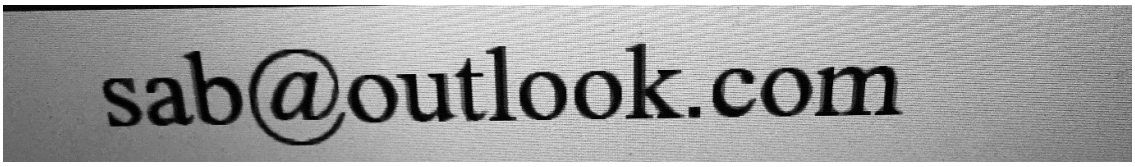


we can see 20 contacts and email below in our list

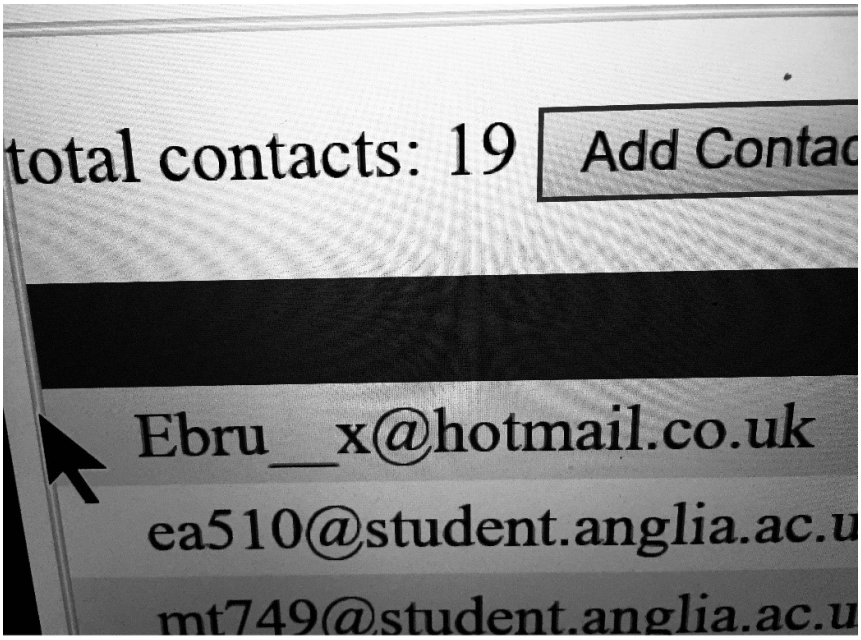
We can also edit and delete an address book contact.



We will change the co at the end to com.



We can delete this email and it should update contacts to 19.



This is the coursework complete. I learnt a lot about new ways of implementing emails and sending them using SMTP, POP3 and a API which I had never used before or even heard of by research. I am pleased because this can be used in other industrial areas and in web-creation.

22/02/18- 28/02/18 Revising coursework 1 to 3

I had completed coursework one to three however, I ensured they met the specification. I revisited these coursework's to tidy them up and to make them look professional and neat. I ensured they had a consistent layout, logo and design.

Coursework 4

01/03/18-14/03/18 Coursework 4

Coursework four was to consume and implement web services. This was a new subject for me to learn as I had no idea on how to consume a web service. However, research videos and tutorials explained what a web service was and how to consume it. I learnt that a web service is a function that can be accessed by programs over the Internet. For example, if I create a PHP website that produces and has HTML, it targets the browser and when we read the page on the internet using Chrome or any browser. Another example is when you try to search your home address on google maps you can view it due to the satellite API extensions used (tutorialspoint.com, 2018).

The web service is made up of XML plus HTTP and all the standard web services work using the following:

- SOAP = simple object access protocol which, is to transfer a message.
- UDDI = Universal Description, Discovery and Integration which tags the data.
- WSDL = Web service description language which, shows the availability of the service.

03/03/18 Part a

The first stage was to research how to consume a web service and I found a video which showed me how to do this and what one was (Part 2 Consuming a web service, n.d.) and (How To Call & Use a Webservice in .NET (C#), n.d.). Therefore, after following this video I was able to consume the following web services.

1. http://www.webservicex.net/ConvertArea.asmx?WSDL

The first web service was to convert area like a square to rectangle area which is an example, you could convert area of square to square if wanted. This was done successfully please refer to the program which is on the USB under coursework 4 using web services to view the code for this. This was a success until logging it to the logbook because the service now is inactive and this is out of my control. However, my lecturer had seen this previously working.

04/03/18

The other part was to consume the following web service.
http://www.webservicex.net/ConvertComputer.asmx?WSDL.

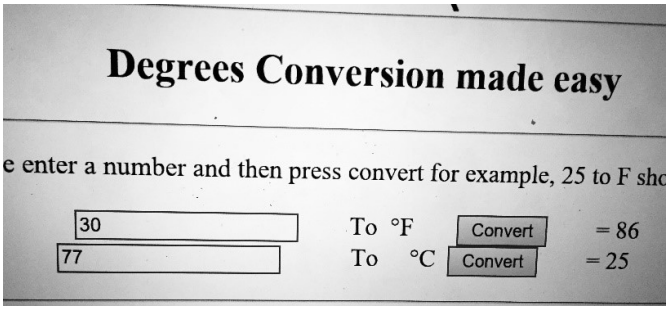
After researching I realised that I had created a C# program before where I was converting megabytes to gigabytes and vice-versa. I used some of that code but consumed the web service from what I had learnt from the video. This was a success until logging it to the logbook because the service now is inactive and this is out of my control. However, my lecturer had seen this previously working.

06/08/18

After a rest from this I decided to start part b. This was to create the following web services which I did successfully by watching the video links from my previous video. This was:

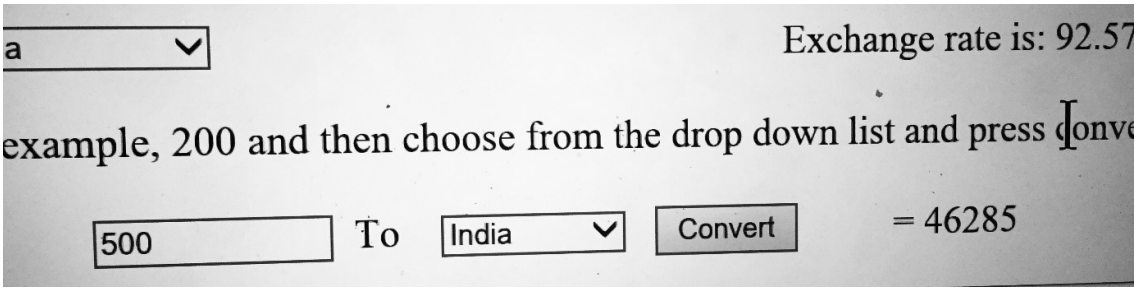
- Convert ºC to ºF
- Convert ºF to ºC
- Based on a country name, return the exchange rate (this must be chosen from a drop-down box)
- Based on a country name and a value in £'s display the resulting currency (this must be chosen from a drop-down box)

The outcome of this was that this was working and we can see C to F and F to C first. As if we put 30 as C to F we should get 86 and if we put 77 as F we should get 25 in C which we do.



Outcome for Based on a country name, return the exchange rate (this must be chosen from a drop-down box). Also, for the based on a country name and a value in £'s display the resulting currency (this must be chosen from a drop-down box)

Results

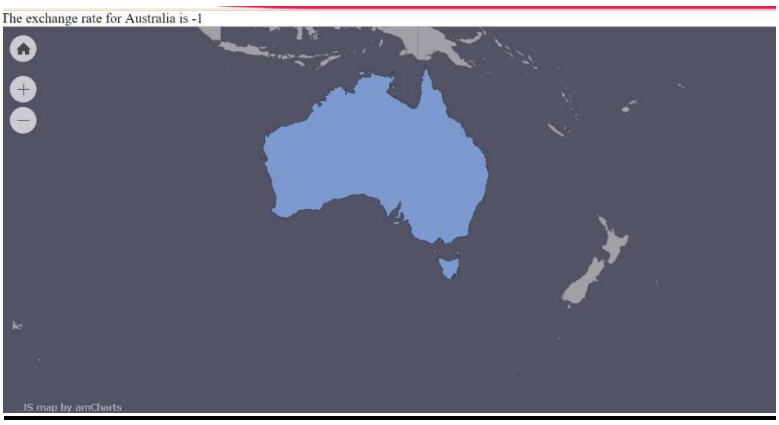


07/03/18-11/03/18

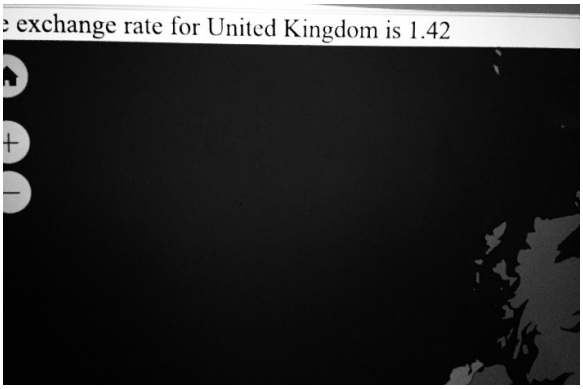
The next task seemed simple as it was to implement the following web services which I had already done before in part a. Implement the web service http://www.webservicex.net/CurrencyConvertor.asmx?WSDLgiving a visual representation of the currency / country

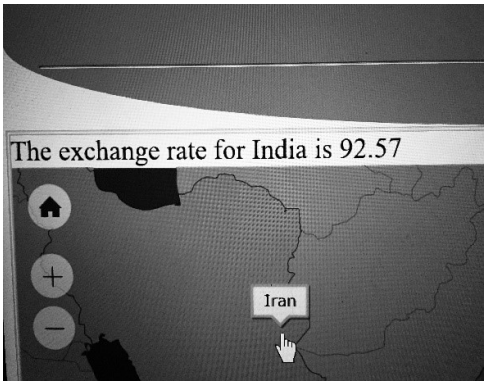
When trying this and showing it on a map as I found that you can do this via arm charts to create a map the web service was out of use and kept showing return -1. First, I felt I was doing something wrong but when speaking to my lecturer it worked for him once or twice and then did not. The best advice was given was to implement another web service like this which I found on the ASP.NET forum. This was the following web service which worked for me http://fx.cloanto.com/webservices/CurrencyServer.asmx?WSDL taken from (Currency Conversion Web Service, 2018).

Old web service which returned -1 codes can be seen in Appendix (please refer).



The working web service outcome



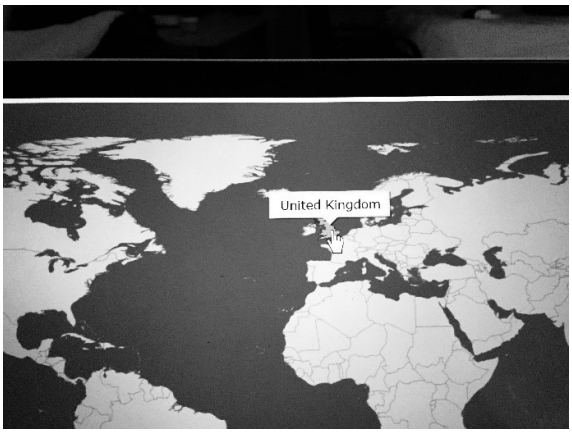


14/02/18 Part c

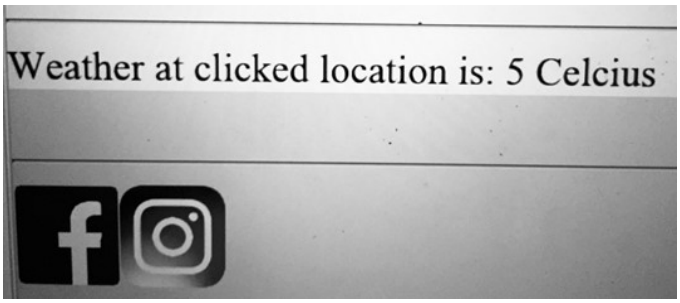
This was to research and implement your own web service from the site http://www.webservicex.net.

Researching and looking at this site I was interested in a weather web service as I use to do a lot of projects at school about weather and this subject interested me. I was able to use a web service found to show the temperature when clicked on the map in Celsius.

Outcome



if we click on United Kingdom we will get the current temperature.



This showed it worked the only disadvantage of this was that for some reason you could not check it multiple times without restarting the program or refreshing it. Web service used was http://www.webservicex.net/globalweather.asmx?WSDL.

Coursework 5

15/03/18 -03/04/18 Coursework 5

Coursework five I had to take a break in between and it was Easter holidays as well. This was because I was stuck on certain aspects of the project and decided to pay more attention to part b. After resuming I learnt a lot from research and implemented it to use the Google APIs as stated for the coursework.

An API stands for Application Programming Interface, which, is a software that allows two applications to interact and talk with each other. An example, of this is when you use social media like Instagram, Snapchat or Facebook or to check the weather on your phone it uses an API.

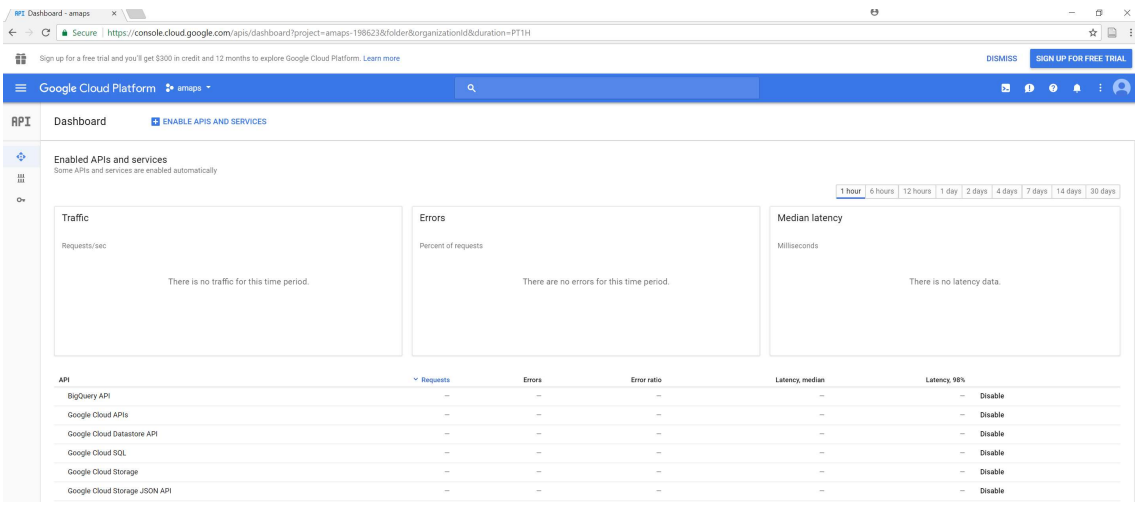
An example, of this is when you use your mobile phone it connects to the internet if you use Facebook or any other application. The data is sent to a server which receives it, interprets it and then performs the necessary action which is required for you to see the information you wanted in a reliable way. This is because the data which is received it is turned into meaningful data which the user can understand and this is what happens in an API.

A good way to explain this is which helped me is when you go to a restaurant we sit on a table and we are presented with a menu full of choices. The kitchen is part of the system which, will prepare the order you chose. This cannot be done without communication so there is a person in between the waiter who takes your order and he brings you to the food to your table once it is cooked. All taken from (What is an API? (Application Programming Interface), 2016).

16/3/18 part 1

The first task was to use the Google Maps API to ask the user their postcode and house number. From, this populate a delivery from with each address element in separate boxes. The second part was to allow the user to select from the map so they can change the displays to satellite, hybrid and display a marker of their address.

The hardest bit for me was to display the marker had I had to revisit this after conducting further research to understand how this was done. I was confident on using the Google API as I had used this for coursework 3. All I had to do was to login to my Google Developer's account and enable the map API services. Screenshots below show this.



I found a link which helped me to understand how to use the API and how to create the Google Map API and I used the link and followed it step-by-step. This helped me to create the project. This was from (C# Tutorial 90: How to Display Google Maps in C# Windows Form, n.d.), (Combining Google Maps with ASP.NET Web Forms – Part 1, 2013) and (Google Maps API V3 for ASP.NET - Code Project, 2018).

Outcome

Please fill in by number and post code your address will be autofilled by google maps

Please Cick Get address after the address has been filled in the textboes to show the location point on the map.

House no 21

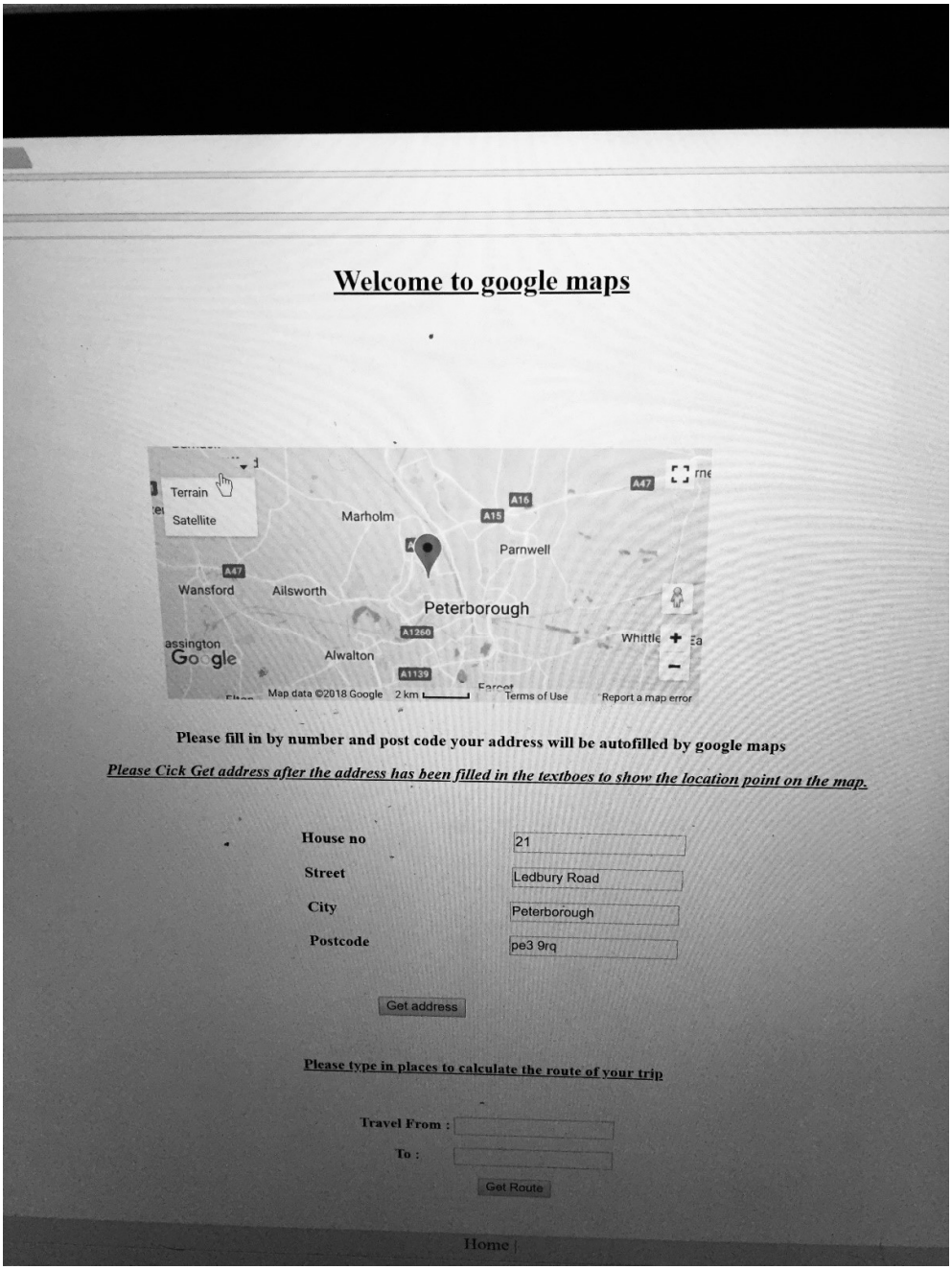
Street

City

Postcode pe3 9rq

Get address

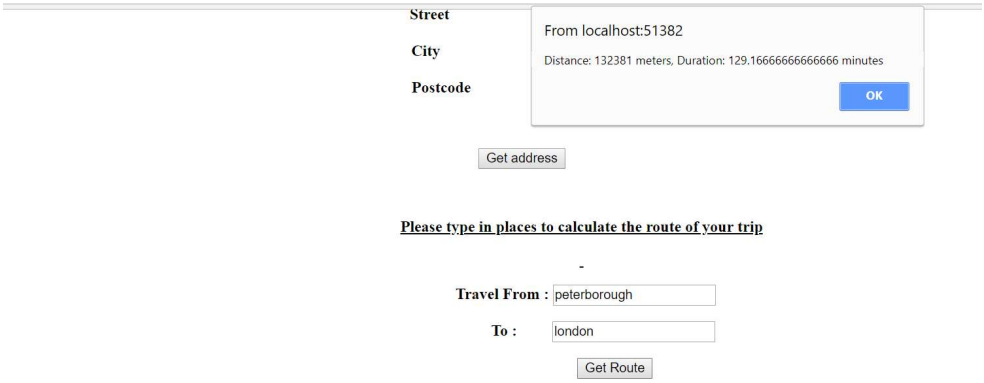
We can put in the house number and the postcode and if we click get address we can see the full address being filled in the text-boxes and with the help of research carried out a marker is also displayed on the map to show the users address.



For the higher marks the coursework required some extra functionality which some I had already implemented and some I had not. The user could zoom in the map which was part of what I had implemented as a result of research.

The first task was to allow the user to enter two postcodes use the Google Map to return the distance between the two points.

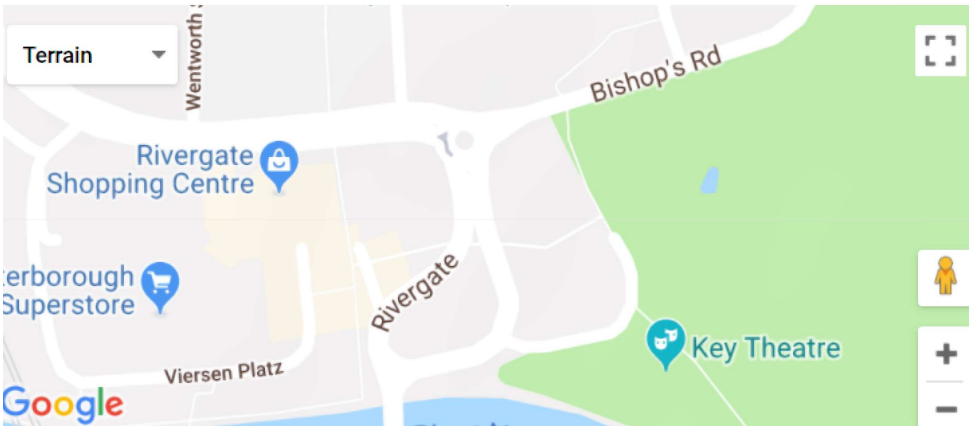
Outcome



The second part was to display markers on the map between the two postcodes. This was done with ease as a result of result of research carried out.

Outcome

The final part was to allow the user to Zoom the map and this was implemented in the first part as because of the research I had carried out. This is shown below



The code used is below

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="default.aspx.cs" Inherits="googlemaps._default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">
    <style type="text/css">
        .auto-style1 {
            text-align: center;
        }
        .auto-style2 {
            text-align: center;
            text-decoration: underline;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder3" runat="server">
</asp:Content>
```

```
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder4" runat="server">

   <script async defer
src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDCEf0Q2dwZ4xg8nZWHXmfUtP
YYLQWqp1A&callback=initialize"></script>

        <script type="text/javascript">

            // for this google map service I have followed this tutuiral and taken some code from
it https://www.codeproject.com/Articles/291499/Google-Maps-API-V-for-ASP-NET

            // gudience from here https://blog.learningtree.com/combining-google-maps-with-
asp-net-web-forms-part-1/ and https://www.youtube.com/watch?v=lRoAjs8RwfE

            // declaration of lcoal variables

                var mapGcode; // google service will be using geocode

                var GMap; // google service will be used for the map

                var googledirecservice; // google direction service will be used

                var googledirecdis; // google direction display will be used

                var messagewindalert; // error message to inform the user if cannot find the
what has been entered


                function initialize() {


                   googledirecservice = new google.maps.DirectionsService();

                   googledirecdis = new google.maps.DirectionsRenderer;

                   messagewindalert = new google.maps.InfoWindow;

                   mapGcode = new google.maps.Geocoder();

                        var latlng = new google.maps.LatLng(52.569499, -0.24053);
```

```
var options =
    {
        zoom: 8,
        center: new google.maps.LatLng(52.569499, -0.24053),
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        mapTypeControl: true,
        mapTypeControlOptions:
        {
            // If style is not specified, it defaults to buttons
            style:
google.maps.MapTypeControlStyle.DROPDOWN_MENU, // display map types as drop down
menu
            poistion:
google.maps.ControlPosition.TOP_RIGHT, // poisiton of the buttons
        // display of map views
            mapTypeIds:
[google.maps.MapTypeId.HYBRID, google.maps.MapTypeId.TERRAIN,
google.maps.MapTypeId.HYBRID, google.maps.MapTypeId.SATELLITE]
        },
        navigationControl: true,
        navigationControlOptions:
        {
            style:
google.maps.NavigationControlStyle.ZOOM_PAN // sytle taken from microsoft
```

```
                    },
                    scaleControl: true,
                    disableDoubleClickZoom: true,
                    streetViewControl: true,
                    draggableCursor: 'move'
                };
            GMap = new google.maps.Map(document.getElementById("map"),
options);
            googledirecdis.setMap(GMap);
        }
    // functions used to ger the adddress for the textboxes
        function doLocation() {
            var houseNo =
document.getElementById('ContentPlaceHolder4_TextBox1').value;
            var street =
document.getElementById('ContentPlaceHolder4_TextBox2').value;
            var city =
document.getElementById('ContentPlaceHolder4_TextBox3').value;
            var postcode =
document.getElementById('ContentPlaceHolder4_TextBox4').value;

            mapGcode.geocode({ componentRestrictions: { country: 'GB',
postalCode: postcode } }, function (res, status) {
                if (status == google.maps.GeocoderStatus.OK) {
```

```
document.getElementById('ContentPlaceHolder4_TextBox2').value =
res[0].address_components[1].long_name;


document.getElementById('ContentPlaceHolder4_TextBox3').value =
res[0].address_components[2].long_name;
                        var address = houseNo + " " + street + " " + city + " " + postcode;
                        mapGcode.geocode({ address: address, componentRestrictions: {
country: 'GB', postalCode: postcode } }, function (cres, cstatus) {
                            if (cstatus == google.maps.GeocoderStatus.OK) {
                                GMap.setCenter(cres[0].geometry.location);
                                GMap.setZoom(11);
                                var markerpoint = new google.maps.Marker({ 'map': GMap,
'position': cres[0].geometry.location });
                                messagewindalert.setContent(cres[0].formatted_address);
                                messagewindalert.open(theMap, markerpoint);



document.getElementById('ContentPlaceHolder4_TextBox1').value =
cres[0].formatted_address;
                            }
                        });
                            }
                            else {
                                    alert("Failed or inccorrect details. Status: " + status);
```

```
                }
            });
        }
    // function used for the travel directions and markers
        function doTravel() {
            var cause = document.getElementById("travelfrom").value; // gets value from the textbox
            var journey = document.getElementById("travelto").value; // gets value from textbox
            var request = {
                origin: cause, // start from the source
                destination: journey,
                travelMode: 'DRIVING' // the travel mode set to driving can be set to walking
            };

            // a lot of help and some code has been taken from https://stackoverflow.com/questions/3251609/how-to-get-total-driving-distance-with-google-maps-api-v3

            googledirecservice.route(request, function (response, status) {
              if (status == google.maps.DirectionsStatus.OK) {
                var EndJourneyDis = 0; // this is the total distance
                var EndJourneydura = 0; // this is the total duration taken
```

```
                    var legs = response.routes[0].legs;
                    for (var i = 0; i < legs.length; ++i) {
                        EndJourneyDis += legs[i].distance.value;
                        EndJourneydura += legs[i].duration.value;
                    }
                    alert('Distance: ' + EndJourneyDis + ' meters, Duration: ' +
EndJourneydura / 60 + ' minutes');
                        googledirecdis.setDirections(response);
                    }
            // else display a message alerting user cannot find what they have entered
                    else {
                        alert("Directions have failed please refresh or try again: "
+ status);
                    }
                });
            }
            //window.onload = initialize;
        </script>
        <h2 class="auto-style2">Welcome to google maps</h2>
    <p> </p>
    <p> </p>
    <p> </p>
    <p> </p>
```

```
<p> </p>

<p>  </p>

<p> </p>

<p> </p>

<p> </p> <hr=2 />

<p class="auto-style1">

     </p>

<p class="auto-style1">

     </p>

<p class="auto-style1">

    <strong>

    <asp:Label ID="Label1" runat="server" Text="Please fill in by number and post code
your address will be autofilled by google maps"></asp:Label>

     </strong></p>

<p class="auto-style2">

    <strong><em>Please Cick Get address after the address has been filled in the textboes to
show the location point on the map.</em></strong></p>

<p class="auto-style1">

      </p> <hr=2 />

<p class="auto-style1"><strong>     House
no            &nb
sp;            &n
bsp;         

    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

```
        </strong>
    </p>
    <p class="auto-style1"><strong>    
Street            
             
;            &nbs
p;  
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
        </strong>
    </p>
    <p class="auto-style1"><strong>    
City            &
nbsp;           
             
;    
        <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
        </strong>
    </p>
    <p class="auto-style1"><strong>    
Postcode           &nbs
p;            &nb
sp;        
        <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
        </strong>
    </p>
    <p> </p>
```

```
<p>            &
nbsp;            
             
;            &nbs
p;            &nb
sp;           &n
bsp;           &
nbsp;           
             
;            &nbs
p;            &nb
sp;           &n
bsp;       
           

      <input type="button" value="Get address" onclick="doLocation()"/>

  </p> <hr=2 />

  <p> </p>

  <p class="auto-style2"><strong>Please type in places to calculate the route of your
trip</strong></p>

  <p class="auto-style2"> </p>

  <p class="auto-style1"> <strong>

                  Travel From : <input id="travelfrom" type="text" name="name" value=""
/>

          </strong>


  </p>
```

```
   <p class="auto-
style1">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; <strong>To
:&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; <input id="travelto" type="text"
name="name" value="" /></strong> <strong>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; </strong>


   </p>

   <p class="auto-
style1">&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp; <strong>&amp;nbsp;<input type="button" value="Get Route"
onclick="doTravel()" /></strong>


   </p>

   <p>&amp;nbsp;</p>

<div id ="map"   style="width: 500px; top: 488px; left: 513px; position: absolute; height:
238px; text-align: center;">

   // above map the sytle and how it will appear on the screen and the size


</div>

</asp:Content>
```

Conclusion

Conclusiion

I can conclude from these coursework is that I had previously used ASP.NET however, I did not know this much despite using MVC and the basic features. From these coursework's, it has given me the confidence to use ASP.NET with ease. I have learnt there is a lot of uses and a lot of projects can be created using this powerful framework. It is used in industry and in work places today is one of the main frameworks used today and has a good future. Many organisations and businesses use this framework for many projects from creation of websites and media for weather services because of all these extensions, web services and APIs can be used to send and receive data with ease.

I have learnt a lot because of research and by using what I have learnt, I was able to apply it to complete the 5-main coursework's and what I had learnt now I can use for part b. This module has extended my knowledge and has given me more confidence and knowledge to extend my expectancies and to use what I learnt to apply in other industrial work places. This is because ASP.NET because it is not only used for websites but it can be used for many other projects as well which we use everyday and we do not even realise it. I have learnt a lot and have will use this framework again and when deciding I will consider this framework first over others because I have learnt a lot and feel comfortable using it. Also, I feel it has a lot to offer and is a powerful framework which can work closely with other programming languages and other frameworks to create a all in one projects with great ideal features and functionality which may not be achieved by other frameworks alone.

References

References

Anon 2013. *Combining Google Maps with ASP.NET Web Forms – Part 1*. [online] Learning Tree Blog. Available at: <https://blog.learningtree.com/combining-google-maps-with-asp-net-web-forms-part-1/> [Accessed 12 Apr. 2018].

Anon 2016. *What is an API? (Application Programming Interface)*. [online] MuleSoft. Available at: <https://www.mulesoft.com/resources/api/what-is-an-api> [Accessed 12 Apr. 2018].

Anon 2018. *Currency Conversion Web Service*. [online] The Official Forums for Microsoft ASP.NET. Available at: <https://forums.asp.net/t/1297305.aspx?Currency+Conversion+Web+Service> [Accessed 11 Apr. 2018].

Anon 2018. *Generate random password in C#.* [online] Available at: <https://madskristensen.net/blog/generate-random-password-in-c/> [Accessed 12 Apr. 2018].

Anon 2018. *Google Maps API V3 for ASP.NET - CodeProject*. [online] Available at: <https://www.codeproject.com/Articles/291499/Google-Maps-API-V-for-ASP-NET> [Accessed 12 Apr. 2018].

Anon 2018. *How To Create Master Page In ASP.NET*. [online] Available at: <https://www.c-sharpcorner.com/article/how-to-create-master-page-in-asp-net/> [Accessed 12 Apr. 2018].

Anon 2018. *OAuth2: Authenticate Users with Google - Google Chrome*. [online] Available at: <https://developer.chrome.com/apps/tut_oauth> [Accessed 12 Apr. 2018].

Anon 2018. *Step by Step working with GitHub Repository and Visual Studio 2015 | Infragistics Blog*. [online] Available at: <https://www.infragistics.com/community/blogs/b/dhananjay_kumar/posts/step-by-step-working-with-github-repository-and-visual-studio-2015> [Accessed 12 Apr. 2018].

Anon 2018. *What is unit testing? - Definition from WhatIs.com*. [online] Available at: <https://searchsoftwarequality.techtarget.com/definition/unit-testing> [Accessed 12 Apr. 2018].

ASP.NET Tutorial 6- Create a Login website - Login page & Validating User and Password in database. n.d. ProgrammingKnowledge Available at: <https://www.youtube.com/watch?v=QoPABrUknsE&index=6&list=PLS1QulWo1RIaM8-S7kTHgWd_pGNu-CyQS.> [Accessed 12 Apr. 2018].

C# Tutorial 90: How to Display Google Maps in C# Windows Form. n.d. ProgrammingKnowledge Available at: <https://www.youtube.com/watch?v=lRoAjs8RwfE> [Accessed 12 Apr. 2018].

Hashing with SHA1 Algorithm in C#. n.d. Tuto4free Available at: <https://www.youtube.com/watch?v=oJpZ5ygg4qQ> [Accessed 12 Apr. 2018].

How To Call & Use a Webservice in .NET (C#). n.d. DevChannel Available at: <https://www.youtube.com/watch?v=lmQ8uMynZP0> [Accessed 11 Apr. 2018].

Khan, M.A., 2018. *ASPSnippets: Code Snippets, Tutorials, Articles, Tips on ASP.Net SQL Server, Windows, C#, VB.Net, AJAX, jQuery, AngularJS and MVC.* [online] Available at: <http://www.aspsnippets.com/Home.aspx> [Accessed 12 Apr. 2018].

Part 2 Consuming a web service. n.d. kudvenkat Available at: <https://www.youtube.com/watch?v=ycKnYOlQDEE> [Accessed 11 Apr. 2018].

tutorialspoint.com, 2018. *What are Web Services*. [online] www.tutorialspoint.com. Available at: <https://www.tutorialspoint.com/webservices/what_are_web_services.htm> [Accessed 11 Apr. 2018].

Appendix for code for the world currency which the server did not work for coursework 4

Web services server errors as server been taken down code for world currency web service.

```
1 using System.Collections.Generic;
2 using System.ComponentModel;
3 using System.Web.Services;
4 using System;
5 using System.Linq;
6 using
usingwebservices.CurrencyConvertor
; 7
8  namespace
usingwebservices 9 {
10      /// <summary>
11      ///    Summary description for CurrencyService
12      /// </summary>
13      [WebService(Namespace = "http://tempuri.org/")]
14      [WebServiceBinding(ConformsTo =
          WsiProfiles.BasicProfile1_1)]
15      [ToolboxItem(false)]
16      // To allow this Web Service to be called from script,
          using ASP.NET AJAX, uncomment the following line.
17      // [System.Web.Script.Services.ScriptService]
18      public class CurrencyService :
WebService 19 {
20          private static readonly Dictionary<Country, double>
ExchangeRates
              = new Dictionary<Country, double>
21          {   23
22              24
```

```
25
26          // list of countires with exchange rates
27          {Country.US, 1.391105},
28          {Country.India, 90.519029},
29          {Country.Australia, 1.782861},
30          {Country.Canada, 1.765812},
31          {Country.Singapore, 1.843563},
32          {Country.Swiss, 1.306380},
        }   {Country.Malaysia, 5.451760},
;           {Country.Japan, 149.354322},
33          {Country.China, 8.832162},

34      [WebMethod]
35      public double ExchangeRate(Country
country) 36    {
37          return
ExchangeRates[country]; 38     }
39
40      [WebMethod]
41      public double CalculateExchangeRate(double value,
Country country) 42{
43          return
value*ExchangeRates[country]; 44    }
45
46
47      public enum Country
48      {
49          // ist of countries entered
50          US,
51          India,
```

```
52          Australi
53          a,
54          Canada,
55          Singapor
56          e,
57          Swiss,
58          Malaysia
59          , Japan,
            China

}
60
61

62      private static readonly Tuple<string,                    ↵
          string, CurrencyConvertor.Currency>[]
          CountriesAndCurrencies =
63
            74
{           75
64
            76
65          77
66
            78
67
68          79
            80
69
70          81
            82
71
            83
72          84
73
```

```
85          new Tuple<string, string,
86            CurrencyConvertor.Currency>
              ("Afghanistan", "Afghan afghani",
87              CurrencyConvertor.Currency.AFA),
            new Tuple<string, string,
88            CurrencyConvertor.Currency>("Albania",
89            "Albanian lek",
                CurrencyConvertor.Currency.ALL),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Algeria",
              "Algerian dinar",
                CurrencyConvertor.Currency.DZD),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Andorra",
              "Euro", CurrencyConvertor.Currency.EUR),
            new Tuple<string, string,
               CurrencyConvertor.Currency>("Antigua and
               Barbuda", "East Caribbean dollar",
               CurrencyConvertor.Currency.XCD),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Argentina",
              "Argentine peso",
                CurrencyConvertor.Currency.ARS),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Australia",
              "Australian dollar",
                CurrencyConvertor.Currency.AUD),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Austria",
              "Euro", CurrencyConvertor.Currency.EUR),
            new Tuple<string, string,
              CurrencyConvertor.Currency>("Bahamas",
```

```
"                string, string,
B               CurrencyConvertor.Currency>("Bahrain",
a               "Bahraini dinar",
h                 CurrencyConvertor.Currency.BHD),
a           new Tuple<string, string,
m             CurrencyConvertor.Currency>
i             ("Bangladesh", "Bangladeshi taka",
a               CurrencyConvertor.Currency.BDT),
n           new Tuple<string, string,
d             CurrencyConvertor.Currency>("Barbados",
o             "Barbadian dollar",
l               CurrencyConvertor.Currency.BBD),
l           new Tuple<string, string,
a             CurrencyConvertor.Currency>("Belgium",
r             "Euro", CurrencyConvertor.Currency.EUR),
"           new Tuple<string, string,
,             CurrencyConvertor.Currency>("Belize",
  Curren      "Belize dollar",
  cyConv        CurrencyConvertor.Currency.BZD),
  ertor.    new Tuple<string, string,
  Curren    CurrencyConvertor.Currency>("Benin",
  cy.BSD
  ),
n
  e
  w
  T
  u
  p
  l
  e
  <
  s
```

```
                "West African CFA franc",
90              CurrencyConvertor.Currency.XOF),
91            new Tuple<string, string,
              CurrencyConvertor.Currency>("Bhutan",
              "Bhutanese ngultrum",
92            CurrencyConvertor.Currency.BTN),
93            new Tuple<string, string,
              CurrencyConvertor.Currency>("Bolivia",
              "Bolivian boliviano",
94            CurrencyConvertor.Currency.BOB),
95            new Tuple<string, string,
              CurrencyConvertor.Currency>("Botswana",
              "Botswana pula",
96            CurrencyConvertor.Currency.BWP),
97            new Tuple<string, string,
              CurrencyConvertor.Currency>("Brazil",
              "Brazilian real",
98            CurrencyConvertor.Currency.BRL),
99            new Tuple<string, string,
              CurrencyConvertor.Currency>("Brunei",
              "Brunei dollar",
100             CurrencyConvertor.Currency.BND),
101           new Tuple<string, string,
              CurrencyConvertor.Currency>("Burkina Faso",
              "West African CFA franc",
102             CurrencyConvertor.Currency.XOF),
103           new Tuple<string, string,
              CurrencyConvertor.Currency>("Burundi",
              "Burundian franc",
104             CurrencyConvertor.Currency.BIF),
105           new Tuple<string, string,
              CurrencyConvertor.Currency>("Cambodia",
              "Cambodian riel",
```

```
106             CurrencyConvertor.Currency.KHR),
107         new Tuple<string, string,
            CurrencyConvertor.Currency>("Cameroon",
            "Central African CFA franc",
108             CurrencyConvertor.Currency.XAF),
109         new Tuple<string, string,
            CurrencyConvertor.Currency>("Canada",
            "Canadian dollar",
110             CurrencyConvertor.Currency.CAD),
111         new Tuple<string, string,
            CurrencyConvertor.Currency>("Cape Verde",
            "Cape Verdean escudo",
112             CurrencyConvertor.Currency.CVE),
113         new Tuple<string, string,
              CurrencyConvertor.Currency>("Central African
              Republic",
114               "Central African CFA franc"
                  ,
                  CurrencyConvertor.Currenc
                  y.XAF),
115         new Tuple<string, string,
            CurrencyConvertor.Currency>("Chad", "Central
            African CFA franc",
116             CurrencyConvertor.Currency.XAF),
117         new Tuple<string, string,
              CurrencyConvertor.Currency>("Chile", "Chilean
              peso",
118               CurrencyConvertor.Currency.CLP),
119         new Tuple<string, string,
              CurrencyConvertor.Currency>("China", "Chinese
              yuan",
120               CurrencyConvertor.Currency.CNY),
121         new Tuple<string, string,
            CurrencyConvertor.Currency>("Colombia",
```

```
                "Colombian peso",
122               CurrencyConvertor.Currency.COP),
123           new Tuple<string, string,
                CurrencyConvertor.Currency>("Comoros",
                "Comorian franc",
```

```
124             CurrencyConvertor.Currency.KMF),
125         new Tuple<string, string,
            CurrencyConvertor.Currency>("Costa Rica",
            "Costa Rican colón",
126             CurrencyConvertor.Currency.CRC),
127         new Tuple<string, string,
            CurrencyConvertor.Currency>("Croatia",
            "Croatian kuna",
128             CurrencyConvertor.Currency.HRK),
129         new Tuple<string, string,
            CurrencyConvertor.Currency>("Cuba", "Cuban
            peso", CurrencyConvertor.Currency.CUP),
130         new Tuple<string, string,
            CurrencyConvertor.Currency>("Cyprus",
            "Euro", CurrencyConvertor.Currency.EUR),
131         new Tuple<string, string,
            CurrencyConvertor.Currency>("Czech Republic",
            "Czech koruna",
132             CurrencyConvertor.Currency.CZK),
133         new Tuple<string, string,
            CurrencyConvertor.Currency>("Denmark",
            "Danish krone",
134             CurrencyConvertor.Currency.DKK),
135         new Tuple<string, string,
            CurrencyConvertor.Currency>("Djibouti",
            "Djiboutian franc",
136             CurrencyConvertor.Currency.DJF),
137         new Tuple<string, string,
            CurrencyConvertor.Currency>("Dominica",
            "East Caribbean dollar",
138             CurrencyConvertor.Currency.XCD),
139         new Tuple<string, string,
            CurrencyConvertor.Currency>("Dominican
```

```
            Republic", "Dominican peso",
140             CurrencyConvertor.Currency.DOP),
141         new Tuple<string, string,
            CurrencyConvertor.Currency>("East Timor",
            "United States dollar",
142             CurrencyConvertor.Currency.USD),
143         new Tuple<string, string,
            CurrencyConvertor.Currency>("Ecuador",
            "United States dollar",
144             CurrencyConvertor.Currency.USD),
145         new Tuple<string, string,
             CurrencyConvertor.Currency>("Egypt", "Egyptian
             pound",
146             CurrencyConvertor.Currency.EGP),
147         new Tuple<string, string,
            CurrencyConvertor.Currency>("El Salvador",
            "United States dollar",
148             CurrencyConvertor.Currency.USD),
149         new Tuple<string, string,
            CurrencyConvertor.Currency>("Equatorial
            Guinea", "Central African CFA franc",
150             CurrencyConvertor.Currency.XAF),
151         new Tuple<string, string,
            CurrencyConvertor.Currency>("Estonia",
            "Euro", CurrencyConvertor.Currency.EUR),
152         new Tuple<string, string,
            CurrencyConvertor.Currency>("Ethiopia",
            "Ethiopian birr",
153             CurrencyConvertor.Currency.ETB),
154         new Tuple<string, string,
            CurrencyConvertor.Currency>("Finland",
            "Euro", CurrencyConvertor.Currency.EUR),
155         new Tuple<string, string,
            CurrencyConvertor.Currency>("France",
```

```
            "Euro", CurrencyConvertor.Currency.EUR),
156         new Tuple<string, string,
            CurrencyConvertor.Currency>("Gabon", "Central
            African CFA franc",
157          CurrencyConvertor.Currency.XAF),
```

```
158             new Tuple<string, string,
                CurrencyConvertor.Currency>("Gambia",
                "Gambian dalasi",
159                 CurrencyConvertor.Currency.GMD),
160             new Tuple<string, string,
                CurrencyConvertor.Currency>("Germany",
                "Euro", CurrencyConvertor.Currency.EUR),
161             new Tuple<string, string,
                CurrencyConvertor.Currency>("Greece",
                "Euro", CurrencyConvertor.Currency.EUR),
162             new Tuple<string, string,
                CurrencyConvertor.Currency>("Grenada",
                "East Caribbean dollar",
163                 CurrencyConvertor.Currency.XCD),
164             new Tuple<string, string,
                CurrencyConvertor.Currency>("Guatemala",
                "Guatemalan quetzal",
165                 CurrencyConvertor.Currency.GTQ),
166             new Tuple<string, string,
                CurrencyConvertor.Currency>("Guinea",
                "Guinean franc",
167                 CurrencyConvertor.Currency.GNF),
168             new Tuple<string, string,
                CurrencyConvertor.Currency>("Guinea- Bissau",
                "West African CFA franc",
169                 CurrencyConvertor.Currency.XOF),
170             new Tuple<string, string,
                CurrencyConvertor.Currency>("Guyana",
                "Guyanese dollar",
171                 CurrencyConvertor.Currency.GYD),
172             new Tuple<string, string,
                 CurrencyConvertor.Currency>("Haiti", "Haitian
                 gourde",
```

```
173             CurrencyConvertor.Currency.HTG),
174         new Tuple<string, string,
            CurrencyConvertor.Currency>("Honduras",
            "Honduran lempira",
175             CurrencyConvertor.Currency.HNL),
176         new Tuple<string, string,
            CurrencyConvertor.Currency>("Hungary",
            "Hungarian forint",
177             CurrencyConvertor.Currency.HUF),
178         new Tuple<string, string,
            CurrencyConvertor.Currency>("Iceland",
            "Icelandic króna",
179             CurrencyConvertor.Currency.ISK),
180         new Tuple<string, string,
             CurrencyConvertor.Currency>("India", "Indian
             rupee",
181             CurrencyConvertor.Currency.INR),
182         new Tuple<string, string,
            CurrencyConvertor.Currency>("Indonesia",
            "Indonesian rupiah",
183             CurrencyConvertor.Currency.IDR),
184         new Tuple<string, string,
            CurrencyConvertor.Currency>("Iraq", "Iraqi
            dinar", CurrencyConvertor.Currency.IQD),
185         new Tuple<string, string,
            CurrencyConvertor.Currency>("Ireland",
            "Euro", CurrencyConvertor.Currency.EUR),
186         new Tuple<string, string,
            CurrencyConvertor.Currency>("Israel",
            "Israeli new shekel",
187             CurrencyConvertor.Currency.ILS),
188         new Tuple<string, string,
             CurrencyConvertor.Currency>("Italy", "Euro",
             CurrencyConvertor.Currency.EUR),
```

```
189             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Ivory Coast",
                  "West African CFA franc",
190                 CurrencyConvertor.Currency.XOF),
191             new Tuple<string, string,
                CurrencyConvertor.Currency>
```

```
            ("Jamaica", "Jamaican dollar",
192             CurrencyConvertor.Currency.JMD),
193         new Tuple<string, string,
            CurrencyConvertor.Currency>("Japan", "Japanese
            yen",
194             CurrencyConvertor.Currency.JPY),
195         new Tuple<string, string,
            CurrencyConvertor.Currency>("Jordan",
            "Jordanian dinar",
196             CurrencyConvertor.Currency.JOD),
197         new Tuple<string, string,
            CurrencyConvertor.Currency>
            ("Kazakhstan", "Kazakhstani tenge",
198             CurrencyConvertor.Currency.KZT),
199         new Tuple<string, string,
            CurrencyConvertor.Currency>("Kenya", "Kenyan
            shilling",
200             CurrencyConvertor.Currency.KES),
201         new Tuple<string, string,
            CurrencyConvertor.Currency>("Kiribati",
            "Australian dollar",
202             CurrencyConvertor.Currency.AUD),
203         new Tuple<string, string,
            CurrencyConvertor.Currency>("Korea, North",
            "North Korean won",
204             CurrencyConvertor.Currency.KPW),
205         new Tuple<string, string,
            CurrencyConvertor.Currency>("Korea, South",
            "South Korean won",
206             CurrencyConvertor.Currency.KRW),
207         new Tuple<string, string,
            CurrencyConvertor.Currency>("Kosovo",
            "Euro", CurrencyConvertor.Currency.EUR),
```

```
208             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Kuwait",
                  "Kuwaiti dinar",
209                 CurrencyConvertor.Currency.KWD),
210             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Laos", "Lao kip",
                  CurrencyConvertor.Currency.LAK),
211             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Latvia",
                  "Euro", CurrencyConvertor.Currency.EUR),
212             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Lebanon",
                  "Lebanese pound",
213                 CurrencyConvertor.Currency.LBP),
214             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Lesotho",
                  "Lesotho loti",
215                 CurrencyConvertor.Currency.LSL),
216             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Liberia",
                  "Liberian dollar",
217                 CurrencyConvertor.Currency.LRD),
218             new Tuple<string, string,
                   CurrencyConvertor.Currency>("Libya", "Libyan
                   dinar",
219                  CurrencyConvertor.Currency.LYD),
220             new Tuple<string, string,
                  CurrencyConvertor.Currency>
                  ("Liechtenstein", "Swiss franc",
221                 CurrencyConvertor.Currency.CHF),
222             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Lithuania",
                  "Euro", CurrencyConvertor.Currency.EUR),
223             new Tuple<string, string,
```

```
                CurrencyConvertor.Currency>("Luxembourg",
                "Euro", CurrencyConvertor.Currency.EUR),
224             new Tuple<string, string,
                CurrencyConvertor.Currency>("Macedonia",
                "Macedonian denar",
```

```
225             CurrencyConvertor.Currency.MKD),
226         new Tuple<string, string,
            CurrencyConvertor.Currency>("Malawi",
            "Malawian kwacha",
227             CurrencyConvertor.Currency.MWK),
228         new Tuple<string, string,
            CurrencyConvertor.Currency>("Malaysia",
            "Malaysian ringgit",
229             CurrencyConvertor.Currency.MYR),
230         new Tuple<string, string,
            CurrencyConvertor.Currency>("Maldives",
            "Maldivian rufiyaa",
231             CurrencyConvertor.Currency.MVR),
232         new Tuple<string, string,
            CurrencyConvertor.Currency>("Mali", "West
            African CFA franc",
233             CurrencyConvertor.Currency.XOF),
234         new Tuple<string, string,
             CurrencyConvertor.Currency>("Malta", "Euro",
             CurrencyConvertor.Currency.EUR),
235         new Tuple<string, string,
            CurrencyConvertor.Currency>("Marshall
            Islands", "United States dollar",
236             CurrencyConvertor.Currency.USD),
237         new Tuple<string, string,
            CurrencyConvertor.Currency>
            ("Mauritania", "Mauritanian ouguiya",
238             CurrencyConvertor.Currency.MRO),
239         new Tuple<string, string,
            CurrencyConvertor.Currency>("Mauritius",
            "Mauritian rupee",
240             CurrencyConvertor.Currency.MUR),
241         new Tuple<string, string,
```

```
                CurrencyConvertor.Currency>("Mexico",
                "Mexican peso",
242             CurrencyConvertor.Currency.MXN),
243           new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Micronesia", "United States dollar",
244             CurrencyConvertor.Currency.USD),
245           new Tuple<string, string,
                CurrencyConvertor.Currency>("Moldova",
                "Moldovan leu",
246             CurrencyConvertor.Currency.MDL),
247           new Tuple<string, string,
                CurrencyConvertor.Currency>("Monaco",
                "Euro", CurrencyConvertor.Currency.EUR),
248           new Tuple<string, string,
                CurrencyConvertor.Currency>("Mongolia",
                "Mongolian tögrög",
249             CurrencyConvertor.Currency.MNT),
250           new Tuple<string, string,
                CurrencyConvertor.Currency>("Montenegro",
                "Euro", CurrencyConvertor.Currency.EUR),
251           new Tuple<string, string,
                CurrencyConvertor.Currency>("Morocco",
                "Moroccan dirham",
252             CurrencyConvertor.Currency.MAD),
253           new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Mozambique", "Mozambican metical",
254             CurrencyConvertor.Currency.MZM),
255           new Tuple<string, string,
                CurrencyConvertor.Currency>("Myanmar",
                "Burmese kyat",
256             CurrencyConvertor.Currency.MMK),
257           new Tuple<string, string,
```

```
                CurrencyConvertor.Currency>("Namibia",
                "Namibian dollar",
258                 CurrencyConvertor.Currency.NAD),
259             new Tuple<string, string,
                CurrencyConvertor.Currency>("Nauru",
```

```
            "Australian dollar",
260             CurrencyConvertor.Currency.AUD),
261         new Tuple<string, string,
            CurrencyConvertor.Currency>("Nepal", "Nepalese
            rupee",
262             CurrencyConvertor.Currency.NPR),
263         new Tuple<string, string,
            CurrencyConvertor.Currency>
                                ("Netherlands", "Euro",
                        CurrencyConvertor.Currency.EUR),
264                         new Tuple<string, string,
                        CurrencyConvertor.Currency>("New
            Zealand", "New Zealand dollar",
265             CurrencyConvertor.Currency.NZD),
266         new Tuple<string, string,
            CurrencyConvertor.Currency>("Nicaragua",
            "Nicaraguan córdoba",
267             CurrencyConvertor.Currency.NIO),
268         new Tuple<string, string,
            CurrencyConvertor.Currency>("Niger", "West
            African CFA franc",
269             CurrencyConvertor.Currency.XOF),
270         new Tuple<string, string,
            CurrencyConvertor.Currency>("Nigeria",
            "Nigerian naira",
271             CurrencyConvertor.Currency.NGN),
272         new Tuple<string, string,
            CurrencyConvertor.Currency>("Norway",
            "Norwegian krone",
273             CurrencyConvertor.Currency.NOK),
274         new Tuple<string, string,
            CurrencyConvertor.Currency>("Oman", "Omani
            rial", CurrencyConvertor.Currency.OMR),
```

```
275             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Pakistan",
                  "Pakistani rupee",
276               CurrencyConvertor.Currency.PKR),
277             new Tuple<string, string,
                   CurrencyConvertor.Currency>("Palau", "United
                   States dollar",
278               CurrencyConvertor.Currency.USD),
279             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Palestine",
                  "Israeli new shekel",
280               CurrencyConvertor.Currency.ILS),
281             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Panama",
                  "Panamanian balboa",
282               CurrencyConvertor.Currency.PAB),
283             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Papua New Guinea",
                  "Papua New Guinean kina",
284                 CurrencyConvertor.Currency.PGK),
285             new Tuple<string, string,
                CurrencyConvertor.Currency>
                  ("Paraguay", "Paraguayan guaraní",
286                 CurrencyConvertor.Currency.PYG),
287             new Tuple<string, string,
                CurrencyConvertor.Currency>("Peru",
                  "Peruvian sol",
                  CurrencyConvertor.Currency.PEN),
288             new Tuple<string, string,
                CurrencyConvertor.Currency>
                  ("Philippines", "Philippine peso",
289                 CurrencyConvertor.Currency.PHP),
290             new Tuple<string, string,
                CurrencyConvertor.Currency>
```

```
                ("Poland", "Polish zloty",
291             CurrencyConvertor.Currency.PLN),
292         new Tuple<string, string,
                CurrencyConvertor.Currency>("Portugal",
                "Euro", CurrencyConvertor.Currency.EUR),
293         new Tuple<string, string,
            CurrencyConvertor.Currency>("Qatar",
```

```
                "Qatari riyal",
294             CurrencyConvertor.Currency.QAR),
295           new Tuple<string, string,
                CurrencyConvertor.Currency>("Republic of
                the Congo", "Central African CFA franc",
296             CurrencyConvertor.Currency.XAF),
297           new Tuple<string, string,
                CurrencyConvertor.Currency>("Romania",
                "Romanian leu",
298             CurrencyConvertor.Currency.ROL),
299           new Tuple<string, string,
                CurrencyConvertor.Currency>("Russia",
                "Russian ruble",
300             CurrencyConvertor.Currency.RUB),
301           new Tuple<string, string,
                CurrencyConvertor.Currency>("Saint Kitts and
                Nevis", "East Caribbean dollar",
302             CurrencyConvertor.Currency.XCD),
303           new Tuple<string, string,
                CurrencyConvertor.Currency>("Saint Lucia",
                "East Caribbean dollar",
304             CurrencyConvertor.Currency.XCD),
305           new Tuple<string, string,
                CurrencyConvertor.Currency>("Saint Vincent and
                the Grenadines",
306               "East Caribbean dollar",
                  CurrencyConvertor.Currency.XCD),
307           new Tuple<string, string,
              CurrencyConvertor.Currency>("Samoa",
                "Samoan tala", CurrencyConvertor.Currency.WST),
308           new Tuple<string, string,
              CurrencyConvertor.Currency>("San
                Marino", "Euro", CurrencyConvertor.Currency.EUR),
```

```
309             new Tuple<string, string,
                CurrencyConvertor.Currency>("São Tomé and
                Príncipe", "São Tomé and Príncipe dobra",
310               CurrencyConvertor.Currency.STD),
311             new Tuple<string, string,
                CurrencyConvertor.Currency>("Saudi Arabia",
                "Saudi riyal",
312               CurrencyConvertor.Currency.SAR),
313             new Tuple<string, string,
                CurrencyConvertor.Currency>("Senegal",
                "West African CFA franc",
314               CurrencyConvertor.Currency.XOF),
315             new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Seychelles", "Seychellois rupee",
316               CurrencyConvertor.Currency.SCR),
317             new Tuple<string, string,
                CurrencyConvertor.Currency>("Sierra Leone",
                "Sierra Leonean leone",
318               CurrencyConvertor.Currency.SLL),
319             new Tuple<string, string,
                CurrencyConvertor.Currency>("Singapore",
                "Singapore dollar",
320               CurrencyConvertor.Currency.SGD),
321             new Tuple<string, string,
                CurrencyConvertor.Currency>("Slovakia",
                "Euro", CurrencyConvertor.Currency.EUR),
322             new Tuple<string, string,
                CurrencyConvertor.Currency>("Slovenia",
                "Euro", CurrencyConvertor.Currency.EUR),
323             new Tuple<string, string,
                 CurrencyConvertor.Currency>("Solomon Islands",
                 "Solomon Islands dollar",
324               CurrencyConvertor.Currency.SBD),
```

```
325             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Somalia",
                  "Somali shilling",
326                 CurrencyConvertor.Currency.SOS),
327             new Tuple<string, string,
                  CurrencyConvertor.Currency>("South
```

```
                Africa", "South African rand",
328                 CurrencyConvertor.Currency.ZAR),
329             new Tuple<string, string,
                    CurrencyConvertor.Currency>("Spain", "Euro",
                    CurrencyConvertor.Currency.EUR),
330             new Tuple<string, string,
                CurrencyConvertor.Currency>("Sri Lanka", "Sri
                Lankan rupee",
331                 CurrencyConvertor.Currency.LKR),
332             new Tuple<string, string,
                    CurrencyConvertor.Currency>("Sudan", "Sudanese
                    pound",
333                 CurrencyConvertor.Currency.SDD),
334             new Tuple<string, string,
                CurrencyConvertor.Currency> ("Swaziland",
                "Swazi lilangeni",
335                 CurrencyConvertor.Currency.SZL),
336             new Tuple<string, string,
                CurrencyConvertor.Currency> ("Sweden",
                "Swedish krona",
337                 CurrencyConvertor.Currency.SEK),
338             new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Switzerland", "Swiss franc",
339                 CurrencyConvertor.Currency.CHF),
340             new Tuple<string, string,
                    CurrencyConvertor.Currency>("Syria", "Syrian
                    pound",
341                 CurrencyConvertor.Currency.SYP),
342             new Tuple<string, string,
                CurrencyConvertor.Currency> ("Taiwan",
                "New Taiwan dollar",
343                 CurrencyConvertor.Currency.TWD),
```

```
344             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Tanzania",
                  "Tanzanian shilling",
345                 CurrencyConvertor.Currency.TZS),
346             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Thailand",
                  "Thai baht",
347                 CurrencyConvertor.Currency.THB),
348             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Togo", "West
                  African CFA franc",
349                 CurrencyConvertor.Currency.XOF),
350             new Tuple<string, string,
                   CurrencyConvertor.Currency>("Tonga", "Tongan
                   pa'anga",
351                 CurrencyConvertor.Currency.TOP),
352             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Trinidad and
                  Tobago", "Trinidad and Tobago dollar",
353                 CurrencyConvertor.Currency.TTD),
354             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Tunisia",
                  "Tunisian dinar",
355                 CurrencyConvertor.Currency.TND),
356             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Turkey",
                  "Turkish lira",
357                 CurrencyConvertor.Currency.TRY),
358             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Tuvalu",
                  "Australian dollar",
359                 CurrencyConvertor.Currency.AUD),
360             new Tuple<string, string,
                  CurrencyConvertor.Currency>("Uganda",
```

```
                "Ugandan shilling",
361               CurrencyConvertor.Currency.UGX),
362           new Tuple<string, string,
                CurrencyConvertor.Currency>
```

```
                ("Ukraine", "Ukrainian hryvnia",
363               CurrencyConvertor.Currency.UAH),
364             new Tuple<string, string,
                CurrencyConvertor.Currency>("United Arab
                Emirates", "United Arab Emirates dirham",
365               CurrencyConvertor.Currency.AED),
366             new Tuple<string, string,
                CurrencyConvertor.Currency>("United Kingdom",
                "British pound",
367               CurrencyConvertor.Currency.GBP),
368             new Tuple<string, string,
                CurrencyConvertor.Currency>("United States",
                "United States dollar",
369               CurrencyConvertor.Currency.USD),
370             new Tuple<string, string,
                CurrencyConvertor.Currency> ("Uruguay",
                "Uruguayan peso",
371               CurrencyConvertor.Currency.UYU),
372             new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Vanuatu", "Vanuatu vatu",
373               CurrencyConvertor.Currency.VUV),
374                                   new Tuple<string, string,
                              CurrencyConvertor.Currency>("Vatican
                 City", "Euro", CurrencyConvertor.Currency.EUR),
375             new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Venezuela", "Venezuelan bolívar",
376               CurrencyConvertor.Currency.VEB),
377             new Tuple<string, string,
                CurrencyConvertor.Currency>
                ("Vietnam", "Vietnamese dong",
```

```
378                 CurrencyConvertor.Currency.VND),
379                                         new Tuple<string, string,
                                    CurrencyConvertor.Currency>("Yemen",
                      "Yemeni rial", CurrencyConvertor.Currency.YER),
380             new Tuple<string, string,
                CurrencyConvertor.Currency>
                  ("Zambia", "Zambian kwacha",
381                 CurrencyConvertor.Currency.ZMK),
382             new Tuple<string, string,
                CurrencyConvertor.Currency>
                  ("Zimbabwe", "United States dollar",
383                 CurrencyConvertor.Currency.USD),
384         };
385
386         [WebMethod]
387         public double CalculateExchangeRateFromService(double
                value,
                  string country)
388         {
389             // guidence from
                    https://www.codeproject.com/Articles/15483/
                    Currency-Conversion-Using-Web-Services
390             // guidence from
                    https://forums.asp.net/t/1297305.aspx?
                    Currency+Conversion+Web+Service
391             // guidence from
                    https://stackoverflow.com/questions/2220158/
                    how-to-get-the-currency-rate-from-the-web-
                    service as this uses vb but can be adpated to c
                    sharp
392             // guidence from
                    https://www.youtube.com/watch?v=ItATfVSz9AI
                    which is created in c sharp but can be adpated
393
```

```
394             if (CountriesAndCurrencies.Any(t => t.Item1 ==
                country))
395             {
396                 var currencyServiceSoapClient = new
                    CurrencyConvertorSoapClient();
```

```
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413 }
```

```
            var rate =
              currencyServiceSoapClient.ConversionRate
              (CurrencyConvertor.Currency.USD,
                CountriesAndCurrencies.Single(t =>
               t.Item1 == country).Item3);

            return value*rate;
        }
        return -1;

    }

    [WebMethod]
    public string[] Countries()
    {
        return CountriesAndCurrencies.Select(t =>
        t.Item1).ToArray();
    }
}
```