# COMP1687 WEB APPLICATION DEVELOPMENT

USMAN BASHARAT 000874782

# Table of Contents

## Introduction

In this report, I will be writing a report upon the website application that I have developed. I will be including a self-assessment sheet, a statement of functionality, any bugs that is necessary, reflection of where strengths and weaknesses within the application, a brief design documentation, and the screenshots of programs in operation.

### 1. Self-Assessment Sheet

**COMP1687 Self-Assessment Sheet for the 201718 Coursework**

**This sheet must be completed and submitted with your report**

**Student name:** Usman Basharat **Student:** ID 000874782

URL: `https://stuweb.cms.gre.ac.uk/~ub2232e/three/register.php`

| **Student Use** | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Level 1 | Account creation | 18 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | **10** |
| Level 2 | Verify account | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | **10** |
| Level 3 | Authentication | 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | **10** |
| Level 4 | Post | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 |
| Level 5 | Image upload | 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 |
| Level 6 | Search | 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | **7** | 8 | 9 | 10 |
| Level 7 | Cookie | 6 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 |
| Level 8 | Report | 16 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 |
| **Staff Use** | | | | | | | | | | | | | |
| Self-Assessment | | 6 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Comments | | | | | | | | | | | | | |

## 2. Statement of functionality

*This section is used to explain what levels I have completed.*

### Level 1: Account Creation

This level was required to create a member's account for visitors. This was complete by adding a CAPTCHA, email address, username and password. As specified by the specification, the register page prevents any duplication of username. The email is also verified. If the user enters an email that is not recognised, it checks it and lets the user know if it is wrong or not. Also, the email address provided by the user, it sends a code to verify the email address. Once the CAPTCHA, email address and username validations are correct; the user gets send to the email verification page where the user verifies the email. The successful register gets put into the database. All explanation and screenshots are below.

### Level 2: Verify Account

**Mail Function**

Once the email gets send to their email, the user uses the code provided and enters the code correctly; they get send straight to the main page of the website. They do not need to authenticate again. The verification account also checks if the code that has been entered is correct or not. If this is not correct, it sends a user a message to try again. The account is remained inactive until this has been verified. Also, the user cannot spam the input. The text box has a maximum of 5 and it limits it.

**Use Case 1**

Returning members whom have not completed their verification of their account cannot log in. Instead, the system detects this and sends another email and lets the user know to verify the email again. Once this is complete, the user would be verified and can log in again. Once the user has logged in, the system deletes the session once the user has logged out.

**Use Case 2**

When users whom have not validated been validated, the user gets redirected; the validation is if the user gets the wrong password, it would show them the message saying enter the correct password. Once entering the correct code, it lets the user onto the home page without authenticating.

### Level 3: Authentication

Having a HTTPS used to add security is important. Once the user enters the URL; it automatically gets the HTTPS to secure. Also, if I was to copy and paste the URL into another browser for the home page, it would automatically get the user to the login page where they need to authenticate. Once the user gets the wrong password, it sends them a message. Also, its successful and directs the user to the home page once successful.

### Level 4: Members Post

Once the users have logged in, they can check their posts and choose to edit, or delete the posts that they have posted. The posts allow the users to post more than one post. Editing one post can enable the user to change what is necessary. Deleting the post allows the users to delete the selected post with a pop-up to make sure they would want to delete the post.  Adding the post is also available and is verified. Selected posts have necessary validations that the user must go through to add a post successfully. These validations are all named at the rest of the report.

### Level 5: Image Upload

The image upload allows the user to upload an image of their selection. Validation has been done for this to upload a successful image. Only images can be detected once uploading an image. For example, if I were to upload a document, it would not let the user. The user has a choice of whether to upload an image or not when adding a post. The user has a choice of whether to delete the selected image, or replace the image that they want. Deleting the image gets deleted from the post straight away and gets deleted from the database too.

### Level 6: Member Search

Unauthorised members can search through content. These contents can filter through the closest possible search. For example, if I were to type in “Po”, it would search up by looking at the database of those who have “Po” in t and display it. Any unauthorised users who wish to view more details of the selected item; it would check if they have logged in or not. If they have not logged in; they would be referred to the log in page. Once they are referred, they can view the post that they wanted. Any posts that have no images are not displayed. Only images that have posts are available.

### Level 7: Cookies

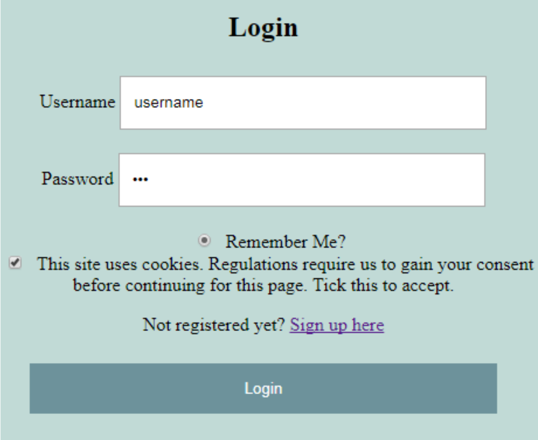
Cookies are enabled at this stage. One cookie is used to remember the username only. In addition, another 4 cookies are used for the last search item that the users have searched. Cookie law has been followed by users have to accept the condition before logging in. If they have not accepted, the cookie does not get used.

## 3. Bugs

*All software has bugs. For my fault within my application, when the user accepts the cookie for the login, or wants to search an item; the user must keep accepting the cookie if they want to update the cookie.*

### 3.1. Cookie Bug

Referring to Figure 1, it demonstrates the user must accept both of the ‘Remember Me?’ and the cookie law in order for the cookie to go through. If one only one of them the user accepts, the cookie would not go through. Referring to Figure 2, it shows that the cookie works.



Login

Username: username

Password: •••

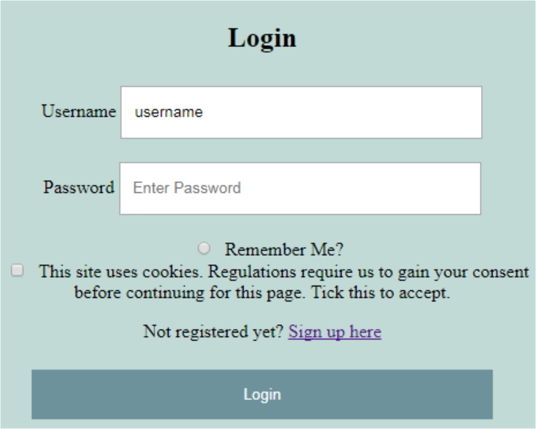
● Remember Me?

☑ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

Login

*Figure 1 shows the user accepting the cookie law and remember me.*



Login

Username: username

Password: Enter Password

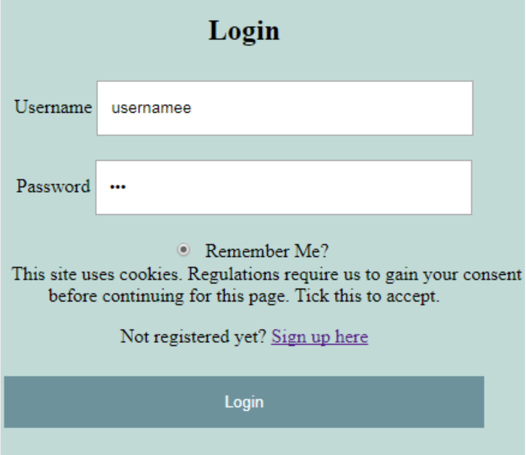
○ Remember Me?

□ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

Login

*Figure 2 shows the cookie being accepted.*



Login

Username: usernamee

Password: •••

● Remember Me?

This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

Login

*Figure 3 shows the details entered.*

Referring to Figure 3, this shows the fault of why this is a bug. If the user does not accept both ‘Remember Me’ and cookie law; the cookie would not work.

Once the user logs in, and later decides to log out; it would show Figure 2 as it has not accepted both of the cookie law and remember me. As specified within the specification, it says to do save the cookie for the last search for it. This is the same bug for the others.

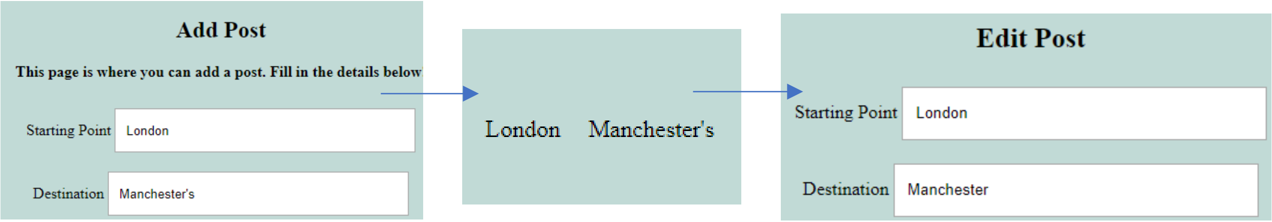
### 3.2. Search Error

Another error that I have within my program is when users chooses to search on the home page. The three choices that the user must fill in. They are starting point, destination and date with time. These choices must be filled in to view a post from another user that has posted it. They can only view the posts that have images only. The user can filter through between starting point and destination. However, the date and time, it does not search. I wanted it so that it searches through the date that they have searched, but the time is not necessary. However, both of these do not work.

One way to solve this bug is to get one FROM date and TO date. And, filter these dates within these dates that the user has selected. For future references, both bugs that have been mentioned would be completed in a different way.

### 3.3. Apostrophe Error

Another is that when the apostrophe has gone in within the database. It adds to the database, and it can be viewed within the database. However, when I click 'Edit Post'; it does not show the apostrophe that is added within the database. The steps are shown from when I add a post, to viewing it and to editing the post. This should not happen when the apostrophe has gone in. This is because once the user has clicked submit in edit, it would end up as the current Manchester and it would make it longer for the user. This would make it longer for the user as they would need to type it in again, when this bug can be fixed easily.



**Add Post**

This page is where you can add a post. Fill in the details below

Starting Point: London

Destination: Manchester's

London Manchester's

**Edit Post**

Starting Point: London

Destination: Manchester

## 4. Reflection of weakness and strength of program

*For this section, I will discuss do a reflection that includes the strengths and weaknesses within this. In addition, I will be including an acceptance testing and I will be evaluating the program.*

### 4.1. Reflection

#### 4.1.1. Strengths

One of the strengths for my application is validation throughout the application. I feel that the validations for when it is required for the user to enter such text; it shows a message for anything that has gone wrong. For example, when a user is registering an account; the application makes sure these validations are correct. If a user enters a wrong type of email, such as '*a@s*', it would recognise that as an invalid email address. These validations are important to recognise the user enters the correct details. Another strength that I feel is necessary is the security of the application. When the user types in the URL for the website, the application makes the website secure. Having the site not secure would potentially make the data that has been inserted insecure. Any site that has not secure within the URL bar would indicate to the users that the data that has been used is insecure.

Another strength that I feel is necessary is that each account is different to each other. This means that when user A uploads one post. User B logs in and can only search through it. The user B cannot edit and delete the post from user B. This is necessary, because the user who uploads the post has the choice to do this. Other users cannot have the authority of editing and deleting.

#### 4.1.1. Weaknesses

One of the weaknesses that I found for my application is that the user has a choice to delete and replace image within the application. However, the user cannot upload more than one image for it. This has been specified within the specification. This weakness could have been done by adding another database for the image and linking the two using another ID for each one. So, to identify who added the images, you can check the id alongside each other.

Another weakness that I have found within my application is that if many users added posts, and a user wants to see all the posts; a logical view is that there will be a limit by adding a paginated with it. For this, I do not have a paginated list for the search post. These weaknesses within the application that I have found. For future improvements, these weaknesses would be considered and added. In addition, the bugs that have been mentioned above will also be considered a different way soon.

## 4.2. Testing

*I have done group of testing for Acceptance testing below. All screenshots of the program in operation is below.*

### 4.2.1. Acceptance Testing

| Test No. Test | Description | Expected Results | Actual Result | Actions |
|---|---|---|---|---|
| 1. | Successful registration | It is expected to put the correct details for the registration and it gets put into the database | The user puts the correct validations and unique username and it gets put into the database. | No action needed. |
| 2. | Validation for email, username, and CAPTCHA. | It is expected for the user to test the registration and put in the wrong details. | The user puts in a duplicate username, a wrong email, and a wrong CAPTCHA code. Each validation gets a message for each one. | No action needed. |
| 3. | Email code sent | Once the user gets send, a code is expected to get sent to the email they have entered. | As expected, the user receives the code from the email that they have typed in. | No action needed. |
| 4. | Email Validation | It is expected for the user to enter a wrong code. | As expected, the user enters the wrong code and a message appears for the user to enter the correct one. | No action needed. |
| 5. | Successful Account | It is expected to enter the correct code and the user to enter the home page straight away. The user does not need to authenticate. | As expected, the user enters the correct code that is sent to the email. And, the user gets referred straight to the home page without authenticated. | No action needed. |
| 6. | Unauthorised login account | It is expected if an unauthenticated account tries to log in, they get referred to the email verification page. | As expected, the user gets referred to the email verification. This is because their account is not verified. | No action needed. |
| 7. | Authorised login | The user is expected to enter correct details. Once they enter the correct details and they got referred to the login page. | Once the user is successful, they enter the correct details and they got referred to the login page. | No action needed. |

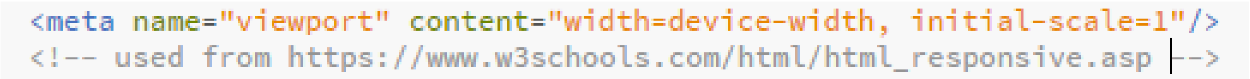
| | | | | |
|---|---|---|---|---|
| 8. | Wrong details for login | It is expected to enter wrong details and a message appears to notify user. | The user entered the wrong details and a message appeared saying to enter the right details | No action needed. |
| 9. | HTTPs used? | It is expected for the site to be secure as soon as the URL is entered. | The HTTPS is used, and the site is secure as soon as the URL has been entered for any browser. | No action needed. |
| 9. | Password encryption | It is expected for the password to be encrypted as soon as the user registers in the database. | The user entered the details and the password entered is encrypted and secure to prevent users knowing each other's password. | No action needed. |
| 10. | Redirect to login page | It is expected for unauthorised users trying to get access for the home page is to be redirected straight away. | The unauthorised user tried to get access to the home page and the application refers the user to the login page straight away. | No action needed. |
| 11. | Adding post | It is expected for the user to add a successful post as all the right details are entered. | All details were correct, and the post was successful. It was successful as it was added to the database and shown on the home page. | No action needed. |
| 12. | Having a choice to upload image or not. | The user is expected to add to posts. Once with image and one without an image. | As expected, the user has a choice to upload an image or not. Both posts were uploaded and successful. | No action needed. |
| 13. | Editing post | Expecting the user to edit the post they have just upload. | As expected, all the details appeared, and they edited it successful. It was edited in the database too. | No action needed. |
| 14. | Deleting post | Expecting the user to delete the post they have just edited and posted. | The user deletes the post. Once the delete link has been clicked, an alert message has appeared for the user to make sure they want to delete the image. It was deleted in the database too. | No action needed. |
| 15. | Replacing image and delete image. | The user has a choice to replace the image. In addition, the user wants to delete the image too. | The user uploads the replacing image and it works. The replacing image is shown. Later, the user deletes the image and it deletes the image straight away. | No action needed. |

|  |  |  |  |  |
|---|---|---|---|---|
| 16. | Unauthorised user trying to search | The user is expected to search for an item. | As expected, the user searches for an item straight away and the correct details appear. | No action needed. |
| 17. | Search with image only | The user is expected to search items that have image only. | The user searches for post with the image and it shows the search straight away. The user tries to do one without image and it does not show. | No action needed. |
| 18. | Cookies | The user is expected to choose to accept the cookies and it remembers it as soon as the user logs out. | The user accepts the cookies and it shows the same username once the user has logged out. | No action needed. |

### 4.3. Evaluation

Overall, I felt for the course, it has taught me well in terms all the levels that have been set and what to do for each. The outcome of the application is for authorised users to be able to post their journeys and others can view it. From Level 1 to Level 3, I felt that the authentication, account creation and login authorisation all works perfectly. However, a few additions that has been mentioned within the weaknesses could be added on to make the website much more better functionality. In addition, the bugs that have been mentioned could be strengthened and it could it better.  These weaknesses and bugs are for future references and are known to me to fix for it.

For the use of tools, I have made sure that my application works on each of the browsers. These are to make sure that different users have different preferences. For example, this code below shows the website on any mobile devices according to the screen.  However, for design purposes, I have only included this code for the navigation, login and register page. Once the user has logged in, they can view it more of a desktop view.



```
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-- used from https://www.w3schools.com/html/html_responsive.asp -->
```

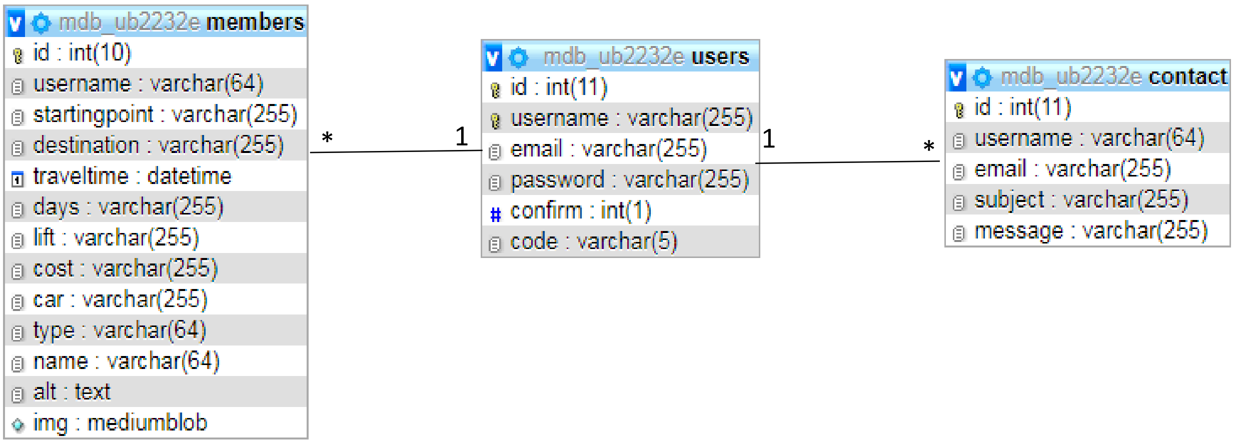
For JavaScript, if a user was to disable JavaScript; the only part that uses JavaScript for when an unauthorised user wants to view more details for search. This is the only part that would be disabled. This is because some devices have JavaScript disabled and some prefer it to be off. The website would adapt it so that it would not work only for the search preference.

## 5. Brief Design Documentation

*Below is a brief design documentation that includes diagrammatic schema for the database, list of all files and supporting UML that includes use-case diagrams for login.*

### 5.1. Diagrammatic schema for the database

This is a diagrammatic schema for the database that is shown below. Below shows a representation of my database of how each table is a relationship with each other. Users to members shows a relationship of one to many. This shows that one user can post many members. This is the same between the relationship from Users to Contact.



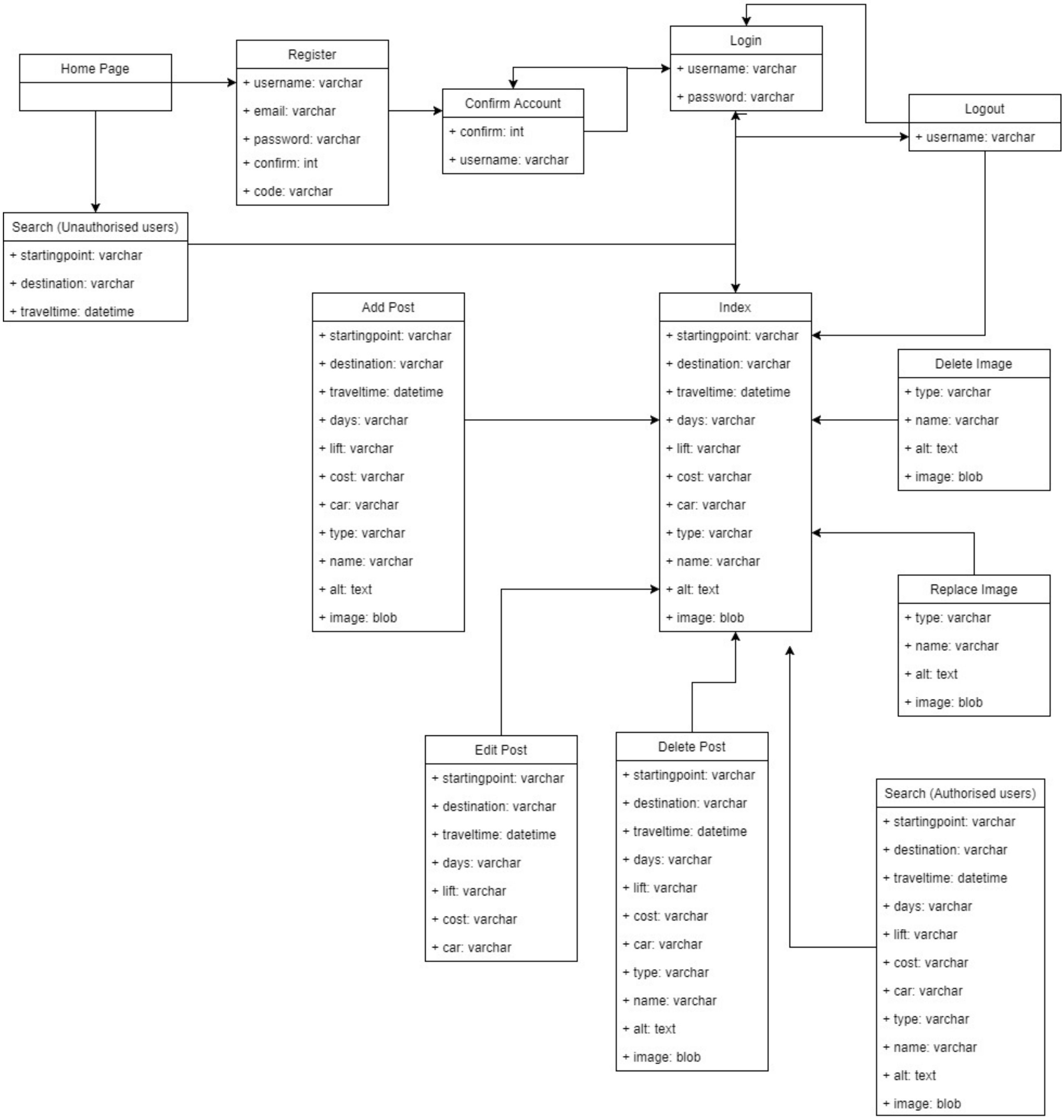
mdb_ub2232e **members**
- id : int(10)
- username : varchar(64)
- startingpoint : varchar(255)
- destination : varchar(255)
- traveltime : datetime
- days : varchar(255)
- lift : varchar(255)
- cost : varchar(255)
- car : varchar(255)
- type : varchar(64)
- name : varchar(64)
- alt : text
- img : mediumblob

mdb_ub2232e **users**
- id : int(11)
- username : varchar(255)
- email : varchar(255)
- password : varchar(255)
- confirm : int(1)
- code : varchar(5)

mdb_ub2232e **contact**
- id : int(11)
- username : varchar(64)
- email : varchar(255)
- subject : varchar(255)
- message : varchar(255)

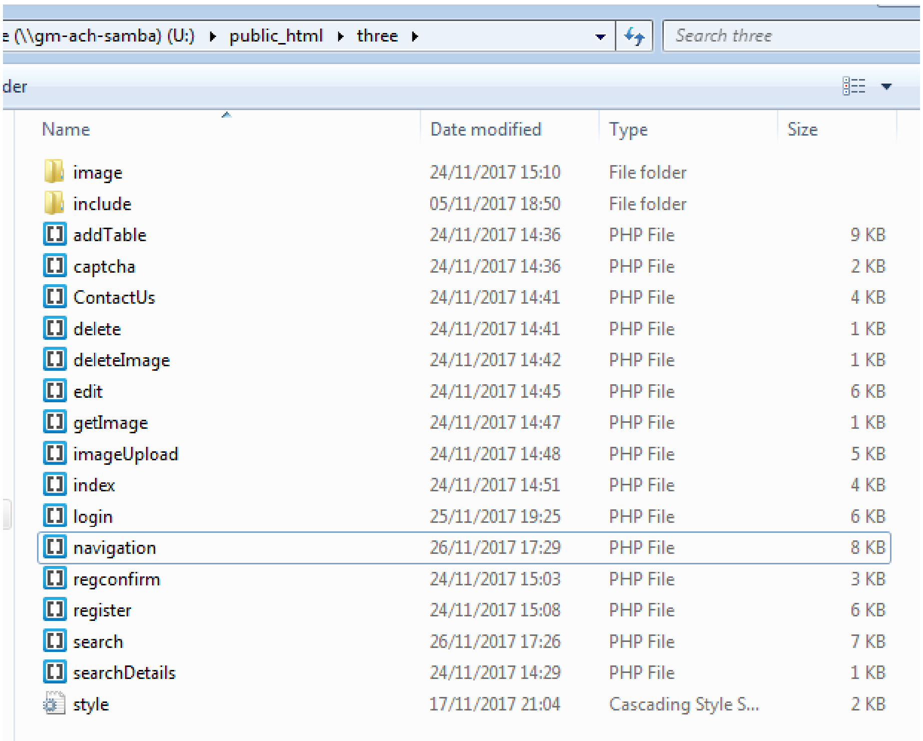
### 5.1.1. Database Relationships

These are the relationship of what is used from the database within each class that I have used. The diagram shows the relationship of what is used, what is included and what type of variable it is.



## 5.2. List of all files

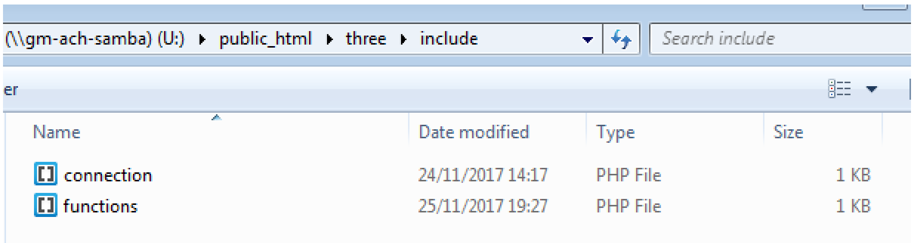
*These are all the files that have been used for my application.*



| Name | Date modified | Type | Size |
|---|---|---|---|
| image | 24/11/2017 15:10 | File folder | |
| include | 05/11/2017 18:50 | File folder | |
| addTable | 24/11/2017 14:36 | PHP File | 9 KB |
| captcha | 24/11/2017 14:36 | PHP File | 2 KB |
| ContactUs | 24/11/2017 14:41 | PHP File | 4 KB |
| delete | 24/11/2017 14:41 | PHP File | 1 KB |
| deleteImage | 24/11/2017 14:42 | PHP File | 1 KB |
| edit | 24/11/2017 14:45 | PHP File | 6 KB |
| getImage | 24/11/2017 14:47 | PHP File | 1 KB |
| imageUpload | 24/11/2017 14:48 | PHP File | 5 KB |
| index | 24/11/2017 14:51 | PHP File | 4 KB |
| login | 25/11/2017 19:25 | PHP File | 6 KB |
| navigation | 26/11/2017 17:29 | PHP File | 8 KB |
| regconfirm | 24/11/2017 15:03 | PHP File | 3 KB |
| register | 24/11/2017 15:08 | PHP File | 6 KB |
| search | 26/11/2017 17:26 | PHP File | 7 KB |
| searchDetails | 24/11/2017 14:29 | PHP File | 1 KB |
| style | 17/11/2017 21:04 | Cascading Style S... | 2 KB |

*Figure 4 shows the list of all my files for my application.*

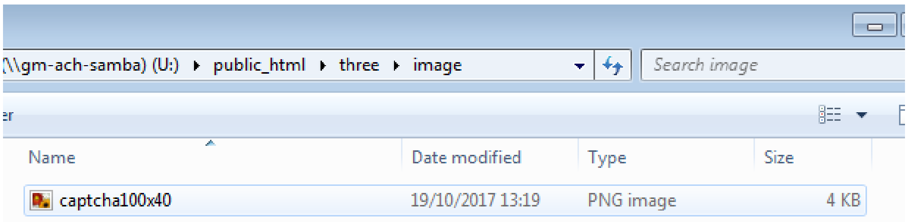
Referring to Figure 6, these are the files for my application. AddTable.php is to add a post. Captcha.php is to generate the code and add the background used within the 'image' folder. ContactUs.php is a folder for users to be able to enquire about anything to the admin. Delete.php is to delete a post. DeleteImage.php is to delete an image. Edit.php is to edit a post. GetImage.php is to get the image from the database. ImageUpload.php is for the user to be able to replace an existing image. Login.php is for the users to be able to login. Navigation is the home page for unauthorised users alongside search. Register.php is for the users to be able to register for the application. Search.php is for the authorised users to be able to search. Searchdetails.php is used within navigation.php to refer unauthorised users to the login page and authorised users to the search page where they use a cookie to be able to view the whole post.



| Name | Date modified | Type | Size |
|---|---|---|---|
| connection | 24/11/2017 14:17 | PHP File | 1 KB |
| functions | 25/11/2017 19:27 | PHP File | 1 KB |

*Figure 5 shows the files within the 'include'.*

Referring to Figure 7, it shows the connection.php and functions.php. These two are used for every class within Figure 6. The connection is for the database to connect between the website. The functions.php only contains session_start() and error_reporting(E_ERROR).



| Name | Date modified | Type | Size |
|---|---|---|---|
| captcha100x40 | 19/10/2017 13:19 | PNG image | 4 KB |

*Figure 6 shows the files within 'image'.*

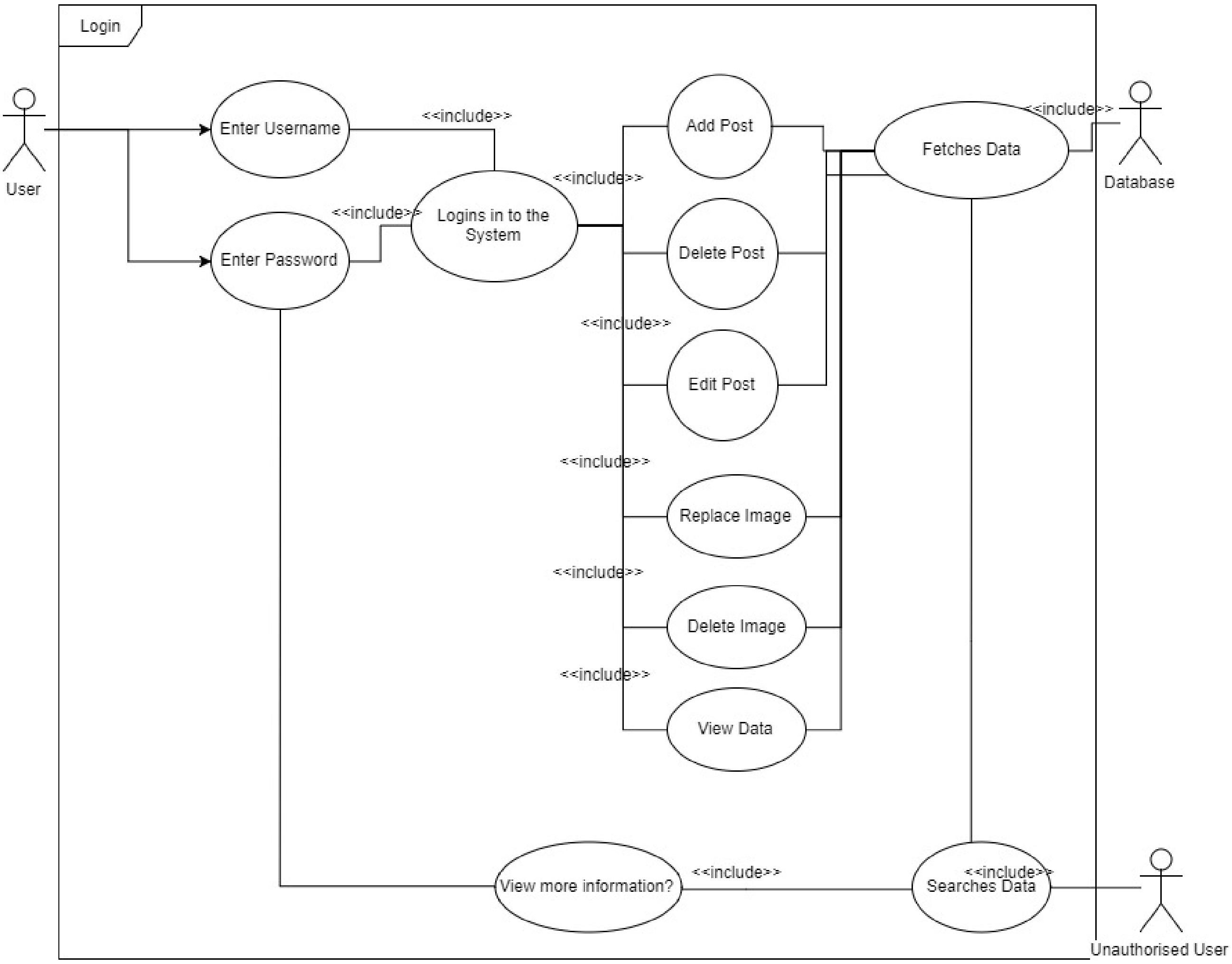
Referring to Figure 8, it shows the background image of the CAPTCHA. This is used within the register.php. The background helps the CAPTCHA for humans to detect the code. The background gives a rough idea of how the code would look like.

## 5.3. Supporting UML

*Below, I am going be showing a use-case diagram that represents how the login works.*

### 5.3.1. Verification Page Account Use-case Diagram

*This is a use-case diagram that shows the verification of login. The diagram shows once the user has logged in, the system checks the authorised user can add, delete, edit, view the post. And, they can delete, and replace image too. However, the unauthorised user can search through the data and if they want to view more information, they can go back to login. All this data is coming from the database and this is all viewed when the authorised user is logged in.*

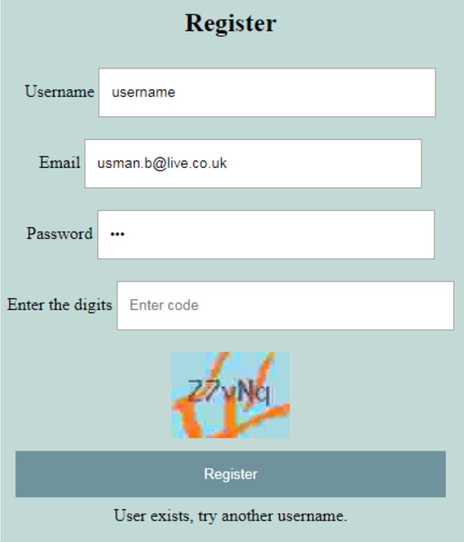


## 6. Screenshots of programs in operation

*The screenshots below show the program being in operation.*

## 6.1. Register

### 6.1.1. Register Validation



**Register**

Username: username

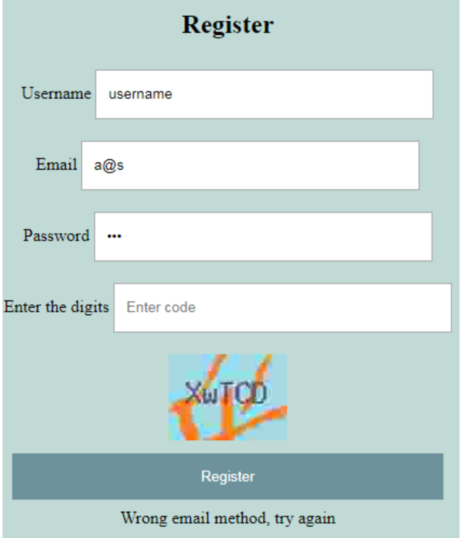
Email: usman.b@live.co.uk

Password: •••

Enter the digits: Enter code

User exists, try another username.

*Figure 8 shows the validation for an existing username*



**Register**

Username: username

Email: a@s

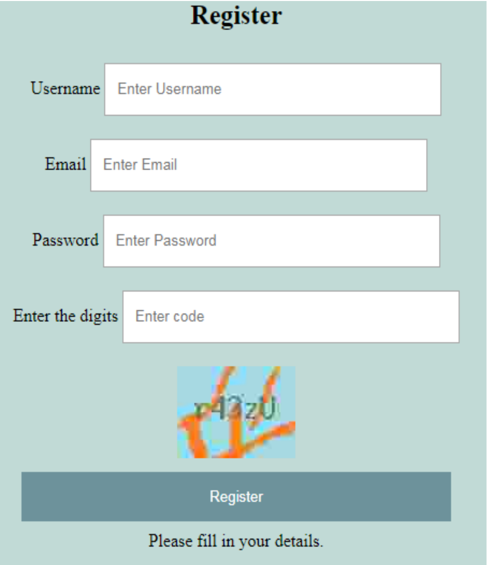
Password: •••

Enter the digits: Enter code

Wrong email method, try again

*Figure 9 shows the validation of the email address.*

Referring to Figure 10, this shows the validation for duplicate username. If another user has the same username as the user trying to register, it will show the following message as displayed. This will prevent other users to trying to register the same username as another one that has been used. Once this message has been appeared, the user must think of another one that has not been used.



**Register**

Username: Enter Username

Email: Enter Email

Password: Enter Password

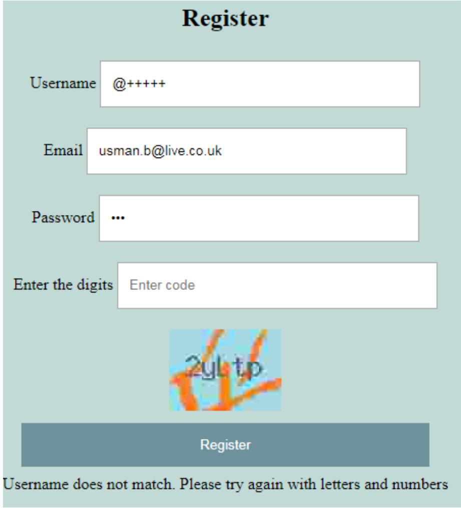
Enter the digits: Enter code

Please fill in your details.

*Figure 7 shows the validation for the empty details.*

Referring to Figure 11, it shows the validation of the email address. As you can see, the user has typed an incorrect email address. This is it is not regarded as an email address. Therefore, the application has detected this and has shown the message to try again.

Referring to Figure 9, this is the same process. As you can see, the user has left all the details empty. Once the user has clicked the button, the following message. Therefore, the user cannot continue until the user has filled the details with the correct validation.



Register

Username @+++++

Email usman.b@live.co.uk

Password •••

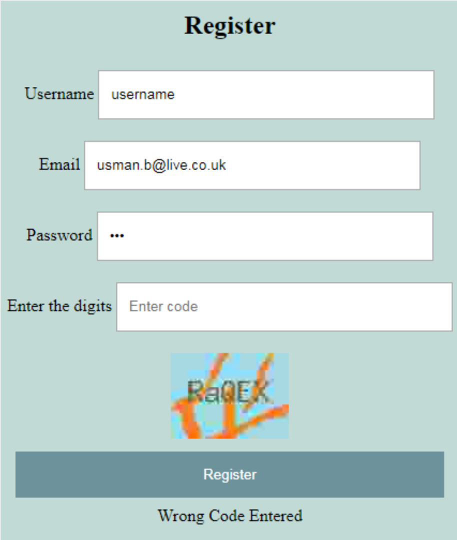
Enter the digits Enter code

Register

Username does not match. Please try again with letters and numbers

*Figure 11 shows the validation of entering the username without numbers or letters.*

Referring to Figure 13, it shows the same validation as mentioned above. The user has entered an incorrect username that has not been recognised. The application has detected this and has recognised the error. Therefore, the message has been placed. The username needs to be without letters and numbers to follow the correct way of the username.



Register

Username username

Email usman.b@live.co.uk

Password •••

Enter the digits Enter code

Register

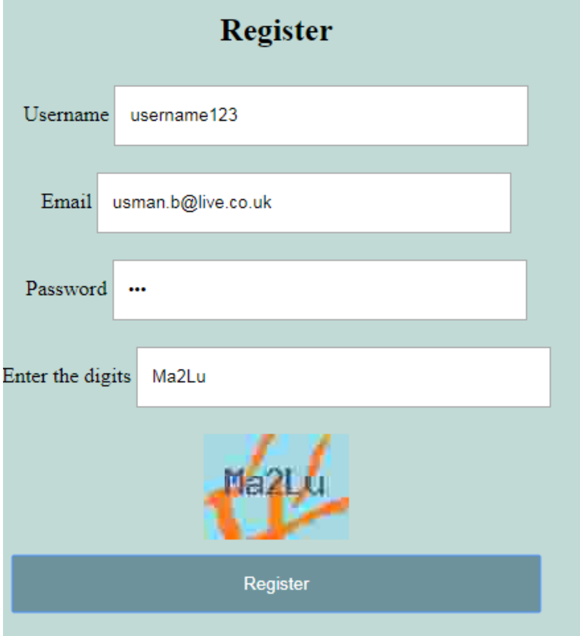
Wrong Code Entered

*Figure 10 shows the incorrect code for CAPTCHA.*

Referring to Figure 12, it shows the incorrect way of entering the wrong CAPTCHA code.

To move forward, the user needs to enter the right code for the system to detect that the username is a human, not a hacker.

### 6.1.2. Successful Register



**Register**

Username: username123

Email: usman.b@live.co.uk

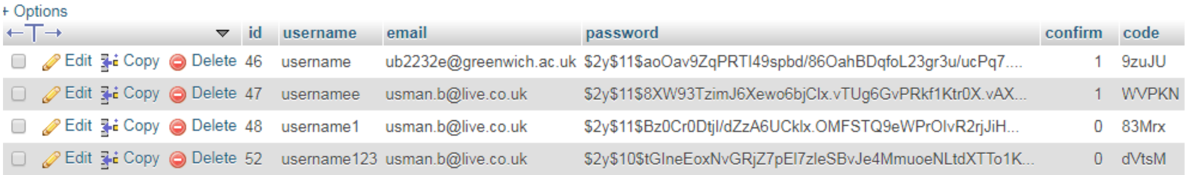
Password: •••

Enter the digits: Ma2Lu

*Figure 12 shows the correct validation for a successful registration*

Referring to Figure 14, it shows that the correct details have been entered and they have all been entered into the database. Another point to remember is that the password that has been entered is encrypted. This makes the security secure as any hackers that have access to the database, they can access it and they would not be able to crack the password so easily. Therefore, users must remember their password that they have entered. Another point is that all these details that have been entered, the ‘0’ is for those who cannot login.

Referring to Figure 15, it shows that it is put into the database.

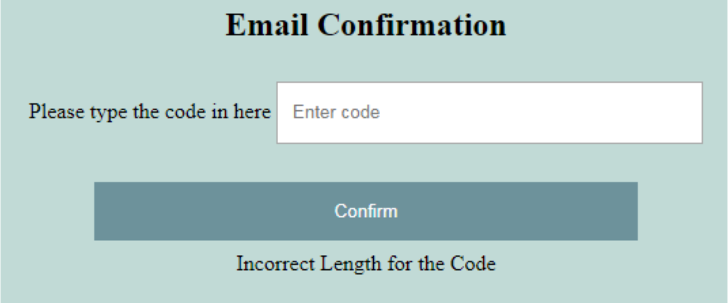


| | id | username | email | password | confirm | code |
|---|---|---|---|---|---|---|
| □ Edit Copy Delete | 46 | username | ub2232e@greenwich.ac.uk | $2y$11$aoOav9ZqPRTI49spbd/86OahBDqfoL23gr3u/ucPq7.... | 1 | 9zuJU |
| □ Edit Copy Delete | 47 | usernamee | usman.b@live.co.uk | $2y$11$8XW93TzimJ6Xewo6bjClx.vTUg6GvPRkf1Ktr0X.vAX... | 1 | WVPKN |
| □ Edit Copy Delete | 48 | username1 | usman.b@live.co.uk | $2y$11$Bz0Cr0Dtjl/dZzA6UCklx.OMFSTQ9eWPrOlvR2rjJiH... | 0 | 83Mrx |
| □ Edit Copy Delete | 52 | username123 | usman.b@live.co.uk | $2y$10$tGIneEoxNvGRjZ7pEI7zleSBvJe4MmuoeNLtdXTTo1K... | 0 | dVtsM |

*Figure 13 shows that it has been entered within the database.*

## 6.2. Email

### 6.2.1. Email Validation

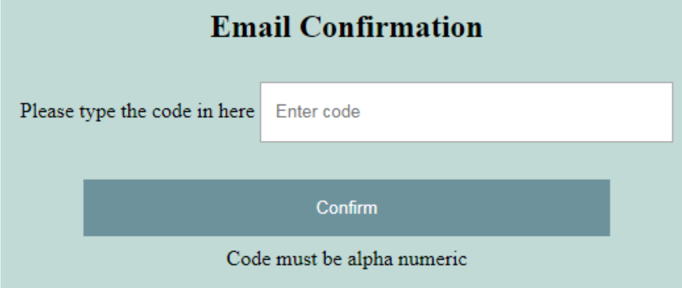


**Email Confirmation**

Please type the code in here: Enter code

Incorrect Length for the Code

*Figure 14 shows the validation for email. Incorrect length entered.*



**Email Confirmation**

Please type the code in here: Enter code

Code must be alpha numeric

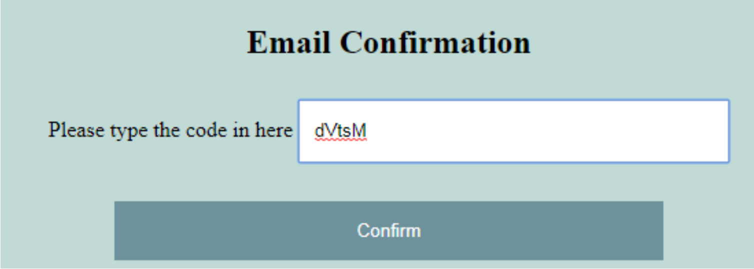
*Figure 15 shows the validation for the code.*

These validations are set in place for users to enter the right details. If any users make a mistake, these details will show the message.

Referring to Figure 16, the incorrect length of the code is entered. This shows that the user has entered a length below 5.

Referring to Figure 17, it shows that the user has entered a non-numeric and alpha code. The application detects this and shows the message.

### 6.2.2. Successful Email Verification

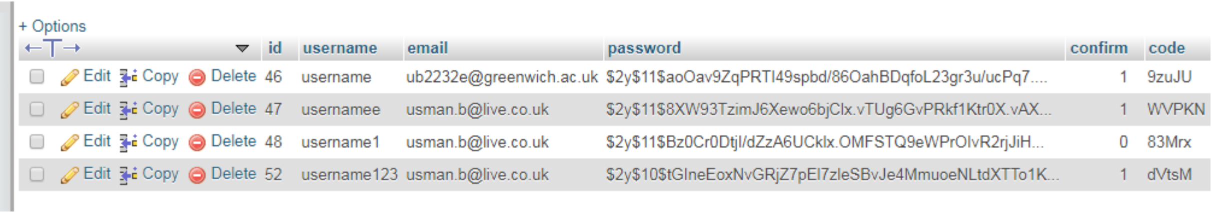


**Email Confirmation**

Please type the code in here dVtsM

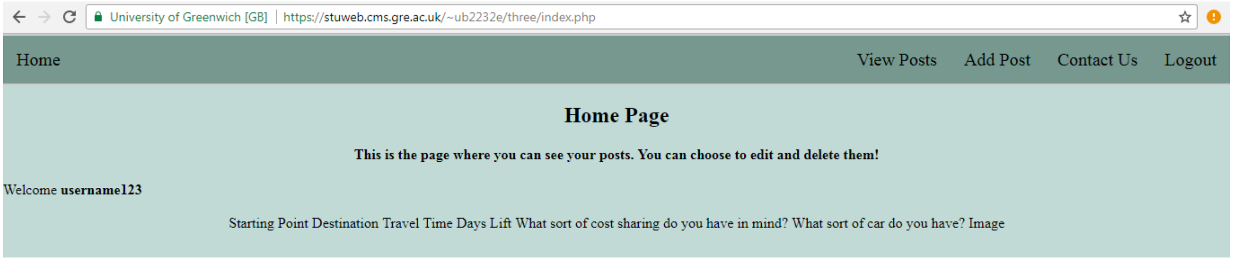
*Figure 16 shows the right code being entered.*

Referring to Figure 18, it shows the correct code being entered. Once the correct code has been entered, the account has been valid as shown in Figure 19. The confirm column goes 1, once the account has been valid.



| | id | username | email | password | confirm | code |
|---|---|---|---|---|---|---|
| □ Edit Copy Delete | 46 | username | ub2232e@greenwich.ac.uk | $2y$11$aoOav9ZqPRTI49spbd/86OahBDqfoL23gr3u/ucPq7.... | 1 | 9zuJU |
| □ Edit Copy Delete | 47 | usernamee | usman.b@live.co.uk | $2y$11$8XW93TzimJ6Xewo6bjClx.vTUg6GvPRkf1Ktr0X.vAX... | 1 | WVPKN |
| □ Edit Copy Delete | 48 | username1 | usman.b@live.co.uk | $2y$11$Bz0Cr0DtjI/dZzA6UCklx.OMFSTQ9eWPrOlvR2rjJiH... | 0 | 83Mrx |
| □ Edit Copy Delete | 52 | username123 | usman.b@live.co.uk | $2y$10$tGIneEoxNvGRjZ7pEI7zleSBvJe4MmuoeNLtdXTTo1K... | 1 | dVtsM |

*Figure 17 shows that the email validation has gone through to validate the account.*



University of Greenwich [GB] | https://stuweb.cms.gre.ac.uk/~ub2232e/three/index.php

**Home Page**

**This is the page where you can see your posts. You can choose to edit and delete them!**

Welcome **username123**

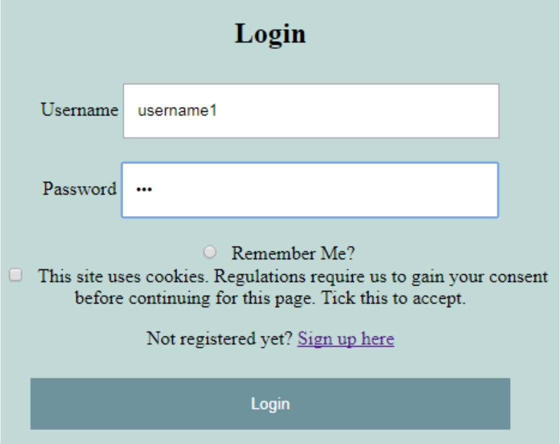
Starting Point Destination Travel Time Days Lift What sort of cost sharing do you have in mind? What sort of car do you have? Image

*Figure 18 shows that the user is logged in straight away.*

As the coursework scenario states, after the email address has verified his email address, the account is active. Referring to Figure 20, this sends the user straight to the home page. The user does not need to re-enter their details again. This makes it easier for the user.

## 6.3. Login

### 6.3.1. Login Validation



**Login**

Username username1

Password •••

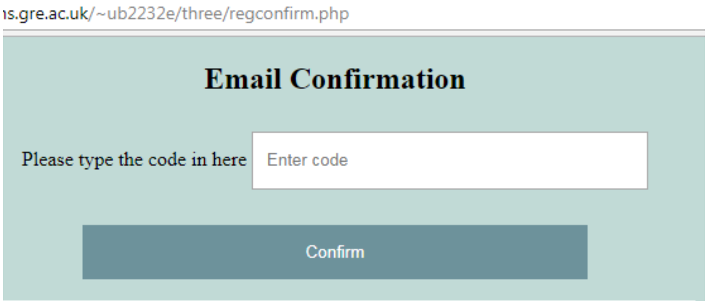
○ Remember Me?

□ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

*Figure 20 shows the unauthorised user trying to login.*

Referring to Figure 19, it shows the details those whom are unauthorised to login. Figure 22 shows that the unauthorised user is trying to login. Referring to Figure 21, the system detects this and sends the user to verify their account to have access to the account.

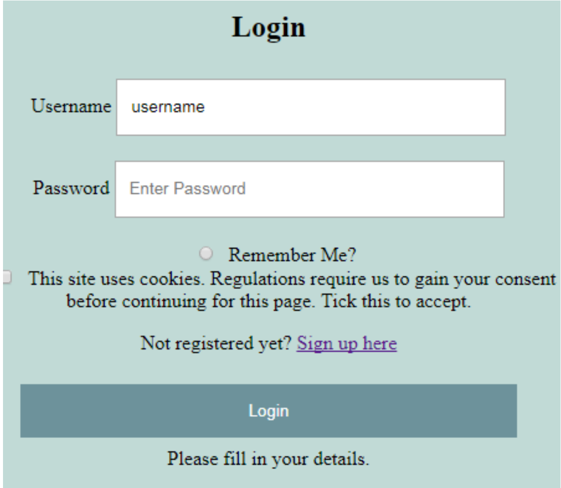


ıs.gre.ac.uk/~ub2232e/three/regconfirm.php

**Email Confirmation**

Please type the code in here Enter code

*Figure 19 shows the unauthorised user gets send to verify account.*



Login

Username username

Password Enter Password

○ Remember Me?

This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

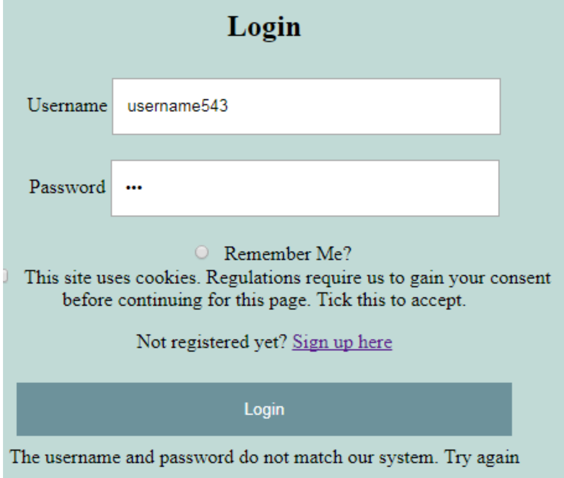
Not registered yet? Sign up here

Please fill in your details.

*Figure 21 shows the validation of the empty details.*

The validation is the same as previously said for the register. However, some of these are different.

Referring to Figure 23, it shows that the user has only entered the username, but not the password. This is detected by the application and has shown the user to enter the details that have been missing.



Login

Username username543

Password •••

○ Remember Me?

This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

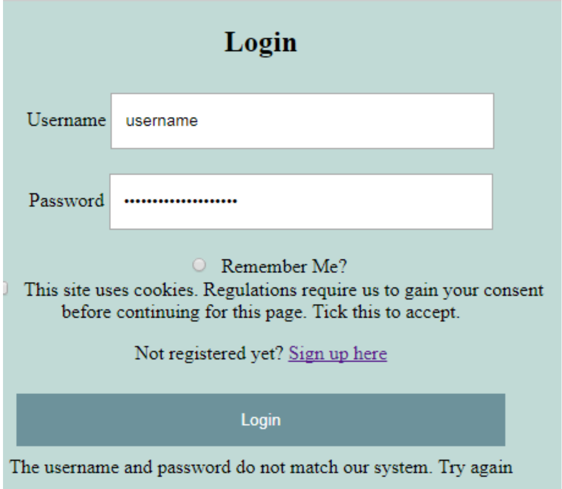
Not registered yet? Sign up here

The username and password do not match our system. Try again

*Figure 22 shows the validation of unknown username and password.*

Referring to Figure 24, it shows that these details are not recognised by the system. As previously shown the details of the database, these details are not recognised, and the application has detected this and has given the message to try again.

These authentications are important as unauthorised users cannot get access to the system.



Login

Username username

Password ••••••••••••••••••••

○ Remember Me?

This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

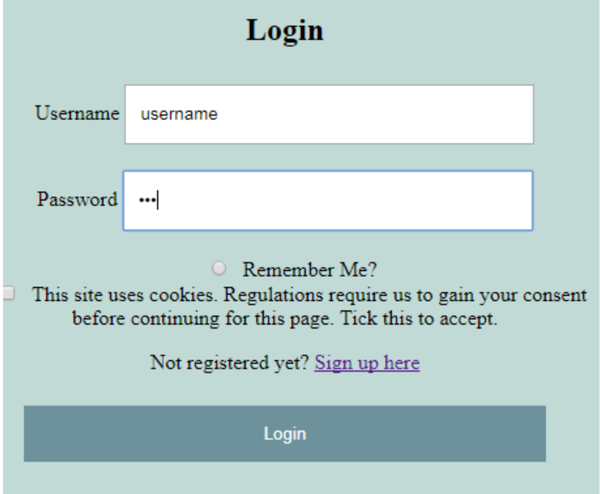
The username and password do not match our system. Try again

*Figure 23 shows the validation of a known username, but wrong password.*

Referring to Figure 25, this is the similar to Figure 24. However, this time, the username has been recognised. However, the password is wrong. This is also detected by the application.

The user must have the correct password to log in. This is the same to authorised users too. The details must match.

### 6.3.2. Successful Login



Login

Username: username

Password: •••

○ Remember Me?

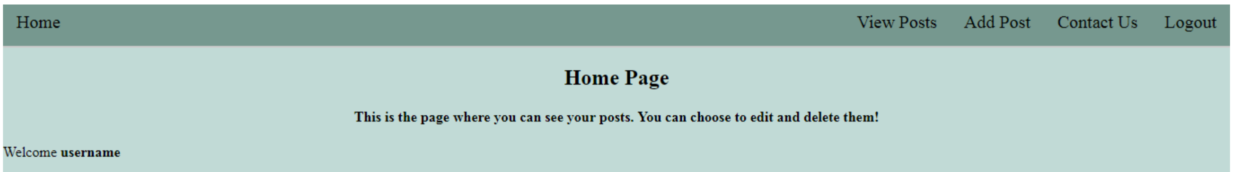
□ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

*Figure 24 shows a successful login details entered.*

As previously shown for a successful register, Figure 26 shows that these correct details grant the user access to the home page of the system. The user has to go through all these validations as stated above in order to get access to the home page.

As shown on the home page, it differentiates each user by welcoming them. It states, “welcome username”.



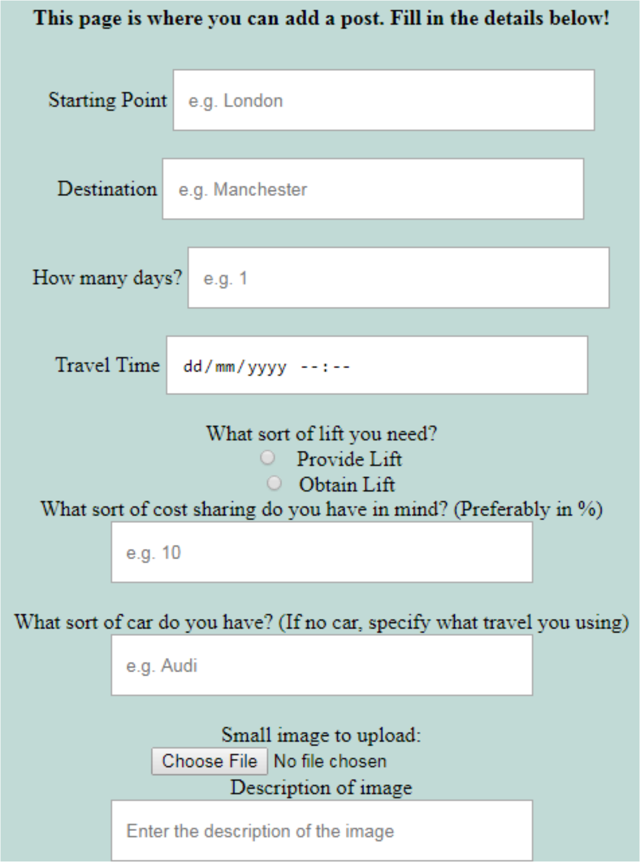
Home Page

This is the page where you can see your posts. You can choose to edit and delete them!

Welcome **username**

*Figure 25 shows that the correct details gets access to the home page.*

## 6.4. Add Post



**This page is where you can add a post. Fill in the details below!**

Starting Point: e.g. London

Destination: e.g. Manchester

How many days?: e.g. 1

Travel Time: dd/mm/yyyy --:--

What sort of lift you need?
○ Provide Lift
○ Obtain Lift

What sort of cost sharing do you have in mind? (Preferably in %): e.g. 10

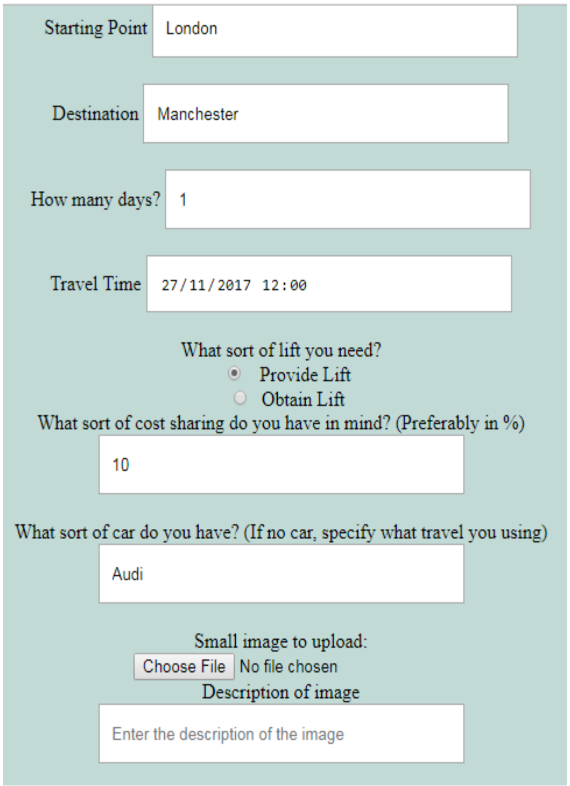
What sort of car do you have? (If no car, specify what travel you using): e.g. Audi

Small image to upload:
Choose File No file chosen

Description of image: Enter the description of the image

*Figure 26 shows the details for validation.*

Referring to Figure 28, these are the details of what to expect to have a successful post. I have put these validations on the placeholder of each textbox to make the user aware of what to type in if they want a successful post. If they do not follow these steps and enter something in correct, the system will detect this and give them an error for each of these details.



Starting Point London

Destination Manchester

How many days? 1

Travel Time 27/11/2017 12:00

What sort of lift you need?<br>
● Provide Lift<br>
○ Obtain Lift

What sort of cost sharing do you have in mind? (Preferably in %)

10

What sort of car do you have? (If no car, specify what travel you using)

Audi

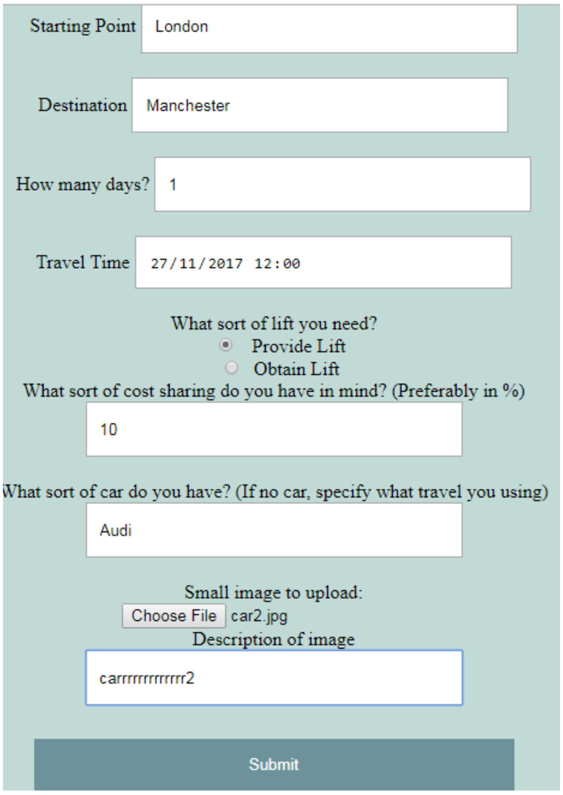
Small image to upload:

No file chosen

Description of image

Enter the description of the image

*Figure 27 shows the post being added successfully, but without an image.*



Starting Point London

Destination Manchester

How many days? 1

Travel Time 27/11/2017 12:00

What sort of lift you need?<br>
● Provide Lift<br>
○ Obtain Lift

What sort of cost sharing do you have in mind? (Preferably in %)

10

What sort of car do you have? (If no car, specify what travel you using)

Audi

Small image to upload:

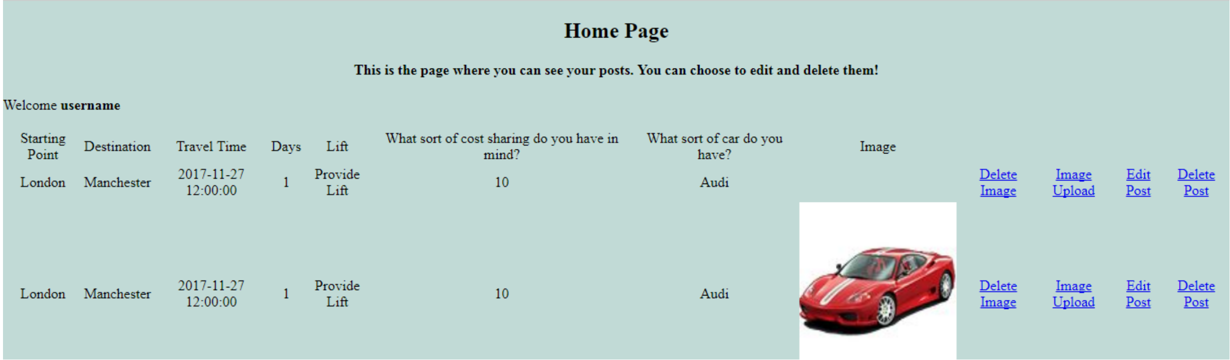
car2.jpg

Description of image

carrrrrrrrrrrr2

*Figure 28 shows the successful post with an image.*

Referring to Figure 29, as specified, it shows that the user has an option to upload an image or not. The user has chosen not to upload an image and it shows within Figure 31. Referring to Figure 30, it shows that the user has uploaded an image and they have the post successful. Figure 31 shows that both posts are successful and have been uploaded alongside the image showing.



**Home Page**

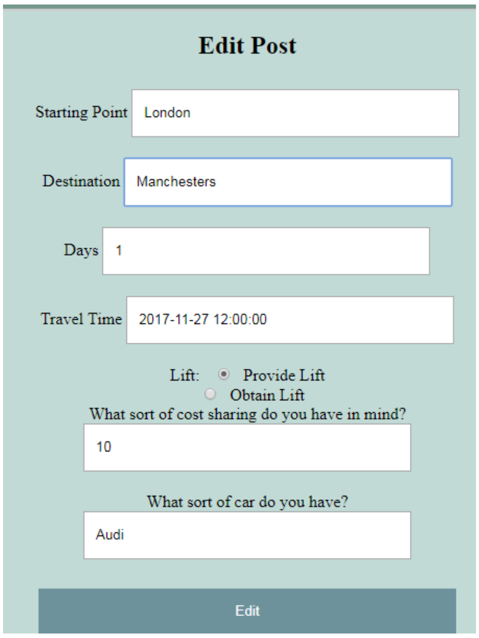
**This is the page where you can see your posts. You can choose to edit and delete them!**

Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 29 shows that both successful posts are uploaded.*

## 6.5. Delete and Edit Post



**Edit Post**

Starting Point: London

Destination: Manchesters

Days: 1

Travel Time: 2017-11-27 12:00:00

Lift: ● Provide Lift ○ Obtain Lift

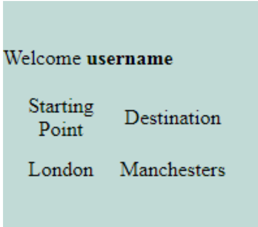
What sort of cost sharing do you have in mind?: 10

What sort of car do you have?: Audi

*Figure 30 shows the user editing the post.*

Referring to Figure 32, it shows that the user has chosen to edit the post. This post is changed by only adding an ‘s’ to Manchester. This was successful. Referring to Figure 33, it shows that the user that has edited, it has shown as Manchesters on its post.

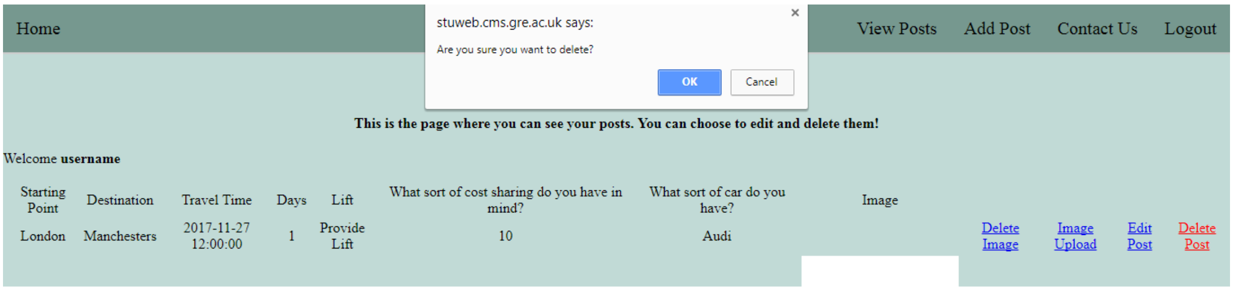
The same principle of validation has been applied here. If the user has entered a wrong cost, such as 10a, it would make the edit unsuccessful.



Welcome **username**

| Starting Point | Destination |
|---|---|
| London | Manchesters |

*Figure 31 shows that the edited post is successful.*



stuweb.cms.gre.ac.uk says:

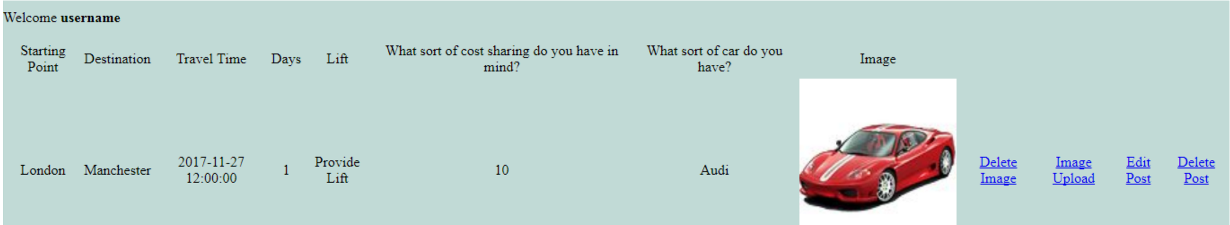
Are you sure you want to delete?

**This is the page where you can see your posts. You can choose to edit and delete them!**

Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchesters | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 32 shows that the user has chosen to delete the post.*



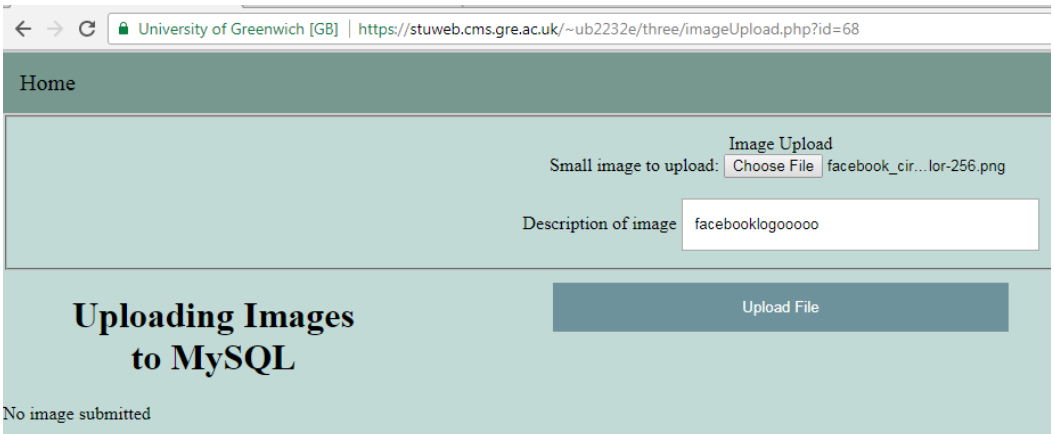
Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 33 shows that the deleted post has been successful.*

Referring to Figure 34, it shows that the user has chosen to delete the post. A message has been appeared to make sure that the user is sure to delete the post. Figure 35 shows that the deleted post has been successful.

## 6.6. Image Upload



University of Greenwich [GB] | https://stuweb.cms.gre.ac.uk/~ub2232e/three/imageUpload.php?id=68

Home

Image Upload

Small image to upload: Choose File facebook_cir...lor-256.png

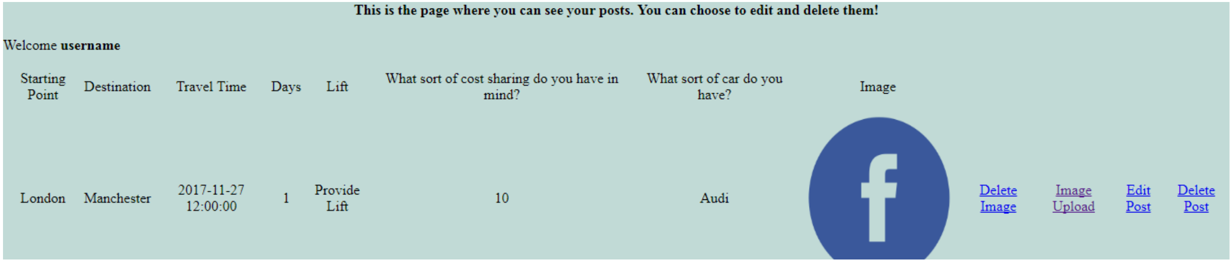
Description of image facebooklogooooo

Upload File

**Uploading Images to MySQL**

No image submitted

*Figure 34 shows that the user has chosen to upload an image.*



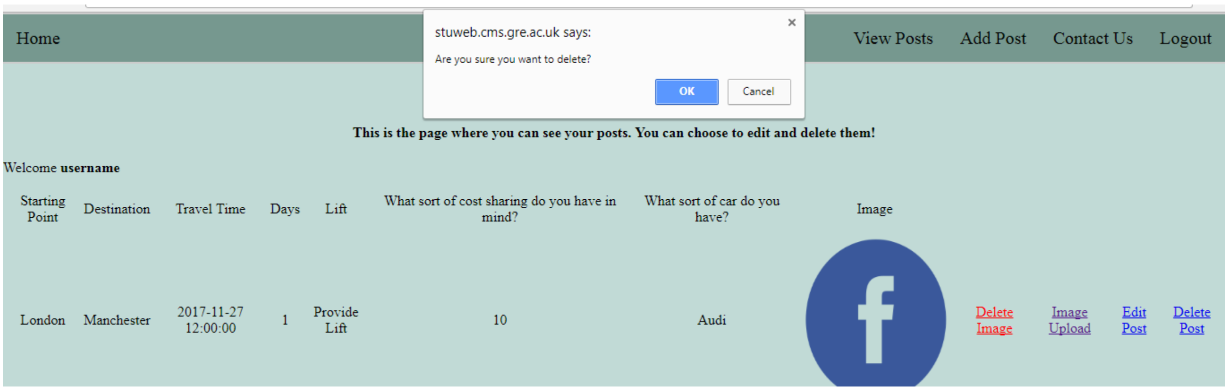
**This is the page where you can see your posts. You can choose to edit and delete them!**

Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 37 shows that the chosen image has been upload.*

Referring to Figure 36, it shows that the user has chosen to upload an image for the chosen post. The user has a choice if the user wants to upload. Referring to Figure 39, it shows that the image upload has been successful.



Home | View Posts | Add Post | Contact Us | Logout

stuweb.cms.gre.ac.uk says:

Are you sure you want to delete?

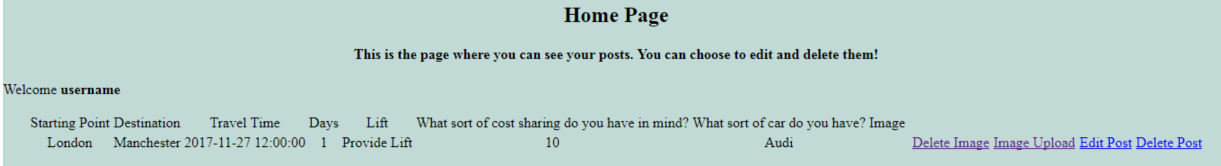
OK | Cancel

**This is the page where you can see your posts. You can choose to edit and delete them!**

Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 36 shows that the user can delete the image.*



**Home Page**

**This is the page where you can see your posts. You can choose to edit and delete them!**

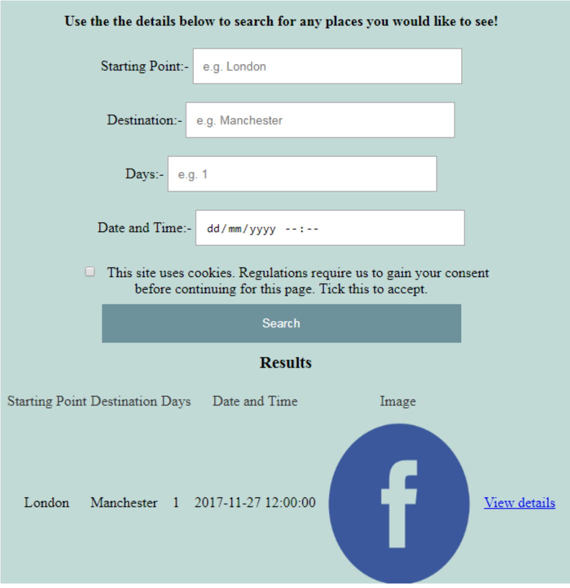
Welcome **username**

| Starting Point | Destination | Travel Time | Days | Lift | What sort of cost sharing do you have in mind? | What sort of car do you have? | Image | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | | Delete Image | Image Upload | Edit Post | Delete Post |

*Figure 35 shows that the image has been deleted.*

Referring to Figure 37, and Figure 38, it shows that the user has deleted the image. It gives the user the option to delete the image. Figure 37 shows that the image has been deleted.

## 6.7. Search



**Use the the details below to search for any places you would like to see!**

Starting Point:- e.g. London

Destination:- e.g. Manchester

Days:- e.g. 1

Date and Time:- dd/mm/yyyy --:--

□ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Search

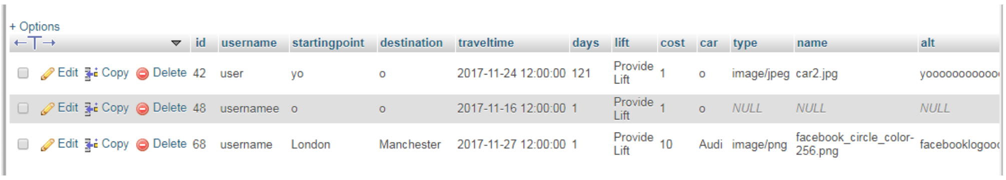
**Results**

| Starting Point | Destination | Days | Date and Time | Image | |
|---|---|---|---|---|---|
| London | Manchester | 1 | 2017-11-27 12:00:00 | | View details |

*Figure 38 shows that the user has searched for an item.*

Referring to Figure 40, it shows that the user has searched for an item. The user has option to view more of the details. However, any unauthorised user gets referred to the login page to enter their details.

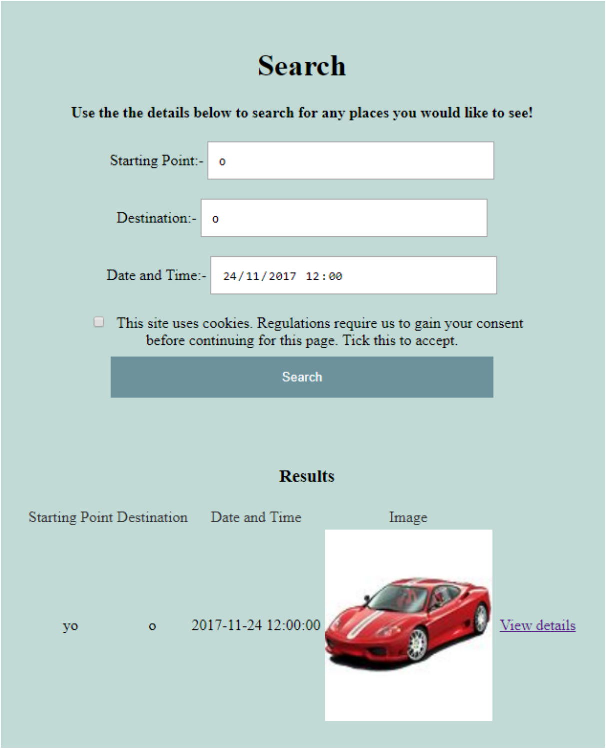
If the user is already authenticated, the user gets referred to the Search page where they can view the whole of the post.



+ Options

| | id | username | startingpoint | destination | traveltime | days | lift | cost | car | type | name | alt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ Edit Copy Delete | 42 | user | yo | o | 2017-11-24 12:00:00 | 121 | Provide Lift | 1 | o | image/jpeg | car2.jpg | yooooooooooo |
| □ Edit Copy Delete | 48 | usernamee | o | o | 2017-11-16 12:00:00 | 1 | Provide Lift | 1 | o | NULL | NULL | NULL |
| □ Edit Copy Delete | 68 | username | London | Manchester | 2017-11-27 12:00:00 | 1 | Provide Lift | 10 | Audi | image/png | facebook_circle_color-256.png | facebooklogooo |

*Figure 39 shows the details of the post in the database.*



**Search**

**Use the the details below to search for any places you would like to see!**

Starting Point:- o

Destination:- o

Date and Time:- 24/11/2017 12:00

□ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Search

**Results**

| Starting Point | Destination | Date and Time | Image | |
|---|---|---|---|---|
| yo | o | 2017-11-24 12:00:00 | | View details |

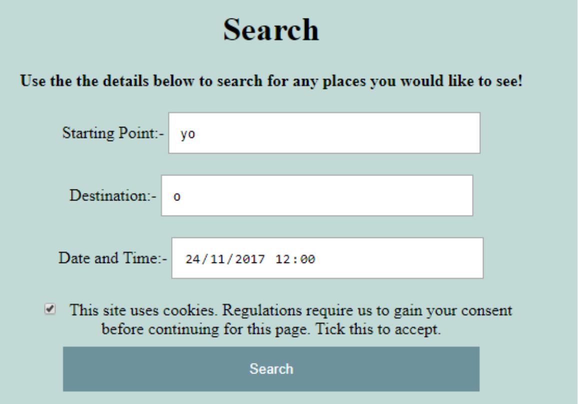
*Figure 40 shows the details being entered for search.*

Referring to Figure 41, it shows all the post of what the user has available to search.

Once the user has searched for the post that has been without an image, it shows that the details only shown is an image that matches what has been searched.

The image is shown by the filtering between the starting point, destination and days. However, the main point is that the search aimed for a post without an image is not shown.

## 6.8. Cookies



Search

Use the the details below to search for any places you would like to see!

Starting Point:- yo

Destination:- o

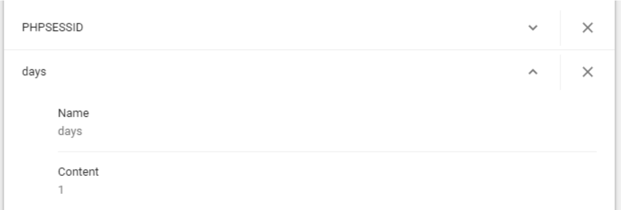
Date and Time:- 24/11/2017 12:00

☑ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

*Figure 41 shows the details being entered and cookie being accepted.*

Referring to Figure 43, it shows that the details being entered, and the cookie law being accepted by the user.

Once this has been accepted and searched, Figure 44 shows that the cookies work and has been accepted.



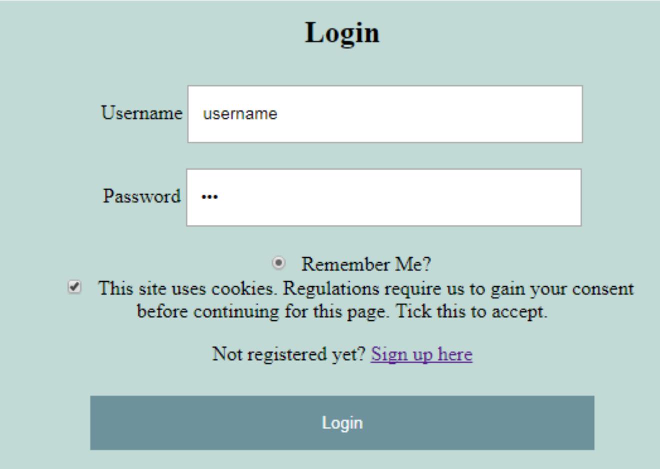
PHPSESSID

days

Name
days

Content
1

*Figure 42 shows that the cookie entered has been entered and accepted.*



Login

Username username

Password •••

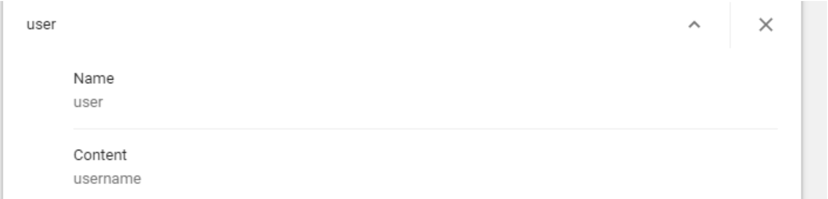
◉ Remember Me?

☑ This site uses cookies. Regulations require us to gain your consent before continuing for this page. Tick this to accept.

Not registered yet? Sign up here

*Figure 43 shows the same process for the login.*

This is the same process as the search that is shown above. Referring to Figure 45, it shows that the user has accepted and Figure 46 shows that the cookie has been accepted and works.



user

Name
user

Content
username

*Figure 44 shows that the username cookie has been accepted and works.*

## 7. References

*These references are used within the code. These references are commented within the code, and are here for references.*

Mcmanus, K. (2017). Uploading Files to MySQL. [Online] stuweb. Available at: http://stuweb.cms.gre.ac.uk/~ha07/web/PHP/imageUpload.html [Accessed 30 Nov. 2017].

Mcmanus, K. (2017). Uploading Files to MySQL. [Online] stuweb. Available at: http://stuweb.cms.gre.ac.uk/~ha07/web/PHP/graphics.html [Accessed 30 Nov. 2017].

PHP. (2017). password_hash. [Online] Available at: //http://php.net/manual/en/function.password-hash.php [Accessed 30 Nov. 2017].

PHP, R. (2017). Redirecting from HTTP to HTTPS with PHP. [Online] Stackoverflow.com. Available at: https://stackoverflow.com/questions/5106313/redirecting-from-http-to-https-with-php [Accessed 30 Nov. 2017].

Php.net. (2017). PHP: password_verify - Manual. [Online] Available at: http://php.net/manual/en/function.password-verify.php [Accessed 30 Nov. 2017].

W3schools.com. (2017). HTML Responsive Web Design. [online] Available at: https://www.w3schools.com/html/html_responsive.asp [Accessed 30 Nov. 2017].