



National  
College *of*  
Ireland

Software Project Report

Tours2Connect

Shane Murphy

BSHCYB4 - BSc (Honors) in Computing

## Contents

Executive Summary.....	4
1 Introduction.....	5
1.1 Background .....	5
1.2 Aims .....	7
1.3 Technologies .....	7
1.4 Structure.....	9
2 Tours2Connect Methodology .....	10
2.1 Tours2Connect Prototype Model.....	10
2.1.1 Phase 1 – Analyse Requirements .....	10
2.1.2 Phase 2 – Design .....	10
2.1.3 Phase 3 – Build prototype .....	10
2.1.4 Phase 4 – Critique prototype .....	11
2.1.5 Phase 5 – Implement.....	11
2.1.6 Phase 6 – Test .....	11
2.1.7 Phase 7 – Operate .....	11
2.2 Advantages of Prototype model.....	12
2.3 Prototype Process .....	12
3 System .....	14
3.1 User Requirements.....	14
3.1.1 Functional requirements .....	15
Requirement 1: User Registration .....	16
Requirement 2: User Login.....	18
Requirement 3: Create Tour.....	20
Requirement 4: Search Tour .....	22
Requirement 5: Location – Google Maps API .....	24
Requirement 6: Contacting users .....	26
Requirement 7: Function Level Access Control.....	28
Requirement 8: PayPal Payment .....	30
3.2 Non-Functional Requirements.....	32
3.2.1 Security requirement .....	32

3.2.2 Performance/Response time requirement .....	33
3.2.3 Availability requirement .....	33
3.2.4 Recover requirement .....	33
3.2.5 Robustness requirement .....	34
3.2.6 Reliability requirement .....	35
3.2.7 Maintainability requirement.....	36
3.2.8 Extendibility requirement .....	36
3.2.9 Reusability requirement.....	36
3.2.10 Data requirements .....	37
3.2.11 User requirements .....	38
3.2.12 Environmental requirements.....	40
3.2.13 Usability requirements .....	40
3.3 Design and Architecture .....	41
3.3.1 Architecture .....	41
3.3.2 Algorithms .....	43
3.3.2.1 Hashing .....	43
3.4 Evolution of Tours2Connect Web Application .....	44
3.5 Implementation .....	45
3.5.1 Hashing .....	45
3.5.2 Google Maps API .....	47
3.5.3 Login.....	50
3.5.3 Register .....	53
3.5.4 Tours Table Extraction .....	56
3.5.5 Users entering tours to database.....	58
3.5.6 Extracting tour information separately .....	59
3.5.5 Technology Overview .....	61
3.5.6 Security .....	63
3.5.9 CIA Triad .....	75
3.6 Testing.....	76
3.6.1 Security Testing.....	76
3.6.2 Standard Testing.....	79
3.6.3. Customer Testing .....	85
3.7 Evaluation.....	90
3.8 Graphical User Interface (GUI) Layout .....	92

4	Conclusions.....	97
5	Further Developments & Research .....	98
6	Appendix .....	99
6.6	Project Proposal .....	99
	Objectives.....	99
	Background .....	100
	Technical Approach.....	101
	Project Plan .....	102
	Technical Details .....	102
	Evaluation.....	103
6.7	Monthly Journals .....	104
	September.....	104
	October.....	105
	November.....	106
	December.....	107
	January.....	108
	February .....	109
	March	110
	April	111
	May	112
7	References .....	113

## **Executive Summary**

*Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution and briefly report the findings of the evaluation.*

Overseas tourists that visited Ireland in 2016 grew by 8.8% to 8.742 million. There are many companies that offer tours around Ireland but nothing like what Tours2Connect can offer. The problem being addressed is to enter a very niche market in which I am yet to find a competitor. Tours2Connect is a web application in which tour guides can register and offer tours to other users. This is different to other tour guiding companies as these tour guides would be self-employed and we are giving them a platform to display their tours that they offer. Other companies that offer tours such as 'Irish day tours' and 'paddy wagon tours' are huge companies, but their tours are very commercialized and the individuals giving the tours are employed by the companies, what Tours2Connect offers differently to the companies mentioned, is the fact that the tour guides aren't employed by a company and they are doing the tours freelance, as they aren't employed by us.

The objective of Tours2Connect is to breach the market and allow personalized tours to be given to tourists or people who wish to see the beautiful scenic tours that the country of Ireland offers, by tour guides who have a love for the country and are passionate about the topic or tour they are offering.

# 1 Introduction

## 1.1 *Background*

The idea of a web application where tour guides can display their up and coming tours with the option for other users to attend came from my time on work placement. My work placement took place in Beaumont Hospital where I had met a lot of nice people, staff and patients, but my idea came from a patient who I had helped in the hospital. He had told me that he loved giving guided tours to locals and visiting families to the area of nearby scenic attractions. From this he had told me he wished he could have had more people on his tours to share his love of mountain walking, and the scenic areas he personally loved etc. After talking to this man, I got an idea to create a web application in which people like the man in Beaumont hospital, could post their tours online and where they could get a much bigger audience instead of a few locals from their town.

When I had gathered the idea, I then ran it by my manager in Beaumont Hospital and a few colleagues, they thought the idea was very unique and they had not heard of a web application that supports a service like this. When I heard them say that it was unique and had not seen another application like this, I thought to myself that the idea of the final year project is to stand out and if the idea is unique, it will stand out, that is why I have chosen to go ahead with it.

Since leaving Beaumont Hospital and starting back in NCI, I had been thinking of different ways to approach the creation of the web application and how I could make it better. I believe I achieved this with my creation of Tours2Connect.

**Tourism Numbers 2013 – 2016**  
*Where did Ireland's tourists come from?*

Numbers (000s)	2013	2014	2015	2016
<b>Britain</b>	<b>2,870</b>	<b>3,007</b>	<b>3,346</b>	<b>3,632</b>
<b>Mainland Europe</b>	<b>2,346</b>	<b>2,490</b>	<b>2,880</b>	<b>3,102</b>
France	409	420	471	494
Germany	466	535	609	624
Italy	226	246	304	326
Spain	249	274	322	370
Netherlands	148	151	174	222
Belgium	95	99	121	127
Denmark	51	55	66	64
Sweden	72	60	64	59
Switzerland	73	84	105	107
Austria	51	57	53	58
Norway	50	50	58	50
Poland	152	140	161	176
All Other Europe	306	318	373	427
<b>North America</b>	<b>1,039</b>	<b>1,146</b>	<b>1,294</b>	<b>1,477</b>
USA	924	1,005	1,129	1,294
Canada	115	140	165	183

From the image above, you can see that tourism numbers are on the rise every year, this allows for innovative business ideas like mine to expand and become a massive competitor in such a niche market. With many people wanting to showcase their love for Irish amenities throughout Ireland, the idea to create a web application so that people can do this is a great idea and the time seems to be right as there are thousands of tourists who take tours across the country.

## **1.2 Aims**

My Project will consist of developing a secure web application for the tourism industry. The secure web application will consist of a multiple page interface that are linked together, in order to allow potential customers to browse through different tours and select the one that appeals to them the most. The tours will be displayed through a table that's linked to a database so that the available tours will be displayed within the table.

Each user will register and sign in, from here the logged in user will have the choice to add a tour or the opportunity to browse the tours available. When a user see's a tour that they like, they can go onto the page and see more information about this tour and if they are interested in the idea, they can enquire about it. From here a secure payment method shall be available to each user.

The idea of the project is to allow users to sign up and register, from here they can act as a tour guide or a tourist. The users who wish to be tour guides can fill in a form and post potential tours that they have planned, these tours will be displayed on the web application through a table that has tour data extracted from a database through PhpMyAdmin. When the user does search and sees a tour that they like, the user will click this, and it brings them to that tours page, where more information is displayed. From here the user can contact the tour guide for more information or pay through PayPal. Also, there will be a price summary of each tour displayed.

The security aspect of the project is obviously high as it is under the specialization of Cyber Security, but it also allows for entry of people's sensitive information like passwords and usernames. These will be hashed using MD5 as the algorithm, and the database PhpMyAdmin, will store the hashed passwords.

None of the tour services or searching will not be available until the user logs in or registers an account.

## **1.3 Technologies**

The web application will be build using PHP, HTML5, JavaScript and Bootstrap to create the relevant content needed to be displayed. The bootstrap template will



give a base to the web application, and I will edit this content using PHP, CSS and HTML5 to ensure the web application is how I want it to look and user friendly.

PHP is a server-side scripting language; Therefore, it can collect data from forms and generate dynamic content too. It is suited for web development and can also be embedded into HTML. All the forms on the web application, to name a few - Register, Login, Create Tour pages all have forms on them, in which the data is sanitized and then entered into the database. Some of the information is then called upon and echoed on the web application, this includes the logged in user, or every tour that is displayed in a table on the tours' page.

HTML5 is a mark-up language used for structuring and presenting content for web applications, It's the newest version and is linked with the PHP to show the web applications display. Browsers don't display HTML tags but use them to interpret and structure the content that I have created on all my web pages.

JavaScript is a programming language that allows interactivity on the web application for the users such as error messages. I used this on a lot of my pages, login and register have a function where when a href is clicked, the forms switch. I also used it on my google maps API.

API's are a set of tools and methods, that allow for building software applications in which users can interact with them. The google maps API is one of the most commonly used API's as it can be manipulated to display location data that I have stored in my database.

I created a database and stored all tour information and user information on it. This is stored within PhpMyAdmin. The information in the database is extracted using PHP methods that are used to implement the body of the tours table.

PhpMyAdmin is created to handle the administration of MySQL databases. This is what I used to store the data entered into the web application.

I developed my Web Application through cloud9, and have it hosted on Heroku. Both have caused major issues with the CSS and database connections.

## **1.4 Structure**

Section 1 of this document discussed the background, aims and technology that have been used to create Tours2Connect.

Section 2 discuss' the prototype methodology, within this it discuss' the prototype model for Tours2Connect and the Prototype process.

Section 3 is about the system, within this we will see user requirements, functional requirements, the use case diagram for tours2connect and also the non-functional requirements related to my project. Within section 3, we talk about the design and architecture, the evolution of tours2connect, the implementation and the graphical user interface layout.

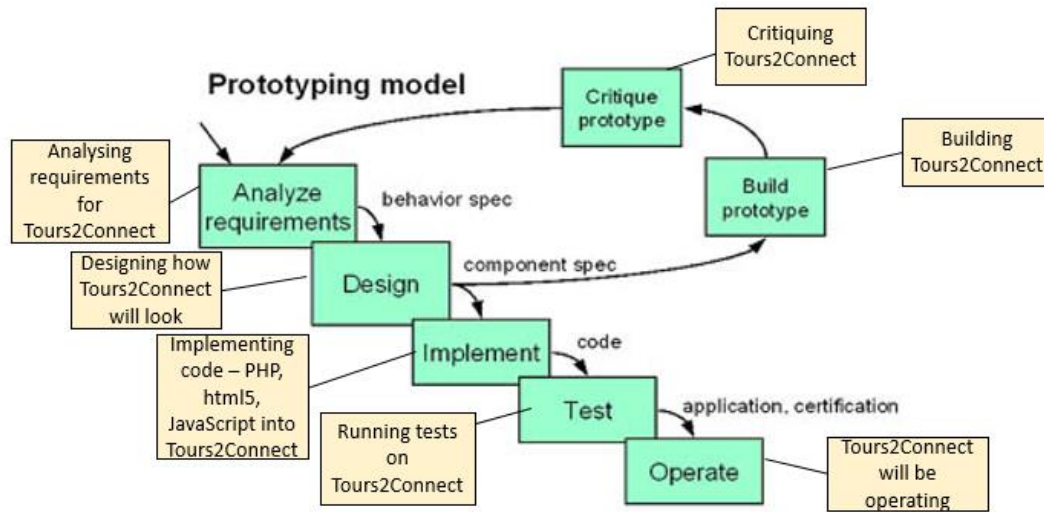
Section 4 discuss' the security aspects of the web application and how they were implemented. This section will contain code snippets and explanations of how they were used.

Section 5 & 6 discuss' the conclusion to the technical report and the further development and research.

Section 7 is the final part of the technical report, here we discuss the project proposal and the monthly journals.

## 2 Tours2Connect Methodology

### 2.1 Tours2Connect Prototype Model



#### 2.1.1 Phase 1 – Analyse Requirements

Start of the prototype model analysing the requirements needed to execute Tours2Connect. The requirements are stated and explained later on in the report. This will discuss technical and non-technical requirements of the whole project.

#### 2.1.2 Phase 2 – Design

Designing begins, this loops in with the next two phases – to build the prototype and critique it and link it back into the requirements. The design aspect of the project changes a lot throughout the process of building the prototype. Different designs suit different pages, and it also depends on how well the code created works.

#### 2.1.3 Phase 3 – Build prototype

This section is linked into the critiquing of the prototype. Building the prototype is a very bare image of how the fully developed application will look. The building phase takes many different twists throughout the entire prototype model. Some aspects of the project won't work or fit in with the rest of the project, ideas may

change about how the building will take part, there are many different ways in which this phase may cause complications to the entire project.

#### **2.1.4 Phase 4 – Critique prototype**

The above loop will repeat itself throughout the process. This is when you will begin to nail down what you do want, and do not want to build. You begin putting all built items into place and you can understand how they will work and look together.

#### **2.1.5 Phase 5 – Implement**

This is when you start to implement code and algorithms into your prototype to improve it. Languages such as PHP, HTML, CSS, are all implemented correctly into your web application. All tags involved in the creation of the web application are implemented so that the end product is fully established. Different algorithms are used within the stage and they bring everything together. Hashing algorithm is the perfect algorithm that is an example of improving your prototype, it makes the server side more secure and reliable.

#### **2.1.6 Phase 6 – Test**

Testing your implementation of code to test for errors and other problems that may be available. This ensures that the finished product is safe and allowed to be deployed for real world users.

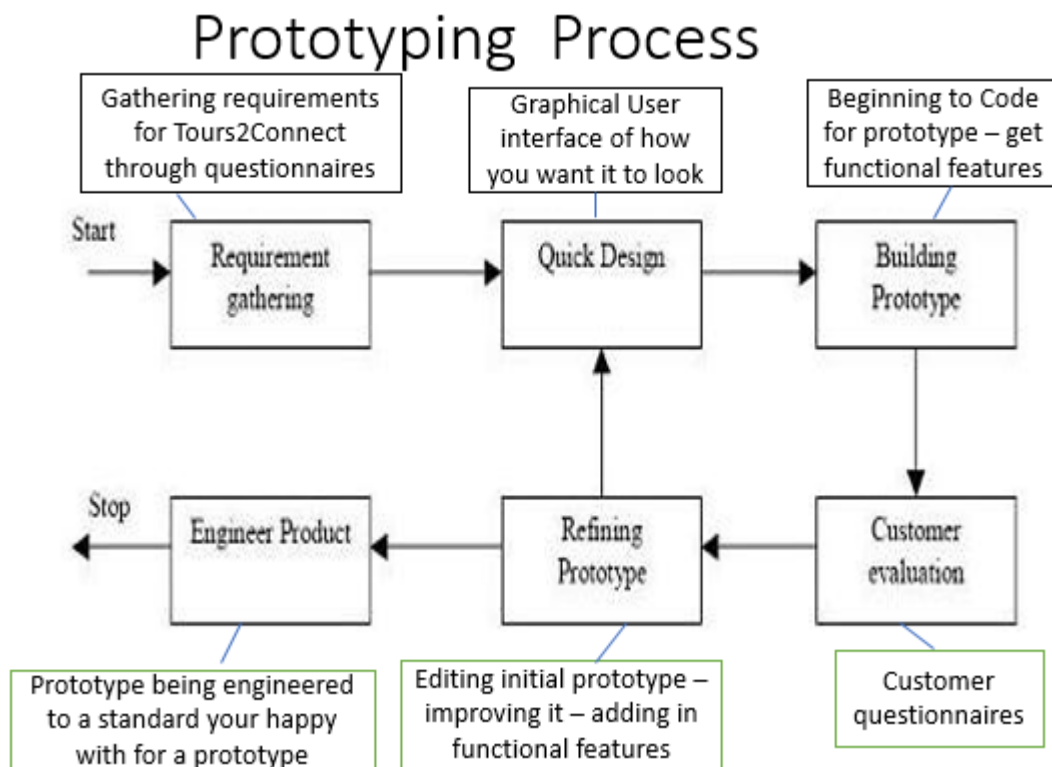
#### **2.1.7 Phase 7 – Operate**

Tours2Connect will be operational after these steps are taken.

## 2.2 Advantages of Prototype model

- Users are actively involved in the development, receiving user inputs and they may find problems that you will not.
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solution.

## 2.3 Prototype Process



The above image created, outlines the 6 steps to the process of prototyping. Requirements gathering was done on friends, friends of friends and fellow students. These people agreed for their data to be used in the evolution of Tours2Connect.

A quick design was created, which was then built and presented for my mid-point presentation. This layout has changed massively since the early days of the prototype at the mid-point.

Customer evaluations were again done by allowing users to have access to the web application, and then getting them to fill out a survey with their feedback, which was then taken on board and changes were made.

Refining the prototype was then done, taking on customer evaluations, changes were made. Additional functions were added into the web application, and some functions were taken out or improved as users didn't like the idea.

Engineer product stage is the process of the prototype turning into a fully functional, well-engineered web application available for use by users.

## 3 System

### 3.1 User Requirements

This section will be similar to your original requirements specification. The requirements have definitely evolved and have been expanded upon since then.

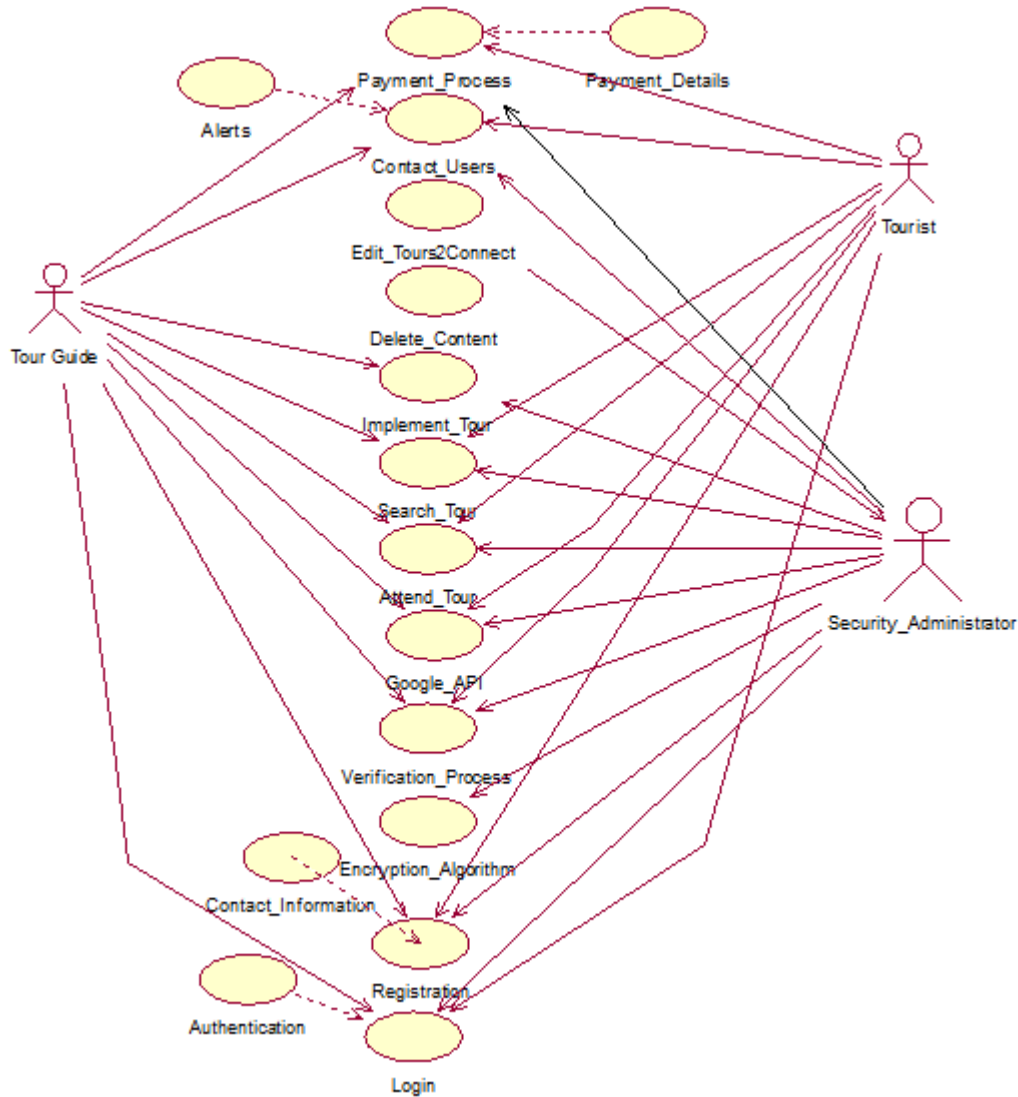
- Hashing algorithms for log-in and registering features within the web application to allow for a secure web application – Stored in the database
- Google maps API connected to database, to display markers and locations of all tours available to the users.
- Access – users cannot view web application further without logging in – OWASP Top 10 Number 7: Missing Function Level Access Control - information stored and encrypted.
- Tour Guides to be able to enter their tour through a form
- Login – Logging in through a form – information will be hashed and then unhashed to give access to the next page.
- Verification of Guide – security reasons – must be safe and secure for all users.
- Tours extracted from the database, to display through a table.
- Each tour is individually extracted from the table displaying tour information by ID – Tour Name displayed, as well as other information.
- Create Tour – Name, category, location etc.
- Search Tour – through database – by category, location etc.
- Contacting Guide – Any queries for contact guide – message box
- Payment Process – PayPal connection.

### 3.1.1 Functional requirements

#### 3.1.1.1 Use Case Diagram

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements





## Requirement 1: User Registration

### 3.1.1.2 Description & Priority

This use case describes the registration of a user on the Web Application. Important use case as users cannot get past the main page without registering.

### 3.1.1.3 Use Case

Registration

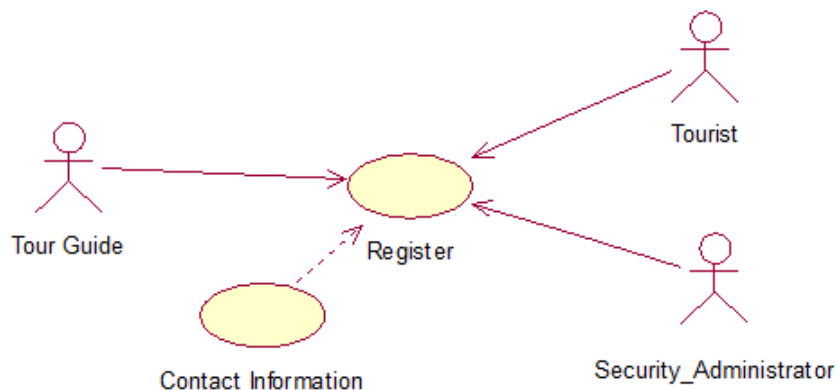
#### Scope

The scope of this use case is for a user of the web application to register to the web application to be able to gain further access

#### Description

This use case describes the process of users registering.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a Tour guide or tourist want to register.

## **Main flow**

1. The system identifies the users on the web app.
2. The Tour guide / tourist can see the main page.
3. The system won't allow the users to gain further access
4. The Tour guide / tourist have to register to gain further access.
5. The Tour guide / tourist will click the register button on the home page.
6. The system will allow the user to enter details.
7. The Tour guide / tourist will enter details and be accepted if the information is viable. (See A1 – invalid credentials)
8. The system will redirect the user to the next page – where tours are displayed – giving that user access.

## **Alternate flow**

A1: <Invalid credentials>

1. The system indicates that the information entered is not valid – prompts for correct details
2. The Tour guide / tourist will have to enter valid details to gain access
3. The system checks newly entered credentials and will accept if correct.
4. Continue to main flow point 8.

## **Exceptional flow**

-

## **Termination**

The system stores all valid information into the database and will redirect users to the next page – where tours are displayed.

## **Post condition**

The system goes into a wait state

## Requirement 2: User Login

### 3.1.1.4 Description & Priority

This use case describes the logging in of a user on the Web Application. Important use case as users cannot get past the main page without logging in

### 3.1.1.5 Use Case

Login

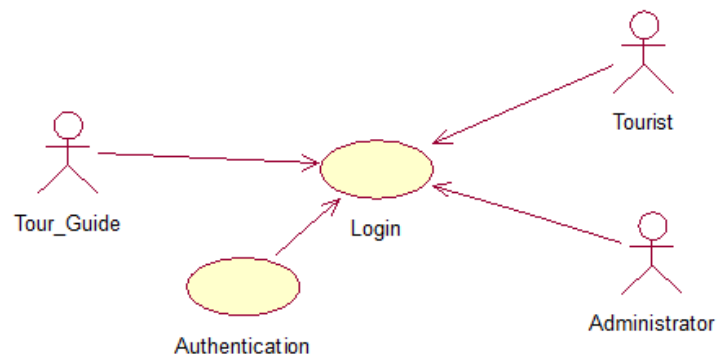
#### Scope

The scope of this use case is for a user of the web application to login to the web application to be able to gain further access.

#### Description

This use case describes the process of users logging in from the home page.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a Tour guide or tourist is on the home page and wants to login

#### Main flow

1. The system identifies the users on the web app.
2. The Tour guide / tourist can see the main page.
3. The system won't allow the users to gain further access.
4. The Tour guide / tourist have to login to gain further access.
5. The Tour guide / tourist will click the login button on the home page.
6. The system will allow the user to enter details already stored on database.  
(See A1 – Not registered)
7. The Tour guide / tourist will enter details they've previously entered in the registration stage.
8. The system will redirect the user to the next page – where tours are displayed – giving that user access.

#### **Alternate flow**

A1: <Not Registered>

1. The system indicates that the information entered is not valid – User not registered
2. The Tour guide / tourist will have to click on register instead of login and register – follow requirement 1.
3. The system checks newly entered credentials and will accept if correct.
4. Continue to main flow point 8.

#### **Exceptional flow**

-

#### **Termination**

The system stores all valid information into the database and will redirect users to the next page – where tours and displayed.

#### **Post condition**

The system goes into a wait state

## Requirement 3: Create Tour

### 3.1.1.6 Description & Priority

This use case describes the Creation of a Tour on the Web Application. Important use case as no tours will be displayed on the Web Application without this function.

### 3.1.1.7 Use Case

Create Tour

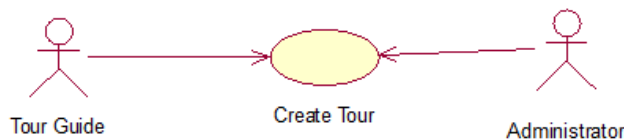
#### Scope

The scope of this use case is for a Tour guide to create a tour to display for tourists to see.

#### Description

This use case describes the process of users creating a tour on the web application.

#### Use Case Diagram



#### Flow Description

##### Precondition

The system is idle

##### Activation

This use case starts when a Tour guide clicks to create a tour.

##### Main flow

1. The system verifies that the user has logged in
2. Tour guides clicks to create a tour
3. Systems redirects user to the Create tour page
4. The Tour guide now has to enter a tour – The attraction name, the location, how many available slots there is available, the price and when the tour is.

5. The system will check that there is sufficient information entered into the creation form. (See A1 – not valid information entered)
6. The form will be accepted by the system and then stored within the encrypted database.
7. The information will be displayed on the page users see when they login.

### **Alternate flow**

A1: <not valid information entered>

1. The system indicates that the information entered is not valid – Field not filled in correctly
2. Tour guide will have to see which field is invalid and enter valid information for the tour to be displayed.
3. If correct information is entered, continue to main flow 6.

### **Exceptional flow**

-

### **Termination**

The system stores all valid information into the database. The tour will now be displayed through the table which is connected through the database.

### **Post condition**

The system goes into a wait state.

## Requirement 4: Search Tour

### 3.1.1.8 Description & Priority

This use case describes the Search of a Tour on the Web Application. Important use case as the tourists will have to be able to search for tours that are available.

### 3.1.1.9 Use Case

Search Tour

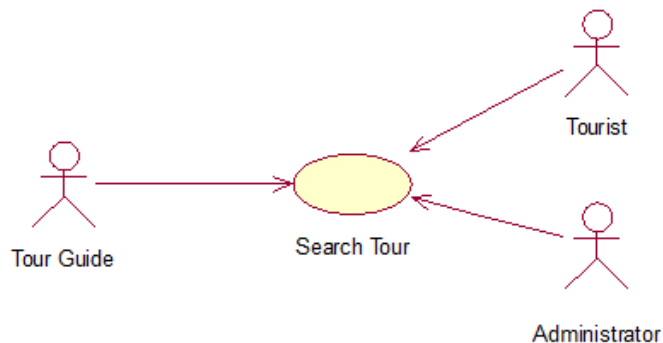
#### Scope

The scope of this use case is for a user of the web application to search for tours available.

#### Description

This use case describes the process of users searching for tours available on the Web Application.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a Tour guide accesses the search feature.

### **Main flow**

1. The system verifies that the user has logged in
2. The user can now see the tours table displayed, to the side of this is the search feature.
3. The user can search through location & category through dropdown menus (See A1 - No tours available).
4. The system will display tours available through the encrypted database that match the criteria.
5. The user can now see tours available to them and can pick which one they want

### **Alternate flow**

A1: <No tours available>

1. The system indicates that there are no tours available to the user matching the criteria entered.
2. No tours will display here.
3. The User will enter in new criteria to search for.
4. If tours matching the criteria entered are available, continue to main flow 4.

### **Exceptional flow**

-

### **Termination**

The system displayed tours that are stored in the database.

### **Post condition**

The system goes into a wait state



## Requirement 5: Location – Google Maps API

### 3.1.1.10 Description & Priority

This use case describes the Locations of the tours being displayed on the Web Application through a google maps API. Important use case as the tourists will be able to see where tours are available around the country.

### 3.1.1.11 Use Case

Location – Google Maps API

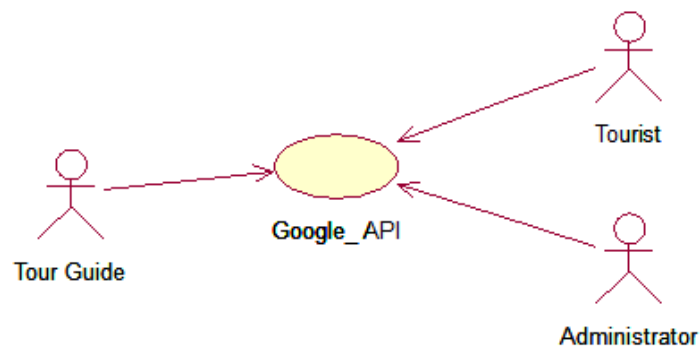
#### Scope

The scope of this use case is for a user of the web application to be able to see a marker, which represents the location of tours available.

#### Description

This use case describes the process of users interacting with the Google maps API to see locations of available tours.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a Tour guide accesses the search feature.

### **Main flow**

1. The system verifies that the user has logged in
2. The system connects to database to display locations of tours through the google maps API. (See A1 – tours must be in database to display)
3. The user can see a map with pins displayed on it.
4. The user can interact with the map to see which tours are at which locations by clicking the pins
5. The system will redirect users to the specific tour's page, if the user double clicks.
6. The system will update with new pins if new tours are added.

### **Alternate flow**

A1: < tours must be in database to display>

1. The system must check to see that there are tours to display within the google maps API.
2. If there are no tours available, nothing will display within the map.
3. If there is tours available, continue to main flow point 3.

### **Exceptional flow**

-

### **Termination**

The system displayed tours that are stored in the database through the API.

### **Post condition**

The system goes into a wait state

## Requirement 6: Contacting users

### 3.1.1.12 Description & Priority

This use case describes how tourists and tour guides can contact each other.

### 3.1.1.13 Use Case

Contacting a Tour users

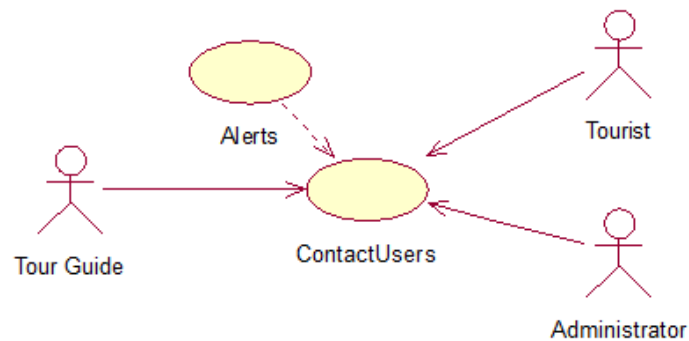
#### Scope

The scope of this use case is for an alert to happen if a user contacts another

#### Description

This use case describes the process of users interacting with other users

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a Tourist shows interest in a tour and decides to contact a tour guide.

#### Main flow

1. The system verifies that the user has logged in
2. The tourist will click attend on a tour they have interest in to contact the tour guide. (See A1 – Different contact method)
3. The tourist will then contact the tour guide from within the Web Application through an encrypted message box.

### **Alternate flow**

A1: < Different contact method>

1. The system will check if the tourist has clicked attend.
2. The tourist may not want to click attend without further information – user can email the tour guide to find out more information.
3. The user will receive more information from the tourist and will continue to main flow point 2 if they are happy to attend.
4. Terminate otherwise.

### **Exceptional flow**

-

### **Termination**

The system will allow encrypted messaging between tour guides and attendees of the tour.

### **Post condition**

The system goes into a wait state

## Requirement 7: Function Level Access Control

### 3.1.1.14 Description & Priority

This use case describes the restriction of access a user has on the web application. Important use case as users cannot get past the main page without registering.

### 3.1.1.15 Use Case

Function Level Access Control

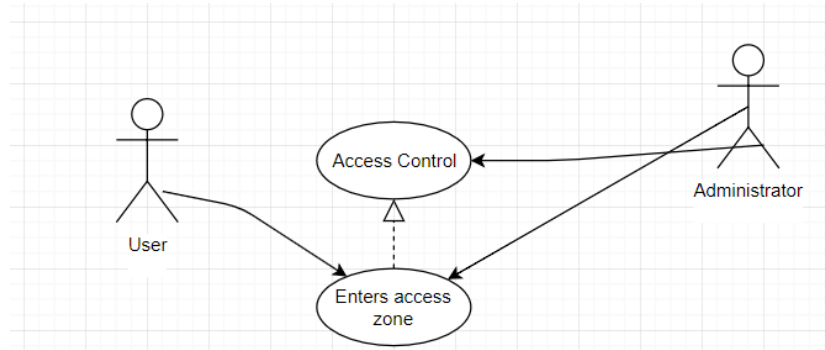
#### Scope

The scope of this use case is for a user of the web application not be able to gain access further than they should be able to without logging in.

#### Description

This use case describes Function Level Access Control within Tours2Connect.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

This use case starts when a user wants to gain further access than main page.

### **Main flow**

1. The system identifies the users on the web app as not logged in.
2. The user can see the main page.
3. The system won't allow the users to gain further access.
4. The user tries to gain further access without logging in. (See A1 – Access level Controls)
5. The user cannot gain further access past index.php & register.php/login.php.
6. User can gain access, when they have created an account with the access to do so.

### **Alternate flow**

A1: <Access level Controls>

1. The system will not allow access to any pages apart from index.php & register.php / login.php.
2. The Tour guide / tourist will have to leave the web application or create an account.
3. The system will register the user if they wish to create an account.
4. Continue to main flow point 5.

### **Exceptional flow**

-

### **Termination**

The system stores all valid information into the database and will allow the correct access for the correct user, if they choose to create an account.

### **Post condition**

The system goes into a wait state

## Requirement 8: PayPal Payment

### 3.1.1.16 Description & Priority

This use case describes payment process of a user looking to book a place on the tour.

### 3.1.1.17 Use Case

PayPal Payment.

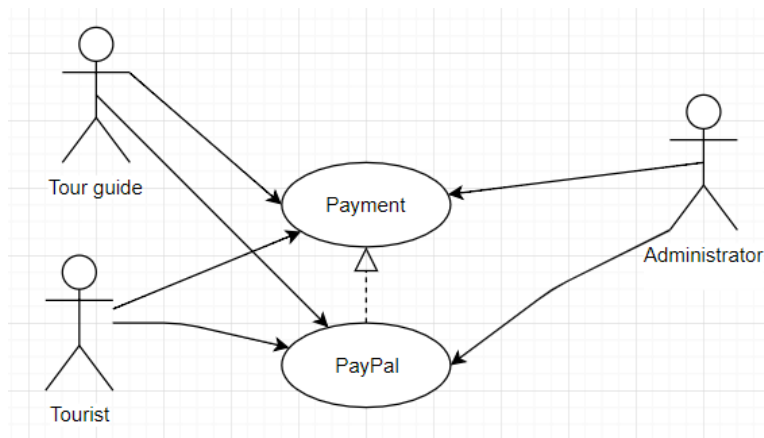
#### Scope

The scope of this use case is for a user of the web application select a tour they wish to attend and pay for the selected tour.

#### Description

This use case describes the process of a user paying for a tour.

#### Use Case Diagram



#### Flow Description

#### Precondition

The system is idle

#### Activation

The use case starts when a user wishes to pay for a tour displayed.

### **Main flow**

1. The system identifies the users on the web app.
2. The user selects a tour from the table displaying tours.
3. The system will redirect the user to that specific tours page.
4. The user can then see that tours information.
5. The user will then select to pay via button.
6. The PayPal page will display upon the click.
7. The Tour guide / tourist will enter their payment details - PayPal. (See A1 – Invalid credentials/funds)
8. The system will redirect the user to the tours page, where they can browse again.

### **Alternate flow**

A1: <Invalid credentials/funds>

1. The system indicates that the information entered is not valid – prompts for correct details
2. The user will have to enter valid details to be able to process further.
3. The system checks newly entered credentials and will accept if correct.
4. Continue to main flow point 7.

### **Exceptional flow**

-

### **Termination**

The PayPal system will take care of all credentials entered.

### **Post condition**

The system goes into a wait state



## **3.2 Non-Functional Requirements**

### **3.2.1 Security requirement**

The administrator is the only person to have access to changing the functions of the web application. The admin will be in control of all security features within the Web Application. The only entry features a user will be able to access is the login/register and the create a tour form. Everything else within the web application, doesn't require a user input.

Information inputted through the web application such as user's login credentials and passwords will be held and hashed through the MySQL database. All tour information will be also held in the database to be stored correctly.

All payments will be taken through PayPal and therefore will be securely taken care of. This means that Tours2Connect doesn't handle any payment information and will not store unnecessary data, which could lead to security issues.

Goal of tours2Connect is to create a fully functional secure web application and with these security requirements I believe it is possible.

Prevent cross site scripting, XSS enables attackers to inject client-side scripts into web pages viewed by other users

Make sure that sessions end when users logout.

### **3.2.2 Performance/Response time requirement**

We require a level of Performance/Response as it is a Web application that will be held live through Heroku. Performance is very important to the project as users will be expecting and relying on a fast responding Web application. If the application is not fast and responsive, then users will grow frustrated and leave the web application half way through their visit.

As there will be a google maps API embedded into the web application, we need a fast performance time because if there isn't, the area where the map is will load after the rest of the page, looking unprofessional. Users will want not just the API to display fast but the whole site to respond in a timely fashion.

As there is a lot of PHP and SQL commands within the code to make the web application fully functional, this is where the performance needs to be up to standard, as loading times for the data within the database to load on the web application would cause users to grow frustrated and possibly make them leave the web application all together.

Overall, performance and response times are a very big requirements to ensure users have a good experience while browsing Tours2Connect, and that they will enjoy their experience.

### **3.2.3 Availability requirement**

The web application's availability will be a very important requirement as users will be expecting that they can access our web page from any device that can browse the web. It will be hosted off of Heroku, therefore will be live on the world wide web. People will be able to visit the Tours2Connect site on the move and this will help the popularity of the Website. Users will expect the web application to be available constantly so that they can use the service as they wish. When the user has access to the web, the application will be fully functional to them. For them to gain further access, they will have to create an account.

If errors appear while the web application is live, they must be repaired straight away and removed to make sure the service remains up and running for all users.

### **3.2.4 Recover requirement**

A lot of our information and data will be stored within an SQL database. This database is accessed through the cloud9 workspace for Tours2Connect. The entire project will be stored and backed up to GitHub where files will be kept if there happens to be an issue. Therefore, they are recoverable. Throughout

development, code has been pushed from cloud9 to GitHub. The account being used for the recovery is: <https://github.com/shanemurphy1996>

```
git remote add origin https://github.com/shanemurphy1996/Tours2Connect.git|
git add .
git commit -m "Example Commit for Tours2Connect"

git push -u origin master
```

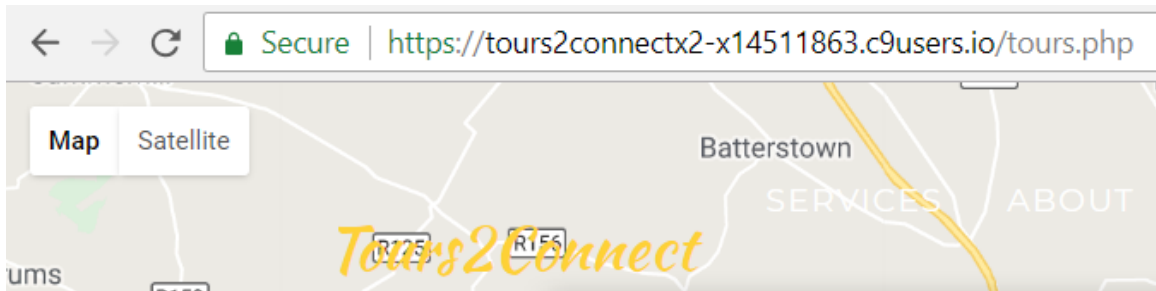
### **3.2.5 Robustness requirement**

Robustness is the ability of a computer system to cope with errors during execution.

The aim of Tours2Connect is to have the ability to deliver service that can justifiably be trusted as a software system. If one part of the Web application isn't working correctly, this shouldn't cause the whole web application to crash i.e. If the google maps API doesn't load during execution, but the rest of the web application runs without any issues, the robustness of the project is good, and this will be the aim.

Robustness has been looked at throughout the entire web application, ensuring that all page redirects are working correctly, that the google API loads correctly, that all tours display within the tour table, and that when a user selects to 'learn more' about a tour, that that tours information loads. Other issues looked at, are the issues that allow for a user to register and login with the same account that they create upon requesting access to Tours2 Connect.

Below you can see that a user is registering, the details are accepted to the database, and from here the user can login with these details. When the user logs in, they are redirected to a new page, the Tours page. This is where the google maps API is and as you can see below, it loads perfectly and very fast.



All tours from the database are extracted using PHP and select statements, these are loaded in good time, causing no errors to be caused. This results in the application having no issues with different aspects used throughout the development.

KEEP TRACK OF UP AND COMING EVENTS!

FIRST NAME	LAST NAME	CATEGORY	TOUR NAME	MEETING POINT	REGION	PRICE €	DATE	LEARN MORE
Shane	Murphy	Sport	Croke Park	1 Jones Rd, Drumcondra, Dublin 3	Leinster	11	2018-07-21 13:30:00	<a href="#">Learn More</a>

### 3.2.6 Reliability requirement

Reliability test results should be stated in terms of measurements. Measurements are taken during testing when we are collecting and analysing data about the performance of the software.

As of yet the reliability of tours2Connect is very high with a low failure rate. As you can see in the robustness requirement section above, different aspects of the web application seem to be working fine and reliably. It will be held on Heroku, which I have used for other projects and I have had no issues with it as a hosting service.

If the web application does have many errors and a high failure rate, many users will be unhappy with the web application.

### **3.2.7 Maintainability requirement**

We will need to maintain our site in order to keep it up to date with the latest technology and to keep it running smoothly. If the technology becomes old and outdated, the maintainability of it becomes outdated and old with it. This will turn users off using Tours2Connect. The application was created for an academic purpose as a final year project, but I still plan on developing the web application and making it into a fully up to date web application, with the best of software being used within in.

### **3.2.8 Extendibility requirement**

Overtime, Tours2Connect may need multiple databases to be able to hold excess information, as well as being able to provide users with fast response times etc. There are so many ideas that could be implemented into Tours2Connect to make it a massive tourist attraction, and this is how the idea will extend.

### **3.2.9 Reusability requirement**

A lot of the content is reused throughout the web application such as each tour having its own page, the pages are the exact same other than the different tours information being displayed. This is done through the use of PHP, SQL statements and the extraction of each tour having an ID. The API code could be reused on multiple pages to display all the tours available to the users, on more than just the tours page. As long as the code being reused is bug free, implementation can be quick and easy.

The documentation for the project has also been reused as it will be edited, updated and improved throughout the course of the project.

The design of the application is also very reusable as it would suit many different categories of projects, whether it be for gaming, tourism or economically focused, it could be implemented to fit many different projects. It also has a register and

login page that has certain requirements to be added into each input field on the form, this can be reused in the future.

### 3.2.10 Data requirements

Data requirements will refer to the MySQL database that will store all data entered into the web application. There are usually four types of data items involved in data requirements, these are columns, rows, schemata and the table itself.

When users login / register, their information will be stored in the name, email address and password fields within the database. Private information entered within the web application gets stored in the database and hashed due to the security aspects of the project.

PhpMyAdmin works perfectly with cloud9 where I am developing Tours2Connect, so this is the database in which all my data will be stored.



Welcome to phpMyAdmin

PhpMyAdmin runs on the server 127.0.0.1 via TCP/IP which is linked in from cloud9 when the connection strings are entered to initially create the relationship between both services.

The database server used is an Ubuntu server version 5.5.57. This links in with the apache web server to create a stack.

```
mysql-ctl install
```

Then install phpMyAdmin:

```
phpmyadmin-ctl install
```


After the installation is complete you'll just want to make sure mysql is running once more:

```
mysql-ctl start
```

## Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MySQL
- Server version: 5.5.57-0ubuntu0.14.04.1 - (Ubuntu)
- Protocol version: 10
- User: x14511863@localhost
- Server charset: UTF-8 Unicode (utf8)

## Web server

- Apache/2.4.7 (Ubuntu)
- Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: bf9ad53b11c9a57efdb1057292d73b928b8c5c77 \$
- PHP extension: mysqli 

### 3.2.11 User requirements

If a user wants to use this application, they must meet the following requirements:

**Internet access:** The user will need to have internet access to use the Web Application.

**An Account:** To gain access past the index page, an account has to be created so that they can see tours available and more features.

**Technological Device:** Users must be able to access a device that can search the web to display Tours2Connect.

#### **What the user says they want:**

Created a survey for past colleagues I've worked with to a client's point of view, also got fellow classmates and year members to take the survey.

<https://www.surveymonkey.com/r/JG6DRSH>

28/11/17 – 3 Questions and answers asked to people who were asked the following:

**Q1**  
 What do you want a web application to have?  
 Answered: 2 Skipped: 0

**RESPONSES (2)** TEXT ANALYSIS TAGS

**PAID FEATURE**  
 Text Analysis lets you search and tag comments and see word clouds of frequent words.  
[UPGRADE](#) [Learn more >](#)

Add Tags Filter by Tag

Showing 2 responses

easy to navigate around. simply display clearly show what is available on the website  
 11/28/2017 8:20 PM

Easy to get around.  
 11/13/2017 3:57 PM

**Q2** [Export](#)

Requirements that a Tour Guide Web application must have to make it useful for you  
 Answered: 2 Skipped: 0

**RESPONSES (2)** TEXT ANALYSIS TAGS

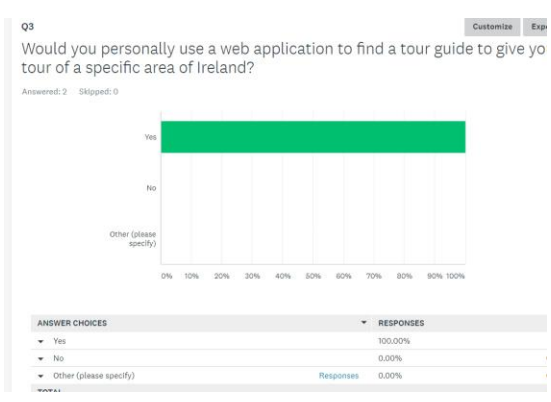
**PAID FEATURE**  
 Text Analysis lets you search and tag comments and see word clouds of frequent words and phrases. To get this feature, upgrade to a paid plan.  
[UPGRADE](#) [Learn more >](#)

Add Tags Filter by Tag Search responses

Showing 2 responses

clear what tours are available. easy method of payment and method to attend  
 11/28/2017 8:20 PM [View respondent's answers](#)

simply website to see whats available  
 11/13/2017 3:57 PM [View respondent's answers](#)



**User must be able to create an account:** Signing in using a username that's not already in use

**Users want easy way of attending tours:** This will be facilitated with through the PayPal payment option.

**Users want a simply displayed web application, that's easy to navigate around:** This can be facilitated by having little content, but exactly what is needed to complete its goal.

Through the survey I asked a question on if people would use a Web application to find a tour guide, the majority of people said yes, they would.



### 3.2.12 Environmental requirements

**Social:** Sharing of an interface. All users are able to see the same interface as the tours will be displayed through a database table. The only difference will be between users who create an account and login as they will see more pages and services than users who do not create an account.

Users will also be able to see interfaces that are just for themselves such as creating a tour, nobody else can access the same creating a tour page as another user as this is just for one user at a time.

Profile page is also for each individual user.

**Organizational:** The security administrator will be top of the hierarchy as that person will control all features of the web application through the code and database access.

The application is developed using PHP, HTML, CSS and more, through cloud9. Cloud9 is the service that offers users the ability to create web applications etc in different languages. It is the environment in which I used to create Tours2Connect. To run the web application off of cloud9, or from outside of cloud9, internet access is required to load it.

### 3.2.13 Usability requirements

- Interface elements e.g. menus should be easy to understand.
- The purpose of the system should be easily understandable, the idea is to create jobs and benefit the tourism industry.
- The system be customisable for different users, such as each users profile page, as well as each tour having their own page available for their own tour.
- Attractiveness of the web application - The screen layout and colour should be appealing and easily used.
- As stated above, the performance should be perfect for all users or else the user will grow frustrated with the slow performance.
- Users require a simple flow within the web application, this would mean being able to go from page to page without being interrupted with errors or slow response times.

Each user to gain access past the index page, will be required to create an account and login. Without an account, the user will not have the ability to use the web application further. It is assumed that the user knows that to gain further access they are required to register and create an account. The forms provided for users are validated so that only email addresses are accepted in the email form field etc.

### **3.3 Design and Architecture**

Tours2Connect is build using a mixture of different languages. PHP, HTML5, CSS, JavaScript are all used throughout the development process. The template used at the beginning was a Bootstrap template to give me a base to work off of. I used a MySQL database to store all the information entered from Tours2Connect. This information is pulled from the database using PHP and SQL statements within the code to display tables of tours, as well as markers on the google maps API. Bootstrap gave me a starting point to the project, in terms of a template and the responsiveness provided by using it, it was a great starting point to the projects development. A google maps API is displayed on my web application using XML and JavaScript. The XML displays information from the database on a separate file by itself and from here, the JavaScript reads the XML and is able to translate it back into readable code, therefore displaying markers on the map. The CSS used was partly from the bootstrap template, but also edited and transformed by myself to result in a colour scheme and look I wanted to display to my users. It also helped to style tables and forms so that they were user friendly. I believe that the CSS that has formed Tours2Connect to look the way it does, creates a great user experience from page to page. The user session is created when a user creates an account and logs in as that account, the session lasts until the timing mechanism of none activity kicks in, or the user ends the session by logging out. PayPal payments are incorporated for the payments within Tours2Connect, this is on the tours page.

#### **3.3.1 Architecture**

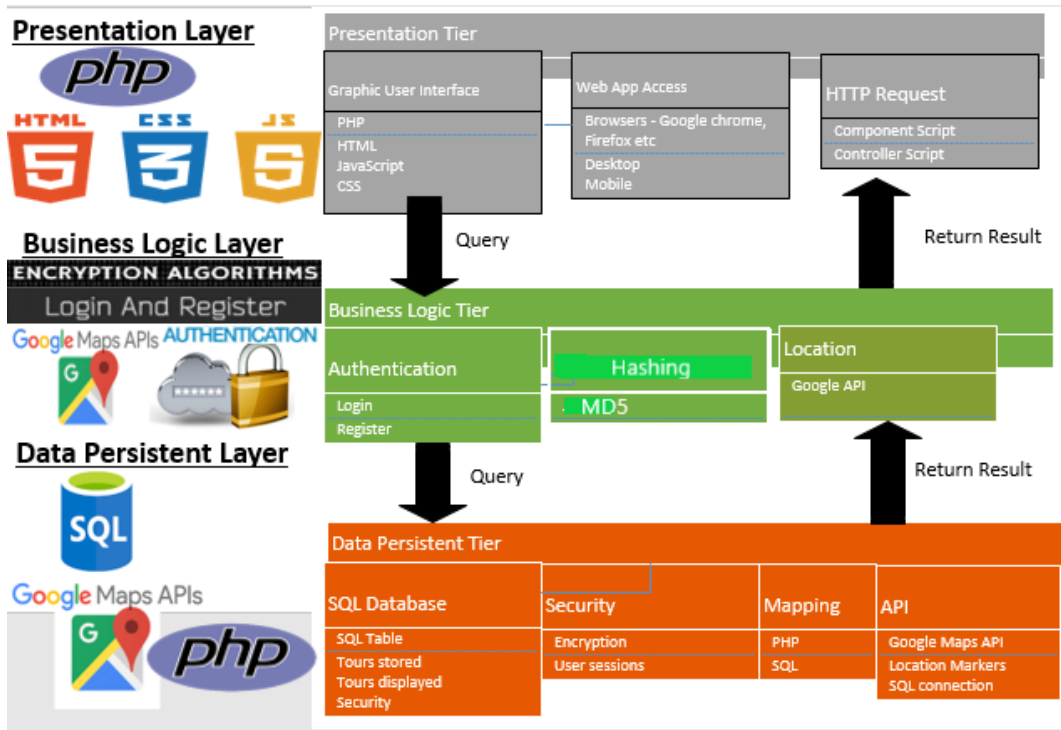
I will be using a 3-tier architecture design. It gives you the ability to update the technology stack of one tier, without impacting other areas of the application.

Presentation layer, Business Logic and the Data Persistent Tier.

Within the presentation tier, we have the languages used to create what is presented, the HTTP requests used to gain access to the database for the markers on the google maps API, and the web application access module, in which the devices or methods to gain access are defined.

The Business logic layer, contains the features that are in between the presentation tier and the data persistent tier. This include the hashing algorithms for the login and registration, aswell as the google maps api that is used to link in with the database, to display the results on the presentation layer.

The data persistent layer is then where the database is. This brings all the PHP functions and SQL statements together. Without this layer, there would be no interaction between the database and the presentation layer. This is how I display tours from my database to the users, by linking all 3 layers together, to create the Tours2Connect that offers users a different tourism experience.



## 3.3.2 Algorithms

### 3.3.2.1 Hashing

The main algorithm within my web application will be the use of Hashing of passwords, and then the validation of the hash to allow users to login with their plaintext passwords. When users enter their details to register, the details will be hashed within the database and then unhashed when the users want to log in, so that the system will recognize their credentials. Below is the code for registering. The user will enter their password into the form, the password is then accepted and hashed before being sent to the database using MD5. These variables are defined on the register.php page and called from register\_handler.php.

MD5 is a message digest algorithm used in multiple different languages. It basically takes an input a message that can be any length and re produces an output of a 128-bit "message digest" of the input. This is how the inputs from the registration fields are digested into a different message and length within the database. The MD5 algorithm is used mainly as a digital signature application, where information must be made smaller in a secure way before being encrypted with a private key from a public key cryptosystem, such as RSA.

```
if(empty($error_array)) {
    $password = md5($password); //Encrypt password before sending to database

    //Generate username by concatenating first name and last name
    $username = strtolower($fname . "_" . $lname);
    $check_username_query = mysqli_query($con, "SELECT username FROM user WHERE username='$username'");
```

The algorithm above is used to hash the password field for the register form. It results in the 'message' being entered as a password, to form into a new message. See below for the message digest of an input.

password  
8d001afc4277e88581ca0f655df488bb  
2ac9cb7dc02b3c0083eb70898e549b63  
dc647eb65e6711e155375218212b3964  
2ac9cb7dc02b3c0083eb70898e549b63  
8d001afc4277e88581ca0f655df488bb  
8d001afc4277e88581ca0f655df488bb  
2ac9cb7dc02b3c0083eb70898e549b63  
863b688d4208d59174766e452f9b6b06  
5f4dcc3b5aa765d61d8327deb882cf99  
8d001afc4277e88581ca0f655df488bb  
5f4dcc3b5aa765d61d8327deb882cf99  
dc647eb65e6711e155375218212b3964  
2ac9cb7dc02b3c0083eb70898e549b63  
5f4dcc3b5aa765d61d8327deb882cf99  
5f4dcc3b5aa765d61d8327deb882cf99

On the login form, the data has to be undigested or unhashed. This is done with a simple line of code. When the user enters their password, this line of code below, gets the password, queries the database for the password, against the email address used to log in. If the password and email match, then the user is allowed to login.

```
$password = md5($_POST['log_password']); //Get password  
  
$check_database_query = mysqli_query($con, "SELECT * FROM user WHERE email='$email' AND password='$password'");  
$check_login_query = mysqli_num_rows($check_database_query);
```

### 3.4 Evolution of Tours2Connect Web Application

#### 1. Questionnaires

These surveys are structured questionnaires to give feedback on what is being asked. Users point of view is essential to success for Tours2Connect. At the beginning of the process, at the prototype stage a survey was completed to see what users would like to see in the web application and how it would suit them.

Another survey was done when the development stage was nearly completed. This survey is asking users to give feedback after they have used the web application. This feedback was taken on board and used to make the web application better.

#### 2. Development

Development for the prototype stage, was merely to get an idea on what I wanted to create, what I thought the idea should be like. After the prototype

stages, I began seeing a bigger and better image of how Tours2Connect should look like. This is what has been created today.

### 3. **System specification evolution**

This would be test cases. Each step in the Business logic layer of the architecture, tests would have to be carried out.

## **3.5 *Implementation***

### **3.5.1 Hashing**

The main algorithm within my web application will be the use of Hashing of passwords, and then the validation of the hash to allow users to login with their plaintext passwords. When users enter their details to register, the details will be hashed within the database and then unhashed when the users want to log in, so that the system will recognize their credentials. Below is the code for registering. The user will enter their password into the form, the password is then accepted

and hashed before being sent to the database using MD5. These variables are defined on the register.php page and called from register\_handler.php.

MD5 is a message digest algorithm used in multiple different languages. It basically takes an input a message that can be any length and re produces an output of an 128-bit "message digest" of the input. This is how the inputs from the registration fields are digested into a different message and length within the database. The MD5 algorithm is used mainly as a digital signature application, where information must be made smaller in a secure way before being encrypted with a private key from a public key cryptosystem, such as RSA.

```
if(empty($error_array)) {
    $password = md5($password); //Encrypt password before sending to database

    //Generate username by concatenating first name and last name
    $username = strtolower($fname . "_" . $lname);
    $check_username_query = mysqli_query($con, "SELECT username FROM user WHERE username='$username'");
```

The algorithm above is used to hash the password field for the register form. It results in the 'message' being entered as a password, to form into a new message. See below for the message digest of an input.

```
password
8d001afc4277e88581ca0f655df488bb
2ac9cb7dc02b3c0083eb70898e549b63
dc647eb65e6711e155375218212b3964
2ac9cb7dc02b3c0083eb70898e549b63
8d001afc4277e88581ca0f655df488bb
8d001afc4277e88581ca0f655df488bb
2ac9cb7dc02b3c0083eb70898e549b63
863b688d4208d59174766e452f9b6b06
5f4dcc3b5aa765d61d8327deb882cf99
8d001afc4277e88581ca0f655df488bb
5f4dcc3b5aa765d61d8327deb882cf99
dc647eb65e6711e155375218212b3964
2ac9cb7dc02b3c0083eb70898e549b63
5f4dcc3b5aa765d61d8327deb882cf99
5f4dcc3b5aa765d61d8327deb882cf99
```

On the login form, the data has to be un digested or unhashed. This is done with a simple line of code. When the user enters their password, this line of code below, gets the password, queries the database for the password, against the email address used to log in. If the password and email match, then the user is allowed to login.

```
$password = md5($_POST['log_password']); //Get password
$check_database_query = mysqli_query($con, "SELECT * FROM user WHERE email='$email' AND password='$password'");
$check_login_query = mysqli_num_rows($check_database_query);
```

### 3.5.2 Google Maps API

My web application will run a Google Maps API within it. The idea is to get markers onto the map that will connect to the locations of tours from within the database. So, if a Tour guide posts a tour that is being held at the Guinness storehouse at St James' Gate, a google map with a marker will be displayed on the API embedded into the web application.

The code below is what makes incorporating a google maps API into Tours2Connect possible. The image to the left, simply customizes markers with specific letters that match the category of the tour. It also creates a map that opens up with specific co-ordinates and at a specific height. The longitude and latitude I chose to specify was just over Dublin, with a zoom level of 11. The image beside it incorporates the xml file, in which the map retrieves tour information from the database. The XML file acts as an intermediary between the database and the map. I have declared all my fields in the database as variables so that the XML can reach them and also read them. The bottom two paragraphs of code, 'var infowincontent' and below, act as the onclick for the markers that are created on



the map. So, when a user clicks a marker, the user will see the tour name and meeting point of the tour in which it is displaying its location.

```
<body>
<div id="map"></div>

<script>

var customLabel = {
  Sport: {
    label: 'Sp'
  },
  History: {
    label: 'H'
  },
  Adventure: {
    label: 'A'
  },
  Fun: {
    label: 'F'
  },
  Scenic: {
    label: 'Sc'
  }
};

function initMap() {
var map = new google.maps.Map(document.getElementById('map'), {
  center: {lat: 53.33833198, lng:-6.285498858},
  zoom: 11
});
var infoWindow = new google.maps.InfoWindow;
```

```
// Change this depending on the name of your PHP or XML file
downloadUrl('/googleapixml.php', function(data) {
var xml = data.responseXML;
var tour = xml.documentElement.getElementsByTagName('marker');
Array.prototype.forEach.call(tour, function(markerElem) {
var id = markerElem.getAttribute('id');
var first_name = markerElem.getAttribute('first_name');
var last_name = markerElem.getAttribute('last_name');
var category = markerElem.getAttribute('category');
var tour_name = markerElem.getAttribute('tour_name');
var region = markerElem.getAttribute('region');
var price = markerElem.getAttribute('price');
var meetingpoint = markerElem.getAttribute('meetingpoint');
var date = markerElem.getAttribute('date');

var point = new google.maps.LatLng(
parseFloat(markerElem.getAttribute('lat')),
parseFloat(markerElem.getAttribute('lng')));

var infowincontent = document.createElement('div');
var strong = document.createElement('strong');
strong.textContent = tour_name
infowincontent.appendChild(strong);
infowincontent.appendChild(document.createElement('br'));

var text = document.createElement('text');
text.textContent = meetingpoint
infowincontent.appendChild(text);
var icon = customLabel[category] || {};
var marker = new google.maps.Marker({
  map: map,
  position: point,
  label: icon.label
});
```

The XML file I created acts as an intermediary between the database and the map. This is the code below is the xml code. It uses outputs the XML with PHP's echo function. It uses a helper function, the 'parseToXML' when calling upon the echo function to encode special entities correctly, such as "<, >, ?, '" etc. The file basically selects all inputs from the table tour and displays them within an XML table to be read.

```

<?php
require("googleapiconfig.php");

function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$htmlStr);
$xmlStr=str_replace('"','&quot;',$htmlStr);
$xmlStr=str_replace("'",'&#39;',$htmlStr);
$xmlStr=str_replace('&','&amp;',$htmlStr);
return $xmlStr;
}

// Opens a connection to a MySQL server
$conn=mysqli_connect('localhost', $username, $password);
if (!$conn) {
die('Not connected : ' . mysqli_error());
}

// Set the active MySQL database
$db_selected = mysqli_select_db($conn, $database);
if (!$db_selected) {
die ('Can't use db : ' . mysqli_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM tour WHERE 1";
$result = mysqli_query($conn, $query);
if (!$result) {
die('Invalid query: ' . mysqli_error());
}

header("Content-type: text/xml");

// Start XML file, echo parent node
echo "<?xml version='1.0' ?>";
echo "<tours>";

```

```

// Start XML file, echo parent node
echo "<?xml version='1.0' ?>";
echo "<tours>";
$ind=0;
// Iterate through the rows, printing XML nodes for each
while ($row = @mysqli_fetch_assoc($result)){
// Add to XML document node
echo '<marker ' ;
echo 'id="' . $row['id'] . ' ' ;
echo 'first_name="' . parseToXML($row['first_name']) . ' ' ;
echo 'last_name="' . parseToXML($row['last_name']) . ' ' ;
echo 'category="' . parseToXML($row['category']) . ' ' ;
echo 'tour_name="' . parseToXML($row['tour_name']) . ' ' ;
echo 'region="' . parseToXML($row['region']) . ' ' ;
echo 'price="' . parseToXML($row['price']) . ' ' ;
echo 'meetingpoint="' . parseToXML($row['meetingpoint']) . ' ' ;
echo 'date="' . parseToXML($row['date']) . ' ' ;
echo 'lat="' . $row['lat'] . ' ' ;
echo 'lng="' . $row['lng'] . ' ' ;

echo '>';
$ind = $ind + 1;
}

// End XML file
echo "</tours>";
?>

```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" ?>
<tours>
  <marker id="1" first_name="Shane" last_name="Murphy" category="Sport" tour_name="Croke Park" region="Leinster" price="11" meetingpoint="1 Jones Rd, Drumcondra, Dublin 3" date="2018-07-21 13:30:00" lat="53.360710" lng="-6.251209"/>
  <marker id="2" first_name="Seamus" last_name="Cowan" category="Sport" tour_name="Thomond Park" region="Munster" price="15" meetingpoint="10 Cratloe Rd, Limerick" date="2018-10-19 18:00:00" lat="0.000000" lng="0.000000"/>
  <marker id="3" first_name="Samantha" last_name="Harding" category="History" tour_name="Killarney Castle" region="Munster" price="22" meetingpoint="Road Road, Ross Island, Killarney, Co. Kerry" date="2018-06-01 18:00:00" lat="52.040558" lng="-9.542500"/>
  <marker id="6" first_name="Sean" last_name="Cowan" category="Adventure" tour_name="Tayto Park" region="Leinster" price="15" meetingpoint="Adrian Geraghty Joinery, Kiltrew, Ashbourne, Co. M" date="2018-05-02 14:00:00" lat="53.545460" lng="-6.469362"/>
  <marker id="7" first_name="Seamus" last_name="Murphy" category="Adventure" tour_name="Guided Tour of the Zoo" region="Leinster" price="0" meetingpoint="The Phoenix Monument, Phoenix Park, Dublin 8" date="2018-09-20 11:00:00" lat="53.357800" lng="-6.334500"/>
  <marker id="13" first_name="James" last_name="Redmond" category="Adventure" tour_name="Pub Crawl of Temple bar - 1 Free Drink" region="Leinster" price="0" meetingpoint="O'Keefe;Connell Bridge, Dublin 2" date="2018-06-21 15:30:00" lat="53.347458" lng="-6.259035"/>
  <marker id="14" first_name="Matt" last_name="Reggs" category="Fun" tour_name="Best bar crawl of your life - Harcourt Street pubs" region="Leinster" price="0" meetingpoint="60 Harcourt St, Saint Kevin's, Dublin 2" date="2018-07-30 20:30:00" lat="53.334427" lng="-6.262743"/>
  <marker id="16" first_name="Ciaran" last_name="Stainton" category="Fun" tour_name="Crazy tour of Finglas" region="Leinster" price="0" meetingpoint="The Full Shilling Pub - 66 Main Street, Finglas, D" date="2018-07-19 20:30:00" lat="53.389503" lng="-6.299699"/>
  <marker id="20" first_name="Derek" last_name="Hansane" category="Fun" tour_name="Tour of Dublin city - Includes stops at pubs, hist" region="Leinster" price="0" meetingpoint="BadAss Bar, 9-11 Crown Alley, Temple Bar, Dublin 2" date="2018-06-27 20:30:00" lat="53.345318" lng="-6.262915"/>
  <marker id="21" first_name="Shane" last_name="Murphy" category="Adventure" tour_name="NCI Final Year celebration tour" region="Leinster" price="20" meetingpoint="The National College of Ireland" date="2018-05-31 21:30:00" lat="53.348816" lng="-6.243179"/>
  <marker id="22" first_name="Liam" last_name="Scully" category="History" tour_name="Swords castle tour - Swords History!" region="Leinster" price="25" meetingpoint="Swords Village " date="2018-06-13 17:30:00" lat="53.440558" lng="-6.220188"/>
  <marker id="23" first_name="Peter" last_name="Jones" category="Architecture" tour_name="Tour of Dublin" region="Leinster" price="45" meetingpoint="Spire" date="2018-05-22 17:30:00" lat="53.349819" lng="-6.260276"/>
  <marker id="24" first_name="fsf" last_name="fsf" category="History" tour_name="sf" region="Leinster" price="3" meetingpoint="dfd" date="2018-05-03 17:30:00" lat="0.000000" lng="0.000000"/>
  <marker id="25" first_name="Keegan" last_name="Butler" category="Scenic" tour_name="Cliffs of Moher" region="Leinster" price="15" meetingpoint="Lislorcan North, Lisconnor, Co. Clare" date="2018-06-08 17:30:00" lat="52.969410" lng="-9.426719"/>
  <marker id="26" first_name="Darryl" last_name="O'Keefe; Meara" category="History" tour_name="Rock of Cashel" region="Connaught" price="10" meetingpoint="Moor, Cashel, Co. Tipperary" date="2018-06-07 17:30:00" lat="52.519218" lng="-7.887713"/>
  <marker id="27" first_name="Peter" last_name="Maycock" category="Sport" tour_name="Dublin Mega Tours" region="Leinster" price="500" meetingpoint="The spire" date="2018-05-22 17:30:00" lat="53.409840" lng="-6.168942"/>
</tours>

```

So, when this is all done, we return to the tours page to the rest of the google maps API code, where it then downloads the URL, googleapixml.php, which is stated in the screenshots above. The call-back function below, starts the 'downloadURL' function depending on the size of the XML file in question. The bigger it is, the longer you're going to have to wait for the function to be complete. When the downloadURL function is defined, it calls it from your code and gives in the name of your PHP file and the call back function. The request then 'GET's' the information and displays it for users.

```

function downloadUrl(url, callback) {
  var request = window.ActiveXObject ?
    new ActiveXObject('Microsoft.XMLHTTP') :
    new XMLHttpRequest;

  request.onreadystatechange = function() {
    if (request.readyState == 4) {
      request.onreadystatechange = doNothing;
      callback(request, request.status);
    }
  };

  request.open('GET', url, true);
  request.send(null);
}

function doNothing() {}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDKzbCU0aHVZQPu4kvEWSdnmsM9je3ywUs&callback=initMap">
</script>

```

To use a google maps API, you must register for an account on the Google API Console, here you can get a google maps API key and add it to your project.

### 3.5.3 Login

The login function for my web application entails in my form linking to my login handler page. The below image is the form, which requires an email address and

a password that have already been submitted through the register page. Here we give the email and the password fields a name of 'log\_email' and 'log\_password'.

```
<div class="login_box">
  <div class="login_header">
    <h1>Tours2Connect</h1>
    Login or sign up below!
  </div>
  <br>
  <div id="first">
    <form action="register.php" method="POST">
      <input type="email" name="log_email" placeholder="Email Address" value="<?php
      if(isset($_SESSION['log_email'])) {
        echo $_SESSION['log_email'];
      }
      >" required>
      <br>
      <input type="password" name="log_password" placeholder="Password">
      <br>
      <?php if(in_array("Email or password was incorrect<br>", $error_array)) echo "Email or password was incorrect<br>"; ?>
      <input type="submit" name="login_button" value="Login">
      <br>
      <a href="#" id="signup" class="signup">Need an account? Register here!</a>
    </form>
  </div>
</div>
```

The button in the login page, has a name of 'login\_button', which is called upon in the login handler page. When a user presses this button, it triggers the handling page for logins and you can see the process below. The email is sanitized so make sure that it isn't a fatal message. This can protect use against cross site scripting attacks as that field is sanitized. The email is then stored into the user's session and the password is checked, an SQL query is executed selected the email and passwords from the table users, if the inputs are found and accepted, the user will be sent to the next page – tours.php

```
<?php
if(isset($_POST['login_button'])) {

    $email = filter_var($_POST['log_email'], FILTER_SANITIZE_EMAIL); //sanitize email

    $_SESSION['log_email'] = $email; //Store email into session variable
    $password = md5($_POST['log_password']); //Get password

    $check_database_query = mysqli_query($con, "SELECT * FROM user WHERE email='$email' AND password='$password'");
    $check_login_query = mysqli_num_rows($check_database_query);

    if($check_login_query == 1) {
        $row = mysqli_fetch_array($check_database_query);
        $username = $row['username'];

        $user_closed_query = mysqli_query($con, "SELECT * FROM user WHERE email='$email' AND user_closed='yes'");
        if(mysqli_num_rows($user_closed_query) == 1) {
            $reopen_account = mysqli_query($con, "UPDATE user SET user_closed='no' WHERE email='$email'");
        }

        $_SESSION['username'] = $username;
        header("Location: ../tours.php");
        exit();
    }
    else {
        array_push($error_array, "Email or password was incorrect<br>");
    }
}
?>
```

### 3.5.3 Register

Below is the code for the registration form. It has all the exceptions required for the user for when they are creating a profile. The first name has to be between certain amount of characters, the last name has to be between certain amount of characters, the email entered must match the second email field that is below it etc.

```
<div id="second">
  <form action="register.php" method="POST">
    <input type="text" name="reg_fname" placeholder="First Name" value="<?php
    if(isset($_SESSION['reg_fname'])) {
      echo $_SESSION['reg_fname'];
    }
    ?>" required>
    <br>
    <?php if(in_array("Your first name must be between 2 and 25 characters<br>", $error_array)) echo "Your first name must be between 2 and 25 characters<br>";

    <input type="text" name="reg_lname" placeholder="Last Name" value="<?php
    if(isset($_SESSION['reg_lname'])) {
      echo $_SESSION['reg_lname'];
    }
    ?>" required>
    <br>
    <?php if(in_array("Your last name must be between 2 and 25 characters<br>", $error_array)) echo "Your last name must be between 2 and 25 characters<br>";

    <input type="email" name="reg_email" placeholder="Email" value="<?php
    if(isset($_SESSION['reg_email'])) {
      echo $_SESSION['reg_email'];
    }
    ?>" required>
    <br>

    <input type="email" name="reg_email2" placeholder="Confirm Email" value="<?php
    if(isset($_SESSION['reg_email2'])) {
      echo $_SESSION['reg_email2'];
    }
    ?>" required>
    <br>

    <?php if(in_array("Email already in use<br>", $error_array)) echo "Email already in use<br>";
    else if(in_array("Invalid email format<br>", $error_array)) echo "Invalid email format<br>";
    else if(in_array("Emails don't match<br>", $error_array)) echo "Emails don't match<br>"; ?>

    <input type="password" name="reg_password" placeholder="Password" required>
    <br>
    <input type="password" name="reg_password2" placeholder="Confirm Password" required>
    <br>
    <?php if(in_array("Your passwords do not match<br>", $error_array)) echo "Your passwords do not match<br>";
    else if(in_array("Your password can only contain english characters or numbers<br>", $error_array)) echo "Your password can only contain english characters or numbers<br>";
    else if(in_array("Your password must be between 5 and 30 characters<br>", $error_array)) echo "Your password must be between 5 and 30 characters<br>";

    <input type="submit" name="register_button" value="Register">
    <br>

    <?php if(in_array("<span style='color: #14C800;'>You're all set! Go ahead and login!</span><br>", $error_array)) echo '
    <a href="#" id="signin" class="signin">Already have an account? Sign in here!</a>
    '
  </form>
</div>
/>
```

When the submit button is pressed, the name 'register\_button' is activated in the register handler page. Within the handler page, all variables on the register page are declared so that errors don't occur. All the variables stored in the database, that are mentioned on the register page are checked and ensure that the message entered will not cause issues or errors. All inputs that have spaces are removed, all spaced are removed, and the information entered is stored into the session variable.

```
<?php
//Declaring variables to prevent errors
$fname = ""; //First name
$lname = ""; //Last name
$email = ""; //email
$email2 = ""; //email 2
$password = ""; //password
$password2 = ""; //password 2
$error_array = array(); //Holds error messages

if(isset($_POST['register_button'])){

    //Registration form values

    //First name
    $fname = strip_tags($_POST['reg_fname']); //Remove html tags
    $fname = str_replace(' ', '', $fname); //remove spaces
    $fname = ucfirst(strtolower($fname)); //Uppercase first letter
    $_SESSION['reg_fname'] = $fname; //Stores first name into session variable

    //Last name
    $lname = strip_tags($_POST['reg_lname']); //Remove html tags
    $lname = str_replace(' ', '', $lname); //remove spaces
    $lname = ucfirst(strtolower($lname)); //Uppercase first letter
    $_SESSION['reg_lname'] = $lname; //Stores last name into session variable

    //email
    $email = strip_tags($_POST['reg_email']); //Remove html tags
    $email = str_replace(' ', '', $email); //remove spaces
    $email = ucfirst(strtolower($email)); //Uppercase first letter
    $_SESSION['reg_email'] = $email; //Stores email into session variable
```

The email entered and the 2<sup>nd</sup> email entered are then checked against each other to ensure that they are the same. This is a security feature. If the email is already in use, then the registration doesn't apply, and the user will be prompted to try login again.

```

//email 2
$email2 = strip_tags($_POST['reg_email2']); //Remove html tags
$email2 = str_replace(' ', '', $email2); //remove spaces
$email2 = ucfirst(strtolower($email2)); //Uppercase first letter
$_SESSION['reg_email2'] = $email2; //Stores email2 into session variable

//Password
$password = strip_tags($_POST['reg_password']); //Remove html tags
$password2 = strip_tags($_POST['reg_password2']); //Remove html tags

$date = date("Y-m-d"); //Current date

if($email == $email2) {
    //Check if email is in valid format
    if(filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $email = filter_var($email, FILTER_VALIDATE_EMAIL);

        //Check if email already exists
        $e_check = mysqli_query($con, "SELECT email FROM user WHERE email='$email'");

        //Count the number of rows returned
        $num_rows = mysqli_num_rows($e_check);

        if($num_rows > 0) {
            array_push($error_array, "Email already in use<br>");
        }

    }
    else {
        array_push($error_array, "Invalid email format<br>");
    }
}
else {
    array_push($error_array, "Emails don't match<br>");
}

```



### 3.5.4 Tours Table Extraction

The tours entered by users, are stored into a database. These tours are then displayed on the 'tours.php' page. Below you can see the database connection on that page.

```
?>
<?php

$db_host = 'localhost'; // Server Name
$db_user = 'root'; // Username
$db_pass = ''; // Password
$db_name = 'tours2connect'; // Database Name

$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
if (!$conn) {
    die ('Failed to connect to MySQL: ' . mysqli_connect_error());
}

$sql = 'SELECT *
      FROM tour';

$query = mysqli_query($conn, $sql);

if (!$query) {
    die ('SQL Error: ' . mysqli_error($conn));
}
?>
```

Firstly, a simple html table has to be created for the table to sit into. For this the table headers are created with all the information I want from the database to be displayed.

```
<h1 style="color:#33ff33">Keep Track of up and coming events!</h1>
<table class="data-table">

  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Category</th>
      <th>Tour Name</th>
      <th>Meeting Point</th>
      <th>Region</th>
      <th>Price €</th>
      <th>Date</th>
      <th> Learn More</th>
    </tr>
  </thead>
  <tbody>
```

After this is done, I then create the rows for which the data will be echoed out onto. Below you can see where I have created the table columns and echoed out the information for each row from the database. Each line is a different heading from the database and here will display a tour on each row of the table.

```
<?php
while ($row = mysqli_fetch_array($query))
{
    echo '<tr>
        <td>'.$row['first_name'].'</td></button>
        <td>'.$row['last_name'].'</td>
        <td>'.$row['category'].'</td>
        <td>'.$row['tour_name'].'</td>
        <td>'.$row['meetingpoint'].'</td>
        <td>'.$row['region'].'</td>
        <td>'.$row['price'].'</td>
        <td>'.$row['date'].'</td>
        <td><button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal"> <a href=tourinfo.php?'.$row[id].'> Learn More</a></td>
    </tr>';
}
}}>
```

This is what it looks like on the tour.php page on Tours2Connect. Each tour is displayed on a different row, containing 9 columns each. The learn more button brings you to a new page where the information about that page is individually displayed.

VIEW TABLE OF ALL COMING EVENTS

FIRST NAME	LAST NAME	CATEGORY	TOUR NAME	MEETING POINT	REGION	PRICE €	DATE	LEARN MORE
Shane	Murphy	Sport	Croke Park	1 Jones Rd, Drumcondra, Dublin 3	Leinster	11	2018-07-21 13:30:00	<a href="#">Learn More</a>
Sean	Cowley	Sport	Thomond Park	10 Cratloe Rd, Limerick	Munster	15	2018-10-19 18:00:00	<a href="#">Learn More</a>
Samantha	Harding	History	Killarney Castle	Road Road, Ross Island, Killarney, Co. Kerry	Munster	22	2018-06-01 18:00:00	<a href="#">Learn More</a>
Sean	Cowley	Adventure	Tayto Park	Adrian Geraghty Joinery, Kilbrew, Ashbourne, Co. M	Leinster	15	2018-05-02 14:00:00	<a href="#">Learn More</a>

### 3.5.5 Users entering tours to database

Within the create tours page, I start by connecting the database to the page in question. Then I declare all variables that will be requested from the form that is going to allow users to enter their tours. The below 2 images act as the handlers for the create tour form. The method gets rid of all special characters that are entered within the form.

```
<?php
require('config.php');
// If form submitted, insert values into the database.
if (isset($_REQUEST['first_name'])){
    $first_name = stripslashes($_REQUEST['first_name']); // removes backslashes from spent entry
    $first_name = mysqli_real_escape_string($con,$first_name); //escapes special characters in a string

    $last_name = stripslashes($_REQUEST['last_name']); //stripslashes removes back slashes
    $last_name = mysqli_real_escape_string($con,$last_name);

    $category = stripslashes($_REQUEST['category']);
    $category = mysqli_real_escape_string($con,$category);

    $tour_name = stripslashes($_REQUEST['tour_name']);
    $tour_name = mysqli_real_escape_string($con,$tour_name);

    $region = stripslashes($_REQUEST['region']);
    $region = mysqli_real_escape_string($con,$region);

    $price = stripslashes($_REQUEST['price']);
    $price = mysqli_real_escape_string($con,$price);

    $meetingpoint = stripslashes($_REQUEST['meetingpoint']);
    $meetingpoint = mysqli_real_escape_string($con,$meetingpoint);

    $date = stripslashes($_REQUEST['date']);
    $date = mysqli_real_escape_string($con,$date);

    $lat = stripslashes($_REQUEST['lat']);
    $lat = mysqli_real_escape_string($con,$lat);

    $lng = stripslashes($_REQUEST['lng']);
    $lng = mysqli_real_escape_string($con,$lng);
```

After all the variables are declared, there is an SQL statement that simply states, whatever is entered into the create tour form, will be entered into the table 'tour' within the database 'Tours2Connect'. When all variables are entered correctly, then an echo will be prompted stating that the tours details are on the tour.php page, and to follow a link to see all the tours available.

```

//md5 function which will transform the code into encrypted data
$query = "INSERT into `tour` (
first_name,
last_name,
category,
tour_name,
region,
price,
meetingpoint,
date,
lat,
lng) VALUES ({$first_name', '{$last_name', '{$category', '{$tour_name', '{$region', '{$price', '{$meetingpoint', '{$date', '{$lat', '{$lng}');
$result = mysqli_query($con,$query);
if($result){
echo "<div class='container'>
<div class='col-md-4 col-md-offset-4'>
<div class='login-panel panel panel-default'><br/><br/>
<h3> Tour details are now on the Tours page, Follow the link.<br/>Click here to view all <a href='tours.php'>Tours Available</a> </h3></div></div></div>";
}
}
?>

```

### 3.5.6 Extracting tour information separately

Extracting individual tours data was a tough task. I believe that it was one of the harder algorithms to get used to. When the tours are displayed on the page, there is a learn more button, within this button there is a request that states to bring across the row id to the next page in which that tours information should be displayed only.

```

<td> <button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal"> <a href=tourinfo.php?'. $row[id]. '> Learn More</a>

```

The link above brings the user to the 'tourinfo.php' page, but also accesses the rows id number and attempts to bring it across to the next page.

Below you can see that I created a variable that equals a server's query string. This interoperates the initial request from the page before, takes the query string, and where the rows 'id' equals the 'myquery', it will display that rows id. So, when I click on each row of the table, the id from 'tours.php' is brought over to the 'tourinfo.php' page, where the 'myquery' query string is executed and can find and display each id's information.

```

$myquery=$_SERVER["QUERY_STRING"];
echo $myquery;

$sql = "SELECT tour_name FROM tour WHERE id = '$myquery'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
// output data of each row
while($row = $result->fetch_assoc()) {
echo " Welcome To The Tour: " . $row["tour_name"]. "<br>"
}
} else {
echo "0 results";
}
?>

```

Below is the outcome of this function. The ID 1, Croke park is displayed on the page by itself when a user clicks on the 'learn more' button from the table of tours. The same happens for ID 13 below.

**1 Welcome To The Tour: Croke Park  
SEE TOUR DETAILS BELOW!**

**13 Welcome To The Tour: Pub Crawl of Temple bar - 1 Free Drink  
SEE TOUR DETAILS BELOW!**

I then had to create a table for which only the specific tour whose ID is called, will display. The below code creates the table which had headings to the left-hand side and information to the right.

```

<th>First Name:</th>
<td> '.$row["first_name"].'</td>
</tr>
<th>Surname</th>
<td> '.$row["last_name"].'</td>
</tr>
<th>Tour Name:</th>
<td> '.$row["tour_name"].'</td>
</tr>
<tr>
<th>Category</th>
<td> '.$row["category"].'</td>
</tr>
<tr>
<th>Meeting Point</th>
<td> '.$row["meetingpoint"].'</td>
</tr>
<tr>
<th>Price</th>
<td> '.$row["price"].'</td>
</tr>
<tr>
<th>
Attendance:
</th>
<td> '.$row["attendance"].'</td>
</tr>

```

First Name:	James
Surname	Redmond
Tour Name:	Pub Crawl of Temple bar - 1 Free Drink
Category	Adventure
Meeting Point	O'Connell Bridge, Dublin 2
Price	0
Attendance:	0

The individual tour information is called like before to extract that specific ID's information and display in into a table. The code below is how this is done. The SQL statement selects the information that I want to extract and calls from the id where it equals the 'myquery' statement that is declared above.

```

<?php
    $sql = "SELECT first_name, last_name, tour_name, category, meetingpoint, price, date, attendance FROM tour WHERE id ='$myquery'";
    // echo $myquery;

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo

```

### **3.5.5 Technology Overview**

The web application was build using PHP, HTML5, JavaScript and Bootstrap to implement the features I was needed. The bootstrap template will give a base to the web application, and I will edit this content using PHP, CSS and HTML5 to ensure the web application is how I want it to look and user friendly.

PHP is a server-side scripting language; Therefore, it can collect data from forms and generate dynamic content too. It is suited for web development and can also be embedded into HTML. All the forms on the web application, to name a few - Register, Login, Create Tour pages all have forms on them, in which the data is sanitized and then entered into the database. Some of the information is then called upon and echoed on the web application, this includes the logged in user, or every tour that is displayed in a table on the tours' page.

HTML5 is a mark-up language used for structuring and presenting content for web applications, It's the newest version and is linked with the PHP to show the web applications display. Browsers don't display HTML tags, but use them to interpret and structure the content that I have created on all my web pages.

JavaScript is a programming language that allows interactivity on the web application for the users such as error messages. I used this on a lot of my pages, login and register have a function where when a href is clicked, the forms switch. I also used it on my google maps API.

API's are a set of tools and methods, that allow for building software applications in which users can interact with them. The google maps API is one of the most commonly used API's as it can be manipulated to display location data that I have stored in my database.

I created a database and stored all tour information and user information on it. This is stored within PhpMyAdmin. The information in the database is extracted using PHP methods that are used to implemented the body of the tours table.

PhpMyAdmin is created to handle the administration of MySQL databases. This is what I used to store the data entered into the web application.

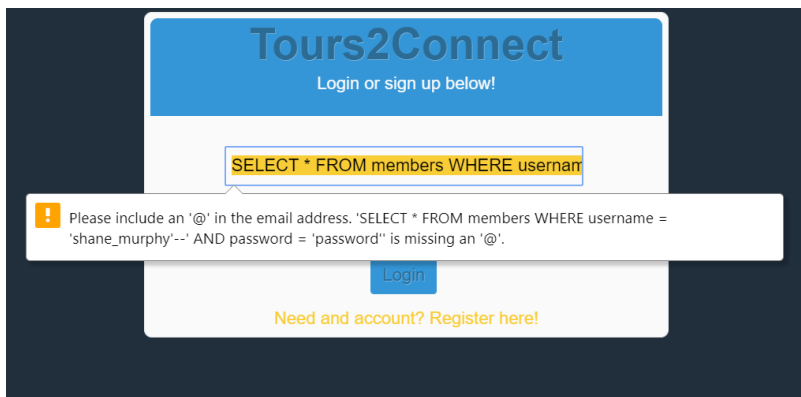
I developed my Web Application through cloud9, and have it hosted on 000WebHost & Heroku. Both have caused major issues with the CSS and database connections.

### 3.5.6 Security

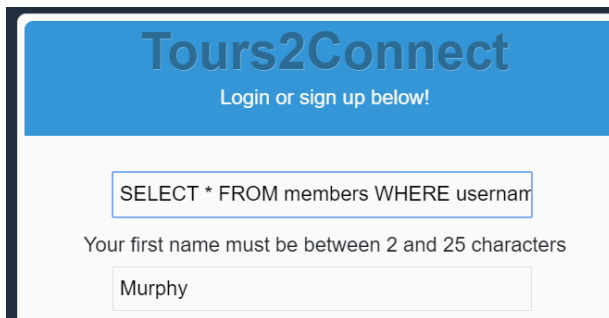
Tours2Connect has to have contact with its hosted database to allow for users to gain access to the functionality of the web application. The user's information is stored in a PhpMyAdmin database which must hash and encrypt the data held inside. At Tours2Connect, we believe that all data should be kept safe and secure, following the guidelines of the GDPR.

#### 3.5.6.1 OWASP No. 1 - injection

Prevention from SQL Injection. Login form below



Registration below



The following code means that an attacker who enters input, CANNOT contain the following characters: ' or "

```
type="text" pattern="[^'\x22]+" title="Invalid input" required/>
```

```
<input class="form-control" id="lng" name="lng" type="text" pattern="[^'\x22]+" title="Cant contain ' Or special characters" required/>
<span class="help-block" id="hint_lng">
  This will display your tour on the map on the tours area.
  Need help with finding your events longitude?
  Find it <a href="https://www.latlong.net/convert-address-to-lat-long.html" style="color:red" target="_blank">here.</a>
</span>
```



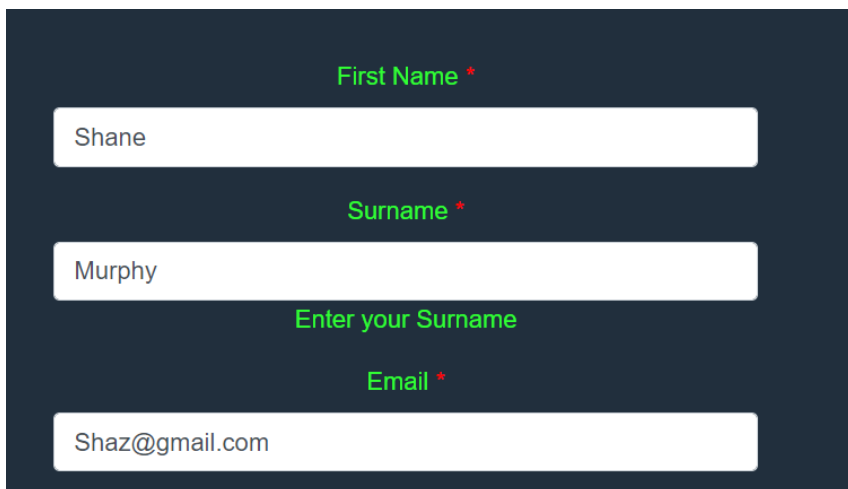
```
<div class="form-group ">
<label class="control-label requiredField" for="price" required>
  Tour Price - €
  <span class="asteriskField">
    *
  </span>
</label>
<input class="form-control" id="price" name="price" type="number" pattern="[^\x22]+" title="Cant contain ' Or special characters" required />
</div>
```

Entering tours to page, doesn't allow special characters on any field, therefore no SQL injection can occur on the page in question.

A screenshot of a web form with a dark background and green text. The form has several input fields, each with a label and a red asterisk indicating it is required. The labels are: "First Name", "Enter your Surname", "Email", "Tour Name", "Tour Price - €", "Meeting Point", and "Latitude of Tour". Each of these fields contains a SQL injection payload: "WHERE username = 'admin'--' AND password = 'password'", "SELECT \* FROM members WHERE username = 'admin'--' /", "SELECT \* FROM members WHERE username = 'admin'--' /", "1", "SELECT \* FROM members WHERE username = 'admin'--' /", "Select the spot in which you want to begin your Tour.", and "SELECT \* FROM members WHERE username = 'admin'--' /". A tooltip with a warning icon is visible over the "First Name" field, containing the text: "Please match the requested format. Cant contain ' Or special characters".

### 3.5.6.2 OWASP – No. 2 Session Management

In PHP, sessions create a method of storing information in variables that can be used or displayed on different pages across your web application. This is different to a cookie session where a user's information is stored to the browser or the device itself. Using sessions helps you to create profile systems, as well as protecting pages from being accessed by none registered users of the web application. Session variables hold information about the logged in user across all pages, hence when the user tries to create a tour, their information is already pre-entered, due to sessions.



A screenshot of a registration form on a dark blue background. The form has three input fields, each with a label above it. The first field is labeled "First Name \*" and contains the text "Shane". The second field is labeled "Surname \*" and contains the text "Murphy". Below the second field is a green prompt "Enter your Surname". The third field is labeled "Email \*" and contains the text "Shaz@gmail.com".

Once the user logs out of Tours2Connect, their session is destroyed, and they are required to login again. The way Tours2Connect is set up is very secure, as users cannot access any pages apart from the login, register or index without logging in.

```
<?php
    session_start();

    if(!isset($_SESSION))
    {
        header('location:tours.php');
        exit;
    }

?>
```

Broken Authentication and Session Management occurs when applications fail to properly protect credentials and session tokens throughout their lifecycle examples of potential exploits are that Session IDs are passed unencrypted, User credentials are passed unencrypted, Poor username and password creation. All of these features are looked at and validated on Tours2Connect by MD5 hashing, and the validation at registration for better security passwords etc.

### 3.5.6.3 OWASP No. 3 Cross Site Scripting (XSS)

XSS enables an attacker to inject scripts into the client side. This can be done through different forms that allow user input. This why validation of all inputs is a must do in a secure web application such as Tours2Connect.

When numbers, characters etc that can be entered into a field, it's a must to sanitize it. Escape data as much as possible on output to avoid XSS and malformed HTML. PHP offers functions for this such as the following for sanitizing email addresses.

```
<?php
if(isset($_POST['login_button'])) {
    $email = filter_var($_POST['log_email'], FILTER_SANITIZE_EMAIL); //sanitize email
    $_SESSION['log_email'] = $email; //Store email into session variable
    $password = md5($_POST['log_password']); //Get password
```

The following code is sanitization for the create tour form. All field inputs escape special characters that can be used for XSS and ensures that they do not reach the database.

```
<?php
require('config.php');
// If form submitted, insert values into the database.
if (isset($_REQUEST['first_name'])){
    $first_name = stripslashes($_REQUEST['first_name']); // removes backslashes from first name entry
    $first_name = mysqli_real_escape_string($con,$first_name); //escapes special characters in a string

    $last_name = stripslashes($_REQUEST['last_name']); //stripslashes removes back slashes
    $last_name = mysqli_real_escape_string($con,$last_name);

    $category = stripslashes($_REQUEST['category']);
    $category = mysqli_real_escape_string($con,$category);

    $tour_name = stripslashes($_REQUEST['tour_name']);
    $tour_name = mysqli_real_escape_string($con,$tour_name);

    $region = stripslashes($_REQUEST['region']);
    $region = mysqli_real_escape_string($con,$region);

    $price = stripslashes($_REQUEST['price']);
    $price = mysqli_real_escape_string($con,$price);

    $meetingpoint = stripslashes($_REQUEST['meetingpoint']);
    $meetingpoint = mysqli_real_escape_string($con,$meetingpoint);

    $date = stripslashes($_REQUEST['date']);
    $date = mysqli_real_escape_string($con,$date);

    $lat = stripslashes($_REQUEST['lat']);
    $lat = mysqli_real_escape_string($con,$lat);
```

Also, the input validation doesn't allow for XSS scripts to be accepted.

First Name \*

Shane

Surname \*

Murphy

Enter your Surname

Email \*

Shaz@gmail.com

Please match the requested format.  
Cant contain ' Or special characters

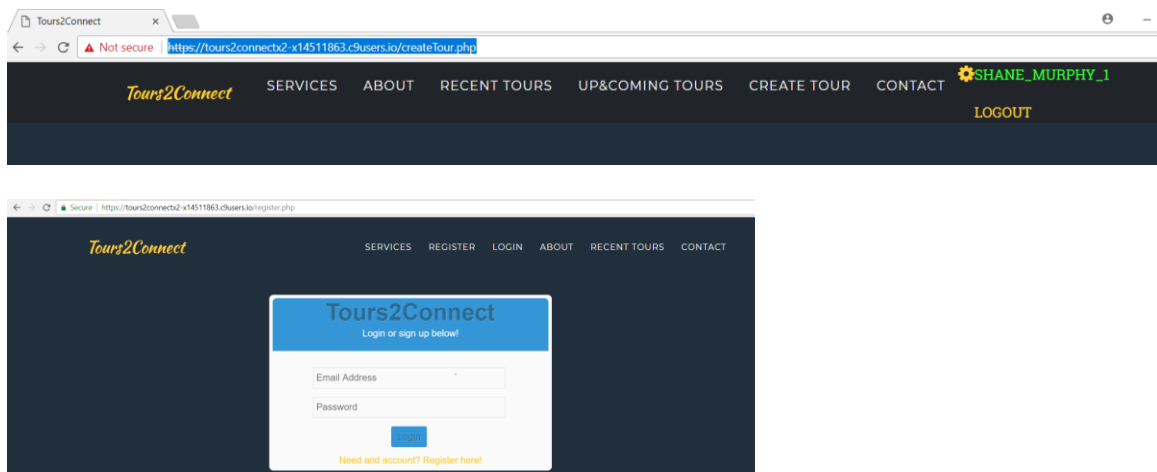
<IMG SRC="javascript:alert('XSS');">

#### 3.5.6.4 OWASP No. 6 - Sensitive Data Exposure

Most web applications don't protect sensitive data, such as credit cards and user credentials. Attackers aim to steal such weakly protected data to conduct credit card fraud or hack an account. Sensitive data should have extra protection such as encryption at rest or in transit, at tours2connect we have hashing. As each user has a different id, by these IDs aren't called upon in the URL, there is no issue with an attacker attempting to access another users account.

### 3.5.6.5 OWASP No. 7 - Access level control

Users must login to gain further access than homepage. This is for security reasons. On Tours2Connect there is no way to bypass the homepage or registration page, without logging in. This is so the functionality hidden beyond the login page cannot be accessed by a malicious user. This kind of attack occurs when an attacker, simply changes the URL or a parameter to a privileged function. This isn't possible on Tours2Connect as pages are not directed to by IDs to privilege pages, only to different tours available. Copying a URL, logging out and pasting the URL, results in being redirected to the registering page.



### 3.5.6.6 Strong User Passwords

A strong password is a must have for secure web applications and their users. This will protect users and Tours2Connect against brute force hacking. Having a very strong password can cause an attacker to grow tired of trying to hack your account. Strong passwords include uppercase, lowercase letters, as well as a series of numbers. This feature is introduced in the registration form for Tours2Connect. A user must have uppercase, lowercase, numbers and at least 8 characters or more to create an account. Strong passwords are rarely cracked or guessed by attackers, so this is a must.

```
<input type="password" name="reg_password" placeholder="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters"
pattern="(?!.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" requi
<br>
<input type="password" name="reg_password2" placeholder="Confirm Password" required>
<br>
<?php if(in_array("Your passwords do not match<br>", $error_array)) echo "Your passwords do not match<br>";
else if(in_array("Your password can only contain english characters or numbers<br>", $error_array)) echo "Your password can only contain english characters or numbers.
else if(in_array("Your password must be between 5 and 30 characters<br>", $error_array)) echo "Your password must be between 5 and 30 characters<br>"; ?>
```

### 3.5.6.7 Password Hashing

Introducing password hashing in PHP is a must have feature of a secure web application. Password hashing is a strong one-way hashing algorithm so that the password cannot be intercepted as plain text when a user tries to login with their password.

This is a major feature in a secure web application due to the fact Tours2Connect is a secure web application and all passwords should be hashed within the database. I have used this method when a new account is created from the register form.

The MD5 algorithm is used mainly as a digital signature application, where information must be made smaller in a secure way before being encrypted with a private key from a public key cryptosystem, such as RSA.

```
if(empty($error_array)) {
    $password = md5($password); //Encrypt password before sending to database

    //Generate username by concatenating first name and last name
    $username = strtolower($fname . "_" . $lname);
    $check_username_query = mysqli_query($con, "SELECT username FROM user WHERE username='$username'");
```

The algorithm above is used to hash the password field for the register form. It results in the 'message' being entered as a password, to form into a new message. See below for the message digest of an input.

```
password
8d001afc4277e88581ca0f655df488bb
2ac9cb7dc02b3c0083eb70898e549b63
dc647eb65e6711e155375218212b3964
2ac9cb7dc02b3c0083eb70898e549b63
8d001afc4277e88581ca0f655df488bb
8d001afc4277e88581ca0f655df488bb
2ac9cb7dc02b3c0083eb70898e549b63
863b688d4208d59174766e452f9b6b06
5f4dcc3b5aa765d61d8327deb882cf99
8d001afc4277e88581ca0f655df488bb
5f4dcc3b5aa765d61d8327deb882cf99
dc647eb65e6711e155375218212b3964
2ac9cb7dc02b3c0083eb70898e549b63
5f4dcc3b5aa765d61d8327deb882cf99
5f4dcc3b5aa765d61d8327deb882cf99
```

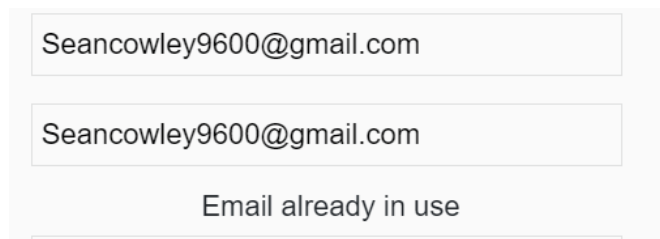
On the login form, the data has to be un digested or unhashed. This is done with a simple line of code. When the user enters their password, this line of code below,

gets the password, queries the database for the password, against the email address used to log in. If the password and email match, then the user is allowed to login.

```
$password = md5($_POST['log_password']); //Get password
$check_database_query = mysqli_query($con, "SELECT * FROM user WHERE email='$email' AND password='$password'");
$check_login_query = mysqli_num_rows($check_database_query);
```

### 3.5.6.8 Email registered

Emails have to be in email format, all emails are then checked against the database to ensure that there are no email addresses that are the same.



Seancowley9600@gmail.com

Seancowley9600@gmail.com

Email already in use

first_name	last_name	username	email	password
Sean	Cowley	sean_cowley	Seancowley9600@gmail.com	8d001afc4277e88581ca0f655df488bb

The code used to search for already created email addresses is an SQL query. This is shown below. It checks for valid email being entered, and then queries the database for an email address of the same name. This is then accepted or rejected with an error message stating 'email address already in use'.

```
if($em == $em2) {
    //Check if email is in valid format
    if(filter_var($em, FILTER_VALIDATE_EMAIL)) {
        $em = filter_var($em, FILTER_VALIDATE_EMAIL);
        //Check if email already exists
        $e_check = mysqli_query($con, "SELECT email FROM user WHERE email='$em'");
        //Count the number of rows returned
        $num_rows = mysqli_num_rows($e_check);
        if($num_rows > 0) {
            array_push($error_array, "Email already in use<br>");
        }
    }
    else {
        array_push($error_array, "Invalid email format<br>");
    }
}
```



### 3.5.6.9 Users can't have same username

Username are changed if that username is already existing – shanemurphy created, shanemurphy\_1 created, shanemurphy\_2 created etc. This is so sensitive data exposure doesn't occur. As each user has a profile, we do not want the user 'shane\_murphy' to have access to 'shane\_murphy\_1' account. Due to the changing factor of everyone with the same name having a different username, gets rid of the issues of different users seeing each other's personal information.

```
if(empty($error_array)) {
    $password = md5($password); //Encrypt password before sending to database

    //Generate username by concatenating first name and last name
    $username = strtolower($fname . "_" . $lname);
    $check_username_query = mysqli_query($con, "SELECT username FROM user WHERE username='$username'");

    $i = 0;
    //if username exists add number to username
    while(mysqli_num_rows($check_username_query) != 0) {
        $i++; //Add 1 to i
        $username = $username . "_" . $i;
        $check_username_query = mysqli_query($con, "SELECT username FROM user WHERE username='$username'");
    }
}
```

Shane	Murphy	shane_murphy
Shane	Murphy	shane_murphy_1
Shane	Murphy	shane_murphy_1_2

### 3.5.6.10 Session time

I have inserted a JavaScript command, that will log a user out and bring them to the registration page after 50000000, which is approximately 83 minutes. This is a very long time, but for the terms of Tours2Connect and the presentation I have kept this number at a high amount.

```
<script type="text/javascript">
function idleLogout() {
  var t;
  window.onload = resetTimer;
  window.onmousemove = resetTimer;
  window.onmousedown = resetTimer; // catches touchscreen presses
  window.onclick = resetTimer;    // catches touchpad clicks
  window.onscroll = resetTimer;   // catches scrolling with arrow keys
  window.onkeypress = resetTimer;

  function logout() {
    window.location.href = 'register.php';
  }

  function resetTimer() {
    clearTimeout(t);
    t = setTimeout(logout, 50000000); // time is in milliseconds
  }
}
idleLogout();
```

### 3.5.8.11 Server-Side Validation

Server-side validation comes into fruition when inputs submitted by users of Tours2Connect are being sent to the server and validated using PHP on the server itself before completing the submission. This is done a lot on my login and registration forms. Below you can see the code that allows for my login form to validate all user inputs so that there are no issues on the server side.

A screenshot of a registration form with several input fields and error messages. The fields are: First Name (containing 'D'), Last Name (containing 'D'), Email (containing 'D@eircom.net'), Confirm Email (containing 'D@f'), Password, and Confirm Password. Error messages are displayed below the fields: 'Your first name must be between 2 and 25 characters' under the first name field, 'Your last name must be between 2 and 25 characters' under the last name field, 'Emails don't match' between the email and confirm email fields, and 'Your passwords do not match' between the password and confirm password fields. A blue 'Register' button is at the bottom.

```
/tours.php
    array_push($error_array, "Emails don't match<br>");
}

if(strlen($fname) > 25 || strlen($fname) < 2) {
    array_push($error_array, "Your first name must be between 2 and 25 characters<br>");
}

if(strlen($lname) > 25 || strlen($lname) < 2) {
    array_push($error_array, "Your last name must be between 2 and 25 characters<br>");
}

if($password != $password2) {
    array_push($error_array, "Your passwords do not match<br>");
}
else {
    if(preg_match('/[^A-Za-z0-9]/', $password)) {
        array_push($error_array, "Your password can only contain english characters or numbers<br>");
    }
}

if(strlen($password) > 30 || strlen($password) < 5) {
    array_push($error_array, "Your password must be between 5 and 30 characters<br>");
}
```

A screenshot of a password field with a validation error. The field is labeled 'Password - 8 Characters or More' and contains a blue border. Below it is a 'Confirm Password' field. A red error message is displayed above the password field: 'Please match the requested format.'

A screenshot of a registration form with a validation error. The form has a 'Register' button and a link 'Already have an account? Sign in here!'. A red error message is displayed above the password field: 'Please match the requested format.'

The form doesn't allow for characters that aren't numbers or letters, therefore SQL Injection and XSS are prevented.

### **3.5.9 CIA Triad**

This is a model created to guide policies for information security within organizations and business ideas. This consists of the three most crucial components of security.

#### **3.5.9.1 Data Confidentiality**

The principle of data confidentiality means granting least privileges. This principle states that access to information should be granted only where essential to perform functions necessary for that user.

*“Authentication, and Authorization are principles which are achieved through various access and privacy controls that support Confidentiality.”* (Mehta, 2016).

#### **3.5.9.2 Data Integrity**

Data integrity ensures that data is not changed or corrupted during the transmission from source to destination, data should remain accurate and consistent. Information should always be transported over a secure connection, not as simple text, but has a hashed or encrypted value. A practice used in within Tours2Connect that ensures the integrity of examined data is to use the MD5 hashing technique on all personal information or private information a user enters.

#### **3.5.9.3 Data Availability**

Data Availability ensures that information is available when and where it is needed, and that is all. Authorized users should have access to the information they need, and nothing else. Availability is possible when all parts of the system or project in question are working properly, therefore there will be no issues, allowing for information to be available at all times.

## 3.6 Testing

Web application testing is a way of software testing so that the application is fully tested, leaving the application interface and other functionalities being tested for.

### 3.6.1 Security Testing

#### 3.6.1.1 OWASP Zed Attack Proxy

Security testing is done to make sure your web application and code is protected and free from bugs or threats. I performed an OWASP Zed Attack analysis via Kali Linux. This is a web application analysis tool. As you can see below, I have performed a standard test on my web applications URL. The results were very good, all the issues that showed up were to do with the browser (Firefox) or with cloud9 itself.

Untitled Session - 20180512-161650 - OWASP ZAP 2.7.0

File Edit View Analyse Report Tools Online Help

Standard Mode

Quick Start Request Response

Contexts

- Default Context
- Sites

### Welcome to the OWASP Zed Attack Proxy

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web

Please be aware that you should only attack applications that you have been specifically t

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack:

Progress: Actively scanning (attacking) the URLs discovered by the spider

For a more in depth test you should explore your application using your browser or autom

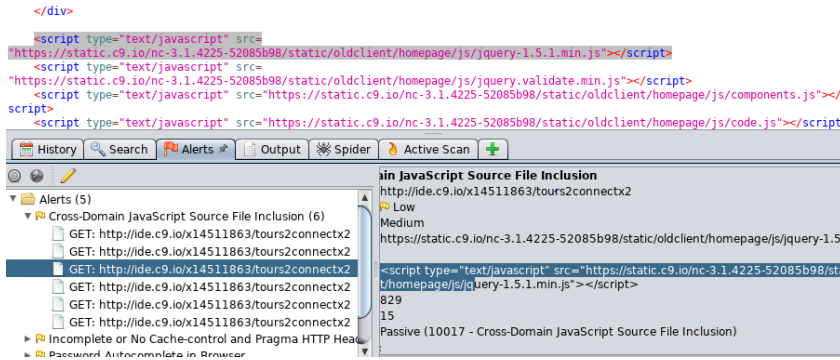
Explore your application:

History Search Alerts Output Spider Active Scan

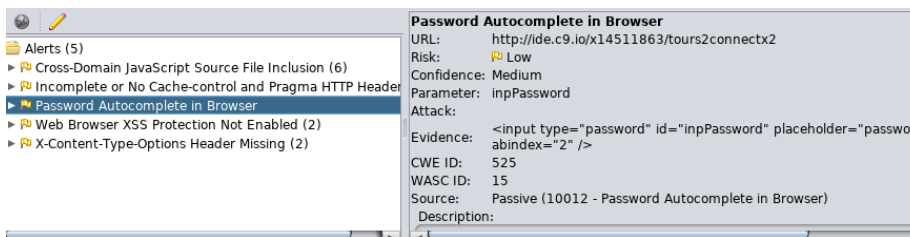
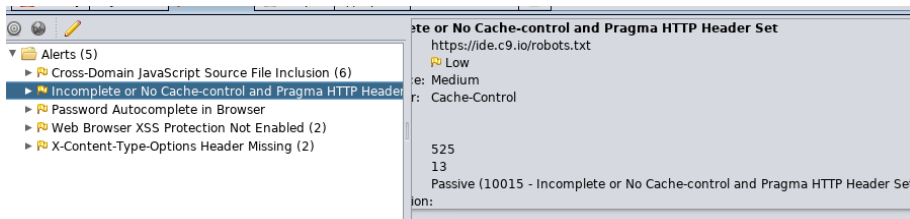
New Scan Progress: 0: http://ide.c9...tours2connect2 5% Current Scans: 1 Num requests: 15 Export

id	Req. Timestamp	Resp. Timestamp	Meth...	URL	Co...	Reason	R...	Size Resp.	Hea...	Size Resp. B...
29	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	6...	853 bytes		3,886 bytes
30	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	7...	855 bytes		3,886 bytes
31	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	9...	853 bytes		3,886 bytes
32	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	7...	857 bytes		3,886 bytes
33	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	7...	857 bytes		3,886 bytes
34	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	3...	853 bytes		3,886 bytes
35	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	1...	859 bytes		3,886 bytes
36	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	7...	863 bytes		3,886 bytes
37	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	6...	849 bytes		3,886 bytes
38	12/05/18 16:1...	12/05/18 16:1...	GET	http://ide.c9.io/x14511863/tours2conn...	200	OK	2...	855 bytes		3,886 bytes

A jQuery inclusion on one the pages was a low risk, but the script is needed for the page to function the way it does. It's a bootstrap script, therefore needed to function the web application.

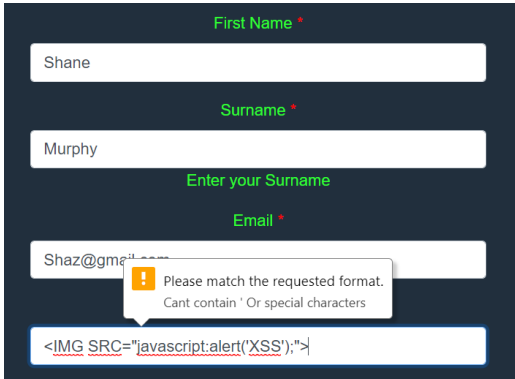


robots.txt page, which is automatically incremented from cloud9 has a low risk also, again it doesn't involve my code for Tours2Connect. Also, a browser issue that auto completes passwords in the browser that was being used, which was Firefox.



### 3.6.1.2 XSS Testing

For security testing I also ensured that Inputs fields protect against SQL injection commands as well as XSS commands. This was done by creating a script to test my inputs against.



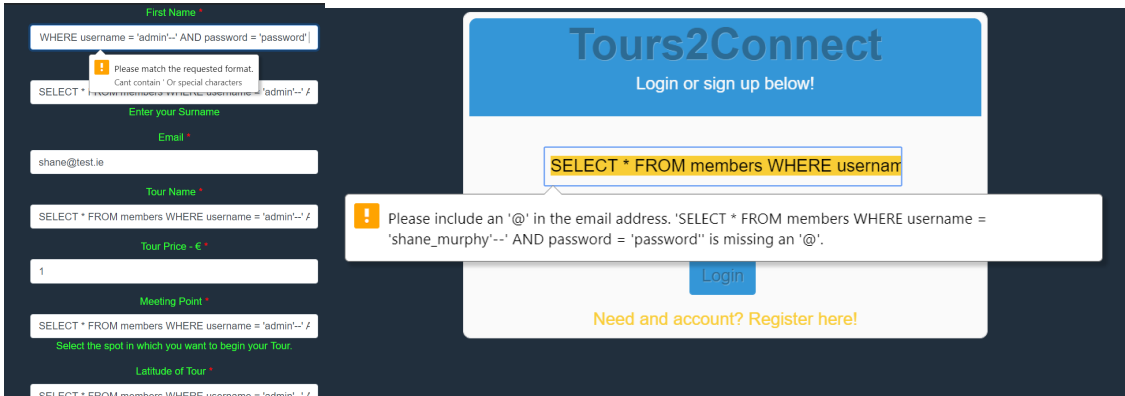
A screenshot of a registration form with the following fields and values:

- First Name: Shane
- Surname: Murphy
- Email: Shaz@gmail.com
- Input field containing the XSS payload: `<IMG SRC="javascript:alert('XSS');">`

An error message is displayed over the email field: "Please match the requested format. Cant contain ' Or special characters".

### 3.6.1.3 SQL Injection Testing

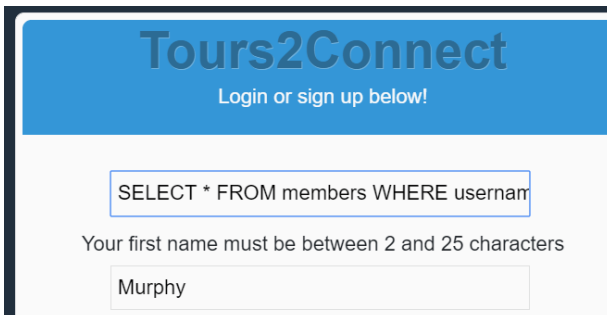
For security testing I also ensured that Inputs fields protect against SQL injection commands as well as XSS commands. This was done by creating a script to test my inputs against.



A screenshot of a registration form with the following fields and values:

- First Name: WHERE username = 'admin'--AND password = 'password'
- Surname: SELECT \* FROM members WHERE username = 'admin'-- /
- Email: shane@test.ie
- Tour Name: SELECT \* FROM members WHERE username = 'admin'-- /
- Tour Price - €: 1
- Meeting Point: SELECT \* FROM members WHERE username = 'admin'-- /
- Latitude of Tour: SELECT \* FROM members WHERE username = 'admin'-- /

An error message is displayed over the email field: "Please include an '@' in the email address. 'SELECT \* FROM members WHERE username = 'shane\_murphy'--' AND password = 'password' is missing an '@'".

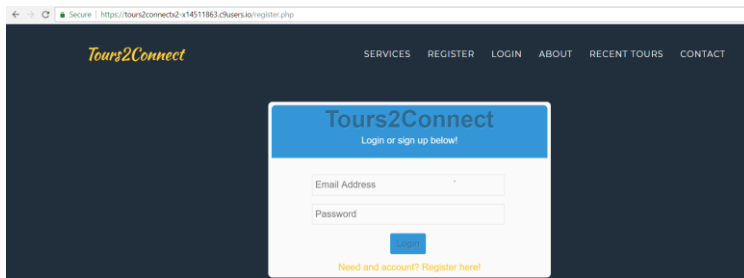
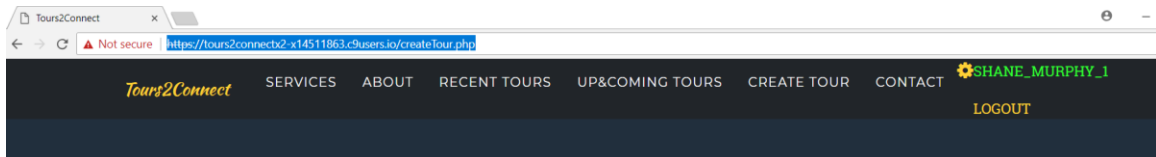


A screenshot of a registration form with the following fields and values:

- Input field containing the SQL injection payload: `SELECT * FROM members WHERE usernam`
- Message: "Your first name must be between 2 and 25 characters"
- Input field containing the value: "Murphy"

### 3.6.1.4 Access Level Control Testing

On Tours2Connect there is no way to bypass the homepage or registration page, without logging in for security reasons. So, I tested this by impersonating an attacker by simply changing the URL or a parameter to a privileged function. This isn't possible on Tours2Connect as pages are not directed to by IDs to privilege pages, only to different tours available. Copying a URL, logging out and pasting the URL, results in being redirected to the registering page.



### 3.6.2 Standard Testing

#### 3.6.2.1 Verification

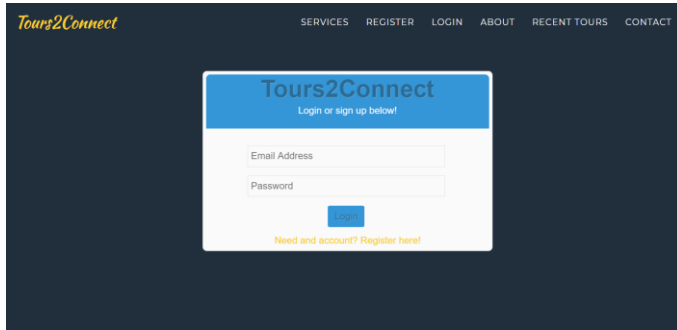
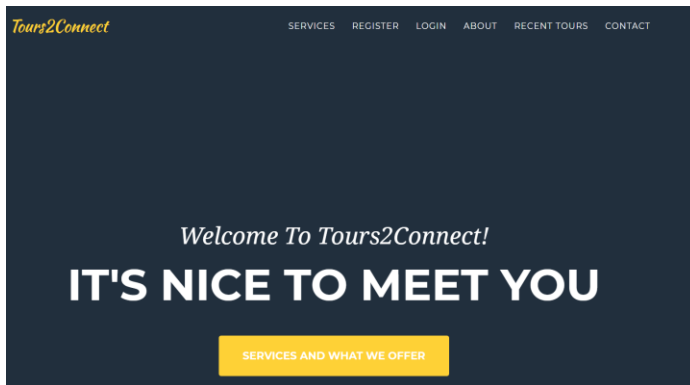
Verification is the process to ensure the application matches and suits the idea and goals at the start of the development stages. It's done to make sure the web application acts the way it's designed to. Verification is conducted to ensure each feature within Tours2Connect behaves the way it should. After testing all the forms and features on the web application, I realised that the create tour didn't protect against SQL injection or XSS attacks. I implemented a feature so that the input fields do not allow " " OR ' " characters. The rest of the features seem to work the way their meant to.



### 3.6.2.2 System Testing

System testing the testing done to ensure that the web application will work and will display all the same features in the exact same way across different environments, in this case, different browsers. This helped me to understand the different layouts of each browser, if there was any. Only difference I could see and was noticeable was the register form was higher up the page on Edge by a small amount. All features worked perfectly, and all functionality was working.

#### Google Chrome



FIRST NAME	LAST NAME	CATEGORY	TOUR NAME	MEETING POINT	REGION	PRICE €	DATE	LEARN MORE
Shane	Murphy	Sport	Croke Park	1 Jones Rd, Drumcondra, Dublin 3	Leinster	11	2018-07-21 13:30:00	<a href="#">Learn More</a>
Sean	Cowley	Sport	Thomond Park	10 Cratloe Rd, Limerick	Munster	15	2018-10-19 18:00:00	<a href="#">Learn More</a>
Samantha	Harding	History	Killarney Castle	Road Road, Ross Island, Killarney, Co. Kerry	Munster	22	2018-06-01 18:00:00	<a href="#">Learn More</a>
Sean	Cowley	Adventure	Tayto Park	Adrian Geraghty Joinery, Kilbrew, Ashbourne, Co. M	Leinster	15	2018-05-02 14:00:00	<a href="#">Learn More</a>
Sean	Murphy	Adventure	Guided Tour of the Zoo	The Phoenix Monument, Pheonix Park, Dublin 8	Leinster	0	2018-09-20 11:00:00	<a href="#">Learn More</a>

Tours2Connect SERVICES ABOUT RECENT TOURS UP&COMING TOURS CREATE TOUR CONTACT SHANE\_MURPHY.J LOGOUT

First Name \*

Surname \*

Enter your Surname

Email \*

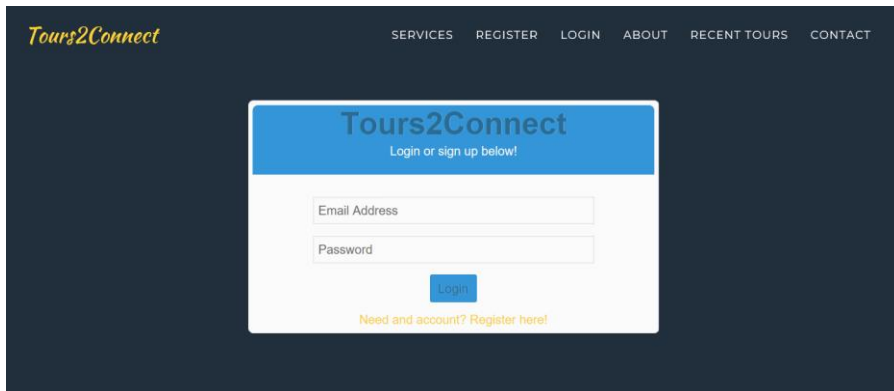
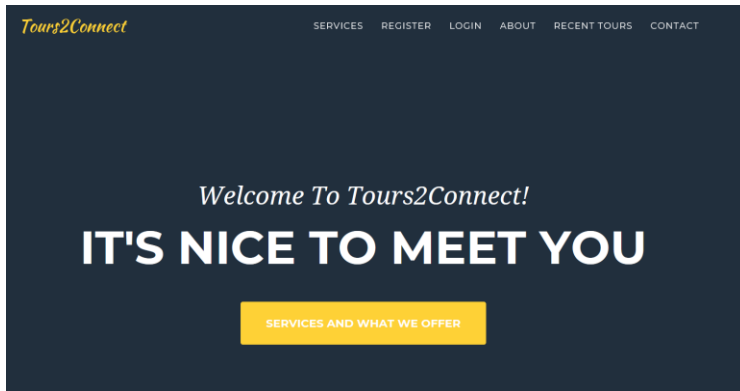
Tour Name \*

Tour Price - € \*

Meeting Point \*

Select the spot in which you want to begin your Tour.

## Microsoft Edge





LIST TOURS OF SHANE MURPHY

FIRST NAME	LAST NAME	CATEGORY	TOUR NAME	MEETING POINT	REGION	PRICE €	DATE	LEARN MORE
Shane	Murphy	Sport	Croke Park	1 Jones Rd, Drumcondra, Dublin 3	Leinster	11	2018-07-21 13:30:00	<a href="#">Learn More</a>
Sean	Cowley	Sport	Thomond Park	10 Cratloe Rd, Limerick	Munster	15	2018-10-19 18:00:00	<a href="#">Learn More</a>
Samantha	Harding	History	Killarney Castle	Road Road, Ross Island, Killarney, Co. Kerry	Munster	22	2018-06-01 18:00:00	<a href="#">Learn More</a>
Sean	Cowley	Adventure	Tayto Park	Adrian Geraghty Joinery, Kilbrev, Ashbourne, Co. M	Leinster	15	2018-05-02 14:00:00	<a href="#">Learn More</a>

Tours2Connect SERVICES ABOUT RECENT TOURS UP&COMING TOURS CREATE TOUR CONTACT SHANE\_MURPHY\_1

LOGOUT

First Name \*

Surname \*

Enter your Surname

Email \*

Tour Name \*

Tour Price - € \*

Meeting Point \*

Select the spot in which you want to begin your Tour.

### 3.6.2.3 Usability Testing

Usability testing ensures that the web application is user friendly. The navigation from page to page brings you to the right destinations and that it is easy to navigate around. I have survey results in 'Customer Testing' below which has users replies and opinions on Tours2Connect. Seven different users were asked to navigate around tours2connect and to use the functionality, then to answer the survey about their experience.

### 3.6.2.4 Integration Testing

Integration testing is simply the stage in which different features of the web application are tested separately and then as a combined unit. This sort of integration testing was performed throughout my project every time a new feature was added to the web application. The main area of the application was linking the Google Maps API to the table which displayed tours, and also linking a database so that both features could read the locations and information of tours.

I added the table first. I added data manually from the database to ensure that the table was echoing data from the database, otherwise the table would have been no use to the integration.

id	int(50)	<input type="text"/>	<input type="text"/>
first_name	varchar(50)	<input type="text"/>	Sean
last_name	varchar(50)	<input type="text"/>	Murray
category	varchar(50)	<input type="text"/>	Adventure
tour_name	varchar(80)	<input type="text"/>	Game of Thrones Tour
region	varchar(50)	<input type="text"/>	Ulster
price	int(50)	<input type="text"/>	17
meetingpoint	varchar(50)	<input type="text"/>	Derry
date	datetime	<input type="text"/>	2018-06-08 16:00:00
lat	float(10,6)	<input type="text"/>	54.983496
lng	float(10,6)	<input type="text"/>	-7.317149

Sean	Murray	Adventure	Game of Thrones Tour	Derry	Ulster	17	2018-06-08 16:00:00
------	--------	-----------	----------------------	-------	--------	----	---------------------

After manual integration worked, a created a form to post its entered information to the table. This would ensure users would be able to enter their tours and other

users would be able to see the tour in question. The map, the table, the form and the database all worked together as a group and integrate perfectly.

Shane

Surname \*

Murphy

Enter your Surname

Email \*

Shaz@gmail.com

Tour Name \*

Test

Tour Price - € \*

5

Meeting Point \*

Test

Select the spot in which you want to begin your Tour.

Map showing Limerick, Ireland, with a red pin and a label 'H'. The map includes labels for Quin, Latoon, Shannon, Limerick, Dooradoyle, and Dublin city. Road markers for N18, N24, M20, and M7 are visible.

### 3.6.3. Customer Testing

#### Question 1

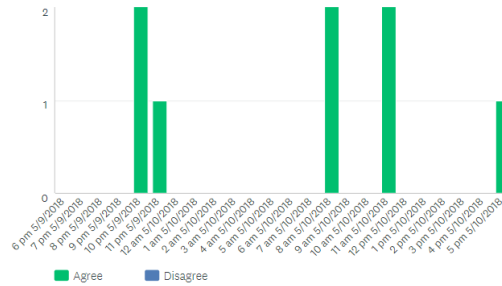
All users agreed to take part in the survey after testing Tours2Connect. They read the waiver provided below and proceeded.

Q1 (by hour)

Chart Type Display Options Trend by... Zoom

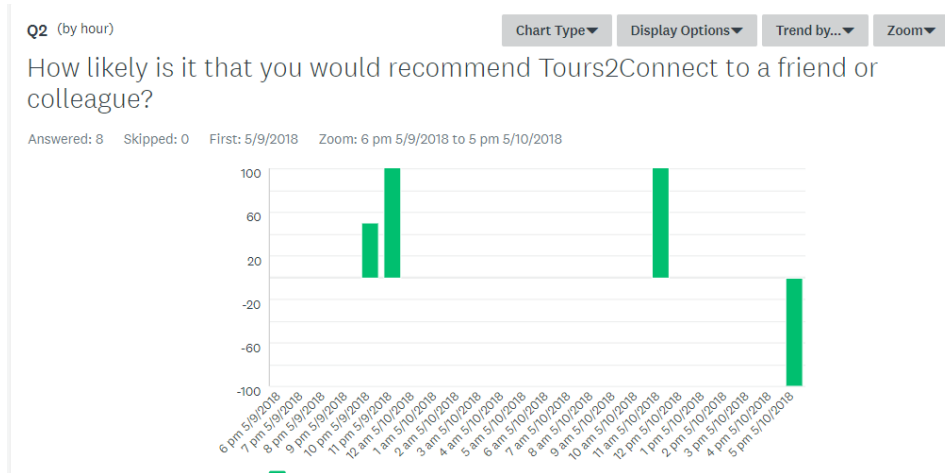
Your participation in this research study is voluntary. Of course, You may choose not to participate, if so turn back now. If you decide to participate in this research survey, you may withdraw at any time for any reasons you may have. The procedure involves answering the questions that follow, within the survey. Your answers and inputs will be confidential, and we do not collect identifying information such as your name etc. The survey questions will be about the my web application Tours2Connect. To help protect your confidentiality, the surveys will not contain information that will personally identify you. The results of this study will be used for an academic report about my web application, Tours2Connect. Please select your choice below. Clicking on the "Agree" button below indicates that: • you have read the above information • you voluntarily agree to participate • you are at least 18 years of age If you do not wish to participate in the research study, please decline participation by clicking on the "Disagree" button.

Answered: 8 Skipped: 0 First: 5/9/2018 Zoom: 6 pm 5/9/2018 to 5 pm 5/10/2018



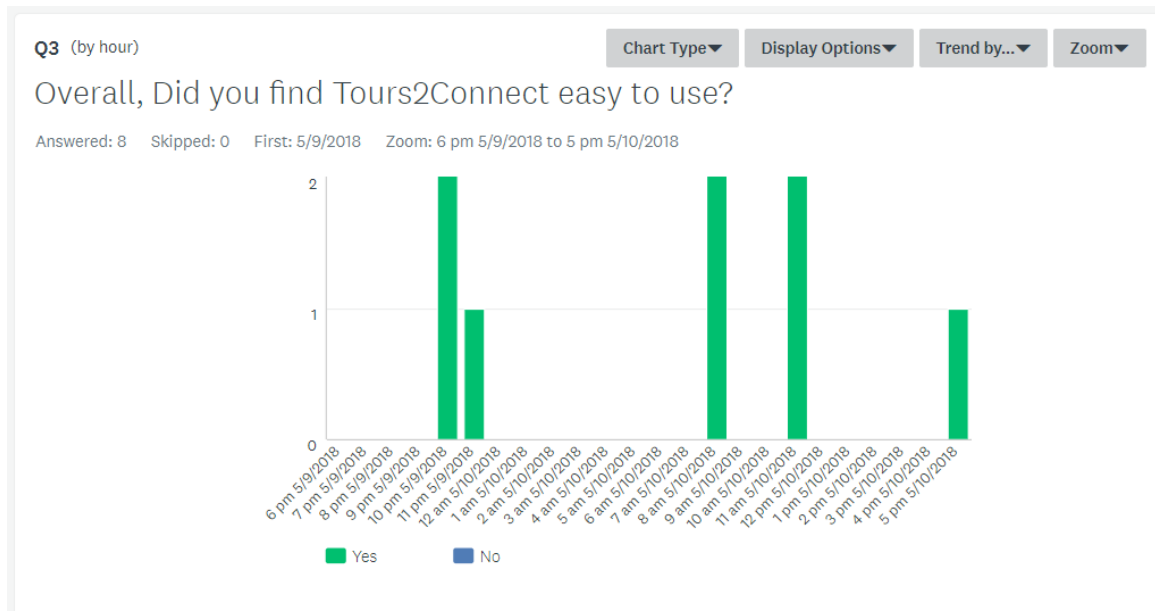
## Question 2

The likelihood of the users to recommend Tours2Connect to a friend is massive, as 5/8 ranked the likelihood to be between 9/10. It is clear the users liked the web application if they are rating it a 9/10 to recommend to a friend.



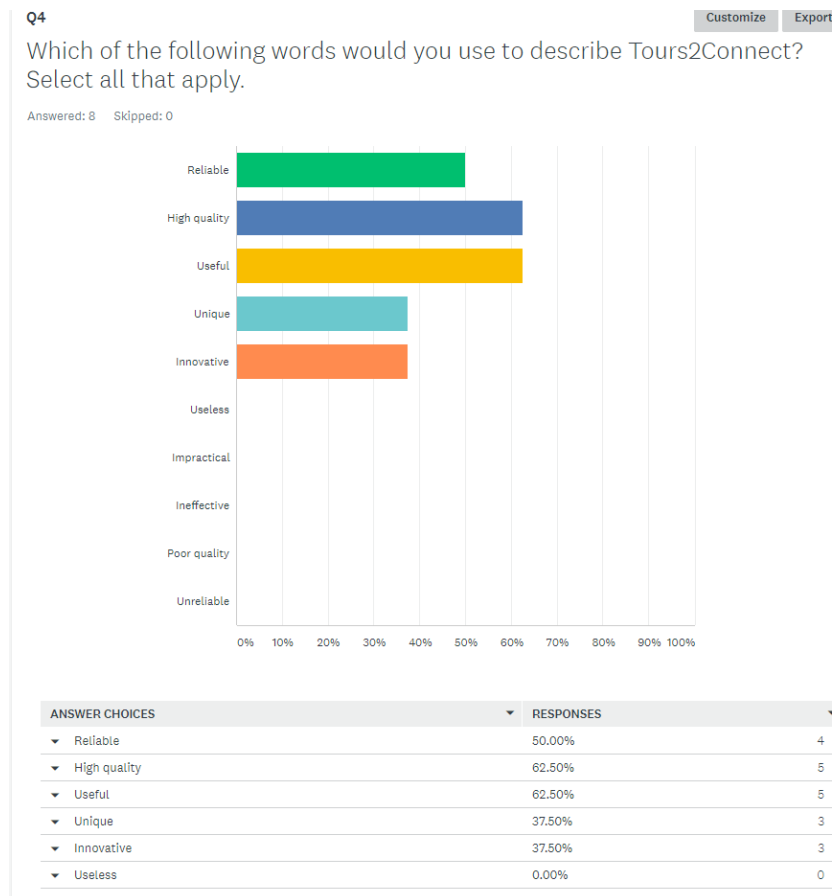
## Question 3

All users voted that Tours2Connect was an easy to use application. This is all participants.



## Question 4

4 users describe Tours2Connect as 'reliable', 5 users ranked us as 'High Quality', 5 users ranked us as 'Useful' as a web application, 3 users described the web application as 'unique' and 3 also said the idea is 'innovative'. No negative replies here.



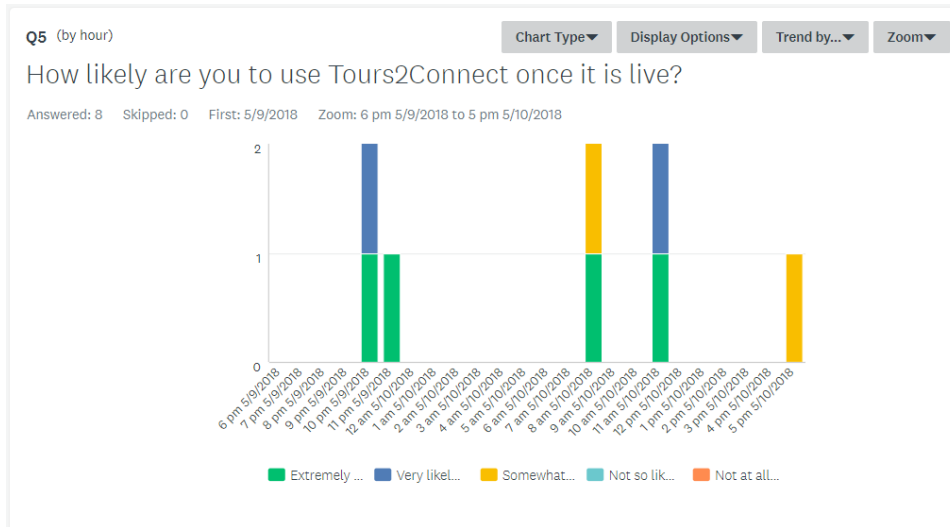


## Question 5

Extremely Likely – 4

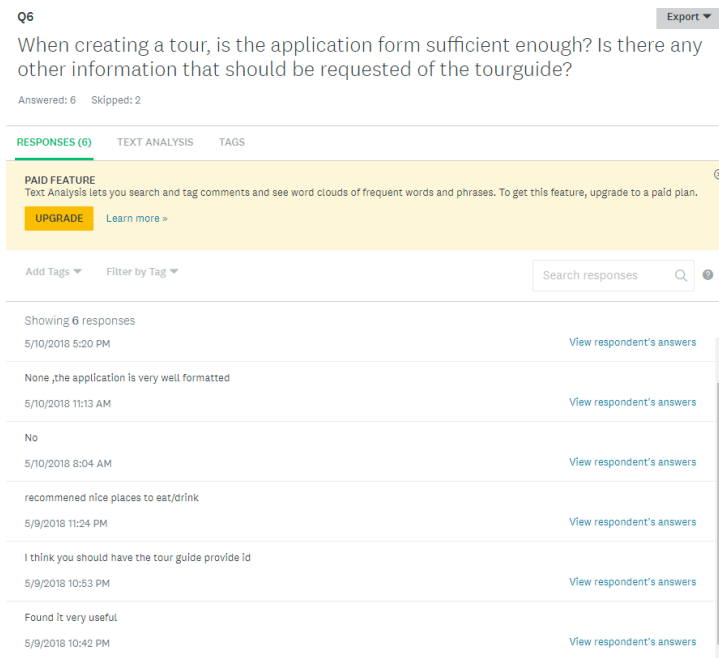
Very Likely – 2

Somewhat Likely – 2



## Question 6

Comments from the users.



## Question 7

Changes from the users if they had the option.

**Q7** Export

What changes would you make to the web application?

Answered: 6 Skipped: 2

**RESPONSES (6)** TEXT ANALYSIS TAGS

**PAID FEATURE**  
Text Analysis lets you search and tag comments and see word clouds of frequent words and phrases. To get this feature, upgrade to a paid plan.  
[UPGRADE](#) [Learn more »](#)

Add Tags Filter by Tag Search responses

Showing 6 responses

A filtering section to filter search tours.

5/10/2018 5:20 PM	<a href="#">View respondent's answers</a>
None ,the web application has a great layout	<a href="#">View respondent's answers</a>
5/10/2018 11:13 AM	<a href="#">View respondent's answers</a>
None	<a href="#">View respondent's answers</a>
5/10/2018 8:04 AM	<a href="#">View respondent's answers</a>
none	<a href="#">View respondent's answers</a>
5/9/2018 11:24 PM	<a href="#">View respondent's answers</a>
make the map smaller	<a href="#">View respondent's answers</a>
5/9/2018 10:53 PM	<a href="#">View respondent's answers</a>
None	

## Question 8

The feedback users left for me about Tours2Connect.

**Q8** Export

Do you have any other comments, questions, or additional feedback for us?

Answered: 6 Skipped: 2

**RESPONSES (6)** TEXT ANALYSIS TAGS

**PAID FEATURE**  
Text Analysis lets you search and tag comments and see word clouds of frequent words and phrases. To get this feature, upgrade to a paid plan.  
[UPGRADE](#) [Learn more »](#)

Add Tags Filter by Tag Search responses

Showing 6 responses

I would like to see more tours that are available in different counties or countries.	<a href="#">View respondent's answers</a>
5/10/2018 5:20 PM	
No	<a href="#">View respondent's answers</a>
5/10/2018 11:13 AM	
No	<a href="#">View respondent's answers</a>
5/10/2018 8:04 AM	
good job	<a href="#">View respondent's answers</a>
5/9/2018 11:24 PM	
other than the above comments I feel the site is very well done	<a href="#">View respondent's answers</a>
5/9/2018 10:53 PM	
No not now.	

### **3.7 Evaluation**

After testing Tours2Connect through multiple different methods was completed, evaluating all test results was the next step.

Security testing involved using OWASP Zed Attack Proxy, testing for XSS injection and SQL injection from within Forms and user input sections.

OWASP Zed Attack resulted in issues with the robots.txt file in which cloud9 has automatically installed, as well as the HTTP headers that cloud9 has too. There were no issues with Tours2Connect from these tests. The XSS and SQL injection tests also had good results for Tours2Connect as users were not able to input special characters in some input fields, and the characters " and ' , are not allowed in any fields.

I also made sure that my sessions were working. When a user logs out and copies the URL from when they were logged in, the user was sent to the registration page, which is exactly what i wanted to happen. From here, the user is required to login again to gain access to where they were before logging out. This is a great example of access control. The fields in which I enabled validation on the inputs work perfectly too. The inputs a user enters in any field on Tours2Connect is checked and validated so that special characters such as ' and " are not allowed. This means that SQL and XSS cannot be performed as they require these special characters.

The environment testing also seems to have turned out good for Tours2Connect. Two browsers were used to see if there was any difference when transitioning between different environments. Microsoft edge and Google chrome were used for this. This resulted in my login, registration, google maps API and all my other features working perfectly and having no issues with the transition between different browsers.

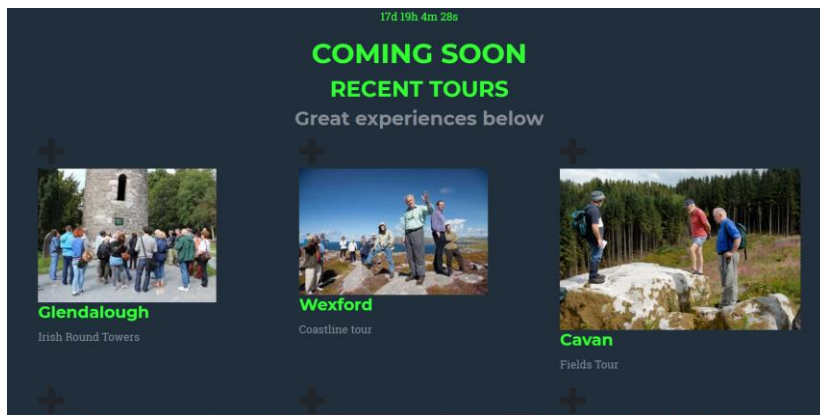
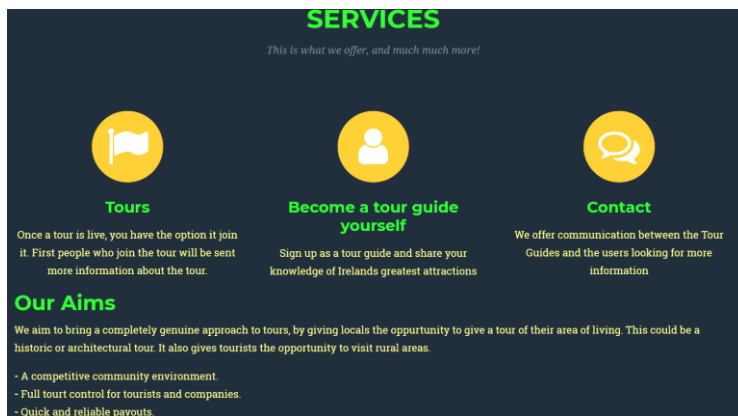
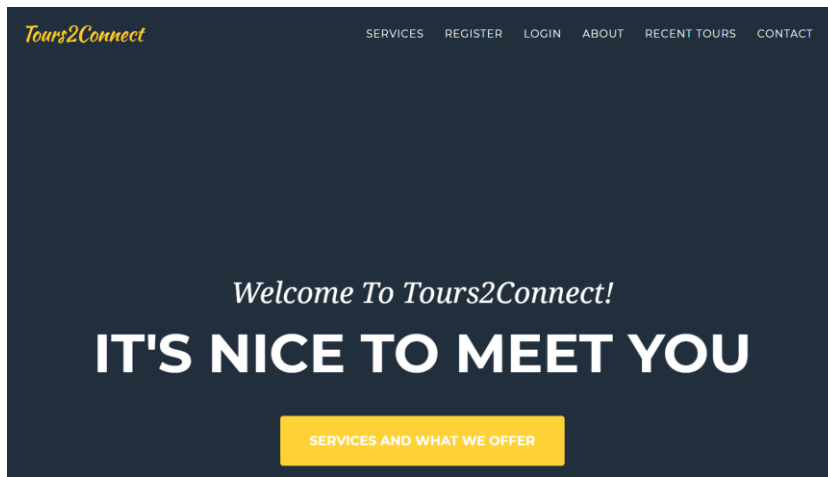
All the features on Tours2Connect were integrated at different stages of the development process, yet they all work together. The google maps API extracts data from the database, which the database then extracts data into the tours table,

and the tours table reads the data entered into the create tour form and displays it. All these features offer users a great, easy experience.

I asked friends, work colleagues and ex colleagues to take part in my customer testing. After this, I asked the customers to take part in a survey in which they would outline their satisfaction or dissatisfaction with their experience of Tours2Connect. The results are above under '3.6.3 Customer Testing', but as you will see, the results are very positive, and this is great news for tours2connect. All users were of different backgrounds, some replies were from IT minded people, some were from design orientated users, and others were from none technical users. This results in fully rounded feedback from customers of all different experiences and knowledge.

## 3.8 Graphical User Interface (GUI) Layout

Index.php





## Register / Login

### Tours2Connect

Login or sign up below!

Need an account? [Register here!](#)

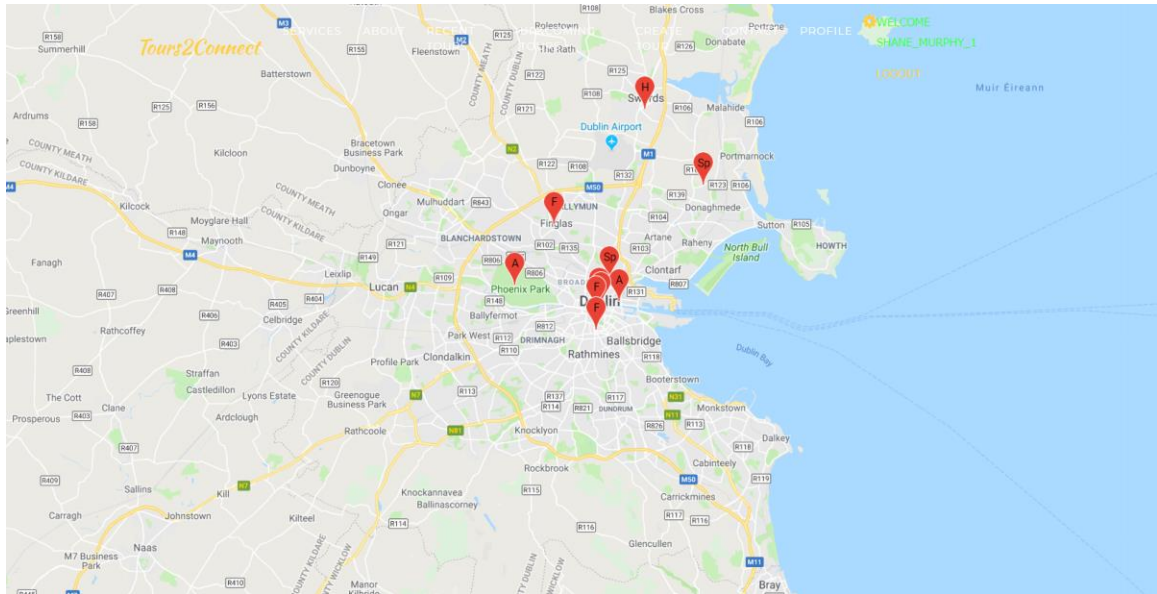
### Tours2Connect

Login or sign up below!

Must contain at least one number and one uppercase an

Already have an account? [Sign in here!](#)

# Tours.php




## TOURS4YOU

KEEP TRACK OF UP AND COMING EVENTS!

FIRST NAME	LAST NAME	CATEGORY	TOUR NAME	MEETING POINT	REGION	PRICE €	DATE	LEARN MORE
Shane	Murphy	Sport	Croke Park	1 Jones Rd, Drumcondra, Dublin 3	Leinster	11	2018-07-21 13:30:00	<a href="#">Learn More</a>
Sean	Cowley	Sport	Thomond Park	10 Cratloe Rd, Limerick	Munster	15	2018-10-19 18:00:00	<a href="#">Learn More</a>
Samantha	Harding	History	Killarney Castle	Road Road, Ross Island, Killarney, Co. Kerry	Munster	22	2018-06-01 18:00:00	<a href="#">Learn More</a>
Sean	Cowley	Adventure	Tayto Park	Adrian Geraghty Joinery, Kilbree, Ashbourne, Co. M	Leinster	15	2018-05-02 14:00:00	<a href="#">Learn More</a>
Sean	Murphy	Adventure	Guided Tour of the Zoo	The Phoenix Monument, Phoenix Park, Dublin 8	Leinster	0	2018-09-20 11:00:00	<a href="#">Learn More</a>
James	Redmond	Adventure	Pub Crawl of Temple bar – 1 Free Drink	O'Connell Bridge, Dublin 2	Leinster	0	2018-06-21 15:30:00	<a href="#">Learn More</a>
Matt	Regga	Fun	Best bar crawl of your life – Harcourt Street pubs	60 Harcourt St, Saint Kevin's, Dublin 2	Leinster	0	2018-07-30 20:30:00	<a href="#">Learn More</a>
Ciaran	Stainton	Fun	Crazy tour of Finglas	The Full Shilling Pub – 66 Main Street, Finglas, D	Leinster	0	2018-07-19 20:30:00	<a href="#">Learn More</a>
Derek	Hansane	Fun	Tour of Dublin city – Includes stops at pubs, hist	BadAss Bar, 9-11 Crown Alley, Temple Bar, Dublin 2	Leinster	0	2018-06-27 20:30:00	<a href="#">Learn More</a>

## TourInfo.php


*Tours2Connect* SERVICES ABOUT RECENT TOURS UP&COMING TOURS CREATE TOUR CONTACT  WELCOME SHANE\_MURPHY\_1 LOGOUT

*1 Welcome To The Tour: Croke Park*  
**SEE TOUR DETAILS BELOW!**

## TOUR INFORMATION

*This is what we offer, and much much more!*

First Name:	Shane
Surname	Murphy
Tour Name:	Croke Park
Category	Sport
Meeting Point	1 Jones Rd, Drumcondra, Dublin 3
Price	11
Attendance:	0

 Checkout



# CreateTour.php

*Tours2Connect* SERVICES ABOUT RECENT TOURS UP&COMING TOURS CREATE TOUR CONTACT SHANE\_MURPHY\_1 LOGOUT

*Welcome Tourguides*

## LET'S CREATE A TOUR!

First Name \*

Surname \*

Enter your Surname

Email \*

Tour Name \*

Tour Price - € \*

Meeting Point \*

Select the spot in which you want to begin your Tour.

Latitude of Tour \*

This will display your tour on the map on the tours area. Need help with finding your events latitude? Find it [here](#).

Longitude of Tour \*

This will display your tour on the map on the tours area. Need help with finding your events longitude? Find it [here](#).

Select Region of Tour \*

Choose from the options above

Select Category of Tour. \*

Please choose one of the options.

Date \*

## 4 Conclusions

There are many advantages to the creation of Tours2Connect. As the tourism industry is growing every year, this allows for such a resource to thrive in these times. In 2016, out-of-state tourist expenditure amounted to €5.1 billion, while Domestic tourism expenditure amounted to €1.8 billion, This is a massive market with a lot of money to be made from the tourism industry.

I believe that Tours2Connect is an idea that allows tourists to receive the best of service when they visit the country of Ireland, as well as offering Tour guides the opportunity to branch out and shared their knowledge of their favourite tours of Ireland, whether it be scenic, historic or sports related.

I know from meeting people on my work placement that there are people out there who love to give tours to locals in their towns or to family members, I believe that tours2connect would allow these people who love giving tours would be interested in the idea and register to offer people their knowledge.

From all the testing, all of the implementation and all of the security aspects of Tours2Connect, I truly believe that this idea is something great and could help the Tourism industry in Ireland. "People who love the country of Ireland, being able to give a guided tour of amenities that they love, to people who are visiting the country". That is what the idea was at the start of the prototype phase, and what I have develop still allows for that quote to become true.

## **5 Further Developments & Research**

The further development possibilities of Tours2Connect and the innovative idea that it is, are endless. Features could be added into Tours2Connect in a month's times, 2 months' time etc and the web application would still be gaining users and adding creative new ideas to interact with the users. The possibilities of expansion are endless with Tours2Connect and with the right amount of resources, this idea could change the way tours are being given in the country. There is a definite need for tour guides to be interested and innovative with their ideas of tours within Ireland, therefore a web application that offers that service is a great starting point for them.

As stated above, the idea is completely based in Ireland. Tours2Connect is only offering a service within the country at the moment. For further development, I would love to expand the idea into different countries and set up a base in another European country so that I could have access and knowledge of the tourism within that country.

A tour guide rating system is also very possible to add in, each user can give a tour guide a rating after being on a tour with that user.

## **6 Appendix**

### **6.6 Project Proposal**

#### **Objectives**

My Project will consist of developing a web application for the tourism industry. The Web application will consist of multiple page interfaces linked together, in order to allow potential customers to browse through different tours and select the one that appeals to them the most.

Each user will register and sign in, from here the logged in user will have the choice to add a tour or the opportunity to browse the tours. When a user see's a tour that they like, they can go onto the page and click the attend button.

The idea of the project is to allow users to sign up and register as a tourist or a tour guide. From here the tour guides can post potential tours that they have planned, These tours will be displayed on the web application through a searching mechanism whos data is held on a database through PHPMyAdmin. When the user does search and sees a tour that they like, the user will click this and bring them to that tours page. From here the user can click the attend button, after which the tours attendee count will go up one, making less available spots available. Also, there will be a price summary of each tour displayed.

The user should then have the opportunity to pay through PayPal or have the option to pay the guide in person.

The security aspect of the project is obviously high as it is under the specialization of Cyber Security, but it also allows for entry of peoples sensitive information like passwords and usernames. These will be encrypted by the PHP password default encryption option PHPMyAdmin gives us.

As the attendee number increases when a user decides to attend the tour, this will therefore be real time updating as other users will be able to see that there is one less space available in the tour.

None of the tour services, or searching will not be available until the user logs in or registers an account.

## **Background**

The idea of the a web application where tour guides can display their up and coming tours with the option for other users to attend came from my time on work placement. My work placement took place in Beaumont Hospital where i had met alot of nice people, staff and patients, but my idea came from a patient who i had helped get into his bed. He had told me that he loved giving guided tours to locals in his town in County Kerry. From this he had told me he wished he could of had more people on his tours to share his love of mountain walking etc. After talking to this man i got an idea to create a Web Application in which men like the man in beaumont hospital, could post their tours online and where they could get a much bigger audience instead of a few locals from his town.

When i had gathered the idea, I then ran it by my manager in Beaumont Hospital and a few colleagues, they thought the idea was very unique and they had not heard of a web application that supports this kind of service. When i heard them say that it was unique and had not seen another application like this, I thought to myself that the idea of the Final year project is to stand out and if the idea is unique, it will stand out, that is why i have chosen to go ahead with it.

Since leaving Beaumont Hospital and starting back in NCI, i have been thinking of different ways to approach going about creating the Web Application and how i should start it.

## **Technical Approach**

I have done a lot of research into the Tour guiding industry and have not found anything like what I have chosen to create. A lot of the commercial tour guiding companies such as PaddyWagon and irishdaytours offer tours from tour guides who are on their payroll. They get paid per hour whether they give 2 tours a day or they give 4 tours a day.

These types of companies offer cheesy commercialized tours with the tour guide displaying no enthusiasm to the landscape as they have done the tour hundreds of times before. The difference between the tour guides on the webpage is that they would be doing the tours for the love of the landscape and the idea of sharing their knowledge with the attendees of the tour.

Tour guide companies also charge astronomical rates such as 40 euro per person for a tour of the Boyne valleys, this is so that the company can make money and also pay the tour guide. These prices can almost be halved on my web application due to the fact the only person making money will be the tour guide.

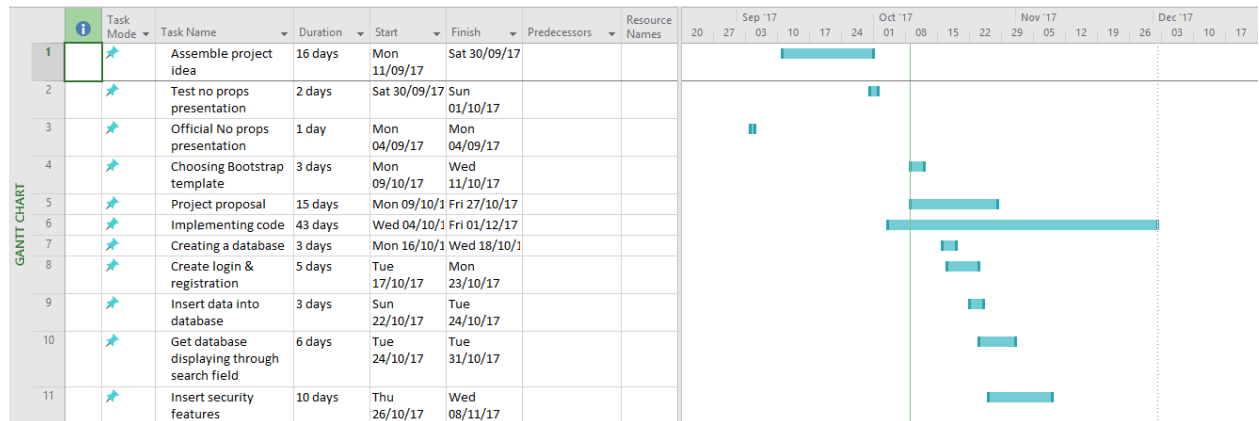
My idea has been accepted so the plan has already started, creating an idea and implementing how I'm going to go about getting to my end goal all had to be looked at and summarized for the no props presentation.

The users can expect a user-friendly web application that is very simple to move around. It'll allow users to register and log in. The users can then be able to see the easy using search mechanism within the web application to see possible tours in each province. The possible users can expect to see different pages for different tours and on these pages, it'll allow users to attend. From there a possibility to pay now or later will arise. The web application will be very easy to use and will cause no problems for users.

Our web application will have the ability to store and save information entered into it, such as peoples usernames and passwords, the data will be collected through the text boxes that allow username and password entry, as well as the tours being displayed on the pages. This will be stored on PhpMyAdmin, a database system.

The relationship with our users is strictly a platform based one. We offer a platform for tour guides to place their tours on the web for others to endure and enjoy. We do not employ the tour guides, nor do we pay them. They offer their skills to other users looking for a tour that will excite them and offer them knowledge.

## Project Plan



## Technical Details

The web application will be build using PHP, HTML5, JavaScript and Bootstrap to create the relevant content needed to be displayed. The bootstrap template will

give a base to the web application, and I will edit this content using php and HTML5 to suit how I want it to look.

I will create a database and store all tour information and user information on it. This will be stored on a database using PhpMyAdmin. The information in the database will be pulled through code that will be implemented where the tours table will be displayed.

I will develop my Web Application through Eclipse and cloud9, and host the it through Heroku or Microsoft Azure, this is still undecided.

## **Evaluation**

System testing is important throughout the process, this determines whether all functioning parts of your project are working correctly or not. This will be done at each major milestone of the project / each additional function added to web app, testing will be done to determine and evaluate how it effects the web app and how it will be a positive for users of the web app.



## **6.7 Monthly Journals**

### **September**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

#### **6.7.2.1 My Achievements**

This month, I was able to come up with my idea.

When I had the idea created in my head I was comfortable that it was the idea that I wanted to do. I had my no props presentation on the 2<sup>nd</sup> of October, so I practiced my presentation and what I was going to say.

My idea is to create a web application for Tour Guides, where they can display tours for people to attend. Through this idea, I achieved what I wanted to do, I also decided on what technology I wanted to use.

To me, this was a successful month towards my achievements as I didn't know what I wanted to create, and I had a blank mind for ideas. So, when I did get an idea I was happy with it and I also felt comfortable in it.

My contributions to the projects included coming up with the idea and deciding on how I will go about creating my idea.

Came up with a presentation for my idea that I knew would make my idea come across well.

#### **6.7.2.2 My Reflection**

Took me a while to come up with my idea. The idea was inspired by the tourism industry. As it took a long time to come up with the idea, I felt it took a longer time to plan other aspects of my project.

## **October**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.3 My Achievements**

I completed my Project Proposal this month. I uploaded it over the reading week. Within this proposal I discussed the background of the idea, where the idea originated from, the objectives, the technical approach to how I will go about completing the project and what I will use. The project plan and technical details were also included in the proposal.

I began to look at my requirements specification after uploading the proposal

I looked into doing my requirements specification, while doing this I concentrated on getting some ideas on what to write for each heading.

I met with my Supervisor and this helped towards my approach to creating all the necessary diagrams as well as how I would layout my web application.

### **6.7.2.4 Intended Changes**

I intended to change my original layout of the web application after meeting with my supervisor. She had great ideas and put on the right path of changing certain aspects of the project.

### **6.7.2.5 Supervisor Meetings:**

Met with Catherine Mulwa on the 15<sup>th</sup> of October and discussed my project idea. She put me on the right direction to completing my requirements document.

## **November**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.6 My Achievements**

I completed my Requirements specification and Technical Report documents this month.

### **6.7.2.7 My Reflection**

I completed both my Requirements specification and technical report this month, it took a lot of time and effort to get this done. I met with my Advisor twice this month to get advice about system architectures etc. A lot of advice was given by Catherine this month on how I should lay out my reports

### **6.7.2.8 Intended Changes**

I intend to continue to build off of my prototype to give me a good base to start coding again in the new year.

### **6.7.2.9 Supervisor Meetings:**

Met with Catherine Mulwa on the 15<sup>th</sup> of November and discussed my project idea. She put me on the right direction to completing my requirements document.

Met with Catherine Mulwa again on the 29<sup>th</sup> of November to talk about the requirements and technical report. Again, great advice and pointed me in the right direction.

## **December**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.10 My Achievements**

Prepared for mid-point presentation. Completed slideshow to present. Tidied up the web application. Presented for mid-point and received good grade.

### **6.7.2.11 My Reflection**

I prepared very accurately for the mid-point presentation. Completed good slides with some images from my report from the previous month. I created a rough idea of how I wanted my web application to look like and I presented this to Vikas and Catherine.

### **6.7.2.12 Intended Changes**

I intend to continue to build off of my prototype to give me a good base to start coding again in the new year.

### **6.7.2.13 Supervisor Meetings:**

Met with Catherine Mulwa in December and talked about my report and which images to include in the presentation. Discussed what I should know and discussed my idea. She put me on the right direction to completing preparing for the midpoint presentation.

## **January**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.14 My Achievements**

Continued with my project slowly. Exams were in this month leaving very little time to concentrate on the project. Still was able to code somewhat.

### **6.7.2.15 My Reflection**

A lot of time was taken from this month due to exams being on. This made it hard to concentrate on the project. Continued with the project for a few days of month and got some coding done.

### **6.7.2.16 Intended Changes**

I intend to get stuck into the project in the coming months to be able to present a working project to be proud of.

## **February**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.17 My Achievements**

Got started on the coding of my project properly. I have decided all the pages that I need to complete my project. Index, login, register, tour pages. These are the basics at the moment of my project. The login and register are on the same page using ajax to change between them. The login and register are created. The register is going to the database, but when logging in with the credentials, it is not working correctly. I have also automatically generated tours in the database and they are being extracted and displayed on the 'tours' page.

### **6.7.2.18 My Reflection**

Need to fix the login issue. The login won't accept the details entered at registering. A lot of work was done this month. PHP causing a few issues with errors with connecting to SQL which caused delays.

### **6.7.2.19 Intended Changes**

Intend to fix login issue. Intent to implement a google maps API to connect to the database.

## **March**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.20 My Achievements**

Fixed login and register. Credentials work for logging in now. Database stores information and displays it onto tours page through a table. The table is used to display all relevant information of a tour.

Implemented a google map API. This is at the top of the tours page. It is implemented but not reading from the database as of yet.

### **6.7.2.21 My Reflection**

Google Maps API is hard to implement. This is what I have noticed from working with the API. The documentation online isn't very clear when you're not using their example that is used. This makes it tough to implement the API for my idea.

### **6.7.2.22 Intended Changes**

Intend to get the google maps API connected to my database through XML so that it reads the longitude and latitude of the addresses in the database to display markers on the map.

## **April**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.23 My Achievements**

This month was great for progression. The google maps API has been connected to the database and is displaying markers from the addresses in the database. This shows the location of all tours available.

I have also created a form to enter the database, this means that the data I automatically created months ago can be deleted, and I can enter my own information through the create tours page available for users.

### **6.7.2.24 My Reflection**

Google maps API is displaying from my database and I am now able to enter information through a form to display on the tours page. Massive progression and great work being done. Massive step in the right direction.

### **6.7.2.25 Intended Changes**

Have to tidy up my pages. I have to also make sure that when a user clicks on a specific tour, it is able to only display the information of that tour only.



## **May**

Student name: Shane Murphy

Programme (e.g., BSc in Computing): BSHCYB4 - BSc (Honours) in Computing

### **6.7.2.26 My Achievements**

Tidied up certain pages. Implemented more styling to the web application so that it wasn't basic anymore. Began working on a rating system and the idea of a user clicking on a specific tour, to only display the information for that tour only.

### **6.7.2.27 My Reflection**

A lot of work still left to do, but work is getting done and fast. Another great start to the month and I have my priorities set in place.

A lot of documentation is yet to be done as of yet, but it will get done.

### **6.7.2.28 Intended Changes**

To get all functionality working and the web application to be completely finished by the deadline.

## 7 References

Borysowich, C. (2017). *Defining Data Requirements*. [online] *It.toolbox.com*. Available at: <http://it.toolbox.com/blogs/enterprise-solutions/defining-data-requirements-23690> [Accessed 30 Nov. 2017].

Failteireland.ie. (2017). *Cite a Website - Cite This for Me*. [online] Available at: [http://www.failteireland.ie/FailteIreland/media/WebsiteStructure/Documents/3\\_Research\\_Insights/3\\_General\\_SurveysReports/Tourism-Facts-2016.pdf?ext=.pdf](http://www.failteireland.ie/FailteIreland/media/WebsiteStructure/Documents/3_Research_Insights/3_General_SurveysReports/Tourism-Facts-2016.pdf?ext=.pdf) [Accessed 30 Nov. 2017].

Google Developers. (2017). *Google Maps APIs | Google Developers*. [online] Available at: <https://developers.google.com/maps/> [Accessed 30 Nov. 2017].

Henderson, A. (2018). *The CIA Triad: Confidentiality, Integrity, Availability - Panmore Institute*. [online] *Panmore Institute*. Available at: <http://panmore.com/the-cia-triad-confidentiality-integrity-availability> [Accessed 12 May 2018].

Vogels, R. (2018). *A 6-Step Guide to Web Application Testing - Usersnap*. [online] *Usersnap Blog*. Available at: <https://usersnap.com/blog/web-application-testing/> [Accessed 09 May 2018].

Owasp.org. (2018). *Category: OWASP Top Ten Project - OWASP*. [online] Available at: [https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project) [Accessed 09 May 2018].