***Complane Airlines***

***Enable unhappier flight passengers to gather on line***

*Benjamin ADELINE*

National College of Ireland Student Number: 16135083

Higher Diploma in Science in Computing Information

Dublin, Ireland

[Benjamin.adeline@gmail.com](mailto:Benjamin.adeline@gmail.com)

*Abstract*

*With the increase of people are experiencing flight issues. It involves huge delays, poor quality services or even cancellations.*

*In order to through different ways*

key words API, HTTPS, Superuser.

# Introduction

With the increase of the airlines traffic and especially low cost, more and more people are experiencing flight issues. It involves huge delays, poor quality services or even cancellations.  
 As customer services from airlines are not always very keen to answer to desperate customers, I had the idea to create a web page to allow them to meet online travelers with the same issues of confronting the same airline.

In order for them to meet online in a secure and simple way, I opted for a google sign in account. By this manner, the user will benefit from Google security and will be reassured.

As he will need to enter a few data to express his issue, the Firebase database also from Google will be performing the background job to store the form with the issue, but also to retrieve the issues from other travelers.

# Background

This project will be deployed on an FTP server with an https certificate.

With is the secure version of HTTP, the 'S' at the end of HTTPS stands for 'Secure'. It will ensure that all communications between the browser and the website are encrypted. The encryption level is … /HTTPS will be used to protect highly confidential online information the data fill out in the online forms.

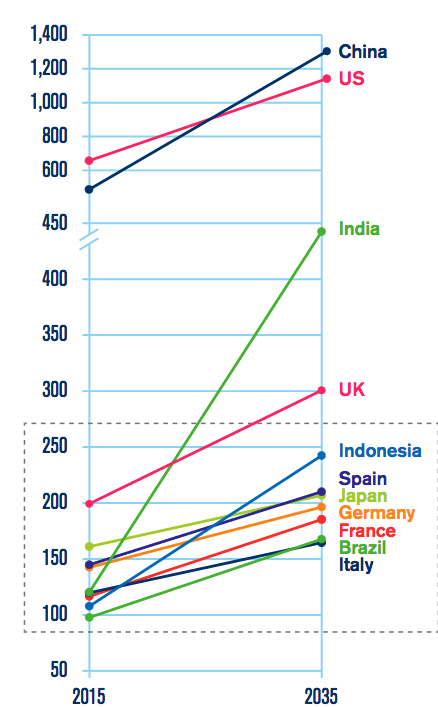
Web browsers such as Internet Explorer, Firefox and Chrome also display a padlock icon in the address bar to visually indicate that a HTTPS connection is in effect.

The authentication and the database management will be handle thanks to google Sign-In API (It will be detailed later in the report)

# Define Context

In a context of a growing demand of flights from passengers IATA forecasts passengers demand to double over the next 20 years. The International Air Transport Association (IATA) expects 7.2 billion passengers to travel in 2035, a near doubling of the 3.8 billion air travelers in 2016.

The prediction is based on a 3.7% annual Compound Average Growth Rate (CAGR) noted in the release of the latest update to the association’s.



Unfortunately the increase of complains will continue follow this trend especially for the low cost companies.



EUclaim said it had received 15,499 compensation enquiries in relation to easyJet, putting it at the top of the worst offenders table.

It said that it has seen a continued increase in the number of claims made against British Airways, meaning that 2016 has been the airline’s worst year for delays and cancellations.

EUclaim helps passengers to claim compensation owed to them because of flight delays and cancellations, but charges fees for its services.

If they don’t want to pay they can complain to the Civil Aviation Authority (CAA), but cannot see the feedback of other passengers. That is why Complane Airlines has plenty of room for opportunities in this niche market

.

# Aims

This project Will allow people to:

* Find travelers experiencing or who have experience the same issue during their flights.
* Express their flight issues and helping other passengers.

Gather with other travelers to request reimbursement from the airlines.

# Technologies

The technologies used are:

* All operating Systems (Windows, Linux and Mac OS)
* FTP server (File Transfer Protocol)
* HTTPS certificate (Hyper Text Transfer Protocol Secure)
* SQL Database (Structured Query Language)
* Google Sign-in API (Application Programming Interface)
* PHP (Hypertext Preprocessor, a server scripting language)
* JavaScript
* HTML

# Functional requirements

This system will:

* Recognize users by their google ID, by a person.
* Register the issue of the traveler
* Allow the user to see the issues of other travelers according to his criteria.

# Data requirements

To create the database, the users will a few information (first name, airlines company, issue and the year the flight took place).

The database will be hosted on the FTP server in SQL language. In order to Protect from SQL injections. This happens when hackers are trying to send request in a form to reach confidential information in the SQL database.

Actually the attacker creates or alters existing SQL commands to expose hidden data, or to override valuable ones, or even to execute dangerous system level commands on the database host. This is accomplished by the application taking user input and combining it with static parameters to build an SQL query.

This is crucial as if the hacker notices the lack of input validation and connecting to the database on behalf of a Superuser or the one who can create users, He mays create a Superuser in your database.

Once a Superuser it will be a piece of cake for him to damage it.

In order to avoid this, I decided to implement a layer of PHP to impede those attacks to occur. For that I use a string (a ligne of text code):

“ string mysql\_real\_escape\_string ( string $unescaped\_string [, resource $link\_identifier = NULL ] ) “

This “mysql\_real\_escape\_string “ Impede special characters in a string to be used in an SQL statement. For that it take into account the current character set of the connection so that it is safe to place it in a [mysql\_query()](http://php.net/manual/en/function.mysql-query.php). If binary data is to be inserted, this function must be used.

mysql\_real\_escape\_string() calls MySQL's library function mysql\_real\_escape\_string, which prepends backslashes to the following characters: *\x00*, *\n*, *\r*, *\*, *'*, *"* and *\x1a*.

This function must always (with few exceptions) be used to make data safe before sending a query to MySQL.

We these lists are maintained by The OWASP SecLists Project

<https://www.owasp.org/index.php/OWASP_SecLists_Project>

# System Requirements

This system can be run on any computer or device with a browser and an internet access. The system was built and run on a two-year-old mac laptop, and the system did not use up significant resources on the laptop.

It also requests a google account as I added a goggle sign in button to authenticate on the first page. Google Sign-In is a secure authentication system that reduces the burden of login for the users, by enabling them to sign in with their Google account.

It is user friendly, and like most of the people already a google account that they use for their emails or android phones with Google play it will reassure the customers.

Google Sign-In is also your gateway to connecting with Google’s users and services in a secure manner. In the future, if I want to add paying services, I will be able to give my clients the opportunity to pay with Android Pay, share with their Google-wide contacts and more.

A safe Android Pay system combined with HTTPS certificate for my website

By Integrating Google’s user-centric APIs and services inside your app to help your users to act and convert.

The good point in the google Sign in is that as I created division on the same page with block in in order to appear in the right order; the authentication will have no need to be targeted to another page (please see below the screenshot with the divisions and block in function in order to make them appear at the right time):

[] screenshot

# User requirements

The user requirements for this webpage are:

* Publish his flight issue in a safe way
* To be able to see if anyone on the webpage experience the same issue than him.

# Purpose of the WebAPP

Project design (done in the interim presentation)

1. **Process Flow**

### Login with a google account

### Write your issue

### See the results

### Sign Out

{process flow}

  - Data model design

{process flow}

Architecture requirements (done in the interim presentation)

- Class diagram

- Activity diagram

Project requirements (done in the interim presentation)

 - Use case diagram

# Usability requirements

A user should understand the system at first glance.

The system only requires the user, a google account and to have an internet access.

# Design and Architecture

To calculate the ability of the password to resist a brute force attack the following algorithm was used. Please note that this algorithm comes from Geek's Password Strength Meter <https://www.sitepoint.com/geeks-password-strength-meter/>

The search algorithm is a simple SQL statement to read the database, and to increase the speed of the search a binary search is conducted on an index field.

# Implementation

The webpage is deployed on a FTTP sever and protected by a HTTPS server main peace of code which calculates the strength of the password can be downloaded from git hub, [https://GitHub/osumartin/NCI-Password-Search.git](https://github.com/osumartin/NCI-Password-Search.git) this code, is available in its original form, along with the modifications, which were made, to match the user interface of this system. The original code is in the folder Complane Airlines.

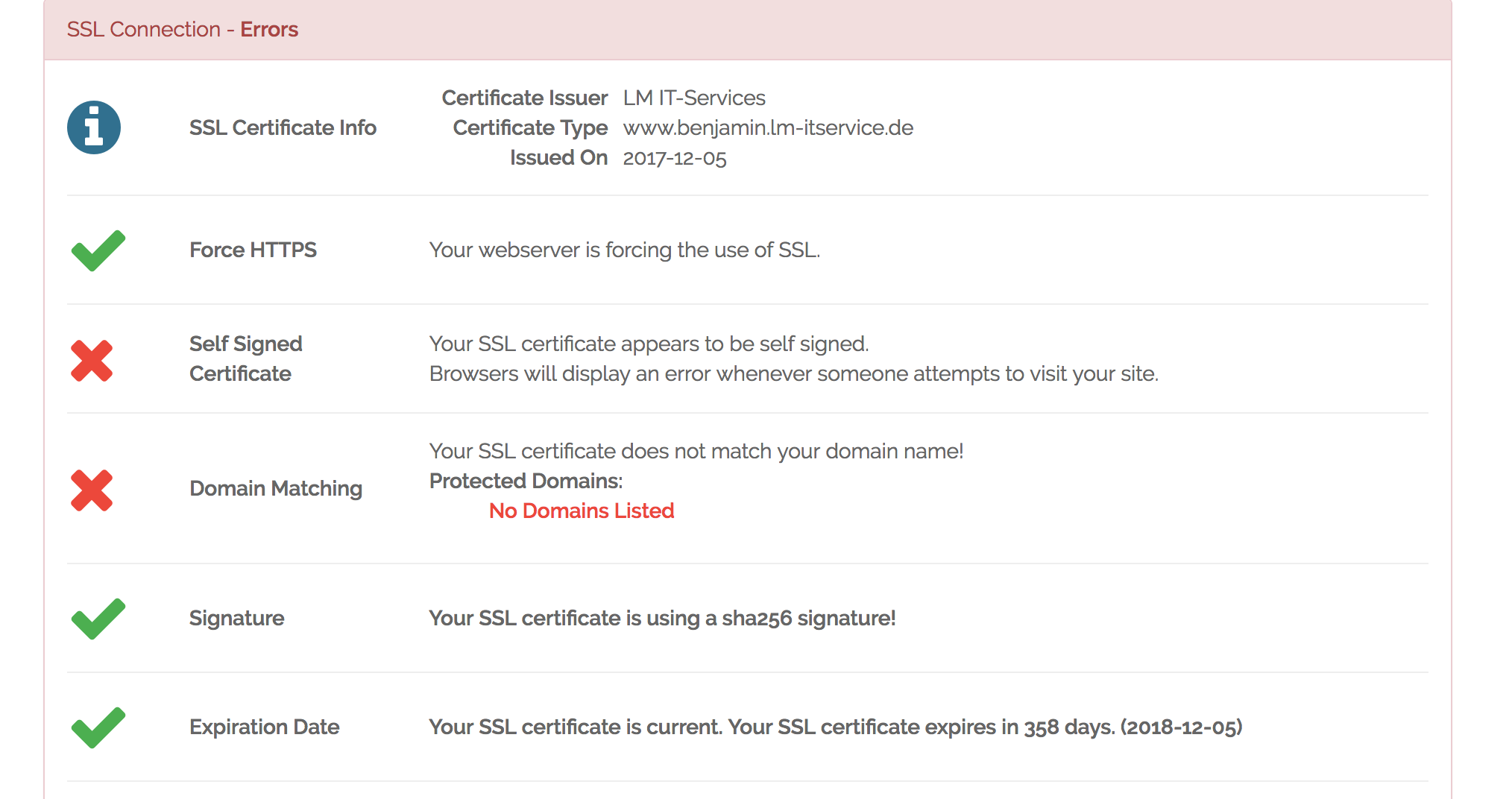
The source code of the entire project, can be downloaded from GitHub, and is licensed under the folder

# Testing

The below table shows the tests conducted, and their results.

| Tests Conducted | Results |
| --- | --- |
| HTTPS testing | The system protected well and avoid any |
| SQL injections | The system impedes any SQL injections because it forbidden the use of the characters necessary to SQL injections. |
| Google Account Security | It was not possible to gain any access to the site via this method. |

# results of the https page testing



Please note that the error “ Your SSL certificate appears to be self signed . Browsers will display an error whenever someone attempts to visit your site” is normal as it

# Security Feature Assessment

- Scope of assessment vs the OWASP Top 10 (chosen OWASP that we will use)

- Tools used (HTTPS page testes with Why not

the overall of the results are available on:

‑‑‑‑[https://www.whynopadlock.com/results/b6d785fd‑c37d‑4db1‑b2e3‑9e515de7e4ce](https://www.whynopadlock.com/results/b6d785fdc37d4db1b2e39e515de7e4ce)

- Define the vulnerable codes (write the sample of codes in the report)

# User Interface

In choosing the interface, the following factors were considered.

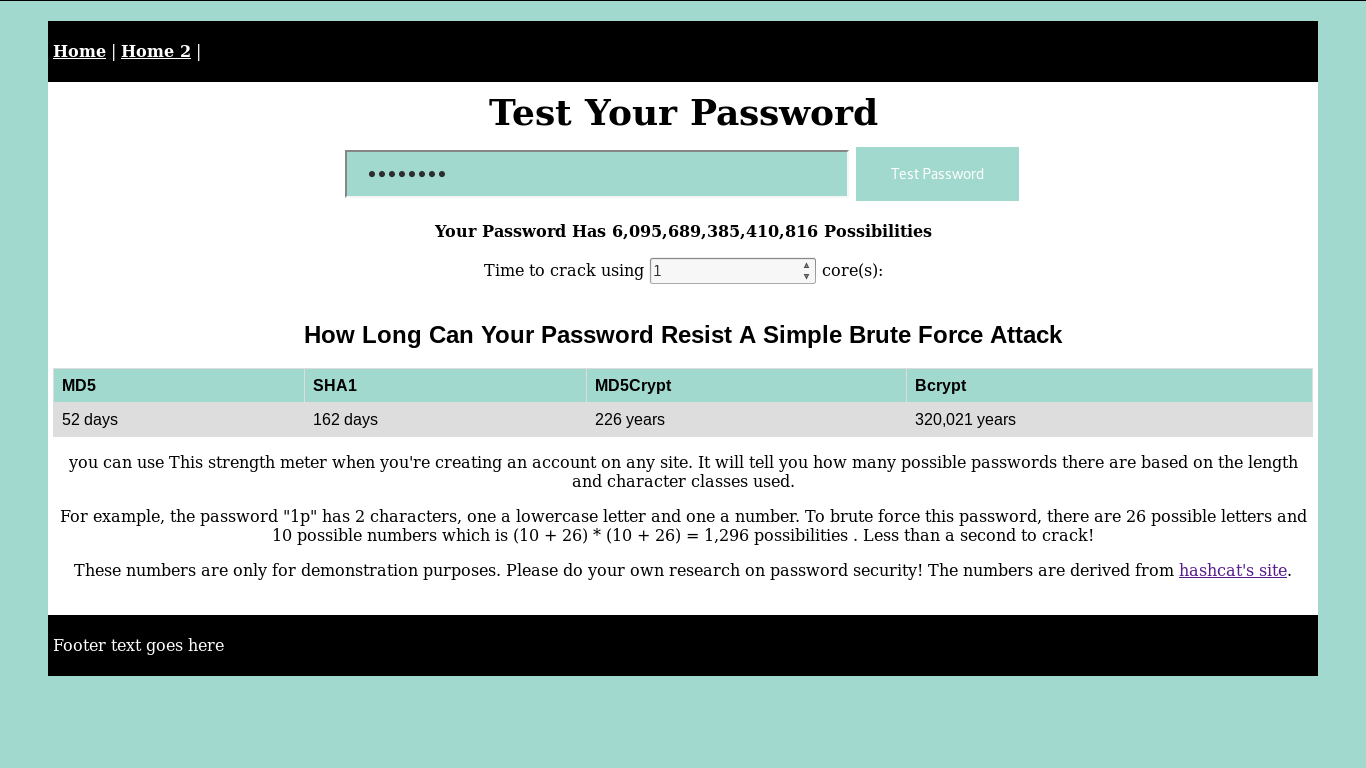
* Ease of use
* Cross Platform usage
* Everything under one Webpage to simplify the loading

Given these criteria the best choice was to use a web interface.

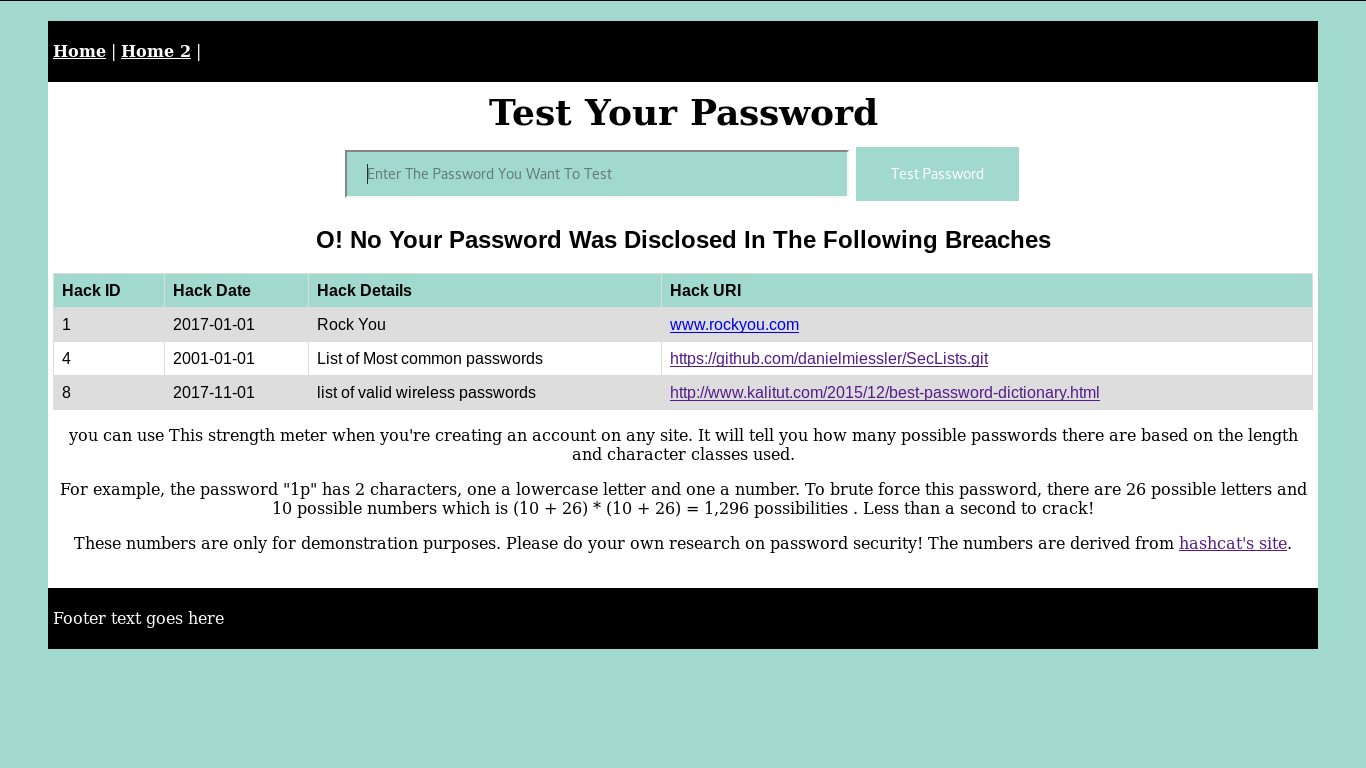
Welcome Screen This is the first screen which greets the user. There are only two elements which the user needs to interact with. The text entry field, and the submit button.



Once the user starts to type, the first test commences. The calculations on how easy it would be to brake their password using just brute force.



Results screen This is where the user finds out whether their password has ever been disclosed.



# Customer testing

This project was tested by a group who campaign against hate speech at a workshop which was conducted with them on security. Out of 10 people who tried the system, 7 had passwords which were both easy to brute force, and on the list of passwords which have already been disclosed.

This system was helpful to them in deciding to change their passwords. All 7 people who saw that their password was week according to the parameters of the tests conducted changed their passwords.

# Evaluation

The following evaluations were conducted on the system:

* Passwords Authentication via Google was really satisfying and worked well
* The HTTPs prevented and allowed the SQL
* Local and remote testing was done to verify that the system could return the results within 2 seconds.
* SQL injection was attempted to try and gain access to the system.
* The system was tested, by a group, who are lightly to be targeted by politically motivated attackers.

# Conclusions

# Further development or research

This system could be used to conduct research on the types of passwords people choose, and determine sociological factors in choosing passwords. Factors which could be considered are:

* Do people who have a background in information security choose better passwords?
* Do people who have suffered a data breach, or had their password compromised, choose more complex passwords?
* Could companies or organizations with a sufficiently large user base, profile people based on the type of password which they use?
* Could class, ethnicity, gender, sexual orientation, or other data be gleamed by looking at your password choice?

# References

Ashley Madison data breach, 2017. . Wikipedia.

Change a Password for MySQL on Linux via Command Line [WWW Document], 2014. . Liquid Web Knowledge Base. URL <https://www.liquidweb.com/kb/change-a-password-for-mysql-on-linux-via-command-line/> (accessed 11.29.17).

Create HTML table using javascript - Stack Overflow [WWW Document], n.d. URL <https://stackoverflow.com/questions/16126357/create-html-table-using-javascript> (accessed 11.29.17).

Firebase Database from Google.

<https://firebase.google.com/docs/database/security/securing-data>

Edit fiddle - JSFiddle [WWW Document], n.d. URL <https://jsfiddle.net/fh9FP/12/> (accessed 11.29.17).

Geek’s Password Strength Meter — SitePoint [WWW Document], n.d. URL <https://www.sitepoint.com/geeks-password-strength-meter/> (accessed 11.29.17).

Guy, 2014. Make Offline Mirror of a Site using `wget`. Guy Rutenberg.

IATA (International Air Transport Association) Forecast, IATA Forecasts Passenger Demand to Double Over 20 Years

<http://www.iata.org/pressroom/pr/Pages/2016-10-18-02.aspx>

HTML table advanced features and accessibility [WWW Document], n.d. . Mozilla Developer Network. URL <https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Advanced> (accessed 11.29.17).

HTTPS What is the difference between HTTP and HTTPS?

<https://www.instantssl.com/ssl-certificate-products/https.html>

HTTPS Testing results available on:

<https://www.whynopadlock.com/results/b6d785fd-c37d-4db1-b2e3-9e515de7e4ce>

Hunt, T., n.d. Have I been pwned? Check if your email has been compromised in a data breach [WWW Document]. URL <http://www.haveibeenpwned.com/> (accessed 11.29.17).

Is This Password Secure [WWW Document], n.d. URL <http://www.passwordtester.net/NCI-Password-Search/> (accessed 11.29.17).

jitpukdebodin, เขียนโดย Sumate, n.d. Howto: Enable Apache SSL in Kali.

Linux Command: List All Users In The System, n.d. . nixCraft.

Miessler, D., 2017. SecLists is the security tester&#39;s companion. It is a collection of multiple types of lists used during security assessments. List types include usernames, passwords, URLs, sensitive data grep s..

mysql - How do I replace all my NULL values in a particular field in a particular table? - Stack Overflow [WWW Document], n.d. URL <https://stackoverflow.com/questions/4629202/how-do-i-replace-all-my-null-values-in-a-particular-field-in-a-particular-table> (accessed 11.29.17).

OWASP SecLists Project - OWASP [WWW Document], n.d. URL <https://www.owasp.org/index.php/OWASP_SecLists_Project> (accessed 11.29.17).

passwords - Wordlists on Kali Linux? - Information Security Stack Exchange [WWW Document], n.d. URL <https://security.stackexchange.com/questions/45850/wordlists-on-kali-linux> (accessed 11.29.17).

php - Count number of iterations in a foreach loop - Stack Overflow [WWW Document], n.d. URL <https://stackoverflow.com/questions/6220546/count-number-of-iterations-in-a-foreach-loop> (accessed 11.29.17).

php - Hyperlink in Table - Stack Overflow [WWW Document], n.d. URL <https://stackoverflow.com/questions/3938903/hyperlink-in-table> (accessed 11.29.17).

PHP: Connections and Connection management - Manual [WWW Document], n.d. URL <http://php.net/manual/en/pdo.connections.php> (accessed 11.29.17).

PHP MySQLi Prepared Statements to Prevent SQL Injection [WWW Document], n.d. URL <https://websitebeaver.com/prepared-statements-in-php-mysqli-to-prevent-sql-injection> (accessed 11.29.17).

PHP: set\_exception\_handler - Manual [WWW Document], n.d. URL <http://php.net/manual/en/function.set-exception-handler.php> (accessed 11.29.17).

Raise of complains with low-cost Airlines

<https://www.thesun.co.uk/living/2594786/easyjet-tops-list-of-compensation-complaints-as-best-and-worst-airlines-revealed/>

Remove Permissions for a MySQL User on Linux via Command Line [WWW Document], 2014. . Liquid Web Knowledge Base. URL <https://www.liquidweb.com/kb/remove-permissions-for-a-mysql-user-on-linux-via-command-line/> (accessed 11.29.17).

Restart Start Stop MySQL Server from Command Line, macOS, Linux, 2015. . Coolest Guides on the Planet.

Salame, W., 2017. Password dictionary. KaliTut Tutorial.

SQL injections: SQL Injection,

<http://php.net/manual/en/security.database.sql-injection.php>

Sqlmap tutorial for beginners – hacking with sql injection, n.d. . BinaryTides.

WebAIM: Creating Accessible Tables - Data Tables [WWW Document], n.d. URL <https://webaim.org/techniques/tables/data> (accessed 11.29.17).

# Appendix

| Discription | File |
| --- | --- |
| Project Proposal |  |
| Project Plan | Keep writing code until it works |
| Requirements specification |  |
| Project Test Plans |  |
| Monthly Journals |  |
|  |  |

