Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

Final Year Project

Research Report

Project Title: ClickNWin

Student: Geoffrey Atkinson

Student Number: C00184861

Project Supervisor: Greg Doyle

Date: 05/04/2017

Table of Contents

1.0 Introduction 3

2.0 Research Overview 3

3.0 Technology 4

3.1 ASP.NET MVC5 4

3.2 Python 5

3.3 HTML, CSS, and JavaScript 6

4.0 Mobile Development 7

4.1 Native Development 7

4.2 Hybrid Development 8

5.0 Database 8

5.1 SQL Server Express 8

5.2 MySQL 8

6.0 Encryption Algorithms 9

6.1 AES 9

6.2 Triple DES 9

6.3 HTTPS 10

7.0 Random Algorithm 10

8.0 Payment System 11

8.1 PayPal 11

8.2 Luhn Algorithm 12

9.0 Hosting Options 13

9.1 Self-Hosted 13

9.2 Microsoft Azure 13

9.3 PythonAnywhere 14

10.0 Conclusions 14

11.0 Glossary 15

References 16

1.0 Introduction

The primary goal of this project is to develop an application that can securely generate scratch cards like the physical national lottery scratch cards, allow users to pay for them, redeem any prizes won and send them as a gift to friends. The application also requires a high degree of security as malicious users must be prevented from spoofing scratch cards to pretend they have won a prize or gaining access to the application database which may contain personal information of users or their payment details.

This report will detail the research that has been carried out to facilitate the development of the application. It will focus mostly on the technologies to be used to develop the application, the primary algorithms that will need to be implemented in the application and the security that is required to protect it.

2.0 Research Overview

The first area of research that was undertaken for this project was looking at the technologies and programming languages that would be used to build the ClickNWin application. It was decided early in the project inception stages that the application would be built as a web application with a view to possibly porting it to the mobile platform as well. To that end, preliminary research was focused on the technologies that could be used to build a web application and would also have easy portability to mobile platforms. The first of these was the ASP.Net MVC framework which uses C# and HTML, CSS, and JavaScript. Research was also conducted on the Flask framework which uses Python along with HTML, CSS, and JavaScript as well.

Research was also done on mobile development and how to achieve it. Two main methods of mobile development were looked at. These were native and hybrid applications. The advantages and disadvantages of both were looked at in the context of the project with attention paid to how easy it was to port applications into them.

As the application was required to store user details along with other information, database technologies were researched to help decide what database would be a right fit for the project and would also integrate well with the chosen project programming languages and technologies. Databases that used SQL technologies were paid close attention to as these would integrate well with the ASP.Net and Flask frameworks that were being considered for the project.

With the high degree of security that is required for the ClickNWin project, a lot of research was put into encryption algorithms to decide which ones would be the most secure and a good fit for the project. Two algorithms were researched for this purpose. They were AES and Triple DES. There was also research done on using Hyper Text Transmission Protocol Secure (HTTPS) to secure transmissions between the user's browser and the application server.

To allow for the random generation of scratch cards, an algorithm was researched to randomly generate whether scratch cards would have a prize or not and what prize it would be. A description of this algorithm and a test that was ran on it to determine its accuracy are contained later in the report.

The application should also be able to take payment from the users for the scratch cards. To achieve this the PayPal sandbox system was researched so it could be used in the project to mock up payments for the cards in a more realistic way.

3.0 Technology

3.1 ASP.NET MVC5

One of the technologies researched for this project is ASP.Net MVC5. ASP is a web framework released and supported by Microsoft. It is used for building web applications with C#, HTML5, CSS and JavaScript. It uses the model-view-controller pattern which is an architectural design model that splits the components of an application into separate parts to allow for easier development and testing [1]. Some of the advantages of using ASP.Net MVC include separation of concerns that is implemented with the MVC design pattern, more control over the HTML that is rendered on the web pages and ease of integration with other technologies [2]. ASP also allows for multiple views to be defined and displayed depending on certain conditions. This includes displaying pages designed for either mobile or desktop clients depending on the device that is being used to view the web application [3].

The MVC pattern separates the application into models, views, and controllers. Models work with the application's data, controllers handle business logic, receive events from the views and provide data to the views. This separation of the various components makes the code more reusable and easier to maintain by removing dependencies between multiple components [22]. Using this software pattern for ClickNWin helps to promote the use of good software design principles in the project and allows readable, easily maintainable code to be created.

For this project, ASP.Net MVC5 is being considered as a framework to build the web application for ClickNWin. The ASP framework is supported by Visual Studio which is the project team's Integrated Development Environment(IDE) of choice. HTML5, CSS and JavaScript are commonly used languages when creating web applications and ASP.Net supports the use of all these technologies.

3.2 Python

Another technology that was researched for this project was Python. Python can be used to build server side scripts for web applications [4] in conjunction with HTML5, CSS and JavaScript used to build the front end and run client side scripts. Python can use several different frameworks for web application development such as Django and Flask. Python comes with a large amount of open source libraries and works well cross platform [5]. Python is also good at dealing with large amounts of data and can handle high traffic on a web server [6]. Python is available to run through the Visual Studio with use of a plugin but can be used in many other IDE's as well. When creating web applications, there are several Python frameworks to choose from but as the scale of this project is quite small the Flask framework was researched. Flask was developed around 2010 and, as stated, is aimed towards smaller web application projects [23]. Unlike Django, Flask allows developers to choose the database and templating style for the application [23]. There are several that work well with Flask such as MongoDB and SQLAlchemy and MySQL [23]. MySQL is being considered as the database of choice for this project along with SQL Server which also works with Flask.

Python does have some issues though. As Python is an interpreted language instead of a compiled one, this can cause some performance issues at runtime compared to compiled languages [7]. Python is also difficult to secure against attack and therefore is not run in browsers [7].

3.3 HTML, CSS, and JavaScript

Going hand in hand with both the ASP.Net framework and Python are three fundamental web based languages, HTML5, CSS and JavaScript. HTML5 is a web mark-up language that is used to create web pages. It uses tags to create different elements on the page and these tags can have attributes which allow for styling, dynamic content etc. HTML5 comes with a slew of benefits including better support for mobile development, a greater level of interactivity and features offline local storage [18]. These features will bring some useful benefits to the ClickNWin project. Mobile development is being considered as a secondary element to the project with a view to creating a hybrid application which would utilise web technologies like HTML5. By using HTML5 for a hybrid application, it would help with cross platform support as most browsers support HTML5 [18] so there would be no need to build several native applications to support different mobile operating systems and different versions of these operating systems. One of the new interactive features in HTML5 is the canvas element. The canvas element allows graphics to be created on the page which can then be accessed with a JavaScript API to draw dynamic graphics [19]. This could be utilised in the application to render scratch cards to the screen with dynamic graphics instead of using pre-saved images.

CSS is styling language that is used in conjunction with HTML to define the visual appearance of a web page and make it easier for developers to separate the functionality of the page from the style [20]. CSS is used to create style sheets which then tell the browser how to render the web page that is going to be displayed [20]. CSS can also be used to define how the web page will look when viewed on different devices such as smart phones [20]. This will be useful for ClickNWin if a mobile app is developed, it will allow the pages to be rendered differently when being view on either a desktop browser or a phone app. CSS will be used in the project to contribute to the user experience by creating a visually appealing web application that will draw users in.

JavaScript is a client side scripting language that can be used with HTML and CSS to create a more dynamic experience for a site's users. JavaScript runs within a user's browser so long as the browser supports it and it is enabled [21]. It can be used to create scripts that can dynamically affect a web page to create an interactive experience. An example would be using JavaScript in a form to display different options for a choice box based on a choice made in a previous choice box. If it is decided to use the canvas functionality in HTML, then JavaScript will be needed in the application to access the API required to use the canvas. Some smaller uses for JavaScript in forms or pages may also crop up as the project progresses. These could include form validation, creating extra input areas or displaying error messages.

JavaScript can also be used to implement Asynchronous JavaScript and XML(AJAX). AJAX is a web application technique that allows the client to make asynchronous requests to the server using JavaScript and the server back end [24]. This means that data can be stored or retrieved without the need to reload the entire web page. It can be implemented quite easily in JavaScript and an API can be set up on the server to receive the AJAX calls and return appropriate data. AJAX would be useful in a few places in ClickNWin. As all the application's users will have to pick a unique username, AJAX could be used to asynchronously check the username a person has picked and tell them if it is already taken without the need to submit the registration form. There will most likely several other areas that AJAX functionality would prove useful in the application so this technology will probably be used in some form.

4.0 Mobile Development

One of the issues that will arise during development of the ClickNWin application is its usage on mobile devices. The mobile application market is expanding rapidly [8] and it must always be considered when developing web applications whether the application is more suited to web or mobile development. When considering mobile app development, there are two main methods that will be considered for this project. These are native development and hybrid development.

4.1 Native Development

The first is native app development. Native applications are developed in the native language of the platform of choice. This provides them with full access to the device's hardware such as camera, GPS etc. [9]. For Android, the native language is Java. On IOS, it is Objective C and Swift.

4.2 Hybrid Development

The other option is hybrid app development. Hybrid allows web applications to be built and then wrapped with a wrapper for the specific phone platform. This allows for a single code base to be used and ported easily across multiple platforms [8]. Hybrid applications also allow easy deployments to different platforms and can be built using a single framework which eliminates the need to use multiple IDE's [11]. This means mobile applications can be built much quicker and far easier by eliminating the need for separate applications to be developed. A development team will not need extensive knowledge of different languages, IDE's, etc. needed to support the development of an application. Hybrid development would be the best option for ClickNWin as there isn't enough time in the project to develop a native application that works across multiple devices.

5.0 Database

5.1 SQL Server Express

The project will require a database to persistently store the various amounts and types of data that the project will require. SQL Server Express is one database that is being considered for use in the project. SQL Server Express is a free edition of the mainstream application SQL Server distributed by Microsoft. Express comes with many of the same features as SQL Server but is missing some advanced features and has some restrictions such as a 10GB size limit and the use of only one CPU [12]. SQL Server Express uses T-SQL to query the database which is Microsoft's extension of the SQL language [13]. SQL Server Express can be used with any of the technologies identified for building the project in.

5.2 MySQL

Another database to be considered is MySQL. MySQL is an open source database management system and is one of the most popular [13]. It is multi-platform, easy to install and has great community support [14]. It uses basic SQL commands for querying the database. However, MySQL does suffer from some poor performance issues and it also doesn't handle scaling very well [14]. If the ClickNWin were to progress to becoming a popular application, this could cause problems in the future. A MySQL database could be used with either ASP.Net or Flask as it is compatible with both.

6.0 Encryption Algorithms

6.1 AES

To protect the application data from malicious attack, the project should implement encryption techniques to protect sensitive user data. Advanced Encryption Standard(AES) is an encryption algorithm that is the national standard of the US government. AES is one of the most popular encryption algorithms due to its speed and security against attack [15]. It is a symmetric key algorithm which uses key lengths of either 128, 192 or 256 bits [15]. So far no one has been able to break the AES encryption directly. It is estimated that a brute force attack against AES using a 128 bit key would take 3.4e + 38 years. Exploits have been used on some AES products to retrieve the encryption keys but the encryption algorithm itself has not been broken. [15]. This makes AES ideal for use in any application that requires encryption security.

6.2 Triple DES

Another encryption standard that could be used for the project is Triple DES. Triple Des extends the old 56 bit key of DES by encrypting the data then decrypting it with a different key before encrypting it again. However, experts are still not convinced of Triple DES's security and believe it may be cracked soon [16]. The level of security required for ClickNWin is quite high so would necessitate the use of an extremely secure encryption algorithm. Therefore, Triple DES should not be considered for the project.

6.3 HTTPS

Encryption algorithms will secure the database contents but a way of securing information sent from the client browser to the server is also needed. When information is sent between the two it is vulnerable to malicious attackers who could be listening in on any device that the information passes through. To prevent this. HTTPS is used to encrypt the transmissions. During the first connection, the client and server use a Diffie-Hellman key exchange to create a shared secret key. This key is then used to encrypt all data flowing between the two and prevent unauthorised access if it were to be intercepted [28]. The use of HTTPS will greatly increase ClickNWin's security as not only will the database contents be encrypted, all data coming to and from the server will be as well.

7.0 Random Algorithm

An important aspect of the ClickNWin project is the ability to generate scratch cards that have varying probabilities of success for the user. Each card, when generated, should have a chance to be a winning card or a losing card. If the card is a winner there should be varying degrees of probability for each prize that could potentially be won.

To implement this aspect of the project, an algorithm has been researched and modified that uses cumulative probability and a randomly generated number to decide if a card is a winner and decide the prize if it is a winner. For the algorithm, a list of five key value pairs was created which contained first prize, second prize, third prize, fourth prize and not a winner. Each prize was then matched with a number between 0.0 and 1 which was that prizes probability of being chosen. A random number between 0.0 and 1 was then generated and a cumulative value is initialised to zero. The probability of each prize is then added one by one on to the cumulative starting with the lowest chance. If the random number is less than this cumulative value, then the prize currently selected is the winning prize.

To test out the accuracy of this algorithm, a sample implementation was created, in the C# language, and ran one thousand times. The amount of times each prize was chosen was then collected and compared against the probabilities that had been given to that prize to test the accuracy of the algorithm. The result of this test are as follows:

Prize	**Probability**	**Expected Results**	**Actual Results**
1st	1%	10	7
2nd	5%	50	54
3rd	11%	110	133
4th	15%	150	147
Not a Winner	68%	680	659

Fig 1. C# results for the cumulative probability algorithm

These results prove that the algorithm is quite accurate at selecting the different prizes based on the assigned probability. While there is deviation from the expected results, this is expected due to the randomly generated number that the algorithm creates. The deviation is not high enough to have any concerns about the algorithm being inaccurate. Therefore, it is strongly recommended to implement this algorithm in the project to randomly generate the prizes for the scratch cards.

8.0 Payment System

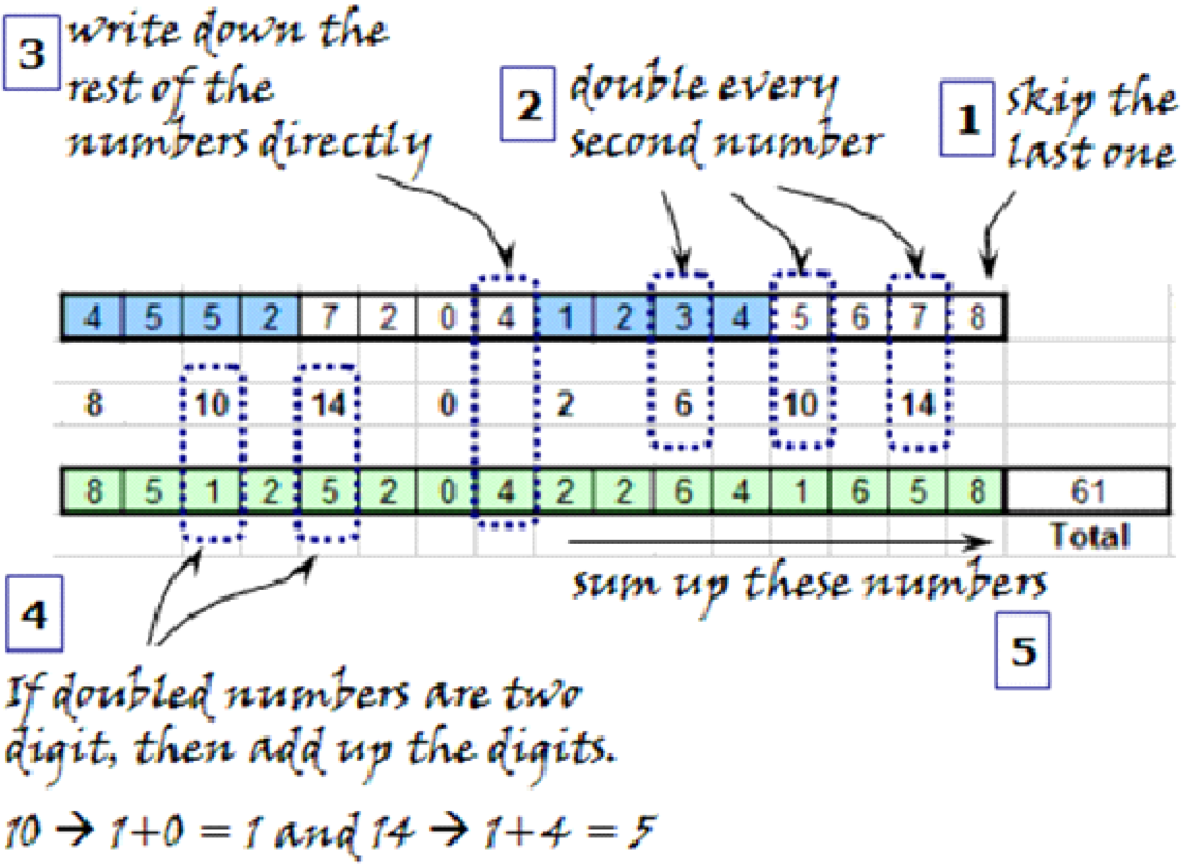
8.1 PayPal

As the ClickNWin will need to generate revenue from the scratch cards it will sell, a payment system must be set up to take payments from buyers and to allow buyers to be paid out with prizes they receive from the game. The best way that has been identified to accomplish this is the PayPal sandbox environment. The PayPal sandbox allows transactions on a site to be fully tested by simulating the real-world actions that would be used to process a transaction without any actual money being transferred between accounts [17].

Test buyer and seller accounts can be created in the sandbox and these accounts can then be used to simulate the real world behaviour of a web application when it is processing these transactions [17].

8.2 Luhn Algorithm

ClickNWin will be storing the credit cards of users who wish to top up their site balance with them. To assist with the storing of the credit card details, some basic client side validation was needed. The Luhn algorithm can be used for this. When financial institutions create new numbers for credit cards, the first six digits are used as an identifier and the next nine randomly generated. The final number is decided by doubling every second digit starting from the first number on the right. If the doubled number is greater than nine, then minus nine from it. Finally, get the sum of all the numbers and divide by ten and the whole number remainder is used as the final digit in the credit card number [29]. This means that by following the same formula on the full number, except starting from the second number from the right, that when the sum is divided by ten it will divide evenly. If a number is accidently swapped or omitted, the algorithm will pick that up and the user can be alerted. The following graphic shows the Luhn algorithm checking an incorrect number. Note that the solution for numbers greater than 9 when doubling is to add the number's two digits together. Doing this or subtracting nine will have the same outcome.



3 write down the rest of the numbers directly
2 double every second number
1 skip the last one
4 5 5 2 7 2 0 4 1 2 3 4 5 6 7 8
8 10 14 0 2 6 10 14
8 5 1 2 5 2 0 4 2 2 6 4 1 6 5 8
61
Total
sum up these numbers
4
5
If doubled numbers are two digit, then add up the digits.
10 → 1+0 = 1 and 14 → 1+4 = 5

Fig 2. Luhn algorithm representation [30]

The Luhn algorithm could be used in ClickNWin to validate card numbers entered by users when they are storing payment details. While the algorithm cannot detect fraudulent credit card numbers which satisfy the algorithm's check, it can detect minor error on the part of the user and allow an error message to be displayed asking them to check their card details. This would be useful in preventing users from unnecessarily storing cards with incorrect numbers.

9.0 Hosting Options

The ClickNWin application, being a web application, will need to be hosted in order for its users to be able to interact with the site. There are some options available for this. These are self-hosting, PythonAnywhere and Microsoft Azure.

9.1 Self-Hosted

Self-hosting is an option for ClickNWin. There are some good benefits to hosting your own application such as control over the application data and what goes onto the server [25]. However, these are outweighed by the negatives. Hosting your own server means expensive setup costs, the technical knowledge to configure the server and the server's maintenance [26]. By shipping out to a cloud provider, these costs can be avoided and the burden put on the provider.

9.2 Microsoft Azure

Azure is Microsoft's player in the cloud computing market. Azure provides support for several different languages and frameworks, has a rich array of features and offers support across a wide variety of services such as web development, access management and big data [27]. Azure's pricing will be a problem for ClickNWin. Azure provides only a one month free trial after which monthly payments must be made. There is no budget in ClickNWin for hosting. In addition, their customer service support is not considered very good [27] and with this being one of the major applications created by this team, a high level of support may be needed.

9.3 PythonAnywhere

PythonAnywhere is a cloud hosting provider exclusively based around the Python language. While it is limited in its choice of languages, PythonAnywhere provides an excellent free solution for hosting Python applications as well as the opportunities to purchase very cheap plans to get some extra features that might be needed. It also provides support for the MySQL database which can be hosted on the cloud alongside your application. It is also compatible with Git to allow web applications to be easily deployed.

10.0 Conclusions

Following the research that was done at the inception of the ClickNWin project, the project team have looked at the different options that were available for building the application and have decided to use Flask and the Python language as the main framework and programming language for building the web application. This framework will be supplemented with the use of HTML5, CSS and JavaScript as the building blocks for the web application's front end design. It was decided to go with Flask and Python because the project team have little experience with this technology and would like to create a project with it as a way of getting familiar with this web framework. Also, Python is can be made compatible with the project team's IDE of choice, Visual Studio, with the use of a plugin.

ClickNWin will be hosted using the PythonAnywhere cloud hosting site. Self-hosting and Azure are not being considered due the prohibitive cost. PythonAnywhere fully supports the use of the Python language and so will be a good fit for ClickNWin as it is being created in Python. A free site can be used to host all features of ClickNWin and to also host its database. If some extra features are required further down the road of the project, then a small fee can be paid to get access to them.

Comparing the two databases that were considered for the project, it has been decided to use MySQL as the database of choice for the application. While SQL Server Express works well with Visual Studio, the restrictions like the 10GB database size have been considered too restrictive for the application and it has been decided to try use MySQL instead as it has features of a full database while still being open source. MySQL is also a good choice for ClickNWin as the hosting platform, PythonAnywhere, only provides support for MySQL.

Mobile development is being considered for the project but with the time constraints that are present in the development of the application it has been decided to use hybrid development if time allows. Hybrid development will allow easy porting of the completed web application onto a mobile platform whereas there would not be enough time to write several different native applications if native development was chosen. Mobile development is not a definite feature that will be included in the project but is an option for the final iteration if time allows.

After comparing the researched encryption algorithms, it has been decided to use AES as the encryption standard of choice for the ClickNWin application. AES is one of the most secure encryption algorithms on the market and is virtually unbreakable without having the secret key. This makes it ideal for protecting the application's data. The database encryption will be supplemented with the use of HTTPS to secure transmissions between client and server.

11.0 Glossary

1. Spoofing: hoax or trick a system.
2. Database: Persistent storage space for an application's data
3. Encryption: The process of encoding data to prevent unauthorised parties from reading it.
4. API: An Application Program Interface defines a set of routines, protocols and tools for building software applications.
5. MVC: Model View Controller is a software pattern that defines a web application as three separate layers, each with its own responsibilities.
6. Algorithm: In computing terms, it is a set of instructions used to complete a pre-defined task.

References

[1] – Aspnet4/Anon. 2011. ASP.net MVC. [ONLINE]. Available at: http://aspnet4.com/asp-net-4-thoery/asp-net-mvc/. [Accessed 11 October 2016].

[2] – Cmarix/Anon. 2016. 6 benefits of ASP.net MVC. [ONLINE]. Available at: http://www.cmarix.com/6-benefits-of-asp-net-mvc/. [Accessed 11 October 2016].

[3] – Simple Talk/Dino Espositio. 2013. Multiple Views and DisplayMode Providers in ASP.NET MVC 4. [ONLINE]. Available at: https://www.simple-talk.com/dotnet/asp-net/multiple-views-and-displaymode-providers-in-asp-net-mvc-4/. [Accessed 11 October 2016].

[4] – Full Stack Python/Matt Makai. 2016. Web Development. [ONLINE]. Available at: https://www.fullstackpython.com/web-development.html. [Accessed 12 October 2016].

[5] – Full Stack Python/Matt Makai. 2016. Why Use Python. [ONLINE]. Available at: https://www.fullstackpython.com/why-use-python.html. [Accessed 12 October 2016].

[6] – Up Work/Yoshitaka Shiotsu. 2014. Web Development 101: Top Web Development Languages in 2014. [ONLINE]. Available at: https://www.upwork.com/blog/2014/03/web-development-101-top-web-development-languages-2014/. [Accessed 12 October 2016].

[7] – Info World/Paul Krill. 2015. A developer's guide to the pros and cons of Python. [ONLINE]. Available at: http://www.infoworld.com/article/2887974/application-development/a-developer-s-guide-to-the-pro-s-and-con-s-of-python.html. [Accessed 12 October 2016].

[8] – Infoq/Daniel Jebaraj. 2013. Hybrid mobile apps with ASP.Net MVC. [ONLINE]. Available at: https://www.infoq.com/articles/Hybrid-Mobile-Apps-MVC. [Accessed 12 October 2016].

[9] – About Tech/Priya Viswanathan. 2016. The Pros and Cons of Native Apps and Mobile Web Apps. [ONLINE]. Available at: http://mobiledevices.about.com/od/additionalresources/qt/The-Pros-And-Cons-Of-Native-Apps-And-Mobile-Web-Apps.htm. [Accessed 12 October 2016].

[10] – Y Media Labs/Robbie Abed. 2016. Hybrid vs Native Mobile Apps – The Answer is Clear. [ONLINE]. Available at: http://www.ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/. [Accessed 12 October 2016].

[11] – Phase 2 Technology/Peter Cho. 2015. The Benefits of Native vs. Hybrid Mobile Apps. [ONLINE]. Available at: https://www.phase2technology.com/blog/the-benefits-ofnative-vs-hybrid-mobile-apps/. [Accessed 18 October 2016].

[12] – Learning SQL Server/Jason Roth. 2011. What is SQL Server Express. [ONLINE]. Available at: https://learningsqlserver.wordpress.com/2011/01/26/what-is-sql-server-express/. [Accessed 18 October 2016].

[13] – SQL with Manoj/Manoj Pandey. 2015. What is SQL, PL/SQL, T-SQL and difference between them. [ONLINE]. Available at: https://sqlwithmanoj.com/2015/05/03/what-is-sql-plsql-t-sql-and-difference-between-them/. [Accessed 18 October 2016].

[14] – Data Realm/John Mack. 2014. Five Advantages & Disadvantages Of MySQL. [ONLINE]. Available at: https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/. [Accessed 18 October 2016].

[15] - Search Security/Margeret Rouse. 2014. Advanced Encryption Standard. [ONLINE] Available at: http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard. [Accessed 7 October 2016].

[16] – Gale Group/James Swann. 2005. Preparing for Triple DES security. [ONLINE]. Available at: http://0-go.galegroup.com.www.library.itcarlow.ie/ps/i.do?p=ITOF&u=itc_web&id=GALE|A140142391&v=2.1&it=r&sid=summon&userGroup=itc_web&authCount=1. [Accessed 19 October 2016].

[17] – Tips and Tricks HQ. 2014. Playing in PayPals's Sandbox. [ONLINE]. Available at: https://www.tipsandtricks-hq.com/playing-in-paypals-sandbox-2880. [Accessed 19 October 2016].

[18] – Patrick Cox/Tympanus. 2011. Top 10 Reason to Use HTML5 Right Now. [ONLINE]. Available at: http://tympanus.net/codrops/2011/11/24/top-10-reasons-to-use-html5-right-now/. [Accessed 4 November 2016].

[19] – Ivaylo Gerchev/Sitepoint. 2014. HTML5 Canvas Tutorial: An Introduction. [ONLINE]. Available at: https://www.sitepoint.com/html5-canvas-tutorial-introduction/. [Accessed 4 November 2016].

[20] – Jenniffer Krynin/AboutTech. 2016. What is CSS and Where is it Used. [ONLINE]. Available at: http://webdesign.about.com/od/beginningcss/a/aa021607.htm. [Accessed 4 November 2016].

[21] – Stephen Chapman/AboutTech. 2016. What is JavaScript. [ONLINE]. Available at: http://javascript.about.com/od/reference/p/javascript.htm. [Accessed 4 November 2016].

[22] – Tom Dalling/Tom Dalling. 2009. Model View Controller Explained. [ONLINE]. Available at: http://www.tomdalling.com/blog/software-design/model-view-controller-explained/. [Accessed 4 November 2016].

[23] – Ryan Brown/AirPair. 2015. Django vs Flask vs Pyramid: Choosing a Python Web Framework. [ONLINE]. Available at: https://www.airpair.com/python/posts/django-flask-pyramid. [Accessed 14 November 2016].

[24] – Seque Technologies. 2013. What is Ajax and Where is it Used in Technology? [ONLINE]. Available at: http://www.seguetech.com/ajax-technology/. [Accessed 28 March 2017].

[25] – Danny Stieben/MakeUseOf. 2012. 5 Reasons Why You Should Make Your Own Server. [ONLINE]. Available at: http://www.makeuseof.com/tag/5-reasons-server/. [Accessed 28 March 2017].

[26] – Sam McKinney/Spinweb. 2015. Pros and Cons of Cloud Hosting vs. Internal Hosting. [ONLINE]. Available at: https://blog.spinweb.net/pros-cons-cloud-hosting-vs-self-hosting. [Accessed 28 March 2017].

[27] – Todd Lyle/Technopedia. 2015. The Four Major Cloud Players: Pros and Cons. [ONLINE]. Available at: https://www.techopedia.com/2/31280/trends/the-four-major-cloud-players-pros-and-cons. [Accessed 28 March 2017].

[28] – Hartley Brody. 2013. How HTTPS Secures Connections: What Every Web Dev Should Know. [ONLINE]. Available at: https://blog.hartleybrody.com/https-certificates/. [Accessed 28 March 2017].

[29] – Jay McDonald/Credit Cards. 2010. How the 'Luhn formula' validates credit card numbers. [ONLINE]. Available at: http://www.creditcards.com/credit-card-news/luhn-formula-credit-card-number-system-1273.php. [Accessed 28 March 2017].

[30] – Rakesh Sreekanth/Baykon Blogspot. 2008. Numbers....we got to deal with. [ONLINE]. Available at: http://baykon.blogspot.ie/2008/03/numberswe-got-to-deal-with.html. [Accessed 28 March 2017].