|  |  |
| --- | --- |
|  |  |
| rgu logo colour |
|  | |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| **COURSEWORK**  **DUE DATE** | 14/12/2019 |
| **MATRIC No** | **1909247** |
| **SURNAME** | **Chatzileftheris** |
| **FIRST NAME(S)** | **Stefanos** |
| **COURSE TITLE** | **IT for Oil and Gas Industry** |
| **MODULE NUMBER**  **& TITLE** | **CMM007 – Intranet System Development Part 3** |
| **LECTURER ISSUING COURSEWORK** | **Stewart Massie** |

**RGU Research Paper Project (CMM007-PART 3)**

**Abstract:**

In this report I am going to analyse with all the necessary details my project. I will recommend some new techniques that I could add into my project and I will explain what kind of system architecture I am using and how can I improve that. Moreover, I will give some scalability solution and give some idea for the security.

Description: This project was designed to implement a web application that will help group project to upload, identification, review, monitoring and following of research papers between the group members. The research activities will be report writing, review and other similar tasks can be to monitor and track the editorial progress of several papers within a limited time.

1. **Analysis of current limitations:**

The name of this project is RGU research paper, in this system they will be three types of users: Administrator, Student Team Leader and Students. All users will have access to the register system i.e. members will be required to log in. The administrator has the highest role to manage the tools on the platform, and he will only have the access to create users, setup project groups, allocate team leader role and assign members to the project. The team leader is responsible for delivering papers to members for review and. Students will be able to upload research papers, this will be submitted to the Student team leader, who will allocate it to any members for review. Furthermore, this project will give the opportunity for the user to create platform that will support group project teams.

***Strengths:***

The created web application will contain the following features. More than one user role, the application will be able to support users with different roles assigned various degrees of permissions and benefits in what a user is able of doing. Some type of file upload system and: This will facilitate the uploading of research papers on the server for storing and download at request. This will be implemented using HTML and PHP for executing upload to remote server. A system for user to input data that is stored and then recalled from a database: The system will support data storage for example documents or reviewing papers and retrieval using HTML, PHP in the application and MySQL(phpMyAdmin) for the database server. The application will be hosted on a local server (Xampp) accessible only with a request.

In addition, the application must contain both front end (client side) and (server side) code. The front-end is the presentation layer that the user will use to perform tasks on the web app. The front-end is the user-friendly Graphical User Interface built. It is built using HTML, CSS, Font awesome and Bootstrap. The backend is the logic and data layer. It executes user's requests, by performing query based on the request and return response to the user on the front-end.

***Weakness:***

In the project you are not able to change the password as a user and team leader and you are not able to change your photo profile. In addition, you are not able to visit profiles. Moreover, I didn’t implement any alert for any action which is a must implement for the future improvements. As users in the login system forum need to have a button if someone forgot the password to login. The application shall be able to function with other hardware devices. All the users should be authorized to use the system and try to avoid cyber and physical attacks. For the database: A backup must be saved to avoid any disaster and be recovery in cases of disaster. Moreover, the cookies: The application could have a cookie so the users should not type every time the password to get to the app.

As an Administrator you can’t not edit the Project but only you can create them. Furthermore, as Admin you cannot change information through the system but only from the database (localhost xampp).

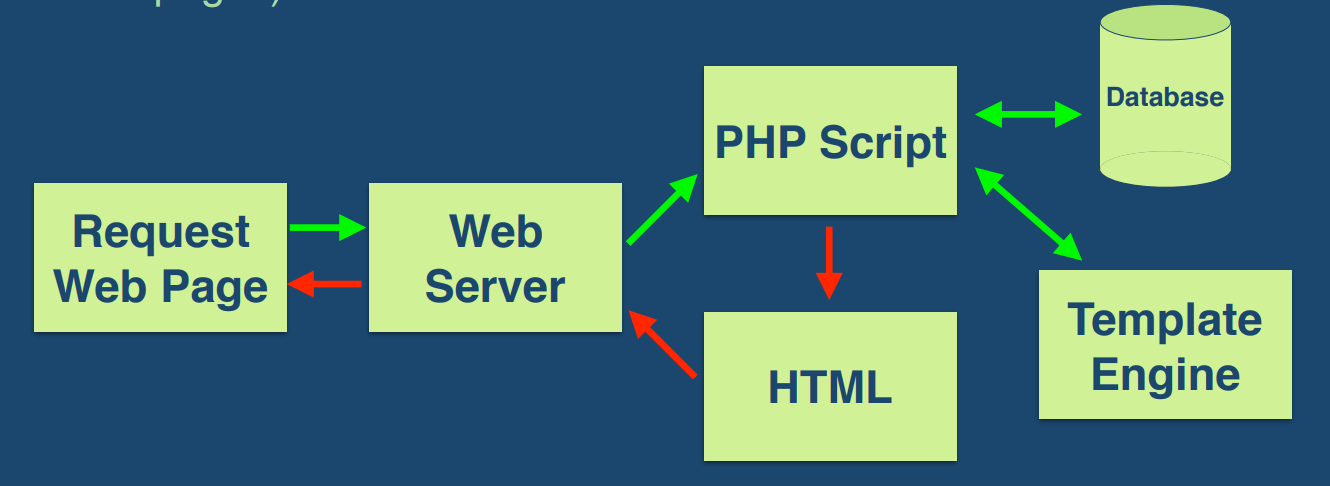
As a User and team leader you are not able to edit your profile and change your password inside the system. A team leader has to view the projects and the users which is not implement into my project. In addition, as a team leader can remove from a project a member if it’s necessary or edit the assigned.

1. **Recommended Technologies:**

***Server-side technologies:***

The term "server technologies" may include several software solutions, mainly: server scripting languages. Database Management Systems (DBMS), web server software such as Apache specific for my RGU research project paper. The essential combination of technologies required to create a service is known as a "software solution stack" and the original and most commonly used web software stack is known as LAMP (Apache, MySQL and PHP)which I am not using into my application but it would be ideal . It is worth noting that the stack must include four elements: an operating system, a web server presence, a database management system and a server batch language.

The application code is usually stored on the server. Customers make requests to the servers. Then the server responds to these requests after gathering the required information.

******

***Data Storage:***

An alternative way to design my database will be using MongoDB. MongoDB is best suitable for hierarchical data storage, but RDBMS is not. MongoDB supports JSON query language along with SQL, but RDBMS supports SQL query language only. Moreover, MongoDB is almost 100 times faster than traditional database system like phpMyAdmin, which is slower in comparison with the NoSQL databases.

***Client-side technologies:***

To improve my project by the side of client-side, I could use Ajax and jQuery. The two huge advances in scripting on the part of the client are Ajax and jQuery. Ajax is not a tool or programming language, but a concept. The idea is to call the server directly from the client without doing Post-Back. The main advantage of Ajax is the storage and retrieval of data from a database, bypassing the web server, which is known as Call-Back which better for my project

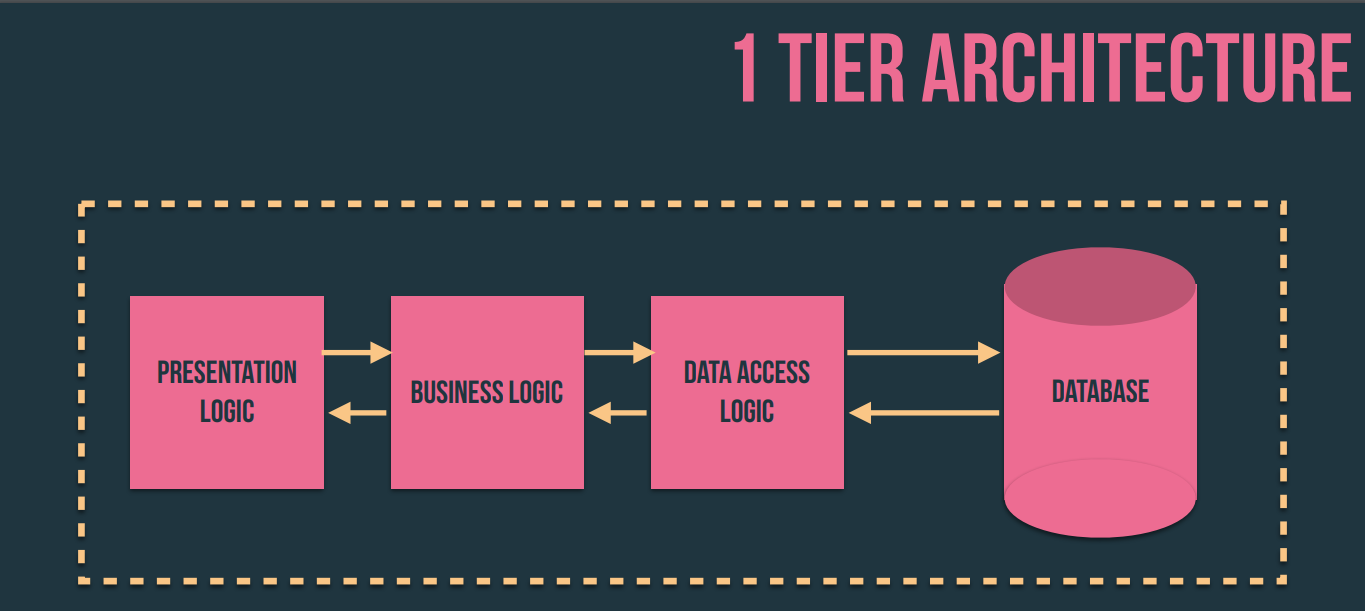
According to their website, jQuery is defined as, a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

In addition, I could improve the style sheets, presentation and style which could be separated, from the information structure and content, Cascading Style Sheet (CSS), Mixed Media.

1. **Proposed system architecture:**

***3.1) What I am using***

I am using the first-tier architecture for my project because of the situation. I only use a localhost from my apartment xampp to do all the necessary changes in php for my project. My project follows the three layers down below which are presentation, business and data layer, which I am going to explain the layer down below.



**Presentation Logic Layer:** Using this level, we can access the websites. The primary function of this layer is to communicate with the application layer. This level transmits the information supplied by the user in terms of keyboard events, mouse-click the application level. In a simple word, it is to see the implementation.

**Application/Business Logic Layer:** It is also known as the Business Logic Layer, which is also known as Layer Logic. According to the example of the login page with the message, after the user clicks the Connect button, the application layer interacts with the level of the database and sends the required information to the presentation level. It checks the functionality of an application by making detailed processing. This layer acts as an intermediary between the level of presentation and database.

**Data Logic Layer:** The data stored at this level. The level of application communicates with the database level to recover the data. It contains methods that link the database and perform the required actions, such as: insert, update, delete, etc. Which in this situation I am using xampp (localhost).

On the other hand, the first tier is very bad for scalability (single processor being used), bad for portability and bad for maintenance (if you want to change one thing).

***3.2) Improvements for system architecture***

In my opinion the best architecture system is the N-tier because every layer can be on a different machine, easier to maintain, components are reusable, and it will have faster division of work. So, there are several benefits to using n-tier architecture for your software.

**Secure:** You can secure each of the three tiers separately using different methods.

**Easy to manage:** You can manage each tier separately, adding or modifying each tier without affecting the other tiers.

**Scalable:** If you need to add more resources, you can do it per tier, without affecting the other tiers.

**Flexible:** Apart from isolated scalability, you can also expand each tier in any manner that your requirements dictate.

In short, the architecture n-tier, you can adopt new technologies and add more information without having to rewrite the entire application, or redesigning your entire software, making it easier to scale and maintain. Meanwhile, in terms of security, you can store sensitive or confidential information in logical level, keeping them away from the presentation level, thus making it safer. Other benefits include: More efficient development, Easy to add new features and Easy to reuse.

1. **Scalability Solutions:**

***4.1) Scaling services:***

We have to two types of scaling our application:

**Vertical scaling**:

Can essentially resize your server with no change to your code. It is the ability to increase the capacity of existing hardware or software by adding resources. Vertical scaling is limited by the fact that you can only get as big as the size of the server.

**Horizontal scaling**:

Affords the ability to scale wider to deal with traffic. It is the ability to connect multiple hardware or software entities, such as servers, so that they work as a single logical unit. This kind of scale cannot be implemented at a moment’s notice.

So, having said all that, for my project it will better fit the horizontal scaling. Furthermore, scaling services up for use in an enterprise-level web program, there are 4 methods:

***Caching:***

Caching is a way to store and reuse the same data multiple times. By data, I mean anything like images, CSS files, JSON, etc. Caching will help you serve more requests per second and save on precious resources like network bandwidth and CPU load.

***Proxies:***

Proxies are good when lots of people are searching for the same thing. Also, good if people are searching for close to the same thing. We can use techniques at the same time to make it even faster. A lot of proxies come with a cache built into them.

***Load Balancing:***

Load balancing refers to the process of distributing a set of tasks over a set of resources (computing units), with the aim of making their overall processing more efficient. Load balancing techniques can optimise the response time for each task, avoiding unevenly overloading compute nodes while other compute nodes are left idle.

***Queuing:***

Queueing theory is the mathematical study of waiting lines, or queues. A queueing model is constructed so that queue lengths and waiting time can be predicted. Queueing theory is generally considered a branch of operations research because the results are often used when making business decisions about the resources needed to provide a service.

***4.2) Scalable designs:***

MVC-(Model-View-Controller). The MVC is one application design model consisting of three interconnected parts. The MVC model or "pattern" is commonly used for the development of modern user interfaces. It provides fundamental pieces for program design for desktop or mobile, and web applications. I could use PHP in MVC which can be:

***Model***

Responsible for managing the data of the application. Responds to the request from the view and responds to instructions from the controller to update itself. The it is the lowest level of the pattern which is responsible for maintaining data and represents the application core.

***View***

The view displays the data and requests information from the model, that it needs for an output. The it presents data in a format. The MVC is often seen in web applications where the view is the HTML page. It can be any output representation of data and chart. We can multiple views of the same data are possible and bar charts for management.

***Controller***

The controller is the part of the application that handles user interaction. The typically controllers read data from a view, control user input and send input data to the model. In addition, we handle input, typically users’ actions and may invoke changes on the model and view.

1. **Application Security**

To begin with, the technology you put in place to secure a system it’s usually based on three aspects:

* Operating System Security - Securing the OS that your system is working on
* Network Security - Securing the connections to your server
* Application Security - Securing the application that you write

Moreover, to secure my code I need to take into consideration

***Least Privilege*:** Only the required access to do a job should be given. Should be limited for the time needed.

***Fail-Safe Defaults*:** Access must be granted for each user.

***Economy of mechanism:*** Systems should be designed to be as simple as possible. Design implementation errors result in unauthorised access.

***Complete Mediation:***Access to every resource must be validated for Authorisation.

***Open design:*** The security of a system should not be dependent on secrecy of its design or implementation

In addition, I need to improve my security into my project when creating my PHP code.

**1) Exposed Access Credentials:**

Most PHP applications interact with a database which involves connecting to database server using credentials, typically in a file db.php included if connection needed, problems arise when file is in document root, file can then be accessed via url . A simple solution is to place file outside my document root, include can now use the file path.

**2) Cross Site Scripting (XSS) Vulnerabilities:**

Attack against an authenticated user using cookies. A user browser tricked into sending requests. Prevented by synchronizing Cookie with an anti-CSRF token.

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications.

XSS enables attackers to inject client-side script into Web pages viewed by other users.

A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

**3) Cross Site Forgery Requests:**

The web application vulnerability that allows attacker to inject code

To prevent that we can input checking to block abnormal requests and can use firewalls with signature filtering

**4) SQL Injection:**

SQL injection is a code injection technique that exploits a security vulnerability in an applications software.

The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. To prevent SQL injection, we can use:

***Parameterized Statements:***

Programming languages talk to SQL databases using database drivers. A driver allows an application to construct and run SQL statements against a database, extracting and manipulating data as needed. Parameterized statements make sure that the parameters (i.e. inputs) passed into SQL statements are treated in a safe manner.

***Object Relational Mapping:***

Many development teams prefer to use Object Relational Mapping (ORM) frameworks to make the translation of SQL result sets into code objects more seamless.

***Escaping Inputs:***

Programming languages have standard ways to describe strings containing quotes within them – SQL is no different in this respect. Typically, doubling up the quote character – replacing ' with '' – means “treat this quote as part of the string, not the end of the string”.

***Sanitizing Inputs:***

Developers should always try to reject inputs that look suspicious out of hand, while taking care not to accidentally punish legitimate users. For instance, your application may clean parameters supplied in GET and POST requests

**Conclusion:**

The project was implemented well with some small issues, but it needs a lot of improvements. First, every use needs to have a functionality to change photo and password. In addition, I will need to have any Alert statement for every forum to warn the user and inform them about the situation, for instance (if they upload the files successful or not).

Furthermore, admin needs to edit the project and the users need to edit the paper before sharing. I need to change the security system and make it encrypt for hackers in my database.

Moreover, the tier architecture must change from 1- tier to n-tier for a better to use. I need to use an alternative way for my technology to improve the client-side, server-side and data storage for research paper project. In addition, the scalability of my project must be easily upgradable to make space for additional content. Lastly the application security must be improved in the system security, network security and application security in order to have a saver system for the future.

# **References:**

Beaumont, D. (2014). *IBM*. Retrieved from How to explain vertical and horizontal scaling in the cloud: https://www.ibm.com/blogs/cloud-computing/2014/04/09/explain-vertical-horizontal-scaling-cloud/

*GEOG5870/1M: Web-based GIS, University of Leeds*. (n.d.). Retrieved from Server Side Technologies: https://www.geog.leeds.ac.uk/courses/postgrad/web/lectures/server-side-technologies/1\_introduction/index.html

*Hacksplaining*. (n.d.). Retrieved from protecting against sql injection: https://www.hacksplaining.com/prevention/sql-injection

Massie, S. (2020). *Intranet System Development Lecture 1.* Aberdeen: Robert Gordon University.

Massie, S. (2020). *MVC and its use in Web Lecture 9.* Aberdeen: Robert Gordon University.

Massie, S. (2020). *Scaling up Web Applications Lecture 7.* Aberdeen: Robert Gordon University.

Massie, S. (2020). *WebSecurity Lecture 10.* Aberdeen: Robert Gordon University.

Punetha Sarmila, G., Gnanambigai, N., & Dinadayalan, P. (2015). *Wikipedia*. Retrieved from Load balancing (computing): https://en.wikipedia.org/wiki/Load\_balancing\_(computing)

Rajkumar. (2016). *Software Testing Material*. Retrieved from Software Architecture: One-Tier, Two-Tier, Three Tier, N Tier: https://www.softwaretestingmaterial.com/software-architecture/

Sundarapandian, V. (2009). *Wikipedia*. Retrieved from Queueing theory: https://en.wikipedia.org/wiki/Queueing\_theory

Technologies, S. (2013). *Segue 3 Technologies*. Retrieved from How Does Client-Side Scripting Improve Web Application Performance?: https://www.seguetech.com/client-side-scripting-performance/