



Ministry of Higher Education and Scientific Research

University of Technology

Computer Engineering Department - Academic Year 2022-2023



Design and Implementation Secure E Learning System Using SQL Injection

Graduation project

**Submitted to the Computer Engineering Department in partial
fulfilment of B.Sc. degree in Information Engineering**

By

Mustafa Basher Hatem

Basseim Hussein Abdul Amir

Supervised by

Assistant Zainab Mahmoud

2022-2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قال تعالى: (اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ * خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ * اقْرَأْ وَرَبُّكَ الْأَكْرَمُ * الَّذِي عَلَّمَ بِالْقَلَمِ * عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ). "سورة العلق"

SUPERVISOR CERTIFICATION

I certify that the preparation of this project entitled (Design and Implementation Secure E Learning System Using SQL Injection) was prepared by (Mustafa Basher Hatem&Basseim Hussein Abdul Amir) under my supervision at the Information Engineering Branch, Computer Engineering Department, University of Technology in partial fulfillment of the requirements for the degree of B. Sc. In information Engineering.

Signature:

Supervisor Name:

Scientific Degree:

Date:

EXAMINATION COMMITTEE CERTIFICATION

We certify that we have read this project entitled (Design and Implementation Secure E Learning System Using SQL Injection), and as an examining committee examined the students (Mustafa Basher Hatem&Basseim Hussein Abdul Amir) in its contents, and our opinion, it meets the standards of B.Sc. in Information Engineering.

Signature:

Signature:

Name:

Name:

Scientific Degree:

Scientific Degree:

Date:

Date:

(Chairman)

(Member)

Signature:

Name:

Scientific Degree:

Date:

(Head of information Engineering Branch)

Abstract

The abstract of this project provides an overview of the research project. It begins by introducing the problem of SQL injection attacks on E-learning systems. It then describes the goal of the research project, which is to develop a secure E-learning system that can withstand SQL injection attacks.

The abstract then briefly describes the methodology used in the project, which involves designing and implementing a secure E-learning system and then testing it for vulnerabilities. It also highlights the importance of this research project, as E-learning systems are increasingly popular and the threat of SQL injection attacks is a major concern.

Finally, the abstract mentions the expected results of the project, which include identifying and addressing vulnerabilities in the E-learning system and demonstrating its security against SQL injection attacks. The overall purpose of the research project is to enhance the security of E-learning systems and provide a model for developing secure web applications that can withstand SQL injection attacks.

DEDICATION

الإهداء

أشكر الله العليّ القدير الذي أنعم عليّ بنعمة العقل والدين. القائل في محكم التنزيل “وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ” سورة يوسف آية 76.... صدق الله العظيم . وقال رسول الله (صلي الله عليه وسلم): “من صنع إليكم معروفاً فكافنوه، فإن لم تجدوا ما تكافنونه به فادعوا له حتى تروا أنكم كافأتموه” (رواه أبو داود) . وأثني ثناء حسنا على من علمني العطاء وإلى من أحمل اسمه بكل افتخار وأرجو من الله أن يمد في عمرك لتري ثمارا قد حان قطافها بعد طول انتظار “والدي العزيز” وإلى من أفضّلها على نفسي، ولمّ لا؛ فلقد ضحّت من أجلي ولم تدّخر جهداً في سبيل إسعادي على الدوام (أمي الحبيبة) وإلى من بهم أكبر وعليهم أعتد وإلى من بوجودهم أكتسب قوة ومحبة لا حدود لها وإلى من عرفت معهم معنى الحياة “إخوتي وأخواتي” وأيضا وفاء وتقديرا وإعترافا مني بالجميل أتقدم بجزيل الشكر لأولئك المخلصين الذين لم يألوا جهداً في مساعدتنا في مجال البحث العلمي، وأخص بالذكر الأستاذة الفاضلة (الست زينب محمود) على هذه الدراسة وصاحبة الفضل في توجيهي ومساعدتي في تجميع المادة البحثية، فجزاها الله كل خير. ولا أنسي أن أتقدم بجزيل الشكر إلى جميع أساتذتي الكرام الذي قامو بتوجيهنا طيلة هذه الدراسة ، وأخيراً ،أتقدم بجزيل شكري إلى كل من مدوا لي يد العون والمساعدة في إخراج هذه الدراسة على أكمل وجه.

TABLE of content

1	CHAPTER ONE	11
1.1	Introduction:	11
1.2	Aim.....	12
1.3	Plan of Design.....	13
1.4	The expected results.....	14
2	CHAPTER TWO.....	17
2.1	Introduction	17
2.1.1	Description:	17
2.1.2	Common variants of SQL injection include:	18
2.2	Finding SQL Injection	18
2.2.1	SQL injection based on HTTP headers:	19
2.2.2	SQL injection based on user input:	20
2.3	How to prevent SQL injection attacks	22
2.4	Cross-Site Scripting or XSS	25
2.4.1	Preventing XSS in HTML and PHP	25
2.5	File upload security	26
3	CHAPTER THREE.....	29
3.1	Thses are functionality performed by the admin users.....	29

3.2	Functionality performed by Teacher user:	30
3.3	Functionality performed by Student user:.....	31
3.4	Tools used in this project	32
3.5	Database	32
4	CHAPTER FOUR	41
4.1	CONCLUSION.....	41
4.2	FUTURE WORK.....	41

LIST OF FIGURE

Figure (1)	Database of E-learning System	36
Figure (2)	Register in page	36
Figure (3)	Log in page.....	37
Figure (4)	Flow Chart Login System	38
Figure (5)	sample design file upload system.....	39

ABBREVIATION

E-learning: Eelectronic learning

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets

JS: JavaScript

PHP: Hypertext Preprocessor

CHAPTER ONE

Introduction

1 CHAPTER ONE

1.1 Introduction:

E-learning has become an essential part of education, especially during the COVID-19 pandemic, where E-learning has become a popular choice for students and professionals alike to access educational resources and training programs from anywhere, at any time. However, with the rise of E-learning platforms, the security of these platforms has become a concern. One of the most significant security threats to E-learning platforms is SQL injection.[1]

SQL injection is a type of security vulnerability that allows attackers to execute malicious SQL statements on a web application's database. These malicious SQL statements can allow attackers to access, modify, or delete sensitive data from the application's database, compromising the security of the E-learning platform. Therefore, it is crucial to develop a secure E-learning system that can prevent SQL injection attacks. This system should implement various security measures, such as input validation, parameterized queries, and stored procedures, to protect against SQL injection attacks.

In this project , we will discuss the importance of secure E-learning systems and the impact of SQL injection attacks on these systems. We will also explore different methods of preventing SQL injection attacks and implementing security measures to ensure the safety of the E-learning platform.

Overall, a secure E-learning system is crucial for protecting the privacy and security of students' data and ensuring that they can learn without fear of cyber attacks.[2]

1.2 Aim

The aim of this topic is to examine the security risks associated with E-learning systems that use SQL databases and the impact of SQL injection attacks on the confidentiality, integrity, and availability of the system and its users' data.

SQL injection is a type of cyber attack that can exploit vulnerabilities in web applications that use SQL databases. Attackers can use SQL injection to execute malicious SQL commands and gain unauthorized access to sensitive data. E-learning systems that use SQL databases are particularly vulnerable to SQL injection attacks, which can compromise student grades, personal information, and other sensitive data.[3]

The aim of this topic is to identify the best practices and security measures that can be implemented to secure E-learning systems against SQL injection attacks. These measures include input validation, parameterized queries, and secure coding practices, among others. By implementing these security measures, E-learning systems can mitigate the risk of SQL injection attacks and other cyber threats, safeguarding the confidentiality, integrity, and availability of the system and its users' data. [11]

In summary, the aim of this topic is to raise awareness about the security risks associated with E-learning systems that use SQL databases, and to promote the implementation of robust security measures to protect against

SQL injection attacks and other potential cyber threats, ultimately enhancing the security and reliability of the system and its users.[4]

1.3 Plan of Design

The plan of design for this topic involves several steps that are necessary to create a secure E-learning system using SQL databases:[5]

1- Risk assessment: This step involves identifying potential security risks associated with E-learning systems that use SQL databases. The risk assessment should include a thorough evaluation of the system's architecture, design, and implementation to identify any vulnerabilities that could be exploited by attackers.

2- Security requirements: Based on the results of the risk assessment, specific security requirements should be identified for the E-learning system. These requirements should cover aspects such as access control, authentication, data protection, and auditing.

3- Secure coding practices: Developers should be trained in secure coding practices, such as avoiding the use of dynamic SQL, validating input data, and using parameterized queries. This can help prevent SQL injection attacks.

4- Input validation: Input validation is a critical step in preventing SQL injection attacks. All user input data should be validated to ensure that it is of the correct data type, length, and format.[6]

5- Parameterized queries: Parameterized queries are an effective way to prevent SQL injection attacks. Parameterized queries use placeholders for user input data, preventing malicious SQL code from being executed.

6- Regular security audits: Regular security audits should be conducted to identify any new security risks or vulnerabilities that may have emerged. The security audits should also review the effectiveness of the security measures that have been implemented.

By following this plan of design, E-learning system developers can create a secure platform that protects the confidentiality, integrity, and availability of the system and its users' data. Additionally, by conducting regular security audits and keeping up-to-date with new security threats, the E-learning system can adapt to evolving security risks and maintain a high level of security for users.[15]

1.4 The expected results

The expected results for this topic include the development of a secure E-learning system that uses SQL databases and is resilient to SQL injection attacks. This system should meet the following criteria:[7]

1- Confidentiality: The system should protect the confidentiality of user data, such as personal information and grades. This can be achieved through access controls and encryption. Access controls can limit access to sensitive data to authorized users only, while encryption can ensure that data is protected in transit and at rest.

2- Integrity: The system should ensure the integrity of user data, meaning that data should not be modified or deleted by unauthorized users. This

can be achieved through secure coding practices, input validation, and auditing. Secure coding practices can help prevent vulnerabilities that could be exploited by attackers, while input validation can ensure that data is in the correct format and within acceptable limits. Auditing can help detect unauthorized changes to data.

3- Availability: The system should ensure the availability of user data and services, meaning that users should be able to access the system and its services when needed. This can be achieved through redundancy, backup and recovery processes, and disaster recovery plans. Redundancy can ensure that there are multiple copies of critical data and services, while backup and recovery processes can ensure that data is recoverable in case of a failure. Disaster recovery plans can help ensure that the system can be quickly restored after a major outage.

4- Prevention of SQL injection attacks: The system should be resilient to SQL injection attacks, which can compromise the confidentiality, integrity, and availability of user data. This can be achieved through input validation, parameterized queries, and secure coding practices. Input validation can ensure that data entered by users is within acceptable limits and cannot be used to execute SQL injection attacks. Parameterized queries can help prevent SQL injection attacks by separating user input from SQL code. Secure coding practices can help prevent vulnerabilities that could be exploited by attackers.[14]

5- Compliance with relevant regulations: The system should comply with relevant regulations, such as data protection laws, to ensure that user data is handled in accordance with legal requirements. Compliance can help ensure that the system is secure and that user data is protected.

CHAPTER TWO

SQL injection

2 CHAPTER TWO

2.1 Introduction

SQL injection is an attack technique that exploits a security vulnerability occurring in the database layer of an application . Hackers use injections to obtain unauthorized access to the underlying data, structure, and DBMS. It is one of the most common web application vulnerabilities.

2.1.1 Description:

A Database is the heart of many, if not all, web-applications and is used to store information needed by the application, such as, credit card information, customer demographics, customer orders, client preferences, etc. Consequently, databases have become attractive and very lucrative targets for hackers to hack into. SQL Injections happen when a developer accepts user input that is directly placed into a SQL Statement and doesn't properly validate and filter out dangerous characters. This can allow an attacker to alter SQL statements passed to the database as parameters and enable her to not only steal data from your database, but also modify and delete it. A database is vulnerable to SQL injections when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed. SQL injection attacks are also known as SQL insertion attacks. Injection vulnerabilities, such as SQL, LDAP, HTTP header injection and OS command injection, have been ranked number one on the OWASP (Open Web Application Security Project) Top 10 Web application vulnerabilities 2010 and the top 25 Most Dangerous Software Errors 2011.

2.1.2 Common variants of SQL injection include:

SQL injection based on user input .

SQL injection based on cookies .

SQL injection based on HTTP headers .

Second-order SQL injection .

A successful SQL injection attack on a PHP application can enable an attacker to:

Steal credentials

Access databases

Alter data

Access network(s) [16]

2.2 Finding SQL Injection

SQL injection can be present in any front-end application accepting data entry from a system or user, which is then used to access a database server. In this section, we will focus on the Web environment, as this is the most common scenario, and we will therefore initially be armed with just a Web browser.

In a Web environment, the Web browser is a client acting as a front end requesting data from the user and sending it to the remote server which will

create SQL queries using the submitted data. Our main goal at this stage is to identify anomalies in the server response and determine whether they are generated by an SQL injection vulnerability. Although you will see many examples and scenarios in this chapter, we will not cover every SQL injection possibility that can be found. Think of it this way: Someone can teach you how to add two numbers, but it is not necessary (or practical) to cover every single possibility; as long as you know how to add two numbers you can apply that knowledge to every scenario involving addition. SQL injection is the same. You need to understand the hows and whys and the rest will simply be a matter of practice. We will rarely have access to the application source code, and therefore we will need to test by inference. Possessing an analytical mindset is very important in understanding and progressing an attack. You will need to be very careful in understanding server responses to gain an idea of what might be happening at the server side. Testing by inference is easier than you might think. It is all about sending requests to the server and detecting anomalies in the response. You might be thinking that finding SQL injection vulnerabilities is about sending random values to the server, but you will see that once you understand the logic and fundamentals of the attack it becomes a straightforward and exciting process .[8]

2.2.1 SQL injection based on HTTP headers:

This time, however, you are going to attempt to inject your own SQL commands appending them to the input parameter val. You can do this by appending the string OR '1'='1 to the URL:

■ `http://www.victim.com/products.php?val=100' OR '1'='1`

2.2.2 SQL injection based on user input:

SQL Injection Example

For this SQL injection example, let's use two database tables, Users and Contacts. The Users table may be as simple as having just three fields: ID, username, and password. The Contacts table has more information about the users, such as UserID, FirstName, LastName, Address1, Email, credit card number, and security code. [17]

The Users table has information used for logins like:

jsmith,P@\$w0rd

sbrown,WinterIsComing!

kcharles,Sup3rSecur3Password\$

Note: Passwords should always be hashed and salted when stored in a database and never in cleartext.

When someone wants to log in, they'll go to the login page and enter their username and password. This information is then sent to the webserver, which will construct a SQL query and send that query to the database server. An example of what that query looks like might be: Select ID from Users where username='jsmith' and password='P@\$w0rd'

The way SQL works is that it will then perform a true or false comparison for each row that the query requests. In our example, the query says to check the Users table and give back the ID value for every row where the

username is jsmith and the password is P@\$w0rd. Often, the webserver will then see what is returned by the database server and if it is a number. In our case, the webserver would receive back a 1 and let the user past the login page.

But, what if we want to get malicious with this? Because the database server performs that true-or-false check, we can trick it into believing that we have successfully authenticated. We can do this by adding an OR to the password. If we log in with x' or 1=1 as our password, that will create a new SQL query that looks like:

Select ID from Users where username='jsmith' and password='x' or 1=1
This will work for us, because while x is not jsmith's password, the database server will then check the second condition. If x isn't jsmith's password, then does 1 equal 1? It does! The ID will be sent back to the application and the user will be successfully authenticated.

This doesn't have to be a 1=1 condition. Any two equal values will work, 2=2, 4726=4726 or even a=a.

If a web page is capable of displaying data, it may also be possible to print additional data to the screen. To access the data, we can try to chain together two SQL requests. In addition to our ' or 1=1, we can add on to that a second statement like UNION SELECT LastName, credit card number, security code from Contacts. Extra clauses like this may take some extra work, but getting access to data is the ultimate goal of a SQL injection attack.

Another technique we can use for blind SQL injection, the one where no data is sent back to the screen is to inject other hints. Similar to our ‘ or 1=1 condition, we can tell the server to sleep. We could add: “ ‘ or sleep(10) ” and this will do what it seems like. It will tell the database server to take a 10-second nap and all responses will be delayed.

2.3 How to prevent SQL injection attacks

The following suggestions can help prevent an SQL injection attack from succeeding:

Don't use dynamic SQL

- Avoid placing user-provided input directly into SQL statements.
- Prefer prepared statements and parameterized queries, which are much safer.
- Stored procedures are also usually safer than dynamic SQL.
- The PDOStatement::bindParam() function is an inbuilt function in PHP that is used to bind a parameter to the specified variable name. This function bound the variables, pass their value as input, and receives the output value, if any, of their associated parameter marker.[18]

For example

```
$addUser=$conn->prepare("INSERT INTO  
  
uesr(FirstName,LastName,username,email,passowrd,type_user)  
VALUES(:FirstName,:LastName,:username,:email,:passowrd,:type_user)  
"); bool PDOStatement::bindParam ( $parameter, $variable, $data_type,  
$length, $driver_options )
```

```
$addUser->bindParam(":FirstName",$fname);
```

```
$addUser->bindParam(":LastName",$lname);
```

```
$addUser->bindParam(":passowrd",$password1);
```

```
$addUser->bindParam(":email",$email);
```

```
$addUser->bindParam(":address",$address);
```

```
$addUser->bindParam(":type_user",$groubid);
```

Sanitize user-provided inputs

- Properly escape those characters which should be escaped.
- Verify that the type of data submitted matches the type expected.

Don't leave sensitive data in plaintext

- Encrypt private/confidential data being stored in the database.
- Salt the encrypted hashes.
- This also provides another level of protection just in case an attacker successfully exfiltrates sensitive data.

Limit database permissions and privileges

- Set the capabilities of the database user to the bare minimum required.
- This will limit what an attacker can do if they manage to gain access.

Avoid displaying database errors directly to the user

- To avoid display error use function (error_reporting(0));
- To prevent the user from exploiting errors.
- Attackers can use these error messages to gain information about the database.

Use a Web Application Firewall (WAF) for web applications that access databases

- This provides protection to web-facing applications.
- It can help identify SQL injection attempts.
- Based on the setup, it can also help prevent SQL injection attempts from reaching the application (and, therefore, the database).

Use a web application security testing solution to routinely test web apps that interact with databases

- Doing so can help catch new bugs or regressions that could allow SQL injection.

Keep databases updated to the latest available patches

- This prevents attackers from exploiting known weaknesses/bugs present in older versions.

SQL injection is a popular attack method for adversaries, but by taking the proper precautions such as ensuring data is encrypted, that you protect and test your web applications, and that you're up to date with patches, you can take meaningful steps toward keeping your data secure.[17]

2.4 Cross-Site Scripting or XSS

is a type of security vulnerability where an attacker gains access to a website and executes a potentially malicious script at the client's side. This is one of the code injections attacks which can be caused by incorrectly validating user data, which usually gets inserted into the page through a web form or using a hyperlink that has been tampered. This code can be inserted via any client-side coding languages such as JavaScript, HTML, PHP, VBScript[18]

2.4.1 Preventing XSS in HTML and PHP

Following are the methods by which we can prevent XSS in our web applications –

- Using htmlspecialchars() function – The htmlspecialchars() function converts special characters to HTML entities. For a majority of web-apps, we can use this method and this is one of the most popular methods to prevent XSS. This process is also known as HTML Escaping.
 - ‘&’ (ampersand) becomes ‘&’
 - “” (double quote) becomes “”
 - ” (greater than) becomes ‘>’

```
<?php
// GET
$input = htmlspecialchars($_GET['input']);
// POST
$input = htmlspecialchars($_POST['input'])
?>
```

- strip_tags() – This function removes content between HTML tags. This function also does not filter or encode non-paired closing angular braces

```
$name=strip_tags($_GET["fname"]);
```

- **Filter Functions:** The filter function is used to filter the data coming from insecure source.

```
$name=filter_var($_GET["fname"],FILTER_SANITIZE_STRING);
```

Can be prevented xss and sql injection http headerThrough the valid data
Sanitize data and encrypt data in query string

2.5 File upload security

What are file upload vulnerabilities? File upload vulnerabilities are when a web server allows users to upload files to its filesystem without sufficiently validating things like their name, type, contents, or size. Failing to properly enforce restrictions on these could mean that even a basic image upload function can be used to upload arbitrary and potentially dangerous files instead. This could even include server-side script files that enable remote code execution. In some cases, the act of uploading the file is in itself enough to cause damage. Other attacks may involve a follow-up HTTP request for the file, typically to trigger its execution by the server.

What is the impact of file upload vulnerabilities? The impact of file upload vulnerabilities generally depends on two key factors: Which aspect of the file the website fails to validate properly, whether that be its size, type, contents, and so on. What restrictions are imposed on the file once it has been successfully uploaded. In the worst case scenario, the file's type isn't

validated properly, and the server configuration allows certain types of file (such as .php and .jsp) to be executed as code. In this case, an attacker could potentially upload a server-side code file that functions as a web shell, effectively granting them full control over the server.

If the filename isn't validated properly, this could allow an attacker to overwrite critical files simply by uploading a file with the same name. If the server is also vulnerable to directory traversal, this could mean attackers are even able to upload files to unanticipated locations.

Failing to make sure that the size of the file falls within expected thresholds could also enable a form of denial-of-service (DoS) attack, whereby the attacker fills the available disk space[19].

CHAPTER THREE

Requirements

3 CHAPTER THREE

3.1 Thses are functionality performed by the admin users

- Admin Registration: Any admin can register on website using the registration module
- Login for admin.
- Confirm from email.
- Forgot password for admin.
- Edit profile for admin .
- Change password for admin .
- Logout functionality .
- Dashboard for admin user .

Mange teachers

- Add new teacher
- View details of the teacher
- List of all teacher

Mange courses

- Add new course
- Edit the exiting course
- View details of the course
- Delete the course
- List from the course

Mange students

- Add new student
- View details of the student
- List of all student

Mange dscussion

- Send message to student
- Send message to teacher

Thses are functionality performed by the admin users

- Login for teacher
- Forgot password for teacher
- Edit profile for teacher
- Change password for teacher
- Logout functionality
- Dashboard for teacher user

3.2 Functionality performed by Teacher user:

- Teacher Registration: Any Teacher can register on website using the registration module.
- Teacher Login: This is the login form, from where Teacher can login into the system
- Teacher Add lessons: the teacher can be create new lessons.
- teacher can be upload and download and delete the lessons and also edit.

- Teacher Assignments Add: This is the Teacher Assignments Add form of the project.
- Teacher Send Message: This is the Teacher Send Message form where Teacher will be able to send message.
- Teacher Quiz Screen: This is Teacher Quiz Screen form where teacher add quiz and see quiz report.
- the teacher can change or update password and image profile
- 10 if forgot password can be change

3.3 Functionality performed by Student user:

- Student Registration: Any Student can register on website using the registration module.
- Student Login: This is the login form, from where Student can login into the system
- student can be upload and download file.
- can be insert the link
- The student can download the answers
- the students can change or update password and image profile
- if forgot password can be change
- Student Assignments Report: This is the Student Assignments Report form of the project
- Student Quiz Screen: This is Student Quiz Screen form where Student see quiz report.

In this project, students are allowed to discuss with the teacher or the doctor

3.4 Tools used in this project

- **HTML:** Page layout has been designed in HTML
- **CSS:** CSS has been used for all the designing part
- **JavaScript:** All the validation task and animations has been developed by JavaScript
- **PHP:** All the business and frontend logic has been implemented in PHP with security
- **MySQL:** MySQL database has been used as database for the project
- **XAMPP:** Project will be run over the Apache server

3.5 Database

E-learning represents an innovative shift in the field of learning, providing rapid access to specific knowledge and information. It offers online instruction that can be delivered anytime and anywhere through a wide range of electronic learning solutions such as webbased courseware, online discussion groups, live virtual classes, video and audio streaming, web chat, online simulations, and virtual mentoring.

In this course we will use this database in my project

Name database :my_project

1-User_form represent admin and teacher and student can be distinguished by Type_user.

2-courses .

3- lessons.

4- Assignment .

5-Answers .

6- degree.

7-discussions.

8- links.

9.Quizzes.

1-FIRST of the ERD process is to add attributes to our entities. user Entity has the following attributes:

user_form contain (

ID , image ,password, email ,

user_type / note :identify the admin and uesr',

name

SECURITY_CODE / note Account Activation cod

ACTIVTION / NOTE :Active account

)

2- Lessons Entity has the following attributes:

lessons (

lesson_id , chapter ,title , course_id ,FILE ,Subject , teacher_id
,time_insert

)

3 - Lessons details Entity has the following attributes:

less_link (

id_link , title , chose , ttext , teacher_id)

4 -Quiz Entity has the following attributes:

Quiz (

ID , name_quiz , file , teacher_id , course_id , time_insert)

5 -Quiz multi choice Entity has the following attributes:

quiz_m (

id , que , option_1 , option_2 , option_3 , option_4

ans , userans , course_id , teacher_id)

6 -Discussion Entity has the following attributes:

discussion(

ID , text , course_i , section , teacher_id , student_id , time_insert

)

7 -assignment Entity has the following attributes:

assignment(

ID , name_ASSINMINT , file , POSITION , FILE_TYPE , teacher_id ,
course_id

time_insert)

8- courses Entity has the following attributes:

courses (

id , name , category , duration , date , image

9- degree Entity has the following attributes:

score` (

id_score , scor_quize , scor_number_mulit , details_id , scor_mid ,
scor_finally ,Degree_of_pursuit , score_assignment

10- degree Entity has the following attributes:

details_q (

id_det , number_of_que , score)

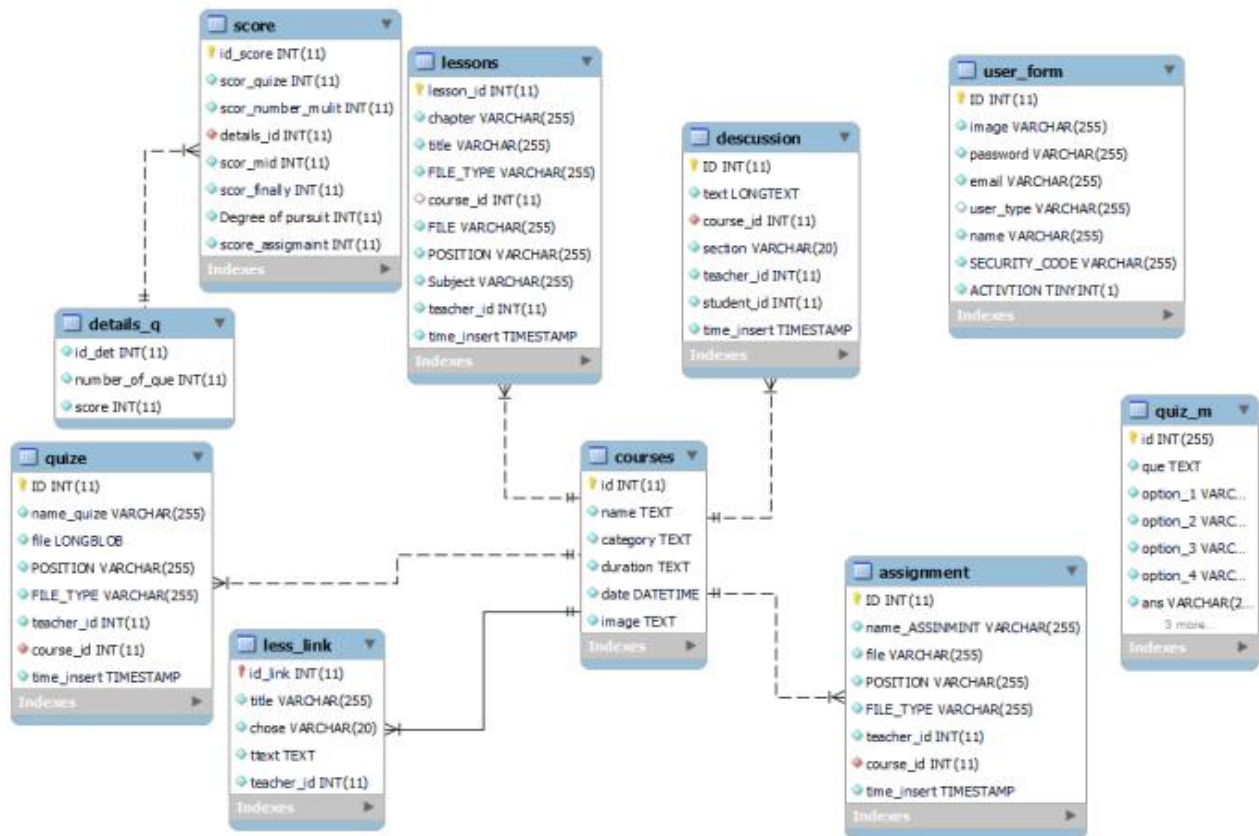



Figure (1) Database E-learning System


 We are The Lotus Team

Please Register Now

User:

Submit

I Already have an account? [log in](#)

Figure (2) Register in page

Register In Page:

Registration page on this page a student, teacher, supervisor (or administrator)can

1- the intended full name must be written.

2- Write an email with it.

3- Write the password and the password must be confirmed by repeating it

4- the type of user must be determined by choosing the user, either if he is a teacher, student or administrator, through this property it can be determined if a student goes to the student page and if a teacher goes to the teacher page and so on .

Note the official e-mail must be confirmed, the page is not accessed until after the e-mail is confirmed, otherwise the page is not accessed Login is done through the login page

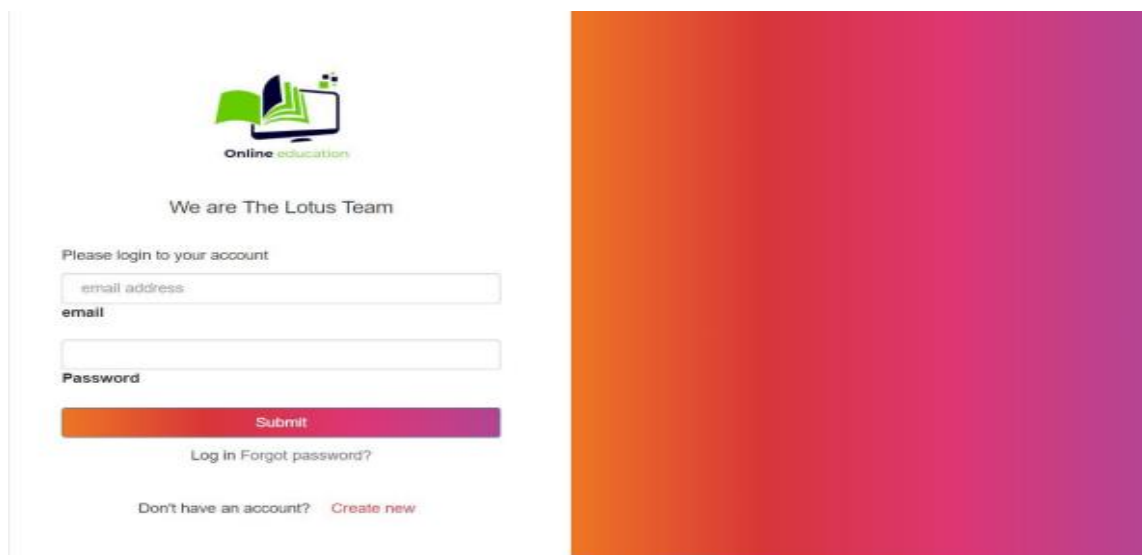
The image shows a login page for 'Online education'. At the top, there is a logo with a green book and a computer monitor, with the text 'Online education' below it. Below the logo, it says 'We are The Lotus Team'. The main heading is 'Please login to your account'. There are two input fields: the first is labeled 'email address' and the second is labeled 'email'. Below these is a password field labeled 'Password'. A red 'Submit' button is at the bottom of the form. Below the button, there is a link 'Log in Forgot password?'. At the very bottom, there is a link 'Don't have an account? Create new'.

Figure (3) Log in Page

Through the login page, you can access the main page, depending on the type of user.

The login process is described in the figure below

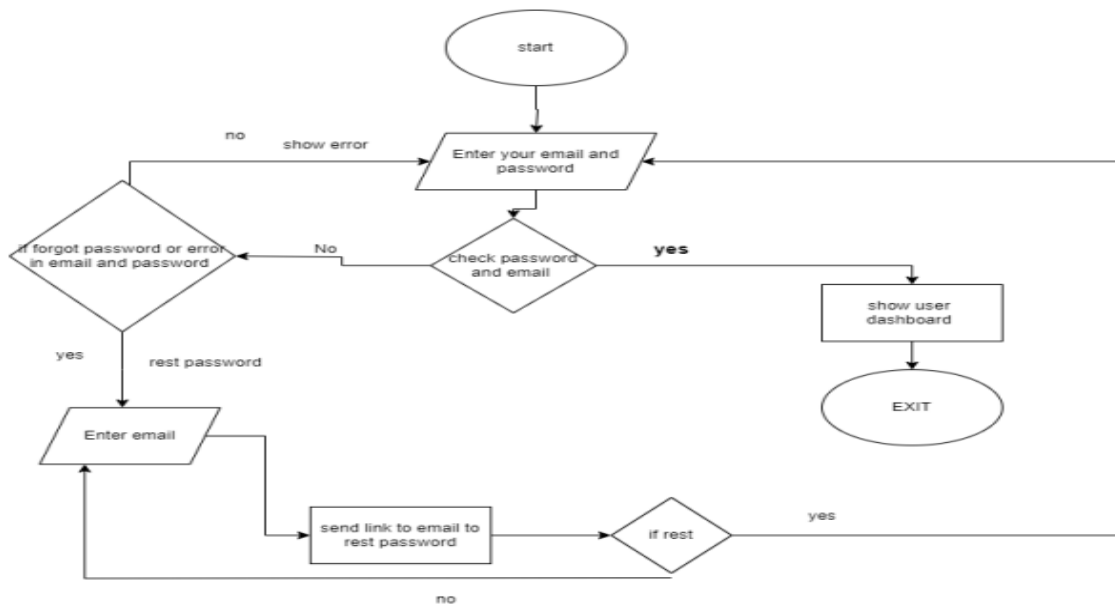


Figure (4) Flow Chart Log in

Upload file:

Upload a file in this part we will discuss uploading a file after logging in, Go to the lessons or assignments section, if the login is not logged in, the file is not uploaded, then the file is selected and the required information is included, if it is as required, the Upload is done otherwise, try again, the file is saved in the folder on the server, and only the name is saved in the database or path

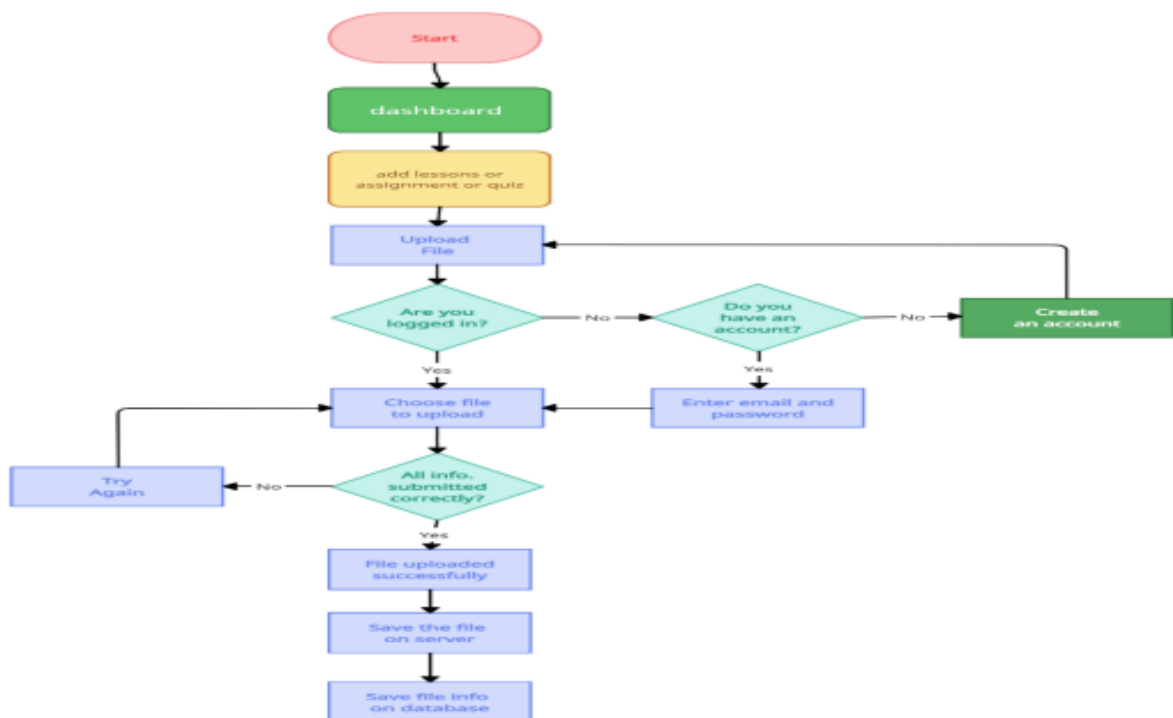


Figure (5) sample design file upload system

CHAPTER FOUR

Conclusion&FutureWork

4 CHAPTER FOUR

4.1 CONCLUSION

In conclusion, the implementation of proper security measures in E-learning systems is crucial to prevent potential SQL injection attacks. It is important to identify and address vulnerabilities in the system by conducting regular security audits and ensuring that all software components are up-to-date with the latest security patches.

In addition, user input validation and sanitization must be implemented to prevent malicious SQL statements from being executed. This can be achieved through the use of prepared statements, parameterized queries, and input validation techniques.

Furthermore, education and training on secure coding practices should be provided to developers and system administrators to ensure that they have the necessary knowledge and skills to implement effective security measures in E-learning systems.[9]

Overall, the implementation of robust security measures and continuous monitoring can greatly reduce the risk of SQL injection attacks in E-learning systems, thereby ensuring the safety and security of sensitive data and user information.

4.2 FUTURE WORK

the we suggest several directions for further research and improvements to the current system. Some of these potential areas for future work include:

Improving the security of the system by implementing more robust security measures beyond SQL injection prevention. This could include incorporating encryption, user authentication, and authorization mechanisms, as well as conducting regular security audits and assessments.

Enhancing the user experience and functionality of the system by adding new features and tools. For example, the authors suggest integrating social media and gamification elements to make the learning experience more interactive and engaging.

Conducting more extensive testing and evaluation of the system to ensure its effectiveness and usability. This could involve conducting user surveys, gathering feedback from stakeholders, and conducting A/B testing to determine the most effective design and functionality.

Exploring the potential for integrating machine learning and artificial intelligence technologies to improve the system's ability to personalize learning and adapt to individual learner needs.

Overall, the "Future Work" section highlights the ongoing need for continuous improvement and innovation in E-learning systems, as technology and user needs continue to evolve.[10]

I. References

- [1] Hodges, C., Moore, S., Lockee, B., Trust, T., & Bond, A. (2020). The difference between emergency remote teaching and online learning. *EDUCAUSE Review*, 27.
- [2] Li, N., & Li, F. (2020). COVID-19 and E-learning: A Review of Effects on the Teaching and Learning Process. *Eurasia Journal of Mathematics, Science and Technology Education*, 16(7), em1861.
- [3] OWASP (Open Web Application Security Project) is a community-driven organization that provides information and resources related to web application security, including SQL injection attacks. Their website contains useful guides and resources on preventing and mitigating SQL injection attacks in web applications.
- [4] W3Schools. SQL Injection. Retrieved from https://www.w3schools.com/sql/sql_injection.asp This web page provides an overview of SQL injection attacks and examples of how they can be prevented using various techniques.
- [5] "SQL Injection Attacks and Defense" by Justin Clarke: This book provides a comprehensive overview of SQL injection attacks and how to defend against them, including best practices for securing E-learning systems.
- [6] "Secure Coding in C and C++" by Robert C. Seacord: This book provides guidance on secure coding practices for developers, including techniques for preventing SQL injection attacks in web applications.
- [7] "OWASP Top Ten Project" by the Open Web Application Security Project (OWASP) - This project provides a list of the top ten most critical web application security risks, including SQL injection attacks. It also provides guidance on how to prevent and mitigate these risks. The OWASP website (<https://owasp.org/>) contains a wealth of resources on web application security.

- [8] J. Clarke, SQL Injection Attack And Defense, United States of America: Syngress Publishing, Inc., 2009.
- [9] Kim, D. H., et al. "Analysis of SQL Injection Attack and Defense Techniques." Proceedings of the 2017 International Conference on Platform Technology and Service, 2017, pp. 1-6
- [10] A. Deokar and D. Chakraborty, "Secure E-learning System Using SQL Injection Prevention Techniques," International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no. 9, pp. 100-110, 2016.
- [11] OWASP (Open Web Application Security Project): <https://owasp.org/>
- [12] W3Schools. SQL Injection. Retrieved from : https://www.w3schools.com/sql/sql_injection.asp
- [13] SQL Injection Attacks and Defense, Second Edition by Justin Clarke: <https://www.amazon.com/Injection-Attacks-Defense-Second-Justin/dp/1597499633>
- [14] "OWASP Top 10 - 2017: A1 Injection" by OWASP, https://owasp.org/Top10/A1_2017-Injection.html
- [15] E-learning by Wikipedia: <https://en.wikipedia.org/wiki/E-learning>
- [16] <https://brightsec.com/blog/php-sql-injection/>
- [17] <https://www.rapid7.com/fundamentals/sql-injection-attacks/>
- [18] <https://www.geeksforgeeks.org/how-to-prevent-xss-with-html-php/>
- [19] <https://portswigger.net/web-security/file-upload>

الملخص

يقدم ملخص هذه المشروع لمحة عامة عن مشروع البحث. يبدأ بإدخال مشكلة هجمات حقن SQL على أنظمة التعلم الإلكتروني. ثم يصف الهدف من مشروع البحث ، وهو تطوير نظام تعليم إلكتروني آمن يمكنه مقاومة هجمات حقن SQL.

يصف الملخص بعد ذلك بإيجاز المنهجية المستخدمة في المشروع ، والتي تتضمن تصميم وتنفيذ نظام تعليم إلكتروني آمن ثم اختباره بحثاً عن نقاط الضعف. كما يسلط الضوء على أهمية هذا المشروع البحثي ، حيث تزداد شعبية أنظمة التعلم الإلكتروني ويصبح تهديد هجمات حقن SQL مصدر قلق كبير.

أخيراً ، يشير الملخص إلى النتائج المتوقعة للمشروع ، والتي تشمل تحديد ومعالجة الثغرات الأمنية في نظام التعلم الإلكتروني وإظهار أمانه ضد هجمات حقن SQL. الغرض العام من مشروع البحث هو تعزيز أمان أنظمة التعلم الإلكتروني وتوفير نموذج لتطوير تطبيقات الويب الآمنة التي يمكنها مقاومة هجمات حقن SQL.



وزارة التعليم العالي والبحث العلمي

جامعة التكنولوجيا



قسم هندسة الحاسوب - العام الدراسي 2022-2023

تصميم وتنفيذ نظام تعليم إلكتروني آمن باستخدام لغة

حقن SQL

مشروع التخرج

مقدم إلى قسم هندسة الحاسبات تنفيذاً جزئياً لدرجة البكالوريوس. شهادة في هندسة المعلومات

بواسطة

مصطفى بشير حاتم

باسم حسين عبد الامير

يشرف عليها

مساعدة زينب محمود

2022-2023