www.lmsportals.com

LMS PORTALS

POWERFUL LEARNING MANAGEMENT SYSTEM PORTALS

Cloud-Based, Multi-Tenant LMS with Centralized Administration

TABLE OF **CONTENTS**

1 SYSTEM SECURITY

LMS Portals leverages XSS to prevent SQL Injections by use of prepared statements, whitelist input validation, escaping all user-supplied inputs, etc.

Because we have implemented XSS, whenever you output dynamic data, the data is cleaned with the htmlspecialchars function.

Strong Cryptography:

LMS Portals uses the latest hashing algorithm sha512 with the longest encryption key storing at a safe place (instead of MD5 or SHA1).

This eliminates the possibility of easily guessed passwords. We have implemented password rules with validation of (8- characters min, lower case, 1 uppercase, 1 number, OR 1 special character).

All sensitive data resting in the database is encrypted using a proprietary encryption/decryption function.

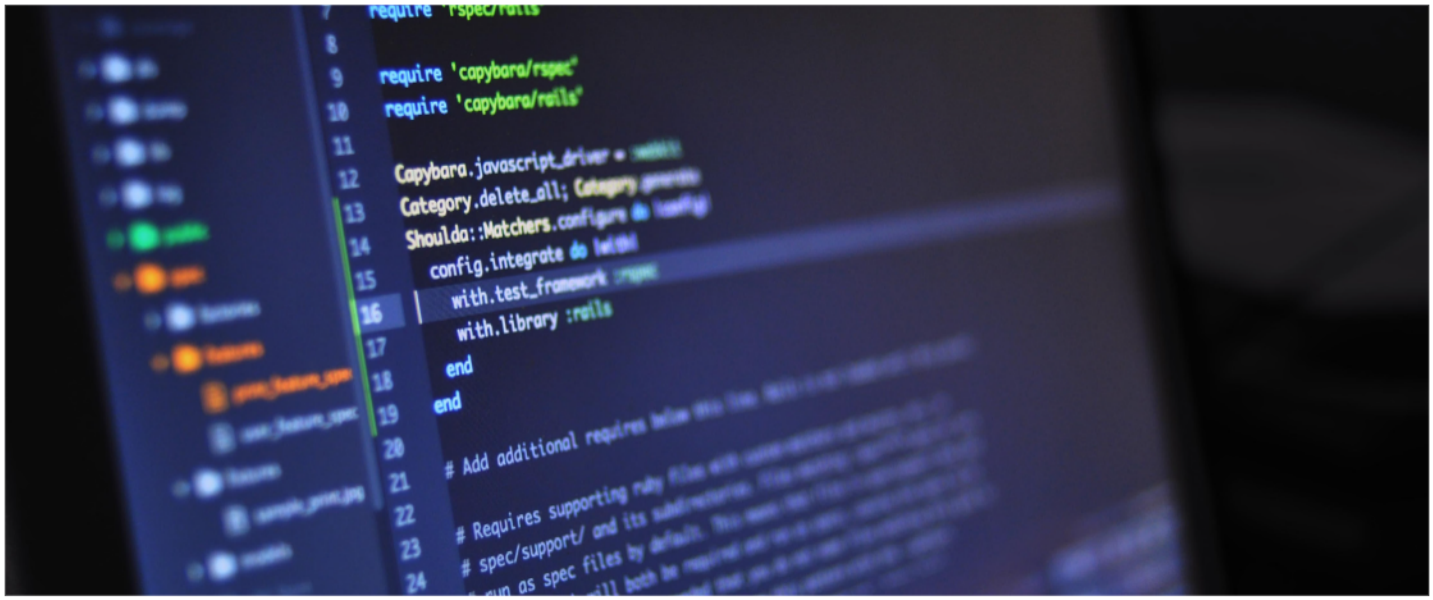
Directory Exposure:

1. We have disabled subdirectories and file listings for any directory of your project.

2. To disable directory listing, we have added .htaccess file to each directory wherever storing uploaded files.

Use of the "Root" User:

We have created a MySQL user (not root!) and utilize that user instead of the root login. We do not offer the same access to this user as root.



Server Hardening:

We have installed fail2ban and configured it to block access attempts. We have configured the value of the max attempt to 3.

We have implemented SSH to require a security key to connect by configuring a private security key and passphrase so it secures the SSH connection by a security key.

Overcome Potentially Leaked Password:

We have removed sensitive information, such as passwords, encryption keys, etc. from code or commented code lines. We have saved sensitive information in encrypted format in database and configuration files.

Overcome missing Encryption:

We store credentials such as API keys, client ids, passwords, etc. into the database with encryption.

Strong Password Controls:

We have implemented strong password validation so that passwords cannot be easily guessed. We have implemented validation to ensure passwords contain at least one uppercase, lowercase, one number, or one special character.

No Improperly Stored Sensitive Data:

Storing sensitive information as plain text can increase the chance of hacking and data theft by cyber-attacks. To overcome improperly stored sensitive data, we have stored sensitive information in the database using strong encryption.

Correct security configuration:

We have implemented security configurations that eliminate dead links from projects and databases.

Secured Admin application:

We have restricted direct access to the PHPMyadmin database. If an unauthorized person accesses the URL directly, they will be redirected to the forbidden page. We have allowed access to the database only to specific IP addresses we want to allow.

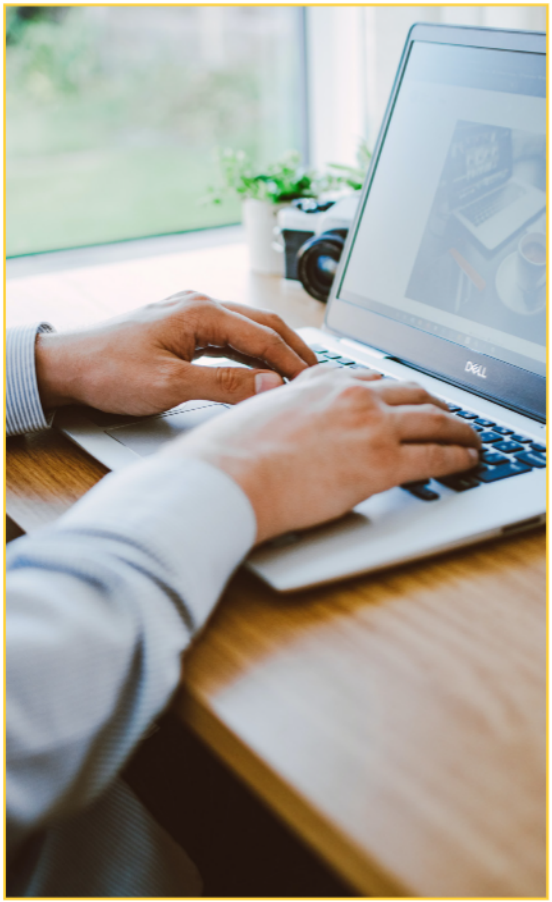
2 DATABASE SECURITY

2.1 DEDICATED DATABASE

LMS Portals creates a dedicated database for each new portal.

A dedicated database for each portal is preferable over managing client data in a single database. The benefits of a dedicated database include:

- Easy debug of any issues in real-time in a live environment
- Maintain high speed and enhanced performance
- Decreased chance of hack of the whole system database



2.2 DEDICATED DATABASE CREDENTIALS

We create all databases with the same credentials. We store database passwords in encrypted format.

2.3 INDEXING

We have implemented indexing for columns on tables of the dedicated database. This improves the speed of query execution which includes where clauses and join queries.

2.4 CASCADING / CONSTRAINTS

We have implemented cascading using foreign key creation. Cascading is a relation between two or more tables. Through cascading, when we delete a specific record we remove child records related to specific records in other tables.

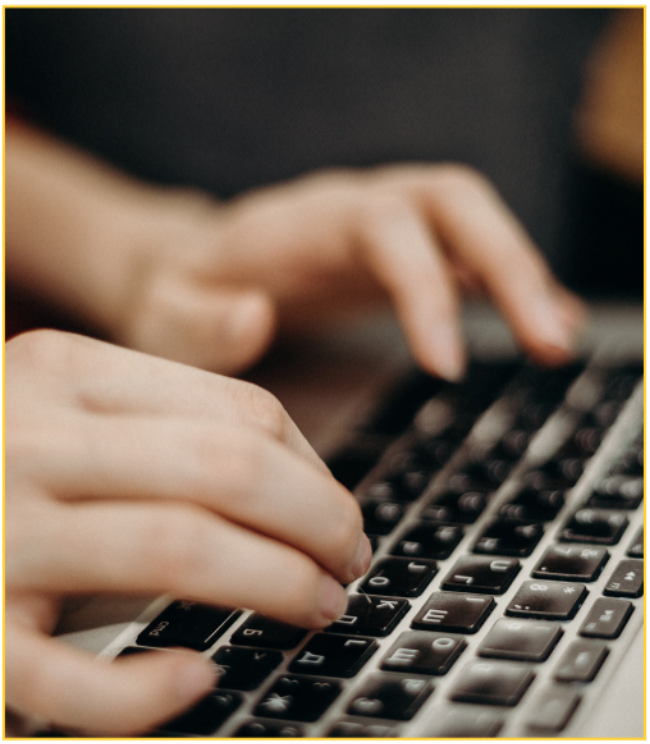
Cascading also provides the benefit of restricting to deletion of child records directly. This minimizes any opportunity to scrape records and errors on the front end.

2.5 MINIMIZE VALUE OF DATABASE

We do not store any confidential information that is not required.
We actively manage the data so we can delete any information that we do not need from the database. Data that must be retained for compliance or other purposes can be moved to more secure storage (eg, offline).

2.6 STRONG PASSWORD

- We have implemented strong and unpredictable passwords.
- We have implemented strong password validation with password must be minimum of 8 characters and use a combination of letters, numbers, cases, and symbols.



2.7 BACK-UPS

A data backup, as part of the database security protocol, makes a copy of your data and stores it on a separate system. This backup allows recovering lost data that may result from hardware failures, data corruption, theft, hacking, or natural disasters. We have implemented a process for taking a backup of databases every day.

2.8 SQL – INJECTION

We leverage the Codeigniter framework so SQL injection does not occur because the framework uses query builders for SQL queries.

2.9 DATABASE PHYSICAL SECURITY

Access controls of the database are in place to prevent unauthorized access. No one has direct access to dedicated database credentials.