

Developer Toolbox

- [XAMPP Backup and Recovery](#)
 - [Backing Up a XAMPP Project](#)
 - [Restoring a XAMPP Project from BackupNew Page](#)
 - [Moving a XAMPP Project to Another Computer](#)
- [Helper Tools](#)
 - [HTML & CSS Fixer](#)
 - [PHP Password Hasher](#)

XAMPP Backup and Recovery

Backing Up a XAMPP Project

A XAMPP project usually consists of two parts:

1. The website files stored in the `htdocs` folder.
2. The MySQL database storing the application's data.

To create a complete backup, both parts must be saved.

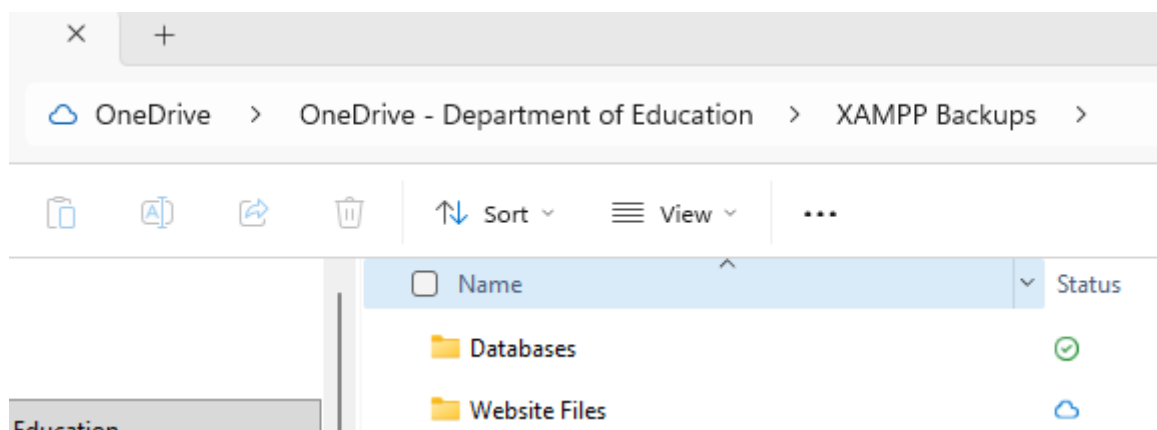
Create a Backup Folder

Create a folder somewhere safe, such as:

OneDrive - Department of Education
└─ XAMPP Backups

Inside it create:

XAMPP Backups
├─ Website Files
└─ Databases



Back Up the Website Files

Open the XAMPP installation folder.

Example:

```
C:\xampp\htdocs\
```

Locate your project folder.

Example:

```
C:\xampp\htdocs\security_dashboard
```

Copy the entire project folder into:

```
XAMPP Backups
└─ Website Files
```

Your backup should now contain:

```
Website Files
└─ security_dashboard
```

This backup includes:

- HTML files
- PHP files
- CSS files
- JavaScript files
- Images
- JSON files

Open phpMyAdmin

Open phpMyAdmin in your browser.

Example:

<http://localhost/phpmyadmin/>

Select the database used by your project.

Example:

project_db

Export the Database

Click:

Export

Choose:

Method: Quick

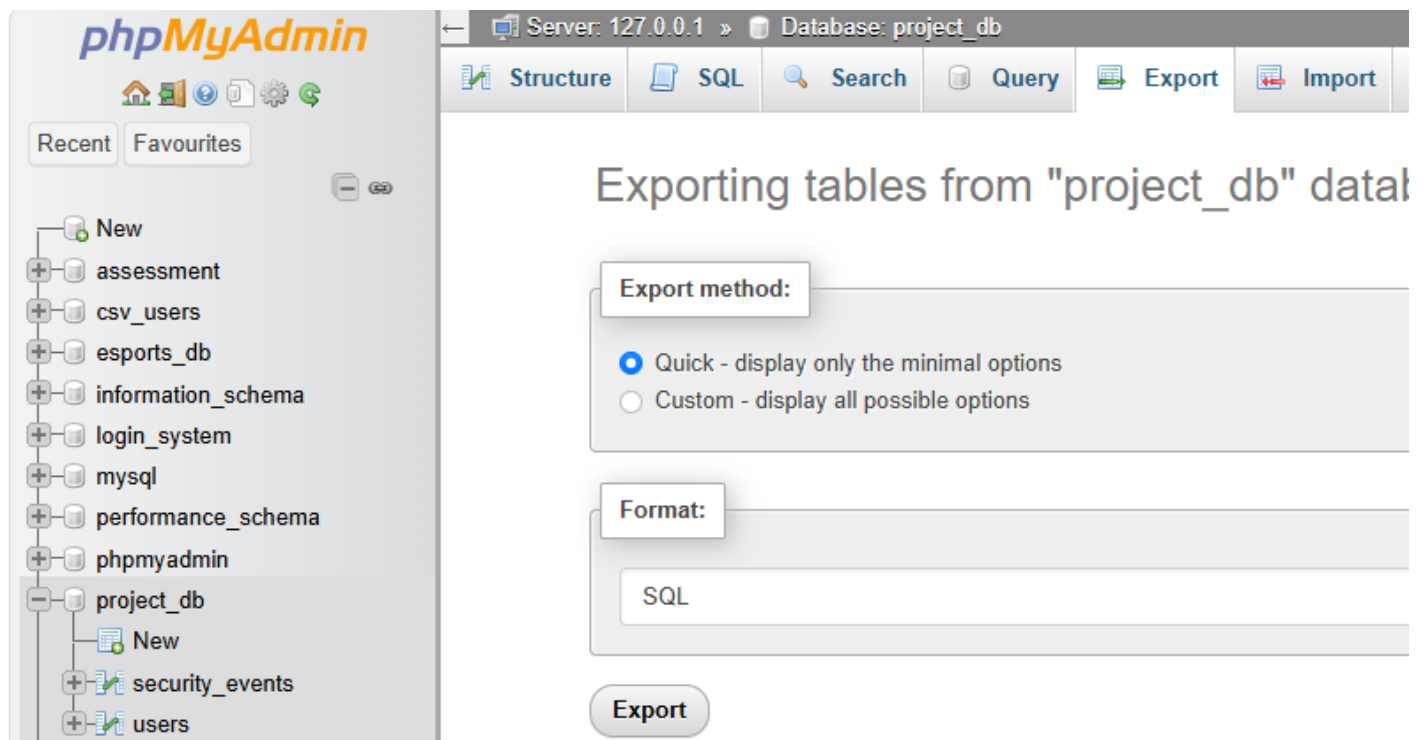
Format: SQL

Click:

Export

A file similar to the following will be downloaded:

project_db.sql



Save the Database Backup

Move the downloaded SQL file into:

```
XAMPP Backups
└─ Databases
```

Your backup folder should now contain:

```
XAMPP Backups
├─ Website Files
│   └─ security_dashboard
│
└─ Databases
    └─ project_db.sql
```

Test the Backup

Before making major changes to a project, check that:

- Website files have been copied successfully.
- The SQL file exists.
- The SQL file is not empty.

A backup is only useful if it can be restored.

Complete Backup Checklist

Website Files

- ✓ Project folder copied
- ✓ Images copied
- ✓ CSS copied
- ✓ JavaScript copied

✓ PHP files copied

Database

- ✓ Database selected
- ✓ Export completed
- ✓ SQL file saved

Backup Structure Example

```
XAMPP Backups
├─ Website Files
│   └─ security_dashboard
│       ├── index.php
│       ├── dashboard.php
│       ├── css
│       ├── js
│       └─ images
└─ Databases
    └─ project_db.sql
```

You now have a complete backup of your XAMPP project.

Next tutorial: **Restoring a XAMPP Project from Backup.**

Restoring a XAMPP Project from BackupNew Page

If a project is accidentally deleted, corrupted, moved to another computer, or you need to continue working from a backup, both the website files and database must be restored.

This tutorial demonstrates how to restore a complete XAMPP project from a backup.

Backup Requirements

Before restoring, you should have:

```
XAMPP Backups
├─ Website Files
│   └─ security_dashboard
│
└─ Databases
    └─ project_db.sql
```

The project folder contains your website files and the SQL file contains your database backup.

Restore the Website Files

Open your backup folder.

Example:

```
XAMPP Backups
└─ Website Files
    └─ security_dashboard
```


Copy the entire project folder.

Copy the Files into htdocs

Open:

C:\xampp\htdocs\

Paste the project folder into the `htdocs` directory.

Example:

C:\xampp\htdocs\security_dashboard

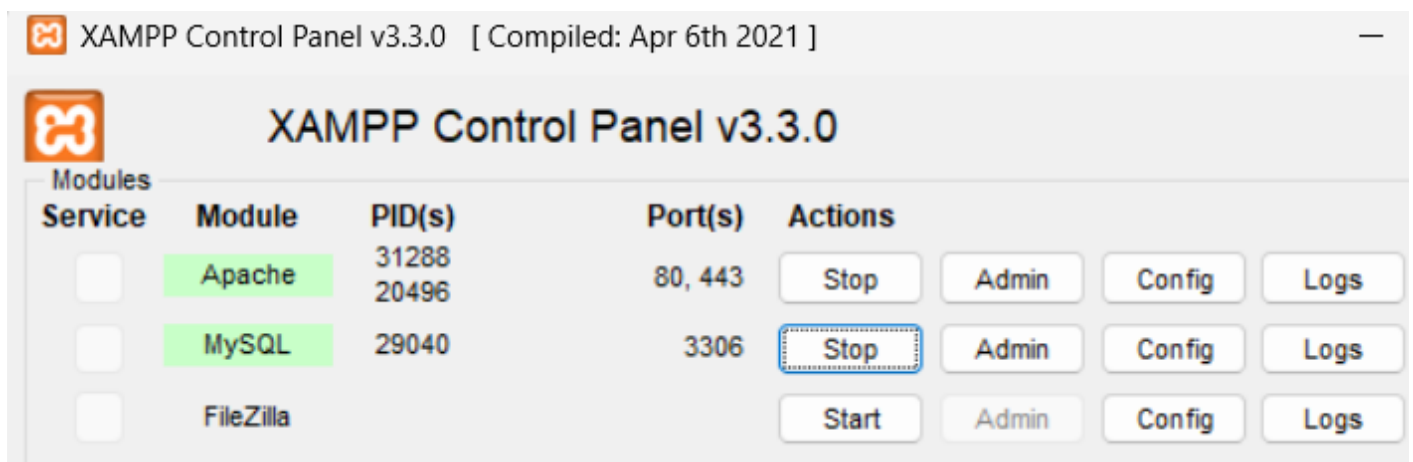
Your project should now appear inside the XAMPP web server folder.

Start XAMPP

Open the XAMPP Control Panel.

Start:

Apache
MySQL



Open phpMyAdmin

Open:

<http://localhost/phpmyadmin/>

You should now see the phpMyAdmin home page.

Create the Database

Click:

New

Create a database using the original database name.

Example:

project_db

Click:

Create

Import the Database Backup

Select the newly created database.

Click:

Import

Choose:

project_db.sql

Click:

Import

phpMyAdmin will restore all tables and records from the backup file.

phpMyAdmin

Server: 127.0.0.1 » Database: project_db

Structure SQL Search Query Export Import Operations

Recent Favourites

New

- assessment
- csv_users
- esports_db
- information_schema
- login_system
- mysql
- performance_schema
- phpmyadmin
- project_db
 - New
 - security_events
 - users
- school_library
- test
- timetables

Importing into the database "project_db"

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 40MiB)

Choose File project_db.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.
This might be a good way to import large files, however it can break transactions.

Skip this number of queries (for SQL) starting from the first one:

0

Other options

☒ Enable foreign key checks

Format

SQL

Format-specific options:

SQL compatibility mode:

NONE

☒ Do not use AUTO_INCREMENT for zero values

Import

Verify the Database

Select the restored database.

You should see the original tables.

Example:

```
project_db
├─ users
└─ security_events
```

Click on a table and verify that records are present.

Test the Website

Open your project in a browser.

Example:

http://localhost/security_dashboard/

Check that:

- Pages load correctly.
- Images display correctly.
- Login systems work.
- Database data appears correctly.

Common Problems

Database Connection Error

Check:

```
Database name
Username
Password
```

inside your PHP connection file.

Example:

```
$conn = new mysqli(
    "localhost",
    "root",
    "",
    "project_db"
);
```

The database name must match the restored database.

Table Not Found

Example error:

```
Table 'project_db.users' doesn't exist
```

Solution:

- Confirm the SQL file imported successfully.
 - Verify the table exists in phpMyAdmin.
 - Re-import the database if required.
-

Page Not Found

Example:

```
404 Not Found
```

Solution:

- Verify the project folder is inside:

```
C:\xampp\htdocs\
```

- Check the URL is correct.
-

Restoration Checklist

Website Files

- ✓ Project folder copied to htdocs
- ✓ Images restored
- ✓ CSS restored
- ✓ JavaScript restored
- ✓ PHP files restored

Database

- ✓ Database created
- ✓ SQL file imported
- ✓ Tables restored
- ✓ Records restored

Testing

- ✓ Website loads
- ✓ Database connection works
- ✓ Login system works
- ✓ Data displays correctly

Complete Restoration Process

```
Backup Folder
    ↓
Copy Project Files
    ↓
Paste into htdocs
    ↓
Create Database
    ↓
Import SQL File
    ↓
```

You have successfully restored a XAMPP project from a backup.

Next tutorial: **Moving a XAMPP Project to Another Computer.**

Moving a XAMPP Project to Another Computer

One of the advantages of using XAMPP is that projects can easily be moved between computers.

This is useful when:

- Working between school and home.
- Upgrading to a new computer.
- Sharing a project with a teacher.
- Creating a backup before reinstalling Windows.

A complete project transfer requires:

1. Website files.
2. Database files.

On the Original Computer

Before moving the project, create a backup.

You should have:

```
XAMPP Backups
├─ Website Files
│   └─ security_dashboard
│
└─ Databases
    └─ project_db.sql
```

If you have not already created a backup, refer to:

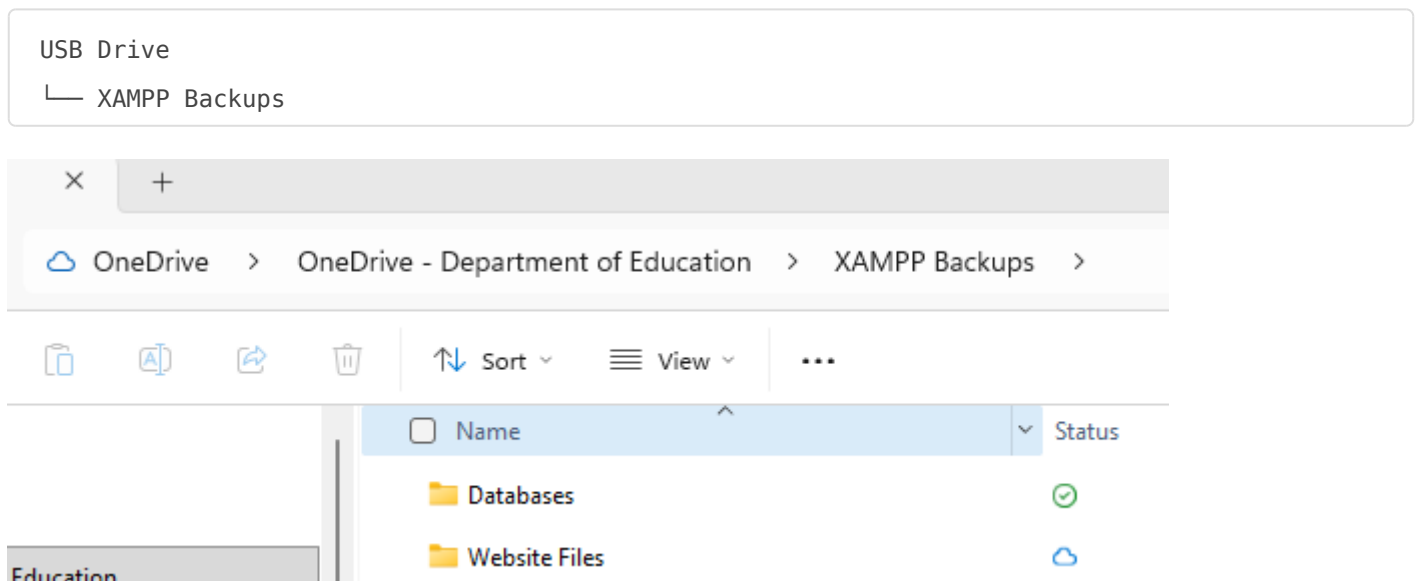
Backing Up a XAMPP Project

Copy the Backup Files

Copy the backup folder to:

- USB drive
- External hard drive
- OneDrive
- Google Drive

Example:



Install XAMPP on the New Computer

Download XAMPP:

<https://www.apachefriends.org/>

Install XAMPP using the default settings.

The installation folder will usually be:

C:\xampp\

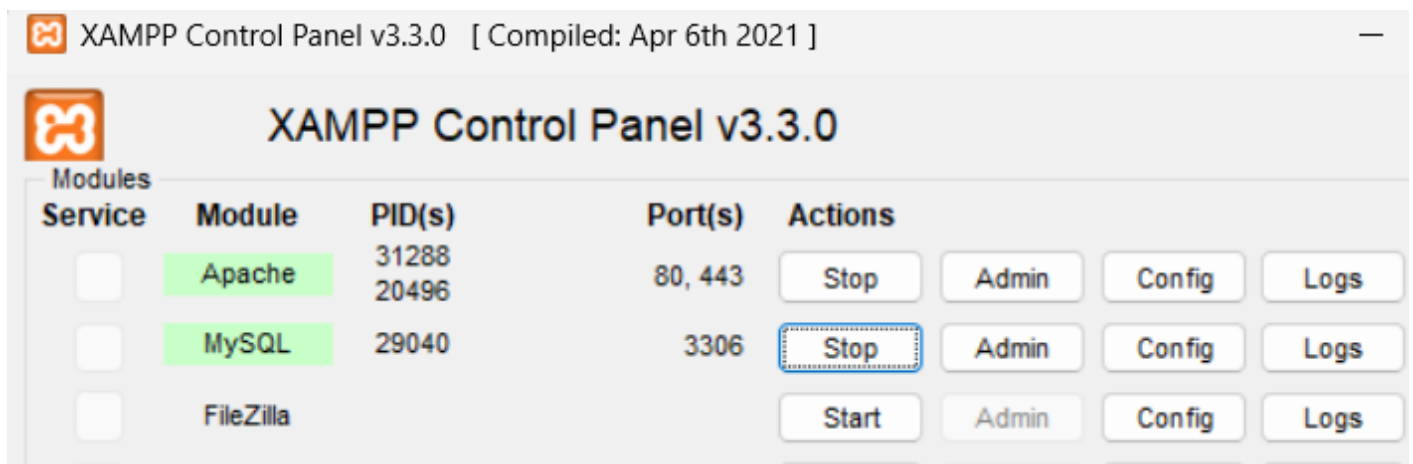
Start XAMPP

Open the XAMPP Control Panel.

Start:

Apache
MySQL

Both services should show a green status.



Copy the Website Files

Open:

C:\xampp\htdocs\

Copy the project folder from your backup.

Example:

security_dashboard

Paste it into:

C:\xampp\htdocs\

Your folder structure should now look like:

C:\xampp\htdocs\
└─ security_dashboard

Create the Database

Open phpMyAdmin:

<http://localhost/phpmyadmin/>

Click:

New

Create a database with the same name as the original project.

Example:

project_db

Click:

Create

Import the Database Backup

Select:

project_db

Click:

Import

Choose:

project_db.sql

Click:

Import

Wait for the import to complete.

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of databases: assessment, csv_users, esports_db, information_schema, login_system, mysql, performance_schema, phpmyadmin, project_db (selected), school_library, test, and timetables. The 'project_db' database is expanded, showing 'New', 'security_events', and 'users' tables. The main area displays the 'Import' tab for the 'project_db' database. The title is 'Importing into the database "project_db"'. The 'File to import:' section has a text input for the file name, currently 'project_db.sql', and a 'Choose File' button. Below this is a checkbox for 'Allow the interruption of an import...' which is checked. The 'Partial import:' section has a text input for 'Skip this number of queries...' with the value '0'. The 'Other options' section has a checkbox for 'Enable foreign key checks' which is checked. The 'Format' section has a dropdown menu set to 'SQL'. The 'Format-specific options:' section has a dropdown for 'SQL compatibility mode' set to 'NONE' and a checkbox for 'Do not use AUTO_INCREMENT for zero values' which is checked. At the bottom is an 'Import' button.

Server: 127.0.0.1 » Database: project_db

Structure SQL Search Query Export Import Operations

Importing into the database "project_db"

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 40MiB)

Choose File project_db.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit
This might be a good way to import large files, however it can break transactions.

Skip this number of queries (for SQL) starting from the first one:

0

Other options

☒ Enable foreign key checks

Format

SQL

Format-specific options:

SQL compatibility mode:

NONE

☒ Do not use AUTO_INCREMENT for zero values

Import

Verify the Tables

After importing, you should see the project tables.

Example:

```
project_db
├─ users
├─ security_events
└─ products
```

Your tables may differ depending on the project.

Check the Database Connection

Open the database connection file used by the project.

Example:

```
$conn = new mysqli(
    "localhost",
    "root",
    "",
    "project_db"
);
```

Verify:

- Database name is correct.
- Username is correct.
- Password is correct.

The default XAMPP configuration uses:

```
Server: localhost
Username: root
Password: (blank)
```

Test the Website

Open:

```
http://localhost/security_dashboard/
```

Check:

- Pages load correctly.
- Images display correctly.
- Database content appears.
- Login systems work.
- Forms submit correctly.

Common Problems

Database Not Found

Example:

```
Unknown database 'project_db'
```

Solution:

- Create the database.
 - Import the SQL backup.
-

Table Not Found

Example:

```
Table 'project_db.users' doesn't exist
```

Solution:

- Verify the SQL file imported successfully.
 - Check the table exists in phpMyAdmin.
-

Access Denied

Example:

```
Access denied for user 'root'
```

Solution:

- Check the database username and password.
 - Verify MySQL is running.
-

Page Not Found

Example:

```
404 Not Found
```

Solution:

- Confirm the project folder is inside:

```
C:\xampp\htdocs\
```

- Verify the URL matches the folder name.
-

Transfer Checklist

Website Files

- ✓ Project folder copied
- ✓ Files restored to htdocs
- ✓ Images restored
- ✓ CSS restored
- ✓ JavaScript restored
- ✓ PHP files restored

Database

- ✓ Database created
- ✓ SQL file imported
- ✓ Tables restored
- ✓ Data restored

Testing

- ✓ Website loads
- ✓ Database connection works
- ✓ Forms work
- ✓ Login works
- ✓ Data displays correctly

Complete Process

```
Original Computer
    ↓
Create Backup
    ↓
Copy to USB / Cloud Storage
    ↓
Install XAMPP
    ↓
Copy Website Files
    ↓
Create Database
    ↓
Import SQL File
    ↓
Test Website
```

You have successfully moved a XAMPP project to another computer.

Helper Tools

HTML & CSS Fixer

The HTML & CSS Fixer helps identify common mistakes in HTML and CSS code.

The tool highlights problems and suggests fixes using comments so you can correct the code yourself.

The Tool Can Help Find

- Missing closing tags
 - Incorrect nesting
 - Missing quotation marks
 - Missing semicolons
 - Missing curly braces
 - Common HTML and CSS syntax errors
-

Accessing the Tool

Open:

<https://tools.napper.au/html-fixer/>

This tool requires an **access key**.

If prompted, please ask your teacher for the current access key.

Remember

The HTML & CSS Fixer is a debugging tool.

It helps you find problems, but you are still responsible for understanding and applying the fixes yourself.

Open Tool: <https://tools.napper.au/html-fixer/>

PHP Password Hasher

The PHP Password Hasher generates secure password hashes for use in PHP login systems.

This tool is useful when:

- Adding users directly in phpMyAdmin.
 - Resetting passwords during development.
 - Testing login systems.
 - Verifying that a password matches a stored hash.
-

Using the Hash in phpMyAdmin

Example:

```
INSERT INTO users (username, password)
VALUES (
    'admin',
    '$2y$10$z2QfYt0B0e9D9I5Yj7M4VuqhZgN3bU5QeK2Yl2sL5vLJ6m6r4Q9u0 '
);
```

Accessing the Tool

[Open PHP Password Hasher](#)

Remember

Passwords should **never** be stored as plain text in a database.

Always use:

```
password_hash()
```

to create passwords and

```
password_verify()
```

to check passwords during login.