

Developer Toolbox

Practical tools and guides for setting up, checking, backing up and recovering local web-development projects.

- [XAMPP Backup and Recovery](#)
 - [Backing Up a XAMPP Project](#)
 - [Restoring a XAMPP Project from Backup](#)
 - [Moving a XAMPP Project to Another Computer](#)
- [Helper Tools](#)
 - [HTML & CSS Fixer](#)
 - [PHP Password Hasher](#)
- [XAMPP Setup and Projects](#)
 - [Installing XAMPP on Windows](#)
 - [Starting and Testing Apache and MySQL](#)
 - [Creating Your First XAMPP PHP Project](#)
 - [Creating a MySQL Database with phpMyAdmin](#)
 - [Checking a Project Before Development](#)

XAMPP Backup and Recovery

Back up, restore and transfer both XAMPP project files and MySQL databases safely.

Backing Up a XAMPP Project

A working XAMPP project normally has two separate parts: website files inside `htdocs` and data stored in MySQL. A complete backup needs both.

Back up project files

1. Stop active editing and save every file.
2. Locate the project folder inside `htdocs`.
3. Copy the entire project folder to an approved backup location.
4. Name the backup clearly with project and date.
5. Open the copy and confirm key PHP, CSS, JavaScript and asset files are present.

Do not use a synchronised copy as the only backup.

Export the database

1. Start Apache and MySQL.
2. Open `http://localhost/phpmyadmin/`.
3. Select the project database.
4. Choose **Export**.
5. Use SQL format and complete the export.
6. Save the `.sql` file beside the project-file backup.

Do not treat XAMPP's raw `mysql` storage folder as a portable database backup.

Protect sensitive information

Use fictional data only. Check that backups and screenshots contain no real student records, personal passwords or hosted-server credentials.

Verify the backup

A backup is not proven until it can be inspected or restored. Confirm:

- project archive opens
- expected folders and files are present
- SQL file is not empty
- SQL contains expected table definitions
- backup location is accessible from another approved device

Check

- ☐ Project files and SQL export are both included.
- ☐ Backup name and date are clear.
- ☐ Original project still runs.
- ☐ Backup contains only fictional data.
- ☐ At least one recoverable copy exists outside the working folder.

Restoring a XAMPP Project from Backup

Restoring a XAMPP Project from Backup recreates both the website files and the MySQL database. Work from a copy of the backup so the original remains unchanged.

Restore project files

1. Install and test XAMPP.
2. Stop Apache and MySQL before replacing project files.
3. Copy the backed-up project folder into `htdocs`.
4. Confirm the folder contains the expected entry page and assets.
5. Start Apache and open the project through localhost.

Restore the database

1. Start Apache and MySQL.
2. Open `http://localhost/phpmyadmin/`.
3. Create the database name expected by the project, using `utf8mb4`.
4. Select the new database and choose **Import**.
5. Choose the backed-up `.sql` file.
6. Complete the import and inspect tables and fictional records.

If the SQL export already creates and selects the database, read it before creating another database manually.

Check configuration

Confirm the PDO connection file uses the restored database name and the local XAMPP credentials approved for class. Do not expose credentials in screenshots or public repositories.

Verify the restoration

Test:

- public entry page
- registration with a fictional user
- login and logout
- role-protected pages
- database queries and processed outputs
- administrator CSV import with a small fictional file

A page loading is not enough; confirm that database changes persist.

Check

- ☐ Files are inside the correct `htdocs` project folder.
- ☐ SQL import completes without unresolved errors.
- ☐ PDO connects to the restored database.
- ☐ Authentication and role checks work.
- ☐ Imported and processed data matches expected fictional results.
- ☐ Original backup remains unchanged.

Moving a XAMPP Project to Another Computer

Moving a XAMPP project requires a verified backup of both the project files and its database.

Prepare the source computer

1. Save all source files.
2. Export the project database as SQL through phpMyAdmin.
3. Copy the complete project folder from `htdocs`.
4. Include the SQL export and a short setup note.
5. Stop MySQL and Apache cleanly.

The setup note should record the project folder, local URL, database name and any approved port changes. Do not record real passwords.

Transfer safely

Use an approved storage location. Keep a separate source backup until the destination has been tested. Do not email sensitive archives or use real personal data.

Restore on the destination

1. Install and test XAMPP.
2. Copy the project folder into the destination `htdocs`.
3. Create or select the expected database.
4. Import the SQL file.
5. check the PDO connection settings.
6. Start Apache and MySQL.
7. open the project through localhost.

Test the complete journey

Use fictional accounts and data to test registration, login, roles, protected pages, CSV import, database storage and meaningful outputs.

Common problems

Problem	Check
Page not found	Folder name and localhost URL
Database connection fails	Database name and PDO settings
Tables missing	Correct SQL file and selected database
Styles or images missing	Relative paths and filename case
Login fails	User table, complete hashes and session setup
Import fails	Role, form encoding, file limits and headers

Check

- ☐ Files and SQL were transferred.
- ☐ Source backup remains recoverable.
- ☐ Destination passes the full user journey.
- ☐ No real credentials or personal data are included.
- ☐ Configuration differences are recorded.

Helper Tools

Classroom tools for checking HTML and CSS and generating fictional test password hashes.

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.

XAMPP Setup and Projects

Install, test and organise a local XAMPP environment for PHP and MySQL projects.

Installing XAMPP on Windows

XAMPP gives you a local web server on your Windows computer. It lets you run PHP pages, use a MySQL-compatible database and manage that database through phpMyAdmin without publishing your project to the internet.

By the end of this tutorial, you will have XAMPP installed and be able to open its Control Panel. Starting and testing the services is covered in the separate **Starting and Testing Apache and MySQL** resource.

“ **Use fictional data only.** Never place real student or teacher information, real passwords or other sensitive information in a classroom project.

Before you begin

You will need:

- a Windows computer
- permission to install software
- enough free storage for XAMPP and your projects
- access to the official Apache Friends website
- your teacher or school IT support if installation is restricted

If you are using a managed school computer, check with your teacher before beginning. Do not try to bypass administrator restrictions.

What XAMPP installs

XAMPP bundles the main tools needed for a local PHP and database project:

Component	Purpose
-----------	---------

Apache	Serves your project pages through a local web address such as <code>http://localhost/</code> .
PHP	Runs the server-side code in your <code>.php</code> files.
MariaDB/MySQL	Stores users, imported CSV records and other application data. XAMPP currently supplies MariaDB, which is compatible with the MySQL work used in these tutorials.
phpMyAdmin	Provides a browser-based interface for creating and inspecting databases and tables.

You do not need to understand every component before installing it. Later tutorials will introduce each tool when you use it.

1. Download XAMPP from the official source

1. Open the [official Apache Friends XAMPP website](#).
2. Choose **XAMPP for Windows**.
3. Download the current Windows installer.
4. Wait for the download to finish before opening the file.

“ **Security check:** Use the official Apache Friends website. Do not download repackaged installers from advertisements, download mirrors or unfamiliar websites.

The exact version number may differ from screenshots or demonstrations. That is normal. Use the current version approved by your teacher or school.

2. Run the installer

1. Open the downloaded installer.
2. If Windows displays a User Account Control prompt, confirm that the publisher and file are the installer you intentionally downloaded.
3. Select **Yes** only if you have permission to install software.
4. If antivirus software displays a warning, follow your school's instructions or ask your teacher rather than disabling security software.

Some managed computers require an administrator to complete this step.

3. Select the required components

Keep these components selected:

- **Apache**
- **MySQL**
- **PHP**
- **phpMyAdmin**

Other components are not required for the projects in this tutorial sequence. Leaving the default selection is acceptable if your teacher has not asked you to change it.

4. Choose the installation folder

Use the default local folder unless your teacher or school IT team specifies another location:

C:\xampp

A short local path helps avoid permission and path problems.

Avoid installing XAMPP inside OneDrive, another synchronised folder or a network drive. Synchronisation can lock or move files while Apache or MySQL is using them.

5. Complete the installation

1. Continue through the installer.
2. Choose your preferred language when prompted.
3. Wait while the required files are copied.
4. Leave the option to open the **XAMPP Control Panel** selected.
5. Select **Finish**.

The first launch may trigger a Windows Firewall prompt. Permit access only for the network types approved by your school. For normal classroom development, XAMPP is used locally on your own computer.

6. Locate the important folders

Open the XAMPP installation folder. You should be able to find:

- `xampp-control.exe` — opens the XAMPP Control Panel
- `htdocs` — stores your PHP web projects
- `mysql` — contains the database software and its files

Do not manually edit or delete files inside the `mysql` folder. You will manage databases through phpMyAdmin and PHP.

Check your installation

Before continuing, confirm each item:

- ☐ XAMPP is installed in an approved local folder.
- ☐ The XAMPP Control Panel opens.
- ☐ The Control Panel shows rows for Apache and MySQL.
- ☐ The `htdocs` folder exists.
- ☐ You know where `xampp-control.exe` is located.
- ☐ You have not entered or copied any real personal data.

You do **not** need to make Apache and MySQL run successfully in this tutorial. Use the standalone **Starting and Testing Apache and MySQL** resource when you are ready to run both services and diagnose common port conflicts.

Common installation problems

Problem	What to do
The installer is blocked	Ask your teacher or school IT team to install or approve XAMPP.
Windows requests administrator access	Continue only if you have permission and recognise the official installer.
The Control Panel shortcut is missing	Open your XAMPP folder and run <code>xampp-control.exe</code> .
XAMPP was installed in a synchronised folder	Ask your teacher whether it should be reinstalled in an approved local folder such as <code>C:\xampp</code> .

Problem	What to do
Security software displays a warning	Do not disable it. Follow the school process or ask for assistance.
Your screen looks slightly different	Version changes are normal. Look for the same component names and actions.

Starting and Testing Apache and MySQL

Apache runs PHP pages and MySQL stores application data. Both services must be running while you develop or demonstrate a database-backed project.

Start the services

1. Open **XAMPP Control Panel**.
2. Select **Start** beside Apache.
3. Select **Start** beside MySQL.
4. Wait until both rows are highlighted and show running process identifiers and ports.

Do not select **Svc** or install Windows services unless your teacher or IT team asks you to.

Test Apache and MySQL

Open `http://localhost/`. A XAMPP dashboard confirms Apache is responding.

Then open `http://localhost/phpmyadmin/`. If phpMyAdmin appears, Apache is serving the interface and MySQL is available.

“ **Local does not mean public.** A localhost page normally runs only on your computer. Continue to use fictional project data.

Understand the status

Status	Meaning	Action
Green/running	Service started	Continue working

Status	Meaning	Action
Starts then stops	A port or file may be unavailable	Read and record the log
Red error text	XAMPP reports a specific problem	Keep the message for troubleshooting
Browser cannot connect	Service or address may be wrong	Recheck both

Apache commonly uses ports 80 and 443; MySQL commonly uses 3306. Ask before changing ports or stopping unfamiliar Windows services. Never disable security software to solve a conflict.

Stop services cleanly

Close database pages, select **Stop** beside MySQL, then stop Apache and close the Control Panel.

Verification checklist

- ☐ Apache and MySQL show as running.
- ☐ Both localhost addresses open.
- ☐ You can stop both services cleanly.
- ☐ Any error is recorded exactly before troubleshooting.

Creating Your First XAMPP PHP Project

A XAMPP project belongs inside the `htdocs` folder so Apache can serve it and process PHP.

Create the project

Inside your XAMPP `htdocs` folder, create `campready`. Use short, meaningful lowercase folder names without spaces.

Create `index.php` inside it:

```
<?php
$projectName = "CampReady";
$currentYear = date("Y");
?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title><?= htmlspecialchars($projectName) ?></title>
</head>
<body>
    <h1><?= htmlspecialchars($projectName) ?></h1>
    <p>Local PHP project running in <?= $currentYear ?>.</p>
</body>
</html>
```

Start Apache and open `http://localhost/campready/`.

Do not double-click the PHP file. A `file:///` address bypasses Apache, so PHP will not run.

What the code demonstrates

- variables hold the project name and year
- PHP output is inserted into HTML
- `htmlspecialchars()` safely encodes displayed text
- `index.php` loads automatically for the folder URL

Suggested structure

```
campready/  
|-- index.php  
|-- css/  
|   |-- style.css  
|-- includes/  
|-- uploads/  
|-- assets/
```

Create only the folders your solution needs. Keep reusable PHP files outside writable upload folders.

Troubleshooting

Symptom	Check
Page not found	Folder spelling and location in <code>htdocs</code>
PHP appears as text	Apache is running and URL starts with localhost
Blank page	PHP syntax and error log
Old content	Save and refresh without cache
Styles missing	Relative path and exact filename

Verification checklist

- ☐ The project is inside `htdocs`.
- ☐ The main file is `index.php`.

- ☐ It opens through localhost.
- ☐ PHP outputs the current year.
- ☐ Files contain no real personal data or credentials.

Creating a MySQL Database with phpMyAdmin

A database organises persistent information in related tables. This practice creates a small fictional database without prescribing an assessed design.

Create the database

- 1. Start Apache and MySQL.
- 2. Open `http://localhost/phpmyadmin/`.
- 3. Select **Databases**.
- 4. Create `campready_db` with a Unicode collation such as `utf8mb4_unicode_ci`.

Use lowercase names and underscores. Avoid spaces and personal names.

Create a practice table

Create `activities` with these columns:

Column	Type	Settings	Purpose
<code>activity_id</code>	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier
<code>activity_name</code>	VARCHAR(100)	NOT NULL	Activity label
<code>location</code>	VARCHAR(100)	NOT NULL	Fictional location
<code>capacity</code>	INT	NOT NULL	Maximum participants

“ This is practice only. Design assessment tables from the actual data, relationships and user needs.

Add fictional rows

activity_name	location	capacity
Navigation challenge	North field	24
Shelter setup	Camp zone A	30
Team cooking	Kitchen shelter	20

Leave the identifier blank because AUTO_INCREMENT creates it.

The equivalent SQL is:

```
CREATE TABLE activities (  
    activity_id INT AUTO_INCREMENT PRIMARY KEY,  
    activity_name VARCHAR(100) NOT NULL,  
    location VARCHAR(100) NOT NULL,  
    capacity INT NOT NULL  
);
```

Design principles

- Give every table a primary key.
- Match data types to values.
- Use `NOT NULL` for essential values.
- Do not store several separate values in one field.
- Separate entities into related tables.
- Use fabricated classroom data only.

Verification checklist

- ☐ The database and table exist.
- ☐ The table has an auto-incrementing primary key.
- ☐ Fields use suitable types.
- ☐ Three fictional rows appear in **Browse**.
- ☐ You can explain why each field exists.

Checking a Project Before Development

A short environment check prevents installation, folder and service problems from consuming development time.

Environment check

Check	Successful evidence
XAMPP opens	Control Panel loads
Apache runs	<code>http://localhost/</code> opens
MySQL runs	phpMyAdmin opens
Project location	Folder is inside <code>htdocs</code>
PHP processing	A PHP page outputs a variable or calculation
Database access	Practice database is visible
File editing	Saved changes appear after refresh

Diagnostic page

Create `environment-check.php` in the project:

```
<?php
$checks = [
    "PHP is running",
    "Files are served by Apache",
    "Server time: " . date("H:i:s")
];
?>
<!doctype html>
```

```
<html lang="en">
<head><meta charset="utf-8"><title>Environment check</title></head>
<body>
<h1>Environment check</h1>
<ul>
<?php foreach ($checks as $check): ?>
    <li><?= htmlspecialchars($check) ?></li>
<?php endforeach; ?>
</ul>
</body>
</html>
```

Open it through localhost. The example uses iteration to output a list; it is a diagnostic tool, not a complete solution.

Safe working practices

- Use fictional data.
- Keep files in a clearly named project folder.
- Never expose passwords in screenshots or public repositories.
- Stop services cleanly.
- Keep recoverable backups of files and databases.
- Record approved configuration changes.

Record the setup

Record the installation path, project folder, local URL, database name, any approved port changes and the most recent successful check. Do not record real passwords.

Final checklist

- ☐ Diagnostic page opens through localhost.
- ☐ PHP outputs the server time.
- ☐ phpMyAdmin displays the database.
- ☐ Only fictional data is used.
- ☐ Important folders and URLs are recorded.

☐ Any unresolved problem includes its exact error message.