# **PHP File Upload Website**

**What this contains**: a simple PHP/MySQL website with public & private folders, user registration/login, file & folder upload, admin deletion for public files, and CSS animations. Drop this into your Apache htdocs (or similar) and follow the setup instructions below.

### **Project structure**

```
project-root/
                      # Database connection (edit credentials)
⊢ db.php
                     # SQL to create database + tables

⊢ init.sql

    index.php

                     # Home page (Upload / View / Login links)

    ⊢ register.php

                     # User registration
├ login.php
                     # User login
⊢ logout.php
                     # Logout
─ upload.php
                      # Upload file / create folder (public or private)
# View files (public, and user's private when logged
in)
─ admin.php
                     # Admin page to manage public files (delete)
                # Basic security rules (optional)
⊢ .htaccess
⊢ assets/
                    # Styling + animations
  ─ style.css
  └ script.js # Small client JS
└ uploads/

⊢ public/
                     # Files visible to everyone (admin can delete)
  └ private/
                   # Private folders per-user, created as private/
{user_id}/
```

## init.sql (create DB + tables)

```
-- Create database (edit name if needed)

CREATE DATABASE IF NOT EXISTS file_portal CHARACTER SET utf8mb4 COLLATE

utf8mb4_unicode_ci;

USE file_portal;

-- users table

CREATE TABLE IF NOT EXISTS users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) NOT NULL UNIQUE,
   email VARCHAR(100) NOT NULL UNIQUE,
```

```
phone VARCHAR(20),
  password VARCHAR(255) NOT NULL,
  is admin TINYINT(1) DEFAULT 0,
  created at TIMESTAMP DEFAULT CURRENT TIMESTAMP
) ENGINE=InnoDB;
-- files table (metadata)
CREATE TABLE IF NOT EXISTS files (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user id INT DEFAULT NULL, -- null for uploads to public by anyone
  filename orig VARCHAR(255) NOT NULL,
  filename disk VARCHAR(255) NOT NULL,
  folder_type ENUM('public','private') NOT NULL,
  folder_path VARCHAR(255) NOT NULL, -- relative path inside uploads/ e.g.
public/ or private/3/
  size INT NOT NULL,
  mime VARCHAR(100),
  uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user id) REFERENCES users(id) ON DELETE SET NULL
) ENGINE=InnoDB;
-- Create a default admin (change password hash after first login or generate
your own)
INSERT INTO users (username, email, phone, password, is_admin) VALUES
('admin', 'admin@example.com','', '" . md5('change_me_please') . "', 1);
```

Important: After importing init.sql, edit the admin password using PHP
password\_hash() or register an admin and update is\_admin=1 in the DB. The sample
INSERT uses an md5 placeholder — replace it.

# db.php

```
} catch (Exception $e) {
    die('Database connection failed: ' . $e->getMessage());
}

// Basic helper: ensure uploads folders exist
$baseUpload = __DIR__ . '/uploads';
$publicDir = $baseUpload . '/public';
$privateDir = $baseUpload . '/private';

if (!is_dir($baseUpload)) mkdir($baseUpload, 0755);
if (!is_dir($publicDir)) mkdir($publicDir, 0755);
if (!is_dir($privateDir)) mkdir($privateDir, 0755);
```

### assets/style.css

```
/* Basic, modern styling with small animations */
*{box-sizing:border-box}
body{font-family:Arial, Helvetica, sans-
serif;background:#f4f4f8;color:#222;margin:0;padding:0}
.header{background:linear-gradient(90deg,#111,#333);color:#fff;padding:18px
24px;display:flex;align-items:center;justify-content:space-between}
.header .brand{font-weight:700;letter-spacing:1px}
.container{max-width:980px;margin:28px auto;padding:18px}
.card{background:#fff;border-radius:10px;box-shadow:0 6px 20px rgba(10,10,20,
0.06);padding:18px;margin-bottom:18px;overflow:hidden}
.actions{display:flex;gap:12px}
.btn{background:#222;color:#fff;padding:10px 14px;border-radius:8px;text-
decoration:none;display:inline-block;transition:transform .15s ease, box-
shadow .15s}
.btn:hover{transform:translateY(-3px);box-shadow:0 10px 30px rgba(0,0,0,0.12)}
.form-row{display:flex;gap:12px;flex-wrap:wrap}
.input{padding:10px;border-radius:8px;border:1px solid #ddd;flex:1}
.small{font-size:13px;color:#666}
.file-list{display:grid;grid-template-columns:repeat(auto-fill,minmax(220px,
1fr));gap:12px}
.file-card{padding:12px;border-radius:8px;border:1px solid
#eee;background:linear-gradient(180deg,#fff,#fafafa);transition:transform .12s}
.file-card:hover{transform:translateY(-6px)}
.meta{font-size:12px;color:#666;margin-top:8px}
.notice{padding:10px;border-radius:8px;background:#fffbe6;border:1px solid
#ffe59e}
/* Simple animation */
@keyframes pop {0%{transform:scale(.98);opacity:.8}100%{transform:none;opacity:
```

```
1}}
.card{animation:pop .18s ease both}

/* responsive */
@media(max-width:600px){.header{flex-direction:column;gap:8px}.actions{flex-direction:column}}
```

## assets/script.js

```
// Simple client-side niceties
function confirmDelete(name){
  return confirm('Delete "' + name + '" ? This action cannot be undone.');
}
```

### index.php (home)

```
<?php
session_start();
include 'db.php';
$logged = isset($_SESSION['user']);
$user = $logged ? $_SESSION['user'] : null;
?>
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>File Portal</title>
  <link rel="stylesheet" href="assets/style.css">
</head>
<body>
  <div class="header">
    <div class="brand">File Portal</div>
    <div class="actions">
      <a class="btn" href="upload.php">Upload</a>
      <a class="btn" href="view.php">View</a>
      <?php if($logged): ?>
        <span class="btn">Hi, <?=htmlspecialchars($user['username'])?></span>
        <a class="btn" href="logout.php">Logout</a>
        <?php if($user['is_admin']): ?>
          <a class="btn" href="admin.php">Admin</a>
```

```
<?php endif; ?>
     <?php else: ?>
       <a class="btn" href="login.php">Login</a>
       <a class="btn" href="register.php">Register</a>
      <?php endif; ?>
   </div>
 </div>
 <div class="container">
   <div class="card">
      <h2>Welcome</h2>
      Public folder is visible to all. Private folders are for
logged-in users only.
     <div style="margin-top:12px">
       <a class="btn" href="upload.php">Go to Upload</a>
       <a class="btn" href="view.php">Browse Files</a>
     </div>
   </div>
 </div>
 <script src="assets/script.js"></script>
</body>
</html>
```

## register.php

```
<?php
session_start();
include 'db.php';
$err = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST'){
  $username = trim($_POST['username']);
  $email = trim($ POST['email']);
  $phone = trim($_POST['phone']);
  $password = $_POST['password'];
  if (!$username || !$email || !$password){ $err = 'Please fill required
fields.'; }
  else {
    // insert
    $hash = password_hash($password, PASSWORD_DEFAULT);
    $stmt = $pdo->prepare('INSERT INTO users (username, email, phone, password)
VALUES (?,?,?,?)');
   try{
      $stmt->execute([$username,$email,$phone,$hash]);
      header('Location: login.php'); exit;
    }catch(Exception $e){ $err = 'Registration failed: ' . $e->getMessage(); }
```

```
}
}
<!doctype html>
<html><head><meta charset="utf-8"><meta name="viewport" content="width=device-
width,initial-scale=1"><title>Register</title>
<link rel="stylesheet" href="assets/style.css"></head><body>
<div class="header"><div class="brand">File Portal</div><div class="actions"><a</pre>
class="btn" href="index.php">Home</a></div></div>
<div class="container"><div class="card">
  <h3>Register</h3>
  <?php if($err): ?><div class="notice"><?=htmlspecialchars($err)?></div><?php</pre>
endif; ?>
  <form method="post">
    <div class="form-row"><input class="input" name="username"</pre>
placeholder="Username" required></div>
    <div class="form-row"><input class="input" name="email" placeholder="Email"</pre>
type="email" required></div>
    <div class="form-row"><input class="input" name="phone"</pre>
placeholder="Phone"></div>
    <div class="form-row"><input class="input" name="password"</pre>
placeholder="Password" type="password" required></div>
    <div style="margin-top:12px"><button class="btn">Register</button></div>
  </form>
</div></div>
</body></html>
```

# login.php

```
<?php
session_start();
include 'db.php';
$err = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST'){
    $email = trim($_POST['email']);
    $password = $_POST['password'];
    $stmt = $pdo->prepare('SELECT * FROM users WHERE email = ? OR username = ?
LIMIT 1');
    $stmt->execute([$email,$email]);
    $user = $stmt->fetch();
    if ($user && password_verify($password, $user['password'])){
        unset($user['password']);
        $_SESSION['user'] = $user;
        header('Location: index.php'); exit;
```

```
} else {
    $err = 'Invalid credentials.';
}
?>
<!doctype html>
<html><head><meta charset="utf-8"><meta name="viewport" content="width=device-
width,initial-scale=1"><title>Login</title>
<link rel="stylesheet" href="assets/style.css"></head><body>
<div class="header"><div class="brand">File Portal</div><div class="actions"><a</pre>
class="btn" href="index.php">Home</a></div></div>
<div class="container"><div class="card">
  <h3>Login</h3>
  <?php if($err): ?><div class="notice"><?=htmlspecialchars($err)?></div><?php</pre>
endif; ?>
  <form method="post">
    <div class="form-row"><input class="input" name="email" placeholder="Email</pre>
or Username" required></div>
    <div class="form-row"><input class="input" name="password"</pre>
placeholder="Password" type="password" required></div>
    <div style="margin-top:12px"><button class="btn">Login</button></div>
  </form>
</div></div>
</body></html>
```

# logout.php

```
<?php
session_start();
session_destroy();
header('Location: index.php');
exit;</pre>
```

# upload.php

```
<?php
session_start();
include 'db.php';
$err = '';
$success = '';
$logged = isset($_SESSION['user']);</pre>
```

```
$user id = $logged ? $ SESSION['user']['id'] : null;
// Allowed max size & types
MAX = 10 * 1024 * 1024; // 10MB
allowed = [
 'image/png','image/jpeg','image/gif','application/pdf','text/
plain','application/zip','application/octet-stream'
1;
if ($ SERVER['REQUEST METHOD'] === 'POST'){
 $folder_type = ($_POST['folder_type'] === 'private' && $logged) ? 'private' :
'public':
 $target_dir = $folder_type === 'public' ? $publicDir : ($privateDir . '/' .
$user id);
 if (!is_dir($target_dir)) mkdir($target_dir, 0755, true);
 // Create a folder inside target if provided
 if (!empty($ POST['create folder'])){
   $create = preg_replace('#[^a-zA-Z0-9_\- ]#','', $_POST['create_folder']);
   if ($create){
      $newPath = $target_dir . '/' . $create;
     if (!is_dir($newPath)) mkdir($newPath, 0755, true);
     $success = 'Folder created: ' . htmlspecialchars($create);
   }
 }
 // Handle file upload
 if (isset($_FILES['file']) && $_FILES['file']['error'] !== UPLOAD_ERR_NO_FILE)
    $f = $ FILES['file'];
    if ($f['error'] !== UPLOAD ERR OK){ $err = 'Upload error code: ' .
$f['error']; }
   elseif ($f['size'] > $MAX) { $err = 'File too large. Max 10MB.'; }
   else {
      $mime = mime content type($f['tmp name']);
      // allow unknown but you can tighten check
     // if (!in_array($mime, $allowed)) { $err = 'File type not allowed: ' .
$mime; }
     // else {
        $orig = basename($f['name']);
        $uniq = time() . '_' . bin2hex(random_bytes(6)) . '_' . preg_replace('/
[^a-zA-Z0-9. -]/','', $orig);
        $subfolder = (!empty($_POST['subfolder'])) ? preg_replace('#[^a-zA-
Z0-9 \- ]#','', $ POST['subfolder']) : '';
        $destFolder = $target_dir . ($subfolder ? '/' . $subfolder : '');
        if (!is dir($destFolder)) mkdir($destFolder, 0755, true);
        $dest = $destFolder . '/' . $uniq;
```

```
if (move uploaded file($f['tmp name'], $dest)){
          // store metadata
          $relPath = ($folder type === 'public') ? ('public' . ($subfolder?'/'.
$subfolder:'')) : ('private/' . $user_id . ($subfolder?'/'.$subfolder:''));
          $stmt = $pdo->prepare('INSERT INTO files (user_id, filename_orig,
filename_disk, folder_type, folder_path, size, mime) VALUES (?,?,?,?,?,?,?)');
          $stmt->execute([$user_id, $orig, $uniq, $folder_type, $relPath,
$f['size'], $mime]);
          $success = 'Uploaded: ' . htmlspecialchars($orig);
        } else { $err = 'Failed to move uploaded file.'; }
     // }
    }
  }
}
?>
<!doctype html>
<html><head><meta charset="utf-8"><meta name="viewport" content="width=device-
width,initial-scale=1"><title>Upload</title>
<link rel="stylesheet" href="assets/style.css"></head><body>
<div class="header"><div class="brand">File Portal</div><div class="actions"><a</pre>
class="btn" href="index.php">Home</a></div></div>
<div class="container"><div class="card">
  <h3>Upload File or Create Folder</h3>
  <?php if($err): ?><div class="notice"><?=htmlspecialchars($err)?></div><?php</pre>
endif; ?>
  <?php if($success): ?><div class="notice"><?=htmlspecialchars($success)?></</pre>
div><?php endif; ?>
  <form method="post" enctype="multipart/form-data">
    <label class="small">Choose target:</label>
    <div class="form-row">
      <select name="folder_type" class="input">
        <option value="public">Public (visible to all)</option>
        <?php if($logged): ?><option value="private">My Private (requires
login)</option><?php endif; ?>
      </select>
      <input class="input" name="subfolder" placeholder="Optional: subfolder</pre>
name" />
    </div>
    <div style="margin-top:10px">
      <input type="file" name="file">
    </div>
    <div style="margin-top:10px">
      <button class="btn">Upload</putton>
    </div>
    <hr>
    <h4>Create folder</h4>
    <div class="form-row">
      <input class="input" name="create_folder" placeholder="Folder name</pre>
```

### view.php

```
<?php
session_start();
include 'db.php';
$logged = isset($ SESSION['user']);
$user_id = $logged ? $_SESSION['user']['id'] : null;
// If download requested
if (!empty($_GET['download'])){
  $id = (int)$_GET['download'];
  $stmt = $pdo->prepare('SELECT * FROM files WHERE id = ? LIMIT 1');
  $stmt->execute([$id]);
  $row = $stmt->fetch();
  if ($row){
    // Check access: if private, ensure owner
   if ($row['folder type'] === 'private' && $row['user id'] != $user id){
      die('Access denied.');
    }
    $path = __DIR__ . '/uploads/' . $row['folder_path'] . '/' .
$row['filename_disk'];
    if (is file($path)){
      header('Content-Type: ' . ($row['mime'] ?? 'application/octet-stream'));
      header('Content-Disposition: attachment; filename="' .
basename($row['filename_orig']) . '"');
      readfile($path); exit;
    }
 }
 die('File not found.');
}
// List public files and user private files
$files = [];
$stmt = $pdo->query('SELECT * FROM files WHERE folder_type = "public" ORDER BY
uploaded_at DESC');
$publicFiles = $stmt->fetchAll();
$privateFiles = [];
```

```
if ($logged){
  $stmt = $pdo->prepare('SELECT * FROM files WHERE folder type = "private" AND
user id = ? ORDER BY uploaded at DESC');
  $stmt->execute([$user id]);
  $privateFiles = $stmt->fetchAll();
}
?>
<!doctype html>
<html><head><meta charset="utf-8"><meta name="viewport" content="width=device-
width,initial-scale=1"><title>View Files</title>
<link rel="stylesheet" href="assets/style.css"></head><body>
<div class="header"><div class="brand">File Portal</div><div class="actions"><a</pre>
class="btn" href="index.php">Home</a></div></div>
<div class="container">
  <div class="card">
    <h3>Public Files</h3>
    <div class="file-list">
      <?php foreach($publicFiles as $f): ?>
        <div class="file-card">
          <strong><?=htmlspecialchars($f['filename orig'])?></strong>
          <div class="meta">Size: <?=round($f['size']/1024,2)?> KB • Uploaded:
<?=$f['uploaded at']?></div>
          <div style="margin-top:10px"><a class="btn" href="view.php?download=<?"</pre>
=$f['id']?>">Download</a>
          <?php if(isset($ SESSION['user']) && $ SESSION['user']['is admin']): ?</pre>
            <a class="btn" href="admin.php?delete=<?=$f['id']?</pre>
>" onclick="return confirmDelete('<?=addslashes($f['filename orig'])?</pre>
>')">Delete</a>
          <?php endif; ?></div>
        </div>
      <?php endforeach; ?>
    </div>
  </div>
  <?php if($logged): ?>
  <div class="card">
    <h3>My Private Files</h3>
    <div class="file-list">
      <?php foreach($privateFiles as $f): ?>
        <div class="file-card">
          <strong><?=htmlspecialchars($f['filename_orig'])?></strong>
          <div class="meta">Size: <?=round($f['size']/1024,2)?> KB • Uploaded:
<?=$f['uploaded_at']?></div>
          <div style="margin-top:10px"><a class="btn" href="view.php?download=<?"</pre>
=$f['id']?>">Download</a></div>
        </div>
      <?php endforeach; ?>
```

```
</div>
</div>
</php endif; ?>
</div>
</cript src="assets/script.js"></script>
</body></html>
```

## admin.php

```
<?php
session start();
include 'db.php';
if (!isset($ SESSION['user']) || !$ SESSION['user']['is admin']){ die('Access
denied.'); }
// Handle deletion of public files
if (!empty($_GET['delete'])){
  $id = (int)$_GET['delete'];
  $stmt = $pdo->prepare('SELECT * FROM files WHERE id = ? AND folder_type =
"public"');
  $stmt->execute([$id]);
  $row = $stmt->fetch();
  if ($row){
    $path = __DIR__ . '/uploads/' . $row['folder_path'] . '/' .
$row['filename disk'];
    if (is_file($path)) unlink($path);
    $pdo->prepare('DELETE FROM files WHERE id = ?')->execute([$id]);
    header('Location: admin.php'); exit;
  }
}
$files = $pdo->query('SELECT * FROM files WHERE folder_type = "public" ORDER BY
uploaded at DESC')->fetchAll();
<!doctype html>
<html><head><meta charset="utf-8"><meta name="viewport" content="width=device-
width,initial-scale=1"><title>Admin</title>
<link rel="stylesheet" href="assets/style.css"></head><body>
<div class="header"><div class="brand">Admin - File Portal</div><div</pre>
class="actions"><a class="btn" href="index.php">Home</a></div></div>
<div class="container"><div class="card">
  <h3>Public files (manage)</h3>
  <div class="file-list">
   <?php foreach($files as $f): ?>
      <div class="file-card">
        <strong><?=htmlspecialchars($f['filename_orig'])?></strong>
```

### .htaccess (optional)

```
# prevent listing
Options -Indexes

# Basic security for uploads
<FilesMatch "\.(php|phtml|php5|phps)$">
    Deny from all
</FilesMatch>
```

Note: This blocks execution of PHP files inside the uploads folder if your server honors .htaccess. You may need to put a separate .htaccess in /uploads with rules to deny script execution.

# **Setup instructions**

- 1. Create the project folder in your webserver's root (e.g., C:/xampp/htdocs/file\_portal).
- 2. Place the files as listed above (create assets/ and uploads/ directories). Make uploads/ writable by the webserver (e.g., chmod 755 or 775).
- 3. Import init.sql into MySQL (phpMyAdmin or mysql CLI): mysql -u root -p < init.sql.
- 4. Edit db.php with your DB credentials.
- 5. Make sure PHP file\_uploads is enabled and upload\_max\_filesize and post\_max\_size support your limits.
- 6. Visit http://localhost/your-folder/index.php and register/login.

## **Security & next steps (recommended)**

- **Sanitize uploads**: The example accepts many types; tighten allowed MIME types and/or scan files for malware depending on use.
- Admin user: Create admin via DB or register and set [is\_admin=1].
- **Use HTTPS** to protect credentials.
- Rate limiting and CSRF protection for forms (add tokens) for production.
- Filename collisions: we used unique disk names; simpler mapping is in files table.

If you'd like, I can now: - convert this into a downloadable ZIP with all files, or - add CSRF tokens, stronger file type checks, or - add thumbnails for images and previews.

Tell me which of these you'd like next and I will update the project.