# view\_folder.php

<?php

// File: view\_folder.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

requireAuth();

$type = $\_GET['type'] ?? '';

if (!$type) {

    header('Location: dashboard.php');

    exit;

}

// Fetch files for this user & folder/type

$stmt = $pdo->prepare("SELECT \* FROM files WHERE user\_id = ? AND file\_type = ? ORDER BY uploaded\_at DESC");

$stmt->execute([$\_SESSION['user\_id'], $type]);

$files = $stmt->fetchAll();

// Count for display

$count = count($files);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title><?= ucfirst(sanitize($type)) ?> – Shona Cloud</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <link rel="stylesheet" href="assets/css/style.css">

</head>

<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

    <div class="container-fluid">

        <a class="navbar-brand" href="dashboard.php">Shona Cloud</a>

        <div class="collapse navbar-collapse">

            <ul class="navbar-nav ms-auto">

                <li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>

            </ul>

        </div>

    </div>

</nav>

<div class="container my-4">

    <div class="d-flex align-items-center mb-3">

        <h2 class="me-3 mb-0"><?= ucfirst(sanitize($type)) ?> (<?= $count ?>)</h2>

        <a href="dashboard.php" class="btn btn-secondary btn-sm me-2">⬅ Back to Folders</a>

        <!-- Folder-level Download (GET) -->

        <?php if ($count > 0): ?>

            <a href="includes/download\_folder.php?type=<?= urlencode($type) ?>"

               class="btn btn-outline-primary btn-sm me-2">Download All</a>

            <!-- Folder-level Delete (POST with confirmation) -->

            <form action="includes/delete\_folder.php" method="post" style="display:inline;" onsubmit="return confirmDeleteFolder('<?= sanitize($type) ?>');">

                <input type="hidden" name="type" value="<?= sanitize($type) ?>">

                <button type="submit" class="btn btn-outline-danger btn-sm">Delete Folder</button>

            </form>

        <?php else: ?>

            <button class="btn btn-outline-secondary btn-sm me-2" disabled>Download All</button>

            <button class="btn btn-outline-secondary btn-sm" disabled>Delete Folder</button>

        <?php endif; ?>

    </div>

    <div class="row g-4">

        <?php if (empty($files)): ?>

            <div class="col-12">

                <div class="alert alert-info text-center">No files in this folder yet.</div>

            </div>

        <?php else: ?>

            <?php foreach ($files as $f): ?>

                <div class="col-sm-6 col-md-4 col-lg-3">

                    <div class="card h-100 shadow-sm">

                        <div class="card-body d-flex flex-column">

                            <h6 class="card-subtitle mb-2 text-truncate"><?= sanitize($f['filename']) ?></h6>

                            <p class="card-text text-muted small mb-3">

                                Uploaded: <?= date('M j, Y', strtotime($f['uploaded\_at'])) ?>

                            </p>

                            <div class="mt-auto d-grid gap-2">

                                <a href="includes/download.php?id=<?= (int)$f['id'] ?>" class="btn btn-outline-primary btn-sm">Download</a>

                                <button type="button" class="btn btn-outline-secondary btn-sm" onclick="share(<?= (int)$f['id'] ?>)">Share</button>

                                <a href="includes/delete.php?id=<?= (int)$f['id'] ?>" class="btn btn-outline-danger btn-sm" onclick="return confirm('Delete <?= sanitize($f['filename']) ?>?');">Delete</a>

                            </div>

                        </div>

                    </div>

                </div>

            <?php endforeach; ?>

        <?php endif; ?>

    </div>

</div>

<script>

function confirmDeleteFolder(folder) {

    return confirm('Delete ALL files in folder "' + folder + '"? This action cannot be undone.');

}

</script>

</body>

</html>

# signup.php

<?php

// FILE: signup.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

// If user already logged in, redirect

if (isLoggedIn()) {

    header('Location: dashboard.php');

    exit;

}

$error = $\_SESSION['error'] ?? '';

unset($\_SESSION['error']);

// Load countries list

require\_once \_\_DIR\_\_ . '/includes/countries.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Signup – Shona Cloud</title>

    <meta name="viewport" content="width=device-width,initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <link rel="stylesheet" href="assets/css/style.css">

    <style>

      /\* small tweak so datalist input matches other controls size-wise \*/

      .country-input { min-width: 0; }

    </style>

</head>

<body class="d-flex vh-100 align-items-center justify-content-center bg-gradient-to-r from-purple-500 via-pink-500 to-red-500">

    <div class="bg-white p-4 rounded shadow w-100" style="max-width: 500px;">

        <h2 class="mb-3 text-center">Create Account</h2>

        <?php if ($error): ?>

            <div class="alert alert-danger"><?= sanitize($error) ?></div>

        <?php endif; ?>

        <form action="includes/register.php" method="post" class="row g-3">

            <div class="col-12">

                <label class="form-label">Full Name</label>

                <input type="text" name="full\_name" class="form-control" required>

            </div>

            <div class="col-md-6">

                <label class="form-label">Email</label>

                <input type="email" name="email" class="form-control" required>

            </div>

            <div class="col-md-6">

                <label class="form-label">Phone Number</label>

                <input type="tel" name="phone" class="form-control" required>

            </div>

            <div class="col-12">

                <label class="form-label">Country</label>

                <!-- Using a datalist gives a searchable dropdown-like UX without extra JS.

                     The input name must be 'country' to match existing register.php handling. -->

                <input list="countries" name="country" class="form-control country-input" placeholder="Start typing your country..." required>

                <datalist id="countries">

                    <?php foreach ($countries as $c): ?>

                        <option value="<?= htmlspecialchars($c, ENT\_QUOTES, 'UTF-8') ?>"></option>

                    <?php endforeach; ?>

                </datalist>

                <!-- If you prefer a classic select (non-searchable), replace the above with:

                <select name="country" class="form-select" required>

                    <option value="" selected disabled>Choose your country</option>

                    <?php foreach ($countries as $c): ?>

                        <option value="<?= htmlspecialchars($c, ENT\_QUOTES, 'UTF-8') ?>"><?= htmlspecialchars($c, ENT\_QUOTES, 'UTF-8') ?></option>

                    <?php endforeach; ?>

                </select>

                -->

            </div>

            <div class="col-md-6">

                <label class="form-label">Password</label>

                <input type="password" name="password" class="form-control" minlength="6" required>

            </div>

            <div class="col-12">

                <button type="submit" class="btn btn-primary w-100">Sign Up</button>

            </div>

        </form>

        <p class="mt-3 text-center">

            Already have an account? <a href="login.php">Login here</a>.

        </p>

    </div>

</body>

</html>

# share.php

<?php

// File: share.php

// Public download via token

if (isset($\_GET['token'])) {

  require 'includes/config.php';

  $stmt = $pdo->prepare("

    SELECT f.filename,f.path

    FROM shares s

    JOIN files f ON s.file\_id=f.id

    WHERE s.token=?

  ");

  $stmt->execute([$\_GET['token']]);

  $file = $stmt->fetch();

  if ($file) {

    header('Content-Disposition: attachment; filename="'.basename($file['filename']).'"');

    readfile($file['path']);

    exit;

  }

}

http\_response\_code(404);

echo 'File not found.';

?>

# profile.php

<?php

// File: profile.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

requireAuth();

// Fetch user info

$stmt = $pdo->prepare("SELECT full\_name, email, phone, country FROM users WHERE id = ?");

$stmt->execute([$\_SESSION['user\_id']]);

$user = $stmt->fetch();

$error = $\_SESSION['error'] ?? '';

$success = $\_SESSION['success'] ?? '';

unset($\_SESSION['error'], $\_SESSION['success']);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Edit Profile – Shona Cloud</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body class="bg-light">

    <div class="container mt-5">

        <div class="col-md-6 offset-md-3">

            <h3>Edit Your Profile</h3>

            <?php if ($error): ?>

                <div class="alert alert-danger"><?= sanitize($error) ?></div>

            <?php elseif ($success): ?>

                <div class="alert alert-success"><?= sanitize($success) ?></div>

            <?php endif; ?>

            <form action="includes/update\_profile.php" method="post" class="mt-4">

                <div class="mb-3">

                    <label class="form-label">Full Name</label>

                    <input type="text" name="full\_name" class="form-control" value="<?= sanitize($user['full\_name']) ?>" required>

                </div>

                <div class="mb-3">

                    <label class="form-label">Phone</label>

                    <input type="text" name="phone" class="form-control" value="<?= sanitize($user['phone']) ?>" required>

                </div>

                <div class="mb-3">

                    <label class="form-label">Country</label>

                    <input type="text" name="country" class="form-control" value="<?= sanitize($user['country']) ?>" required>

                </div>

                <div class="mb-3">

                    <label class="form-label">New Password <small class="text-muted">(leave blank to keep current)</small></label>

                    <input type="password" name="password" class="form-control" minlength="6">

                </div>

                <button type="submit" class="btn btn-primary">Update Profile</button>

                <a href="dashboard.php" class="btn btn-secondary ms-2">Cancel</a>

            </form>

        </div>

    </div>

</body>

</html>

# logout.php

<?php

//File: logout.php

session\_start();

session\_unset();

session\_destroy();

header('Location: index.php');

exit;

?>

# login.php

<?php

//File: login.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

if (isLoggedIn()) {

    header('Location: dashboard.php');

    exit;

}

$error = $\_SESSION['error'] ?? '';

unset($\_SESSION['error']);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Log In – Shona Cloud</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body class="d-flex vh-100 align-items-center justify-content-center bg-light">

    <div class="card p-4 shadow-sm w-100" style="max-width: 400px;">

        <h3 class="mb-3 text-center">Log In</h3>

        <?php if ($error): ?>

            <div class="alert alert-danger"><?= sanitize($error) ?></div>

        <?php endif; ?>

        <form action="includes/login.php" method="post">

            <div class="mb-3">

                <label class="form-label">Email address</label>

                <input type="email" name="email" class="form-control" required autofocus>

            </div>

            <div class="mb-3">

                <label class="form-label">Password</label>

                <input type="password" name="password" class="form-control" required>

            </div>

            <button type="submit" class="btn btn-primary w-100">Log In</button>

        </form>

        <p class="mt-3 text-center">

            Don’t have an account? <a href="signup.php">Sign up</a>

        </p>

    </div>

</body>

</html>

# index.php

<?php

// File: index.php

// Landing page

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,initial-scale=1">

    <title>Shona Cloud</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <link rel="stylesheet" href="assets/css/style.css">

</head>

<body class="d-flex vh-100 align-items-center justify-content-center bg-gradient-to-r from-purple-500 via-pink-500 to-red-500">

    <div class="text-center text-white p-5 rounded shadow-lg bg-opacity-75 bg-dark">

        <h1 class="display-4">Welcome to Shona Cloud</h1>

        <p class="lead">Securely store, share, and manage your files.</p>

        <a href="login.php" class="btn btn-light m-2">Login</a>

        <a href="signup.php" class="btn btn-outline-light m-2">Signup</a>

    </div>

</body>

</html>

# delete\_account.php

<?php

//File: delete\_account.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

require\_once \_\_DIR\_\_ . '/includes/helpers.php';

// Double safeguard

if (empty($\_SESSION['user\_id'])) {

    header('Location: login.php');

    exit;

}

requireAuth();

?>

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Delete Account</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body class="container py-5">

  <h2 class="text-danger">Are you sure you want to delete your account?</h2>

  <p>This action is irreversible. All your files will be permanently deleted.</p>

  <form action="includes/delete\_user.php" method="POST">

    <button type="submit" class="btn btn-danger">Yes, Delete My Account</button>

    <a href="profile.php" class="btn btn-secondary">Cancel</a>

  </form>

</body>

</html>

# dashboard.php

<?php

// File: dashboard.php

require\_once \_\_DIR\_\_ . '/includes/config.php';

requireAuth();

$success = $\_SESSION['success'] ?? '';

unset($\_SESSION['success']);

// Determine display name: prefer session-stored full name, otherwise fetch from DB

$displayName = '';

if (!empty($\_SESSION['user\_full\_name'])) {

    $displayName = sanitize($\_SESSION['user\_full\_name']);

} elseif (!empty($\_SESSION['user\_id'])) {

    // fallback DB query (in case session value not set)

    $stmtName = $pdo->prepare("SELECT full\_name FROM users WHERE id = ? LIMIT 1");

    $stmtName->execute([$\_SESSION['user\_id']]);

    $rowName = $stmtName->fetch();

    $displayName = $rowName && !empty($rowName['full\_name']) ? sanitize($rowName['full\_name']) : 'User';

} else {

    $displayName = 'User';

}

// Search handling

$q = trim($\_GET['q'] ?? '');

$searchResults = [];

$searchCount = 0;

if ($q !== '') {

    // Limit results to avoid huge payloads; adjust as needed

    $limit = 500;

    $like = '%' . $q . '%';

    // NOTE: LIMIT cannot reliably be bound as a parameter in many MySQL/MariaDB drivers,

    // so we inject an integer-casted value into the SQL string (safe because of intval()).

    $sql = "

        SELECT id, filename, file\_type, uploaded\_at

        FROM files

        WHERE user\_id = ? AND filename LIKE ?

        ORDER BY uploaded\_at DESC

        LIMIT " . intval($limit);

    $stmt = $pdo->prepare($sql);

    $stmt->execute([$\_SESSION['user\_id'], $like]);

    $searchResults = $stmt->fetchAll(PDO::FETCH\_ASSOC);

    $searchCount = count($searchResults);

}

// If no search, get folder types and counts for the logged-in user

$fileTypes = [];

if ($q === '') {

    $stmt = $pdo->prepare("SELECT file\_type, COUNT(\*) AS cnt FROM files WHERE user\_id = ? GROUP BY file\_type ORDER BY file\_type");

    $stmt->execute([$\_SESSION['user\_id']]);

    $fileTypes = $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

// Optional message from actions (delete, etc.)

$msg = $\_GET['msg'] ?? '';

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Dashboard – Shona Cloud</title>

    <meta name="viewport" content="width=device-width,initial-scale=1">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <link rel="stylesheet" href="assets/css/style.css">

    <style>

        .folder-card {

            text-align: center;

            padding: 20px;

            background: #f8f9fa;

            border: 1px solid #ddd;

            border-radius: 8px;

            transition: background 0.2s ease;

            min-height: 170px;

            display: flex;

            flex-direction: column;

            justify-content: space-between;

        }

        .folder-card:hover { background: #e9ecef; }

        .folder-icon { font-size: 44px; color: #ffc107; }

        .folder-meta { margin-top: 8px; color: #6c757d; font-size: 0.9rem; }

        .file-card .card-subtitle { white-space: nowrap; overflow: hidden; text-overflow: ellipsis; }

        @media (max-width: 576px) { .folder-card { min-height: 160px; } }

    </style>

</head>

<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

    <div class="container-fluid">

        <a class="navbar-brand" href="dashboard.php">Shona Cloud</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navMain" aria-controls="navMain" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navMain">

            <ul class="navbar-nav me-auto">

                <!-- left side if needed -->

            </ul>

            <!-- Search form in navbar -->

            <form class="d-flex me-3" method="get" action="dashboard.php" role="search" onsubmit="return true;">

                <input class="form-control form-control-sm me-2" type="search" name="q" placeholder="Search files..." aria-label="Search files" value="<?= sanitize($q) ?>">

                <?php if ($q !== ''): ?>

                    <a href="dashboard.php" class="btn btn-sm btn-outline-light me-2">Clear</a>

                <?php endif; ?>

                <button class="btn btn-sm btn-light" type="submit">Search</button>

            </form>

            <ul class="navbar-nav ms-auto">

                <li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>

            </ul>

        </div>

    </div>

</nav>

<div class="container my-4">

    <?php if ($success): ?>

        <div class="alert alert-success"><?= sanitize($success) ?></div>

    <?php endif; ?>

    <?php if ($msg): ?>

        <div class="alert alert-info"><?= sanitize($msg) ?></div>

    <?php endif; ?>

    <div class="mb-4 d-flex flex-column flex-md-row align-items-start gap-3">

        <div>

            <h2 class="text-secondary mb-1">Welcome, <?= $displayName ?>!</h2>

            <p class="text-muted mb-0">Your file library is organised into folders by type.</p>

        </div>

        <div class="ms-auto d-flex gap-2">

            <a href="profile.php" class="btn btn-outline-secondary btn-sm align-self-start">Edit Profile</a>

            <a href="delete\_account.php" class="btn btn-outline-danger btn-sm align-self-start"

               onclick="return confirm('Are you sure you want to delete your account? All your files will be lost.');">Unsubscribe</a>

        </div>

    </div>

    <div class="card mb-4 shadow-sm">

        <div class="card-body">

            <h5 class="card-title">Upload a File</h5>

            <form action="includes/upload.php" method="post" enctype="multipart/form-data" class="d-flex gap-2 align-items-center">

                <input class="form-control" type="file" name="file" required>

                <button class="btn btn-success" type="submit">Upload</button>

            </form>

        </div>

    </div>

    <?php if ($q !== ''): ?>

        <div class="mb-3">

            <h4>Search results for "<?= sanitize($q) ?>" <small class="text-muted">(<?= $searchCount ?> found)</small></h4>

            <p class="text-muted mb-0">Showing up to <?= intval($limit) ?> most recent matches. Try a more specific phrase if you get too many results.</p>

        </div>

        <?php if (empty($searchResults)): ?>

            <div class="alert alert-info">No files matched your search.</div>

        <?php else: ?>

            <div class="row g-4">

                <?php foreach ($searchResults as $f): ?>

                    <div class="col-sm-6 col-md-4 col-lg-3">

                        <div class="card h-100 shadow-sm file-card">

                            <div class="card-body d-flex flex-column">

                                <h6 class="card-subtitle mb-2 text-truncate"><?= sanitize($f['filename']) ?></h6>

                                <p class="card-text text-muted small mb-3">

                                    Type: <?= sanitize($f['file\_type']) ?><br>

                                    Uploaded: <?= date('M j, Y', strtotime($f['uploaded\_at'])) ?>

                                </p>

                                <div class="mt-auto d-grid gap-2">

                                    <a href="includes/download.php?id=<?= (int)$f['id'] ?>" class="btn btn-outline-primary btn-sm">Download</a>

                                    <button type="button" class="btn btn-outline-secondary btn-sm" onclick="share(<?= (int)$f['id'] ?>)">Share</button>

                                    <a href="includes/delete.php?id=<?= (int)$f['id'] ?>" class="btn btn-outline-danger btn-sm" onclick="return confirm('Delete <?= sanitize($f['filename']) ?>?');">Delete</a>

                                </div>

                            </div>

                        </div>

                    </div>

                <?php endforeach; ?>

            </div>

        <?php endif; ?>

    <?php else: ?>

        <!-- Folder overview when no search -->

        <div class="row g-4">

            <?php if (empty($fileTypes)): ?>

                <div class="col-12">

                    <div class="alert alert-info text-center">No files yet. Upload something to get started!</div>

                </div>

            <?php else: ?>

                <?php foreach ($fileTypes as $row):

                    $type = $row['file\_type'];

                    $count = (int)$row['cnt'];

                    $typeEsc = urlencode($type);

                ?>

                    <div class="col-sm-6 col-md-4 col-lg-3">

                        <div class="folder-card shadow-sm d-flex flex-column">

                            <div>

                                <div class="folder-icon">📁</div>

                                <h5 class="mt-2 mb-0 text-truncate"><?= ucfirst(sanitize($type)) ?></h5>

                                <div class="folder-meta"><?= $count ?> file<?= $count !== 1 ? 's' : '' ?></div>

                            </div>

                            <div class="mt-3">

                                <div class="d-grid gap-2">

                                    <a href="view\_folder.php?type=<?= $typeEsc ?>" class="btn btn-primary btn-sm">Open</a>

                                    <a href="includes/download\_folder.php?type=<?= $typeEsc ?>" class="btn btn-outline-primary btn-sm" <?= $count === 0 ? 'aria-disabled="true" tabindex="-1"' : '' ?>>Download</a>

                                    <form action="includes/delete\_folder.php" method="post" onsubmit="return confirmDeleteFolder('<?= sanitize($type) ?>');" class="m-0">

                                        <input type="hidden" name="type" value="<?= sanitize($type) ?>">

                                        <button type="submit" class="btn btn-outline-danger btn-sm" <?= $count === 0 ? 'disabled' : '' ?>>Delete Folder</button>

                                    </form>

                                </div>

                            </div>

                        </div>

                    </div>

                <?php endforeach; ?>

            <?php endif; ?>

        </div>

    <?php endif; ?>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

<script src="assets/js/app.js"></script>

<script>

function confirmDeleteFolder(folder) {

    return confirm('Delete ALL files in folder "' + folder + '"? This action cannot be undone.');

}

</script>

</body>

</html>

# includes/config.php

<?php

// File: includes/config.php

session\_start();

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

define('DB\_HOST', 'localhost');

define('DB\_NAME', 'onlinestorage');

define('DB\_USER', 'root');

define('DB\_PASS', '');

require\_once \_\_DIR\_\_ . '/db.php';

require\_once \_\_DIR\_\_ . '/helpers.php';

?>

# includes/countries.php

<?php

// File: includes/countries.php

// A simple list of country names (common world countries).

// You can use these values as option values or datalist entries.

$countries = [

    "Afghanistan",

    "Albania",

    "Algeria",

    "Andorra",

    "Angola",

    "Antigua and Barbuda",

    "Argentina",

    "Armenia",

    "Australia",

    "Austria",

    "Azerbaijan",

    "Bahamas",

    "Bahrain",

    "Bangladesh",

    "Barbados",

    "Belarus",

    "Belgium",

    "Belize",

    "Benin",

    "Bhutan",

    "Bolivia",

    "Bosnia and Herzegovina",

    "Botswana",

    "Brazil",

    "Brunei",

    "Bulgaria",

    "Burkina Faso",

    "Burundi",

    "Cabo Verde",

    "Cambodia",

    "Cameroon",

    "Canada",

    "Central African Republic",

    "Chad",

    "Chile",

    "China",

    "Colombia",

    "Comoros",

    "Congo (Congo-Brazzaville)",

    "Costa Rica",

    "Côte d'Ivoire",

    "Croatia",

    "Cuba",

    "Cyprus",

    "Czechia (Czech Republic)",

    "Democratic Republic of the Congo",

    "Denmark",

    "Djibouti",

    "Dominica",

    "Dominican Republic",

    "Ecuador",

    "Egypt",

    "El Salvador",

    "Equatorial Guinea",

    "Eritrea",

    "Estonia",

    "Eswatini (fmr. \"Swaziland\")",

    "Ethiopia",

    "Federated States of Micronesia",

    "Fiji",

    "Finland",

    "France",

    "Gabon",

    "Gambia",

    "Georgia",

    "Germany",

    "Ghana",

    "Greece",

    "Grenada",

    "Guatemala",

    "Guinea",

    "Guinea-Bissau",

    "Guyana",

    "Haiti",

    "Honduras",

    "Hungary",

    "Iceland",

    "India",

    "Indonesia",

    "Iran",

    "Iraq",

    "Ireland",

    "Israel",

    "Italy",

    "Jamaica",

    "Japan",

    "Jordan",

    "Kazakhstan",

    "Kenya",

    "Kiribati",

    "Kuwait",

    "Kyrgyzstan",

    "Laos",

    "Latvia",

    "Lebanon",

    "Lesotho",

    "Liberia",

    "Libya",

    "Liechtenstein",

    "Lithuania",

    "Luxembourg",

    "Madagascar",

    "Malawi",

    "Malaysia",

    "Maldives",

    "Mali",

    "Malta",

    "Marshall Islands",

    "Mauritania",

    "Mauritius",

    "Mexico",

    "Moldova",

    "Monaco",

    "Mongolia",

    "Montenegro",

    "Morocco",

    "Mozambique",

    "Myanmar (Burma)",

    "Namibia",

    "Nauru",

    "Nepal",

    "Netherlands",

    "New Zealand",

    "Nicaragua",

    "Niger",

    "Nigeria",

    "North Korea",

    "North Macedonia",

    "Norway",

    "Oman",

    "Pakistan",

    "Palau",

    "Panama",

    "Papua New Guinea",

    "Paraguay",

    "Peru",

    "Philippines",

    "Poland",

    "Portugal",

    "Qatar",

    "Romania",

    "Russia",

    "Rwanda",

    "Saint Kitts and Nevis",

    "Saint Lucia",

    "Saint Vincent and the Grenadines",

    "Samoa",

    "San Marino",

    "Sao Tome and Principe",

    "Saudi Arabia",

    "Senegal",

    "Serbia",

    "Seychelles",

    "Sierra Leone",

    "Singapore",

    "Slovakia",

    "Slovenia",

    "Solomon Islands",

    "Somalia",

    "South Africa",

    "South Korea",

    "South Sudan",

    "Spain",

    "Sri Lanka",

    "Sudan",

    "Suriname",

    "Sweden",

    "Switzerland",

    "Syria",

    "Taiwan",

    "Tajikistan",

    "Tanzania",

    "Thailand",

    "Timor-Leste",

    "Togo",

    "Tonga",

    "Trinidad and Tobago",

    "Tunisia",

    "Turkey",

    "Turkmenistan",

    "Tuvalu",

    "Uganda",

    "Ukraine",

    "United Arab Emirates",

    "United Kingdom",

    "United States of America",

    "Uruguay",

    "Uzbekistan",

    "Vanuatu",

    "Vatican City",

    "Venezuela",

    "Vietnam",

    "Yemen",

    "Zambia",

    "Zimbabwe"

];

# includes/db.php

<?php

// File: includes/db.php

try {

    $dsn = "mysql:host=".DB\_HOST.";dbname=".DB\_NAME.";charset=utf8mb4";

    $pdo = new PDO($dsn, DB\_USER, DB\_PASS, [

      PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

      PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

    ]);

} catch (Exception $e) {

    die('DB connection error: ' . $e->getMessage());

}

?>

# includes/delete\_folder.php

<?php

// File: includes/delete\_folder.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if ($\_SERVER['REQUEST\_METHOD'] !== 'POST') {

    http\_response\_code(405);

    echo 'Method not allowed.';

    exit;

}

$type = $\_POST['type'] ?? '';

if (!$type) {

    http\_response\_code(400);

    echo 'Missing folder type.';

    exit;

}

// Fetch files for this user & type

$stmt = $pdo->prepare("SELECT id, filename, path FROM files WHERE user\_id = ? AND file\_type = ?");

$stmt->execute([$\_SESSION['user\_id'], $type]);

$files = $stmt->fetchAll();

if (empty($files)) {

    // Nothing to delete — redirect back

    header('Location: ../dashboard.php?msg=' . urlencode('nothing\_to\_delete'));

    exit;

}

$pdo->beginTransaction();

$deletedFiles = [];

$failedUnlinks = [];

foreach ($files as $f) {

    $path = $f['path'];

    $id = (int)$f['id'];

    // Attempt to unlink file if exists

    if (is\_file($path)) {

        if (@unlink($path)) {

            $deletedFiles[] = $id;

        } else {

            // mark as failed unlink but continue

            $failedUnlinks[] = $f['filename'];

        }

    } else {

        // File missing on disk — still delete DB record

        $deletedFiles[] = $id;

    }

}

// Delete DB records for collected ids

if (!empty($deletedFiles)) {

    $in  = str\_repeat('?,', count($deletedFiles) - 1) . '?';

    $sql = "DELETE FROM files WHERE id IN ($in) AND user\_id = ?";

    $params = array\_merge($deletedFiles, [$\_SESSION['user\_id']]);

    $stmtDel = $pdo->prepare($sql);

    $stmtDel->execute($params);

}

$pdo->commit();

// Build redirect message

if (!empty($failedUnlinks)) {

    $msg = 'Deleted records, but some files could not be removed from disk: ' . implode(', ', array\_slice($failedUnlinks, 0, 5));

    // Note: For long lists, only show first 5 filenames

    header('Location: ../dashboard.php?msg=' . urlencode($msg));

    exit;

}

// Success

header('Location: ../dashboard.php?msg=' . urlencode('folder\_deleted'));

exit;

# includes/delete\_user.php

<?php

// File: includes/delete\_user.php

require\_once \_\_DIR\_\_ . '/config.php';

require\_once \_\_DIR\_\_ . '/helpers.php';

if (empty($\_SESSION['user\_id'])) {

    header('Location: ../login.php');

    exit;

}

requireAuth();

$user\_id = $\_SESSION['user\_id'];

// Delete files from DB and storage

$stmt = $pdo->prepare("SELECT path FROM files WHERE user\_id = ?");

$stmt->execute([$user\_id]);

$files = $stmt->fetchAll();

foreach ($files as $file) {

    if (file\_exists($file['path'])) {

        unlink($file['path']);

    }

}

// Remove all subfolders and storage directory

$userDir = \_\_DIR\_\_ . '/../storage/' . $user\_id;

if (is\_dir($userDir)) {

    $it = new RecursiveDirectoryIterator($userDir, RecursiveDirectoryIterator::SKIP\_DOTS);

    $files = new RecursiveIteratorIterator($it, RecursiveIteratorIterator::CHILD\_FIRST);

    foreach ($files as $file) {

        if ($file->isDir()) {

            rmdir($file->getRealPath());

        } else {

            unlink($file->getRealPath());

        }

    }

    rmdir($userDir);

}

// Delete DB records

$pdo->prepare("DELETE FROM files WHERE user\_id = ?")->execute([$user\_id]);

$pdo->prepare("DELETE FROM shares WHERE file\_id IN (SELECT id FROM files WHERE user\_id = ?)")->execute([$user\_id]);

$pdo->prepare("DELETE FROM users WHERE id = ?")->execute([$user\_id]);

session\_destroy();

header("Location: ../index.php");

exit;

?>

# includes/delete.php

<?php

// File: includes/delete.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if (isset($\_GET['id'])) {

    $id = (int)$\_GET['id'];

    // Fetch file path

    $stmt = $pdo->prepare("SELECT path FROM files WHERE id = ? AND user\_id = ?");

    $stmt->execute([$id, $\_SESSION['user\_id']]);

    if ($f = $stmt->fetch()) {

        if (file\_exists($f['path'])) {

            unlink($f['path']);

        }

        $pdo->prepare("DELETE FROM files WHERE id = ?")->execute([$id]);

    }

}

header('Location: ../dashboard.php');

exit;

?>

# includes/download\_folder.php

<?php

// File: includes/download\_folder.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

$type = $\_GET['type'] ?? '';

if (!$type) {

    http\_response\_code(400);

    echo 'Missing folder type.';

    exit;

}

// Fetch files for user & type

$stmt = $pdo->prepare("SELECT filename, path FROM files WHERE user\_id = ? AND file\_type = ? ORDER BY uploaded\_at DESC");

$stmt->execute([$\_SESSION['user\_id'], $type]);

$files = $stmt->fetchAll();

if (empty($files)) {

    http\_response\_code(404);

    echo 'No files to download in this folder.';

    exit;

}

if (!class\_exists('ZipArchive')) {

    http\_response\_code(500);

    echo 'Server missing Zip extension.';

    exit;

}

// Create a temp zip file

$tmpFile = tempnam(sys\_get\_temp\_dir(), 'zip\_');

$zip = new ZipArchive();

if ($zip->open($tmpFile, ZipArchive::OVERWRITE) !== true) {

    http\_response\_code(500);

    echo 'Could not create zip file.';

    exit;

}

// Add files to zip

foreach ($files as $f) {

    $path = $f['path'];

    $displayName = $f['filename'];

    // Only add real files

    if (is\_file($path) && is\_readable($path)) {

        // Ensure unique names in zip (if duplicates, prefix with id or timestamp)

        $zip->addFile($path, $displayName);

    }

}

$zip->close();

// Stream the zip as attachment

if (filesize($tmpFile) === 0) {

    // Cleanup and error

    @unlink($tmpFile);

    http\_response\_code(500);

    echo 'Zip creation failed (empty zip).';

    exit;

}

// Clean any output buffers

if (ob\_get\_level()) {

    ob\_end\_clean();

}

$filenameSafe = preg\_replace('/[^a-zA-Z0-9\_\-\.]/', '\_', $type);

$downloadName = $filenameSafe . '\_' . date('Ymd\_His') . '.zip';

header('Content-Type: application/zip');

header('Content-Disposition: attachment; filename="' . $downloadName . '"');

header('Content-Length: ' . filesize($tmpFile));

header('Pragma: public');

header('Cache-Control: must-revalidate, post-check=0, pre-check=0');

header('Expires: 0');

readfile($tmpFile);

// remove temp file

@unlink($tmpFile);

exit;

# includes/download.php

<?php

// File: includes/download.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if (!isset($\_GET['id']) || !is\_numeric($\_GET['id'])) {

    http\_response\_code(400);

    echo 'Invalid request.';

    exit;

}

$id = (int)$\_GET['id'];

// Fetch the file and ensure it belongs to the logged-in user

$stmt = $pdo->prepare("SELECT filename, path FROM files WHERE id = ? AND user\_id = ?");

$stmt->execute([$id, $\_SESSION['user\_id']]);

$file = $stmt->fetch();

if (!$file) {

    http\_response\_code(404);

    echo 'File not found.';

    exit;

}

$path = $file['path'];

$filename = $file['filename'];

// Check that file exists on disk

if (!is\_file($path) || !is\_readable($path)) {

    http\_response\_code(404);

    echo 'File not found on server.';

    exit;

}

// Clear output buffers (important for large files)

if (ob\_get\_level()) {

    ob\_end\_clean();

}

// Determine mime type (best-effort)

$mime = 'application/octet-stream';

if (function\_exists('finfo\_open')) {

    $finfo = finfo\_open(FILEINFO\_MIME\_TYPE);

    if ($finfo) {

        $m = finfo\_file($finfo, $path);

        if ($m) $mime = $m;

        finfo\_close($finfo);

    }

} elseif (function\_exists('mime\_content\_type')) {

    $m = mime\_content\_type($path);

    if ($m) $mime = $m;

}

// Send headers

header('Content-Description: File Transfer');

header('Content-Type: ' . $mime);

header('Content-Disposition: attachment; filename="' . basename($filename) . '"');

header('Content-Transfer-Encoding: binary');

header('Expires: 0');

header('Cache-Control: must-revalidate');

header('Pragma: public');

header('Content-Length: ' . filesize($path));

// If you have large files and the server supports X-Sendfile / X-Accel-Redirect, prefer that approach.

// For now, simple streaming:

readfile($path);

exit;

# includes/helpers.php

<?php

// File: includes/helpers.php

function isLoggedIn() {

  return !empty($\_SESSION['user\_id']);

}

function requireAuth() {

  if (!isLoggedIn()) {

    header('Location: login.php');

    exit;

  }

}

function sanitize($data) {

  return htmlspecialchars($data, ENT\_QUOTES, 'UTF-8');

}

?>

# includes/login.php

<?php

// File: includes/login.php

require\_once \_\_DIR\_\_ . '/config.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $email    = filter\_var($\_POST['email'], FILTER\_VALIDATE\_EMAIL);

    $password = $\_POST['password'];

    if (!$email || !$password) {

        $\_SESSION['error'] = 'Please enter both email and password.';

        header('Location: ../login.php');

        exit;

    }

    // Look up user by email (also select full\_name)

    $stmt = $pdo->prepare("SELECT id, password\_hash, full\_name FROM users WHERE email = ?");

    $stmt->execute([$email]);

    $user = $stmt->fetch();

    if (!$user) {

        // No account with that email

        $\_SESSION['error'] = 'No account found for that email. Please sign up first.';

        header('Location: ../login.php');

        exit;

    }

    // Verify password

    if (!password\_verify($password, $user['password\_hash'])) {

        $\_SESSION['error'] = 'Incorrect password. Please try again.';

        header('Location: ../login.php');

        exit;

    }

    // Success! Store useful session values

    $\_SESSION['user\_id'] = $user['id'];

    // store full name for display on dashboard

    $\_SESSION['user\_full\_name'] = $user['full\_name'];

    // optional: store email too (used elsewhere)

    $\_SESSION['user\_email'] = $email;

    header('Location: ../dashboard.php');

    exit;

}

# includes/register.php

<?php

// File: includes/register.php

require\_once \_\_DIR\_\_ . '/config.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    // collect & validate

    $fullName = trim($\_POST['full\_name']);

    $email    = filter\_var($\_POST['email'], FILTER\_VALIDATE\_EMAIL);

    $phone    = trim($\_POST['phone']);

    $country  = trim($\_POST['country']);

    $password = $\_POST['password'];

    if (!$fullName || !$email || !$phone || !$country || strlen($password) < 6) {

        $\_SESSION['error'] = 'Please fill in all fields (password ≥ 6 chars).';

        header('Location: ../signup.php');

        exit;

    }

    // check for existing email

    $stmt = $pdo->prepare("SELECT id FROM users WHERE email = ?");

    $stmt->execute([$email]);

    if ($stmt->fetch()) {

        $\_SESSION['error'] = 'Email already registered.';

        header('Location: ../signup.php');

        exit;

    }

    // hash & insert

    $hash = password\_hash($password, PASSWORD\_DEFAULT);

    $stmt = $pdo->prepare("

      INSERT INTO users (full\_name, email, phone, country, password\_hash)

      VALUES (?, ?, ?, ?, ?)

    ");

    $stmt->execute([$fullName, $email, $phone, $country, $hash]);

    // redirect to login

    header('Location: ../login.php');

    exit;

}

?>

# includes/share\_link.php

<?php

// File: includes/share\_link.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if (isset($\_GET['id'])) {

  $id    = (int)$\_GET['id'];

  $token = bin2hex(random\_bytes(16));

  $stmt  = $pdo->prepare("INSERT INTO shares (file\_id, token, created\_at) VALUES (?, ?, NOW())");

  $stmt->execute([$id, $token]);

  header('Content-Type: application/json');

  echo json\_encode([

    'url' => "https://{$\_SERVER['HTTP\_HOST']}/share.php?token=$token"

  ]);

}

?>

# includes/update\_profile.php

<?php

// File: includes/update\_profile.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $fullName = trim($\_POST['full\_name']);

    $phone    = trim($\_POST['phone']);

    $country  = trim($\_POST['country']);

    $password = $\_POST['password'];

    // Validate required fields

    if (!$fullName || !$phone || !$country) {

        $\_SESSION['error'] = 'Please fill all required fields.';

        header('Location: ../profile.php');

        exit;

    }

    // Validate password length if provided

    if ($password && strlen($password) < 6) {

        $\_SESSION['error'] = 'Password must be at least 6 characters.';

        header('Location: ../profile.php');

        exit;

    }

    // Perform update

    if ($password) {

        $hash = password\_hash($password, PASSWORD\_DEFAULT);

        $stmt = $pdo->prepare("

            UPDATE users

            SET full\_name = ?, phone = ?, country = ?, password\_hash = ?

            WHERE id = ?

        ");

        $stmt->execute([$fullName, $phone, $country, $hash, $\_SESSION['user\_id']]);

    } else {

        $stmt = $pdo->prepare("

            UPDATE users

            SET full\_name = ?, phone = ?, country = ?

            WHERE id = ?

        ");

        $stmt->execute([$fullName, $phone, $country, $\_SESSION['user\_id']]);

    }

    // Redirect back to dashboard with success message

    $\_SESSION['success'] = 'Profile updated successfully.';

    header('Location: ../dashboard.php');

    exit;

}

?>

# includes/upload.php

<?php

// File: includes/upload.php

require\_once \_\_DIR\_\_ . '/config.php';

requireAuth();

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && !empty($\_FILES['file']['name'])) {

    $userDir = \_\_DIR\_\_ . '/../storage/' . $\_SESSION['user\_id'];

    if (!is\_dir($userDir)) mkdir($userDir, 0755, true);

    $filename = basename($\_FILES['file']['name']);

    $ext = strtolower(pathinfo($filename, PATHINFO\_EXTENSION));

    // File type categories

    $categories = [

        'images'    => ['jpg', 'jpeg', 'png', 'gif', 'bmp', 'svg', 'webp'],

        'documents' => ['pdf', 'doc', 'docx', 'xls', 'xlsx', 'ppt', 'pptx', 'txt', 'csv'],

        'videos'    => ['mp4', 'avi', 'mov', 'mkv', 'flv', 'wmv'],

        'audio'     => ['mp3', 'wav', 'aac', 'ogg', 'flac']

    ];

    $folder = 'others';

    foreach ($categories as $cat => $exts) {

        if (in\_array($ext, $exts)) {

            $folder = $cat;

            break;

        }

    }

    $typeDir = $userDir . '/' . $folder;

    if (!is\_dir($typeDir)) mkdir($typeDir, 0755, true);

    $target = $typeDir . '/' . $filename;

    if (move\_uploaded\_file($\_FILES['file']['tmp\_name'], $target)) {

        $stmt = $pdo->prepare("

            INSERT INTO files (user\_id, filename, path, file\_type, uploaded\_at)

            VALUES (?, ?, ?, ?, NOW())

        ");

        $stmt->execute([$\_SESSION['user\_id'], $filename, $target, $folder]);

    }

}

header('Location: ../dashboard.php');

exit;

?>

# assets/css/style.css

body {

  font-family: sans-serif;

}

.bg-gradient-to-r {

  background: linear-gradient(to right, #8e2de2, #c32dcf);

}

# assets/js/app.js

function share(fileId) {

  fetch(`includes/share\_link.php?id=${fileId}`)

    .then(res => res.json())

    .then(json => {

      prompt('Share this link:', json.url);

    })

    .catch(console.error);

}