File 1

```
about
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dominic A Cooper's Site</title>
    <!-- Link to external CSS file -->
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Hello World!</h1>
        <hr>
    </header>

    <main>
        <section>
            <p>My name is Dominic A Cooper</p>
            <p>This site is being created ad-infinitum. Dedicated to completing the study and use of Mathematics. To be an Engineer.</p>
        </section>
        <hr>
    </main>

    <footer>
        <a href="index.php"><button>Home</button></a>
    </footer>
</body>
</html>
```

File 2

```
admin
```

```
<?php
session_start();
```

```php
// Check if the user is logged in
if (!isset($_SESSION['logged_in']) || $_SESSION['logged_in'] !== true) {
    header('Location: login.php');
    exit;
}

// Define the base directory
$base_dir = __DIR__;

// Get the current directory from the query string or default to the base directory
$current_dir = isset($_GET['dir']) ? realpath($base_dir . '/' . $_GET['dir']) : $base_dir;
if (strpos($current_dir, $base_dir) !== 0) {
    $current_dir = $base_dir; // Prevent directory traversal attacks
}

// Function to evaluate script (mock for educational purposes)
function evaluate_script($code) {
    // IMPORTANT: In a real system, replace this with a secure sandboxed interpreter
    eval($code);
}

// Breadcrumb navigation
function breadcrumb($dir) {
    $base_dir = __DIR__;
    $parts = explode(DIRECTORY_SEPARATOR, str_replace($base_dir, '', $dir));
    echo "<nav><a href='?dir=" . urlencode($base_dir) . "'>Home</a> / ";
    $path = $base_dir;
    foreach ($parts as $part) {
        if ($part) {
            $path .= DIRECTORY_SEPARATOR . $part;
            echo "<a href='?dir=" . urlencode($path) . "'>" . htmlspecialchars($part) . "</a> / ";
        }
    }
    echo "</nav><br>";
}

// List files and directories
function list_files($dir, $filter = '*') {
    $files = glob("$dir/$filter");
    $dirs = glob("$dir/*", GLOB_ONLYDIR);

    echo "<div><button onclick=\"location.href='?dir=" . urlencode(dirname($dir)) . "'\">Back</button></div>";

    if (realpath($dir) !== realpath(__DIR__)) {
        echo "<div><a href='?dir=" . urlencode(dirname($dir)) . "'>.. (parent directory)</a></div>";
    }
    echo "<table>";
    echo "<tr><th>Name</th><th>Type</th><th>Size</th><th>Modified</th>
<th>Actions</th></tr>";

    foreach ($dirs as $directory) {
        echo "<tr><td>📁 <a href='?dir=" . urlencode($directory) . "'>" .
basename($directory) . "</a></td>";
        echo "<td>Directory</td><td>-</td><td>-</td>";
        echo "<td><a href='?edit_dir=" . urlencode($directory) . "'>Rename</a>
| ";
        echo "<a href='?delete_dir=" . urlencode($directory) . "'
onclick='return confirm(\"Are you sure you want to delete this directory?
\")'>Delete</a></td></tr>";
    }

    foreach ($files as $file) {
        echo "<tr><td>📄 " . basename($file) . "</td>";
        echo "<td>File</td><td>" . filesize($file) . " bytes</td><td>" .
date("F d Y H:i:s.", filemtime($file)) . "</td>";
        echo "<td><a href='?edit=" . urlencode($file) . "'>Edit</a> | ";
        echo "<a href='?download=" . urlencode($file) . "'>Download</a> | ";
        echo "<a href='?delete=" . urlencode($file) . "' onclick='return
confirm(\"Are you sure you want to delete this file?\")'>Delete</a></td></tr>";
    }

    echo "</table>";
}

// File and directory handling functions
function edit_file($file) {
    $content = file_exists($file) ? file_get_contents($file) : '';
    echo "<form method='post'>";
    echo "<textarea name='content' rows='10' cols='50'>" .
htmlspecialchars($content) . "</textarea><br>";
    echo "<input type='hidden' name='file' value='" . htmlspecialchars($file) .
"'>";
    echo "<input type='submit' name='save' value='Save'>";
    echo "</form>";
}

function save_file($file, $content) {
    file_put_contents($file, $content);
}

function delete_file($file) {
    unlink($file);
}

function delete_dir($dir) {
    rmdir($dir);
}
```

```php
function rename_item($old_name, $new_name) {
    rename($old_name, $new_name);
}

function handle_file_upload($dir) {
    if (isset($_FILES['upload_file']) && $_FILES['upload_file']['error'] === UPLOAD_ERR_OK) {
        $upload_file = $dir . '/' . basename($_FILES['upload_file']['name']);
        move_uploaded_file($_FILES['upload_file']['tmp_name'], $upload_file);
    }
}

function download_file($file) {
    if (file_exists($file)) {
        header('Content-Description: File Transfer');
        header('Content-Type: application/octet-stream');
        header('Content-Disposition: attachment; filename=' . basename($file));
        header('Expires: 0');
        header('Cache-Control: must-revalidate');
        header('Pragma: public');
        header('Content-Length: ' . filesize($file));
        readfile($file);
        exit;
    }
}

// Handling script submissions
if (isset($_POST['execute'])) {
    evaluate_script($_POST['script']);
}

// Handle file creation
if (isset($_POST['create_file'])) {
    $newFileName = $current_dir . '/' . $_POST['newFileName'];
    if (!empty($newFileName) && !file_exists($newFileName)) {
        file_put_contents($newFileName, ''); // Create a new file
    }
}

// Handle directory creation
if (isset($_POST['create_dir'])) {
    $newDirName = $current_dir . '/' . $_POST['newDirName'];
    if (!empty($newDirName) && !file_exists($newDirName)) {
        mkdir($newDirName); // Create a new directory
    }
}

// Handle file operations
if (isset($_GET['edit'])) {
    $edit_file = $_GET['edit'];
} elseif (isset($_GET['delete'])) {
    delete_file($_GET['delete']);
    header('Location: admin.php?dir=' . urlencode($current_dir));
    exit;
} elseif (isset($_GET['download'])) {
    download_file($_GET['download']);
    exit;
}

// Handle directory operations
if (isset($_GET['edit_dir'])) {
    $edit_dir = $_GET['edit_dir'];
} elseif (isset($_GET['delete_dir'])) {
    delete_dir($_GET['delete_dir']);
    header('Location: admin.php?dir=' . urlencode($current_dir));
    exit;
}

if (isset($_POST['save'])) {
    save_file($_POST['file'], $_POST['content']);
    header('Location: admin.php?dir=' . urlencode($current_dir));
    exit;
}

if (isset($_POST['rename'])) {
    rename_item($_POST['old_name'], $_POST['new_name']);
    header('Location: admin.php?dir=' .
urlencode(dirname($_POST['old_name'])));
    exit;
}

// Handle file uploads
if (isset($_POST['upload'])) {
    handle_file_upload($current_dir);
    header('Location: admin.php?dir=' . urlencode($current_dir));
    exit;
}

// File type filter
$filter = isset($_POST['filter']) ? $_POST['filter'] : '*';
?>
<!DOCTYPE html>
<html>
<head>
    <title>Admin - Portfolio File Manager</title>
    <link rel="stylesheet" href="astyle.css">
    <script>
        function confirmAction(message) {
            return confirm(message);
        }
    </script>
</head>
<body>

<h2>Portfolio File Manager</h2>
```

```
<?php breadcrumb($current_dir); ?>

<form method="post">
    <input type="text" name="newFileName" placeholder="Enter new file name (e.g., file.html)">
    <input type="submit" name="create_file" value="Create New File">
</form>

<form method="post">
    <input type="text" name="newDirName" placeholder="Enter new directory name">
    <input type="submit" name="create_dir" value="Create New Directory">
</form>

<form method="post">
    <input type="text" name="filter" placeholder="Enter file type filter (e.g., *.txt)" value="<?= htmlspecialchars($filter) ?>">
    <input type="submit" value="Filter Files">
</form>

<form method="post" enctype="multipart/form-data">
    <input type="file" name="upload_file">
    <input type="submit" name="upload" value="Upload File">
</form>

<?php if (isset($edit_file)): ?>
    <?php edit_file($edit_file); ?>
<?php elseif (isset($edit_dir)): ?>
    <form method="post">
        <input type="text" name="new_name" placeholder="New directory name">
        <input type="hidden" name="old_name" value="<?= htmlspecialchars($edit_dir) ?>">
        <input type="submit" name="rename" value="Rename Directory">
    </form>
<?php else: ?>
    <div>
        <h3>Files and Directories in <?= htmlspecialchars($current_dir) ?>: </h3>
        <?php list_files($current_dir, $filter); ?>
    </div>
<?php endif; ?>

<!-- Script Execution Form -->
<h3>Execute Script:</h3>
<form method="post">
    <textarea name="script" rows="5" cols="50" placeholder="Enter your script here..."></textarea><br>
    <input type="submit" name="execute" value="Execute Script">
</form>

<br><br>
<form action="logout.php" method="post"><button type="submit" name="logout">Logout</button></form>
```

```
</body>
</html>
```

File 3

```
alpha
```

```
/* alpha.css */

body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-color: #f5f5f5;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    background-color: #fff;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.input-section label {
    display: block;
    margin-top: 10px;
}

.input-section textarea {
    width: 100%;
    padding: 8px;
    box-sizing: border-box;
}

#compute-button, #generate-button {
    margin-top: 15px;
    padding: 10px 15px;
    font-size: 16px;
}

.result-section, .generator-section {
    margin-top: 20px;
}

.mappings {
    display: flex;
    justify-content: space-between;
}

.mappings div {
    width: 48%;
}

.mappings pre {
    background-color: #f0f0f0;
    padding: 10px;
    overflow: auto;
    max-height: 200px;
}

.generator-section pre {
    background-color: #f0f0f0;
    padding: 10px;
    overflow: auto;
    max-height: 200px;
}

.hidden {
    display: none;
}
```

File 4

```
alpha
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Alpha Mappings Generator</title>
    <link rel="stylesheet" href="alpha.css">
</head>
<body>
    <div class="container">
        <h1>Alpha Mappings Generator</h1>
        <div class="input-section">
            <label for="lhs-input">Enter the LHS string (any symbols):</label>
            <textarea id="lhs-input" rows="3"></textarea>

            <label for="rhs-input">Enter the RHS string (any symbols):</label>
            <textarea id="rhs-input" rows="3"></textarea>
            <button id="compute-button">Compute Mappings</button>
        </div>

        <div id="result-section" class="result-section hidden">
            <h2>Results</h2>
            <p id="i-value"></p>

            <div class="mappings">
                <div>
                    <h3>LHS Symbol Mappings:</h3>
                    <pre id="lhs-mappings"></pre>
                </div>
                <div>
                    <h3>RHS Symbol Mappings:</h3>
                    <pre id="rhs-mappings"></pre>
                </div>
            </div>

            <button id="generate-button">Generate Text Sequences</button>

            <div id="generator-section" class="generator-section hidden">
                <h2>Generated Text Sequences</h2>
                <div>
                    <h3>LHS Strings:</h3>
                    <pre id="lhs-strings"></pre>
                </div>
                <div>
                    <h3>RHS Strings:</h3>
                    <pre id="rhs-strings"></pre>
                </div>
            </div>
        </div>
    </div>

    <script src="alpha.js"></script>
</body>
</html>
```