1. PHP/debugging_optimization.php

```
<?php
function getUserPosts($userId, $db) {
    $stmt = $db->prepare("SELECT * FROM posts WHERE user_id = ?");
    $stmt->execute([$userId]);
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

// Use a persistent database connection
$db = new PDO('mysql:host=localhost;dbname=test', 'root', '', [
    PDO::ATTR_PERSISTENT => true,
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
]);

$userPosts = getUserPosts(1, $db);
?>
```

2. PHP/security_injection.php

```
<?php
$id = $_GET['id'];
$stmt = $conn->prepare("SELECT * FROM users WHERE id = ?");
$stmt->bind_param("i", $id);
$stmt->execute();
$result = $stmt->get_result();
?>
```

3. PHP/large_data_handling.php

```
<?php
function processLargeCSV($filePath, $pdo) {
    $handle = fopen($filePath, "r");
    $batchSize = 1000;
    $rows = [];

    while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
        $rows[] = $data;
        if (count($rows) >= $batchSize) {
            insertBatch($pdo, $rows);
            $rows = [];
        }
    }

    if (!empty($rows)) {
        insertBatch($pdo, $rows);
    }

    fclose($handle);
}

function insertBatch($pdo, $rows) {
    $stmt = $pdo->prepare("INSERT INTO table_name (col1, col2) VALUES (?, ?)");
    foreach ($rows as $row) {
        $stmt->execute($row);
    }
}
?>
```

4. PHP/error_handling_logging.php

```
<?php
$json = '{"name": "John"}';
$data = json_decode($json, true);

$name = $data['name'] ?? 'Unknown';
$age = $data['age'] ?? 0;

if (json_last_error() !== JSON_ERROR_NONE) {
    error_log("JSON Decode Error: " . json_last_error_msg());
}
?>
```

5. PHP/async_processing.php

```
<?php
exec("php process_file.php > /dev/null 2>&1 &");
?>
```

6. PHP/camel_case_conversion.php

```
<?php
function camelToSentence($string) {
    return strtolower(preg_replace('/([a-z])([A-Z])/', '$1 $2', $string));
}

echo camelToSentence("camelCaseString");
?>
```

7. PHP/session_management.php

```
<?php
ini_set('session.save_handler', 'redis');
ini_set('session.save_path', 'tcp://127.0.0.1:6379');
session_start();
?>
```

8. PHP/secure_file_upload.php

```
<?php
function uploadImage($file) {
    $allowedTypes = ['image/jpeg', 'image/png'];

    if (!in_array($file['type'], $allowedTypes)) {
        die("Invalid file type");
    }

    $uploadDir = "uploads/";
    $fileName = basename($file["name"]);
    $targetFilePath = $uploadDir . $fileName;

    move_uploaded_file($file["tmp_name"], $targetFilePath);
}
?>
```