

## Secure File Upload System - Test Document

### Functionality Testing:

#### 1. File Type Validation:

- Test Case 1: Upload a valid image file with the extension `.jpg`.
  - Expected Result: File is successfully uploaded.
- Test Case 2: Attempt to upload a file with an invalid extension, e.g., `.exe`.
  - Expected Result: The system rejects the upload with an error message.

#### 2. File Size Limit:

- Test Case 1: Upload a file with a size less than the maximum limit (e.g., 4MB).
  - Expected Result: File is successfully uploaded.
- Test Case 2: Attempt to upload a file exceeding the maximum size limit (e.g., 6MB).
  - Expected Result: The system rejects the upload with an error message.

#### 3. File Name Sanitization:

- Test Case 1: Upload a file with a valid name, including alphanumeric characters and underscores.
  - Expected Result: File is successfully uploaded with the same name.
- Test Case 2: Attempt to upload a file with a potentially harmful name containing special characters.
  - Expected Result: The system sanitizes the file name and replaces harmful characters, then successfully uploads the file.

#### 4. File Upload Directory:

- Test Case 1: Upload a file and check its storage location on the server.
  - Expected Result: The file is stored in a secure directory located outside the web server's root directory.
- Test Case 2: Attempt to access an uploaded file directly through the web server.
  - Expected Result: The server denies access to the file, preventing execution as a script.

#### 5. Logging:

- Test Case 1: Upload a file and check the system logs for the uploader's IP address, file name, and upload timestamp.
  - Expected Result: The log entry contains the necessary information.
- Test Case 2: Upload a file with an invalid extension (e.g., `.exe`), triggering a rejection.

- Expected Result: The system logs the suspicious or rejected upload with details.

#### 6. Security Headers:

- Test Case 1: Attempt to inject malicious code in a file name and upload it.
  - Expected Result: The server sends appropriate security headers in the HTTP response, mitigating potential XSS attacks.

#### 7. User Authentication:

- Test Case 1: Attempt to upload a file without being authenticated as a user.
  - Expected Result: The system restricts file uploads to authorized users only and prompts for authentication.