### ****1. Risk Management (risk\_management.txt)****

**Risk Management Plan**

This section outlines potential problems and how to handle them when using the **File Organizer Script**.

#### ****Possible Risks and Solutions****

|  |
| --- |
| **Risk** **What Could Happen** **How to Prevent/Handle** |

|  |  |  |
| --- | --- | --- |
| **Permission Issues** | Some files might not be able to be moved because of restricted permissions. | Use sudo to ensure proper permissions or let the user know to fix permissions. |
| **Unknown File Types** | Files with extensions not listed in the script might be missed. | Put these files in an "Others" folder or let users add new types to the script. |
| **Missing Files** | If a file is moved or deleted before the script runs, it won't be found. | Check if the file exists before trying to move it. |
| **Overwriting Files** | A file could be accidentally replaced by another with the same name. | Check for duplicate files before moving them and avoid overwriting. |
| **Wrong Directory Path** | If the user types the wrong directory, the script will fail. | Ask the user to check the path and make sure the directory exists before running. |

### ****2. Performance Testing (performance\_testing.txt)****

**Performance Testing Plan**

This section explains how to check if the script runs smoothly, even with a large number of files.

#### ****How to Test Performance****

1. **Execution Time**: Measure how long it takes for the script to complete when running on directories with different numbers of files.
2. **System Resources**: Check how much CPU and memory the script uses during its run to make sure the system isn't overloaded.

#### ****Tools to Use****

1. **time**: To check how long the script takes to run.

time ./file\_organizer.sh /path/to/directory

1. **top** or **htop**: To check the CPU and memory usage.

top

1. **iostat**: To check disk usage and I/O activity.

iostat

#### ****Test Examples****

**1. Test with a Few Files:**

1. Run the script on a small directory with only a few files.
2. Check the time it takes using time.
3. Check the CPU and memory usage with top.

**2. Test with Many Files:**

1. Run the script on a directory with many files (1000+).
2. Check the time it takes and how much system resources are used.

#### ****Expected Results****

1. **Small Directory**: The script should run quickly (a few seconds), using little system resources.
2. **Large Directory**: The script will take longer (maybe 30 seconds or more) and will use more CPU and memory.

#### ****Improvement Suggestions****

1. If the script is slow with a lot of files, consider breaking the task into smaller parts.
2. Look for ways to reduce the load on your system, like moving files in batches.