# Stage 2: Testing

## Testing Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| What was tested | Test Data | Expected output | Actual output | Revisions |
| **Check password function** | **Password:** Password!123 | True | True |  |
| **“** | **Password:**  password | False | True | Instead of multiple nested if statements, they are now replaced with if statements that check if the criteria is not met, then it will return false.  Update: Instead of returning false, it will add an error to a list. Once all of the checks have been made, it will return it. It the list is empty, it will then return true. |
| **“** | **Password­:**  password | [  "Password must contain one capital letter.",  "Password must contain a number.", "Password must contain one special character."  ] | [  "Password must contain one capital letter.",  "Password must contain a number.", "Password must contain one special character."  ] |  |
| **Check password range minimum** | **Password:**  1234567 | [  “Password must be at least 8 characters long.”,  “Password must contain one capital letter.”,  “Password must contain one special character.”  ] | [  “Password must be at least 8 characters long.”,  “Password must contain one capital letter.”,  “Password must contain one special character.”  ] |  |
| **“** | **Password:**  12345678 | [  “Password must contain one capital letter.”,  “Password must contain one special character.”  ] | [  “Password must contain one capital letter.”,  “Password must contain one special character.”  ] |  |
| **Check password range maximum** | **Password:**  Test!Test!Test!Test!Test!T | True | ["Password must be less than 25 characters long."] | Changed > to >= symbol as the negation of the maximum characters criteria would be as such. |
| **“** | **“** | True | True |  |
| **“** | **Password:**  Test!Test!Test!Test!Test! | ["Password must be less than 25 characters long."] | ["Password must be less than 25 characters long."] |  |
| **Gallery File Picker Input** | **File:**  flowers.png | Should upload | Uploads successfully |  |
| **“** | **File:**  words.txt | Should not be able to upload | Uploads successfully | Added an if statement to check the extension of the file being uploaded. If the extension is not one of the image file extensions, then it will reject the request. |
| **Gallery Upload then View** | **File:**  flowers.png | Image should be shown | Console error: <HTMLElement>.src cannot be set. | Backend was not returning the correct base64 string representation of the image, instead it was missing the metadata string which states what image type it is. Added a base64 prefix to the string to identify its type. |
| **“** | **“** | Image should be shown | Image is shown. |  |
| **Sorting shows correct output** | **Images:** [“flowers.png”, “ant.jpeg”, “scene.png”, “test.png”, “basketball.jpg”] | [“ant.jpeg”, “basketball.jpg”, “flowers.png”, ”scene.png”, “test.png”] | [“ant.jpeg”, “basketball.jpg”, “flowers.png”, ”scene.png”, “test.png”] |  |
| **Deselect all button** | **Selected:**  [“scene.png”, “test.png” | **Selected:** []  All select buttons go invisible | **Selected:** []  All select buttons go invisible |  |
| **Gallery upload with duplicate files** | **Stored files:**  [ant.jpeg, Flowers.png]  **File:**  Flowers.png | Show notification saying that there are duplicate files. | File is overwritten | ‘/uploadImage’ route now returns response code 300 if a duplicate file exists. |
| **“** | **“** | Shows notification saying that there are duplicate files. | Shows notification saying that there are duplicate files. |  |
| **Gallery upload with duplicate files** | **Stored files:**  [ant.jpeg, Flowers.png]  **File:**  Flowers.png  **Action:**  User clicks overwrite button | Original image should be replaced with the new image | New image is rotated 90 degrees with only some images. | Extract image metadata from the base64 string using Python module ‘exif’. Detect the orientation and check that if the orientation is 8, it will manipulate the image to rotate it 90 degrees anti-clockwise. |
| **“** | **“** | Original image should be replaced with the new image in the correct orientation | Original image is replaced with the new image in the correct orientation. |  |
| **Gallery items on click** | **Image:**  Flowers.png  **Action:**  User clicks image | Downsized image should be displayed until original quality image is fetched. | Downsized image is displayed and does not change. | ‘/getImage’ was not returning any image data. It now responds with the base64 string of the image data back to the web client. |
| **“** | **“** | Downsized image should be displayed until original quality image is fetched. | Downsized image shows for a split second and then changes to the original quality. |  |
| **Gallery multiple file upload** | **Files:** [“flowers.png”, “ant.jpeg”] | An upload notification is shown. Files are uploaded, then the notification changes to say that the files have been uploaded. | File picker does not allow multiple files | Changed a property on HTML <input> to allow multiple files to be inputted. |
| **“** | **“** | An upload notification is shown. Files are uploaded, then the notification changes to say that the files have been uploaded. | Only the first image sent through is uploaded to the database. | The upload request was being sent before all of the files were being read, due to JS asynchronous subroutines. Added a check after each is read to tally up and compare with the total file count before sending the upload request. |
| **“** | **“** | An upload notification is shown. Files are uploaded, then the notification changes to say that the files have been uploaded. | An upload notification is shown. Files are successfully uploaded, then the notification changes to say that the files have been uploaded. |  |
| **Sorting in Recently Deleted Page - Name** | **Files:**  [“script.txt”, “Powerpoint.pptx”, “Words.docx”, “Cheesecake recipe.doc”]  **Action:**  Click sort by name button | [“Cheesecake recipe.doc”,  “Powerpoint.pptx”, “script.txt”, “Words.docx”] | [“Cheesecake recipe.doc”,  “Powerpoint.pptx”, “script.txt”, “Words.docx”] |  |
| **Sorting in Recently Deleted Page – Date Uploaded** | **Files:**  [“24/08/4”, “23/09/13”, “23/02/25”, “24/01/12”]  **Action:**  Click sort by date button | [“23/02/25”, “23/09/13”, “24/01/12”, “24/08/4”] | [“24/08/4”, “24/01/12”, “23/09/13”, “23/02/25”] | The original function was sorting by the file name. Added a check for parameter ‘sortBy’. If it was date\_uploaded, then it will parse the date strings to Unix time and then compare the integers. |
| **“** | **“** | [“23/02/25”, “23/09/13”, “24/01/12”, “24/08/4”] | [“23/02/25”, “23/09/13”, “24/01/12”, “24/08/4”] |  |
| **Archiving files** | **Selected:**  [“flowers.png”]  **Action:** Click archive button | Property ‘archived’ is added to the file and image is deleted from the gallery.  File should show up in recently deleted | Archived is added to file properties and then the image is removed.  It can be viewed in the recently deleted section. |  |
| **Restoring files** | **Selected:** [“flowers.png”]  **Action:** Click restore button | ‘archived’ property should be removed and the file should be removed from the page. | ‘archived’ property is removed and the file cannot be viewed in recently deleted. |  |
| **Restoring multiple files** | **Selected:** [“flowers.png”, “ant.jpeg”, “cake.png”]  **Action:** Click restore button | The three files with the corresponding names are removed from the page and the ‘archived’ property is removed from all three files. | Only ‘flowers.png’ is unarchived. | The restore request was only restoring the first file in the list of file names given. Added a for loop to iterate over each name and perform the restore function. |
| **“** | **“** | The three files with the corresponding names are removed from the page and the ‘archived’ property is removed from all three files. | The three files with the corresponding names are removed from the page and the ‘archived’ property is removed from all three files. |  |
| **File manager upload images** | **Files:** [“flowers.png”] | **Notification:**  Image files can only be uploaded in 'All Files' or the 'Gallery': flowers.png | Image is uploaded when it should not | ‘/uploadFile’ route now returns with a response code to tell the web client that the image cannot be uploaded on this page. |
| **“** | **“** | **Notification:**  Image files can only be uploaded in 'All Files' or the 'Gallery': flowers.png | **Notification:**  Image files can only be uploaded in 'All Files' or the 'Gallery': flowers.png |  |
| **File manager upload image and file** | **Files:** [“words.txt”, “ant.jpeg”] | **Notification:**  Image files can only be uploaded in 'All Files' or the 'Gallery': ant.jpeg | **Notification:**  Image files can only be uploaded in 'All Files' or the 'Gallery': ant.jpeg |  |
| **File manager duplicate files** | **File store:** [“words.txt”, “test.docx”, “Maths.pdf”]  **Upload files:** [“Maths.pdf”] | **Notification:**  Files are conflicting with existing files, do you want to overwrite them?  Maths.pdf  Yes - No | All three files are uploaded, Maths.pdf replaces the original Maths.pdf file. | ‘/uploadFile’ returns response code 300 for a duplicate file. The web client catches that and stores it in a list ‘duplicateFiles’. After all the requests have been made, if the length of duplicateFiles is greater than 0, then it will raise a notification that there are conflicting files. |
| **“** | **“** | **Notification:**  Files are conflicting with existing files, do you want to overwrite them?  Maths.pdf  Yes - No | **Notification:**  Files are conflicting with existing files, do you want to overwrite them?  Maths.pdf  Yes - No |  |
| **File manager duplicate files** | **File store:**  [“words.txt”, “test.docx”, “Maths.pdf”]  **Upload files:** [“Maths.pdf”]  **Action:**  Click Yes button | Maths.pdf should be replaced with new uploaded file. | Maths.pdf is replaced with new uploaded file. |  |
| **File manager duplicate files** | **File store:** [“words.txt”, “test.docx”, “Maths.pdf”]  **Upload files:** [“Maths.pdf”]  **Action:**  Click No button | Only words.txt and test.docx are uploaded. | Only words.txt and test.docx are uploaded. |  |
| **File manager with archived files** | **Files:**  [“words.txt”, “test.docx”, “Maths.pdf”]  **Action:**  Archive test.docx then reload page | Only words.txt and Maths.pdf should show | All three files are still shown. | Fixed an issue in ImageLoader.load\_thumbnails which takes in archived as a Boolean parameter yet does not check it. It will now filter out the archived files when shown in the file manager. |
| **“** | **“** | Only words.txt and Maths.pdf should show | Only words.txt and Maths.pdf are shown |  |
| **File manager preview - .exe** | **Action:**  Click  **“**paint.net Installer.exe” | “File format is not supported for preview.” | “File format is not supported for preview.” |  |
| **File manager preview - code** | **Action:** Clickscript.py | Code preview should show up with script.py contents. | **Preview text**: “File format is not supported for preview.” | When requesting a get for a file’s contents, if it is in the list of code file extensions, it will return the contents of the file by converting its value from base64 to plain text. Then the text is shown using prism.js’ code preview. |
| **“** | **“** | Code preview should show up with script.py contents. | Code preview shows up with script.py contents. |  |
| **File manager preview - code** | **Action:**  Clickscreensaver.ahk | Code preview should show up with screensaver.ahk contents. | Code preview shows up with screensaver.ahk contents. |  |
| **File manager preview - text** | **Action:**  Clickwords.txt | Text preview should show up with the contents of words.txt. | Text preview is populated with the contents of words.txt successfully. |  |
| **Gallery preview GIF** | **Action:**  Click giphy.gif | Image viewer should show an animated gif image | Image viewer shows an animated gif preview. |  |
| **All files preview – image** | **Action:**  Click flowers.png | Image preview should show flowers.png | Image preview shows a file icon which is usually for non-image formats. | Added an exclusion for files with the image file extensions where if they pass, then it returns the base64 string representation of the image file back to the web client. |
| **“** | **“** | Image preview should show flowers.png | Image preview is showing the image ‘flowers.png’. |  |
| **All files preview – GIF** | **Action:**  Click giphy.gif | Image preview should show the animated GIF file ‘giphy.gif’ | Image preview shows the animated GIF file ‘giphy.gif’. |  |
| **All files – sort by date taken** | **Action:**  Click date header 3 times until it reads “Date Taken” | Images should be sorted by their date taken, while files will be at the bottom | Images are sorted by their date taken. However, files are randomly in between the sorted image files. | Files without the property “date\_taken” will now be represented with a date of 0 (in Unix time). |
| **“** | **“** | Images should be sorted by their date taken, while files will be at the bottom. | Images are sorted by their date taken and below them are the rest of the files. |  |
| **Context menu – Select** | **Action:**  Right click “flowers.png” and click select | Only “flowers.png” should be selected. The file selection panel should show up. | “flowers.png” is not selected. The file selection panel still shows up. | Added event listener such that clicking the select button in the context menu will call the select event. |
| **“** | **“** | Only “flowers.png” should be selected. The file selection panel should show up. | Only “flowers.png” is selected. The file selection panel does show up. |  |
| **Context menu – Copy** | **Selected:** []  **Action:**  Right click “flowers.png” and click copy | User’s clipboard should be populated with “flowers.png” | User’s clipboard text is now “flowers.png” |  |
| **Context menu – copy multiple files** | **Selected:** [flowers.png, ant.jpeg, words.txt]  **Action:**  Right click “words.txt” and click copy | User’s clipboard should be populated with “flowers.png, ant.jpeg, words.txt” | User’s clipboard text is only “words.txt” | If the list ‘selected’ is not empty, then the copy event will populate the user’s clipboard with the selected items separated by commas. |
| **“** | **“** | User’s clipboard should be populated with “flowers.png, ant.jpeg, words.txt” | User’s clipboard is populated with “flowers.png, ant.jpeg, words.txt” |  |
| **Context menu – paste single file** | **Clipboard:** flowers.png  **Action:** rightclick and click paste | File manager should gain another file with the same name but with “ – Copy” concatenated on it | File manager gains another file with copied name |  |
| **Context menu – paste multiple files** | **Clipboard:** flowers.png, ant.jpeg  **Action:** rightclick and click paste | Creates “flowers – Copy.png” and “ant – Copy.jpeg” | Creates “flowers – Copy.png” and “ant – Copy.jpeg” |  |
| **Download button works** | **Selected:** [flowers.png]  **Action:** Click download button and confirm | Download should start on user’s web client with zipped folder containing flowers.png | No download starts on user’s web client. | In JavaScript, it will run “window.location.href = zipPath;” which starts the download after the download request is made. |
| **“** | **“** | Download should start on user’s web client with zipped folder containing flowers.png | Download started with zipped folder. Zipped folder contains one file named flowers.png. Opening flowers.png, it is the same file. Successful. |  |
| **Download button works – multiple files** | **Selected:** [flowers.jpeg, SAT GANTT.pdf, Glossary of Terms.docx]  **Action:** Click download button and confirm | Download should start and the zipped folder will contain three files, flowers.jpeg, SAT GANTT.pdf, and Glossary of Terms.docx. |  |  |
| **Downloaded zipped folder deletes** | **Selected:** [flowers.jpeg, SAT GANTT.pdf, Glossary of Terms.docx]  **Action:** Click download button and confirm | After 10 seconds, the zipped folder deletes. | The zipped folder does not delete after 10 seconds. | Created a function called ‘confirmDownloadRequest’ which fetches ‘/downloadFiles’ with the zipped folder path in the request data. The backend detects this and deletes the zipped folder accordingly. |
| **“** | **“** | After 10 seconds, the zipped folder deletes. | After 10 seconds, the zipped folder deletes. |  |
| **Context menu – Archive single file** | **Selected:** flowers.png  **Action:** rightclick, click archive and confirm | File removed from file manager page and archived property added to file in database. | File removed on file manager page and archived property added to file in database. |  |
| **Context menu – Archive multiple files** | **Selected:** flowers.png, ant.jpeg, words.txt  **Action:** rightclick, click archive and confirm | All three file rows removed from file manager page and the archived property is added to the files in the database. | All three file rows removed from file manager page and the archived property is added to the files in the database. |  |
| **Context menu – Delete multiple files in recently deleted** | **Selected:** flowers.png, ant.jpeg, words.txt  **Action:** rightclick, click delete and confirm | All three file rows removed from file manager page and files are permanently deleted from the database. | All three file rows removed from file manager page and files are permanently deleted from the database. |  |
| **Image pixelated setting** | **Profile:** Image quality -> pixelated  **Action:** View flowers.png in gallery | Image preview is pixelated after zooming in. | Image preview is pixelated after zooming in. |  |
| **Image pixelated setting** | **Profile:** Image quality -> normal  **Action:** View flowers.png in gallery | Image preview is blurry after zooming in. | Image preview is pixelated after zooming in. | Added an if statement to check if the user settings has enable the setting for image pixelated quality. If so, it will set the style of all images to be ‘image-rendering: pixelated;’. |
| **“** | **“** | Image preview is blurry after zooming in. | Image preview is blurry after zooming in. |  |
| **Total file storage calculation** | **Files:** [  ..: 5.8MB, ..: 6.5MB, ..: 97.9MB]  **Action:** View profile | 110.2MB / 10GB | 110.2MB / 10GB |  |
| **Image zoom speed setting** | **Profile:** Image zoom speed -> slow  **Action:** View flowers.png in gallery | Image zooms quite slow | Image does not zoom at all | When setting the state of the setting slider/switch in profile settings, it checks if the value is not true, but this also holds true for values which are zero (which is what happened in this case). |
| **“** | **“** | Image zoomed quite slow | Image preview zooms slowly |  |
| **Image zoom speed setting** | **Profile:** Image zoom speed -> fast  **Action:** View flowers.png in gallery | Image zooms quite fast comparatively | Image zooms quite fast comparatively |  |
| **Quick Sort algorithm computation** | **Files:** [Glossary of Terms.docs, SAT GANTT.pdf, IMG\_9839.jpg, flowers.jpg, mindmap.pdf, 2020-janson-units-3-4.pdf]  **Action:** Sort by name | [2020-janson-units-3-4.pdf, Glossary of Terms.docx, IMG\_9839.JPG, SAT GANTT.pdf, flowers.JPG, mindmap.pdf] | [flowers.JPG, IMG\_9839.JPG, Glossary of Terms.docx, SAT GANTT.pdf, 2020-janson-units-3-4.pdf, mindmap.pdf] | The loop in the partition method where it would calculate the index smallest of the value in the array was not working. Rewrote the code for it. |
| **“** | **“** | [2020-janson-units-3-4.pdf, Glossary of Terms.docx, IMG\_9839.JPG, SAT GANTT.pdf, flowers.JPG, mindmap.pdf] | [2020-janson-units-3-4.pdf, Glossary of Terms.docx, IMG\_9839.JPG, SAT GANTT.pdf, flowers.JPG, mindmap.pdf] |  |
| **Quick Sort algorithm computation** | **Files:** [Glossary of Terms.docs, SAT GANTT.pdf, IMG\_9839.jpg, flowers.jpg, mindmap.pdf, 2020-janson-units-3-4.pdf]  **Action:**  Sort by date uploaded | [Glossary of Terms.docx, mindmap.pdf, SAT GANTT.pdf, 2020-janson-units-3-4.pdf, flowers.JPG, IMG\_9839.JPG] | [Glossary of Terms.docx, mindmap.pdf, SAT GANTT.pdf, 2020-janson-units-3-4.pdf, flowers.JPG, IMG\_9839.JPG] |  |
| **Quick Sort algorithm computation** | **Files:** [Glossary of Terms.docs, SAT GANTT.pdf, IMG\_9839.jpg, flowers.jpg, mindmap.pdf, 2020-janson-units-3-4.pdf]  **Action:**  Sort by type | [flowers.JPG, IMG\_9839.JPG, Glossary of Terms.docx, SAT GANTT.pdf, 2020-janson-units-3-4.pdf, mindmap.pdf] | [flowers.JPG, IMG\_9839.JPG, Glossary of Terms.docx, SAT GANTT.pdf, 2020-janson-units-3-4.pdf, mindmap.pdf] |  |
| **Quick Sort algorithm computation** | **Files:** [mindmap.pdf, flowers.JPG, SAT GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, 2020-janson-units-3-4.pdf]  **Action:** Sort by size descending | [2020-janson-units-3-4.pdf, mindmap.pdf, flowers.JPG, IMG\_9839.JPG, SAT GANTT.pdf, Glossary of Terms.docx] | [2020-janson-units-3-4.pdf, mindmap.pdf, flowers.JPG, IMG\_9839.JPG, SAT GANTT.pdf, Glossary of Terms.docx] |  |
| **Reverse function** | [2020-janson-units-3-4.pdf, Glossary of Terms.docx, IMG\_9839.JPG, SAT GANTT.pdf, flowers.JPG, mindmap.pdf] | [mindmap.pdf, flowers.JPG, SAT GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, 2020-janson-units-3-4.pdf] | [flowers.JPG, GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, SAT, 2020-janson-units-3-4.pdf, mindmap.pdf] | The indices that were being passed into the swap function were wrong, it must be a two indices which mirror each other according to the size of the array. |
| **“** | “ | [mindmap.pdf, flowers.JPG, SAT GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, 2020-janson-units-3-4.pdf] | [mindmap.pdf, flowers.JPG, SAT GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, 2020-janson-units-3-4.pdf] |  |
| **Binary search** | **Files:**  [mindmap.pdf, flowers.JPG, SAT GANTT.pdf, IMG\_9839.JPG, Glossary of Terms.docx, 2020-janson-units-3-4.pdf]  **Item:**  flower.JPG | **Index:** 1  <File Object 2> | None | The algorithm was originally discarding the half in which flowers.JPG was in. This was because I would make all of the file names lowercase first to make comparisons easily. However, this meant that the correctly sorted order was unsorted when it came to filenames with capital letters. Therefore, I made the parameter ‘name’ case-sensitive to avoid this problem. |
| **“** | **“** | **Index:** 1  <File Object 2> | **Index:** 1  <File Object 2> |  |
| **Logged out user** | **Action:**  Click logout button | User should not be able to access the file and gallery routes. | User cannot access file and image routes anymore. |  |
| **Previewing files in one click setting** | **Profile:** One click preview-> No  **Action:** View main.py in file manager | Preview button should appear at the top right hand side of the page. No preview should be shown. | Code preview is shown. | The setting for one click preview was not correctly loaded in on window initialisation. Fixed this and added an if statement to check if the setting is disabled, then it will not preview the file immediately upon clicking. |
| **“** | **“** | Preview button should appear at the top right hand side of the page. No preview should be shown. | Preview button shows up and no preview is shown. |  |
| **Previewing files in one click setting** | **Profile:** One click preview-> Yes  **Action:** View main.py in file manager | Code preview should be shown immediately on click | Code preview is shown immediately. |  |