File Selection Test Cases

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC001 |
| **Test Case Description** | File Selection – Select the CSV file  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. |
| **Test Data** | The CSV file. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Click the ‘Choose File’ button at the top left to select the CSV file. |
| **Expected Results** | At Step 2:   1. Only files that are in .csv format are selectable.   After Step 2:   1. The 'No file chosen' text in the file input button should be replaced with the selected file name, 'data.csv'. 2. No data should be displayed. 3. The progress bar, pagination, and search textbox should remain hidden. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC002 |
| **Test Case Description** | File Selection – Cancel file selection  (Positive Test Case) |
| **Preconditions** | 1. The app must be running and working. |
| **Test Data** | N/A |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Click the ‘Choose File’ button at the top left to open file selection dialog. 3. Click the ‘Cancel’ button to exit the file selection dialog. |
| **Expected Results** | * + - 1. None of the UI elements, including the pagination, search textbox, progress bar, or table, should be displayed.       2. The "Choose File" button should remain in its default state. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

|  |  |
| --- | --- |
| **Test Case ID** | TC003 |
| **Test Case Description** | File Selection – Cancel file selection to reset the data  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. |
| **Test Data** | The CSV file. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Select the CSV file using the ‘Choose File’ button. 3. Click ‘UPLOAD’ button to read the data from the CSV file. 4. Wait for the data to appear. 5. Click the ‘Choose File’ button again to reopen the file selection dialog. 6. Click ‘Cancel’ to exit the file selection dialog. |
| **Expected Results** | After Step 4:   1. The progress bar, pagination, search textbox, and data in table format should appear.   After Step 6:   1. The progress bar, pagination, and search textbox should disappear. 2. The data in the table should be cleared, and a ‘No Records’ message should be shown. 3. The "Choose File" button should reset to its default state. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC004 |
| **Test Case Description** | File Selection – Select a file of a different type  (Negative Test Case) |
| **Preconditions** | 1. User must have at least one file that is not in .csv format. 2. The app must be running and working. |
| **Test Data** | A file that is not in .csv format. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Click the ‘Choose File’ button at the top left. 3. Select ‘All Files’ from the file type dropdown to view all available files. 4. Select a file that is not in .csv format. |
| **Expected Results** | 1. ‘No file chosen’ text in the file input button replaced by the selected file name. 2. An ‘Only CSV file is allowed.’ message should be displayed in a light-red alert card below the file selection box. 3. No data should be displayed. 4. The progress bar, pagination, and search textbox should not be rendered. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC005 |
| **Test Case Description** | File Selection – Select a non-CSV file while data is already displayed  (Negative Test Case) |
| **Preconditions** | * + - 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv>  1. User must have at least one file that is not in .csv format. 2. The app must be running and working |
| **Test Data** | The CSV file and a file that is not in .csv format |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Select the CSV file using the ‘Choose File’ button. 3. Click ‘UPLOAD’ button to read the data from the CSV file. 4. Wait for the data to appear. 5. Click the ‘Choose File’ button again. 6. Select another file that is not in .csv format. |
| **Expected Results** | After Step 4:   1. The progress bar, pagination, search textbox, and data in table format should appear.   After Step 6:   1. The progress bar, pagination, and search textbox should disappear. 2. The table data should be cleared, and a 'No Records' message should be displayed. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

File Upload Test Cases

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC006 |
| **Test Case Description** | File Upload – No file selected  (Positive Test Case) |
| **Preconditions** | 1. The app must be running and working. |
| **Test Data** | N/A |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Click ‘UPLOAD’ button when no file selected. |
| **Expected Results** | None of the UI elements (pagination, search textbox, progress bar, or table) should be displayed. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC007 |
| **Test Case Description** | File Upload – Valid CSV File  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. |
| **Test Data** | The CSV file. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Select the CSV file using the ‘Choose File’ button. 3. Click the ‘UPLOAD’ button to initiate data upload. 4. Observe the progress bar behaviour. 5. Wait for the progress bar to reach 100%. 6. Verify that the data from the CSV file is displayed in a table with correct headers. |
| **Expected Results** | At Step 4:   1. The progress bar with animation indicates the upload is in progress. 2. A number in percentage next to the progress bar indicates the upload progress. 3. The progress bar may briefly display '100%' if the upload speed is very fast.   After Step 5:   1. The "No Records" message disappears. 2. The table displays the CSV data with correct headers. 3. Pagination and search textbox appear. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC008 |
| **Test Case Description** | File Upload – Invalid file format  (Negative Test Case) |
| **Preconditions** | 1. User must have at least one file that is not in .csv format. 2. The app must be running and working |
| **Test Data** | A file that is not in .csv format. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Click the ‘Choose File’ button. 3. Select ‘All Files’ from the file type dropdown to view all available files. 4. Select a file that is not in .csv format. 5. Click the ‘UPLOAD’ button. |
| **Expected Results** | The system should not process or display any data from the invalid file. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC009 |
| **Test Case Description** | File Upload – Upload a non-CSV file while data is already displayed  (Negative Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. User must have at least one file that is not in .csv format. 3. The app must be running and working |
| **Test Data** | The CSV file and a file that is not in .csv format. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Select the CSV file using the ‘Choose File’ button. 3. Click the ‘UPLOAD’ button to initiate data upload. 4. Wait for data to display. 5. Click the ‘Choose File’ button again to open the file selection dialog. 6. Select ‘All Files’ from the file type dropdown to view all available files. 7. Select a file that is not in .csv format. 8. Click the ‘UPLOAD’ button. |
| **Expected Results** | After Step 7:   * + - 1. An ‘Only CSV file is allowed.’ Message is shown.       2. The 'No file chosen' text in the file input button should be replaced with the selected file name.       3. The table data should be cleared, and a 'No Records' message should be displayed.       4. The progress bar, pagination, and search textbox should disappear.   After Step 8:  The system should not display any data from the invalid file. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

Pagination Test Cases

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC010 |
| **Test Case Description** | Pagination – With the data.csv provided  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. |
| **Test Data** | The CSV file. |
| **Test Steps** | 1. Use npm start to run the application at localhost:3000. 2. Select the CSV file using the ‘Choose File’ button. 3. Click the ‘UPLOAD’ button to initiate data upload. 4. Verify that the data from the CSV file is displayed in a table with correct headers. 5. Click any page number, or the Previous and Next buttons in the pagination controls. |
| **Expected Results** | At Step 4:   1. The pagination should appear above the data table at the right. 2. Page 1 should be selected by default, and there should be a total of 100 pages.   At Step 5:   1. The data should be displayed according to the page or buttons clicked. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

Search Function Test Cases

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC011 |
| **Test Case Description** | Search Function – Records found  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. 3. The data in CSV is retrieved successfully and displayed in table. |
| **Test Data** | Keyword to search: “Jeremy”, “23”, “voluptatem” |
| **Test Steps** | 1. Copy & paste “Jeremy” in the search textbox. 2. Change capital “J” to “j”. 3. Highlight “jeremy” in the textbox and type “23”. 4. Go to page 2. 5. Copy or key in the keyword “voluptatem” 6. Delete all the characters in the search textbox |
| **Expected Results** | After Step 1:   * + - 1. Only 1 record (id 46) shows because the email includes the keyword “Jeremy” **[email search test]**       2. The pagination controls should display only one page.   After Step 2:   * + - 1. Only 1 record (id 283) should be displayed **[case sensitive test]**   After Step 3:   * + - 1. There is 3 pages of search results that matches the keyword “23” **[postId and id search test]**   After Step 5:   * + - 1. There are 40 pages of record found **[name and body search test]**       2. The active page should reset to page 1. **[pagination reset test]** |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |

| **Field** | **Details** |
| --- | --- |
| **Test Case ID** | TC011 |
| **Test Case Description** | Search Function – No records found  (Positive Test Case) |
| **Preconditions** | 1. User must download the CSV from <https://github.com/tqwdan82/Technical_Assessment/blob/main/data.csv> 2. The app must be running and working. 3. The data in CSV is retrieved successfully and displayed in table. |
| **Test Data** | Keyword to search: “asdfa” or any random text |
| **Test Steps** | 1. Key in any random text such as “asdfa” in the search textbox. |
| **Expected Results** | * + - 1. An 'No Records' message should appear when no record matches the keyword.       2. Pagination controls should be hidden when no records are found. |
| **Actual Results** |  |
| **Pass/Fail Status** |  |
| **Remarks/Comments** |  |