<!-- Filename: QA_Checklist.md -->

<!-- Last Edited: 2025-10-24 -->

JobAssist Application Quality Assurance Checklist

This checklist outlines the mandatory tests to be executed across the application's three main interfaces following any code deployment or significant change.

1. Dashboard & Filtering (index.html)

Test Case	Feature/Endpoint	Steps	Expected Result	Pass/Fail
1.1 Initial Load	GET /api/companies	Load index.html.	All standardized company profiles load and display in the table.	[ ]
1.2 Target Filter	Filter Control	Select **Target Interest: Yes** from the dropdown.	Only companies flagged as target_interest: true are visible.	[ ]
1.3 Name Search	Search Input	Enter a partial clean name (e.g., "Micr").	The table instantly filters to show only matching company names (e.g., "Microsoft").	[ ]
1.4 Column Sorting	Data Manipulation	Click the **Clean Company Name** header,	The data correctly sorts alphabetically, then	[ ]

		then click the **Annual Revenue** header.	numerically (Asc/Desc).	
1.5 Navigation	Header Link	Click the link/button to the Management Tool for an arbitrary company.	User is redirected to management.html with the correct companyId in the URL.	[ ]

2. Company Cleanup Interface (cleanup.html)

Test Case	Feature/Endpoint	Steps	Expected Result	Pass/Fail
2.1 Raw List Load	GET /api/unmapped_list	Load cleanup.html.	The left sidebar populates with a list of unique unmapped raw company names.	[ ]
2.2 Select Name	selectRawName()	Click an unmapped raw name in the list.	The name appears in the central #rawNameDisplay; all action panels become visible.	[ ]
2.3 Suggestion Search	GET /api/suggestions	Type a partial name (e.g., "Goog") into the search box under "Map to Existing."	A dropdown list of existing clean company names appears.	[ ]

2.4 Map to Existing	POST /api/map_company (action: map)	Select an existing company from suggestions and click **Map & Standardize**.	Success message displays, the raw name disappears from the list, and the list refreshes. (Data check: company_name e_mapping record created.)	[ ]
2.5 Create New Profile	POST /api/map_company (action: map)	Enter a unique name into **New Clean Name** input and click **Create New Profile & Map**.	Success message displays, the raw name disappears from the list, and the list refreshes. (Data check: New companies record created and mapped.)	[ ]
2.6 Skip & Self-Map	POST /api/map_company (action: skip)	Click the **Skip for Now** button.	Success message displays, the raw name disappears, a new companies profile is created, and the target_interest flag is set to **TRUE**.	[ ]

2.7 Batch Process	POST /api/map/batch (or multiple POST /api/map_company)	Ensure several names are unmapped, then click **Batch Map All Remaining Names**.	All remaining raw names are processed, the list becomes empty, and a successful batch completion message is shown.	[ ]

3. Company Management Tool (management.html)

Test Case	Feature/Endpoint	Steps	Expected Result	Pass/Fail
3.1 Company List	GET /api/companies	Load management.html.	The left list of clean company profiles populates correctly.	[ ]
3.2 Detail Load	GET /api/companies/{id}	Click a company name in the left list.	All three central panels (Profile Enrichment, Raw Names, Contacts) populate with the correct data for that company.	[ ]
3.3 Profile Update	PUT /api/companies/{id}	Edit a non-critical field (e.g., Headquarters or Internal Notes). Click **Save Changes**.	A success message displays, and if you refresh the page or re-select the company, the changes	[ ]

			persist.	
3.4 Target Toggle	PUT /api/companies /{id}	Toggle the **Target Interest** flag. Click **Save Changes**.	The change is reflected immediately and persists on reload.	[ ]
3.5 Linked Data	Detail Logic	Verify that the **Research & Networking Data** column shows the expected raw name variations and contacts/URLs associated with the selected company.	The data in Column 3 is accurate according to the database mappings.	[ ]

4. Code Quality & UI Consistency (Mandatory Checks)

Test Case	Component	Steps	Expected Result	Pass/Fail
4.1 File Header	All Files (HTML, JS, Py)	Review the top of the file source code.	Each file must contain a comment with the file name (e.g., app.py) and the date it was last edited (e.g., 2025-10-24).	[ ]
4.2 Global Navigation	All HTML Pages	Navigate between index.html, cleanup.html,	A consistent, functional navigation bar (Header) must	[ ]

| | | and management.html. | be present on all pages, linking to the others. | |

5. API and Data Integrity

Test Case	Feature/Endpoint	Steps	Expected Result	Pass/Fail
5.1 ID Consistency	All Endpoints	Map a raw name (e.g., "Foo Ltd") to an existing company (ID 10).	The company_name_mapping table must show raw_name: "Foo Ltd", company_id: 10.	[ ]
5.2 Data Types	PUT /api/companies/{id}	Attempt to save an update with a non-numeric value in size_employees or annual_revenue.	The API should return a 500 or 400 error and not corrupt the database fields.	[ ]
5.3 Error Handling	Any API Call	Simulate a failed database connection during any operation.	The frontend displays a user-friendly error message, and the backend logs the specific psycopg2.Error.	[ ]
5.4 Missing	Detail Logic	Verify a	All three	[ ]

Contacts		company profile with no associated contacts still loads correctly.	columns load, and the **Linked Contacts** panel displays a "No contacts found" message (no crashes).	