### **Detailed Function and API Analysis**

This table breaks down each function in the provided Streamlit application code, from high-level UI sections to low-level utility and API-calling functions.

| **Front End Component** | **Location of the API/Function** | **API / Function Name** | **Input Parameters** | **Output [Datatype]** | **Agent/Tool Used** | **Dependencies** | **Comments** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **URL Normalization** (Internal) | client\_tab.py | normalize\_url | url: str | str | N/A | re | A utility function to clean and format URLs by adding https:// and .com if missing. |
| **RFI Document Upload** | client\_tab.py | save\_uploaded\_file\_and\_get\_path | uploaded\_file, logger, client\_enterprise\_name: str | str (file path) | N/A | os, set\_global\_message | Saves the uploaded file to a directory specific to the client's enterprise name and returns its path. |
| **Data Validation** (Internal) | client\_tab.py | validate\_client\_mandatory\_fields | None | bool | N/A | client\_state\_manager, set\_global\_message | Checks if all mandatory fields in the application state have been filled out. |
| **Client Name Input** | client\_tab.py | render\_client\_name\_section | logger, client\_data, is\_locked: bool | str (client name) | N/A | st, client\_state\_manager, set\_global\_message | Renders the text input for the client's name and the "Find Website" button. |
| **Find Website Button** | ./client\_utils | get\_urls\_list | client\_enterprise\_name: str | List[str] | Web Search Agent | client\_state\_manager | *Called by render\_client\_name\_section*. Searches the web for website URLs matching the company name. |
| **Client Website Dropdown** | client\_tab.py | render\_client\_website\_section | logger, client\_data, is\_locked: bool | None | N/A | st, client\_state\_manager, normalize\_url, set\_global\_message | Renders the dropdown for website selection, a refresh button, and the "Get Details" button. |
| **Get Website Details Button** | ./client\_utils | get\_scraped\_data | client\_website\_url: str | Custom Object | Website Scraping Agent | client\_state\_manager | *Called by render\_client\_website\_section*. Scrapes the selected website to extract company details. |
| **Layout: Name & Website** | client\_tab.py | render\_first\_section | logger, client\_data, is\_locked: bool | str (client name) | N/A | st, render\_client\_name\_section, render\_client\_website\_section | A layout function that arranges the client name and website sections into two columns. |
| **Enterprise Details Text Area** | client\_tab.py | enterprise\_content | logger, client\_data, is\_locked: bool | bool (name provided) | N/A | st, client\_state\_manager, set\_global\_message | Renders the text area where scraped or manually entered enterprise details are displayed. |
| **RFI Document Uploader** | client\_tab.py | doc\_upload\_section | logger, client\_data, is\_locked: bool | None | N/A | st, save\_uploaded\_file\_and\_get\_path, set\_global\_message | Renders the file uploader component and the "Get pain points" button. |
| **Analyze RFI Document Button** | ./client\_utils | get\_pain\_points | file\_path: str, client\_enterprise\_name: str | Dict | Document Analysis Agent | client\_state\_manager | *Called by doc\_upload\_section*. Analyzes the uploaded RFI document to extract key pain points. |
| **Layout: Upload & Details** | client\_tab.py | render\_second\_section | logger, client\_data, is\_locked: bool | bool (name provided) | N/A | st, doc\_upload\_section, enterprise\_content | A layout function that arranges the document upload and enterprise details sections into two columns. |
| **Client Requirements Text Area** | client\_tab.py | render\_client\_requirements\_section | logger, client\_data, is\_locked: bool | None | N/A | st, client\_state\_manager, set\_global\_message | Renders the primary text area for capturing the client's requirements. |
| **Client Pain Points List** | client\_tab.py | render\_client\_pain\_points\_section | logger, client\_data, is\_locked: bool | None | N/A | st, client\_state\_manager, set\_global\_message | Renders the interactive list of pain points. Users can click ➕/❌ to add/remove them from the requirements text area. |
| **Layout: Requirements & Pain Points** | client\_tab.py | render\_third\_section | logger, client\_data, is\_locked: bool | None | N/A | st, render\_client\_requirements\_section, render\_client\_pain\_points\_section | A layout function that arranges the requirements and pain points sections into two columns. |
| **SPOC Name Input** | client\_tab.py | render\_spoc\_name\_section | logger, client\_data, is\_locked: bool | str (SPOC name) | N/A | st, client\_state\_manager, set\_global\_message | Renders the text input for the SPOC's name and the "Get LinkedIn" button. |
| **Get LinkedIn Profile Button** | Search.Linkedin.linkedin\_serp | get\_linkedin | spoc\_name: str | Dict or List | LinkedIn Search Agent | client\_state\_manager | *Called by render\_spoc\_name\_section*. Searches LinkedIn for profiles matching the SPOC's name. |
| **SPOC LinkedIn Profile Dropdown** | client\_tab.py | render\_linkedin\_profile\_section | logger, client\_data, is\_locked: bool, spoc\_name: str | str (profile URL) | N/A | st, client\_state\_manager, set\_global\_message | Renders the dropdown to select a specific LinkedIn profile from the search results. |
| **Selected Profile Info Display** | client\_tab.py | render\_selected\_profile\_info | logger, client\_data, spoc\_name\_provided: bool, spoc\_linkedin\_profile: str | None | N/A | client\_state\_manager, set\_global\_message | Displays info from the selected profile and dynamically updates the SPOC Role and Business Priorities lists based on the profile's data. |
| **Layout: SPOC Name & Profile** | client\_tab.py | render\_fourth\_section | logger, is\_locked: bool, client\_data | (bool, str) | N/A | st, render\_spoc\_name\_section, render\_linkedin\_profile\_section, render\_selected\_profile\_info | A layout function that arranges the SPOC name and LinkedIn profile sections. |
| **SPOC Role Dropdown** | client\_tab.py | render\_spoc\_role\_section | spoc\_name\_provided: bool, spoc\_linkedin\_profile: str, client\_data, logger, is\_locked: bool | None | N/A | st, client\_state\_manager, get\_roles\_list, set\_global\_message | Renders the dropdown for selecting the SPOC's role and the "Get Priorities" button. The role list is dynamically populated from a default list and the selected LinkedIn profile. |
| **Get Default Roles** | Recommendation.recommendation\_utils | get\_roles\_list | None | List[str] | N/A | N/A | *Called by render\_spoc\_role\_section*. Fetches a default, static list of business roles. |
| **Get Priorities Button** | Recommendation.recommendation\_utils | get\_ai\_business\_priorities | selected\_target\_role: str | List[Dict] | AI Recommendation Agent | st.session\_state | *Called by render\_spoc\_role\_section*. Fetches AI-suggested business priorities based on the selected role. |
| **SPOC Business Priorities List** | client\_tab.py | render\_spoc\_business\_priorities\_section | spoc\_name\_provided: bool, client\_data, logger, is\_locked: bool | None | N/A | st, client\_state\_manager, set\_global\_message | Renders the interactive list of business priorities. Users can click ➕/❌ to manage their selections. |
| **Layout: SPOC Role & Priorities** | client\_tab.py | render\_fifth\_section | spoc\_name\_provided: bool, spoc\_linkedin\_profile: str, client\_data, logger, is\_locked: bool | None | N/A | st, render\_spoc\_role\_section, render\_spoc\_business\_priorities\_section | A layout function that arranges the SPOC role and business priorities sections. |
| **Additional Requirements & Specs** | client\_tab.py | render\_sixth\_section | logger, is\_locked: bool, client\_data | None | N/A | st, client\_state\_manager | Renders two text areas for additional requirements and specifications, with interactive lists similar to the pain points section. |
| **Main Application Tab** | client\_tab.py | client\_tab | st, logger, is\_locked: bool | ClientTabState object | N/A | st, client\_state\_manager, all render\_\* functions | The main entry point function that orchestrates the rendering of all sections in the "Client" tab and manages the overall state. |
| **Tab Navigation Handler** | client\_tab.py | handle\_client\_tab\_navigation | None | None | N/A | st, client\_state\_manager | A utility function to reset temporary states when the user navigates to this tab. |
| **Tab Data Validator** | client\_tab.py | validate\_client\_tab\_data | None | (bool, str) | N/A | client\_state\_manager | A utility function to check if all necessary data on the tab has been provided. |
| **Tab Data Summary** | client\_tab.py | get\_client\_tab\_summary | None | Dict | N/A | client\_state\_manager | A utility function to export the current state of the tab as a dictionary. |
| **Clear Tab Data** | client\_tab.py | clear\_client\_tab\_data | None | bool | N/A | client\_state\_manager, set\_global\_message | A utility function to clear all data entered in the client tab. |