|  |  |  |
| --- | --- | --- |
| Function | Description | Functionality |
| showLoginPrompt() | Displays the login overlay prompt if the user is not logged in | - Sets an interval to show the login overlay every 5 seconds. - Listens for a click event on the overlay to hide it and stop the interval if clicked outside the login form. |
| toggleButtons() | Toggles the visibility and state of buttons based on the login status | - If logged in, hides the login/register buttons and enables the exportHistoryButton. - If not logged in, shows the login/register buttons and disables the exportHistoryButton. |
| enableHistoryScroll() | Enables scrolling in the history panel with a max height | - Sets the maximum height of the history panel to 50% of the window height. - Makes the history panel scrollable by applying overflowY: 'auto'. |
| checkLoginStatus() | Checks the user's login status by sending a request to the server | - Sends a fetch request to /check\_login to verify if the user is logged in. - Updates the UI (hides login overlay, shows username and session). - Calls toggleButtons() to update button states. - Displays login prompt if not logged in. - Handles errors if the request fails. |
| fetchSessions() | Fetches and populates the session dropdown with unique session names | - Fetches chat history from the server (/get\_chat\_history). - Extracts unique session names and populates them in a dropdown (session\_select). - Ensures that each session name appears only once using a Set. |
| showMessage() | Displays the message and any attached file in the chat interface | - Creates a new list item (<li>) to represent the message. - Adds an icon based on whether the message is incoming or outgoing. - Displays the text message and any attached file (image or regular file). - Scrolls the chatbox to show the new message. |
| sendMessageToAPI() | Sends the message and file to the backend API for processing | - Creates a FormData object and appends the message text and/or file (if available). - Sends the data to the server via a POST request to /process. - If successful, shows the response message from the API in the chat. - If there’s an error, displays a fallback message |
| sendMessage() | Sends a user’s message and/or file to the backend and shows it in the chat | - Collects the message input and file data from the user. - If there’s a message or file, it adds the message to the chat as an outgoing message. - Sends the message and file to the API using sendMessageToAPI(). - Clears the input field and resets the file input after sending. |
| resetFileInput() | Resets the file input and the file upload UI | - Clears the uploaded file from messageData.  - Resets the file preview and text in the file upload wrapper. - Sets the file input value back to an empty state. - Ensures that the file input is ready for new file selection. |
| Debounce() | Debounces a function call to prevent redundant fetches | - Prevents multiple quick calls to a function (e.g., loadChatHistory()). - Uses setTimeout to ensure that only one request is made after a delay, avoiding unnecessary rapid calls. |
| updateFileName() | Updates the file name in the UI and automatically triggers report generation | - Updates the displayed file name in the UI based on the file selected. - Immediately calls generateReport() to trigger report generation once a file is chosen. |
| generateReport() | Initiates the report generation process and shows a progress bar | - Checks if the user is logged in before allowing the report generation. - Sends the selected report file to the server via a POST request to /report. - Displays a progress bar, simulating upload progress. - If successful, allows the user to download the generated report file. - If there’s an error, shows an alert. |
| loadChatHistory() | Loads and displays the chat history dynamically | - Fetches the chat history from the server (/get\_chat\_history). - Displays the history by creating <li> elements for each entry with session, message, response, and timestamp. - Handles errors if history loading fails. |
| deleteHistory() | Deletes chat history based on the selected session name | - Sends a DELETE request to clear the chat history for the selected session (/clear\_chat\_history). - Prompts the user for confirmation before performing the deletion. - Reloads sessions and chat history after successful deletion. |
| validateFile(file) | Validates the file type and extension before uploading | - Checks if the file type and extension are supported. - Validates against a list of allowed MIME types and extensions (e.g., .jpg, .png, .txt, .pdf, etc.). - Alerts the user if the file type is not supported and returns false to prevent further processing. |
| logout() | Logs the user out by sending a request to the server endpoint /logout | - Sends a GET request to the server to log the user out. - If successful, alerts the user that they have been logged out, sets isLoggedIn to false, and calls toggleButtons(). - Redirects the user to the homepage (/). - If there’s an error, displays an error message |
| deleteAccount() | Handles account deletion process | - Sends a DELETE request to the server (/delete\_account). - If the account is successfully deleted, the user is alerted and redirected to the home page. - If the deletion fails, an error message is shown. |