Approach to Building Features

1. Add User Feature:

HTML Structure:

- Create an HTML form with input fields for user details (in this case, just the name).
- Include a "Submit" button within the form.
- When the form is submitted, it will capture user input and add a new user to the list.

JavaScript Implementation:

- Add an event listener to the form's submit button to prevent the default form submission.
- Capture the input value (user's name) from the input field.
- Create a new user object with a unique ID (in this case, using Date.now()) and the user's name.
- Push the new user object to the users array.
- Call the displayUsers function to update the user list on the page with the new user's information.

2. Delete User Feature:

HTML Structure:

- For each user displayed in the list, add a "Delete" button next to their name.
- The "Delete" button will include a data-id attribute containing the user's ID to identify which user to delete.

JavaScript Implementation:

- Add an event listener to the .user-list-container to listen for clicks on the "Delete" buttons.
- Use event delegation to identify when a "Delete" button is clicked.
- Retrieve the data-id attribute to determine which user to delete.
- Call the deleteUser function, passing the user's ID as a parameter.
- In the deleteUser function, use the filter method to remove the user with the specified ID from the users array.
- After deleting the user, call the displayUsers function to update the user list on the page.

3. Search User Feature:

HTML Structure:

- Create an HTML input box (search bar) where users can type to search for other users.
- Add a placeholder in the input box to instruct users to search by name.

JavaScript Implementation:

- Add an event listener to the search input to detect changes in its value (i.e., user input).
- Capture the search term entered by the user and convert it to lowercase for case-insensitive matching.
- Use the filter method to create a new array (filteredUsers) containing only users whose names include the search term.
- Call the displayUsers function, passing the filteredUsers array to update the displayed user list with the search results.
- The user list is dynamically updated as the user types, allowing for real-time searching by name.

4. Pagination Feature:

JavaScript Implementation:

- Define constants for the number of items per page (itemsPerPage) and the current page (currentPage).
- Calculate the startIndex and endIndex based on the currentPage and itemsPerPage to display the appropriate range of users.
- When users navigate to a different page, the displayUsers function is called with the updated user list to show a different set of users.
- The updatePagination function dynamically generates page buttons based on the total number of users and the itemsPerPage. Users can click on these buttons to navigate through the user list.
- Pagination ensures that the user list is divided into manageable chunks, improving user experience for larger datasets.

Overall, this approach separates concerns by modularizing the code into functions for adding, deleting, searching, and displaying users. Event listeners and delegation are used to handle user interactions, and pagination provides a more organized user interface for managing a growing list of users.