|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [view\_verticallist.html]** | | |
| **Summary***:*  The manage\_verticallists.html file is a template within the Razda Market platform designed for administrators or authorized users to manage "Vertical Lists." This page includes functionalities for creating new vertical lists by entering a list name, description, and items. Additionally, it displays existing lists with their details and allows users to view or delete each list. The page facilitates straightforward list management, enabling better categorization and organization of items within the platform. | | |
| ***Processes*** | | |
| * **Render Create New List Form** | **Displays a form for creating a new vertical list. The form requires inputs such as the list name, description, and items (one per line).** | |
| * **Submit new List Data** | **The form submission triggers a POST request to verticallist\_bp.manage\_verticallists, adding a new list with user-defined details to the database.** | |
| * **Display Existing Vertical Lists** | **Retrieves and displays all existing vertical lists in a table, including each list's ID, name, description, creation date, and available actions.** | |
| * **Manage Existing Lists** | **Provides actions for each list, including viewing detailed list content and a delete option that prompts a confirmation before proceeding with the deletion.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **CSS file (styles.css)** (if linked separately): Manages the styling of input fields, table, and buttons for consistency with the Razda Market design. | * **New List Creation (verticallist\_bp.manage\_verticallists)**: A POST request is sent when the user submits the form to create a new vertical list. This route is expected to handle list creation in the backend. | |
| * **header.html** and **footer.html** (if included later): For navigation and consistency in site appearance. | * **View Vertical List (verticallist\_bp.view\_verticallist)**: Links each list to a detailed view where the user can see all items in the list, accessed via a "View" link. | |
|  | * **Delete Vertical List (verticallist\_bp.delete\_verticallist)**: A POST request is sent to this route when a user confirms the deletion of a list, triggering removal from the database. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **verticallists**: A list of dictionaries or objects representing each vertical list, containing fields such as verticallist\_id, list\_name, description, and created\_at. This data is used to populate the table of existing lists * **Form Data: User-provided input in the "Create New Vertical List" form, including list\_name, description, and items.** | | * **Vertical Lists Management Page**: Generates an organized HTML layout displaying the form to create new lists and a table listing all current lists, each with options to view or delete. * **User Feedback and Actions**: Confirmation prompts before deletion and feedback on actions (e.g., successful list creation or deletion) are managed by the backend and rendered here if implemented. * **Dynamic Display**: Based on the verticallists input, each list is displayed with its ID, name, description, and created date for quick reference. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Form Submission Handling: Ensure that verticallist\_bp.manage\_verticallists properly receives and processes the POST data to create a new list.** |  |  | | --- | | **2. List Retrieval: Verify that verticallists data is correctly fetched and populated into the template to display existing lists accurately.** |  |  | | --- | | **3. Confirmation for Deletion: The delete form includes onsubmit="return confirm('Are you sure you want to delete this vertical list?');" to prevent accidental deletions. Confirm that this prompt functions as expected.** |  |  | | --- | | **4. Error Handling: Handle scenarios where lists are missing or input fields are incorrectly formatted. Display appropriate error messages if needed.** |  |  | | --- | | **5. Styling and Responsiveness: Integrate styling (from styles.css if linked) and ensure the layout works well across different screen sizes.** |  |  | | --- | | **6. Security and Validation: Implement necessary CSRF tokens in forms (if not already in place) and validate form inputs to prevent potential injection attacks.** |  |  | | --- | | **7. Backend Integration: Ensure backend routes (manage\_verticallists, view\_verticallist, and delete\_verticallist) are functional and correctly handle their respective actions.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |