|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [verticallist.py]** | | |
| **Summary***:*  This file manages the functionality for creating, viewing, and deleting vertical lists. Each vertical list can contain multiple items, which are stored in the database with associated metadata (like list\_name, description, and created\_at). This enhances the platform by providing an organized method for categorizing content or products. | | |
| ***Processes*** | | |
| * **Manage Vertical Lists (GET/POST)** | **Allows admins to create new vertical lists with items, display existing lists, and handle potential errors in data submission.** | |
| * **Delete Vertical List** | **Enables the removal of a specific vertical list and all its associated items, with logging and feedback on success or failure.** | |
| * **View Vertical List** | **Displays the items in a specific vertical list, providing a detailed view of each list's content for users or administrators.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **User Login** (user\_login.py): Ensures restricted access for authenticated users, if necessary. | * **Log File** (verticallist.log): Logs actions like creating, deleting, or viewing vertical lists for tracking and auditing purposes. | |
| * **Product Catalog** (product\_catalog.py): (If needed for linking products or items within vertical lists). | * **Database**: Inserts new lists and their items, retrieves lists for display, and deletes specified lists. | |
| **Expected input into file:** | | **Expected output from file:** |
| Form Data: For creating new vertical lists, it receives list\_name, description, and items (a list of items to be included).  List ID: The ID of the vertical list, especially for delete and view actions, is provided in the URL path.  Session (optional): Accesses session data for potential user-specific customizations or authorizations. | | Template Rendering: Displays manage\_verticallists.html and view\_verticallist.html to show lists and items to the user or admin.  Flash Messages: Shows feedback messages (success or error) for operations like list creation, deletion, or retrieval.  Log File Entries: Logs each action for creating, viewing, or deleting lists to facilitate monitoring and issue resolution. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Session and Permission Management: Use session data to determine user access permissions and ensure only authorized users can perform actions, especially creation and deletion.** |  |  | | --- | | **● Logging Actions: Log each view, create, and delete action for tracking vertical list usage, which is essential for monitoring and auditing.** |  |  | | --- | | **● Feedback Provision: Show flash messages to inform users or admins of the success or failure of their actions. This helps improve user experience by providing immediate feedback.** |  |  | | --- | | **● Error Handling: Handle cases where database access or list manipulation may fail, such as if a list is not found, or an error occurs while saving data.** |  |  | | --- | | **● Database Synchronization: Ensure data consistency by committing each change to the database and ensuring each list and item insertion, retrieval, or deletion is handled correctly.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| **Contribution to the Entire Project:**  The **verticallist.py** file enriches the organizational structure on the platform by enabling the categorization of items in vertical lists, which can be used for showcasing products, categories, or other structured content. It improves user experience by presenting information in an organized format and assists administrators in managing content efficiently.  Would you like to proceed with another file, or is further clarification needed on this one? | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: Initial creation of the vertical list management functionality, including routes for adding, viewing, and deleting lists with logging and feedback mechanisms | | |