|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [verticalbox.py]** | | |
| **Summary***:*  This file handles the creation, display, updating, and deletion of vertical categories on the platform. These categories are displayed in a vertical arrangement, enhancing the browsing experience by allowing users to view and select categories organized for easy navigation. This functionality provides a structured way to categorize products or content in a user-friendly format. | | |
| ***Processes*** | | |
| * **Manage Vertical Categories (GET/POST)** | **Allows administrators to create new vertical categories and view existing ones. Validates category data, logs actions, and handles any errors during data submission.** | |
| * **Delete Vertical Category** | **Enables the deletion of a specific category, removing it from the database and providing feedback to the user.** | |
| * **Edit Vertical Category** | **Facilitates updating category details like category\_name and description, enabling admins to keep categories relevant and up-to-date. Retrieves category data for editing and commits updates to the database.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Database**: Retrieves existing categories from the database for display and editing. | * **Log File** (verticalcat.log): Logs category creation, deletion, updates, and retrievals for monitoring and auditing. | |
| * **User Login** (user\_login.py): May restrict access to certain admin functions if necessary. | * **Database**: Inserts, updates, and deletes category records as per user actions. | |
| * **Session Data** (optional): Accesses session data to customize admin/user-specific actions. |  | |
| **Expected input into file:** | | **Expected output from file:** |
| **Form Data: Admin-provided box\_title, content, link, and optionally an image file for box creation or editing.**  **Box ID**: URL-based parameter passed for editing or deleting a specific box by ID. | | **Template Rendering: Renders the manage\_verticalboxes.html for admin management (add/edit/delete) and display\_verticalboxes.html for user viewing.**  **Flash Messages**: Provides feedback to the admin after each operation (success or error) to indicate the outcome of their action.  **Log Entries**: Records the creation, update, deletion, and viewing actions for tracking and monitoring purposes.  **Image Files**: Stores uploaded images in the designated folder (static/verticalboxes) and deletes old images when replaced or the associated box is deleted. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Session and Permission Verification: If admin-only access is required, session data should be used to enforce permissions, allowing only authorized users to manage vertical boxes.** |  |  | | --- | | **● File Management: Ensures that each vertical box with an image has the image stored in static/verticalboxes, and deletes images if a box is deleted or if the image is updated.** |  |  | | --- | | **● Logging: Each action (creation, deletion, updating, and viewing) should be logged to verticalbox.log to facilitate debugging and maintain an audit trail.** |  |  | | --- | | **● Database Error Handling: All database interactions (inserts, updates, deletions) are wrapped with error handling, and user feedback (flash messages) is provided on success or failure.** |  |  | | --- | | **● Validation and File Security: Validates image file extensions to avoid invalid uploads and uses secure\_filename to prevent potential file path issues or malicious file names.** |  |  | | --- | | **● User Feedback: The file provides user feedback through flash messages, informing the admin of any issues (like missing required fields or invalid image types) or successful actions.** | | | |
| Edit log (update each time you make changes to doc or file). | | |
| **Contribution to the Entire Project:**  The **verticalbox.py** file enhances user experience by providing a streamlined, vertically arranged layout for showcasing products, which is ideal for narrow screens or specific aesthetic preferences. By enabling admins to create and manage these vertical displays, the file contributes to the platform's customization capabilities, allowing different visual layouts for various screen sizes and user engagement strategies.  Is there any additional detail you would like on this file, or shall we proceed to another one? | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: Set up vertical box management to allow admin users to create, edit, and delete vertically-styled product displays with optional images. Integrated logging for tracking each action and configured file storage to handle associated image files. | | |