|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [manage\_verticalcategories.html]** | | |
| **Summary***:*  The manage\_verticalcategories.html file serves as the interface for managing vertical categories within Razda Market. This page allows admins to create new categories, view and edit existing categories, and delete categories as needed. Each category is displayed with relevant information such as ID, name, description, creation date, and last updated date, facilitating organized management of categorized items. | | |
| ***Processes*** | | |
| * **Create New Vertical Category** | **Provides a form for admins to enter details for a new category, including category name and description. The form sends a POST request to create a new category.** | |
| * **View Existing Vertical Categories** | **Displays a table of all current categories, showing details like ID, name, description, creation date, and last updated date. This enables admins to view and manage all categories in one place.** | |
| * **Edit Vertical Category** | **Each category has an "Edit" link, directing admins to a form where they can modify category details such as name and description.** | |
| * **Delete Vertical Category** | **The delete option sends a POST request to remove a category from the system, with an onsubmit confirmation prompt to prevent accidental deletions.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **CSS file (styles.css)** (if linked): Adds styling to the form, table, and buttons, ensuring the layout is consistent with Razda Market’s design. | * **Create Category Route (verticalcat\_bp.manage\_verticalcategories)**: The form for creating a new category sends a POST request to this route, which processes the new category creation in the backend. | |
| * **header.html** and **footer.html** (if included later): For a consistent interface layout across the site. | * **Edit Category Route (verticalcat\_bp.edit\_verticalcategory)**: Each category's "Edit" link directs to a form where admins can update category details. | |
|  | * **Delete Category Route (verticalcat\_bp.delete\_verticalcategory)**: The delete button sends a POST request to remove a category, with confirmation to prevent accidental deletions. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **categories**: A list of dictionaries or objects containing details for each vertical category, including fields such as category\_id, category\_name, description, created\_at, and updated\_at. This data populates the table of existing categories. * **Form Data**: Input from the "Create New Vertical Category" form, including category\_name and description. | | * **Category Management Interface**: Displays a form for creating categories and a table listing all current categories with actions to edit or delete each entry. * **Category Management Actions**: Provides functionalities for creating, editing, and deleting categories. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Data Loading: Ensure categories is passed correctly to the template, allowing the table to display all existing categories accurately.** |  |  | | --- | | **2. Create Category Handling: Verify that the verticalcat\_bp.manage\_verticalcategories route correctly processes POST requests to create a new category based on form inputs.** |  |  | | --- | | **3. Edit Functionality: Confirm that each category’s "Edit" link navigates to the correct route and that the update functionality processes the changes as expected.** |  |  | | --- | | **4. Delete Functionality: Ensure the delete\_verticalcategory route processes POST requests for category deletion and that the confirmation prompt functions correctly to prevent accidental deletions.** |  |  | | --- | | **5. Error Handling: Provide user-friendly error handling for cases where category creation, editing, or deletion fails, guiding the admin on the next steps.** |  |  | | --- | | **6. Data Persistence: Confirm that all actions (creation, editing, and deletion) reflect correctly in the backend and are displayed upon page reload or navigation.** |  |  | | --- | | **7. Styling and Responsiveness: Confirm that the form and table are styled consistently and responsively, maintaining readability on different screen sizes, especially for mobile.** |  |  | | --- | | **8. Form Validation: Implement server-side validation for the form inputs, ensuring that the category name is required and preventing the submission of invalid data.** |  |  | | --- | | **9. Audit Trail for Updates: Ensure the updated\_at field is accurately updated for each modification, providing an audit trail for when and how categories have been altered.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |