|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [manage\_verticalboxes.html]** | | |
| **Summary***:*  The manage\_verticalboxes.html file serves as the management interface for vertical boxes within Razda Market. This page allows admins to create new vertical boxes with customizable titles, content, links, and optional images. The interface displays existing boxes, with options to edit or delete them, helping maintain organized, visually engaging content blocks for the marketplace. | | |
| ***Processes*** | | |
| * **Create New Vertical Box:** | **Provides a form for adding new vertical boxes. The form includes fields for title, content, optional link, and optional image upload, which submits a POST request to create the box.** | |
| * **Display Existing Vertical boxes:** | **Displays a table of existing boxes, including each box’s title, content, link, and image, allowing admins to review current vertical boxes.** | |
| * **Edit Vertical box:** | **Each vertical box entry has an "Edit" link, directing admins to a form where they can modify box details such as title, content, link, or image.** | |
| * **Delete Vertical Box:** | **Each box includes a delete button, sending a POST request to remove the box from the system. A confirmation dialog can prevent accidental deletions.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **CSS file (admin.css)**: Adds styling to the form, table, and buttons, ensuring the layout is clean and consistent with the Razda Market admin interface. | * **Create Box Route (verticalbox\_bp.manage\_verticalboxes)**: The form submission for creating a new box sends a POST request, handled by this route to save the box in the backend | |
| * **header.html** and **footer.html** (if included): For a consistent layout across the site. | * **Edit Box Route (verticalbox\_bp.edit\_verticalbox)**: Each box’s "Edit" link directs to a form for updating the box’s details. | |
|  | * **Delete Box Route (verticalbox\_bp.delete\_verticalbox)**: The delete button sends a POST request to remove the selected box from the backend. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **verticalboxes**: A list of dictionaries or objects containing details for each vertical box, including fields such as box\_id, box\_title, content, link, and image\_filename. This data is used to populate the table of existing boxes. * **Form Data**: User-provided input from the "Create New Vertical Box" form, including box\_title, content, optional link, and optional image. | | * **Vertical Box Management Interface**: Displays a form for creating vertical boxes and a table listing all current boxes with options to edit or delete. * **Box Management Actions**: Allows for creation, editing, and deletion of vertical boxes. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Data Loading: Ensure verticalboxes is passed to the template to allow the table to display all existing boxes accurately.** |  |  | | --- | | **2. Create Box Handling: Verify that the verticalbox\_bp.manage\_verticalboxes route processes the POST request to create a new vertical box based on the form inputs.** |  |  | | --- | | **3. Edit Functionality: Confirm that each box’s "Edit" link navigates to the correct route, allowing admins to update box details as needed.** |  |  | | --- | | **4. Delete Functionality: Ensure the delete\_verticalbox route processes the POST request for deleting boxes, and that the delete button has a confirmation prompt to prevent accidental deletions.** |  |  | | --- | | **5. Image Display: If an image is uploaded, ensure it is displayed in the table under the correct box. Verify that the image is stored in the correct directory (e.g., verticalboxes/ folder) and displayed with a valid path.** |  |  | | --- | | **6. Error Handling: Provide user-friendly error handling for cases where box creation, editing, or deletion fails, with guidance for the admin on corrective actions.** |  |  | | --- | | **7. Data Persistence: Ensure that actions such as creation, updating, and deletion reflect immediately in the backend and display upon page reload or navigation.** |  |  | | --- | | **8. Styling and Responsiveness: Confirm that the form and table are styled for readability and accessibility, ensuring responsiveness on various screen sizes.** |  |  | | --- | | **9. Form Validation: Validate form inputs to prevent invalid data from being submitted, particularly for required fields like box\_title and content, as well as URL and file type validation for optional fields.** |  |  | | --- | | **10. File Handling and Security: Confirm that image files are processed securely to prevent file injection or malicious uploads, and consider restricting file size and type.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |