|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [manage\_order.html]** | | |
| **Summary***:*  The manage\_order.html file is a page within the Razda Market Admin dashboard that enables administrators to view and manage individual customer orders. It displays order details, including order ID, user ID, status, total amount, and ordered items. Admins can update the order status using a dropdown menu and submit their changes, allowing for efficient order tracking and management within the marketplace. | | |
| ***Processes*** | | |
| * **Display Order Details:** | **Shows order-specific details such as Order ID, User ID, Status, Total Amount, and Order Date for clarity and easy reference.r** | |
| * **List Ordered Items:** | **Displays a table of items in the order, including Product Name, Price at Purchase, Quantity, and Subtotal. Each item row includes a thumbnail image for quick visual identification.** | |
| * **Update Order Status:** | **Provides a dropdown menu to change the order status (e.g., Pending, Processing, Shipped, Delivered, Canceled). The form submission updates the status in the backend.** | |
| * **Display Flash messages:** | **Uses get\_flashed\_messages to display feedback (success, error messages) for actions taken, such as updating the order status. Messages are styled with Bootstrap alerts for visibility.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **SS file (styles.css)**: Provides styling to ensure that tables, form fields, and buttons are consistent with the Razda Market admin interface. | * **Update Order Status Route (manage\_order)**: The status update form sends a POST request to update the order status in the backend database, changing it to the newly selected value. | |
| * **header.html** and **footer.html** (if included later): For a consistent layout and navigation experience across the admin site. | * **View All Orders (order\_management.view\_all\_orders)**: The "Back to Orders" button links to the main orders page, allowing easy navigation back to the full order list. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **order**: An object or dictionary containing the order's details, such as order\_id, user\_id, order\_status, total\_amount, and created\_at. These values are displayed in the order summary section. * **order\_items**: A list of dictionaries or objects representing each item in the order, including details like name, price\_at\_purchase, quantity, and image\_path. This data populates the ordered items table. * **Form Data**: Order status selection from the dropdown menu for updating the order’s current status. | | * **Order Summary Display**: Displays the current order details, including the user and order ID, order status, total amount, and ordered date, ensuring admins have a clear overview of the order at hand. * **Order Items Table**: Lists all items in the order with their price, quantity, subtotal, and an image thumbnail, providing clarity on what the customer has purchased. * **Order Status Update**: Submits the new status selection to the backend, updating the order's status in the system. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **1. Data Loading: Ensure that order and order\_items are correctly passed to the template, enabling accurate display of order details and itemized lists.** |  |  | | --- | | **2. Order Status Update Handling: Confirm that the POST request from the status update form is processed by the backend, updating the order’s status based on the selected value.** |  |  | | --- | | **3. Flash Messages: Implement get\_flashed\_messages for feedback on actions, such as success or error messages for status updates, ensuring admins receive immediate feedback on their actions.** |  |  | | --- | | **4. Subtotal Calculation: Verify that each item’s subtotal (price\_at\_purchase \* quantity) is calculated and displayed accurately, formatted to two decimal places for consistency.** |  |  | | --- | | **5. Image Loading: Confirm that each item’s image loads correctly from image\_path. Implement a fallback image or error handling if any image path is broken or missing.** |  |  | | --- | | **6. Responsive Layout: Ensure the layout is responsive and accessible across different screen sizes, especially for tablet and desktop views used by admins.** |  |  | | --- | | **7. Error Handling: Implement error handling for scenarios where data fails to load or status updates do not succeed, providing useful feedback and options for resolution.** |  |  | | --- | | **8. Security and CSRF Protection: Ensure that forms, particularly the status update form, are CSRF-protected to prevent unauthorized actions.** |  |  | | --- | | **9. Navigation and Usability: Confirm the "Back to Orders" button accurately links to order\_management.view\_all\_orders for seamless navigation back to the main orders page.** |  |  | | --- | | **10. Audit Logging: (Optional) Consider logging status changes for order tracking and accountability, helping maintain a history of actions taken on each order.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |