|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [checkout.py]** | | |
| **Summary***:*  checkout.py manages the checkout process, including order confirmation, payment processing, and inventory management. It validates user login, calculates the cart total, processes payments, and updates the database with order details. This ensures a seamless checkout experience and maintains data consistency across inventory and order records. | | |
| ***Processes*** | | |
| * **Initialize Blueprint** | **checkout\_bp is created as a Flask Blueprint for managing checkout-specific routes, allowing consistent, modular use across the application.** | |
| * **Cart Verification** | **Ensures the user is logged in and has items in their cart. Redirects users if the cart is empty or they are not logged in.** | |
| * **Calculate Total Amount** | **Loops through cart\_items to compute total\_amount, used to charge the user and create an order record.** | |
| * **Process Payment** | **Calls process\_payment() (mock function here) to process the total amount, simulating interaction with a real payment gateway for demonstration.** | |
| * **Create Order** | **If payment succeeds, creates a new order in the database for the user, logging the total order amount and any issues encountered.** | |
| * **Add Order Items** | **For each item in the cart, it records details (product, quantity, price) in the database, associating them with the order ID.** | |
| * **Reduce Inventory** | **Adjusts inventory based on ordered items to ensure stock availability, updating item quantities.** | |
| * **Clear Cart** | **Clears the user's cart upon successful order completion, removing items and preventing accidental reorder.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment (.env)**: loads configurations with load\_dotenv() for environment-specific settings like database or logging preferences. | * **Templates (checkout.html)**: renders checkout.html with cart items and total amount for the user to review before confirming the order. | |
| * **Database (Products, Orders)**: interacts with tables to retrieve cart items, manage inventory, and create order records in the database. | * **Order History**: upon order placement, redirects users to order history view to access their order details. | |
| * **Inventory Module**: uses inventory.reduce\_inventory() to manage stock for ordered items. | * **Logs**: writes logs of each action to checkout.log to track user transactions and system errors. | |
| **Expected input into file:** | | **Expected output from file:** |
| * **User Authentication**: requires user session to check login and retrieve user\_id for order processing. * **Cart Items**: retrieves items from get\_user\_cart\_items() for the authenticated user, including quantity and price per item. * **Total Amount Calculation**: computes total\_amount by multiplying the price and quantity of each item. | | * **Successful Checkout**: redirects users to the order history upon successful payment and order placement. * **Error/Failure Messages**: displays appropriate error messages if payment fails or if there’s an issue creating the order. * **Log Output**: logs significant actions and errors (e.g., payment success, order creation failure) to checkout.log. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **- Payment Integration: Replace process\_payment() placeholder with a real integration to a payment provider (e.g., Stripe, PayPal) to handle transactions securely.** |  |  | | --- | | **- Inventory Consistency: Ensure real-time stock updates to avoid overselling by calling inventory.reduce\_inventory() after each successful checkout.** |  |  | | --- | | **- Cart Synchronization: Clear the user's cart once an order is created and inventory is adjusted to prevent accidental reorders or mischarges.** |  |  | | --- | | **- Error Handling and Messaging: Provide clear error messages for payment failures or backend issues, allowing users to resolve or retry their checkout.** |  |  | | --- | | **- Checkout Security: Ensure data integrity and session security during checkout, using HTTPS for sensitive interactions and secure session management.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: | | |