Search Bar

For a search bar on your website, which typically involves a text input field and sometimes additional filters or options, we can use equivalence partitioning to ensure comprehensive testing. Here’s how you might outline possible test cases based on typical inputs for such a search bar.

Requirements Specification:

Search Input: Typically a text field where users can enter keywords, phrases, or specific queries.

Advanced Options: Optional filters such as category, date range, or other relevant criteria.

Derived Equivalence Classes

Valid Classes

Search input contains relevant keywords or phrases in plain text (valid).

Search input with special characters typically used in search queries, like hyphens, underscores, or quotation marks (valid).

Advanced options are correctly selected and compatible with the input (valid).

Invalid Classes

Search input is empty (invalid).

Search input contains only special characters not typically used in search queries, like emoji or non-printable characters (invalid).

Advanced options selected are contradictory or not applicable (invalid).

Black Box Test Cases for Search Bar Based on Equivalence Classes

#	Test Data	Expected Outcome	Classes Covered
1	“apple”, no filters selected	True, returns results for “apple”	1
2	“user@example.com”, no filters	True, returns results for email-like query	1, 2
3	“2023-01-01”, “date range selected”	True, returns results from specific date	1, 3
4	“”, no filters selected	False, no results	4
5	“😊”, no filters selected	False, no results	5
6	“apple”, “category: electronics”, “category: clothing”	False, conflicting filters	6

These test cases aim to cover both valid and invalid input scenarios, ensuring that the search functionality processes relevant searches correctly and rejects or handles errors for inappropriate inputs effectively. Each test case is designed to target different equivalence classes to ensure full coverage of the functionalities of your search bar. Adjustments might be needed based on specific features, such as auto-complete or search suggestions, which could influence how inputs are handled and validated.

Shopping Cart

For a shopping cart on an e-commerce website, which typically involves operations such as adding items, removing items, updating quantities, and sometimes applying promo codes, equivalence partitioning can be used to ensure comprehensive testing. Below is an outline of possible test cases based on typical interactions with a shopping cart.

Requirements Specification:

Add Items: Users can add items to the cart with varying quantities.

Remove Items: Users can remove items from the cart.

Update Quantities: Users can change the quantity of the items already in the cart.

Apply Promo Codes: Users can apply promo codes for discounts.

Derived Equivalence Classes

Valid Classes

Adding a valid item in a valid quantity (1 or more) to the cart (valid).

Removing an item that is already in the cart (valid).

Updating an existing item's quantity within allowed limits (e.g., 1 to 10) (valid).

Applying a valid promo code that is active and applicable to the items in the cart (valid).

Invalid Classes

Adding an item with a quantity of zero or negative numbers (invalid).

Removing an item not present in the cart (invalid).

Updating an item's quantity to zero or a negative number, or beyond the stock limit (invalid).

Applying an expired or invalid promo code, or a code not applicable to any items in the cart (invalid).

Black Box Test Cases for Shopping Cart Based on Equivalence Classes

#	Test Data	Expected Outcome	Classes Covered
1	Add “Beer”, quantity 1	True, item “Beer” added with quantity 1	1
2	Remove “Beer” already in cart	True, “Beer” removed from cart	2
3	Update “Beer” quantity from 1 to 2	True, “Beer” quantity updated to 2	3
4	Apply promo code “20OFF”	True, promo code applied, discount reflected	4
5	Add “Beer”, quantity 0	False, no change to cart	5
6	Remove “Beer” (not in cart)	False, no change to cart	6
7	Update “Beer” quantity to 15 (stock limit 10)	False, quantity remains unchanged	7
8	Apply expired promo code “10OFF2010”	False, promo code not applied	8

These test cases are designed to cover both valid and invalid scenarios, ensuring that the shopping cart functionality handles all operations correctly, efficiently blocks invalid actions, and provides appropriate feedback. This comprehensive approach to testing helps ensure that users have a smooth and reliable experience managing their shopping cart. Adjustments may be required based on additional features or specific business rules related to inventory management or promo code restrictions.