Web Engineering

Assignment Question:

Design and implement an interactive e-commerce shopping cart system using vanilla JavaScript that demonstrates advanced event handling techniques, including event delegation, bubbling, capturing, and propagation control. The system should allow users to add items to a cart, remove items, and apply discount coupons, with all interactions dynamically updating the cart's total price

To ensure efficiency, use event delegation to handle all "Add to Cart" button clicks from a single parent listener rather than binding individual event handlers to each button. Log the event propagation path (using event.target and event.currentTarget) to demonstrate how events bubble up the DOM. Implement a toggle mechanism to switch between event bubbling and capturing, logging the differences in propagation order. For the "Remove" buttons, use stopPropagation() to prevent unintended side effects from parent event listeners.

The system should also include a coupon validation feature, where users can enter discount codes (e.g., "SAVE10" for 10% off) and see the total price adjust accordingly.

All user interactions—such as adding items, removing items, and applying coupons—should be logged to the console with timestamps for analytics. Add java applets from a cdn server.

For real-time UI updates, ensure the cart recalculates the total price whenever items are added, removed, or discounts are applied. The solution should avoid redundant DOM manipulations and inefficient event binding.

How to submit ?

A fully functional vanilla JavaScript implementation (no libraries/frameworks). A detailed report explaining:

The benefits of event delegation over direct event binding. When and why stopPropagation() is necessary. The differences between bubbling and capturing and their practical implications.

A short demo video showcasing: Event propagation logs. Coupon application and price updates. The toggle between bubbling and capturing.