1. **Server-side storage**: Implement a backend server that stores the shopping cart information for each user. This can be done using a database or any other persistent storage mechanism.
2. **Session management**: When a user visits the website and opens the shopping cart, create a unique session identifier for that user. Associate this session identifier with the user's shopping cart data stored on the server.
3. **Client-side synchronization**: On the client side, when a user adds an item to the cart, send an asynchronous request to the server to update the shopping cart for that session. The server should receive the request, update the cart data, and store it in the database.
4. **Real-time updates**: To ensure that the cart is updated across multiple pages without relying on page refreshes, you can utilize a real-time communication mechanism such as WebSockets. When an item is added to the cart, notify all other open instances of the shopping cart in real-time. This can be done by broadcasting a message through a WebSocket connection to all connected clients with the same session identifier.
5. **Client-side rendering**: Each page that displays the shopping cart should establish a WebSocket connection to the server upon loading. When the WebSocket receives an update notification, update the cart data on the client-side without needing to refresh the page.
6. **Fallback mechanisms**: In case WebSocket connections are not supported or fail, you can use alternative techniques like long polling or server-sent events (SSE) to achieve a similar real-time update behavior. These techniques involve establishing a long-lived HTTP connection and periodically checking the server for updates.
7. **Polling on focus event**: As an additional fallback mechanism, you can implement polling when the web browser's focus event occurs. When a user switches back to a tab, send a quick request to the server to check for any updates since the last update or refresh. If there are updates, retrieve the latest cart data and update the client-side representation accordingly.

By combining server-side storage, real-time updates through WebSockets (or alternative techniques), and periodic polling, you can ensure that the shopping cart stays up to date across multiple pages and even when a user refreshes or switches tabs.