```
out">Edit</button>

                        <button
onclick="deleteProduct('${product.id}')"
class="px-3 py-1 bg-red-500 hover:bg-
red-600 text-white rounded-md text-sm
font-medium transition duration-300
ease-in-out">Delete</button>

                    </div>

                </div>

            `).join('');

        }



        // Expose functions to the global
scope for onclick attributes
```

```
    window.editProduct = (productId) => {

        const product = productsData.find(p => p.id === productId);

        if (product) {

            editingId = productId;

            productNameInput.value = product.name;

            costPriceInput.value = product.costPrice;

            sellingPriceInput.value = product.sellingPrice;

            formButton.textContent = "Update Product";
            formButton.classList.remove('bg-indigo-600', 'hover:bg-indigo-700');
            formButton.classList.add('bg-green-600', 'hover:bg-green-700');
        }
    };

    window.deleteProduct = async (productId) => {
        try {
            const productDocRef = doc(db, `artifacts/${appId}/public/data/paint_shop_products`, productId);
            await deleteDoc(productDocRef);
        } catch (error) {
            console.error("Error deleting document:", error);
        }
    };

    // Handle form submission
    productForm.addEventListener('submit', async (e) => {
        e.preventDefault();
        const productName = productNameInput.value;
        const costPrice = parseFloat(costPriceInput.value);
        const sellingPrice =
parseFloat(sellingPriceInput.value);


        if (isNaN(costPrice) ||
isNaN(sellingPrice)) {

            console.error("Cost and Selling
prices must be valid numbers.");

            return;

        }


        const productData = {

            name: productName,

            costPrice: costPrice,

            sellingPrice: sellingPrice
        };

        try {
            if (editingId) {
                const productDocRef = doc(db,
`artifacts/${appId}/public/data/
paint_shop_products`, editingId);
                await
updateDoc(productDocRef, productData);
                editingId = null;
                formButton.textContent =
"Add Product";
                formButton.classList.remove('b
g-green-600', 'hover:bg-green-700');
```

```
                    formButton.classList.add('bg-indigo-600', 'hover:bg-indigo-700');

                } else {

                    const productCollectionRef = collection(db, `artifacts/${appId}/public/data/paint_shop_products`);

                    await addDoc(productCollectionRef, productData);

                }

                productForm.reset();

            } catch (error) {

                console.error("Error saving document:", error);
```

```
        }

      });

      // Add event listeners for sorting and search

      sortByNameBtn.addEventListener('click', () => {

        currentSortKey = 'name';

        renderProducts(currentSortKey);

      });

      sortByCostBtn.addEventListener('click', () => {

        currentSortKey = 'costPrice';
      renderProducts(currentSortKey);

    });

    sortBySellingBtn.addEventListener('cl
ick', () => {

      currentSortKey = 'sellingPrice';

      renderProducts(currentSortKey);

    });



    searchInput.addEventListener('input
', () => {

      renderProducts(currentSortKey);

    });
```

```
        // Register the service worker for PWA functionality
        if ('serviceWorker' in navigator) {
            window.addEventListener('load', () => {
                navigator.serviceWorker.register('/service-worker.js').then(registration => {
                    console.log('ServiceWorker registration successful with scope: ', registration.scope);
                }, err => {
                    console.log('ServiceWorker registration failed: ', err);
                });
```

```
      });

    }


    // Start the app when the window
loads

    window.onload = initializeAndAuth;

  </script>

</body>

</html>
```