



The banner features a blue background with a white diagonal stripe. On the left, there is a CSS3 logo (a blue shield with a white 'E' and '3') and the jQuery logo (a blue circle with a white 'j' and 'Query' in black). On the right, there is an HTML5 logo (an orange shield with a white '5') and a JavaScript logo (a yellow shield with a white 'J' and 'S'). In the center, the text 'ASSIGNMENT OF AWT' is written in large, bold, black letters, followed by 'ON' in smaller letters, and 'INVENTORY MANAGEMENT SYSTEM' in bold, black letters. Below the jQuery logo, the text 'write less, do more.' is written in a smaller, italicized font. At the bottom left, a green box contains the submission details: 'SUBMITTED TO : MR.YOGESH BAJPAI', 'SUBMITTED BY : SUDHANSHU DHANIK', 'COURSE : B.TECH CSE', 'SEM : IV SEC : 'C'', and 'ADMISSION ID : 22BTCSE0277'. On the right side, there is a large purple letter 'B' surrounded by various colorful icons representing different web development concepts like a lightbulb, a gear, a smiley face, a bug, a download arrow, and a person icon.

CSS3

ASSIGNMENT OF AWT

ON

INVENTORY MANAGEMENT SYSTEM

jQuery
write less, do more.

HTML5

JavaScript

B

SUBMITTED TO : MR.YOGESH BAJPAI
SUBMITTED BY : SUDHANSHU DHANIK
COURSE : B.TECH CSE
SEM : IV SEC : 'C'
ADMISSION ID : 22BTCSE0277

SOURCE CODE LINK:

https://drive.google.com/drive/folders/198zJruWsf_0X1GxNNZR6AUvZHuw9AUyc?usp=sharing

CODE STARTS HERE :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Inventory Management System</title>

  <!-- Bootstrap CSS -->
```

```
<link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

<!-- jQuery -->

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<style>

    .product-list {

        margin-top: 30px;

    }

</style>
</head>
<body>
<div class="container-fluid bg-light">

    <h1 class=" text-center p-3 bg-info">Inventory Management System</h1>

    <!-- Product List -->

    <div class="product-list">

        <h3 class="display">Product List</h3>

        <table class="table table-striped">

            <thead>

                <tr class="bg-dark text-white">

                    <th>Product Name</th>

                    <th>Description</th>

                    <th>SKU</th>

                    <th>Price</th>

                    <th>Quantity</th>

                </tr>

            </thead>

            <tbody id="productList">

                <!-- Products will be dynamically added here -->
```

```

        </tbody>

    </table>

    <!-- Add Product Button -->

    <button type="button" class="btn btn-primary mt-3" data-toggle="modal" data-
target="#addProductModal">Add Product</button>

    </div>

</div>



<!-- Add Product Modal -->

<div class="modal fade" id="addProductModal" tabindex="-1" role="dialog" aria-
labelledby="addProductModalLabel" aria-hidden="true">

    <div class="modal-dialog" role="document">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title mark" id="addProductModalLabel" >Add Product</h5>

                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">

                    <span aria-hidden="true">&times;</span></button>

            </div>

            <div class="modal-body">

                <!-- Form to add a new product -->

                <form id="addProductForm" class="bg-light text-warning lead">

                    <div class="form-group">

                        <label for="productName">Product Name</label>

                        <input type="text" class="form-control" id="productName" required>

                    </div>

                    <div class="form-group">

                        <label for="productDescription">Description</label>

                        <textarea class="form-control" id="productDescription"
required></textarea>

```

```

    </div>

    <div class="form-group">

        <label for="productSKU">SKU</label>

        <input type="text" class="form-control" id="productSKU" required>

    </div>

    <div class="form-group">

        <label for="productPrice">Price</label>

        <input type="number" class="form-control" id="productPrice" required>

    </div>

    <div class="form-group">

        <label for="productQuantity">Quantity</label>

        <input type="number" class="form-control" id="productQuantity"
required>

    </div>

</form>

</div>

<div class="modal-footer">

    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>

    <button type="button" class="btn btn-primary" id="addProductBtn">Add
Product</button>

</div>

</div>

</div>
</div>

```

```

<!-- Bootstrap JS -->

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></sc
ript>

```

```
<!-- Custom JavaScript -->

<script>

    $(document).ready(function(){

        $("#addProductBtn").click(function(){

            // Retrieve input values

            var productName = $("#productName").val();

            var productDescription = $("#productDescription").val();

            var productSKU = $("#productSKU").val();

            var productPrice = $("#productPrice").val();

            var productQuantity = $("#productQuantity").val();


            // Validate input values (e.g., check for empty fields)

            if(productName === "" || productDescription === "" || productSKU === "" ||
productPrice === "" || productQuantity === "") {

                alert("Please fill in all fields.");

                return;

            }


            // Check if product with the same SKU already exists

            var existingProduct = getProductBySKU(productSKU);

            if (existingProduct) {

                // If product already exists, update its quantity

                var newQuantity = parseInt(existingProduct.quantity) +
parseInt(productQuantity);

                updateProductListItem(existingProduct.sku, newQuantity);

            } else {

                // If product doesn't exist, add it to the list

                var newProduct = {

                    name: productName,

                    description: productDescription,

                    sku: productSKU,
```

```

        price: productPrice,

        quantity: productQuantity
    };

    addProductToList(newProduct);
}

// Clear the form fields
$("#addProductForm")[0].reset();

// Close the modal
$("#addProductModal").modal("hide");

// Check for low stock
if (productQuantity < 10) {
    alert("Low stock alert for " + productName);
}

});

// Function to add a new product to the list
function addProductToList(product) {
    var productRow = `
        <tr id="product-${product.sku}">
            <td>${product.name}</td>
            <td>${product.description}</td>
            <td>${product.sku}</td>
            <td>${product.price}</td>
            <td>${product.quantity}</td>
        </tr>
    `;

    $("#productList").append(productRow);

```

```
}
```

```
// Function to update the quantity of an existing product in the list
```

```
function updateProductListItem(sku, newQuantity) {  
    $("#product-" + sku + " td:last-child").text(newQuantity);  
}
```

```
// Function to get a product by SKU
```

```
function getProductBySKU(sku) {  
    var products = $("tbody#productList tr");  
    for (var i = 0; i < products.length; i++) {  
        var productSKU = $(products[i]).attr("id").split("-")[1];  
        if (productSKU === sku) {  
            return {  
                name: $(products[i]).find("td:eq(0)").text(),  
                description: $(products[i]).find("td:eq(1)").text(),  
                sku: sku,  
                price: $(products[i]).find("td:eq(3)").text().substring(1),  
                quantity: $(products[i]).find("td:eq(4)").text()  
            };  
        }  
    }  
    return null;  
}  
  
$("input").css("background-color", "cyan");  
$("textarea").css("background-color", "cyan");
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

OUTPUT :

Inventory Management System

Product List

Product Name	Description	SKU	Price	Quantity
--------------	-------------	-----	-------	----------

Add Product

AFTER CLICKING ON Add Product Button:

Product List

Product Name

Description

Price

Quantity

Add Product

Add Product

Product Name

Description

SKU

Price

Quantity

Close

Add Product

ALERT POP UP WHEN ANY FIELD IS EMPTY :

This page says

Please fill in all fields.

OK

Ir A

Pr

Desc

Price

Description

The phone comes with a 90 Hz refresh rate 6.60-inch display offering a resolution of 1080x2408 pixels (FHD+).

SKU

SM-A146B

Price

199

Quantity

Close

Add Product

ALERT WHEN NUMBER OF QUANTITY<10 :

This page says

Low stock alert for Samsung Galaxy A14 5G

OK

Samsung Galaxy A14 5G

Description

The phone comes with a 90 Hz refresh rate 6.60-inch display offering a resolution of 1080x2408 pixels (FHD+).

SKU

SM-A146B

Price

199

Quantity

8

Close

Add Product

AFTER ADDING PRODUCTS ON LIST :

Inventory Management System

Product List

Product Name	Description	SKU	Price	Quantity
Vivo Y21	Vivo Y21 The phone comes with a 6.50-inch touchscreen display offering a resolution of 720x1600 pixels. Vivo Y21 is powered by an octa-core MediaTek Helio P35 (MT6765) processor. It comes with 4GB of RAM. The Vivo Y21 runs Android 11 and is powered by a 5000mAh battery.	YB-134	\$123	12
Samsung Galaxy A14 5G	The phone comes with a 90 Hz refresh rate 6.60-inch display offering a resolution of 1080x2408 pixels (FHD+). Samsung Galaxy A14 5G is powered by a 2.2Ghz MHz octa-core processor. It comes with 4GB, 6GB, 8GB of RAM.	ST-42313	\$199.7	35
Redmi Note 10	The phone comes with a 60 Hz refresh rate 6.70-inch touchscreen display offering a resolution of 720x1648 pixels (HD+). The display sports Gorilla Glass for protection. Redmi 10 is powered by an octa-core Qualcomm Snapdragon 680 processor.	WSQ-56432	\$105	32

Add Product