Javascript Task

**index.html :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>JavaScript Task: Transaction Management System</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h2>Transaction Management System</h2>

<div class="content">

<div class="form-container">

<form id="transactionForm">

<label for="transactionId">Transaction ID:</label>

<input type="text" id="transactionId" name="transactionId" disabled>

<label for="payee">Payee:</label>

<input type="text" id="payee" name="payee" required>

<label for="amount">Amount:</label>

<input type="number" id="amount" name="amount" step="0.01" required>

<label for="reason">Reason:</label>

<input type="text" id="reason" name="reason">

<button type="submit">Add Transaction</button>

</form>

</div>

<div class="table-container">

<table id="transactionTable">

<thead>

<tr>

<th>ID</th>

<th>Payee</th>

<th>Amount</th>

<th>Reason</th>

<th>Edit</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<tr id="noTransactionMessage">

<td colspan="6">No Transactions found.</td>

</tr>

</tbody>

</table>

</div>

</div>

<!-- Total Transaction Amount Display -->

<div class="total-container">

<h3>Total Transaction Amount:</h3>

<div id="totalTransactionAmount">0.00</div> <!-- Initially set to 0.00, will be updated dynamically -->

</div>

</div>

<script src="script.js"></script>

</body>

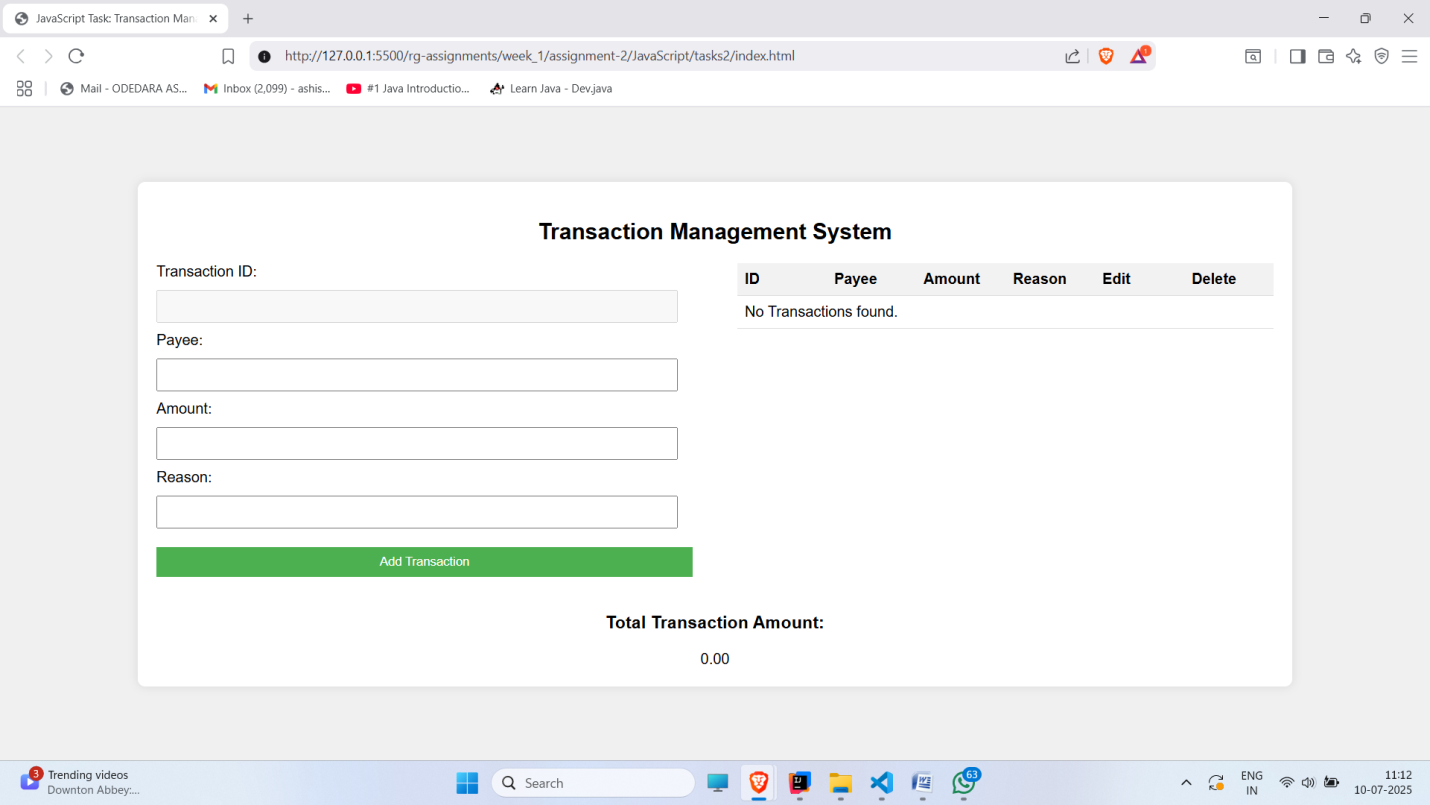
</html>

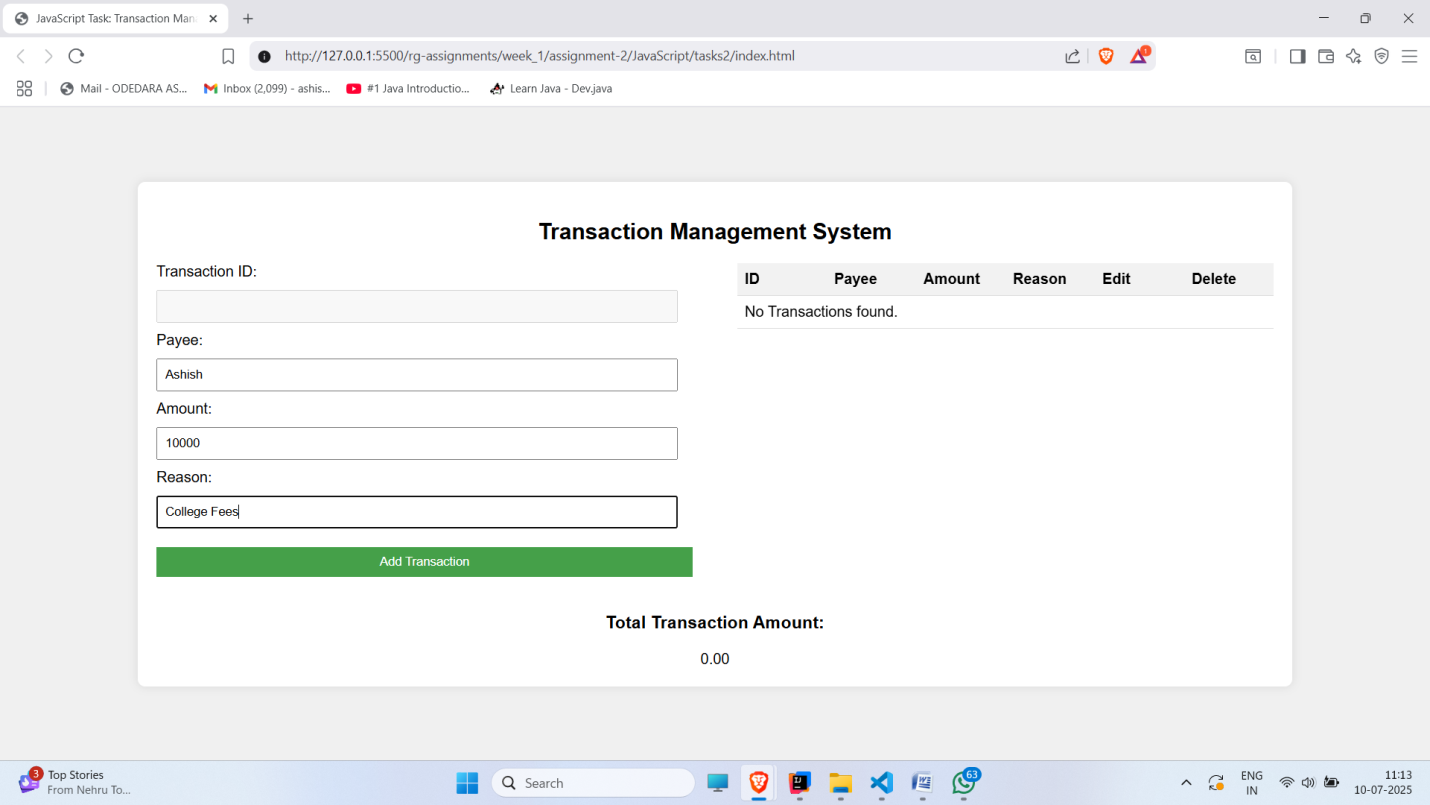
**styles.css :**

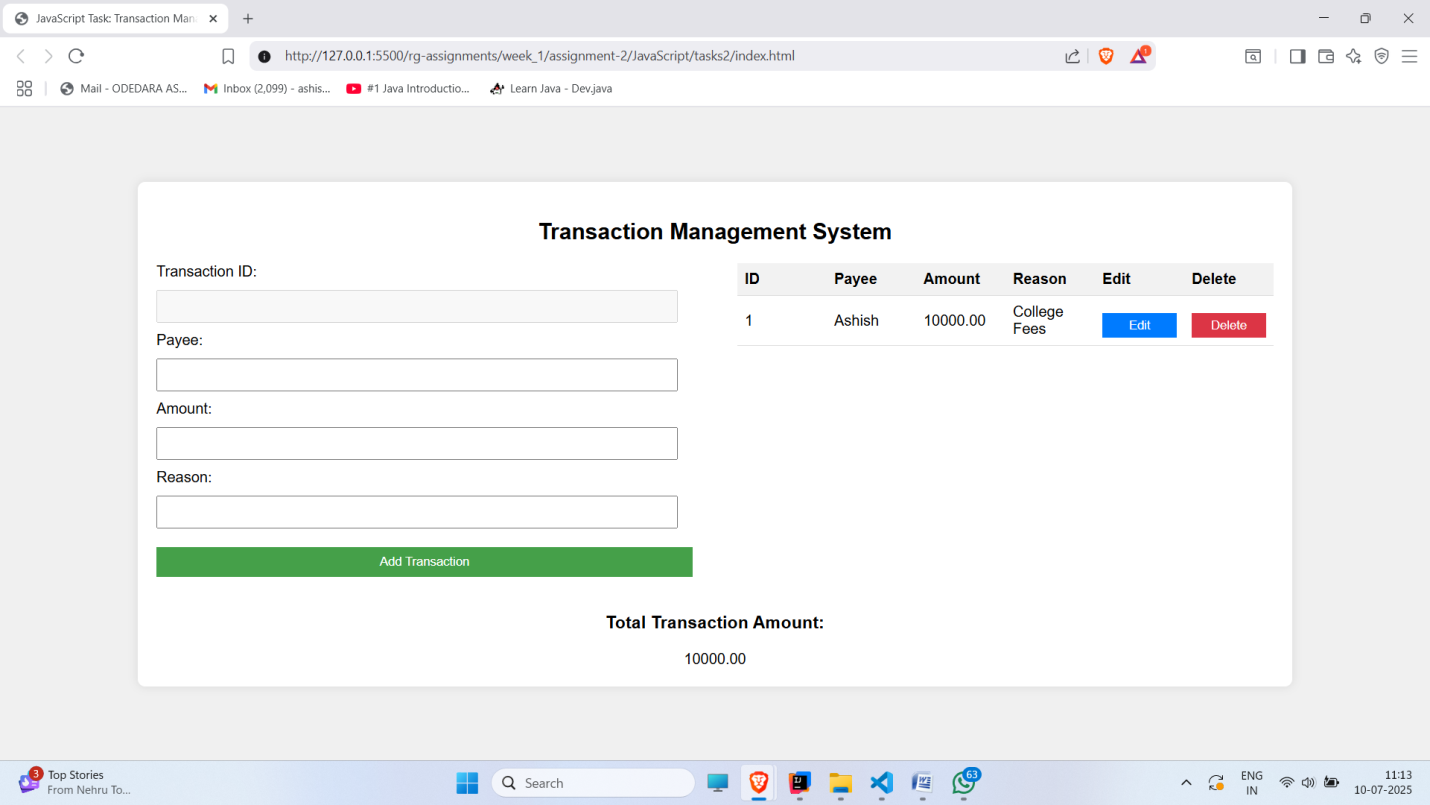
body { font-family: Arial, sans-serif; background-color: #f0f0f0; display: flex; justify-content: center; align-items: center; height: 100vh; margin: 0;}.container { background-color: #fff; padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); width: 1200px; /\* Increased width \*/ text-align: center;}h2 { margin-bottom: 20px;}.content { display: flex; justify-content: space-between;}.form-container, .table-container { width: 48%;}form { margin-bottom: 20px; text-align: left;}input[type="text"], input[type="number"] { padding: 8px; margin: 10px 0; width: calc(100% - 16px); box-sizing: border-box;}button { padding: 8px; background-color: #4CAF50; color: white; border: none; cursor: pointer; width: 100%; margin-top: 10px;}button:hover { background-color: #45a049;}.table-container { max-height: 400px; /\* Set a maximum height \*/ overflow-y: auto; /\* Enable vertical scrolling \*/}table { width: 100%; border-collapse: collapse; table-layout: fixed; /\* Ensure cells do not expand beyond table width \*/ word-wrap: break-word; /\* Ensure long text wraps within cells \*/}th, td { padding: 8px; text-align: left; border-bottom: 1px solid #ddd; overflow-wrap: break-word; /\* Ensure long text breaks within cells \*/}th { background-color: #f2f2f2;}tbody tr:hover { background-color: #f5f5f5;}.edit-button { background-color: #007BFF; /\* Blue color for edit button \*/ color: white; border: none; padding: 5px 10px; cursor: pointer;}.edit-button:hover { background-color: #0056b3; /\* Darker blue on hover \*/}.delete-button { background-color: #DC3545; /\* Red color for delete button \*/ color: white; border: none; padding: 5px 10px; cursor: pointer;}.delete-button:hover { background-color: #c82333; /\* Darker red on hover \*/}

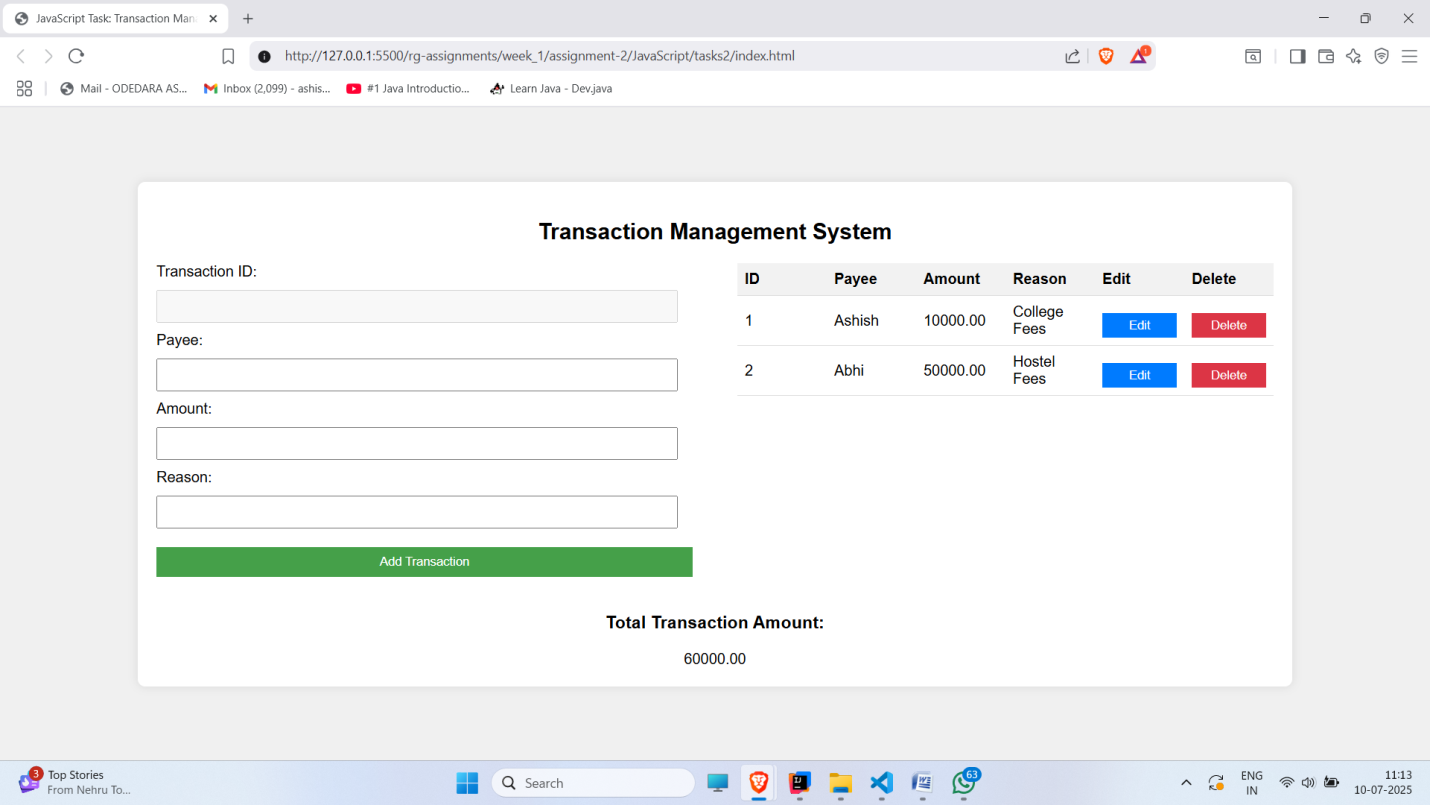
**script.js**

// task1: Complete the below statement to initialize the variable 'transactions' as an empty array.let transactions = [];let isEditing = false;let editingTransactionId = null;document.getElementById('transactionForm').addEventListener('submit',async function (event) { event.preventDefault(); // Prevent default form submission // task2: Retrieve 'payee', 'amount', and 'reason' values from form inputs using document.getElementById(). const payee = document.getElementById('payee').value; const amount = parseFloat(document.getElementById('amount').value); const reason = document.getElementById('reason').value; if (payee === '' || amount <= 0 || isNaN(amount)) { alert('Please enter valid transaction details or check your script code.'); return; } if (isEditing) { updateTransaction(editingTransactionId, payee, amount, reason); isEditing = false; editingTransactionId = null; document.querySelector('button[type="submit"]').textContent = 'Add Transaction'; await updateTotalTransactionAmount(); } else { addTransaction(payee, amount, reason); } clearForm(); updateTransactionTable(); await updateTotalTransactionAmount();});function addTransaction(payee, amount, reason) { // Note: Generated a unique 'id' for each new transaction const id = transactions.length > 0 ? transactions[transactions.length - 1].id + 1 : 1; // task3: Complete the 'newTransaction' object by replacing the placeholder values with appropriate data. const newTransaction = { id: id, payee: payee, amount: amount, reason: reason }; // task4: Complete the below statement to add the 'newTransaction' object into the 'transactions' array using the 'push' method. transactions.push(newTransaction);}function updateTransaction(id, payee, amount, reason) { // task5: Complete the below statement to find the transaction object in the 'transactions' array that matches the provided 'id' parameter. const transaction = transactions.find(t => t.id === id); if (transaction) { transaction.payee = payee; transaction.amount = amount; transaction.reason = reason; }}function editTransaction(id) { const transaction = transactions.find(t => t.id === id); if (transaction) { document.getElementById('payee').value = transaction.payee; document.getElementById('amount').value = transaction.amount; document.getElementById('reason').value = transaction.reason; isEditing = true; editingTransactionId = id; document.querySelector('button[type="submit"]').textContent = 'Edit Transaction'; }}async function deleteTransaction(id) { const index = transactions.findIndex(t => t.id === id); if (index !== -1) { // task6: Complete the below statement to remove a transaction from the 'transactions' array by the index value transactions.splice(index, 1); updateTransactionTable(); await updateTotalTransactionAmount(); }}// task7: Convert below function 'updateTotalTransactionAmount' to use async/await./\* a. Convert the 'updateTotalTransactionAmount' function to an async function. b. Update all occurrences of 'updateTotalTransactionAmount' across the script to use await and ensure their calling functions are marked as async.\*/function updateTotalTransactionAmount() { let totalAmount = 0; transactions.forEach(transaction => { totalAmount += transaction.amount; }); document.getElementById('totalTransactionAmount').textContent = totalAmount.toFixed(2);}function updateTransactionTable() { const tbody = document.querySelector('#transactionTable tbody'); tbody.innerHTML = ''; if (transactions.length === 0) { const noTransactionMessage = document.createElement('tr'); noTransactionMessage.innerHTML = `<td colspan="6">No Transactions found.</td>`; tbody.appendChild(noTransactionMessage); } else { transactions.forEach(transaction => { const row = document.createElement('tr'); row.innerHTML = ` <td>${transaction.id}</td> <td>${transaction.payee}</td> <td>${transaction.amount.toFixed(2)}</td> <td>${transaction.reason}</td> <td><button class="edit-button" onclick="editTransaction(${transaction.id})">Edit</button></td> <td><button class="delete-button" onclick="deleteTransaction(${transaction.id})">Delete</button></td> `; tbody.appendChild(row); }); }}function clearForm() { document.getElementById('payee').value = ''; document.getElementById('amount').value = ''; document.getElementById('reason').value = '';}**Output :**









Repository : <https://github.com/AshishOdedra/rg-assignments/tree/main/week_1/assignment-2/JavaScript/tasks2>