Step1: Create HTML CSS files with proper given ID

index.html > html

```html
<html lang="en">
<body>

  <div class="container">
    <h1>Expense Tracker</h1>
    <form id="expense-form">
      <input type="text" id="expense-name" placeholder="Expense name" required />
      <input type="number" id="expense-amount" placeholder="Amount" required />
      <button type="submit" class="submit-btn">Add Expense</button>
    </form>


    <h2>Expenses</h2>
    <ul id="expense-list">
      <!-- Expenses will be dynamically added here -->
    </ul>


    <div id="total">
      <h3>Total: $<span id="total-amount">0.00</span></h3>
    </div>
  </div>



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

</html>
```

Step2: Create JS file(script.js)

Step3: document.eventListener(DOMContentLoaded, arrow function)

Step4: get all IDs in script.js with proper variable names

JS script.js > ...

```javascript

document.addEventListener('DOMContentLoaded', () => {

  const expenseForm = document.getElementById("expense-form");
  const expenseName = document.getElementById("expense-name");
  const expenseAmount = document.getElementById("expense-amount");

  const expenseListDisplay = document.getElementById("expense-list");
  const totalAmountDisplay = document.getElementById("total-amount");




})
```

Step5: **Declare expense as an empty Array + Declare totalAmount = calculateTotal();**

```
let expenses = [];
let totalAmount = calculateTotal();
```

Step6: **Add event listener on form, prevent default, access & store expense Name and expense Amount in a variable.** *(number from input arrives in **string** format ➔ convert it to **number** or **float**)*

```
expenseForm.addEventListener('submit', (e) => {
  e.preventDefault();

  const name = expenseName.value.trim();
  const amount = parseFloat(expenseAmount.value.trim());

```

Step7: **Check on name-input & amount-input (if conditional)**
(expense not blank, amount not NaN & amont > 0)

```
  if (expenseAmount.value !== "" && !isNaN(amount) && amount > 0){

  }
```

Step8: **Create an *newExpense Object* inside if-conditional & push on expense array**

```
  if (expenseAmount.value !== "" && !isNaN(amount) && amount > 0) {
    const newExpense = {
      id: Date.now(),
      name,
      amount,
    };

    expenses.push(newExpense);
  }
```

```
(2) [{…}, {…}]                                          script.js:35
  0: {id: 1743207671636, name: 'Bijuli', amount: '20'}
  1: {id: 1743207691101, name: 'Momo', amount: '170'}
  length: 2
  [[Prototype]]: Array(0)
```

Step9: Clear input fields ➔ at the end of expenseForm add event listener

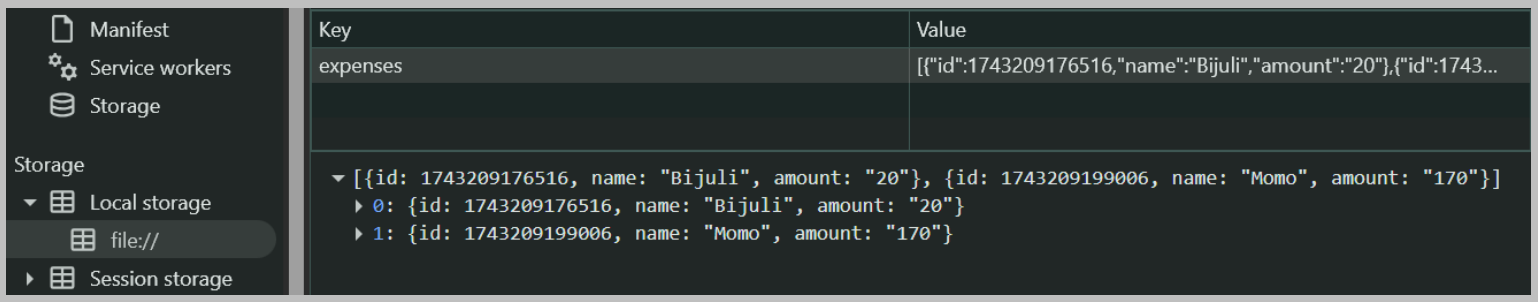
```
35      expenseName.value = "";
36      expenseAmount.value = "";
37
38  })
```

Step10: Define saveExpenseToLocal function and call in side the submit event

save the array in local storage, before it --- do JSON.stringify(array)

```
32      expenses.push(newExpense);
33      saveExpensesToLocal();
34    }
```

```
47  function saveExpensesToLocal() {
48    localStorage.setItem('expenses', JSON.stringify(expenses))
49  }
```



Step11: Work on Total Expenses [Calculate Total ➔ Display Total]

Step12: Define calculateTotal function with return and store it in totalAmount variable

```
13  let expenses = [];
14  let totalAmount = calculateTotal();
15
```

*(do Calculate total amount using **array.reduce()** method)*

```
52  function calculateTotal() {
53    return expenses.reduce((sum, eachExpenses) => (sum + eachExpenses.amount), 0)
54  }
```

Step13: Define displayTotal() and call it inside the form-submit event

```
30      expenses.push(newExpense);
31      saveExpensesToLocal();
32      displayTotal();
33    }
```

(bring the calculateTotal first then show it in HTML)

```
59  function displayTotal() {
60    totalAmount = calculateTotal();
61    totalAmountDisplay.textContent = totalAmount.toFixed(2);
62  }
```

Step14: To save data permanently in local storage [do getItems & re-store in the save array]

(re-convert the string data from local-storage in to JS object to re-assign in the same array)

```
12
13  let expenses = JSON.parse(localStorage.getItem('expenses')) || [];
14  let totalAmount = calculateTotal();
```

Step15: Display all data *[Define displayList() & apply arr.forEach() for each item of expense array]*

```
69  function displayList() {
70    expenseListDisplay.innerHTML = "";
71    expenses.forEach((exp) => {
72
73
74    });
75  }
76
```

Step16: Declare li *then* add inner HTML in *li*

```
69  function displayList() {
70    expenseListDisplay.innerHTML = "";
71    expenses.forEach((exp) => {
72      const li = document.createElement('li');
73      li.innerHTML = `;
74
75    });
76  }
```

Step17: Pass back Quots for *li.innerHTML* with ➔ *expense name, amount & delete button inside*

```
73      li.innerHTML = `
74      ${exp.name}: $${exp.amount.toFixed(2)}
75      <button>Delete</button>
76      `;
77
78  });
```

Step18: Append the li in HTML to display [*expenseListDisplay.appendChild(li)*] *inside forEach()*

```
69  function displayList() {
70    expenseListDisplay.innerHTML = "";
71    expenses.forEach((exp) => {
72      const li = document.createElement('li');
73      li.innerHTML = `
74      ${exp.name}: $${exp.amount.toFixed(2)}
75      <button>Delete</button>
76      `;
77      expenseListDisplay.appendChild(li);
78
79
80    });
81
82  }
```

Step19: To display all data constantly after-submit and while-pageload *[call displayList()]*

*(define displayList() function and call it **inside and outside the submit event**)*

```
18  displayList();
19  expenseForm.addEventListener('submit', (e) => {
20    e.preventDefault();
21
22    const name = expenseNameInput.value.trim();
23    const amount = parseFloat(expenseAmountInput.value.trim());
24
25
26    if (name !== "" && !isNaN(amount) && amount > 0) {
27      const newExpense = {
28        id: Date.now(),
29        name,
30        amount,
31      };
32
33      expenses.push(newExpense);
34      saveExpensesToLocal();
35      displayTotal();
36      displayList();
```

Step20: Delete Items [*add click event on delete button with if-clause*]

```
87  //delete items ==> define click event on 'Delete'
88  expenseListDisplay.addEventListener('click', (e) => {
89    if (e.target.tagName === 'BUTTON') {
90      //write delete code here
91    }
92  })
93
```

Step21: Add data-id attribute on delete button in displayList() function

```
69  function displayList() {
70    expenseListDisplay.innerHTML = "";
71    expenses.forEach((exp) => {
72      const li = document.createElement('li');
73      li.innerHTML = `
74      ${exp.name}: $${exp.amount.toFixed(2)}
75      <button data-id="${exp.id}">Delete</button>
76      `;
77      expenseListDisplay.appendChild(li);
78
79
80    });
```

Step22: Declare variable of id by [getting attribute of delete button] for clicked expenses ➔inside the expenseListDisplay-clicked-event , *#note: convert id into integer*

```
87  //delete items ==> define click event on 'Delete'
88  expenseListDisplay.addEventListener('click', (e) => {
89    if (e.target.tagName === 'BUTTON') {
90      const clickedId = parseInt(e.target.getAttribute('data-id')); // get & store id of clicked item
91    }
92  })
```

Step23: Apply arr.filter() ➔ filter in the same array ➔ save only which [clicked-id !== expense-id]

```
87  //delete items ==> define click event on 'Delete'
88  expenseListDisplay.addEventListener('click', (e) => {
89    if (e.target.tagName === 'BUTTON') {
90      const clickedId = parseInt(e.target.getAttribute('data-id')); // get & store id of clicked item
91      expenses = expenses.filter(ex => ex.id !== clickedId);
92    }
93  })
```

Step24: Then after click on Delete *(filter happens)* saveToLocalStorage, displayList & displayTotal

```
87  //delete items ==> define click event on 'Delete'
88  expenseListDisplay.addEventListener('click', (e) => {
89    if (e.target.tagName === 'BUTTON') {
90      const clickedId = parseInt(e.target.getAttribute('data-id')); // get & store id of clicked item
91      expenses = expenses.filter(ex => ex.id !== clickedId);
92    }
93
94    saveExpensesToLocal();
95    displayList();
96    displayTotal();
97
98  })
99
```