Here’s the **complete implementation** for the **Fitness Tracker - Simple Activity Tracking Application** as per your project requirements and test case conditions.

**Project Structure**

fitness-tracker/

│

├── index.html

├── script.js

└── style.css (optional)

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fitness Tracker</title>

</head>

<body>

<h2>Fitness Tracker</h2>

<div>

<label for="activity">Activity Description:</label>

<input type="text" id="activity" required placeholder="Enter activity description">

<label for="duration">Duration (minutes):</label>

<input type="number" id="duration" required min="1" placeholder="Enter duration">

<button onclick="addActivity()">Add Activity</button>

</div>

<br>

<table border="1" id="activityTable">

<thead>

<tr>

<th>Activity Description</th>

<th>Duration (minutes)</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<h3>Total Duration: <span id="totalDuration">0</span> minutes</h3>

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

</body>

</html>

**script.js**

let totalDuration = 0;

function addActivity() {

const activityInput = document.getElementById("activity");

const durationInput = document.getElementById("duration");

const activity = activityInput.value.trim();

const duration = parseInt(durationInput.value);

if (activity === "" || isNaN(duration) || duration <= 0) {

alert("Please enter a valid activity and duration.");

return;

}

const tableBody = document.querySelector("#activityTable tbody");

const newRow = document.createElement("tr");

const activityCell = document.createElement("td");

activityCell.textContent = activity;

const durationCell = document.createElement("td");

durationCell.textContent = duration;

newRow.appendChild(activityCell);

newRow.appendChild(durationCell);

tableBody.appendChild(newRow);

totalDuration += duration;

document.getElementById("totalDuration").textContent = totalDuration;

// Reset input fields

activityInput.value = "";

durationInput.value = "";

}

**(Optional) style.css**

You can include a simple style file to make it more presentable (not required for test cases):

body {

font-family: Arial, sans-serif;

margin: 20px;

}

table {

width: 50%;

border-collapse: collapse;

}

th, td {

padding: 8px;

text-align: center;

}

button {

margin-left: 10px;

padding: 5px 10px;

cursor: pointer;

}

**How to Run the Project**

1. Open your terminal.
2. Run:
3. npm start
4. When the server starts, open the link shown (should be **port 8081**).
5. The web page will load showing the **Fitness Tracker**.

**How to Run the Test Cases**

* Click on the **“Run Test Case”** button in your environment.
* Make sure the app is running on **port 8081**.
* All test cases will pass if:
  + You used the exact text **“Add Activity”** for the button.
  + You have a <h3> displaying **Total Duration: X minutes**.
  + Inputs are validated correctly.
  + Table displays both **activity description** and **duration**.

Would you like me to add a small CSS enhancement version (for a cleaner look without breaking test cases)?

Top of Form

Bottom of Form