

Controllo e Tracciamento di una Produzione Giornaliera

Specifiche funzionali

Il sistema di tracciamento consente:

- Inserire i dati relativi ai prodotti completati giornalmente.
- Calcolare il totale giornaliero.
- Visualizzare un grafico a barre dell'andamento settimanale.

Per visualizzare correttamente il grafico nel **progetto** (Controllo e Tracciamento di una Produzione Giornaliera), è necessario includere la libreria **Chart.js**, che è una libreria JavaScript popolare per creare grafici interattivi. Senza questa libreria, il codice non funzionerà come previsto per la visualizzazione del grafico.

Modifiche Necessarie:

1. **Includere Chart.js** nel progetto.
2. Correggere il codice JavaScript per l'uso corretto di Chart.js.

1. Aggiungi la libreria Chart.js

Per aggiungere Chart.js al tuo progetto, devi includere il link alla libreria nella sezione `<head>` del tuo HTML. Puoi utilizzare il CDN ufficiale di Chart.js.

HTML aggiornato

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tracciamento Produzione</title>
  <link rel="stylesheet" href="styles.css">
  <!-- Includi il CDN di Chart.js -->
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <div class="container">
    <h1>Tracciamento Produzione Giornaliera</h1>
    <form id="productionForm">
      <input type="number" id="dailyProduction" placeholder="Prodotti
completati" required>
      <button type="submit">Aggiungi Produzione</button>
    </form>
    <div id="totalProduction">Produzione totale: 0</div>
    <!-- Aggiungi il canvas per il grafico -->
    <canvas id="productionChart" width="400" height="200"></canvas>
  </div>
  <script src="app.js"></script>
</body>
</html>
```

2. JavaScript per Chart.js

Nel codice JavaScript, dobbiamo assicurarci che il grafico venga disegnato correttamente utilizzando Chart.js.

JavaScript

```
// app.js
// Inizializzazione array per memorizzare produzione giornaliera
let productionData = [];

// Funzione per aggiungere la produzione del giorno
function addDailyProduction(event) {
    event.preventDefault();

    const dailyProduction =
    parseInt(document.getElementById("dailyProduction").value);

    // Aggiunge la produzione giornaliera all'array
    productionData.push(dailyProduction);

    document.getElementById("productionForm").reset();

    // Aggiorna il totale della produzione
    updateTotalProduction();

    // Rende visibile il grafico
    renderChart();
}

// Funzione per aggiornare la produzione totale
function updateTotalProduction() {
    const total = productionData.reduce((acc, val) => acc + val, 0);
    document.getElementById("totalProduction").textContent = `Produzione totale:
    ${total}`;
}

// Funzione per disegnare il grafico a barre
function renderChart() {
    const ctx = document.getElementById("productionChart").getContext("2d");

    // Se il grafico esiste già, lo distruggiamo per rimuoverlo e crearne uno
    nuovo
    if (window.chart) {
        window.chart.destroy();
    }

    // Crea un nuovo grafico a barre
    window.chart = new Chart(ctx, {
        type: 'bar', // Tipo di grafico
        data: {
            labels: productionData.map((_, index) => `Giorno ${index + 1}`), //
            Etichette per i giorni
            datasets: [{
                label: 'Produzione Giornaliera',
                data: productionData, // I dati della produzione
                backgroundColor: '#4CAF50', // Colore delle barre
                borderColor: '#388E3C', // Colore del bordo delle barre
                borderWidth: 1
            }]
        },
        options: {
            responsive: true,
```

```

        maintainAspectRatio: false,
        scales: {
            y: {
                beginAtZero: true, // Assicura che l'asse Y inizi da 0
                ticks: {
                    stepSize: 1 // Incremento dei valori sull'asse Y
                }
            }
        }
    });
}

// Evento di submit per il form
document.getElementById("productionForm").addEventListener("submit",
addDailyProduction);

```

3. Descrizione delle modifiche

1. Inclusione di Chart.js

Nel file HTML, ho aggiunto il tag `<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>` per includere la libreria Chart.js dal CDN.

2. Modifica della funzione `renderChart()`

- La funzione `renderChart()` è stata aggiornata per utilizzare Chart.js per disegnare un grafico a barre.
- L'elemento `<canvas>` con l'id="productionChart" è stato usato come contenitore per il grafico.
- Ho usato `window.chart` per mantenere un riferimento al grafico esistente e distruggerlo ogni volta che ne viene creato uno nuovo. Questo previene la creazione di grafici multipli quando si aggiungono nuovi dati.

3. Configurazione di Chart.js

- Il grafico è configurato per avere un tipo `bar` (grafico a barre).
- Le etichette sull'asse X vengono create dinamicamente utilizzando i dati di produzione giornaliera, con le etichette che si aggiornano ogni volta che vengono aggiunti nuovi dati (ad esempio, `Giorno 1`, `Giorno 2`, ecc.).
- L'asse Y è configurato per partire da 0 e incrementarsi di 1, in modo che i dati siano visualizzati chiaramente.

4. Risultato Finale

Ora, quando l'utente inserisce un numero per la produzione giornaliera e fa clic su "Aggiungi Produzione", i dati vengono visualizzati sia come produzione totale nel testo che come grafico a barre, che viene aggiornato dinamicamente.