

# Dashboard

## 1. Dashboard

## 2. Dynamic Dashboards

### JS: (matchCountryCapital.js)

```
import { LightningElement } from 'lwc';

const countries = [
  { id: 1, name: 'India', capital: 'New Delhi' },
  { id: 2, name: 'USA', capital: 'Washington D.C.' },
  { id: 3, name: 'Japan', capital: 'Tokyo' }
];

const capitals = [
  { id: 1, name: 'New Delhi' },
  { id: 2, name: 'Washington D.C.' },
  { id: 3, name: 'Tokyo' }
];

export default class MatchCountryCapital extends LightningElement {

  countries=countries ;
  capitals= capitals ;
  resultMessage = '';

  // Handle drag start event
  handleDragStart(event) {
    event.dataTransfer.setData('capital', event.target.dataset.capital);
  }

  // Allow drop
```

```

allowDrop(event) {
  event.preventDefault();
}

// Handle drop event
handleDrop(event) {
  event.preventDefault();
  const draggedCapital = event.dataTransfer.getData('capital');
  const countryName = event.target.dataset.id;

  // Find the country and check if the dropped capital is correct
  const country = this.countries.find(c => c.name === countryName);
  if (country.capital === draggedCapital) {
    this.resultMessage = `Correct! ${draggedCapital} is the capital of ${countryName}`;
  } else {
    this.resultMessage = `Incorrect! ${draggedCapital} is not the capital of ${countryName}`;
  }
}
}

```

Html: (matchCountryCapital.html)

```

<template>
  <div class="container">
    <div class="countries">
      <h3>Countries</h3>
      <template for:each={countries} for:item="country">
        <div key={country.id} class="drop-zone"
          data-id={country.name}
          ondragover={allowDrop}
          ondrop={handleDrop}>
          {country.name}
        </div>
      </template>
    </div>
    <div class="result">
      {resultMessage}
    </div>
  </div>
</template>

```

```

        </template>
    </div>

    <div class="capitals">
        <h3>Capitals</h3>
        <template for:each={capitals} for:item="capital">
            <div key={capital.id} class="draggable" draggable
                data-id={capital.name}
                ondragstart={handleDragStart}>
                {capital.name}
            </div>
        </template>
    </div>
</div>

<template if:true={resultMessage}>
    <div class="result">{resultMessage}</div>
</template>
</template>

```

CSS: (matchCountryCapital.css)

```

.container {
    display: flex;
    justify-content: space-around;
    margin-top: 20px;
}

.countries, .capitals {
    width: 200px;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

```

```

}

.draggable {
  background-color: #f4f4f4;
  padding: 10px;
  margin-bottom: 10px;
  cursor: pointer;
}

.drop-zone {
  background-color: #eef;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px dashed #888;
}

.result {
  margin-top: 20px;
  font-weight: bold;
  color: green;
}

```

O/p:

[https://www.linkedin.com/posts/hari-sankar-r-5353oe\\_happylearninglwc-lwc-lwc-activity-725110222728511488-tfK5?utm\\_source=share&utm\\_medium=member\\_desktop](https://www.linkedin.com/posts/hari-sankar-r-5353oe_happylearninglwc-lwc-lwc-activity-725110222728511488-tfK5?utm_source=share&utm_medium=member_desktop)

## 1. Finance and Accounting Training

Scenario: Match financial terms to their definitions or processes to accounting practices.

Purpose: Help finance professionals or trainees match financial transactions with the correct practices or terminologies.

## 2. Procurement and Vendor Management

Scenario: Match vendors to services/products or procurement steps to compliance procedures.

Purpose: Simplify vendor management by ensuring the correct vendor is matched with the corresponding product or service, and procurement steps adhere to compliance regulations.

## 3. IT and Software Development

Scenario: Match software development tasks to development stages or security vulnerabilities to solutions.

Purpose: Facilitate the planning and management of software development projects or cybersecurity protocols.

## 4. Event Planning and Management

Scenario: Match event components (e.g., catering, venue, guest lists) to the corresponding event type or service provider.

Purpose: Help event planners organize events by ensuring the right resources are allocated to specific event needs.

## 5. Retail Training and Store Management

Scenario: Match products to store sections or customer queries to in-store services.

Purpose: Train retail staff to better understand store layouts and where specific products or services are located.

What's the output of the code below - and why?

```
const a = [7, 2, 5];
const b = [4, 9, 1];

const c = [...a, ...b].map(num => {
  return num % 2 === 0 ? num * 2 : num * 3;
});
```

```
});
```

```
console.log(c);
```

output:

```
[]
```