# Exercise 7 - Table Manipulation

**Objective**: To master table column manipulation using pure JavaScript. You will dynamically hide/show table columns and restore the table to its original visibility.

**Instructions**:

1. **Setup**:

   ○ You given an HTML table element with an ID of `dataTable`.
   ○ There should be buttons that, when clicked, will call the `toggleColumn(index)` function, passing the column index as an argument.

2. **Toggling Column Visibility**:

   ○ **Function**: `toggleColumn(index)`

   ▪ **Purpose**: This function's primary role is to toggle the visibility of a specific column of the table.

   ▪ **Parameters**: The function accepts one argument:

   ▪ `index`: This is a zero-based integer that represents the column number you want to hide or show.

   ▪ **Steps**:

   1. **Fetching All Rows**: Use `document.querySelectorAll('#dataTable tr')` to grab all the table rows. The `querySelectorAll` function returns a NodeList of all elements that match the provided selector. Store all the table rows in a `rows` variable.
   2. **Loop through Rows**: Using the `forEach` method, iterate through the `rows` NodeList to access individual cells. The callback function you provide to `forEach` receives the current `row` as a parameter.
   3. **Accessing the Cell**: For every row, store the cell corresponding to the `index` in a `cell` variable using `row.cells[index]`.
   4. **Toggling Visibility**: Check the `display` style property of the cell:
      ▪ If it's `'none'` (meaning the column is currently hidden), set it to an empty string (`''`), making the column visible.

- If it's not `'none'` (meaning the column is currently visible), set it to `'none'`, hiding the column.

3. **Resetting the Table**:

- **Function**: `resetTable()`

  - **Purpose**: To reset all columns of the table to their original visible state.

  - **Steps**:

    1. **Fetching All Rows**: Similar to the toggle function, store all rows in a `rows` variable with `document.querySelectorAll('#dataTable tr')`.
    2. **Iterating Through Each Row**: Using the `forEach` method, iterate over each row.
    3. **Iterating Through Each Cell**: For each row, you want to go through every cell. To do this, convert the row's `cells` collection to an array using `Array.from(row.cells)` and then iterate over it using another `forEach`.
    4. **Restoring Visibility**: For each cell in the row, set its `display` style property to `''` (empty string). This ensures that every column in the table is visible.

4. **HTML**:

- Use the attached HTML code for this exercise.

5. **Testing**:

- **Visibility Toggle**:

  1. Load your web page containing the table.
  2. Click on one of the "Toggle" buttons. The associated column should either disappear or reappear based on its current state.
  3. Experiment by toggling different columns and observing the changes.

- **Reset Functionality**:

  1. After toggling a few columns, click the "Reset" button.
  2. The table should restore to its original state, with all columns visible.