**Team Budget Algorithm**

**Initialization:**

Create a class named TeamBudget with a constructor that initializes the budget to 0 and an empty array for deals.

**Add methods to the class:**

1. addDeal: Accepts vendor, expense, description, and date. Adds a deal to the deals array and updates the budget.

2. getBudget: Returns the current budget.

3. getDeals: Returns the array of deals.

Create an instance of the TeamBudget class named teamBudget.

**Form Submission Handling:**

1. Add an event listener to the form with the ID 'dealForm' for the 'submit' event.

2. Prevent the default form submission behavior.

3. Retrieve values from the form fields (vendor, expense, description, and date).

4. Validate the input.

5. If the input is valid, call teamBudget.addDeal with the obtained values to add a deal and update the Budget.

6.If the input is invalid, show an alert.

7. Reset the form fields.

**Update Budget Display Function:**

1. Create a function named updateBudgetDisplay.

2. Get references to the HTML elements for budget display, budget span, and create a list for deals.

3. Update the displayed budget value.

4. Create list items for each deal and append them to the list.

5. Clear previous content in the budget display and append the updated content.

**Webpack Configuration:**

1. Create a webpack.config.js file to configure Webpack.

2. Specify the entry point as 'app.js' and the output as 'bundle.js' in the 'dist' directory.

3. Optionally, set the 'mode' option to either 'development' or 'production'.