# Algorithm:

* Initialize:
  + Check if data exists in local storage (`localStorage.getItem("myData")`).
  + If not:
  + Initialize `myData` array with sample `Deal` objects.
  + Set `currentDealId` to the length of `myData`.
  + Store `myData` in local storage (`localStorage.setItem("myData", JSON.stringify(myData))`).
  + Else:
  + Retrieve `myData` from local storage.
  + Set `currentDealId` to the length of `myData`.
* Create Table from JSON:
  + Call `CreateTableFromJSON` function.
  + Clear existing table body content (`$('tbody').empty()`).
  + Parse `myData` from local storage.
  + Loop through each `Deal` in `myData`:
  + Append a table row with `Deal` data to the table body.
* Add New Deal:
  + Get input values for `clientName`, `projectName`, `projectManager`, and `projectCost`.
  + Clear input fields.
  + Create a new `Deal` object.
  + Add the new `Deal` to the `myData` array.
  + Increment `currentDealId`.
  + Clear local storage (`localStorage.clear()`).
  + Store updated `myData` in local storage.
  + Call `CreateTableFromJSON` to update the displayed table.
* Delete Deal:
  + Loop through each `Deal` in `myData`:
  + If `dealId` matches specified `dealId`:
  + Display confirmation.
  + If confirmed:
  + Remove the `Deal` from `myData`.
  + Clear local storage.
  + Store updated `myData` in local storage.
  + Call `CreateTableFromJSON` to update the displayed table.
* Update Project Costs:
  + Call `UpdateCost` function.
  + Initialize an empty array `ary`.
  + Loop through each row in the table body with class `update`:
  + Extract `dealId`, `client\_name`, `project\_name`, `project\_manager`, and updated `project\_cost`.
  + Create a new `Deal` object with updated cost.
  + Add the new `Deal` object to `ary`.
  + Clear local storage.
  + Store updated `ary` in local storage.
  + Call `CreateTableFromJSON` to refresh the table.