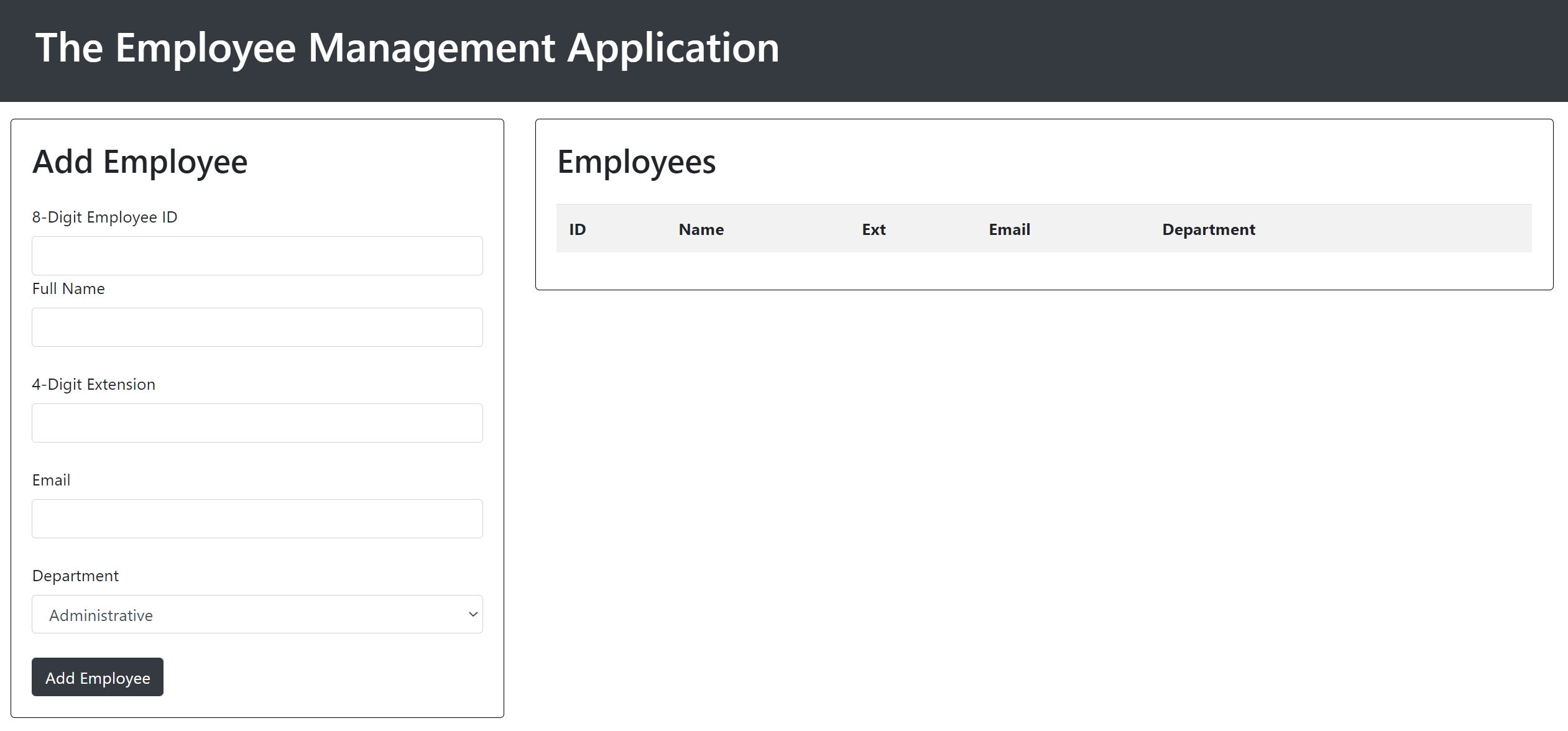
**Assignment 9**

**The Employee Management System**

In this assignment you will build on assignment 7. If you recall, in assignment 7, the user was able to add an employee, view that employee within the grid, and then delete an employee by clicking the delete button that appears to the right of each row. While this was a great learning opportunity as it relates to DOM scripting, it wasn’t realistic as it didn’t allow you to persist the employee data in any way. When the browser was closed, the data was gone and had to be reentered again. In this version of the assignment, you will modify the application to use arrays and then store the populated array within web storage (localStorage) in order to persist the data across browser sessions.

**The Interface**

As was the case in assignment 7, the UI has been created for you using Bootstrap and will look identical to assignment 7. You will not need to touch the HTML for this assignment. The example below outlines what the UI looks like.



Pay close attention to the table markup of this version of the assignment however. You will create the row structure differently in this assignment than what you did in assignment 7. More on this in a bit.

**Loading an Initial Set of Employees**

For this assignment you will use arrays to structure your data. You will need to create an initial array and populate it with at least 5 employees. So when the page loads, the grid should automatically populate with those initial 5 employees. The data for each employee should be structured so that you’re storing the employee ID (number), employee name (string), 4 digit extension (number), email (string), and department (string).

**Building the Grid**

In assignment 7 you used the table’s DOM methods insertRow(), insertCell(), and deleteRow() to manipulate the row structure for the table. In this assignment you will take a different approach. In this assignment, you’ll see the table includes a <tbody> tag. Use this tag as a ‘hook’ and rely on the innerHTML property to programmatically construct the row and cell structure for the table using a template literal string.

Considerations:

* This will be its own function. It will be called when the page loads, when an employee is added, and when an employee is deleted.
* Use a for / of loop here to loop through the array and build each row in code
* Use the appendChild() method to append the constructed row to the <tbody> tag.

**Adding and Removing Data**

You’ll need two separate functions for adding and removing employees. Remember to pass the array into these functions and then use array specific methods to add / remove employees from the array. Don’t forget to call the function to build the grid once an employee has been added or removed.

Considerations:

* When a new employee is added, you’ll need to create a new array. This array is what will be added into the main employees array.

**Storing Data**

Use web storage (localStorage) to persist the array. It should be stored when the page loads / when the grid is built, when a new employee is added, and when an employee is deleted. When the page loads, make sure to check to see if the object exists in storage before you attempt to extract the data from storage.

Considerations

* You’ll need to use JSON.stringify() to store the array in storage.
* You’ll need to use JSON.parse() to retrieve the array from storage.