***Exercise details***

*1. Create a web page with the below details*

*a) Employees, Departments as listing page. (****rem*** *you need to have separate navigation via menu to employees listing page and then for department listing page only)*

*b) Use Session Storage as storage medium/persistence medium. (in conjunction with JS Array/classes)*

*c) Can use JS array/classes for storage of data.*

*d) Implement Pagination, Sorting, Searching (user can search on name & age for Employee and DeptName for Department, try to use a generic logic for the same).*

*e) Should have a Add/Edit for Employees and Department (on click of edit you should navigate to the respective Add/Edit html page)*

*f) Make use of local-storage/session-storage to store the temp-data when user is doing addition or modification, so that in case of power failure or similar issue no data should be lost. (you need to make use of page/DOM events, few keywords to search isDirty, change etc)*

*g) The listing page should have facility to delete records upon confirmation, the list should refresh after successful deletion.*

*H) Assuming that you are on page –2 of Employees listing page, from there if the user clicks on Edit you should navigate to respective page Add/Edit and at the same time you should save the state for every listing page and update it on page navigation, so that when we click the back button and we are navigating to the respective index/listing page we land on the same page number that is ‘2’.*

***NOTE: Please  go through all parts before start of the exercise, so that you can collate all points/data prior to your start with minimum iterations.***

1.Create sidebar using HTML, CSS or Bootstrap

1.1 Create two menus Employee and Departments (like clients, instruments etc you can follow the below images only for illustrations).

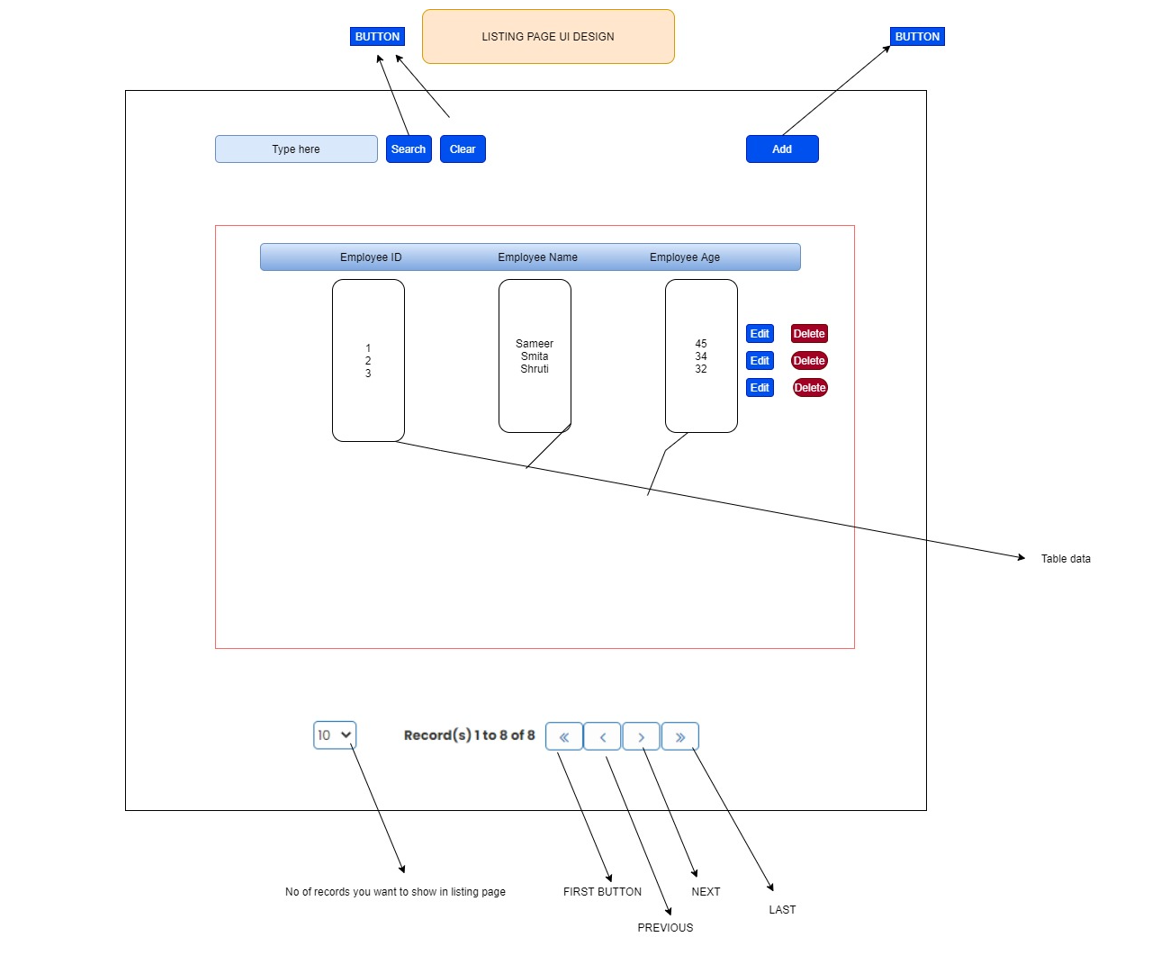
**MENU (for illustration)**

A screenshot of a computer screen

Description automatically generated with medium confidence

1.2 When you click the Employee or Department menu it will navigate to that Index page (Listing Page).

1.3. Inside listing page show ADD, EDIT, DELETE, SEARCH & CLEAR BUTTON. Then below show the Pagination part. Now I will show you only Employee listing data same as you can create for Department (Columns are DeptId,DeptName)

z 

a. ADD Functionality : When you click the add button it will navigate to Add/Edit form. Inside add/edit form create three text box .

Age

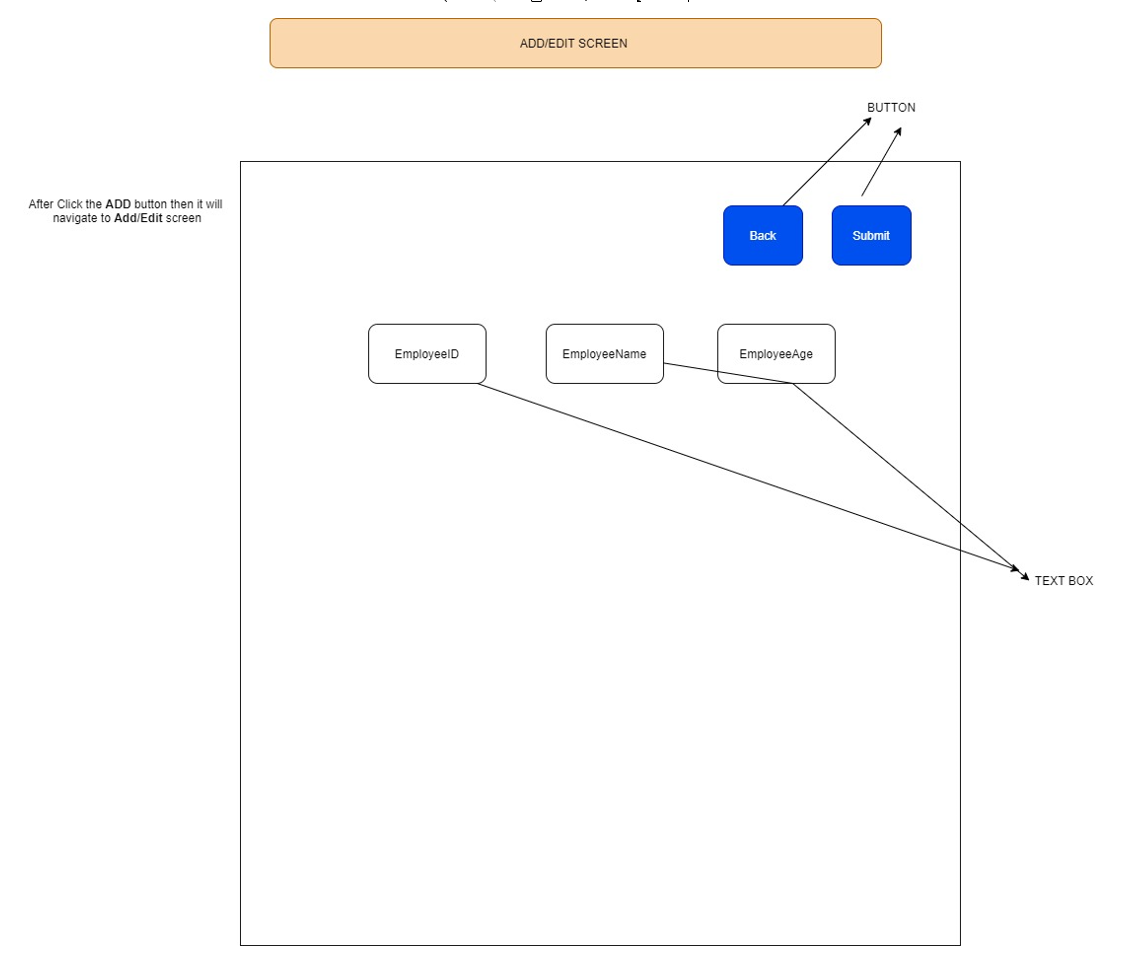
EmployeeName

Employee ID

EmployeeId:String

EmployeeName:String

Age: number



---> Those are the Input fields. After insert data then click the submit button. When you submit the data the submit method will call and that data will store in either session or local storage.

-->Then retrive that data from session or local storage then show in listing page.

***Main Requirements:***

***1.When we are in next page (page no-2) and we try to edit a record. After edit a record then we are clicking on Back button so it should navigate to the same page (page no-2). [****Please use relevant logic to store the state of that page and make sure you update that on every navigation****]***

***2. When we search the data, we want to search the specific row based on search applied on all relevant column. Whether it’s a string or integer. Eg. If the user wants to search by some number you need to detect if that column’s data is int or string)***

**LISTING PAGE (for illustration) :**

Table

Description automatically generated with medium confidence

**ADD/EDIT SCREEN (for illustration)**

Graphical user interface

Description automatically generated