**Component Name**: Dynamic Data Table for React Applications

**Component Functionality**:

Dynamic Table with following features:

1. Dynamic Table Header and Body Data based on any JSON Format Data.

2) Searching Data in Table

3) Pagination

4) Sorting Data based on each column Sort Type.

5) Limitation of Data per Page.

6) Easy Readable Format of Table Data in Mobile & Tablet View.

**Usage Areas**:

Reusable Component in any React Applications.

Ex: E- Commerce Application, Banking and Any Organisation Employee and Tabular Data.

**Component Technical details**:

* React version 16.8 above
* ES6

There is multiple component in the current app and main component is DataTable component.

**Component Name**: DataTable

**Folder & Child Components**: contents (TableHeader, TableBody), features(PageLimit, Pagination, SearchFilter), actions(sorting, searchfilter, getHeader functionalities), Reducer (for maintaining the global state based on functionalities) and styles folder.

All the business logic is present under this DataTable Component and features and action Folder.

**How this component can be integrated with other Applications**:

1. Install the React version **16.8** above.

2. Download the **DataTable Folder** from global components:

global->components->DataTable

2. Required **data** as props in JSON Format.

3. Provide the **pageSize** as a props to Dynamic Table Component for populating the data in table and pagination lists.

**Usage Example**:

 <DataTable

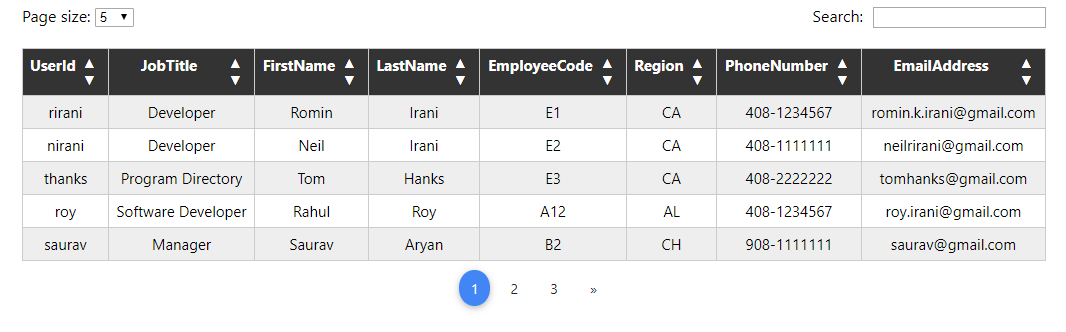
data ={data}

pageSize="5"

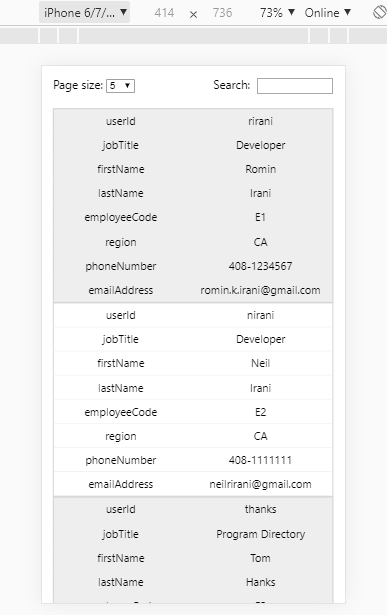
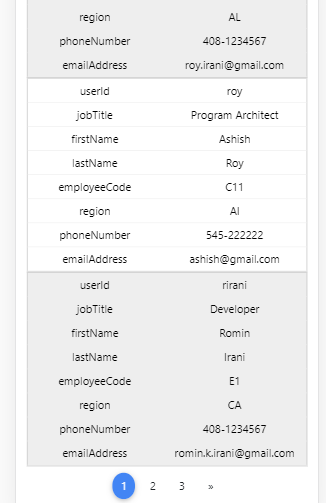
/>

**Screenshots**:

1. **Initial Screen : Desktop View (Dynamic Header list and Table Body Data based on JSON format which is coming as props) -**

****

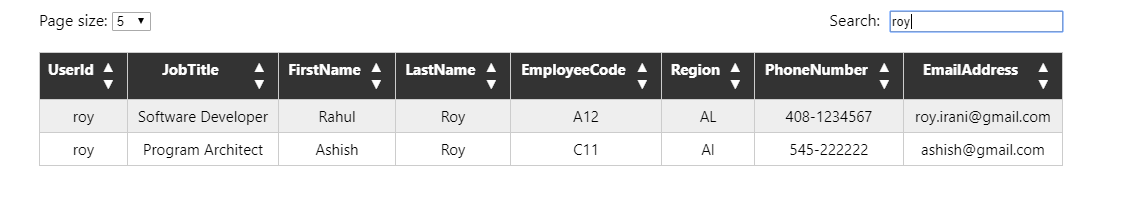
**Mobile View:**

** **

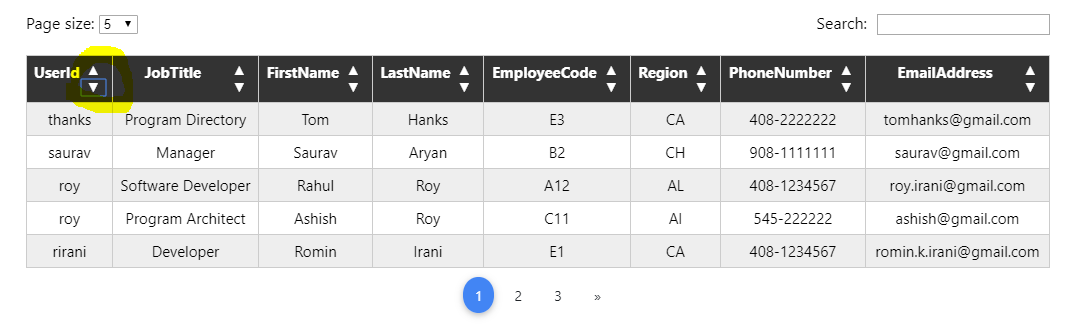
1. **If Data is less than page size which you selected from dropdown then no pagination list will come below of table and the pages looks like that:**



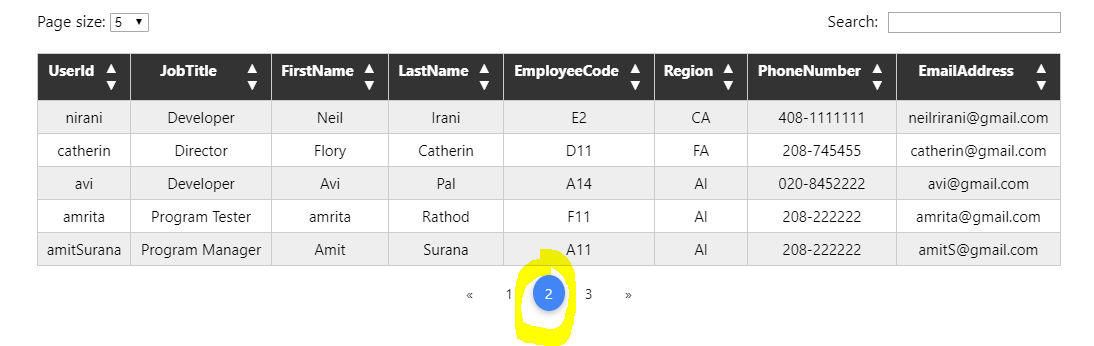
3. Searching the data from all lists of data which is not showing in current screen:

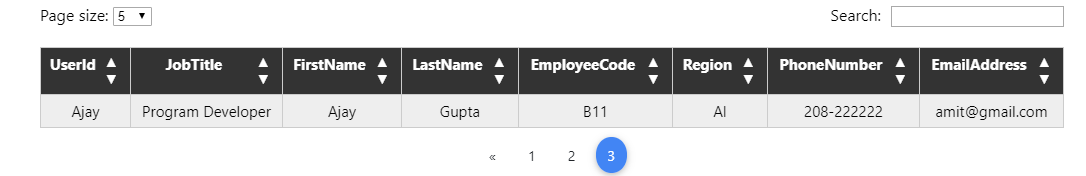


4. Sorting the data based on each column up and down button in ascending and descending order format:

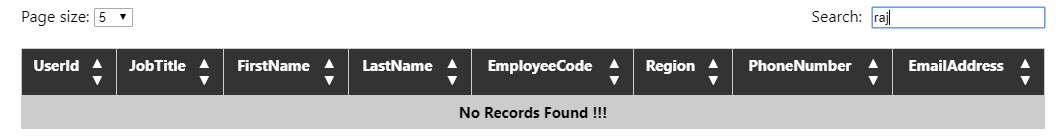


5. Pagination features based on page size and based on first and last pagination list prev and next button will be act:

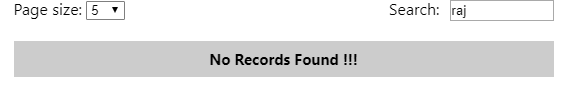




6. DeskTop View on Data Failure based on searching and Page Load:



7. Mobile View on Data Failure based on searching and Page Load:

****