

## Develop Employee Management System

- In Real world scenarios, many small to large scale organizations use proprietary or in-house developed EMS to manage employee's data. With your knowledge about MERN stack technology you will develop EMS. EMS contains following features:
  - User should be able to view list of employees within the organization. List can be filtered using various parameters like age, department, job title etc
  - User should be able to create new Employee with some default parameters or list of actual parameters
  - User should be able to update/edit Employee data, only few specific data should be edited/updated
  - User should be able to delete Employee data
  - Employee object should contain:
    - FirstName(string), LastName(string), Age(Integer), DateOfJoining(Date), Title(string), Department(string), EmployeeType(string), CurrentStatus(bool) (ie working=1, retired=0)

### Assignment -1

- Assignment – 1, covers some of the concepts you learnt from chapter 1 – 6
- In this assignment, you start creating 4 Components:
- EmployeeDirectory => Parent Component, and it includes all other components,
- EmployeeSearch => This component, helps user to search Employee data based on certain conditions and parameters
- EmployeeTable => This component, will list Employee data based on the conditions declared in EmployeeSearch component
- EmployeeCreate => This component will help to create new Employee record within EMS
- For your convenience you may want to add any other Component to further breakdown the work
- **Please use appropriate, techniques to pass data across Child components and also between Parent to Child components. Also avoid creating any Hard Coded components, make sure you iterate over list of Employees data and dynamically display data, create components etc (Ref: Dynamic Composition: page 55-57)**
- More about each component:
- **EmployeeSearch:**
- **EmployeeTable:** (Sample of EmployeeTable)

FirstName	LastName	Age	DateOfJoining	Title	Department	EmployeeType	CurrentStatus
Demo	Demo	20	Jan 1 2000	Manager	IT	FullTime	1

- **EmployeeCreate:** You allow following items to be entered from the user (Please add appropriate validation so that user will not add any unwanted values). You may keep some assumption ie Employee age can be between 20-70 etc, and offer to add age data between 20-70 only
- FirstName, LastName, Age, DateOfJoining, Title, Department, EmployeeType
- Title: Employee, Manager, Director, VP (only these values are allowed)
- Department: IT, Marketing, HR, Engineering (Only these values are allowed)
- EmployeeType: FullTime, PartTime, Contract, Seasonal (Only these values are allowed)
- You don't want to offer to enter CurrentStatus, you can assume that when Employee is created the default value for CurrentStatus would be 1 (working)
- Create appropriate UserModel and setup DataBase within MongoDB
- You need to install few tools ie graphql, apollo-server-express to develop GraphQL api
- Use GraphQL Schema ie Query and Mutation and develop UserSchema (**Ref: page 92-96**)
- You may use GraphQL playground to explore GraphQL API, tool is available at <http://localhost:3000/graphql>
- Create separate Schema file for User data
- Create Employee List API: Fetch all the employee data ie similar to SQL => Select \* from Tbl
- Create Employee Create API: To create Employee data ie similar to SQL => Insert into Tbl
- Integrate APIs with Components, perform appropriate Input validation and display any errors
- **Summary: In this assignment, you will create multiple React components, you will use GraphQL to write APIs to fetch Employee data, Create Employee data from MongoDB, you will also do simple validation and display error message. No search/update/delete functionality yet, it would be in next assignment.**