**Key Summary:**

* **DotNet Core 6 WebAPI and MVC with CQRS pattern using MediatR**
* **Use of Repository pattern in MediatR and Service**
* **single project architecture in WebAPI project**
* **CRUD with Razor pages and MVC and WebAPI**
* **Use of class libray projects for separate 4 layers :**

1. **core, 2) data, 3) service , 4) application**

* **Migration and Code first approach using Entity Framework Core**
* **Use of MVC Datagrid(SyncFusiton CDN approach) with api call /users endpoint using ajax**
* **UI VideoLink and Source Code Github link**

**Video Link for Demo UI Play on web :**

<https://screenrec.com/share/zM2QiOU4yB> Time: 6 min 27 second

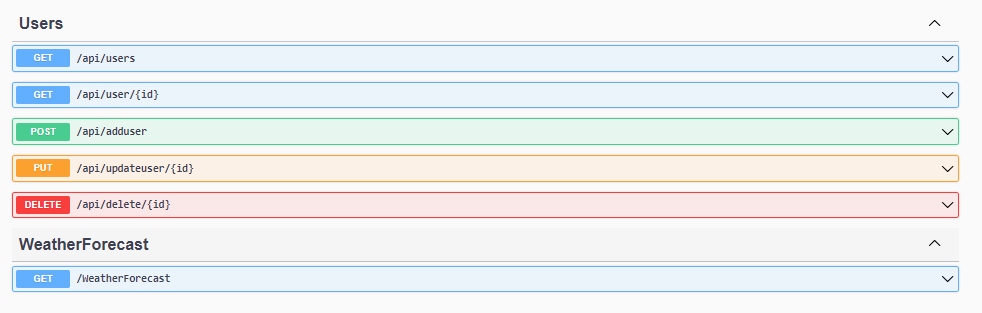
1. Datagrid ( CQRS Pattern MediatR Pattern with WebApi Project )
2. CRUD MVC DotNet Core – CQRS MediatR Pattern

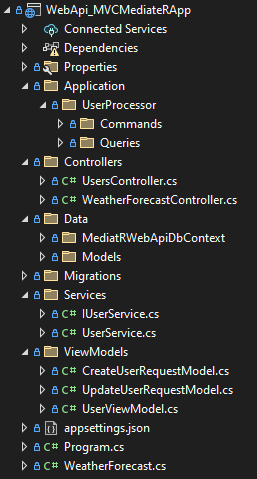
**GitHub Source Code :**

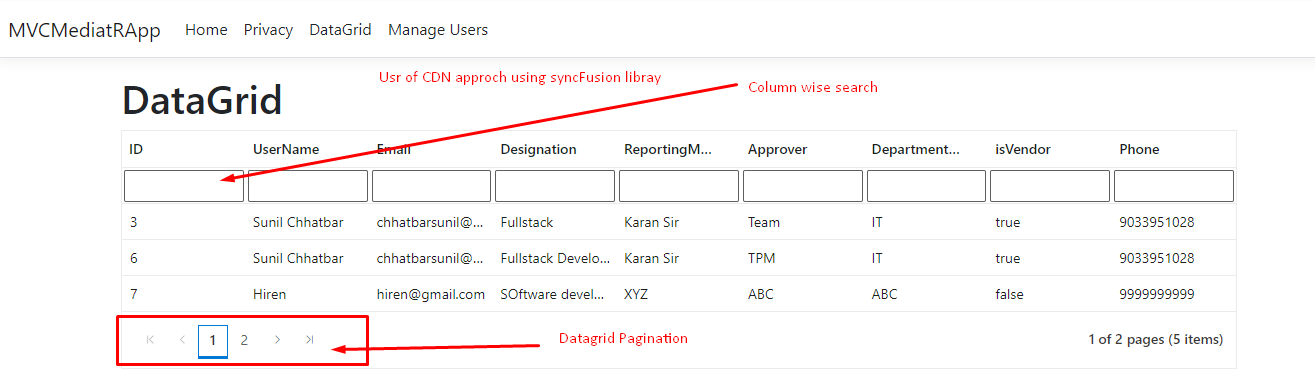
<https://github.com/chhatbarsunil/DotNetRepo/tree/master/MVCMediatRApp>

1. **WebApi Project: WebApi\_MVCMediateRApp**

**( Architecture CQRS pattern with single project )**

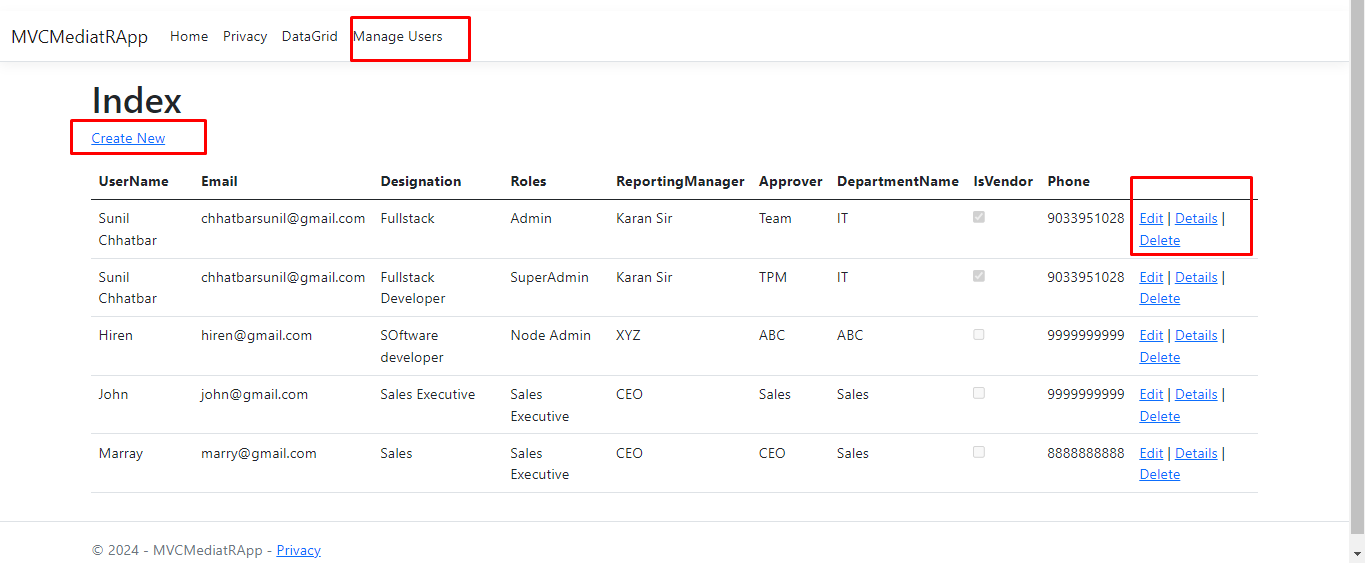


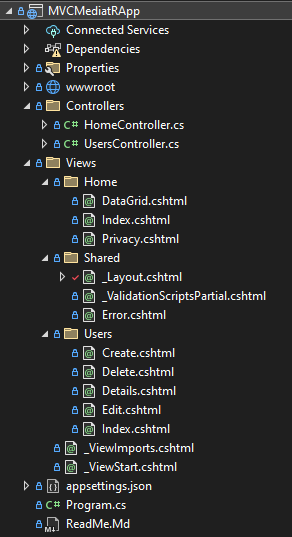


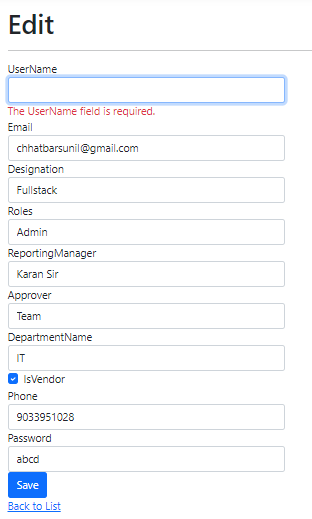
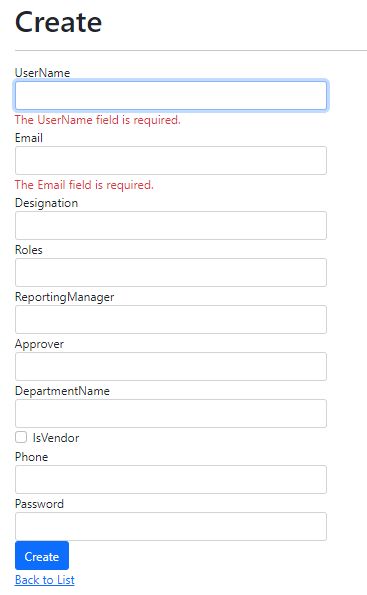


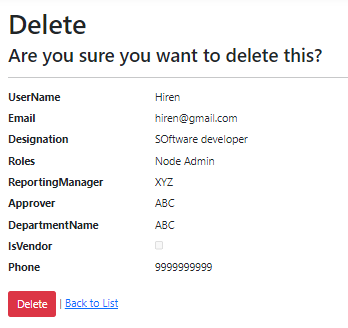
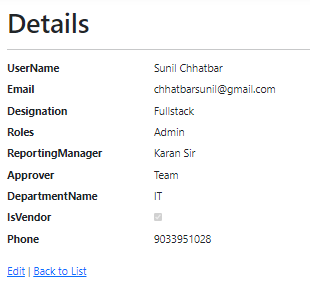
**Projects: MVCMediatRApp Repository ( MVC Project and WebApi )**

1. **MVCMediatRApp ( MVC project with CQRS Pattern )**



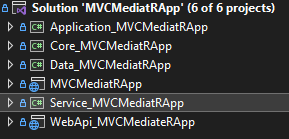






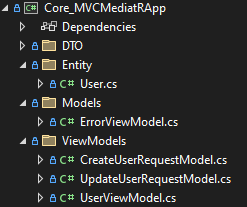
1. **Application Class Library Project**

**( commands and queries ):**



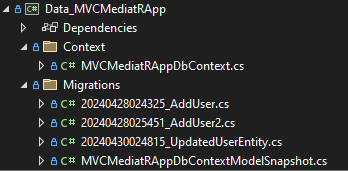
**Objective** : To separate the model read and write operation with MediatR pattern to improve performance and for separation of concern.

1. **Core Layer Class Library Project ( Entity Model, Request Response model and ViewModels )**

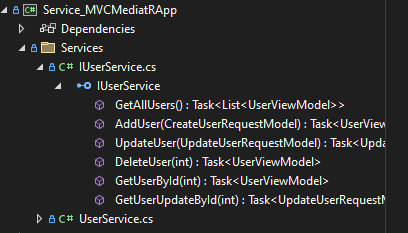


1. **Data Layer Class Library Project**

**(DbContext and Migrations – Code first approach )**



1. **Service Layer Class Library Project - Service\_MVCMediatRApp**



**Thank you**