

```

@page "/CMS1500/Forms"
@using Medical.DTO.Cms1500
@using Medical.BLL
@inject MedicalService MedicalService

<div class="container">
    <h1 class="my-4 text-center">CMS-1500 Forms</h1>
</div>

<DisplayError ErrorList="@MedicalService.Response.ErrorList" />

@if (isLoading)
{
    <p><em>Loading...</em></p>
}
else
{
    <div class="row mb-3">
        <div class="col-md-3">
            <div class="mb-3 form-floating">
                <input type="number" class="form-control" @bind="searchParams.FormId" />
                <label>Search Form ID</label>
            </div>
        </div>
        <div class="col-md-3">
            <div class="mb-3 form-floating">
                <input type="date" class="form-control" @bind="searchParams.CreatedDate" />
                <label>Search Created Date</label>
            </div>
        </div>
        <div class="col-md-3">
            <div class="mb-3 form-floating">
                <input type="date" class="form-control" @bind="searchParams.ServiceDateFrom" />
                <label>Service Date From</label>
            </div>
        </div>
        <div class="col-md-3">
            <div class="mb-3 form-floating">
                <input type="date" class="form-control" @bind="searchParams.ServiceDateTo" />
                <label>Service Date To</label>
            </div>
        </div>
    </div>
}

```

```

<div class="mb-3">
    <button class="btn btn-primary" @onclick="ShowAddFormModal">Add
CMS-1500</button>
    <button class="btn btn-secondary" @onclick="SearchForms">Search</button>
    <button class="btn btn-danger" @onclick="ResetSearch">Reset</button>
</div>

<table class="table table-striped table-hover">
    <thead>
        <tr>
            <th>Form ID</th>
            <th>Patient Name</th>
            <th>Created Date</th>
            <th>Total Charge</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var form in forms ?? [])
        {
            <tr>
                <td>@form.FormId</td>
                <td>@form.Patient?.FirstName @form.Patient?.LastName</td>
                <td>@form.CreatedDate.ToString("yyyy-MM-dd")</td>
                <td>@form.TotalCharge.ToString("C")</td>
                <td>
                    <button class="btn btn-secondary" @onclick="() =>
ShowEditModal(form)">Edit</button>
                    <button class="btn btn-danger" @onclick="() =>
DeleteForm(form)">Delete</button>
                </td>
            </tr>
        }
    </tbody>
</table>

<Pagination TItem="Cms1500FormDTO" TotalItems="@totalItems" PageSize="@pageSize"
CurrentPage="@currentPage"
OnPageChanged="OnPageChanged"
OnPageSizeChangedCallback="OnPageSizeChanged" />
}

<EditModalCMS1500 @ref="editModal" OnFormSaved="LoadForms" />

```

```

@code {
    private List<Cms1500FormDTO>? forms;
    private bool isLoading = true;
    private EditModalCMS1500? editModal;
    private int totalItems;
    private int pageSize = 10;
    private int currentPage = 0;
    private Cms1500SearchDTO searchParams = new Cms1500SearchDTO();

    protected override async Task OnInitializedAsync()
    {
        await LoadForms();
    }

    private async Task LoadForms()
    {
        isLoading = true;
        searchParams.PageNumber = currentPage;
        searchParams.PageSize = pageSize;

        await MedicalService.GetListAsync(searchParams);
        forms = MedicalService.Response.List ?? [];
        totalItems = MedicalService.Response.TotalItems;
        isLoading = false;
    }

    private void ShowAddFormModal()
    {
        editModal?.ShowModal(new Cms1500FormDTO
        {
            Patient = new PatientInfoDTO(),
            Insured = new InsuredInfoDTO(),
            DiagnosisCodes = new List<DiagnosisCodeDTO>(),
            ServiceLines = new List<ServiceLineDTO>()
        });
    }

    private void ShowEditModal(Cms1500FormDTO form)
    {
        editModal?.ShowModal(form);
    }

    private async Task DeleteForm(Cms1500FormDTO form)
    {

```

```
        await MedicalService.DeleteItemAsync(form);
        await LoadForms();
    }

    // Pagination and search methods (similar to original)
    private async Task OnPageChanged(int page) { /*...*/ }
    private async Task OnPageSizeChanged(int size) { /*...*/ }
    private async Task SearchForms() { /*...*/ }
    private async Task ResetSearch() { /*...*/ }
}
```