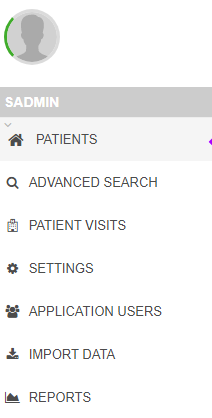
From Views\Shared\\_Layout.cshtml

Line 25: await Html.RenderPartialAsync("\_Aside");

Views\Shared\\_Aside.cshtml

Renders DropDown Side Menu@



The **IMPORT DATA** Option:

@if (User.IsInRole("Admin Role"))

{

<li is-active-route asp-controller="Imports" asp-action="Index">

<a href="@Url.Action("Index", "imports")" title="Imports">

<i class="fa fa-download"></i>

<span class="menu-item-parent">Import data</span>

</a>

</li>

Runs **Index** Action Method on **ImportsController**

public IActionResult Index()

{

return View();

}

Views\Imports\**Index.cshtml**

@Html.Partial("/Views/Imports/\_Index.cshtml")

Views\Imports**\\_Index.cshtml**

@if (User.IsInRole("Admin Role"))

{

<div class="button">

<a href="@Url.Action("New", "Imports")" class="btn btn-success add-user">

<i class="fa fa-plus"></i> Import patients into database

</a>

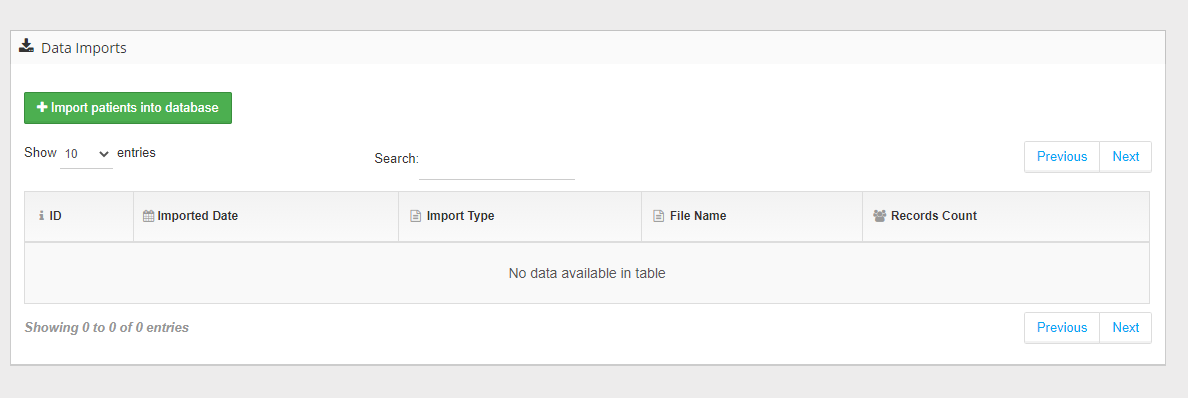
</div>

}

@Html.Partial("/Views/Imports/\_DataTable.cshtml")

</div>

Renders the following DataTable page:



Clicking on the button Runs **New** Action Method on **ImportsController**

public IActionResult New()

{

string batchClinicalLettersDir = \_hostingEnvironment.ContentRootPath + \_configuration.GetSection("clinicLettersDirectory").Value.ToString();

var batchUploadVM = new BatchUploadViewModel();

batchUploadVM.ClinicLettersBatchDirectory = batchClinicalLettersDir;

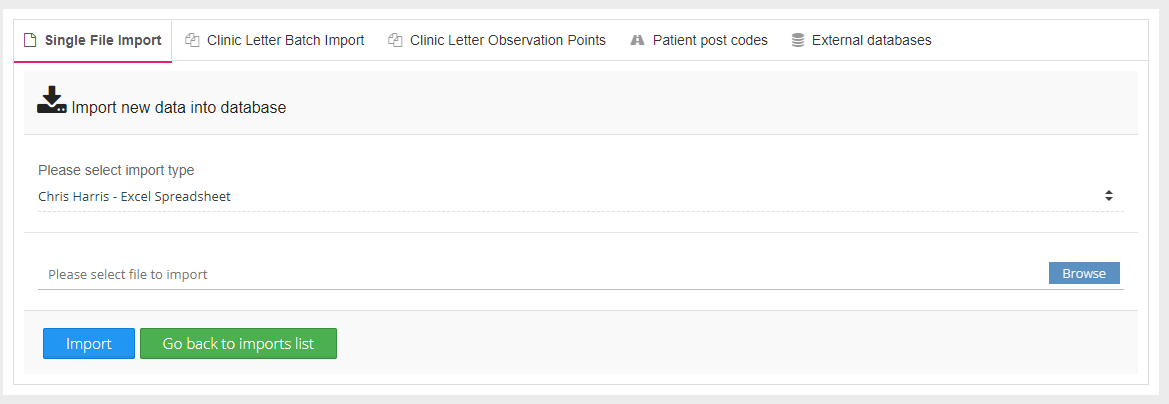
batchUploadVM.ClinicLettersPdfDirectory = \_configuration.GetSection("DataDirectory").Value;

PopulateDbImportTypesDropdownList();

return View(batchUploadVM);

}

Renders a tab-pane Views\Imports\**New.cshtml**



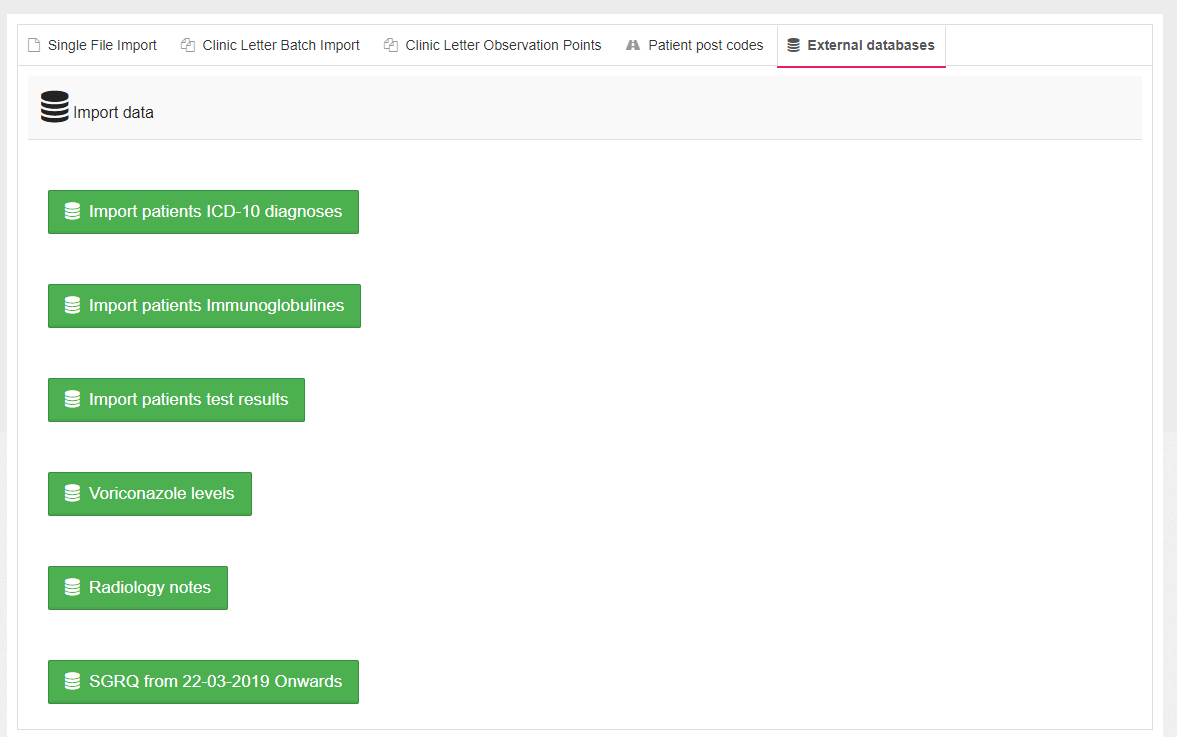
The External Databases Tab Renders:

<div class="tab-pane fade" id="external">

@Html.Partial("/Views/Imports/\_External.cshtml")

</div>

/Views/Imports \_**External.cshtml**



The SGRQ button :

<div class="col-12" style="margin: 0 auto; padding-top: 50px; padding-left: 20px">

<button class="btn btn-lg btn-success" id="external-sgrq">

<i class="fa fa-database"></i>&nbsp; SGRQ from 22-03-2019 Onwards

</button>

</div>

The button event cor this button is set in **Import.js**

var initializeSGRQImport = function () {

$('button#external-sgrq').on('click', function (e) {

e.preventDefault();

$.ajax({

type: "POST",

**url: "/ExternalImports/SGRQ",**

contentType: false,

processData: false,

success: function (response) {

if (response.length === 0) {

LoadingIndicator.hide();

alert('Some error occured while uploading');

} else {

$('#batch-upload-response').html("<div class='alert alert-info'><i class='fa fa-info-circle'></i> &nbsp; Processed (added or updated)" + response.result + " records from orginal file</div>");

LoadingIndicator.hide();

}

},

error: function (e) {

LoadingIndicator.hide();

$('#batch-upload-response').html(e.responseText);

}

});

});

};

This calls the **SQRQ** action method on Controllers\**ExternalImportControllers**

[HttpPost]

public async Task<IActionResult> SGRQ()

{

var sqrqs = \_sqrqContext.QuestionnaireResults

.Include(q => q.Questionnaire)

.ThenInclude(q => q.Patient)

.Where(q => q.Questionnaire.DateOfEntry > SGRQ\_START\_IMPORT\_DATE);

foreach (var externalSGRQ in sqrqs)

{

var districtNumber = externalSGRQ.Questionnaire.Patient.Identifier.ToString();

var patient = \_context.Patients

.Include(p => p.STGQuestionnaires)

.Where(p => p.DistrictNumber.Equals(districtNumber)).FirstOrDefault();

if (patient == null) continue;

var existingDates = patient.STGQuestionnaires.Select(pi => pi.DateTaken.Date).ToList();

bool dateDoesNotExist = existingDates.FindAll(d => d.Date == externalSGRQ.Questionnaire.DateOfEntry.Date).ToList().Count == 0;

if (dateDoesNotExist)

{

var sgrq = new PatientSTGQuestionnaire();

sgrq.PatientId = patient.ID;

sgrq.SymptomScore = (decimal) externalSGRQ.SymptomScore;

sgrq.ImpactScore = (decimal)externalSGRQ.ImpactScore;

sgrq.ActivityScore = (decimal)externalSGRQ.ActivityScore;

sgrq.TotalScore = (decimal)externalSGRQ.TotalScore;

sgrq.DateTaken = externalSGRQ.Questionnaire.DateOfEntry;

sgrq.OriginalImportedId = externalSGRQ.Questionnaire.ID.ToString();

\_context.PatientSTGQuestionnaires.Add(sgrq);

}

}

await \_context.SaveChangesAsync();

return Ok();

}

}