repeatedly create view mvc

<div class="row">

<div class="col-md-8">

<form **asp-action**="Create">

<div **asp-validation-summary**="ModelOnly" class="text-danger"></div>

<div class="form-group">

<label **asp-for**="Title" class="control-label"></label>

<input **asp-for**="Title" class="form-control" />

<span **asp-validation-for**="Title" class="text-danger"></span>

</div>

<div class="form-group">

<label **asp-for**="Description" class="control-label"></label>

<input **asp-for**="Description" class="form-control" />

<span **asp-validation-for**="Description" class="text-danger"></span>

</div>

<div class="form-group">

<label **asp-for**="SubjectId" class="control-label"></label>

<select **asp-for**="SubjectId" **asp-items**="@await service.GetSubjectsToSelectListItem()" class="form-control">

<option selected="selected" **value**="">Please Select</option>

</select>

<span **asp-validation-for**="SubjectId" class="text-danger"></span>

</div>

<div class="form-group">

<label **asp-for**="LevelId" class="control-label"></label>

<select **asp-for**="LevelId" **asp-items**="@await service.GetLevelsToSelectListItem()" class="form-control">

<option selected="selected" **value**="">Please Select</option>

</select>

<span **asp-validation-for**="LevelId" class="text-danger"></span>

</div>

<div id="authorFormGroup">

<label **asp-for**="Authors" class="text-info"></label>

<div class="form-row border border-light pt-1 mb-2">

<div class="form-group col-md-4">

<label **asp-for**="Authors.ElementAt(0).FirstName" class="control-label"></label>

<input **asp-for**="Authors.ElementAt(0).FirstName" class="form-control">

</div>

<div class="form-group col-md-4">

<label **asp-for**="Authors.ElementAt(0).LastName" class="control-label"></label>

<input **asp-for**="Authors.ElementAt(0).LastName" class="form-control">

</div>

<div class="form-group col-md-3">

<label **asp-for**="Authors.ElementAt(0).BirthDay" class="control-label"></label>

<input **asp-for**="Authors.ElementAt(0).BirthDay" class="form-control">

</div>

<div class="form-group">

<button type="button" class="btn" id="addRowBtn"><span class="text-success"><i class="fa fa-plus-circle"></i></span></button>

</div>

</div>

</div>

<div class="form-group">

<input type="submit" value="Create" class="btn btn-primary" />

</div>

</form>

</div>

</div>

<div>

<a **asp-action**="Index">Back to List</a>

</div>

public IActionResult Create()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create(LiteratureCreateViewModel viewModel)

{

var client = clientFactory.CreateClient("LiteratureClient");

var requestMsg = new HttpRequestMessage(HttpMethod.Post, "literatures");

requestMsg.Content = JsonContent.Create(viewModel);

var responseMsg = await client.SendAsync(requestMsg);

if (responseMsg.IsSuccessStatusCode)

{

return RedirectToAction(nameof(Index));

}

else

{

ModelState.AddModelError(string.Empty, $"Server error at {nameof(Create)}. Status code: {responseMsg.StatusCode}.");

}

return View(viewModel);

}

$(document).ready(function () {

let container = $('#authorFormGroup');

let count = 0;

console.log('Before Literature Create method: ' + count);

function AddAuthorRowOnForm(e) {

e.preventDefault();

let maxNoOfRows = 9;

if (count < maxNoOfRows) {

$(container).append('<div class="form-row border border-light pt-1 mb-2">' +

'<div class= "form-group col-md-4">' +

'<label for="inputEmail4">First name</label>' +

'<input type="text" class="form-control" id="inputEmail4">' +

'</div>' +

'<div class= "form-group col-md-4" > ' +

'<label for="inputPassword4">Last name</label>' +

'<input type="text" class="form-control" id="inputPassword4">' +

'</div>' +

'<div class="form-group col-md-3">' +

'<label for="inputPassword4">Birthday</label >' +

'<input type="date" class="form-control" id="inputPassword4">' +

'</div>' +

'<div class="form-group">' +

'<button type="button" class="btn" id="removeRowBtn"><span class="text-danger"><i class="fa fa-minus-circle"></i></span></button >' +

'</div>' +

'</div>');

count++;

console.log('Add: ' + count);

}

else {

alert('Limit for adding authors has been reached!');

}

}

$('#addRowBtn').click(AddAuthorRowOnForm);

$(container).on('click', '#removeRowBtn', function (e) {

e.preventDefault();

e.target.closest('.form-row').remove();

count--;

console.log('Remove: ' + count);

});

}); // end doc ready

$(document).on('click', '#BtnCreate', function (e) {

var course = getCourse();

var moduleSets = getModulesets();

$.ajax({type: 'POST',url: "[https://localhost:44314/Courses/Create"](https://localhost:44314/Courses/Create%22),data: {

"coursevm": course,

"modulesetsvm":moduleSets },

success: function (response) {window.location.href = response.redirectToUrl;},

error: function (err) {;}});});