Paisa2u.API

**Changes to VendorsController** (add new one)

[HttpGet("GetAllProductsOfAllVendor")]

public async Task<IEnumerable<ProductByVendor>> GetAllProductsOfAllVendor(CancellationToken cancellationToken)

{

return await \_vendorService.GetAllProductsOfAllVendor(cancellationToken);

}

**CategoriesController**

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using paisa2u.common.Resources;

using paisa2u.common.Services;

using System.Text.Json;

namespace paisa2u.API.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class CategoriesController : ControllerBase

{

private readonly ICategoryService \_categoryService;

public CategoriesController(ICategoryService categoryService)

{

\_categoryService = categoryService;

}

// GET: api/Categories

[HttpGet("GetCategories")]

public async Task<ActionResult<IEnumerable<CategoryResource>>> GetCategories(CancellationToken cancellationToken)

{

var a = await \_categoryService.GetCategories(cancellationToken);

var jsonString = JsonSerializer.Serialize(a);

return Ok(jsonString);

}

// GET: api/Categories/5

[HttpGet("GetCategory")]

public async Task<ActionResult<CategoryResource>> GetCategory(int id, CancellationToken cancellationToken)

{

try

{

var response = await \_categoryService.GetCategory(id, cancellationToken);

return Ok(response);

}

catch (Exception e)

{

return BadRequest(new { ErrorMessage = e.Message });

}

}

// PUT: api/Category/5

[HttpPut]

public async Task<IActionResult> UpdateCategory([FromBody] CategoryResource resource, CancellationToken cancellationToken)

{

try

{

var response = await \_categoryService.UpdateCategory(resource.catid, resource, cancellationToken);

return Ok(response);

}

catch (Exception e)

{

return BadRequest(new { ErrorMessage = e.Message });

}

}

// DELETE: api/Categories/5

[HttpDelete]

public async Task<IActionResult> DeleteCategory([FromBody] CategoryResource resource, CancellationToken cancellationToken)

{

try

{

var response = await \_categoryService.DeleteCategory(resource, cancellationToken);

return Ok(response);

}

catch (Exception e)

{

return BadRequest(new { ErrorMessage = e.Message });

}

}

}

}

**Program.cs**

builder.Services.AddScoped<ICategoryService, CategoryService>(); //register Icategoryservice

var app = builder.Build();

//new code here

app.Use((ctx, next) =>

{

ctx.Response.Headers["Access-Control-Allow-Origin"] = "https://localhost:7257";

ctx.Response.Headers["Access-Control-Allow-Headers"] = "Origin, X-Requested-With, Content-Type, Accept, Authorization";

return next();

}

);

app.UseAuthorization();

//new code here if not there already

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers()

.RequireCors("CorsApi");

});

Index.html, home.controller on ui->for getsearch

Grid, changes to ui-usercontroller to get regid in tmpdata

Changes to refe

Paisa2u.Common

Services folder

**ICategoryService.cs**

using paisa2u.common.Resources;

namespace paisa2u.common.Services

{

public interface ICategoryService

{

Task<IEnumerable<CategoryResource>> GetCategories(CancellationToken cancellationToken);

Task<CategoryResource> GetCategory(int Catid, CancellationToken cancellationToken);

Task<CategoryResource> DeleteCategory(CategoryResource resource, CancellationToken cancellationToken);

Task<CategoryResource> UpdateCategory(int catid, CategoryResource resource, CancellationToken cancellationToken);

}

}

Services folder

**CategoryService.cs**

using Microsoft.EntityFrameworkCore;

using paisa2u.common.Models;

using paisa2u.common.Resources;

namespace paisa2u.common.Services

{

public class CategoryService : ICategoryService

{

public readonly PaisaDbContext \_context;

public CategoryService(PaisaDbContext context)

{

\_context = context;

}

public async Task<IEnumerable<CategoryResource>> GetCategories(CancellationToken cancellationToken)

{

List<CategoryResource> categoryList = new List<CategoryResource>();

var a = await \_context.Categories.ToListAsync(cancellationToken);

foreach(Category category in a) {

categoryList.Add

(

new CategoryResource(

category.Catid,

category.Catdesc

)

);

}

return categoryList;

}

public async Task<CategoryResource> GetCategory(int catid, CancellationToken cancellationToken)

{

var category = await \_context.Categories

.FirstOrDefaultAsync(x => x.Catid == catid, cancellationToken);

if (category == null)

throw new Exception("category does not exist!");

return new CategoryResource(

category.Catid,

category.Catdesc

);

}

public async Task<CategoryResource> DeleteCategory(CategoryResource resource, CancellationToken cancellationToken)

{

var category = await \_context.Categories

.FirstOrDefaultAsync(x => x.Catid == resource.catid, cancellationToken);

//check for cascade

if (category.Catid != resource.catid)

{

throw new Exception("Category to remove does not exist");

}

var cat = new CategoryResource(

category.Catid,

category.Catdesc

);

\_context.Entry(category).State = EntityState.Deleted;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

throw new Exception("User can not be deleted!");

}

return cat;

}

public async Task<CategoryResource> UpdateCategory(int catid, CategoryResource resource, CancellationToken cancellationToken)

{

var edCategory = await \_context.Categories

.FirstOrDefaultAsync(x => x.Catid == catid, cancellationToken);

if (edCategory.Catid != catid)

{

throw new Exception("Category to update does not exist");

}

edCategory.Catid = resource.catid;

edCategory.Catdesc = resource.catdesc;

\_context.Entry(edCategory).State = EntityState.Modified;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

throw;

}

return new CategoryResource(

edCategory.Catid,

edCategory.Catdesc

);

}

}

}

Resources (folder)

namespace paisa2u.common.Resources

{

public sealed record CategoryResource

(

int catid,

string catdesc

);

}

**IvendorServices (add)**

Task<IEnumerable<ProductByVendor>> GetAllProductsOfAllVendor(CancellationToken cancellationToken);

Paisa2u.UI

Home->Home Controller (Need help here to provide IDs to vendor, category and product

using Microsoft.AspNetCore.Mvc;

using Newtonsoft.Json;

using paisa2u.common.Models;

using paisa2u.common.Resources;

using paisa2u.UI.Models;

using System.Diagnostics;

using System.Text;

namespace paisa2u.UI.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public async Task<ActionResult<ProductByVendor>> Index()

{

List<ProductByVendor> productByVendor = new List<ProductByVendor>();

var client = new HttpClient();

client.DefaultRequestHeaders.Clear();

using (var response = await client.GetAsync("https://localhost:7172/api/Vendors/GetSingleProductByVendor"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

var a = JsonConvert.DeserializeObject<ProductByVendor[]>(apiResponse);

return View("../Home/Index", a);

}

}

public async Task<ActionResult> GetSearch(string Myselect , string Option , IFormCollection collection)

{

var select = collection["Myselect"];

var option = collection["Option"];

//products of all vendors

if (select == "All" && option == "Vendor" )

{

List<ProductByVendor> productByVendors = new List<ProductByVendor>();

var client1 = new HttpClient();

client1.DefaultRequestHeaders.Clear();

using (var response = await client1.GetAsync("https://localhost:7172/api/Vendors/GetAllProductsOfAllVendor"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

var a = JsonConvert.DeserializeObject<ProductByVendor[]>(apiResponse);

return View("../Shared/Vendors", a);

}

}

//products of a vendor

else if (select != "All" && option == "Vendor")

{

List<ProductByVendor> productByVendors = new List<ProductByVendor>();

StringContent content = new StringContent(JsonConvert.SerializeObject(productByVendors), Encoding.UTF8, "application/json");

var client = new HttpClient();

client.DefaultRequestHeaders.Clear();

using (var response = await client.GetAsync("https://localhost:7172/api/Vendors/GetAllProductsByVendor"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

var a = JsonConvert.DeserializeObject<VendorResource[]>(apiResponse);

return View("../Shared/Vendors", a);

}

}

//code to be inserted here

//else if (select == "All" && option == "Product")

//{

//}

//else if (select != "All" && option == "Product")

//{

//}

//else if (select == "All" && option == "Category")

//{

//}

//code to change here to find single category

else

{

List<ProductByVendor> productByVendors = new List<ProductByVendor>();

StringContent content = new StringContent(JsonConvert.SerializeObject(productByVendors), Encoding.UTF8, "application/json");

var client = new HttpClient();

client.DefaultRequestHeaders.Clear();

using (var response = await client.GetAsync("https://localhost:7172/api/Vendors/GetAllProductsByVendor"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

var a = JsonConvert.DeserializeObject<VendorResource[]>(apiResponse);

return View("../Shared/Vendors", a);

}

}

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

**Views->Shared->Vendor**

@{

ViewData["Title"] = "Search Page";

}

@using paisa2u.common.Models

@model IEnumerable<ProductByVendor>

<section class="py-5" id="Opportuanities">

@\*<div class="bg-holder d-none d-sm-block" style="background-image:url(assets/img/illustrations/bg.png);background-position:top left;background-size:225px 755px;margin-top:-17.5rem;">

</div>\*@

<!--/.bg-holder-->

<div class="container" style="top: 50px;">

<div class="row">

<div id="welcome" @\*style="display:none"\*@>

@if (@ViewBag.message != null)

{

<label id="welcomemessage"> @ViewBag.message </label>

}

</div>

</div>

<div class="row align-items-center min-vh-75 min-vh-lg-100">

<div class="col-md-7 col-lg-6 col-xxl-5 py-6 text-sm-start text-center">

<h1 class="mt-6 mb-sm-4 fw-semi-bold lh-sm fs-4 fs-lg-5 fs-xl-6">A New Way to Invest <br class="d-block d-lg-block" />in Agriculture</h1>

<p class="mb-4 fs-1">Zou provides farmes, ranchers, private foresters, and agricultural producers with online self service applications and educational materials.</p><a class="btn btn-lg btn-success" href="#" role="button">Invest Now</a>

</div>

</div>

</div>

<div class="container">

<div class="row flex-center h-100">

<div class="col-xl-9">

<div class="row">

<!--Loop -->

@foreach (var i in Model)

{

<div class="col-md-4 mb-5">

<div class="card h-100 shadow px-4 px-md-2 px-lg-3 card-span pt-6">

<div class="text-center text-md-start card-hover">

<img class="ps-3 icons" src=" assets/img/icons/farmer.svg" height="60" alt="" />

<div class="card-body">

<h6 class="fw-bold fs-1 heading-color">@i.Firstname</h6>

<p class="mt-3 mb-md-0 mb-lg-2">@i.Catdesc</p>

<p class="mt-3 mb-md-0 mb-lg-2">@i.Productname</p>

</div>

</div>

</div>

</div>

}

<!--Loop -->

@\*<div class="col-md-4 mb-5">

<div class="card h-100 shadow px-4 px-md-2 px-lg-3 card-span pt-6">

<div class="text-center text-md-start card-hover">

<img class="ps-3 icons" src="assets/img/icons/growth.svg" height="60" alt="" />

<div class="card-body">

<h6 class="fw-bold fs-1 heading-color">Grow your business</h6>

<p class="mt-3 mb-md-0 mb-lg-2">Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam.</p>

</div>

</div>

</div>

</div>

<div class="col-md-4 mb-5">

<div class="card h-100 shadow px-4 px-md-2 px-lg-3 card-span pt-6">

<div class="text-center text-md-start card-hover">

<img class="ps-3 icons" src="assets/img/icons/planting.svg" height="60" alt="" />

<div class="card-body">

<h6 class="fw-bold fs-1 heading-color">Social Impact Invesment</h6>

<p class="mt-3 mb-md-0 mb-lg-2">At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos</p>

</div>

</div>

</div>

</div>\*@

</div>

</div>

</div>

</div>

</section>

@section Scripts {

<script type="text/javascript">

$('#Find').onclick(function () {

var Option = $('#options').val;

var MySelect = $('#list1').val;

$.ajax({

type: "GET",

url: "https://localhost:7257/Home/GetSearch",

contentType: "application/json; charset=utf-8",

data: { Option, MySelect }

dataType: "json",

success: function (data) {

alert(JSON.stringify(data));

$.each(data, function (index, response) {

//alert(response.Firstname);

MySelect.append("<option value='" + response.productid + "'>" + response.productname + "</option>");

}); //each

},

failure: function (data) {

alert('failure in getting Products');

alert(data.responseText);

},

error: function (xhr, ajaxOptions, thrownError) {

alert(xhr.status);

alert(thrownError);

} //error

});//ajax closure

}

}

</script>

}