**Add ef core relationship in models**

 public ICollection<BlogPost> BlogPosts {get; set;}

public ICollection<Category> Categories { get; set; }

**Change add blog post action method- accept categories**

using codepulseapi.API.Models.Domain;

using codepulseapi.API.Models.DTO;

using codepulseapi.API.Repositories.Interface;

using codepulseapi.API.Repositories.Repository;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.IdentityModel.Tokens;

namespace codepulseapi.API.Controllers

{

    [Route("api/[controller]")]

    [ApiController]

    public class BlogPostsController : ControllerBase

    {

        private readonly IBlogPostRepository blogPostRepository;

        public BlogPostsController(IBlogPostRepository blogPostRepository)

        {

            this.blogPostRepository = blogPostRepository;

        }

[HttpPost]

        public async Task<IActionResult> CreateBlogPost([FromBody]CreateBlogPostRequestDto request)

       {

           var blogPost = new BlogPost

           {

            Title = request.Title,

            ShortDescription = request.ShortDescription,

            Content = request.Content,

            FeaturedImageUrl = request.FeaturedImageUrl,

            UrlHandle = request.UrlHandle,

            PublishedDate = request.PublishedDate,

            Author = request.Author,

            IsVisible = request.IsVisible,

            Categories = new List<Category>()

            };

            foreach (var categoryInt in request.Categories)

            {

                var existingCategory = await categoryRepository.GetById(categoryInt);

                if(existingCategory is not null)

                {

                    blogPost.Categories.Add(existingCategory);

                }

            }

            blogPost = await blogPostRepository.CreateAsync(blogPost);

            var response = new BlogPostDto

            {

                Id = blogPost.Id,

                Title = blogPost.Title,

                ShortDescription = blogPost.ShortDescription,

                Content = blogPost.Content,

                FeaturedImageUrl = blogPost.FeaturedImageUrl,

                UrlHandle = blogPost.UrlHandle,

                PublishedDate = blogPost.PublishedDate,

                Author = blogPost.Author,

                IsVisible = blogPost.IsVisible,

                Categories = blogPost.Categories.Select(x=> new CategoryDto

                {

                    Id = x.Id,

                    Name = x.Name,

                    UrlHandle = x.UrlHandle

                }).ToList()

            };

            return Ok();

        }

**Change get all post action method – return categories**

 [HttpGet]

        public async Task<IActionResult> GetAllBlogPosts()

        {

           var blogPosts = await blogPostRepository.GetAllAsync();

           var response = new List<BlogPostDto>();

           foreach (var blogPost in blogPosts)

           {

            response.Add(new BlogPostDto

            {

                Id = blogPost.Id,

                Title = blogPost.Title,

                ShortDescription = blogPost.ShortDescription,

                Content = blogPost.Content,

                FeaturedImageUrl = blogPost.FeaturedImageUrl,

                UrlHandle = blogPost.UrlHandle,

                PublishedDate = blogPost.PublishedDate,

                Author = blogPost.Author,

                IsVisible = blogPost.IsVisible,Categories = blogPost.Categories.Select(x=> new CategoryDto

                {

                    Id = x.Id,

                    Name = x.Name,

                    UrlHandle = x.UrlHandle

                }).ToList()

            });

           }

           return Ok(response);

        }

**Get all post service -UI**

getAllBlogPosts() : Observable<BlogPost[]>{

    return this.http.get<BlogPost[]>('http://localhost:5125/api/blogposts');

  }

**Display categories in blogpost list**

**BlogPostList.component.html**

<div class="container">

    <h1 class="mt-3">BlogPost List </h1>

    <div class="d-flex justify-content-end mt-3">

        <a [routerLink]="['/admin/blogposts/add']" class="btn btn-primary"> Add BlogPost</a>

    </div>

    <ng-container \*ngIf="blogPosts$ | async as blogPosts">

        <table class="table table-bordered mt-3">

            <thead>

                <tr>

                    <th> Title </th>

                    <th> Short Desciption </th>

                    <th> Is Visible </th>

                    <th> Categories </th>

                    <th>  </th>

                </tr>

            </thead>

            <tbody>

                <tr \*ngFor="let blogPost of blogPosts">

                 <td>{{blogPost.title}} </td>

                 <td>{{blogPost.shortDescription}} </td>

                 <td>{{blogPost.isVisible}} </td>

                 <td >

                    <span \*ngFor="let category of blogPost.categories">

                    <span class ="badge bg-secondary"> {{category.name}} </span>

                    </span>

                 </td>

                 <td>

                    <a class="btn btn-light" [routerLink]="['/admin/blogposts', blogPost.id]"> Edit </a>

                 </td>

                </tr>

            </tbody>

        </table>

    </ng-container>

</div>

**Display categories multiselect dropdown**

**Ass blogpost.component.html**

 <div class="mt-3">

                <label for="categories" class="form-label"> Categories</label>

                <ng-container \*ngIf="categories$ | async as categories">

                  <select class="form-control" [(ngModel)]="model.categories"

                  name="categories"

                  id="categories" multiple>

                  <option \*ngFor="let category of categories" [value]="category.id"> {{category.name}} </option>

                </select>

                </ng-container>

              </div>