# Project

* Site-20-Api-Ng-PhotoManagement
  + Copied from: Site-19-Api-Ng-UnitOfWork-OptQueries-CofirmService-MemMessage
* For the “MySocialConnect-SPA” issue npm install command to install all packages
  + ng serve : to run the spa

# New & Updates

|  |  |
| --- | --- |
| New | Updates |
| /core/models-interfaces/photo-for-approval-dto.model.ts | /core/models-intrfaces/photo-dto.model.ts |
|  | /core/services/helper.service.ts |
|  | /core/services/admin.service.ts |
|  | /site/admin/photo-management.componenet |
|  | /site/members/photo-editor.component |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Clear Database

If you want to clear database then drop it and recreate it

dotnet ef database drop

dotnet ef database update

# /core/models-interfaces

## photo-dto.model.ts

export interface PhotoDto {

    id: number;

    url: string;

    isMain: boolean;

    //Added IsApproved with with PhotoManagement and removed PublicId since this is not being used on the client.

    //publicId: string;

    isApproved: boolean;

}

## photo-for-approval.model.ts

> ng g interface core/models-interfaces/photoForApprovalDto --type=model

export interface PhotoForApprovalDto {

    id: number;

    url: string;

    userName: string;

    userId: number;

    userGuid: string;

    isApproved: boolean;

}

# /core/services

## helper.service.ts

Add the urs for approve and reject urls under admin section urls

  public urlAdminApprovePhoto: string = `${this.urlAdmin}/photo-to-approve/${this.keyPhotoId}`;

  public urlAdminRejectPhoto: string = `${this.urlAdmin}/photo-to-reject/${this.keyPhotoId}`;

## admin.service.ts

Add the three methods for getting photos for approval, accept and reject

  getPhotosForApproval(){

    const url = this.helperService.urlAdminGetPhotosToModerate;

    this.helperService.logIfFrom(url, "AdminService getPhotosForApproval");

    return this.httpClientService.get<PhotoForApprovalDto[]>(url);

  }

  approvePhoto(photoId: number){

    const url = this.helperService.replaceKeyValue(this.helperService.urlAdminApprovePhoto, this.helperService.keyPhotoId, photoId);

    this.helperService.logIfFrom(url, "AdminService approvePhoto");

    return this.httpClientService.put<any>(url, {});

  }

  rejectPhoto(photoId: number){

    const url = this.helperService.replaceKeyValue(this.helperService.urlAdminRejectPhoto, this.helperService.keyPhotoId, photoId);

    this.helperService.logIfFrom(url, "AdminService rejectPhoto");

    return this.httpClientService.put<any>(url, {});

  }

# /site/admin/photo-management.componenet

## photo.management.component.ts

import { Component, OnDestroy, OnInit } from '@angular/core';

import { AdminService } from '../../../core/services/admin.service';

import { PhotoForApprovalDto } from '../../../core/models-interfaces/photo-for-approval-dto.model';

import { Subscription } from 'rxjs';

@Component({

  selector: 'app-photo-management',

  templateUrl: './photo-management.component.html',

  styleUrls: ['./photo-management.component.css']

})

export class PhotoManagementComponent implements OnInit, OnDestroy {

  photos: PhotoForApprovalDto[] = [];

  getPhotosSubscription!: Subscription;

  constructor(private adminService: AdminService){}

  ngOnDestroy(): void {

    if(this.getPhotosSubscription) this.getPhotosSubscription.unsubscribe();

  }

  ngOnInit(): void {

    this.getPhotosForApproval();

  }

  getPhotosForApproval(){

    this.getPhotosSubscription = this.adminService.getPhotosForApproval().subscribe({

      next: (photos: PhotoForApprovalDto[]) => {

        this.photos = photos;

      },

      error: e => {},

      complete: () => {}

    });

  }

  approvePhoto(photoId: number){

    this.adminService.approvePhoto(photoId).subscribe({

      next: () => {

        this.photos.splice(this.photos.findIndex(p => p.id == photoId), 1);

      }

    });

  }

  rejectPhoto(photoId: number){

    this.adminService.rejectPhoto(photoId).subscribe({

      next: () => {

        this.photos.splice(this.photos.findIndex(p => p.id == photoId), 1);

      }

    });

  }

}

## photo-management.component.css

img.img-thumbnail {

    height: 175px;

    min-width: 175px !important;

    margin-bottom: 2px;

}

## photo-management.component.html

<h4 \*ngIf="!photos || photos.length <= 0">No photos to moderate</h4>

<div class="row" \*ngIf="photos">

    <div class="col-sm-2" \*ngFor="let photo of photos">

      <h4>{{ photo.userName }}</h4>

      <img src='{{photo.url}}' class='img-thumbnail p-1' alt='{{photo.userName}}'>

      <div class='text-center'>

        <button class='btn btn-sm btn-success me-1' (click)='approvePhoto(photo.id)'>Approve</button>

        <button class='btn btn-sm btn-danger' (click)='rejectPhoto(photo.id)'>Reject</button>

      </div>

    </div>

</div>

# /site/members/photo-editor.component

## photo-editor.component.css

.nv-file-over{

    border: dotted 3px red;

}

input[type=file]{

    color: transparent;

}

.img-wrapper-approval{

    position: relative;

}

.not-approved {

    opacity: 0.2;

}

.img-text-approval {

    position: absolute;

    left: 0;

    right: 0;

    margin-left: auto;

    margin-right: auto;

    bottom: 30%;

}

## photo-editor.component.html

<div class="row mb-2" \*ngIf="member">

    <!--

        In the photo-editor when a user uploads photos ensure that some text is positioned

        absolutely on any currently unapproved photos as “awaiting approval” in red. Also

        reduce the opacity to 0.2 for any unapproved photos.

        with photo approval add classes

        div = img-wrapper-approval

        img = not-approved

        div awaiting approval

    -->

    <div class="col-2 mb-1 img-wrapper-approval" \*ngFor="let photo of member?.photos">

        <img src="{{photo.url}}" alt="Photo of user" class="img-thumbnail mb-1"

            [class.not-approved]="!photo.isApproved">

        <div class="text-center img-text-approval" \*ngIf="!photo.isApproved">

            <span class="text-danger">Awaiting approval</span>

        </div>

        <div class="text-center">

            <!--do not let the user make it main when not approved-->

            <button class="btn btn-sm"

                [disabled]="photo.isMain || !photo.isApproved"

                (click)="onSetMainPhoto(photo)"

                [ngClass]="photo.isMain ? 'btn-success active' : 'btn-outline-success'">

                Main</button>

            <button class="btn-sm btn-danger ms-2" \*ngIf="!photo.isMain"

            (click)="onDeletePhoto(photo)"><i class="fa fa-trash"></i></button>

        </div>

    </div>

</div>