.NET Core Models for Little Sprouts Nursery Website (Updated with User Roles)

Now, let's reflect these changes in your C# models, using EF Core conventions for relationships.

---

1. Core Nursery Information Model

C#
```
namespace LittleSprouts.Web.Models
{
    public class NurseryInfo
    {
        [Key]
        public int Id { get; set; }
        public string Name { get; set; } = "Little Sprouts Nursery";
        public string Address { get; set; }
        public string PhoneNumber { get; set; }
        public string Email { get; set; }
        public string OpeningHours { get; set; }
        public string EthosStatement { get; set; }
        public string MissionStatement { get; set; }
        public string WhyChooseUsText { get; set; }
        public string OfstedRating { get; set; }
        public string OfstedReportLink { get; set; }
    }
}
```

---

2. People & Team Models

C#
```
namespace LittleSprouts.Web.Models
{
    public class TeamMember
    {
        [Key]
        public int Id { get; set; }
        public string Name { get; set; }
        public string Role { get; set; } // This is the nursery staff role
(e.g., Manager, Room Leader)
        public string Bio { get; set; }
        public string PhotoUrl { get; set; }
        public int OrderDisplay { get; set; }
    }
}
```

---

3. Content Management Models

C#

```
using System.ComponentModel.DataAnnotations;
using System.Collections.Generic; // For navigation properties

namespace LittleSprouts.Web.Models
{
    public class Page
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string Slug { get; set; }
        public string Content { get; set; }
        public DateTime LastUpdated { get; set; }

        // Foreign Key to the User who last updated the page
        public int? LastUpdatedByUserId { get; set; }
        public ApplicationUser? LastUpdatedByUser { get; set; } // Navigation
property
    }

    public class CurriculumItem
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string Content { get; set; }
        public string AgeGroup { get; set; }
        public bool IsSpecialProgram { get; set; }
    }

    public class Policy
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string Description { get; set; }
        public string FileUrl { get; set; }
        public string Category { get; set; }
        public DateTime LastUpdated { get; set; }
    }

    public class FAQ
    {
        [Key]
        public int Id { get; set; }
        public string Question { get; set; }
        public string Answer { get; set; }
        public string Category { get; set; }
        public int OrderDisplay { get; set; }
    }

    public class Vacancy
    {
        [Key]
        public int Id { get; set; }
```

```
        public string JobTitle { get; set; }
        public string Description { get; set; }
        public string Requirements { get; set; }
        public string ApplicationProcess { get; set; }
        public DateTime DatePosted { get; set; }
        public DateTime? ClosingDate { get; set; }
        public bool IsActive { get; set; }
    }
}
```

---

4. Dynamic Content Models

C#

```
using System.ComponentModel.DataAnnotations;
using System.Collections.Generic;

namespace LittleSprouts.Web.Models
{
    public class NewsPost
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string Slug { get; set; }
        public string Content { get; set; }
        public DateTime PublicationDate { get; set; }

        // Foreign Key to the User who authored the post
        public int? AuthorUserId { get; set; }
        public ApplicationUser? AuthorUser { get; set; } // Navigation property
        public string? ThumbnailImageUrl { get; set; }
    }

    public class Event
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string Description { get; set; }
        public DateTime EventDate { get; set; }
        public TimeSpan EventTime { get; set; }
        public string Location { get; set; }
        public string ImageUrl { get; set; }
        public bool IsFeatured { get; set; }
    }

    public class Testimonial
    {
        [Key]
        public int Id { get; set; }
        public string Quote { get; set; }
        public string ParentName { get; set; }
        public string? ChildName { get; set; }
        public string? ChildAge { get; set; }
        public DateTime DateSubmitted { get; set; }
```

```
        public bool IsApproved { get; set; }
    }
}
```

---

5. Media & Assets Model

C#
```
using System.ComponentModel.DataAnnotations;
using System.Collections.Generic;

namespace LittleSprouts.Web.Models
{
    public class GalleryImage
    {
        [Key]
        public int Id { get; set; }
        public string Title { get; set; }
        public string FileUrl { get; set; }
        public string ThumbnailUrl { get; set; }
        public string Category { get; set; }
        public bool ParentalConsentConfirmed { get; set; } // CRUCIAL for
nurseries!
        public DateTime UploadDate { get; set; }

        // Foreign Key to the User who uploaded the image
        public int? UploadedByUserId { get; set; }
        public ApplicationUser? UploadedByUser { get; set; } // Navigation
property
    }
}
```

---

6. User Interactions Models

C#
```
using System.ComponentModel.DataAnnotations;

namespace LittleSprouts.Web.Models
{
    public class ContactSubmission
    {
        [Key]
        public int Id { get; set; }
        public string SenderName { get; set; }
        public string SenderEmail { get; set; }
        public string? SenderPhone { get; set; }
        public string? ChildAge { get; set; }
        public string Message { get; set; }
        public DateTime SubmissionDate { get; set; }
        public bool IsRead { get; set; }
    }

    public class NewsletterSubscriber
    {
```

```
        [Key]
        public int Id { get; set; }
        public string EmailAddress { get; set; }
        public DateTime SubscriptionDate { get; set; }
        public bool IsConfirmed { get; set; }
        public DateTime? UnsubscribedDate { get; set; }
    }
}
```

---

7. User & Role Management Models (NEW)

C#

```
using System.ComponentModel.DataAnnotations;
using System.Collections.Generic;

namespace LittleSprouts.Web.Models
{
    // Represents a user who can log into the website's admin backend
    public class ApplicationUser
    {
        [Key]
        public int Id { get; set; } // Primary Key
        [Required]
        [StringLength(50)]
        public string Username { get; set; } // Unique, for login
        [Required]
        public string PasswordHash { get; set; } // Store hashed password
        [Required]
        [StringLength(100)]
        public string Email { get; set; }
        [StringLength(50)]
        public string? FirstName { get; set; }
        [StringLength(50)]
        public string? LastName { get; set; }
        public bool IsActive { get; set; } = true;
        public DateTime DateCreated { get; set; } = DateTime.UtcNow;
        public DateTime? LastLoginDate { get; set; } // Nullable

        // Navigation property for roles (many-to-many relationship)
        public ICollection<UserRole> UserRoles { get; set; } = new
List<UserRole>();
    }

    // Defines the different roles a user can have (e.g., Administrator,
ContentEditor)
    public class ApplicationRole
    {
        [Key]
        public int Id { get; set; } // Primary Key
        [Required]
        [StringLength(50)]
        public string Name { get; set; } // e.g., "Administrator",
"ContentEditor"
        public string? Description { get; set; }
```

```
        // Navigation property for users (many-to-many relationship)
        public ICollection<UserRole> UserRoles { get; set; } = new
List<UserRole>();
    }

    // Join table for the many-to-many relationship between Users and Roles
    public class UserRole
    {
        // Composite Primary Key configured in DbContext's OnModelCreating
        public int UserId { get; set; }
        public int RoleId { get; set; }

        // Navigation properties
        public ApplicationUser User { get; set; } = null!; // null-forgiving
operator
        public ApplicationRole Role { get; set; } = null!; // null-forgiving
operator
    }
}
```

Updated Entity Framework Core `DbContext` Setup

Now, we need to add the new `DbSet` properties for `ApplicationUser`, `ApplicationRole`, and `UserRole`, and also configure the many-to-many relationship in `OnModelCreating`.

C#

```
using Microsoft.EntityFrameworkCore;
using LittleSprouts.Web.Models; // Ensure this namespace is included

namespace LittleSprouts.Web.Data
{
    public class ApplicationDbContext : DbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
options)
            : base(options)
        {
        }

        // DbSets for existing models
        public DbSet<NurseryInfo> NurseryInfo { get; set; }
        public DbSet<TeamMember> TeamMembers { get; set; }
        public DbSet<Page> Pages { get; set; }
        public DbSet<CurriculumItem> CurriculumItems { get; set; }
        public DbSet<Policy> Policies { get; set; }
        public DbSet<FAQ> FAQs { get; set; }
        public DbSet<Vacancy> Vacancies { get; set; }
        public DbSet<NewsPost> NewsPosts { get; set; }
        public DbSet<Event> Events { get; set; }
        public DbSet<Testimonial> Testimonials { get; set; }
        public DbSet<GalleryImage> GalleryImages { get; set; }
        public DbSet<ContactSubmission> ContactSubmissions { get; set; }
        public DbSet<NewsletterSubscriber> NewsletterSubscribers { get; set; }
```

```
        // NEW DbSets for User and Role Management
        public DbSet<ApplicationUser> Users { get; set; }
        public DbSet<ApplicationRole> Roles { get; set; }
        public DbSet<UserRole> UserRoles { get; set; } // Explicitly declare
the join table

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            base.OnModelCreating(modelBuilder);

            // Configure the many-to-many relationship for Users and Roles
            modelBuilder.Entity<UserRole>()
                .HasKey(ur => new { ur.UserId, ur.RoleId }); // Composite
Primary Key

            modelBuilder.Entity<UserRole>()
                .HasOne(ur => ur.User)
                .WithMany(u => u.UserRoles)
                .HasForeignKey(ur => ur.UserId);

            modelBuilder.Entity<UserRole>()
                .HasOne(ur => ur.Role)
                .WithMany(r => r.UserRoles)
                .HasForeignKey(ur => ur.RoleId);

            // Optional: Configure unique index for Username and RoleName
            modelBuilder.Entity<ApplicationUser>()
                .HasIndex(u => u.Username)
                .IsUnique();

            modelBuilder.Entity<ApplicationRole>()
                .HasIndex(r => r.Name)
                .IsUnique();

            // Optional: Configure foreign key for content modified/created by
users
            modelBuilder.Entity<Page>()
                .HasOne(p => p.LastUpdatedByUser)
                .WithMany() // Or specify a collection if you track all pages
modified by a user
                .HasForeignKey(p => p.LastUpdatedByUserId)
                .IsRequired(false); // Can be nullable

            modelBuilder.Entity<NewsPost>()
                .HasOne(np => np.AuthorUser)
                .WithMany() // Or specify a collection for all posts by an
author
                .HasForeignKey(np => np.AuthorUserId)
                .IsRequired(false);

            modelBuilder.Entity<GalleryImage>()
                .HasOne(gi => gi.UploadedByUser)
                .WithMany() // Or specify a collection for all images uploaded
by a user
                .HasForeignKey(gi => gi.UploadedByUserId)
                .IsRequired(false);
        }
```

```
        }
    }
```