Controllers

Categorycontroller.cs

using Microsoft.AspNetCore.Mvc;

using TrainingFPTCo.DataDBContext;

using TrainingFPTCo.Models;

using TrainingFPTCo.Models.Queries;

using TrainingFPTCo.Helpers;

namespace TrainingFPTCo.Controllers

{

public class CategoryController : Controller

{

private readonly TrainingDbContext \_dbContext;

public CategoryController(TrainingDbContext dbContext)

{

\_dbContext = dbContext;

}

[HttpGet]

public IActionResult Index(string SearchString, string FilterStatus)

{

CategoryViewModel categoryModel = new CategoryViewModel();

categoryModel.CategoryDetailList = new List<CategoryDetail>();

var dataCategory = new CategoryQuery().GetAllCategories(SearchString, FilterStatus);

foreach (var item in dataCategory)

{

categoryModel.CategoryDetailList.Add(new CategoryDetail

{

Id = item.Id,

Name = item.Name,

Description = item.Description,

PosterNameImage = item.PosterNameImage,

Status = item.Status,

CreatedAt = item.CreatedAt,

UpdatedAt = item.UpdatedAt

});

}

ViewData["currentFilter"] = SearchString;

ViewBag.FilterStatus = FilterStatus;

return View(categoryModel);

}

[HttpGet]

public IActionResult Add()

{

CategoryDetail model = new CategoryDetail();

return View(model);

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Add(CategoryDetail category, IFormFile PosterImage)

{

if (ModelState.IsValid)

{

try

{

// upload file

string uniquePoster = UploadFileHelper.UpLoadFile(PosterImage,"images");

int idCategory = new CategoryQuery().InsertItemCategory(category.Name, category.Description, uniquePoster, category.Status);

if (idCategory > 0)

{

// thong bao trang thai

TempData["saveStatus"] = true; // luu thanh cong

}

else

{

// thong bao trang thai

TempData["saveStatus"] = false; // luu that bai

}

}

catch (Exception ex)

{

TempData["saveStatus"] = false; // luu that bai

}

// quay ve trang list categories

return RedirectToAction(nameof(CategoryController.Index), "Category");

}

return View(category);

}

[HttpGet]

public IActionResult Update(int id = 0)

{

CategoryDetail categoryDetail = new CategoryQuery().GetDataCategoryById(id);

return View(categoryDetail);

}

[HttpPost]

public IActionResult Update(CategoryDetail category, IFormFile PosterImage)

{

try

{

var detail = new CategoryQuery().GetDataCategoryById(category.Id);

string uniqueFilePoster = detail.PosterNameImage;

// nguoi dung co upload file ko?

if (category.PosterImage != null)

{

// co upload de thay anh poster

uniqueFilePoster = UploadFileHelper.UpLoadFile(PosterImage, "images");

}

bool updateCategory = new CategoryQuery().UpdateCategoryById(category.Name, category.Description, uniqueFilePoster, category.Status, category.Id);

if (updateCategory)

{

TempData["updateStatus"] = true;

}

else

{

TempData["updateStatus"] = false;

}

return RedirectToAction(nameof(CategoryController.Index), "Category");

}

catch (Exception ex)

{

//return Ok(ex.Message);

return View(category);

}

}

[HttpGet]

public IActionResult Delete(int id = 0)

{

// xu ly xoa danh muc khoa hoc

bool deleteCategory = new CategoryQuery().DeleteItemCatgoryById(id);

if (deleteCategory)

{

TempData["statusDelete"] = true;

}

else

{

TempData["statusDelete"] = false;

}

return RedirectToAction(nameof(CategoryController.Index), "Category");

}

}

}

Coursescontroller.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using TrainingFPTCo.Helpers;

using TrainingFPTCo.Models;

using TrainingFPTCo.Models.Queries;

namespace TrainingFPTCo.Controllers

{

public class CoursesController : Controller

{

[HttpGet]

public IActionResult Index()

{

CourseViewModel course = new CourseViewModel();

course.CourseDetailsList = new List<CourseDetail>();

var dataCourses = new CourseQuery().GetAllDataCourses();

foreach (var data in dataCourses)

{

course.CourseDetailsList.Add(new CourseDetail

{

Id = data.Id,

Name = data.Name,

CategoryId = data.CategoryId,

Description = data.Description,

Status = data.Status,

ViewStartDate = data.ViewStartDate,

ViewEndDate = data.ViewEndDate,

StartDate = data.StartDate,

EndDate = data.EndDate,

ViewImageCouser = data.ViewImageCouser,

ViewCategoryName = data.ViewCategoryName

});

}

return View(course);

}

[HttpGet]

public IActionResult Add()

{

CourseDetail course = new CourseDetail();

List<SelectListItem> itemCategories = new List<SelectListItem>();

var dataCategory = new CategoryQuery().GetAllCategories(null, null);

foreach (var item in dataCategory)

{

itemCategories.Add(new SelectListItem

{

Value = item.Id.ToString(),

Text = item.Name

});

}

ViewBag.Categories = itemCategories;

return View(course);

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Add(CourseDetail course, IFormFile Image)

{

if (ModelState.IsValid)

{

// xu ly insert course vao database

try

{

string nameImageCourse = UploadFileHelper.UpLoadFile(Image, "images");

int idCourse = new CourseQuery().InsertCourse(

course.Name,

course.CategoryId,

course.Description,

course.StartDate,

course.EndDate,

course.Status,

nameImageCourse

);

if (idCourse > 0)

{

TempData["saveStatus"] = true;

}

else

{

TempData["saveStatus"] = false;

}

// quay lai trang danh sach courses

return RedirectToAction(nameof(CoursesController.Index), "Courses");

}

catch (Exception ex)

{

return Ok(ex.Message);

}

}

List<SelectListItem> itemCategories = new List<SelectListItem>();

var dataCategory = new CategoryQuery().GetAllCategories(null, null);

foreach (var item in dataCategory)

{

itemCategories.Add(new SelectListItem

{

Value = item.Id.ToString(),

Text = item.Name

});

}

ViewBag.Categories = itemCategories;

return View(course);

}

[HttpPost]

public JsonResult Delete(int id = 0)

{

bool deleteCourse = new CourseQuery().DeleteCourseById(id);

if (deleteCourse)

{

return Json(new {cod = 200, message = "Successfully"});

}

return Json(new { cod = 500, message = "Failure" });

}

[HttpGet]

public IActionResult Update(int id)

{

CourseDetail detail = new CourseQuery().GetDetailCourseById(id);

List<SelectListItem> itemCategories = new List<SelectListItem>();

var dataCategory = new CategoryQuery().GetAllCategories(null, null);

foreach (var item in dataCategory)

{

itemCategories.Add(new SelectListItem

{

Value = item.Id.ToString(),

Text = item.Name

});

}

ViewBag.Categories = itemCategories;

return View(detail);

}

[HttpPost]

public IActionResult Update(CourseDetail courseDetail, IFormFile Image)

{

try

{

var infoCourse = new CourseQuery().GetDetailCourseById(courseDetail.Id);

string imageCourse = infoCourse.ViewImageCouser;

// check xem nguoi co thay anh hay ko?

if (courseDetail.Image != null)

{

// co muon thay anh

imageCourse = UploadFileHelper.UpLoadFile(Image, "images");

}

bool update = new CourseQuery().UpdateCourseById(

courseDetail.CategoryId,

courseDetail.Name,

courseDetail.Description,

imageCourse,

courseDetail.StartDate,

courseDetail.EndDate,

courseDetail.Status,

courseDetail.Id

);

if (update)

{

TempData["updateStatus"] = true;

}

else

{

TempData["updateStatus"] = false;

}

return RedirectToAction("Index", "Courses");

}

catch (Exception ex)

{

return Ok(ex.Message);

}

List<SelectListItem> itemCategories = new List<SelectListItem>();

var dataCategory = new CategoryQuery().GetAllCategories(null, null);

foreach (var item in dataCategory)

{

itemCategories.Add(new SelectListItem

{

Value = item.Id.ToString(),

Text = item.Name

});

}

ViewBag.Categories = itemCategories;

return View(courseDetail);

}

}

}

Dashboardcontroller.cs

using Microsoft.AspNetCore.Mvc;

namespace TrainingFPTCo.Controllers

{

public class DashboardController : Controller

{

public IActionResult Index()

{

if (string.IsNullOrEmpty(HttpContext.Session.GetString("SessionUserId")))

{

return RedirectToAction(nameof(LoginController.Index), "Login");

}

return View();

}

}

}

Homecontroller.cs

using Microsoft.AspNetCore.Mvc;

using System.Diagnostics;

using TrainingFPTCo.Models;

namespace TrainingFPTCo.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

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 });

}

}

}

Logincontroller.cs

using Microsoft.AspNetCore.Mvc;

using TrainingFPTCo.Models;

using TrainingFPTCo.Models.Queries;

namespace TrainingFPTCo.Controllers

{

public class LoginController : Controller

{

[HttpGet]

public IActionResult Index()

{

LoginViewModel model = new LoginViewModel();

return View(model);

}

[HttpPost]

public IActionResult Index(LoginViewModel model)

{

model = new LoginQuery().CheckUserLogin(model.UserName, model.Password);

if (string.IsNullOrEmpty(model.Id) || string.IsNullOrEmpty(model.Email))

{

// nguoi dung dang nhap linh tinh

// co mot thong bao loi ra view

ViewData["MessageLogin"] = "Account invalid";

return View(model);

}

// luu thong tin nguoi dung vao session

if (string.IsNullOrEmpty(HttpContext.Session.GetString("SessionUserId")))

{

HttpContext.Session.SetString("SessionUserId", model.Id);

HttpContext.Session.SetString("SessionUsername", model.UserName);

HttpContext.Session.SetString("SessionRolesId", model.RolesId);

HttpContext.Session.SetString("SessionEmail", model.Email);

}

// cho chuyen vao trang Home

return RedirectToAction(nameof(DashboardController.Index), "Dashboard");

}

[HttpPost]

public IActionResult Logout()

{

// xoa session da tao ra o login

// quay ve trang dang nhap

if (!string.IsNullOrEmpty(HttpContext.Session.GetString("SessionUserId")))

{

// xoa session

HttpContext.Session.Remove("SessionUserId");

HttpContext.Session.Remove("SessionUsername");

HttpContext.Session.Remove("SessionRolesId");

HttpContext.Session.Remove("SessionEmail");

}

// quay ve trang dang nhap

return RedirectToAction(nameof(LoginController.Index), "Login");

}

}

}

DataDBcontext

Category.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Diagnostics.CodeAnalysis;

namespace TrainingFPTCo.DataDBContext

{

public class Category

{

[Key]

public int Id { get; set; }

[Column("Name", TypeName ="Varchar(50)"), Required]

public string Name { get; set; }

[Column("Description", TypeName="Varchar(200)"), AllowNull]

public string? Description { get; set; }

[Column("ParentId", TypeName ="integer"), Required]

public int ParentId { get; set; }

[Column("PosterImage", TypeName="Varchar(200)"), Required]

public string PosterImage { get; set; }

[Column("Status", TypeName = "Varchar(20)"), Required]

public string Status { get; set; }

[AllowNull]

public DateTime? CreatedAt { get; set; }

[AllowNull]

public DateTime? UpdatedAt { get; set; }

[AllowNull]

public DateTime? DeletedAt { get; set; }

}

}

Roles.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Diagnostics.CodeAnalysis;

namespace TrainingFPTCo.DataDBContext

{

public class Roles

{

// tao bang roles trong database DemoTraining

// su dung migrations cua ASP.net core

[Key]

public int Id { get; set; }

[Column("NameRole", TypeName="Varchar(50)"), Required]

public required string NameRole { get; set; }

[Column("Description", TypeName="Varchar(200)"), AllowNull]

public string? Description { get; set; }

[Column("Status", TypeName ="Varchar(20)"), Required]

public required string Status { get; set; }

[AllowNull]

public DateTime? CreatedAt { get; set; }

[AllowNull]

public DateTime? UpdatedAt { get;set; }

[AllowNull]

public DateTime? DeletedAt { get; set; }

}

}

trainingDbContext.cs

using Microsoft.EntityFrameworkCore;

namespace TrainingFPTCo.DataDBContext

{

public class TrainingDbContext : DbContext

{

public TrainingDbContext(DbContextOptions<TrainingDbContext> options) : base(options) { }

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

}

}

users.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace TrainingFPTCo.DataDBContext

{

public class Users

{

[Key]

public int Id { get; set; }

[ForeignKey("RolesId"), Required]

public required Roles Roles { get; set; }

[Column("Username", TypeName="Varchar(100)"), Required]

public required string UserName { get; set; }

[Column("Password", TypeName = "Varchar(100)"), Required]

public required string Password { get; set; }

[Column("Extracode", TypeName = "Varchar(100)"), Required]

public required string Extracode { get; set; }

[Column("Email", TypeName = "Varchar(100)"), Required]

public required string Email { get; set; }

[Column("Phone", TypeName = "Varchar(20)"), Required]

public required string Phone { get; set; }

}

}

Dbcontext

CategoryDBContext.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Diagnostics.CodeAnalysis;

namespace TrainingFPTCo.DBContext

{

public class CategoryDBContext

{

[Key]

public int Id { get; set; }

[Column("Name", TypeName ="Varchar(100)")]

public string? Name { get; set; }

[Column("Description", TypeName ="Varchar(200)")]

public string? Description { get; set; }

[Column("PosterImage", TypeName ="Varchar(200)")]

public string? PosterImage { get; set; }

[Column("ParentId", TypeName ="Integer")]

public int? ParentId { get; set; }

[Column("Status", TypeName = "Varchar(20)")]

public string? Status { get; set; }

[AllowNull]

public DateTime? CreatedAt { get; set; }

[AllowNull]

public DateTime? UpdatedAt { get; set; }

[AllowNull]

public DateTime? DeletedAt { get; set; }

}

}

CourseDBContext.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Diagnostics.CodeAnalysis;

namespace TrainingFPTCo.DBContext

{

public class CourseDBContext

{

// noi dinh nghia cac truong cua bang du lieu Courses

[Key]

public int Id { get; set; }

[ForeignKey("CategoryId"), Required]

public required CategoryDBContext categories { get; set; }

public required int CategoryId { get; set; }

[Column("NameCourse", TypeName = "Varchar(60)"), Required]

public required string NameCourse { get; set; }

[Column("Description", TypeName = "Varchar(200)"), AllowNull]

public string? Description { get; set; }

[Column("Image", TypeName = "Varchar(200)"), Required]

public required string Image { get; set; }

[Column("LikeCourse", TypeName = "Integer"), AllowNull]

public int? LikeCourse { get; set; }

[Column("StarCourse", TypeName = "Integer"), AllowNull]

public int? StarCourse { get; set; }

[Column("Status", TypeName = "Varchar(20)"), Required]

public required string Status { get; set; }

[AllowNull]

public DateTime? CreatedAt { get; set; }

[AllowNull]

public DateTime? UpdatedAt { get; set; }

[AllowNull]

public DateTime? DeletedAt { get; set; }

}

}

RolesDBContext.cs

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Diagnostics.CodeAnalysis;

namespace TrainingFPTCo.DBContext

{

public class RolesDBContext

{

// tao bang roles trong database DemoTraining

// su dung migrations cua ASP.net core

[Key]

public int Id { get; set; }

[Column("NameRole", TypeName="Varchar(50)"), Required]

public required string NameRole { get; set; }

[Column("Description", TypeName="Varchar(200)"), AllowNull]

public string? Description { get; set; }

[Column("Status", TypeName ="Varchar(20)"), Required]

public required string Status { get; set; }

[AllowNull]

public DateTime? CreatedAt { get; set; }

[AllowNull]

public DateTime? UpdatedAt { get;set; }

[AllowNull]

public DateTime? DeletedAt { get; set; }

}

}

TrainingDBContext.cs

using Microsoft.EntityFrameworkCore;

namespace TrainingFPTCo.DBContext

{

public class TrainingDBContext : DbContext

{

public TrainingDBContext(DbContextOptions<TrainingDBContext> options) : base(options) { }

public DbSet<RolesDBContext> Roles { get; set; }

public DbSet<CategoryDBContext> Categories { get; set; }

public DbSet<CourseDBContext> Courses { get; set; }

}

}

Helpers

Commontext.cs

using System.Text.RegularExpressions;

namespace TrainingFPTCo.Helpers

{

public static class CommonText

{

public static string GenerateSlug(this string phrase)

{

string str = phrase.RemoveAccent().ToLower();

// invalid chars

str = Regex.Replace(str, @"[^a-z0-9\s-]", "");

// convert multiple spaces into one space

str = Regex.Replace(str, @"\s+", " ").Trim();

// cut and trim

str = str.Substring(0, str.Length <= 45 ? str.Length : 45).Trim();

str = Regex.Replace(str, @"\s", "-"); // hyphens

return str;

}

public static string RemoveAccent(this string txt)

{

byte[] bytes = System.Text.Encoding.GetEncoding("Cyrillic").GetBytes(txt);

return System.Text.Encoding.ASCII.GetString(bytes);

}

}

}

Uploadfilehelper.cs

namespace TrainingFPTCo.Helpers

{

public static class UploadFileHelper

{

public static string UpLoadFile(IFormFile file, string folderName)

{

string uniqueFileName;

try

{

string pathUploadServer = string.Empty;

switch (folderName.ToLower())

{

case "images":

pathUploadServer = "wwwroot\\uploads\\images";

break;

case "videos":

pathUploadServer = "wwwroot\\uploads\\videos";

break;

case "documents":

pathUploadServer = "wwwroot\\uploads\\documents";

break;

default:

pathUploadServer = "wwwroot\\uploads\\images";

break;

}

string fileName = file.FileName;

string fileNameWithoutExtension = Path.GetFileNameWithoutExtension(fileName);

fileNameWithoutExtension = CommonText.GenerateSlug(fileNameWithoutExtension);

string ext = Path.GetExtension(fileName);

// tao ten file khong trung nhau

string uniqueStr = Guid.NewGuid().ToString(); // random string

string fileNameUpload = uniqueStr + "-" + fileNameWithoutExtension + ext;

string uploadPath = Path.Combine(Directory.GetCurrentDirectory(), pathUploadServer, fileNameUpload);

var stream = new FileStream(uploadPath, FileMode.Create);

file.CopyToAsync(stream);

uniqueFileName = fileNameUpload;

}

catch (Exception ex)

{

uniqueFileName = ex.Message.ToString();

}

return uniqueFileName;

}

}

}

Migrations

20240131021514\_Roles.cs

using System;

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace TrainingFPTCo.Migrations

{

/// <inheritdoc />

public partial class Roles : Migration

{

/// <inheritdoc />

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "Roles",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

NameRole = table.Column<string>(type: "Varchar(50)", nullable: false),

Description = table.Column<string>(type: "Varchar(200)", nullable: true),

Status = table.Column<string>(type: "Varchar(20)", nullable: false),

CreatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

UpdatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

DeletedAt = table.Column<DateTime>(type: "datetime2", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_Roles", x => x.Id);

});

}

/// <inheritdoc />

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "Roles");

}

}

}

20240131024959\_Categories.cs

using System;

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace TrainingFPTCo.Migrations

{

/// <inheritdoc />

public partial class Categories : Migration

{

/// <inheritdoc />

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "Categories",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

NameCategory = table.Column<string>(type: "Varchar(100)", nullable: false),

Description = table.Column<string>(type: "Varchar(200)", nullable: true),

PosterImage = table.Column<string>(type: "Varchar(200)", nullable: false),

ParentId = table.Column<int>(type: "Integer", nullable: false),

Status = table.Column<string>(type: "Varchar(20)", nullable: false),

CreatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

UpdatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

DeletedAt = table.Column<DateTime>(type: "datetime2", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_Categories", x => x.Id);

});

}

/// <inheritdoc />

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "Categories");

}

}

}

20240131032331\_Courses.cs

using System;

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace TrainingFPTCo.Migrations

{

/// <inheritdoc />

public partial class Courses : Migration

{

/// <inheritdoc />

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "Courses",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

CategoryId = table.Column<int>(type: "int", nullable: false),

NameCourse = table.Column<string>(type: "Varchar(60)", nullable: false),

Description = table.Column<string>(type: "Varchar(200)", nullable: true),

Image = table.Column<string>(type: "Varchar(200)", nullable: false),

LikeCourse = table.Column<int>(type: "Integer", nullable: true),

StarCourse = table.Column<int>(type: "Integer", nullable: true),

Status = table.Column<string>(type: "Varchar(20)", nullable: false),

CreatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

UpdatedAt = table.Column<DateTime>(type: "datetime2", nullable: true),

DeletedAt = table.Column<DateTime>(type: "datetime2", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_Courses", x => x.Id);

table.ForeignKey(

name: "FK\_Courses\_Categories\_CategoryId",

column: x => x.CategoryId,

principalTable: "Categories",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateIndex(

name: "IX\_Courses\_CategoryId",

table: "Courses",

column: "CategoryId");

}

/// <inheritdoc />

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "Courses");

}

}

}

TrainingDBContextModelSnapshot.cs

// <auto-generated />

using System;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Infrastructure;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

using TrainingFPTCo.DBContext;

#nullable disable

namespace TrainingFPTCo.Migrations

{

[DbContext(typeof(TrainingDBContext))]

partial class TrainingDBContextModelSnapshot : ModelSnapshot

{

protected override void BuildModel(ModelBuilder modelBuilder)

{

#pragma warning disable 612, 618

modelBuilder

.HasAnnotation("ProductVersion", "8.0.1")

.HasAnnotation("Relational:MaxIdentifierLength", 128);

SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

modelBuilder.Entity("TrainingFPTCo.DBContext.CategoryDBContext", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<DateTime?>("CreatedAt")

.HasColumnType("datetime2");

b.Property<DateTime?>("DeletedAt")

.HasColumnType("datetime2");

b.Property<string>("Description")

.HasColumnType("Varchar(200)")

.HasColumnName("Description");

b.Property<string>("NameCategory")

.IsRequired()

.HasColumnType("Varchar(100)")

.HasColumnName("NameCategory");

b.Property<int>("ParentId")

.HasColumnType("Integer")

.HasColumnName("ParentId");

b.Property<string>("PosterImage")

.IsRequired()

.HasColumnType("Varchar(200)")

.HasColumnName("PosterImage");

b.Property<string>("Status")

.IsRequired()

.HasColumnType("Varchar(20)")

.HasColumnName("Status");

b.Property<DateTime?>("UpdatedAt")

.HasColumnType("datetime2");

b.HasKey("Id");

b.ToTable("Categories");

});

modelBuilder.Entity("TrainingFPTCo.DBContext.CourseDBContext", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<int>("CategoryId")

.HasColumnType("int");

b.Property<DateTime?>("CreatedAt")

.HasColumnType("datetime2");

b.Property<DateTime?>("DeletedAt")

.HasColumnType("datetime2");

b.Property<string>("Description")

.HasColumnType("Varchar(200)")

.HasColumnName("Description");

b.Property<string>("Image")

.IsRequired()

.HasColumnType("Varchar(200)")

.HasColumnName("Image");

b.Property<int?>("LikeCourse")

.HasColumnType("Integer")

.HasColumnName("LikeCourse");

b.Property<string>("NameCourse")

.IsRequired()

.HasColumnType("Varchar(60)")

.HasColumnName("NameCourse");

b.Property<int?>("StarCourse")

.HasColumnType("Integer")

.HasColumnName("StarCourse");

b.Property<string>("Status")

.IsRequired()

.HasColumnType("Varchar(20)")

.HasColumnName("Status");

b.Property<DateTime?>("UpdatedAt")

.HasColumnType("datetime2");

b.HasKey("Id");

b.HasIndex("CategoryId");

b.ToTable("Courses");

});

modelBuilder.Entity("TrainingFPTCo.DBContext.RolesDBContext", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<DateTime?>("CreatedAt")

.HasColumnType("datetime2");

b.Property<DateTime?>("DeletedAt")

.HasColumnType("datetime2");

b.Property<string>("Description")

.HasColumnType("Varchar(200)")

.HasColumnName("Description");

b.Property<string>("NameRole")

.IsRequired()

.HasColumnType("Varchar(50)")

.HasColumnName("NameRole");

b.Property<string>("Status")

.IsRequired()

.HasColumnType("Varchar(20)")

.HasColumnName("Status");

b.Property<DateTime?>("UpdatedAt")

.HasColumnType("datetime2");

b.HasKey("Id");

b.ToTable("Roles");

});

modelBuilder.Entity("TrainingFPTCo.DBContext.CourseDBContext", b =>

{

b.HasOne("TrainingFPTCo.DBContext.CategoryDBContext", "categories")

.WithMany()

.HasForeignKey("CategoryId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.Navigation("categories");

});

#pragma warning restore 612, 618

}

}

}

Models

Queries

CategoryQuery.cs

using Microsoft.Data.SqlClient;

namespace TrainingFPTCo.Models.Queries

{

public class CategoryQuery

{

public bool UpdateCategoryById (

string nameCategory,

string description,

string image,

string status,

int id

)

{

bool statusUpdate = false;

using (SqlConnection connection = Database.GetSqlConnection())

{

string sqlQuery = "UPDATE [Categories] SET [Name] = @nameCategory, [Description] = @description, [PosterImage] = @posterImage, [Status] = @status, [UpdatedAt] = @updatedAt WHERE [Id] = @id AND [DeletedAt] IS NULL";

connection.Open();

SqlCommand cmd = new SqlCommand( sqlQuery, connection );

cmd.Parameters.AddWithValue("@nameCategory", nameCategory ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@description", description ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@posterImage", image ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@status", status ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@id", id);

cmd.Parameters.AddWithValue("@updatedAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

cmd.ExecuteNonQuery();

connection.Close();

statusUpdate = true;

}

return statusUpdate;

}

public bool DeleteItemCatgoryById(int id = 0)

{

bool statusDelete = false;

using (SqlConnection connection = Database.GetSqlConnection())

{

// khong xoa cung du lieu

// xoa mem du lieu, danh dau truong DeletedAt(update sql)

string sqlQuery = "UPDATE [Categories] SET [DeletedAt] = @deletedAt WHERE [Id] = @id";

SqlCommand command = new SqlCommand(sqlQuery, connection);

connection.Open();

command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@deletedAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

command.ExecuteNonQuery(); // thuc thi cau lenh sql

connection.Close(); // ngat ket noi toi database

statusDelete = true;

}

return statusDelete;

}

public int InsertItemCategory(

string nameCategory,

string description,

string image,

string status

) {

int lastInsertId = 0;

string sqlInsertCategory = "INSERT INTO [Categories]([Name],[Description], [PosterImage], [ParentId], [Status], [CreatedAt]) VALUES(@nameCategory, @description, @image, @parentId, @status, @createdAt) SELECT SCOPE\_IDENTITY()";

// SELECT SCOPE\_IDENTITY(): lay ra id vua dc them moi.

using (SqlConnection connection = Database.GetSqlConnection())

{

connection.Open(); // mo ket noi database

SqlCommand command = new SqlCommand( sqlInsertCategory, connection );

command.Parameters.AddWithValue("@nameCategory", nameCategory ?? DBNull.Value.ToString());

command.Parameters.AddWithValue("@description", description ?? DBNull.Value.ToString());

command.Parameters.AddWithValue("@image", image ?? DBNull.Value.ToString());

command.Parameters.AddWithValue("@parentId", 0);

command.Parameters.AddWithValue("@status", status ?? DBNull.Value.ToString());

command.Parameters.AddWithValue("@createdAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

lastInsertId = Convert.ToInt32( command.ExecuteScalar() );

connection.Close();// ngat ket noi database

}

return lastInsertId;

}

public CategoryDetail GetDataCategoryById(int id = 0)

{

CategoryDetail categoryDetail = new CategoryDetail();

using (SqlConnection connection = Database.GetSqlConnection())

{

string sqlQuery = "SELECT \* FROM [Categories] WHERE [Id] = @id AND [DeletedAt] IS NULL";

connection.Open(); // mo ket noi toi database

SqlCommand command = new SqlCommand(sqlQuery, connection);

command.Parameters.AddWithValue("@id", id);

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

categoryDetail.Id = Convert.ToInt32(reader["Id"]);

categoryDetail.Name = reader["Name"].ToString();

categoryDetail.Description = reader["Description"].ToString();

categoryDetail.PosterNameImage = reader["PosterImage"].ToString();

categoryDetail.Status = reader["Status"].ToString();

}

connection.Close(); // ngat ket noi toi database

}

}

return categoryDetail;

}

public List<CategoryDetail> GetAllCategories(string? SearchString, string? FilterStatus)

{

List<CategoryDetail> category = new List<CategoryDetail>();

using (SqlConnection conn = Database.GetSqlConnection())

{

// chi lay ra danh sach khong bi xoa

string sqlData = string.Empty;

if (FilterStatus != null)

{

sqlData = "SELECT \* FROM [Categories] WHERE [Name] LIKE @keyWord AND [DeletedAt] IS NULL AND [Status] = @status";

}

else

{

sqlData = "SELECT \* FROM [Categories] WHERE [Name] LIKE @keyWord AND [DeletedAt] IS NULL";

}

conn.Open();

SqlCommand command = new SqlCommand(sqlData, conn);

command.Parameters.AddWithValue("@keyWord", "%"+SearchString+"%" ?? DBNull.Value.ToString());

if (FilterStatus != null)

{

command.Parameters.AddWithValue("@status", FilterStatus ?? DBNull.Value.ToString());

}

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

CategoryDetail cate = new CategoryDetail();

cate.Id = Convert.ToInt32(reader["Id"]);

cate.Name = reader["Name"].ToString();

cate.Description = reader["Description"].ToString();

cate.PosterNameImage = reader["PosterImage"].ToString();

cate.Status = reader["Status"].ToString();

cate.CreatedAt = Convert.ToDateTime(reader["CreatedAt"]);

category.Add(cate);

}

conn.Close();

}

}

return category;

}

}

}

Coursequery.cs

using Microsoft.Data.SqlClient;

namespace TrainingFPTCo.Models.Queries

{

public class CourseQuery

{

public bool UpdateCourseById(

int categoryId,

string nameCourse,

string? description,

string image,

string startDate,

string? endDate,

string status,

int id

)

{

bool checkUpdate = false;

using (SqlConnection connection = Database.GetSqlConnection())

{

string sql = "UPDATE [Courses] SET [CategoryId] = @CategoryId, [Name] = @Name, [Description] = @Description, [Image] = @Image, [StartDate] = @StartDate, [EndDate] = @EndDate, [Status] = @Status, [UpdatedAt] = @UpdatedAt WHERE [Id] = @Id AND [DeletedAt] IS NULL";

connection.Open();

SqlCommand cmd = new SqlCommand( sql, connection );

cmd.Parameters.AddWithValue("@CategoryId", categoryId);

cmd.Parameters.AddWithValue("@Name", nameCourse);

cmd.Parameters.AddWithValue("@Description", description ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@Image", image);

cmd.Parameters.AddWithValue("@StartDate", startDate);

cmd.Parameters.AddWithValue("@EndDate", endDate ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@Status", status);

cmd.Parameters.AddWithValue("UpdatedAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

cmd.Parameters.AddWithValue("@Id", id);

cmd.ExecuteNonQuery();

checkUpdate = true;

connection.Close();

}

return checkUpdate;

}

public bool DeleteCourseById(int id)

{

bool checkDelete = false;

using (SqlConnection connection = Database.GetSqlConnection())

{

string sqlQuery = "UPDATE [Courses] SET [DeletedAt] = @DeletedAt WHERE [Id] = @id";

connection.Open();

SqlCommand cmd = new SqlCommand(sqlQuery, connection);

cmd.Parameters.AddWithValue("@DeletedAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

cmd.Parameters.AddWithValue("@Id", id);

cmd.ExecuteNonQuery();

checkDelete = true;

connection.Close();

}

return checkDelete;

}

public CourseDetail GetDetailCourseById(int id)

{

CourseDetail detail = new CourseDetail();

using (SqlConnection connection = Database.GetSqlConnection())

{

string sql = "SELECT \* FROM [Courses] WHERE [Id] = @id AND [DeletedAt] IS NULL";

connection.Open();

SqlCommand cmd = new SqlCommand (sql, connection);

cmd.Parameters.AddWithValue("@id", id);

using (SqlDataReader reader = cmd.ExecuteReader())

{

while (reader.Read())

{

detail.Id = Convert.ToInt32(reader["Id"]);

detail.Name = reader["Name"].ToString();

detail.CategoryId = Convert.ToInt32(reader["CategoryId"]);

detail.Description = reader["Description"].ToString();

detail.StartDate = reader["StartDate"].ToString();

detail.EndDate = reader["EndDate"].ToString();

detail.Status = reader["Status"].ToString();

detail.ViewImageCouser = reader["Image"].ToString();

}

}

connection.Close ();

}

return detail;

}

public List<CourseDetail> GetAllDataCourses()

{

List<CourseDetail> courses = new List<CourseDetail>();

using (SqlConnection connection = Database.GetSqlConnection())

{

string sql = "SELECT [co].\*, CONVERT(VARCHAR(10), [co].[StartDate], 101) AS Start\_Date,CONVERT(VARCHAR(10), [co].[EndDate], 101) AS End\_Date, [ca].[Name] AS [CategoryName] FROM [Courses] AS [co] INNER JOIN [Categories] AS [ca] ON [co].[CategoryId] = [ca].[Id] WHERE [co].[DeletedAt] IS NULL";

connection.Open();

SqlCommand cmd = new SqlCommand(sql, connection);

using (SqlDataReader reader = cmd.ExecuteReader())

{

while (reader.Read())

{

CourseDetail detail = new CourseDetail();

detail.Id = Convert.ToInt32(reader["Id"]);

detail.Name = reader["Name"].ToString() ?? DBNull.Value.ToString();

detail.Description = reader["Description"].ToString();

detail.CategoryId = Convert.ToInt32(reader["CategoryId"]);

detail.ViewStartDate = reader["Start\_Date"].ToString();

detail.ViewEndDate = reader["End\_Date"].ToString();

detail.StartDate = reader["StartDate"].ToString();

detail.EndDate = reader["EndDate"].ToString();

detail.ViewImageCouser = reader["Image"].ToString();

detail.ViewCategoryName = reader["CategoryName"].ToString();

detail.Status = reader["Status"].ToString() ?? DBNull.Value.ToString();

courses.Add(detail);

}

connection.Close();

}

}

return courses;

}

public int InsertCourse(

string nameCourse,

int categoryId,

string? description,

string startDate,

string? endDate,

string status,

string imageCourse

)

{

int IdCourse = 0;

using (SqlConnection connection = Database.GetSqlConnection())

{

string sqlQuery = "INSERT INTO [Courses]([CategoryId], [Name], [Description],[Image], [StartDate], [EndDate], [Status], [CreatedAt]) VALUES(@CategoryId, @Name, @Description, @Image, @StartDate, @EndDate, @Status, @CreatedAt) SELECT SCOPE\_IDENTITY()";

SqlCommand cmd = new SqlCommand( sqlQuery, connection );

connection.Open();

cmd.Parameters.AddWithValue("@CategoryId", categoryId);

cmd.Parameters.AddWithValue("@Name", nameCourse);

cmd.Parameters.AddWithValue ("@Description", description ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@Image", imageCourse);

cmd.Parameters.AddWithValue("@StartDate", startDate);

cmd.Parameters.AddWithValue("@EndDate", endDate ?? DBNull.Value.ToString());

cmd.Parameters.AddWithValue("@Status", status);

cmd.Parameters.AddWithValue("@CreatedAt", DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));

IdCourse = Convert.ToInt32(cmd.ExecuteScalar());

connection.Close();

}

return IdCourse;

}

}

}

Loginquery.cs

using Microsoft.Data.SqlClient;

namespace TrainingFPTCo.Models.Queries

{

public class LoginQuery

{

// viet ham kiem tra dang nhap vao he thong

public LoginViewModel CheckUserLogin(string? username, string? password)

{

LoginViewModel dataUser = new LoginViewModel();

using (SqlConnection conn = Database.GetSqlConnection())

{

string querySql = "SELECT \* FROM [Users] WHERE [Username] = @username AND [Password] = @password";

// @usernam va @password : tham so truyen cau lenh sql

SqlCommand cmd = new SqlCommand(querySql, conn);

cmd.Parameters.AddWithValue("@username", username ?? (object)DBNull.Value);

cmd.Parameters.AddWithValue("@password", password ?? (object)DBNull.Value);

// mo ket noi database

conn.Open();

// thuc thi cau lenh sql

using (SqlDataReader reader = cmd.ExecuteReader())

{

while (reader.Read())

{

// do du lieu thu table users sang model LoginViewModel

dataUser.Id = reader["Id"].ToString();

dataUser.UserName = reader["Username"].ToString();

dataUser.ExtraCode = reader["ExtraCode"].ToString();

dataUser.Email = reader["Email"].ToString();

dataUser.RolesId = reader["RolesId"].ToString();

dataUser.Phone = reader["Phone"].ToString();

dataUser.FullName = reader["FullName"].ToString();

}

// ngat ket noi database

conn.Close();

}

}

return dataUser;

}

}

}

CategoryViewModel.cs

using System.ComponentModel.DataAnnotations;

using System.Diagnostics.CodeAnalysis;

using TrainingFPTCo.Validations;

namespace TrainingFPTCo.Models

{

public class CategoryViewModel

{

public List<CategoryDetail> CategoryDetailList { get; set; }

}

public class CategoryDetail

{

public int Id { get; set; }

[Required(ErrorMessage = "Enter name's Category, please")]

public string Name { get; set; }

public string? Description { get; set; }

[Required(ErrorMessage ="Choose Status, please")]

public string Status { get; set; }

[Required(ErrorMessage = "Choose file, please")]

[AllowExtensionFile(new string[] { ".png", ".jpg", ".jpeg" })]

[AllowSizeFile(5 \* 1024 \* 1024)]

public IFormFile? PosterImage { get; set; }

// view ten anh

[AllowNull]

public string? PosterNameImage { get; set; }

public DateTime? CreatedAt { get; set; }

public DateTime? UpdatedAt { get; set; }

public DateTime? DeletedAt { get; set; }

}

}

CourseViewModel.cs

using System.ComponentModel.DataAnnotations;

using TrainingFPTCo.Validations;

namespace TrainingFPTCo.Models

{

public class CourseViewModel

{

public List<CourseDetail> CourseDetailsList { get; set; }

}

public class CourseDetail

{

public int Id { get; set; }

[Required(ErrorMessage = "Choose Category, please")]

public int CategoryId { get; set; }

public string? ViewCategoryName { get; set; }

[Required(ErrorMessage = "Enter name's course, please")]

public string Name { get; set; }

public string? Description { get; set; }

[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

public string StartDate { get; set; }

public string? ViewStartDate { get; set; }

[DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]

public string? EndDate { get; set; }

public string? ViewEndDate { get; set; }

[Required(ErrorMessage = "Choose file image, please")]

[AllowExtensionFile(new string[] {".png", ".jpg", ".jpeg"})]

[AllowSizeFile(5 \* 1024 \* 1024)]

public IFormFile Image { get; set; }

public string? ViewImageCouser { get; set; }

public int? LikeCourse { get; set; }

public int? StarCourse { get; set; }

[Required(ErrorMessage = "Choose Status, please")]

public string Status { get; set; }

public DateTime? CreatedAt { get; set; }

public DateTime? UpdatedAt { get; set; }

public DateTime? DeletedAt { get; set; }

}

}

ErrorViewModel.cs

namespace TrainingFPTCo.Models

{

public class ErrorViewModel

{

public string? RequestId { get; set; }

public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

}

}

LoginViewModel.cs

namespace TrainingFPTCo.Models

{

public class LoginViewModel

{

public string? Id { get; set; }

public string? RolesId { get; set; }

public string? UserName { get; set; }

public string? Password { get; set; }

public string? Email { get; set; }

public string? Phone { get; set;}

public string? ExtraCode { get; set; }

public string? FullName { get; set; }

}

}

Validations

AllowExtensionFile.cs

using System.ComponentModel.DataAnnotations;

namespace TrainingFPTCo.Validations

{

public class AllowExtensionFile : ValidationAttribute

{

private readonly string[] \_extensions;

public AllowExtensionFile(string[] extensions)

{

\_extensions = extensions;

}

protected override ValidationResult IsValid(object value, ValidationContext validationContext)

{

var file = value as IFormFile;

if (file != null)

{

var extension = Path.GetExtension(file.FileName);

if (!\_extensions.Contains(extension.ToLower()))

{

return new ValidationResult(GetErrorMessage());

}

}

return ValidationResult.Success;

}

private string GetErrorMessage()

{

return "This photo extension is not allowed !";

}

}

}

AllowSizeFile.cs

using System.ComponentModel.DataAnnotations;

namespace TrainingFPTCo.Validations

{

public class AllowSizeFile : ValidationAttribute

{

private readonly int \_maxSizeFile;

public AllowSizeFile(int maxSizeFile)

{

\_maxSizeFile = maxSizeFile;

}

protected override ValidationResult IsValid(object value, ValidationContext validationContext)

{

var file = value as IFormFile;

if (file != null)

{

if (file.Length > \_maxSizeFile)

{

return new ValidationResult(GetErrorMessage());

}

}

return ValidationResult.Success;

}

private string GetErrorMessage()

{

return $"Maximum a allowed file size is {\_maxSizeFile} bytes. ";

}

}

}

View

Category

Add.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CategoryDetail

@{

ViewData["title"] = "Add Category";

ViewData["breadcrumb"] = "Category";

ViewData["breadcrumb\_active"] = "Create";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3> Add new Category page !</h3>

<a class="btn btn-primary" href="@Url.Action("Index", "Category")"> Back to List Categories</a>

<div class="card mt-5">

<div class="card-header">

<h5>Create Category</h5>

</div>

<div class="card-body">

<**form** enctype="multipart/form-data" **asp-action**="Add">

<div class="row">

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

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

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

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

</div>

<div class="form-group mb-3">

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

<**select** class="form-control" **asp-for**="Status">

<**option** **value**="Active">Active</**option**>

<**option** **value**="Deactive">Deactive</**option**>

</**select**>

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

</div>

<div class="form-group mb-3">

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

<**input** **type**="file" class="form-control" **asp-for**="PosterImage" />

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

</div>

</div>

<div class="col-sm-12 col-md-6">

<div class="form-group">

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

<**textarea** rows="5" class="form-control" **asp-for**="Description"></**textarea**>

</div>

<button class="my-3 btn btn-primary" type="submit"> Save </button>

</div>

</div>

</**form**>

</div>

</div>

</div>

</div>

Index.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CategoryViewModel

@{

ViewData["title"] = "Categories";

ViewData["breadcrumb"] = "Home";

ViewData["breadcrumb\_active"] = "Category";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

bool statusAdd = Convert.ToBoolean(TempData["saveStatus"]);

bool statusDel = Convert.ToBoolean(TempData["statusDelete"]);

bool statusUpdate = Convert.ToBoolean(TempData["updateStatus"]);

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3> This is Category page !</h3>

<a class="btn btn-primary" href="@Url.Action("Add", "Category")"> Add new Category</a>

<form asp-action="Index" method="get" class="my-3">

<div class="form-actions no-color">

<p class="mb-0">

Find by Name:<input value="@ViewData["currentFilter"]" type="text" name="SearchString" />

<span> | </span>

<select name="FilterStatus">

<option value=""> -- Status --</option>

@if (ViewBag.FilterStatus == "Active")

{

<option value="Active" selected="selected"> Active </option>

}

else

{

<option value="Active"> Active </option>

}

@if (ViewBag.FilterStatus == "Deactive")

{

<option value="Deactive" selected="selected"> Deactive </option>

}

else

{

<option value="Deactive"> Deactive </option>

}

</select>

<span> | </span>

<button type="submit" class="btn btn-primary btn-sm mb-0"> Filter and Search </button>

<span> | </span>

<a asp-action="Index"> Back To list </a>

</p>

</div>

</form>

@if (TempData["saveStatus"] != null)

{

@if (statusAdd)

{

<div class="text-success my-3"> Add Successfully </div>

}

else

{

<div class="text-danger my-3"> Add Failure </div>

}

}

@if (TempData["statusDelete"] != null)

{

@if (statusDel)

{

<div class="text-success my-3"> Delete Successfully </div>

}

else

{

<div class="text-danger my-3"> Delete Failure </div>

}

}

@if (TempData["updateStatus"] != null)

{

@if (statusUpdate)

{

<div class="text-success my-3"> Uodate Successfully </div>

}

else

{

<div class="text-danger my-3"> Update Failure </div>

}

}

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

<thead class="table-primary">

<tr>

<th>ID</th>

<th>Name</th>

<th>Image</th>

<th>Description</th>

<th>Status</th>

<th>Created At</th>

<th>Updated At</th>

<th colspan="2" width="10%">Action</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model.CategoryDetailList)

{

<tr>

<td>@item.Id</td>

<td>@item.Name</td>

<td width="10%">

<img class="img-fluid" src="/uploads/images/@item.PosterNameImage" alt="@item.Name" />

</td>

<td>@item.Description</td>

<td>@item.Status</td>

<td>@item.CreatedAt</td>

<td>@item.UpdatedAt</td>

<td>

<a href="@Url.Action("Update", "Category", new {id = @item.Id})" class="btn btn-info">Edit</a>

</td>

<td>

<a href="@Url.Action("Delete", "Category", new {id = @item.Id})" class="btn btn-danger">Delete</a>

</td>

</tr>

}

</tbody>

</table>

</div>

</div>

Update.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CategoryDetail

@{

ViewData["title"] = "Update Category";

ViewData["breadcrumb"] = "Category";

ViewData["breadcrumb\_active"] = "Update";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3> Update Category page !</h3>

<a class="btn btn-primary" href="@Url.Action("Index", "Category")"> Back to List Categories</a>

<div class="card mt-5">

<div class="card-header">

<h5>Update Category</h5>

</div>

<div class="card-body">

<form enctype="multipart/form-data" asp-action="Update">

<div class="row">

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

<label class="control-label" asp-for="Name"></label>

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

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

@\* id cua category \*@

<input type="hidden" asp-for="Id" />

</div>

<div class="form-group mb-3">

<label asp-for="Status" class="control-label"></label>

<select class="form-control" asp-for="Status">

<option value="Active">Active</option>

<option value="Deactive">Deactive</option>

</select>

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

</div>

<div class="form-group mb-3">

<label asp-for="PosterImage" class="control-label"></label>

<input type="file" class="form-control" asp-for="PosterImage" />

@\* <span asp-validation-for="PosterImage" class="text-danger"></span> \*@

<br/>

<img class="img-fluid" width="50%" src="/uploads/images/@Model.PosterNameImage" />

</div>

</div>

<div class="col-sm-12 col-md-6">

<div class="form-group">

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

<textarea rows="5" class="form-control" asp-for="Description"></textarea>

</div>

<button class="my-3 btn btn-primary" type="submit"> Save </button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

Courses

Add.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CourseDetail

@{

ViewData["title"] = "Add Courses";

ViewData["breadcrumb"] = "Home";

ViewData["breadcrumb\_active"] = "Add New Courses";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3> Add new Courses page !</h3>

<a class="btn btn-primary" href="@Url.Action("Index", "Courses")"> Back to List Courses</a>

<div class="card mt-5">

<div class="card-header">

<h5>Create Course</h5>

</div>

<div class="card-body">

<form enctype="multipart/form-data" method="post" asp-action="Add">

<div class="row">

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

<label asp-for="Name"></label>

<input type="text" class="form-control" asp-for="Name" />

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

</div>

<div class="form-group mb-3">

<label asp-for="CategoryId">Category</label>

<select

asp-for="CategoryId"

class="form-control"

asp-items="@(new SelectList(ViewBag.Categories, "Value", "Text"))"></select>

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

</div>

<div class="form-group mb-3">

<label asp-for="Description"></label>

<textarea rows="5" asp-for="Description" class="form-control"></textarea>

</div>

</div>

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

<label asp-for="StartDate">Start date</label>

<input class="form-control" type="date" asp-for="StartDate" />

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

</div>

<div class="form-group mb-3">

<label asp-for="EndDate">End date</label>

<input class="form-control" type="date" asp-for="EndDate" />

</div>

<div class="form-group mb-3">

<label asp-for="Status"></label>

<select asp-for="Status" class="form-control">

<option value="Active"> Active</option>

<option value="Deactive"> Deactive</option>

</select>

<span asp-validation-for="Status"></span>

</div>

<div class="form-group mb-3">

<label asp-for="Image"></label>

<input type="file" asp-for="Image" class="form-control" />

<span asp-validation-for="Image"></span>

</div>

<button type="submit" class="my-3 btn btn-primary"> Save </button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

Index.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CourseViewModel

@{

ViewData["title"] = "Courses";

ViewData["breadcrumb"] = "Home";

ViewData["breadcrumb\_active"] = "Courses";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

bool saveStatus = Convert.ToBoolean(TempData["saveStatus"]);

bool updateStatus = Convert.ToBoolean(TempData["updateStatus"]);

// lay ra giay o thoi diem hien tai

var Timestamp = new DateTimeOffset(DateTime.UtcNow).ToUnixTimeSeconds();

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3 class="text-center"> This is Courses page !</h3>

<a class="btn btn-primary" href="@Url.Action("Add", "Courses")"> Add new Courses</a>

@if (TempData["saveStatus"] != null)

{

@if (saveStatus)

{

<div class="text-success my-3"> Add Successfully</div>

}

else

{

<div class="text-danger my-3"> Add Failure</div>

}

}

@if (TempData["updateStatus"] != null)

{

@if (updateStatus)

{

<div class="text-success my-3"> Update Successfully</div>

}

else

{

<div class="text-danger my-3"> Update Failure</div>

}

}

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

<thead class="table-primary">

<tr>

<th>ID</th>

<th>Name</th>

<th>Category</th>

<th>Image</th>

<th>Description</th>

<th>Start Date</th>

<th>End Date</th>

<th>Status</th>

<th width="10%" class="text-center" colspan="2">

Actions

</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model.CourseDetailsList)

{

<tr class="row-course-@item.Id">

<td>@item.Id</td>

<td>@item.Name</td>

<td>@item.ViewCategoryName</td>

<td width="15%">

<img class="img-fluid img-thumbnail" alt="@item.Name" src="/uploads/images/@item.ViewImageCouser" />

</td>

<td>@item.Description</td>

<td>@item.ViewStartDate</td>

<td>@item.ViewEndDate</td>

<td>@item.Status</td>

<td>

<a class="btn btn-info btn-sm" href="@Url.Action("Update", "Courses",new {id = item.Id})"> Edit</a>

</td>

<td>

<button id="@item.Id" class="btn btn-danger btn-sm js-delete-course"> Delete</button>

</td>

</tr>

}

</tbody>

</table>

</div>

</div>

@section javascripts

{

<script src="/js/course.js?v=@Timestamp"></script>

}

Update.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.CourseDetail

@{

ViewData["title"] = "Update Course";

ViewData["breadcrumb"] = "Home";

ViewData["breadcrumb\_active"] = "Update Course";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3> Add new Courses page !</h3>

<a class="btn btn-primary" href="@Url.Action("Index", "Courses")"> Back to List Courses</a>

<div class="card mt-5">

<div class="card-header">

<h5>Update Course</h5>

</div>

<div class="card-body">

<form enctype="multipart/form-data" method="post" asp-action="Update">

<div class="row">

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

<label asp-for="Name"></label>

<input type="text" class="form-control" asp-for="Name" />

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

<input type="hidden" asp-for="Id" />

</div>

<div class="form-group mb-3">

<label asp-for="CategoryId">Category</label>

<select asp-for="CategoryId"

class="form-control"

asp-items="@(new SelectList(ViewBag.Categories, "Value", "Text"))"></select>

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

</div>

<div class="form-group mb-3">

<label asp-for="Description"></label>

<textarea rows="5" asp-for="Description" class="form-control"></textarea>

</div>

</div>

<div class="col-sm-12 col-md-6">

<div class="form-group mb-3">

<label>Start date</label>

<input class="form-control" type="date" asp-for="StartDate" asp-format="{0:yyyy-MM-dd}" />

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

</div>

<div class="form-group mb-3">

<label>End date</label>

<input class="form-control" type="date" asp-for="EndDate" asp-format="{0:yyyy-MM-dd}" />

</div>

<div class="form-group mb-3">

<label asp-for="Status"></label>

<select asp-for="Status" class="form-control">

<option value="Active"> Active</option>

<option value="Deactive"> Deactive</option>

</select>

<span asp-validation-for="Status"></span>

</div>

<div class="form-group mb-3">

<label asp-for="Image"></label>

<input type="file" asp-for="Image" class="form-control" />

<span asp-validation-for="Image"></span>

<br/>

<img width="50%" class="img-fluid" src="/uploads/images/@Model.ViewImageCouser" />

</div>

<button type="submit" class="my-3 btn btn-primary"> Save </button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

Dashboard

Index.cshtml

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@{

ViewData["title"] = "Dashboard";

ViewData["breadcrumb"] = "Home";

ViewData["breadcrumb\_active"] = "Dashboard";

Layout = "~/Views/Shared/\_LayoutMain.cshtml";

}

<div class="row">

<div class="col-sm-12 col-md-12">

<h3>This is dashboard page !</h3>

</div>

</div>

Home

Index.cshtml

@{

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

}

<div class="text-center">

<h1 class="display-4">Welcome</h1>

<p>Learn about <a href="https://learn.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>

</div>

Privacy.cshtml

@{

ViewData["Title"] = "Privacy Policy";

}

<h1>@ViewData["Title"]</h1>

<p>Use this page to detail your site's privacy policy.</p>

Login

@\*

For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@model TrainingFPTCo.Models.LoginViewModel

@{

ViewData["title"] = "Login";

Layout = "~/Views/Shared/\_LayoutLogin.cshtml";

}

<div class="row justify-content-center">

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

<div class="card-group d-block d-md-flex row">

<div class="card col-md-7 p-4 mb-0">

<div class="card-body">

<h1>Login FPT Training</h1>

<p class="text-medium-emphasis">Sign In to your account</p>

@if (ViewData["MessageLogin"] != null)

{

<p class="text-danger text-center">

@ViewData["MessageLogin"]

</p>

}

@using(Html.BeginForm("Index", "Login", FormMethod.Post))

{

<div class="input-group mb-3">

<span class="input-group-text">

<svg class="icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-user"></use>

</svg>

</span>

@\* <input class="form-control" type="text" placeholder="Username"> \*@

@Html.TextBoxFor(m => m.UserName, new { @class

= "form-control", @id

= "username"

})

</div>

<div class="input-group mb-4">

<span class="input-group-text">

<svg class="icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-lock-locked"></use>

</svg>

</span>

@\* <input class="form-control" type="password" placeholder="Password"> \*@

@Html.PasswordFor(m => m.Password, new { @class = "form-control", @id="password" })

</div>

<div class="row">

<div class="col-6">

<button class="btn btn-primary px-4" type="submit">Login</button>

</div>

</div>

}

</div>

</div>

</div>

</div>

</div>

Shared

Layout.cshtml

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - TrainingFPTCo</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/TrainingFPTCo.styles.css" asp-append-version="true" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">TrainingFPTCo</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2024 - TrainingFPTCo - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

Layoutlogin.cshtml

<!DOCTYPE html>

<html lang="en">

<head>

<base href="./">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">

<meta name="description" content="FPT Training">

<meta name="author" content="IT0502 - BTec">

<meta name="keyword" content="FPT Training">

<title>FPT Training - @ViewData["title"]</title>

<link rel="apple-touch-icon" sizes="57x57" href="~/training/assets/favicon/apple-icon-57x57.png">

<link rel="apple-touch-icon" sizes="60x60" href="~/training/assets/favicon/apple-icon-60x60.png">

<link rel="apple-touch-icon" sizes="72x72" href="~/training/assets/favicon/apple-icon-72x72.png">

<link rel="apple-touch-icon" sizes="76x76" href="~/training/assets/favicon/apple-icon-76x76.png">

<link rel="apple-touch-icon" sizes="114x114" href="~/training/assets/favicon/apple-icon-114x114.png">

<link rel="apple-touch-icon" sizes="120x120" href="~/training/assets/favicon/apple-icon-120x120.png">

<link rel="apple-touch-icon" sizes="144x144" href="~/training/assets/favicon/apple-icon-144x144.png">

<link rel="apple-touch-icon" sizes="152x152" href="~/training/assets/favicon/apple-icon-152x152.png">

<link rel="apple-touch-icon" sizes="180x180" href="~/training/assets/favicon/apple-icon-180x180.png">

<link rel="icon" type="image/png" sizes="192x192" href="~/training/assets/favicon/android-icon-192x192.png">

<link rel="icon" type="image/png" sizes="32x32" href="~/training/assets/favicon/favicon-32x32.png">

<link rel="icon" type="image/png" sizes="96x96" href="~/training/assets/favicon/favicon-96x96.png">

<link rel="icon" type="image/png" sizes="16x16" href="~/training/assets/favicon/favicon-16x16.png">

<link rel="manifest" href="~/training/assets/favicon/manifest.json">

<meta name="msapplication-TileColor" content="#ffffff">

<meta name="msapplication-TileImage" content="/training/assets/favicon/ms-icon-144x144.png">

<meta name="theme-color" content="#ffffff">

<!-- Vendors styles-->

<link rel="stylesheet" href="~/training/vendors/simplebar/css/simplebar.css">

<link rel="stylesheet" href="~/training/css/vendors/simplebar.css">

<!-- Main styles for this application-->

<link href="~/training/css/style.css" rel="stylesheet">

<!-- We use those styles to show code examples, you should remove them in your application.-->

<link href="~/training/css/examples.css" rel="stylesheet">

</head>

<body>

<div class="bg-light min-vh-100 d-flex flex-row align-items-center">

<div class="container">

@RenderBody()

</div>

</div>

<!-- CoreUI and necessary plugins-->

<script src="~/training/vendors/coreui/coreui/js/coreui.bundle.min.js"></script>

<script src="~/training/vendors/simplebar/js/simplebar.min.js"></script>

</body>

</html>

layoutMain.cshtml

@using Microsoft.AspNetCore.Session

@inject Microsoft.AspNetCore.Http.IHttpContextAccessor HttpContextAccessor

<!DOCTYPE html>

<html lang="en">

<head>

<base href="./">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">

<meta name="description" content="FPT Training">

<meta name="author" content="IT0502 - BTec">

<meta name="keyword" content="FPT Training">

<title>FPT Training - @ViewData["title"]</title>

<link rel="apple-touch-icon" sizes="57x57" href="/training/assets/favicon/apple-icon-57x57.png">

<link rel="apple-touch-icon" sizes="60x60" href="/training/assets/favicon/apple-icon-60x60.png">

<link rel="apple-touch-icon" sizes="72x72" href="/training/assets/favicon/apple-icon-72x72.png">

<link rel="apple-touch-icon" sizes="76x76" href="/training/assets/favicon/apple-icon-76x76.png">

<link rel="apple-touch-icon" sizes="114x114" href="/training/assets/favicon/apple-icon-114x114.png">

<link rel="apple-touch-icon" sizes="120x120" href="/training/assets/favicon/apple-icon-120x120.png">

<link rel="apple-touch-icon" sizes="144x144" href="/training/assets/favicon/apple-icon-144x144.png">

<link rel="apple-touch-icon" sizes="152x152" href="/training/assets/favicon/apple-icon-152x152.png">

<link rel="apple-touch-icon" sizes="180x180" href="/training/assets/favicon/apple-icon-180x180.png">

<link rel="icon" type="image/png" sizes="192x192" href="/training/assets/favicon/android-icon-192x192.png">

<link rel="icon" type="image/png" sizes="32x32" href="/training/assets/favicon/favicon-32x32.png">

<link rel="icon" type="image/png" sizes="96x96" href="/training/assets/favicon/favicon-96x96.png">

<link rel="icon" type="image/png" sizes="16x16" href="/training/assets/favicon/favicon-16x16.png">

<link rel="manifest" href="/training/assets/favicon/manifest.json">

<meta name="msapplication-TileColor" content="#ffffff">

<meta name="msapplication-TileImage" content="/training/assets/favicon/ms-icon-144x144.png">

<meta name="theme-color" content="#ffffff">

<!-- Vendors styles-->

<link rel="stylesheet" href="/training/vendors/simplebar/css/simplebar.css">

<link rel="stylesheet" href="/training/css/vendors/simplebar.css">

<!-- Main styles for this application-->

<link href="/training/css/style.css" rel="stylesheet">

<!-- We use those styles to show code examples, you should remove them in your application.-->

<link href="/training/css/examples.css" rel="stylesheet">

@\* <link href="/training/vendors/coreui/chartjs/css/coreui-chartjs.css" rel="stylesheet"> \*@

</head>

<body>

<div class="sidebar sidebar-dark sidebar-fixed" id="sidebar">

<div class="sidebar-brand d-none d-md-flex">

<svg class="sidebar-brand-full" width="118" height="46" alt="CoreUI Logo">

<use xlink:href="/training/assets/brand/coreui.svg#full"></use>

</svg>

<svg class="sidebar-brand-narrow" width="46" height="46" alt="CoreUI Logo">

<use xlink:href="/training/assets/brand/coreui.svg#signet"></use>

</svg>

</div>

<ul class="sidebar-nav" data-coreui="navigation" data-simplebar="">

<li class="nav-item">

<a class="nav-link" href="@Url.Action("Index","Dashboard")">

<svg class="nav-icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-speedometer"></use>

</svg> Dashboard<span class="badge badge-sm bg-info ms-auto">NEW</span>

</a>

</li>

<li class="nav-title">Management</li>

<li class="nav-item">

<a class="nav-link" href="@Url.Action("Index","Account")">

<svg class="nav-icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-pencil"></use>

</svg> Accounts

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="@Url.Action("Index","Category")">

<svg class="nav-icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-pencil"></use>

</svg>Categories

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="@Url.Action("Index","Courses")">

<svg class="nav-icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-pencil"></use>

</svg>Courses

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="@Url.Action("Index","Topic")">

<svg class="nav-icon">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-pencil"></use>

</svg>Topic

</a>

</li>

</ul>

<button class="sidebar-toggler" type="button" data-coreui-toggle="unfoldable"></button>

</div>

<div class="wrapper d-flex flex-column min-vh-100 bg-light">

<header class="header header-sticky mb-4">

<div class="container-fluid">

<button class="header-toggler px-md-0 me-md-3" type="button" onclick="coreui.Sidebar.getInstance(document.querySelector('#sidebar')).toggle()">

<svg class="icon icon-lg">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-menu"></use>

</svg>

</button><a class="header-brand d-md-none" href="#">

<svg width="118" height="46" alt="CoreUI Logo">

<use xlink:href="/training/assets/brand/coreui.svg#full"></use>

</svg></a>

<ul class="header-nav d-none d-md-flex">

<li class="nav-item"><a class="nav-link" href="javascript:void(0)">Hi : </a></li>

<li class="nav-item">

<a class="nav-link" href="javascript:void(0)">

@HttpContextAccessor.HttpContext.Session.GetString("SessionEmail")

</a>

</li>

</ul>

<ul class="header-nav ms-auto">

<li class="nav-item"><a class="nav-link" href="#">

<svg class="icon icon-lg">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-bell"></use>

</svg></a></li>

<li class="nav-item"><a class="nav-link" href="#">

<svg class="icon icon-lg">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-list-rich"></use>

</svg></a></li>

<li class="nav-item"><a class="nav-link" href="#">

<svg class="icon icon-lg">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-envelope-open"></use>

</svg></a></li>

</ul>

<ul class="header-nav ms-3">

<li class="nav-item dropdown"><a class="nav-link py-0" data-coreui-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">

<div class="avatar avatar-md"><img class="avatar-img" src="/training/assets/img/avatars/8.jpg" alt="user@email.com"></div>

</a>

<div class="dropdown-menu dropdown-menu-end pt-0">

<div class="dropdown-header bg-light py-2">

<div class="fw-semibold">Account</div>

</div><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-bell"></use>

</svg> Updates<span class="badge badge-sm bg-info ms-2">42</span></a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-envelope-open"></use>

</svg> Messages<span class="badge badge-sm bg-success ms-2">42</span></a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-task"></use>

</svg> Tasks<span class="badge badge-sm bg-danger ms-2">42</span></a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-comment-square"></use>

</svg> Comments<span class="badge badge-sm bg-warning ms-2">42</span></a>

<div class="dropdown-header bg-light py-2">

<div class="fw-semibold">Settings</div>

</div><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-user"></use>

</svg> Profile</a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-settings"></use>

</svg> Settings</a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-credit-card"></use>

</svg> Payments<span class="badge badge-sm bg-secondary ms-2">42</span></a><a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-file"></use>

</svg> Projects<span class="badge badge-sm bg-primary ms-2">42</span></a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">

<svg class="icon me-2">

<use xlink:href="vendors/coreui/icons/svg/free.svg#cil-lock-locked"></use>

</svg> Lock Account

</a>

@using(Html.BeginForm("Logout", "Login", FormMethod.Post))

{

<button class="dropdown-item" type="submit">

<svg class="icon me-2">

<use xlink:href="/training/vendors/coreui/icons/svg/free.svg#cil-account-logout"></use>

</svg> Logout

</button>

}

</div>

</li>

</ul>

</div>

<div class="header-divider"></div>

<div class="container-fluid">

<nav aria-label="breadcrumb">

<ol class="breadcrumb my-0 ms-2">

<li class="breadcrumb-item">

<span>@ViewData["breadcrumb"]</span>

</li>

<li class="breadcrumb-item active">

<span>@ViewData["breadcrumb\_active"]</span>

</li>

</ol>

</nav>

</div>

</header>

<div class="body flex-grow-1 px-3">

<div class="container-lg">

@RenderBody()

</div>

</div>

<footer class="footer">

<div>

<a href="https://btec.fpt.edu.vn/">FPT Training </a>

<a href="https://btec.fpt.edu.vn/">BTec FPTPolytechnic</a> © 2024 creativeLabs.

</div>

<div class="ms-auto">

Powered by&nbsp;

<a href="https://btec.fpt.edu.vn/">IT0502</a>

</div>

</footer>

</div>

<!-- CoreUI and necessary plugins-->

<script src="/training/vendors/coreui/coreui/js/coreui.bundle.min.js"></script>

<script src="/training/vendors/simplebar/js/simplebar.min.js"></script>

<!-- Plugins and scripts required by this view-->

@\* <script src="/training/vendors/chart.js/js/chart.min.js"></script>

<script src="/training/vendors/coreui/chartjs/js/coreui-chartjs.js"></script> \*@

<script src="/training/vendors/coreui/utils/js/coreui-utils.js"></script>

@\* <script src="/training/js/main.js"></script> \*@

<script src="/js/jquery-3.7.1.min.js"></script>

@await RenderSectionAsync("javascripts", required: false)

</body>

</html>

\_ValidationScriptsPartial.cshtml

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>

Error.cshtml

@model ErrorViewModel

@{

ViewData["Title"] = "Error";

}

<h1 class="text-danger">Error.</h1>

<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)

{

<p>

<strong>Request ID:</strong> <code>@Model.RequestId</code>

</p>

}

<h3>Development Mode</h3>

<p>

Swapping to <strong>Development</strong> environment will display more detailed information about the error that occurred.

</p>

<p>

<strong>The Development environment shouldn't be enabled for deployed applications.</strong>

It can result in displaying sensitive information from exceptions to end users.

For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE\_ENVIRONMENT</strong> environment variable to <strong>Development</strong>

and restarting the app.

</p>

Topic

\_ViewImports.cshtml

@using TrainingFPTCo

@using TrainingFPTCo.Models

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

\_ViewStart.cshtml

@{

Layout = "\_Layout";

}

Appsettings.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"ConnectionStrings": {

"myconn": "Server=21AK22-COM\\KHANHDNBH0021;Initial Catalog=Training;Integrated Security=True;"

},

"AllowedHosts": "\*"

}

Database.cs

using Microsoft.Data.SqlClient;

namespace TrainingFPTCo

{

public class Database

{

public static string GetStrConnection()

{

string strConnection = @"Data Source=21AK22-COM\KHANHDNBH0021;Initial Catalog=Training;Integrated Security=True";

return strConnection;

}

public static SqlConnection GetSqlConnection()

{

string strConnection = Database.GetStrConnection();

SqlConnection conn = new SqlConnection(strConnection);

return conn;

}

}

}

Program.cs

using TrainingFPTCo.DataDBContext;

using Microsoft.EntityFrameworkCore;

namespace TrainingFPTCo

{

public class Program

{

public static void Main(string[] args)

{

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

builder.Services.AddDistributedMemoryCache();

builder.Services.AddSession(options =>

{

options.Cookie.Name = ".AdventureWorks.Session";

options.IdleTimeout = TimeSpan.FromSeconds(10);

//options.Cookie.HttpOnly = true;

options.Cookie.IsEssential = true;

});

builder.Services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();

var provider = builder.Services.BuildServiceProvider();

var configuration = provider.GetRequiredService<IConfiguration>();

builder.Services.AddDbContext<TrainingDbContext>(item => item.UseSqlServer(configuration.GetConnectionString("myconn")));

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseSession();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Login}/{action=Index}/{id?}");

app.Run();

}

}

}