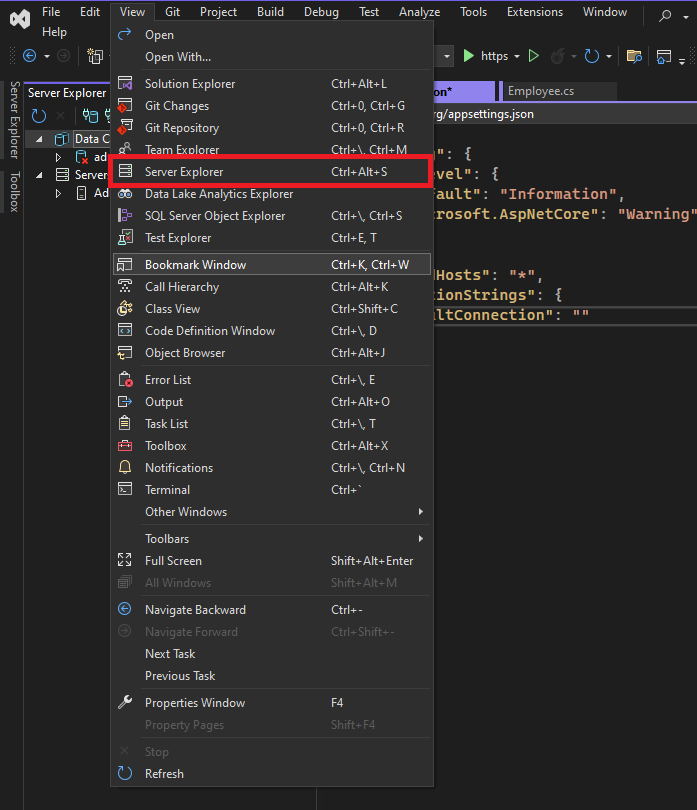
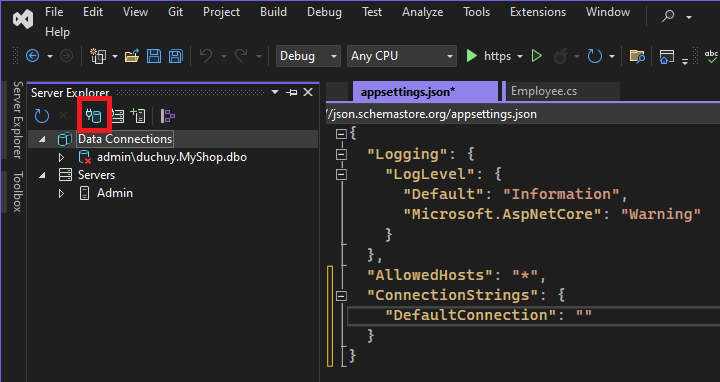
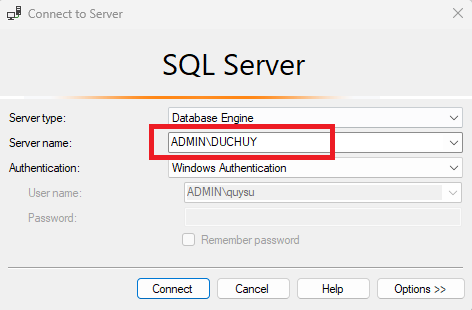
**Kết nối ASP.Net Core và SQL Server**

Sau khi tạo thành công Database, mở Project -> View -> Server Explorer -> Connect to Database

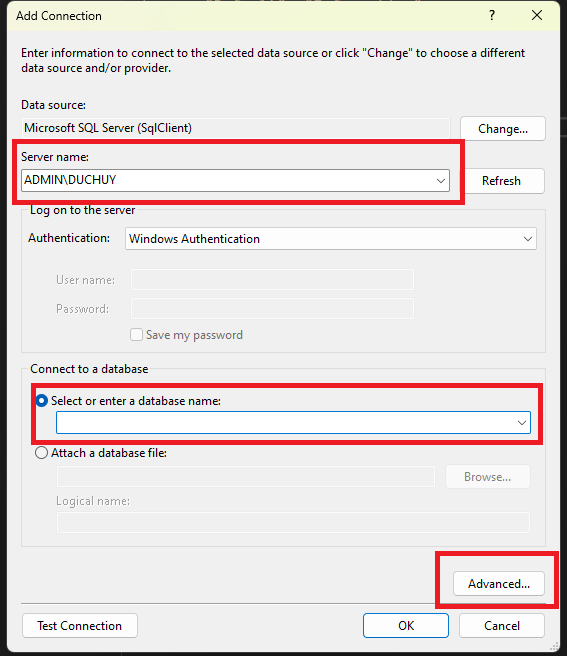




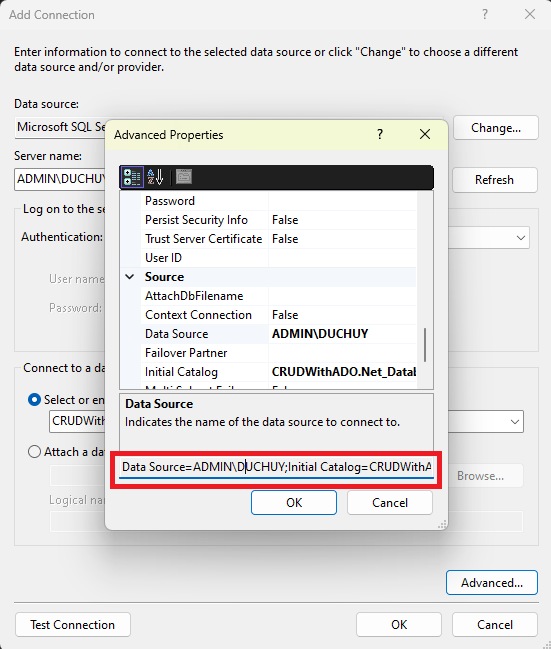
Mở SQL Server để lấy Server Name



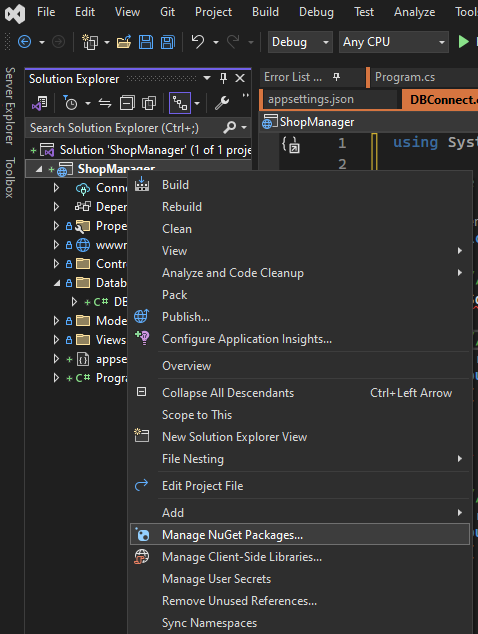
Paste vào ô server name và chọn database vừa tạo -> Advanced



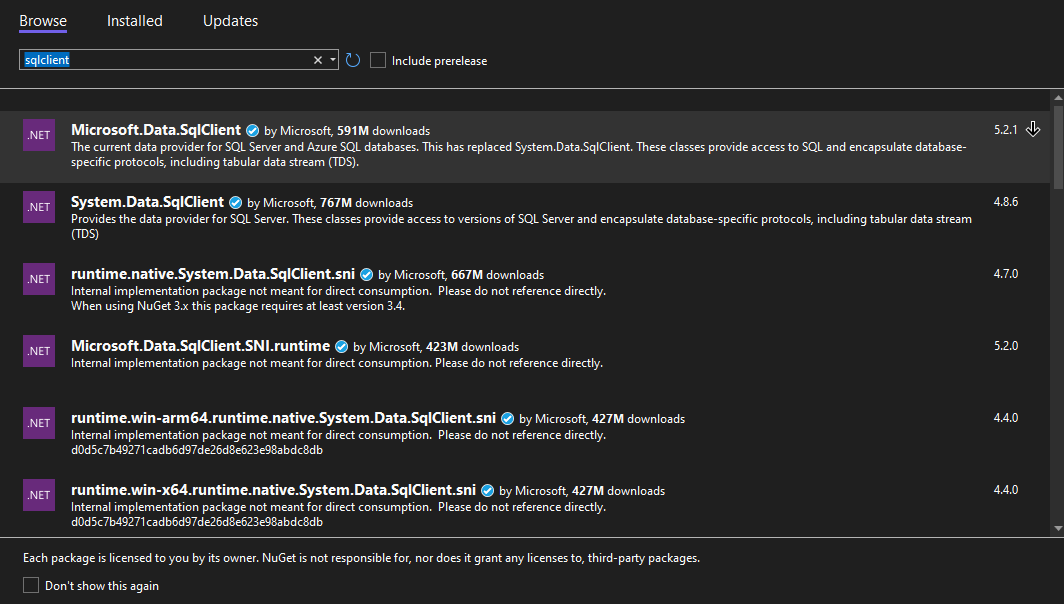
Copy Connection String

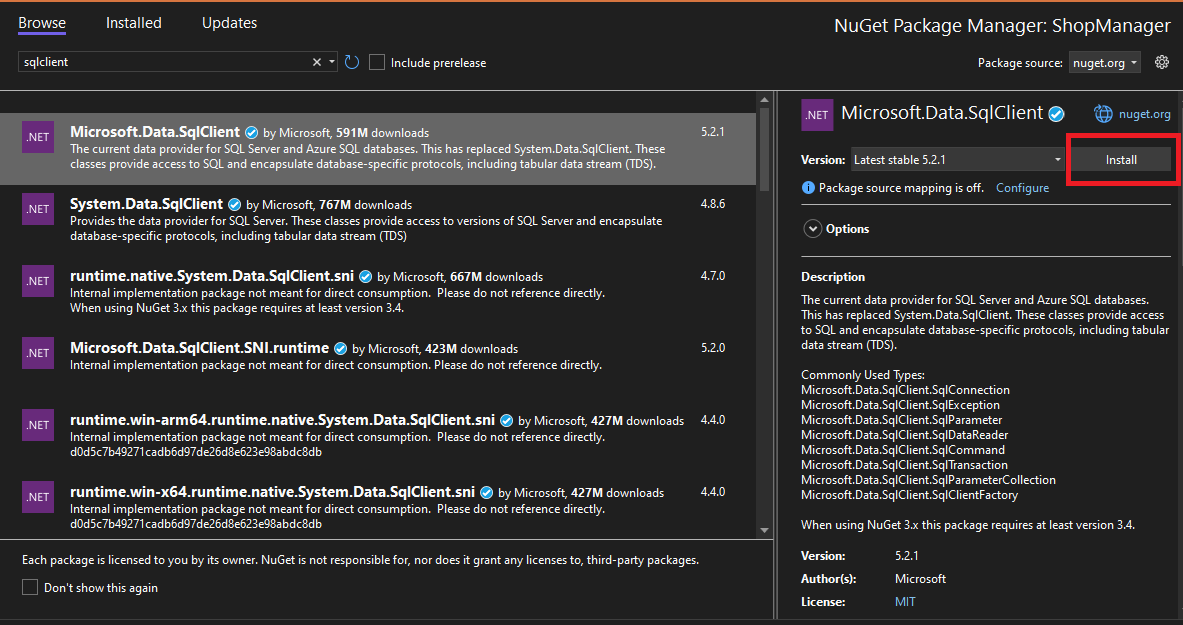


Right Click tại tên Project -> Manage NuGet Packages…



Chọn Brower -> Nhập **sqlclient ->** Chọn Microsoft.Data.SqlClient -> Install -> Apply -> I Accept





Tạo Folder mới: Database

Tạo Class mới: DBConnect.cs -> Thay đổi chuỗi kết nối vừa copy

**Thêm vào cuối chuỗi kết nối:**

;TrustServerCertificate=False;Encrypt=false

using Microsoft.Data.SqlClient;

namespace ShopManager.Database

{

public class DbConnect

{

//to create connection

SqlConnection connect = new SqlConnection("Data Source=ADMIN\\DUCHUY;Initial Catalog=DH\_Shop;Integrated Security=True;TrustServerCertificate=False;Encrypt=false ");

//to get connection

public SqlConnection getConnecttion()

{

return connect;

}

//create a function to Open connection

public void openConnection()

{

if (connect.State == System.Data.ConnectionState.Closed)

{

connect.Open();

}

}

//create a function to Close connection

public void closeConnection()

{

if (connect.State == System.Data.ConnectionState.Open)

{

connect.Close();

}

}

}

}

**Hiển thị danh sách Category từ Database**

Tại Folder Models -> Thêm mới class Category.cs

namespace ShopManager.Models

{

public class Category

{

public int Id { get; set; }

public string Title { get; set; }

public string Content { get; set; }

public DateTime CreateAt { get; set; }

public DateTime UpdateAt { get; set; }

}

}

Tạo mới Folder: DAL -> Tạo mới class: CategoryDAL.cs

DAL/CategoryDAL.cs

using Microsoft.Data.SqlClient;

using ShopManager.Database;

using ShopManager.Models;

using System.Diagnostics.CodeAnalysis;

namespace ShopManager.DAL

{

public class CategoryDAL

{

DBConnect connect = new DBConnect();

public List<Category> getAll()

{

connect.openConnection();

List<Category> list = new List<Category>();

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

command.CommandType = System.Data.CommandType.Text;

string query = @"select \* from category ";

command.CommandText = query;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Category category = new Category()

{

Id = Convert.ToInt32(reader["Id"]),

Title = reader["Title"].ToString() ?? "",

Content = reader["TenHH"].ToString() ?? "",

CreateAt = DateTime.Parse(reader["CreateAt"]?.ToString()),

UpdateAt = DateTime.Parse(reader["UpdateAt"]?.ToString()),

};

list.Add(category);

}

}

connect.closeConnection();

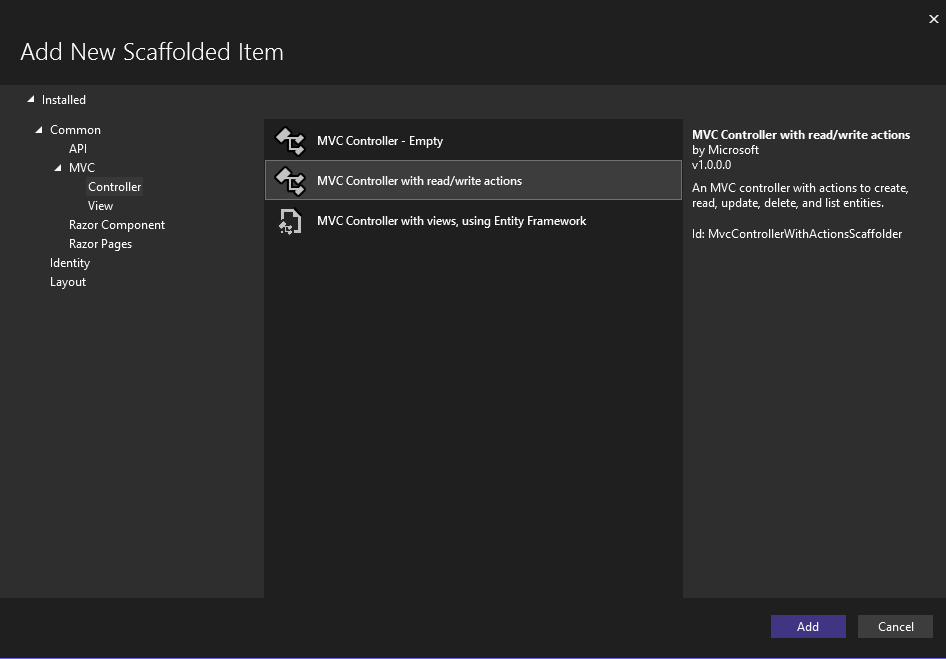
return list;

}

}

}

Tạo Controller mới -> Right Click Folder Controllers -> Add -> Controller -> MVC Controller with read/write actions -> Add -> Nhập tên Controller: CategoryController.cs



Tại Function Index: Lấy danh sách Category từ Database và truyền cho View Index

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class CategoryController : Controller

{

CategoryDAL categoryDAL = new CategoryDAL();

// GET: CategoryController

public ActionResult Index()

{

List <Category> categories = new List<Category>();

categories = categoryDAL.getAll();

return View(categories);

}

// GET: CategoryController/Details/5

public ActionResult Details(int id)

{

return View();

}

// GET: CategoryController/Create

public ActionResult Create()

{

return View();

}

// POST: CategoryController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

// GET: CategoryController/Edit/5

public ActionResult Edit(int id)

{

return View();

}

// POST: CategoryController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

// GET: CategoryController/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: CategoryController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

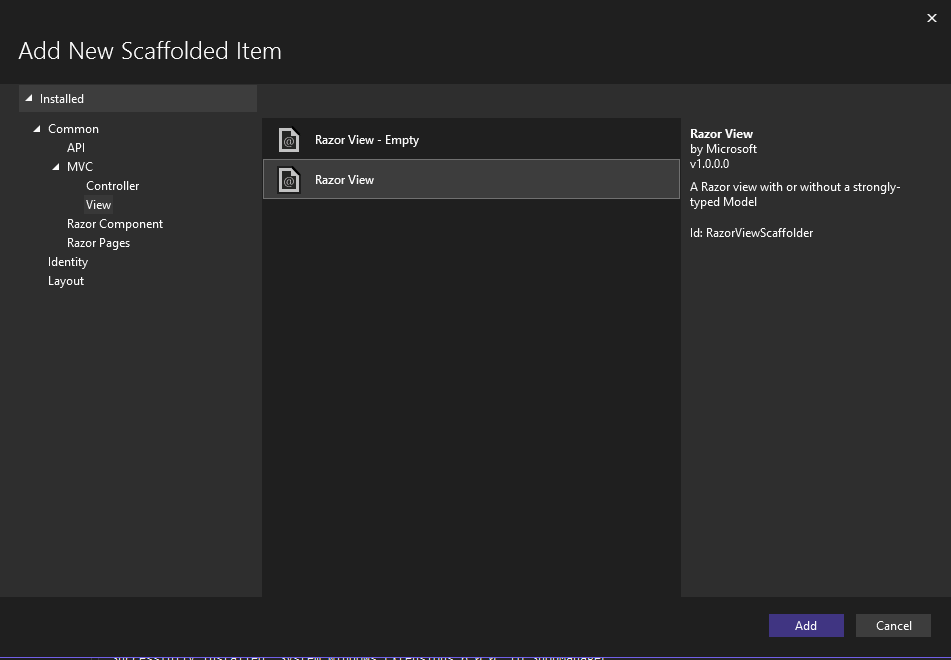
}

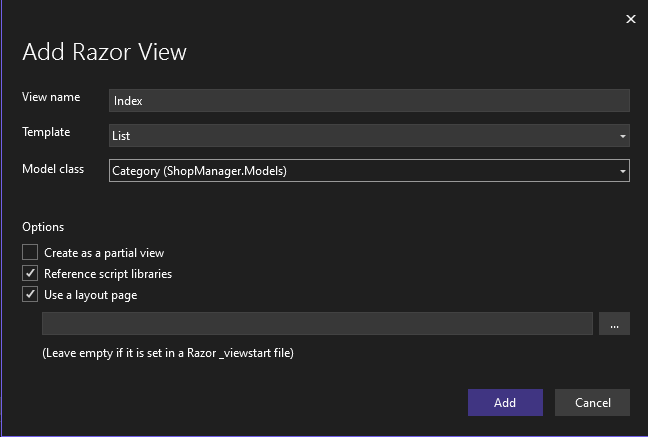
}

}

}

Tạo mới View cho function Index: Right Click tại function Index -> Add View-> Razor View -> Chọn Template: List, Model: Category (ShopManager.Models) -> Add





**Build Project ->**  mở **https://localhost:7018/category/index**

**Thêm Menu Category tại trang chủ**

Views/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"] - ShopManager</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**="~/ShopManager.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">ShopManager</**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>

<li class="nav-item">

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

**Tạo trang thêm mới Category**

DAL/CategoryDAL.cs

using Microsoft.Data.SqlClient;

using Microsoft.Win32;

using ShopManager.Database;

using ShopManager.Models;

using System.Diagnostics.CodeAnalysis;

namespace ShopManager.DAL

{

public class CategoryDAL

{

DBConnect connect = new DBConnect();

public List<Category> getAll()

{

…

}

public bool AddNew(Category categoryAdd)

{

connect.openConnection();

int id = 0;

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

command.CommandType = System.Data.CommandType.Text;

string query = @"insert into category(title, content, createAt, updateAt)

values(@title, @content, @createAt, @updateAt); ";

command.CommandText = query;

command.Parameters.AddWithValue("@title", categoryAdd.Title);

command.Parameters.AddWithValue("@content", categoryAdd.Content);

command.Parameters.AddWithValue("@createAt", categoryAdd.CreateAt);

command.Parameters.AddWithValue("@updateAt", categoryAdd.UpdateAt);

Console.WriteLine("command Insert Category: " + command.CommandText);

id = command.ExecuteNonQuery();

}

connect.closeConnection();

return id > 0;

}

}

}

Controllers/CategoryController.cs

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class CategoryController : Controller

{

CategoryDAL categoryDAL = new CategoryDAL();

// GET: CategoryController

public ActionResult Index()

{

List <Category> categories = new List<Category>();

categories = categoryDAL.getAll();

return View(categories);

}

// GET: CategoryController/Details/5

public ActionResult Details(int id)

{

return View();

}

// GET: CategoryController/Create

public ActionResult Create()

{

return View();

}

// POST: CategoryController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Category categoryNew)

{

try

{

bool IsInserted = false;

if (ModelState.IsValid)

{

DateTime now = DateTime.Now;

categoryNew.CreateAt = now;

categoryNew.UpdateAt = now;

IsInserted = categoryDAL.AddNew(categoryNew);

if (IsInserted)

{

TempData["SuccessMessage"] = "Insert Success";

}

else

{

TempData["ErrorMessage"] = "Insert Fail";

}

}

return RedirectToAction(nameof(Index));

}

catch (Exception ex)

{

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: CategoryController/Edit/5

public ActionResult Edit(int id)

{

return View();

}

// POST: CategoryController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

// GET: CategoryController/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: CategoryController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

}

}

Tạo mới View cho function Create (Không có HttpPost): Right Click tại function Index -> Add View-> Razor View -> Chọn Template: List, Model: Category (ShopManager.Models) -> Add

@model ShopManager.Models.Category

@{

ViewData["Title"] = "Create";

}

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

{

<div class="alert alert-danger">

<strong>Information: </strong> @TempData["ErrorMessage"]

<button type="button" class="btn-close float-end" data-bs-dismiss="alert"

aria-label="Close"></button>

</div>

}

<h1>Create</h1>

<h4>Category</h4>

<hr />

<div class="row">

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

<**form** **asp-action**="Create">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

~~@\* <div class="form-group">~~

~~<label asp-for="Id" class="control-label"></label>~~

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

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

</div> \*@

<div class="form-group">

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

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

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

</div>

<div class="form-group">

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

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

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

</div>

~~@\* <div class="form-group">~~

~~<label asp-for="CreateAt" class="control-label"></label>~~

~~<input asp-for="CreateAt" type="datetime-local" class="form-control" />~~

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

~~</div>~~

~~<div class="form-group">~~

~~<label asp-for="UpdateAt" class="control-label"></label>~~

~~<input asp-for="UpdateAt" type="datetime-local" class="form-control" />~~

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

~~</div> \*@~~

<br />

<div class="form-group">

<input type="submit" value="Create" class="btn btn-primary" />

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Back to List</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

* **Trong category, CreateAt và UpdateAt tự động lấy thời gian tự hệ thống, không cần nhập liệu**

**Tạo trang Detail cho Category**

Tạo Query lấy Category theo Id

DAL/CategoryDAL.cs

using Microsoft.AspNetCore.Http.HttpResults;

using Microsoft.Data.SqlClient;

using Microsoft.Win32;

using ShopManager.Database;

using ShopManager.Models;

using System.Diagnostics.CodeAnalysis;

namespace ShopManager.DAL

{

public class CategoryDAL

{

DBConnect connect = new DBConnect();

public List<Category> getAll()

{

…

}

public bool AddNew(Category categoryAdd)

{

…

}

public Category getCategoryById(int id)

{

connect.openConnection();

Category category = new Category();

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

command.CommandType = System.Data.CommandType.Text;

string query = @"select \* from category where Id = " + id;

command.CommandText = query;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

category.Id = id;

category.Title = reader["Title"].ToString() ?? "";

category.Content = reader["Content"].ToString() ?? "";

category.CreateAt = DateTime.Parse(reader["CreateAt"]?.ToString());

category.UpdateAt = DateTime.Parse(reader["UpdateAt"]?.ToString());

}

}

connect.closeConnection();

return category;

}

}

}

Controllers/CategoryController.cs

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class CategoryController : Controller

{

CategoryDAL categoryDAL = new CategoryDAL();

// GET: CategoryController

public ActionResult Index()

{

List <Category> categories = new List<Category>();

categories = categoryDAL.getAll();

return View(categories);

}

// GET: CategoryController/Details/5

public ActionResult Details(int id)

{

Category category = new Category();

category = categoryDAL.getCategoryById(id);

return View(category);

}

// GET: CategoryController/Create

public ActionResult Create()

{

return View();

}

// POST: CategoryController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Category categoryNew)

{

…

}

// GET: CategoryController/Edit/5

public ActionResult Edit(int id)

{

return View();

}

// POST: CategoryController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

// GET: CategoryController/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: CategoryController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

}

}

Tạo mới View cho function Detail: Right Click tại function **Detail** -> **Add View**-> **Razor View** -> Chọn Template: **Detail**, Model: **Category (ShopManager.Models)** -> Add

@model ShopManager.Models.Category

@{

ViewData["Title"] = "Details";

}

<h1>Details</h1>

<div>

<h4>Category</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Id)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Title)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Title)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Content)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Content)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.CreateAt)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.CreateAt)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.UpdateAt)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.UpdateAt)

</dd>

</dl>

</div>

<div>

@\* ~~@Html.ActionLink("Edit", "Edit", new { id = Model.Id }) | \*@~~

@Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |

<**a** **asp-action**="Index">Back to List</**a**>

</div>

Tại trang Views/Category/Index.cshtml -> truyền Id Category được chọn khi nhấn button Detail

@model IEnumerable<ShopManager.Models.Category>

@{

ViewData["Title"] = "Index";

}

<h1>Index</h1>

<p>

<**a** **asp-action**="Create">Create New</**a**>

</p>

<table class="table">

<thead>

<tr>

<th>

@Html.DisplayNameFor(model => model.Id)

</th>

<th>

@Html.DisplayNameFor(model => model.Title)

</th>

<th>

@Html.DisplayNameFor(model => model.Content)

</th>

<th>

@Html.DisplayNameFor(model => model.CreateAt)

</th>

<th>

@Html.DisplayNameFor(model => model.UpdateAt)

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Id)

</td>

<td>

@Html.DisplayFor(modelItem => item.Title)

</td>

<td>

@Html.DisplayFor(modelItem => item.Content)

</td>

<td>

@Html.DisplayFor(modelItem => item.CreateAt)

</td>

<td>

@Html.DisplayFor(modelItem => item.UpdateAt)

</td>

<td>

~~@\* @Html.ActionLink("Edit", "Edit", new { /\* id=item.PrimaryKey \*/ }) | \*@~~

@Html.ActionLink("Edit", "Edit", new { id = item.Id }) |

~~@\* @Html.ActionLink("Details", "Details", new { /\* id=item.PrimaryKey \*/ }) | \*@~~

@Html.ActionLink("Details", "Details", new { id=item.Id }) |

~~@\* @Html.ActionLink("Delete", "Delete", new { /\* id=item.PrimaryKey \*/ }) \*@~~

@Html.ActionLink("Delete", "Delete", new { id=item.Id })

</td>

</tr>

}

</tbody>

</table>

Build Project

**Tạo trang Edit cho Category**

Views/Category/Index.cshtml

@model IEnumerable<ShopManager.Models.Category>

@{

ViewData["Title"] = "Index";

}

<h1>Index</h1>

<p>

<**a** **asp-action**="Create">Create New</**a**>

</p>

<table class="table">

<thead>

<tr>

<th>

@Html.DisplayNameFor(model => model.Id)

</th>

<th>

@Html.DisplayNameFor(model => model.Title)

</th>

<th>

@Html.DisplayNameFor(model => model.Content)

</th>

<th>

@Html.DisplayNameFor(model => model.CreateAt)

</th>

<th>

@Html.DisplayNameFor(model => model.UpdateAt)

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Id)

</td>

<td>

@Html.DisplayFor(modelItem => item.Title)

</td>

<td>

@Html.DisplayFor(modelItem => item.Content)

</td>

<td>

@Html.DisplayFor(modelItem => item.CreateAt)

</td>

<td>

@Html.DisplayFor(modelItem => item.UpdateAt)

</td>

<td>

~~@\* @Html.ActionLink("Edit", "Edit", new { /\* id=item.PrimaryKey \*/ }) | \*@~~

@Html.ActionLink("Edit", "Edit", new { id = item.Id }) |

@Html.ActionLink("Details", "Details", new { /\* id=item.PrimaryKey \*/ }) |

@Html.ActionLink("Delete", "Delete", new { /\* id=item.PrimaryKey \*/ })

</td>

</tr>

}

</tbody>

</table>

Truyền Id của Category được chọn vào trang Edit

Sau đó tạo truy vấn SQL từ Id Category vừa nhận được

DAL/CategoryDAL.cs

using Microsoft.AspNetCore.Http.HttpResults;

using Microsoft.Data.SqlClient;

using Microsoft.Win32;

using ShopManager.Database;

using ShopManager.Models;

using System.Diagnostics.CodeAnalysis;

namespace ShopManager.DAL

{

public class CategoryDAL

{

DBConnect connect = new DBConnect();

public List<Category> getAll()

{

…

}

public bool AddNew(Category categoryAdd)

{

…

}

public Category getCategoryById(int id)

{

…

}

public bool updateCategoryById(int id, Category categoryUpdate)

{

connect.openConnection();

int isSuccess = 0;

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

command.CommandType = System.Data.CommandType.Text;

string query = @"update category

set title = @title, content = @content, updateAt = @updateAt

where Id = @id";

command.CommandText = query;

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

command.Parameters.AddWithValue("@title", categoryUpdate.Title);

command.Parameters.AddWithValue("@content", categoryUpdate.Content);

command.Parameters.AddWithValue("@updateAt", categoryUpdate.UpdateAt);

Console.WriteLine("command update Category: " + command.CommandText);

isSuccess = command.ExecuteNonQuery();

}

connect.closeConnection();

return isSuccess > 0;

}

}

}

Controllers/CategoryController.cs

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class CategoryController : Controller

{

CategoryDAL categoryDAL = new CategoryDAL();

// GET: CategoryController

public ActionResult Index()

{

…

}

// GET: CategoryController/Details/5

public ActionResult Details(int id)

{

…

}

// GET: CategoryController/Create

public ActionResult Create()

{

return View();

}

// POST: CategoryController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Category categoryNew)

{

…

}

// GET: CategoryController/Edit/5

public ActionResult Edit(int id)

{

Category category = new Category();

category = categoryDAL.getCategoryById(id);

return View(category);

}

// POST: CategoryController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, Category categoryNew)

{

try

{

bool IsInserted = false;

if (ModelState.IsValid)

{

Console.WriteLine("Update Category Id = ", id);

DateTime now = DateTime.Now;

categoryNew.UpdateAt = now;

IsInserted = categoryDAL.updateCategoryById(id,categoryNew);

if (IsInserted)

{

Console.WriteLine("Update Success");

TempData["SuccessMessage"] = "Insert Success";

}

else

{

Console.WriteLine("Update Fail");

TempData["ErrorMessage"] = "Insert Fail";

}

}

return RedirectToAction(nameof(Index));

}

catch (Exception ex)

{

Console.WriteLine("Update error "+ex.Message);

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

// GET: CategoryController/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: CategoryController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

}

}

Views/Category/Edit.cshtml

@model ShopManager.Models.Category

@{

ViewData["Title"] = "Edit";

}

<h1>Edit</h1>

<h4>Category</h4>

<hr />

<div class="row">

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

<**form** **asp-action**="Edit">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

~~@\* <div class="form-group">~~

~~<label asp-for="Id" class="control-label"></label>~~

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

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

~~</div> \*@~~

<div class="form-group">

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

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

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

</div>

<div class="form-group">

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

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

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

</div>

~~@\* <div class="form-group">~~

~~<label asp-for="CreateAt" class="control-label"></label>~~

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

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

~~</div> \*@~~

~~@\* <div class="form-group">~~

~~<label asp-for="UpdateAt" class="control-label"></label>~~

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

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

~~</div> \*@~~

<br />

<div class="form-group">

<input type="submit" value="Save" class="btn btn-primary" />

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Back to List</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

**Tạo trang Delete Category**

Tạo truy vấn Delete Category

DAL/CategoryDAL.cs

using Microsoft.AspNetCore.Http.HttpResults;

using Microsoft.Data.SqlClient;

using Microsoft.Win32;

using ShopManager.Database;

using ShopManager.Models;

using System.Diagnostics.CodeAnalysis;

namespace ShopManager.DAL

{

public class CategoryDAL

{

DBConnect connect = new DBConnect();

public List<Category> getAll()

{

…

}

public bool AddNew(Category categoryAdd)

{

…

}

public Category getCategoryById(int id)

{

…

}

public bool updateCategoryById(int id, Category categoryUpdate)

{

…

}

public bool deleteCategoryById(int id)

{

connect.openConnection();

int isSuccess = 0;

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

command.CommandType = System.Data.CommandType.Text;

string query = @"DELETE FROM category WHERE id = @id;";

command.CommandText = query;

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

Console.WriteLine("command delete Category: " + command.CommandText);

isSuccess = command.ExecuteNonQuery();

}

connect.closeConnection();

return isSuccess > 0;

}

}

}

Controllers/CategoryController.cs

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class CategoryController : Controller

{

CategoryDAL categoryDAL = new CategoryDAL();

// GET: CategoryController

public ActionResult Index()

{

…

}

// GET: CategoryController/Details/5

public ActionResult Details(int id)

{

…

}

// GET: CategoryController/Create

public ActionResult Create()

{

return View();

}

// POST: CategoryController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Category categoryNew)

{

…

}

// GET: CategoryController/Edit/5

public ActionResult Edit(int id)

{

…

}

// POST: CategoryController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, Category categoryNew)

{

…

}

// GET: CategoryController/Delete/5

public ActionResult Delete(int id)

{

Category category = new Category();

category = categoryDAL.getCategoryById(id);

return View(category);

}

// POST: CategoryController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

bool IsInserted = false;

if (ModelState.IsValid)

{

Console.WriteLine("Update Category Id = ", id);

IsInserted = categoryDAL.deleteCategoryById(id);

if (IsInserted)

{

Console.WriteLine("Delete Success");

TempData["SuccessMessage"] = "Delete Success";

}

else

{

Console.WriteLine("Delete Fail");

TempData["ErrorMessage"] = "Delete Fail";

}

}

return RedirectToAction(nameof(Index));

}

catch (Exception ex)

{

Console.WriteLine("Delete error " + ex.Message);

TempData["ErrorMessage"] = ex.Message;

return View();

}

}

}

}

Tạo mới View cho function Delete (Không có HttpPost): Right Click tại function **Delete** -> **Add View**-> **Razor View** -> Chọn Template: **Delete**, Model: **Category (ShopManager.Models)** -> Add

@model ShopManager.Models.Category

@{

ViewData["Title"] = "Delete";

}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>

<div>

<h4>Category</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Id)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Title)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Title)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Content)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Content)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.CreateAt)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.CreateAt)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.UpdateAt)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.UpdateAt)

</dd>

</dl>

<**form** **asp-action**="Delete">

<input type="submit" value="Delete" class="btn btn-danger" /> |

<**a** **asp-action**="Index">Back to List</**a**>

</**form**>

</div>

Build Project