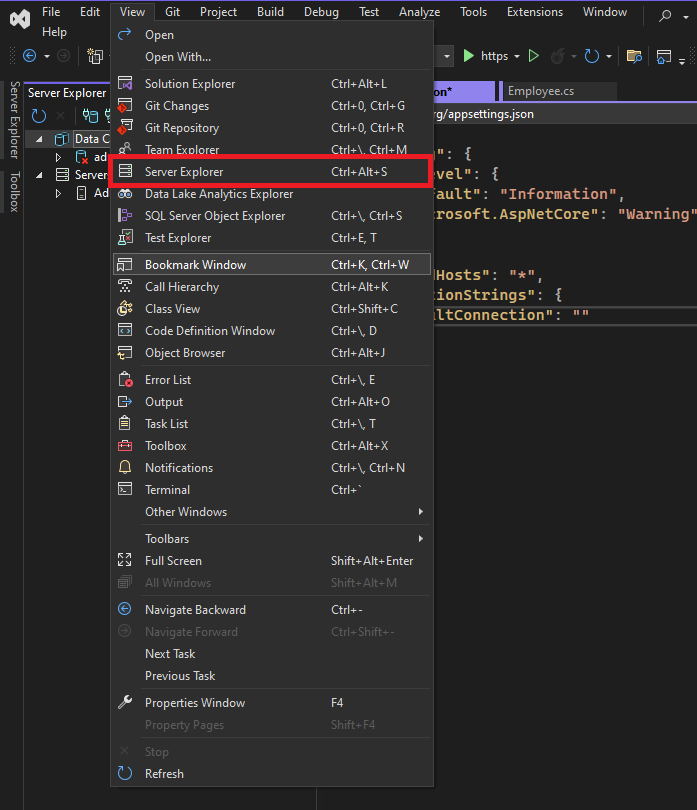
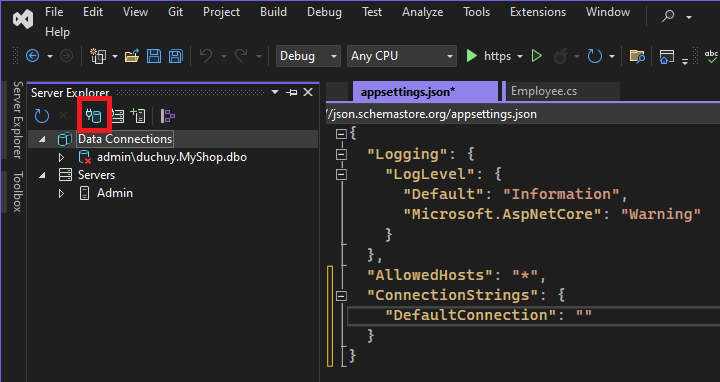
# TẠO trang quản lý kết nối tới database

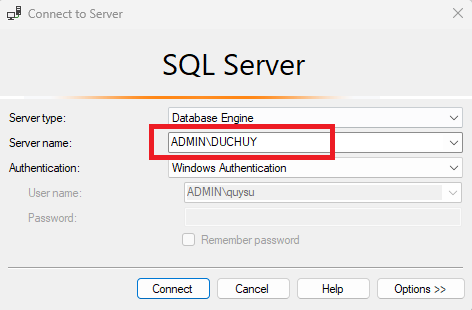
## Tạo chuỗi Kết nối SQL Server

* Xây dựng Database (file **DH\_Shop\_Database.sql**)
* Sau khi tạo thành công Database, Tạo mới **Project (**tên Project: **ShopManager)** -> **View** -> **Server Explorer** -> **Connect to Database**





* Khởi động **SQL Server** để lấy **Server Name**



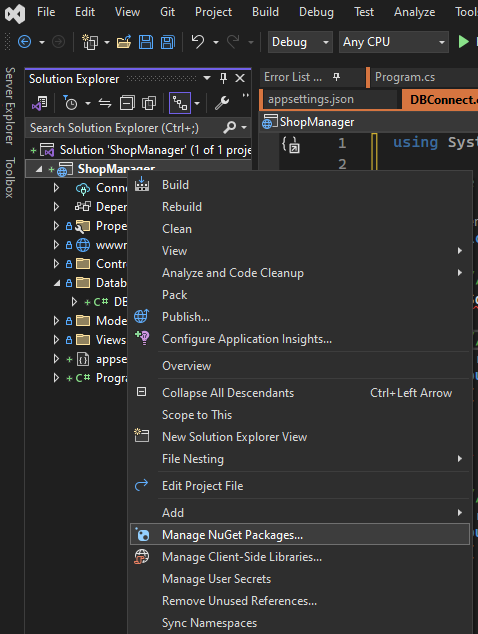
* **Paste** vào ô **server name** và **chọn database** vừa tạo -> **Advanced**

|  |  |
| --- | --- |
| Screenshot 2024-06-18 153847 | C:\Users\quysu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot 2024-06-18 154045.png |

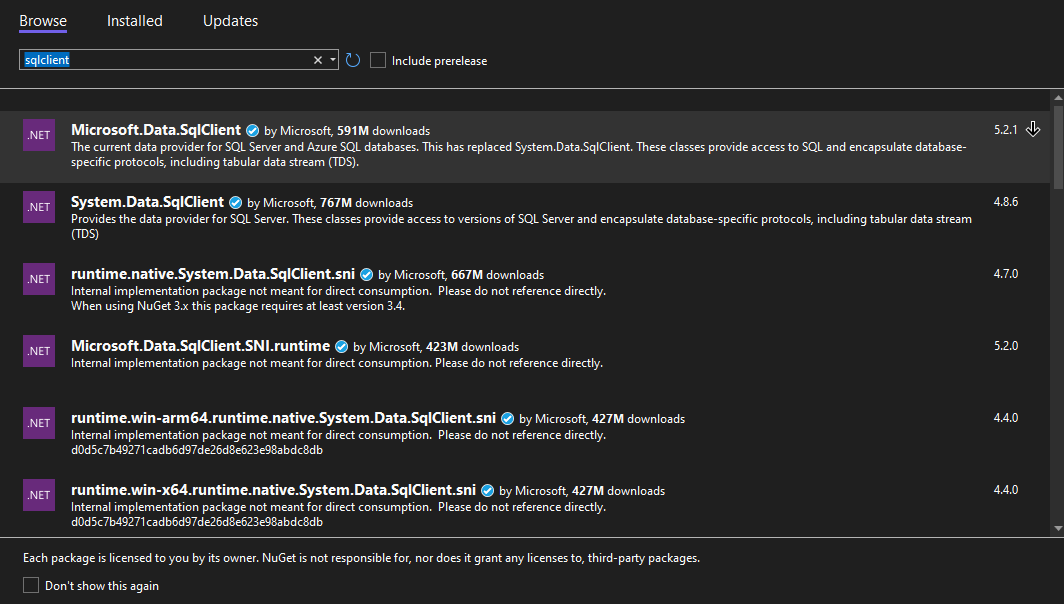
* **Copy** Connection String (**Data Source=…**)

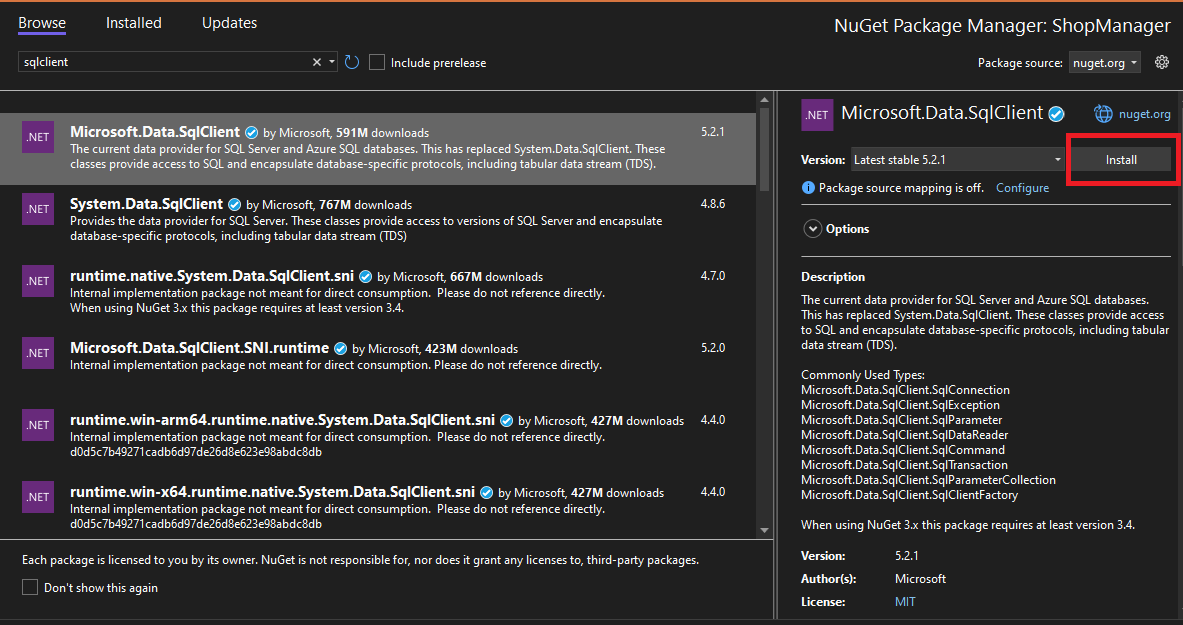
## Cài đặt thư viện SQL Client

* Right Click tại tên Project -> **Manage NuGet Packages**…



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





## Kết nối tới Database bằng thư viện SQL Client

* 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

**Database/DBConnect.cs**

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

**Models/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();

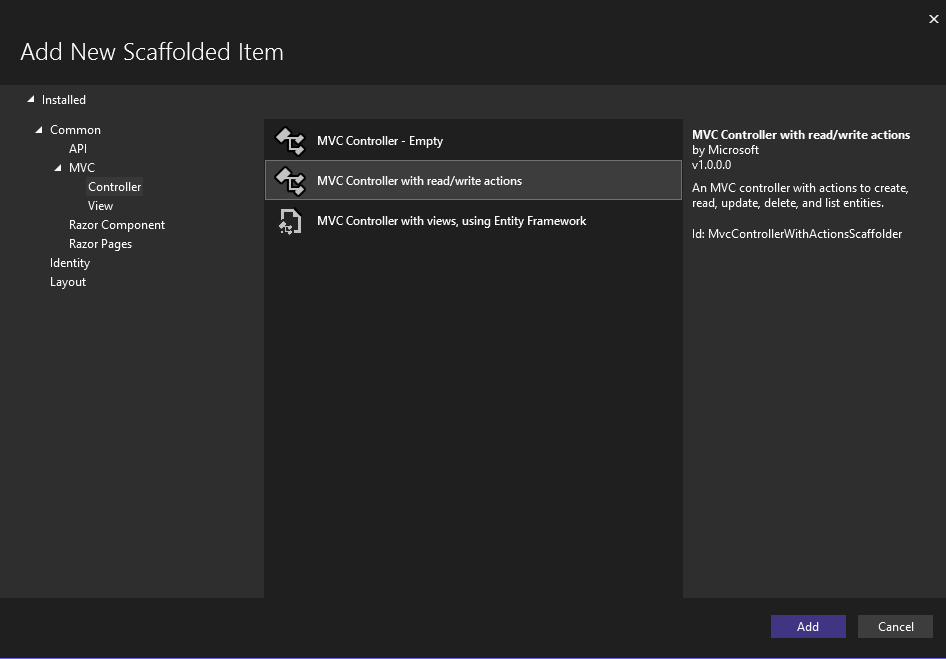
return list;

}

}

}

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: **Create**, 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**
* **Id** được SQL Server tự động sinh **(Auto Increment)**

## 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: **Details**, 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, Edit, Delete

@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

Tạo truy vấn SQL từ **Id Category** vừa nhận được

**DAL/CategoryDAL.cs**

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**

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();

}

}

}

}

Tạo **Razor View Edit** (**Model Category**) cho fucntion **Edit** (**Không có HttpPost**)

**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**

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**

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

## Hiển thị danh sách Product từ Database

Tạo **Model Product.cs**

**Models/Product.cs**

namespace ShopManager.Models

{

public class Product

{

[Key]

public int Id { get; set; }

[Display(Name = "Title")]

[Required(ErrorMessage = "\*")]

[MaxLength(100, ErrorMessage = "Tối đa 100 kí tự")]

public string Title { get; set; }

[Display(Name = "Content")]

[Required(ErrorMessage = "\*")]

[MaxLength(500, ErrorMessage = "Tối đa 500 kí tự")]

public string Content { get; set; }

public string Img { get; set; }

[Display(Name = "Price")]

[Required(ErrorMessage = "\*")]

public int Price { get; set; }

[Display(Name = "Discount")]

[Required(ErrorMessage = "\*")]

public double Discount { get; set; }

[Display(Name = "Create At")]

public DateTime CreateAt { get; set; }

[Display(Name = "Update At")]

public DateTime UpdateAt { get; set; }

[Display(Name = "Category Id")]

[Required(ErrorMessage = "\*")]

public int CategoryId { get; set; }

[Display(Name = "Category Title")]

[Required(ErrorMessage = "\*")]

public string CategoryTitle { get; set; }

}

}

Tạo DAL mới: **ProductDAL.cs**

**DAL/ProductDAL.cs**

using Microsoft.Data.SqlClient;

using ShopManager.Database;

using ShopManager.Models;

namespace ShopManager.DAL

{

public class ProductDAL

{

DBConnect connect = new DBConnect();

public List<Product> getAll()

{

connect.openConnection();

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

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

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

string query = @"select a.id as ProductId, a.title as ProductTitle,

a.content as ProductContent, a.img as ProductImg, a.price as ProductPrice,

a.discount as ProductDiscount, a.createAt as ProductCreateAt, a.updateAt as ProductUpdateAt,

b.id as CategoryId, b.title as CategoryTitle

from product a join category b on a.categoryId = b.Id

";

command.CommandText = query;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

// ================== Cách 1 ===============

Product product = new Product()

{

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

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

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

Img = reader["ProductImg"].ToString() ?? "",

Price = Convert.ToInt32(reader["ProductPrice"]),

Discount = Convert.ToDouble(reader["ProductDiscount"].ToString()),

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

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

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

CategoryTitle = reader["CategoryTitle"].ToString(),

};

// ================= Cách 2 ===============

//Product product = new Product()

//{

// Id = Convert.ToInt32(reader[0]),

// Title = reader[1].ToString() ?? "",

// Content = reader[2].ToString() ?? "",

// Img = reader[3].ToString() ?? "",

// Price = Convert.ToInt32(reader[4]),

// Discount = Convert.ToDouble(reader[5].ToString()),

// CreateAt = DateTime.Parse(reader[6]?.ToString()),

// UpdateAt = DateTime.Parse(reader[7]?.ToString()),

// CategoryId = Convert.ToInt32(reader[8]),

// CategoryTitle = reader[9].ToString(),

//};

list.Add(product);

}

}

connect.closeConnection();

return list;

}

}

}

Tạo Controller mới: ProductController.cs

**ProductController.cs**

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using ShopManager.DAL;

using ShopManager.Models;

namespace ShopManager.Controllers

{

public class ProductController : Controller

{

ProductDAL productDAL = new ProductDAL();

// GET: ProductController

public ActionResult Index()

{

List<Product> products = new List<Product>();

products = productDAL.getAll();

Console.WriteLine("---");

Console.WriteLine(products);

Console.WriteLine("---");

return View(products);

}

…

}

}

Tạo **Razor View Index** cho function Index

**Views/Product/Index.cshtml**

@model IEnumerable<ShopManager.Models.Product>

@{

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 style="width: 400px">

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</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.Img)

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

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

</td>

</tr>

}

</tbody>

</table>

## Tạo trang thêm mới Product

Tạo Folder mới: **Helper**

Tạo File class mới trong thư mục Helper: **ImageHelper.cs** để xử lý Upload Image

**Helper/ImageHelper.cs**

namespace ShopManager.Helper

{

public class ImageHelper

{

public static string UpLoadImage(IFormFile Hinh, string folder)

{

try

{

//Tạo tên file theo thời gian upload -> tránh trùng tên file

var FileName = DateTime.Now.ToString("yyyyMMddHHmmss") + Hinh.FileName;

//Upload Image vào wwwroot, folder Images

var fullPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot","Images", folder, FileName);

using (var myFile = new FileStream(fullPath, FileMode.CreateNew))

{

Hinh.CopyTo(myFile);

}

return FileName;

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

return string.Empty;

}

}

}

}

Tạo mới Folder trong thư mục wwwroot để lưu ảnh Upload

**wwwroot/Images/SanPham**

Tạo Model mới ProductForm

**Models/Product.cs**

using Microsoft.AspNetCore.Mvc.Rendering;

using System.ComponentModel.DataAnnotations;

namespace ShopManager.Models

{

public class Product

{

[Key]

public int Id { get; set; }

[Display(Name = "Title")]

[Required(ErrorMessage = "\*")]

[MaxLength(100, ErrorMessage = "Tối đa 100 kí tự")]

public string Title { get; set; }

[Display(Name = "Content")]

[Required(ErrorMessage = "\*")]

[MaxLength(500, ErrorMessage = "Tối đa 500 kí tự")]

public string Content { get; set; }

public string Img { get; set; }

[Display(Name = "Price")]

[Required(ErrorMessage = "\*")]

public int Price { get; set; }

[Display(Name = "Discount")]

[Required(ErrorMessage = "\*")]

public double Discount { get; set; }

[Display(Name = "Create At")]

public DateTime CreateAt { get; set; }

[Display(Name = "Update At")]

public DateTime UpdateAt { get; set; }

[Display(Name = "Category Id")]

[Required(ErrorMessage = "\*")]

public int CategoryId { get; set; }

[Display(Name = "Category Title")]

[Required(ErrorMessage = "\*")]

public string CategoryTitle { get; set; }

}

public class ProductForm : Product

{

public string IdCategorySelected { get; set; }

public List<SelectListItem>? ListCategory { get; set; }

}

}

**DAL/ProductDAL.cs**

namespace ShopManager.DAL

{

public class ProductDAL

{

DBConnect connect = new DBConnect();

public List<Product> getAll()

{

…

}

public bool AddNew(ProductForm productNew)

{

connect.openConnection();

int id = 0;

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

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

string query = @"insert into product (title, content, img, price, discount, createAt, updateAt, categoryId)

values (@title, @content, @img, @price, @discount, @createAt, @updateAt, @categoryId); ";

command.CommandText = query;

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

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

command.Parameters.AddWithValue("@img", productNew.Img);

command.Parameters.AddWithValue("@price", productNew.Price);

command.Parameters.AddWithValue("@discount", productNew.Discount);

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

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

command.Parameters.AddWithValue("@categoryId", productNew.CategoryId);

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

id = command.ExecuteNonQuery();

}

connect.closeConnection();

return id > 0;

}

}

}

**Controller/ProductController.cs**

namespace ShopManager.Controllers

{

public class ProductController : Controller

{

ProductDAL productDAL = new ProductDAL();

CategoryDAL categoryDAL = new CategoryDAL();

// GET: ProductController

public ActionResult Index()

{

…

}

// GET: ProductController/Create

public ActionResult Create()

{

// Khai báo Model Category

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

// Lấy danh sách Category từ DataBase

categories = categoryDAL.getAll();

// Khai báo Model

ProductForm productAddNew = new ProductForm();

// Tạo danh sách Input Select

productAddNew.ListCategory = new List<SelectListItem>();

foreach (var item in categories)

{

productAddNew.ListCategory.Add(

new SelectListItem

{

Text = item.Title,

Value = item.Id.ToString()

}

);

}

return View(productAddNew);

}

// POST: ProductController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ProductForm productAddNew, IFormFile Img)

{

try

{

//tự động lấy thời gian CreateAt và UpdateAt theo giờ hệ thống

DateTime now = DateTime.Now;

productAddNew.CreateAt = now;

productAddNew.UpdateAt = now;

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(Img, "SanPham");

productAddNew.Img = ImageName;

//lấy Id Category

int IdCategory = Convert.ToInt32(productAddNew.IdCategorySelected);

productAddNew.CategoryId = IdCategory;

// truy vấn tới CSDL

bool IsInserted = productDAL.AddNew(productAddNew);

// Kiểm tra truy vấn SQL thành công hay không?

if (IsInserted)

{

// Truy vấn Thành công

Console.WriteLine("Insert Product Success");

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

return RedirectToAction(nameof(Index));

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Insert Product Fail");

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

return View();

}

}

catch (Exception ex)

{

//Error

Console.WriteLine("Insert Product error ", ex.Message);

return View();

}

}

…

}

}

Thêm **Razor View Create** (**Models: ProductForm**) mới cho function Create (**Không có HttpPost**)

**Views/Product/Create.cshtml**

@model ShopManager.Models.ProductForm

@{

ViewData["Title"] = "Create";

}

<h1>Create</h1>

<h4>Add New Product </h4>

<hr />

<div class="row">

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

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

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

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

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

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

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

~~</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**="Img" class="control-label"></**label**>

<input name="Img" type="file" accept="image/gif, image/jpeg, image/png" class="form-control" />

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

</div>

<div class="form-group">

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

<**input** **asp-for**="Price" **type**="number" class="form-control" />

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

</div>

<div class="form-group">

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

<**input** **asp-for**="Discount" **type**="number" class="form-control" />

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

</div>

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

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

~~<input asp-for="CreateAt" type="datetime" 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> \*@~~

<div class="form-group">

<label class="control-label">Loại Hàng Hóa</label>

<**select** class="form-select"

**asp-items**="@Model.ListCategory" **asp-for**="IdCategorySelected">

</**select**>

</div>

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

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

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

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

~~</div>~~

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

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

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

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

~~</div> \*@~~

<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");}

}

## Tạo trang xem chi tiết Product

Tạo query truy vấn lấy thông tin Product từ CSDL bằng Id Product

**DAL/ProductDAL.cs**

namespace ShopManager.DAL

{

public class ProductDAL

{

DBConnect connect = new DBConnect();

public List<Product> getAll()

{

…

}

public bool AddNew(ProductForm productNew)

{

…

}

public Product GetProductById(int Id)

{

connect.openConnection();

Product product = new Product();

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

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

string query = @"select a.id as ProductId, a.title as ProductTitle,

a.content as ProductContent, a.img as ProductImg, a.price as ProductPrice,

a.discount as ProductDiscount, a.createAt as ProductCreateAt, a.updateAt as ProductUpdateAt,

b.id as CategoryId, b.title as CategoryTitle

from product a join category b on a.categoryId = b.Id

where a.Id = @Id ";

command.CommandText = query;

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

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

product = new Product()

{

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

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

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

Img = reader["ProductImg"].ToString() ?? "",

Price = Convert.ToInt32(reader["ProductPrice"]),

Discount = Convert.ToDouble(reader["ProductDiscount"].ToString()),

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

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

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

CategoryTitle = reader["CategoryTitle"].ToString(),

};

}

}

connect.closeConnection();

return product;

}

}

}

**Controllers/ProductController.cs**

namespace ShopManager.Controllers

{

public class ProductController : Controller

{

ProductDAL productDAL = new ProductDAL();

CategoryDAL categoryDAL = new CategoryDAL();

// GET: ProductController

public ActionResult Index()

{

…

}

// GET: ProductController/Details/5

public ActionResult Details(int id)

{

// Lấy Thông tin Product từ Id

Product product = new Product();

product = productDAL.GetProductById(id);

return View(product);

}

// GET: ProductController/Create

public ActionResult Create()

{

…

}

// POST: ProductController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ProductForm productAddNew, IFormFile Img)

{

…

}

}

}

Tạo **Razor View Detail** (**Models Product**) cho function Details

@model ShopManager.Models.Product

@{

ViewData["Title"] = "Details";

}

<h1>Details</h1>

<div>

<h4>Product</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.Img)

</dt>

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

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

</dd>

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

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

</dt>

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

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

</dd>

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

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

</dt>

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

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

</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>

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

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

</dt>

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

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

</dd>

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

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

</dt>

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

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

</dd>

</dl>

</div>

<div>

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

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

</div>

## Tạo trang Edit Product

Tạo Query truy vấn Update tới CSDL với Id Product

DAL/ProductDAL.cs

namespace ShopManager.DAL

{

public class ProductDAL

{

DBConnect connect = new DBConnect();

public List<Product> getAll()

{

…

}

public bool AddNew(ProductForm productNew)

{

…

}

public Product GetProductById(int Id)

{

…

}

public bool UpdateProduct(ProductForm productNew, int Id)

{

connect.openConnection();

int isSuccess = 0;

using (SqlCommand command = new SqlCommand())

{

command.Connection = connect.getConnecttion();

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

string query = @"update product

set title = @title, content = @content, img = @img, price = @price, discount = @discount, updateAt = @updateAt, categoryId = @categoryId

where id = @id

";

command.CommandText = query;

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

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

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

command.Parameters.AddWithValue("@img", productNew.Img);

command.Parameters.AddWithValue("@price", productNew.Price);

command.Parameters.AddWithValue("@discount", productNew.Discount);

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

command.Parameters.AddWithValue("@categoryId", productNew.CategoryId);

isSuccess = command.ExecuteNonQuery();

}

connect.closeConnection();

return isSuccess > 0;

}

}

}

**Controllers/ProductController.cs**

namespace ShopManager.Controllers

{

public class ProductController : Controller

{

ProductDAL productDAL = new ProductDAL();

CategoryDAL categoryDAL = new CategoryDAL();

// GET: ProductController

public ActionResult Index()

{

…

}

// GET: ProductController/Details/5

public ActionResult Details(int id)

{

…

}

// GET: ProductController/Create

public ActionResult Create()

{

…

}

// POST: ProductController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ProductForm productAddNew, IFormFile Img)

{

…

}

// GET: ProductController/Edit/5

public ActionResult Edit(int id)

{

// Lấy Thông tin Product từ Id

Product product = new Product();

product = productDAL.GetProductById(id);

//Khai báo Model ProductForm

ProductForm productForm = new ProductForm();

productForm.Id = id;

productForm.Title = product.Title;

productForm.Content = product.Content;

productForm.Img = product.Img;

productForm.Price = product.Price;

productForm.CreateAt = product.CreateAt;

productForm.UpdateAt = product.UpdateAt;

productForm.Discount = product.Discount;

productForm.CategoryId = product.CategoryId;

productForm.CategoryTitle = product.CategoryTitle;

//lấy danh sách Category Từ DB

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

categories = categoryDAL.getAll();

// Tạo danh sách Input Select

productForm.ListCategory = new List<SelectListItem>();

foreach (var item in categories)

{

productForm.ListCategory.Add(

new SelectListItem

{

Text = item.Title,

Value = item.Id.ToString()

}

);

}

return View(productForm);

}

// POST: ProductController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, ProductForm productEdit, IFormFile ImageUpload)

{

try

{

//tự động lấy thời gian UpdateAt theo giờ hệ thống

DateTime now = DateTime.Now;

productEdit.UpdateAt = now;

//Nếu có hình ảnh được Upload

if(ImageUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImageUpload, "SanPham");

productEdit.Img = ImageName;

}

//lấy Id Category

int IdCategory = Convert.ToInt32(productEdit.IdCategorySelected);

productEdit.CategoryId = IdCategory;

// truy vấn tới CSDL

bool IsInserted = productDAL.UpdateProduct(productEdit, id);

// Kiểm tra truy vấn SQL thành công hay không?

if (IsInserted)

{

// Truy vấn Thành công

Console.WriteLine("Update Product Success");

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

return RedirectToAction(nameof(Index));

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Product Fail");

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

return View();

}

}

catch

{

return View();

}

}

// GET: ProductController/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: ProductController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

}

}

Tạo Razor View **Edit** (Model **ProductForm**) cho function Edit (Không có HttpPost)

Views/Product/Edit.cshtml

@model ShopManager.Models.ProductForm

@{

ViewData["Title"] = "Edit";

}

<h1>Edit</h1>

<h4>ProductForm</h4>

<hr />

<div class="row">

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

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

<**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" disabled 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 class="control-label">Image</label>

@\* Lấy dữ liệu Img khi người dùng không upload ảnh mới \*@

@\* style="display:none" để hide input đi \*@

<**input** **asp-for**="Img" class="form-control" style="display:none" />

<br>

<**img** src="~/Images/SanPham/@Model.Img" class="img-fluid rounded" alt="Image"

width="300" height="300">

@\*input File->lấy ảnh người dùng upload gửi vào biến ImageUpload \*@

<input class="form-control"

type="file" accept="image/gif, image/jpeg, image/png"

name="ImageUpload"

/>

</div>

<div class="form-group">

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

<**input** **asp-for**="Price" **type**="number" class="form-control" />

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

</div>

<div class="form-group">

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

<**input** **asp-for**="Discount" **type**="number" class="form-control" />

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

</div>

<div class="form-group">

<label class="control-label">Loại Hàng Hóa</label>

<**select** class="form-select"

**asp-items**="@Model.ListCategory" **asp-for**="IdCategorySelected">

</**select**>

</div>

<div class="form-group">

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

<**input** **asp-for**="CreateAt" disabled 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> \*@~~

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

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

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

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

~~</div>~~

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

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

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

~~<span asp-validation-for="CategoryTitle" 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 Xóa Product

DAL/ProductDAL.cs

using Microsoft.Data.SqlClient;

using ShopManager.Database;

using ShopManager.Models;

namespace ShopManager.DAL

{

public class ProductDAL

{

DBConnect connect = new DBConnect();

public List<Product> getAll()

{

…

}

public bool AddNew(ProductForm productNew)

{

…

}

public Product GetProductById(int Id)

{

…

}

public bool UpdateProduct(ProductForm productNew, int Id)

{

…

}

public bool DeleteProduct(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 product

WHERE id = @id ";

command.CommandText = query;

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

isSuccess = command.ExecuteNonQuery();

}

connect.closeConnection();

return isSuccess > 0;

}

}

}

Controllers/ProductController.cs

namespace ShopManager.Controllers

{

public class ProductController : Controller

{

ProductDAL productDAL = new ProductDAL();

CategoryDAL categoryDAL = new CategoryDAL();

// GET: ProductController

public ActionResult Index()

{

…

}

// GET: ProductController/Details/5

public ActionResult Details(int id)

{

…

}

// GET: ProductController/Create

public ActionResult Create()

{

…

}

// POST: ProductController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ProductForm productAddNew, IFormFile Img)

{

…

}

// GET: ProductController/Edit/5

public ActionResult Edit(int id)

{

…

}

// POST: ProductController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, ProductForm productEdit, IFormFile ImageUpload)

{

…

}

// GET: ProductController/Delete/5

public ActionResult Delete(int id)

{

// Lấy Thông tin Product từ Id

Product product = new Product();

product = productDAL.GetProductById(id);

return View();

}

// POST: ProductController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, IFormCollection collection)

{

try

{

//Truy vấn SQL

var IsSuccess = productDAL.DeleteProduct(id);

// Kiểm tra truy vấn SQL thành công hay không?

if (IsSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Delete Product Success");

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

return RedirectToAction(nameof(Index));

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Delete Product Fail");

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

return View();

}

}

catch

{

return View();

}

}

}

}

Tạo Razor View **Delete** (Model **Product**) cho function Edit (Không có HttpPost)

Views/Product/Delete.cshtml

@model ShopManager.Models.Product

@{

ViewData["Title"] = "Delete";

}

<h1>Delete</h1>

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

<div>

<h4>Product</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.Img)

</dt>

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

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

</dd>

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

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

</dt>

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

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

</dd>

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

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

</dt>

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

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

</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>

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

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

</dt>

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

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

</dd>

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

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

</dt>

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

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

</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>