# Bài thực hành 11: xây dựng trang quản trị cho MENUS

# Menu EDIT

Khi click chuyển đổi trạng thái trên nhóm menu chức năng,

Về giao diện trang Edit cũng giống như trang Create (chỉ Update một số trường thông tin).

## MENU CONTROLLER

/////////////////////////////////////////////////////////////////////////////////////

// Admin/Menus/Edit: Thay doi mot mau tin

public ActionResult Edit(int? id)

{

ViewBag.MenuList = new SelectList(menusDAO.getList("Index"), "Id", "Name");

ViewBag.OrderList = new SelectList(menusDAO.getList("Index"), "Order", "Name");

Menus menu = menusDAO.getRow(id);

return View("Edit", menu);

}

### Post HTTP

Bỏ qua 3 trường: CreateAt/By, TypeMenu, TableId

/////////////////////////////////////////////////////////////////////////////////////

// Admin/Menu/Edit: Thay doi mot mau tin

public ActionResult Edit(int? id)

{

ViewBag.ParentList = new SelectList(menusDAO.getList("Index"), "Id", "Name");

ViewBag.OrderList = new SelectList(menusDAO.getList("Index"), "Order", "Name");

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Menus menus = menusDAO.getRow(id);

if (menus == null)

{

return HttpNotFound();

}

return View("Edit",menus);

}

// POST HTTP

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(Menus menus)

{

if (ModelState.IsValid)

{

if (menus.ParentId == null)

{

menus.ParentId = 0;

}

if (menus.Order == null)

{

menus.Order = 1;

}

else

{

menus.Order += 1;

}

//Xy ly cho muc UpdateAt

menus.UpdateAt = DateTime.Now;

//Xy ly cho muc UpdateBy

menus.UpdateBy = Convert.ToInt32(Session["UserId"]);

//Thong bao thanh cong

TempData["message"] = new XMessage("success", "Cập nhật thành công");

//Cap nhat du lieu

menusDAO.Update(menus);

return RedirectToAction("Index");

}

ViewBag.ParentList = new SelectList(menusDAO.getList("Index"), "Id", "Name");

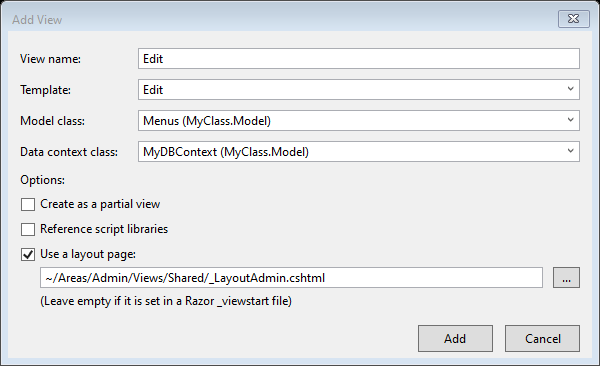
ViewBag.OrderList = new SelectList(menusDAO.getList("Index"), "Order", "Name");

return View(menus);

## }

## Edit.cshmtl

### Tạo mới View từ Controller



#### Lưu ý 2 vấn đề:

### Bỏ qua trường CreateAt/By và menu thêm các Menu con

@using (Html.BeginForm())

{ @Html.AntiForgeryToken()

@Html.HiddenFor(model => model.Id);

@Html.HiddenFor(model => model.CreateAt);

@Html.HiddenFor(model => model.CreateBy);

@Html.HiddenFor(model => model.TypeMenu);

@Html.HiddenFor(model => model.TableId);

…

}

### Tiêu đề trang

@model MyClass.Model.Menus

@{

ViewBag.Title = "Thay đổi thông tin Menu";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

### Nội dung trang

Sao chép từ trang Edit.cshtml của mục Topics

Bỏ qua 3 trường: CreateAt/By, TypeMenu, TableId

@model MyClass.Model.Posts

@{

ViewBag.Title = "Thay đổi thông tin trang đơn";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.Id);

@Html.HiddenFor(model => model.CreateAt);

@Html.HiddenFor(model => model.CreateBy);

@Html.HiddenFor(model => model.TypeMenu);

@Html.HiddenFor(model => model.TableId);

…

}

# Chi tiết

@model MyClass.Model.Menus

@{

ViewBag.Title = "Thay đổi thông tin Menu";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm("Edit", "Menu"))

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.Id);

@Html.HiddenFor(model => model.CreateAt);

@Html.HiddenFor(model => model.CreateBy);

@Html.HiddenFor(model => model.TypeMenu);

@Html.HiddenFor(model => model.TableId);

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Main content -->

<section class="content my-2">

<!-- Default box -->

<div class="card">

<div class="card-header">

<div class="row">

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

<strong class="text-danger">THAY ĐỔI THÔNG TIN MENU</strong>

</div>

<div class="col-md-6 text-right">

<a href="@Url.Action("Index", "Menu")" class="btn-sm btn-info"> <i class="fa fa-long-arrow-alt-left"></i> Quay lại</a>

</div>

</div>

</div>

<div class="card-body">

@\*Phan NOI DUNG\*@

<div class="row">

@\* Phần nhập tay \*@

<div class="col-md-9">

@\* Ngan khong cho edit noi dung cua phan Name va Link \*@

@if (Model.TypeMenu.Equals("custom"))

{

<div class="form-group">

@Html.LabelFor(model => model.Name)

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

</div>

<div class="form-group">

@Html.LabelFor(model => model.Link)

@Html.EditorFor(model => model.Link, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Link, "", new { @class = "text-danger" })

</div>

}

else

{

<div class="form-group">

<label>Tên chủ đề </label>

<div class="col-md-10">

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" , @readonly = "true" } })

@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<label>Liên kết </label>

<div class="col-md-10">

@Html.EditorFor(model => model.Link, new { htmlAttributes = new { @class = "form-control", @readonly = "true" } })

@Html.ValidationMessageFor(model => model.Link, "", new { @class = "text-danger" })

</div>

</div>

}

</div>

@\* Phần lựa chọn \*@

<div class="col-md-3">

<div class="form-group">

<label>Vị trí <span class="text-danger"></span></label>

<div class="col-md-10">

@Html.DropDownListFor(model => model.Position, new[] { new SelectListItem { Text = "Main Menu", Value = "MainMenu" }, new SelectListItem { Text = "Footer", Value = "Footer" } }, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.Position, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<label>Cấp cha <span class="text-danger"></span></label>

<div class="col-md-10">

@Html.DropDownListFor(model => model.ParentId, (SelectList)@ViewBag.ParentList, "Cấp 0", new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.ParentId, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<label>Sắp xếp <span class="text-danger"></span></label>

<div class="col-md-10">

@Html.DropDownListFor(model => model.Order, (SelectList)@ViewBag.OrderList, "Sắp xếp", new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.Order, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<label>Trạng thái <span class="text-danger"> (\*)</span></label>

<div class="col-md-10">

@Html.DropDownListFor(model => model.Status, new[] { new SelectListItem { Text = "Không xuất bản", Value = "2" }, new SelectListItem { Text = "Xuất bản", Value = "1" } }, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.Status, "", new { @class = "text-danger" })

</div>

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<button class="btn-sm btn-success" type="submit" name="create"> Lưu [Thêm] <i class="fa fa-save"></i> </button>

</div>

</div>

</div><!-- /.card-body -->

</div><!-- /.card -->

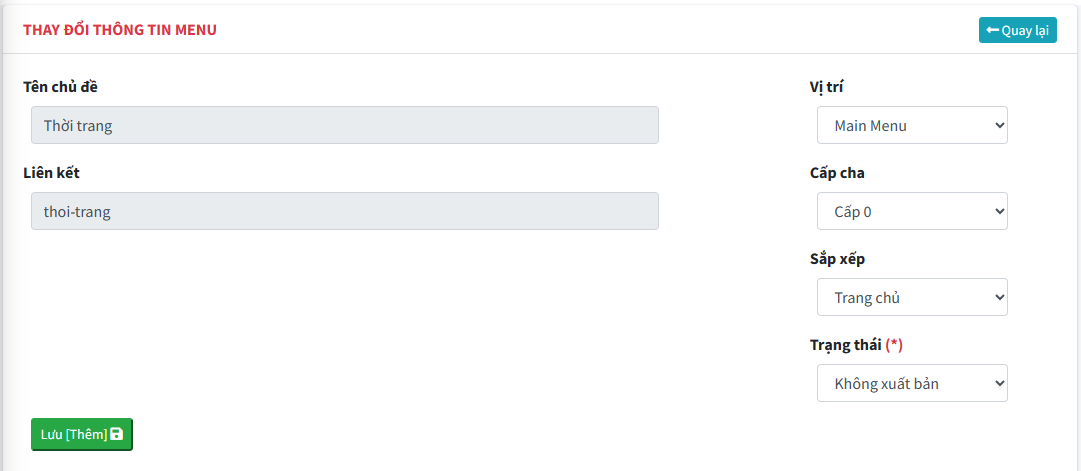
</section><!-- /.content -->

</div><!-- /.content-wrapper -->

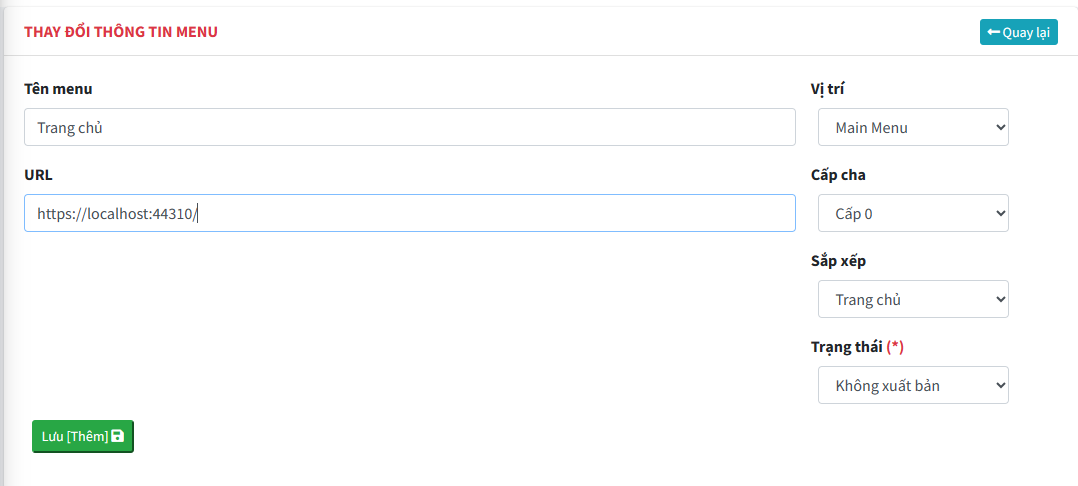
}

# Thử nghiệm

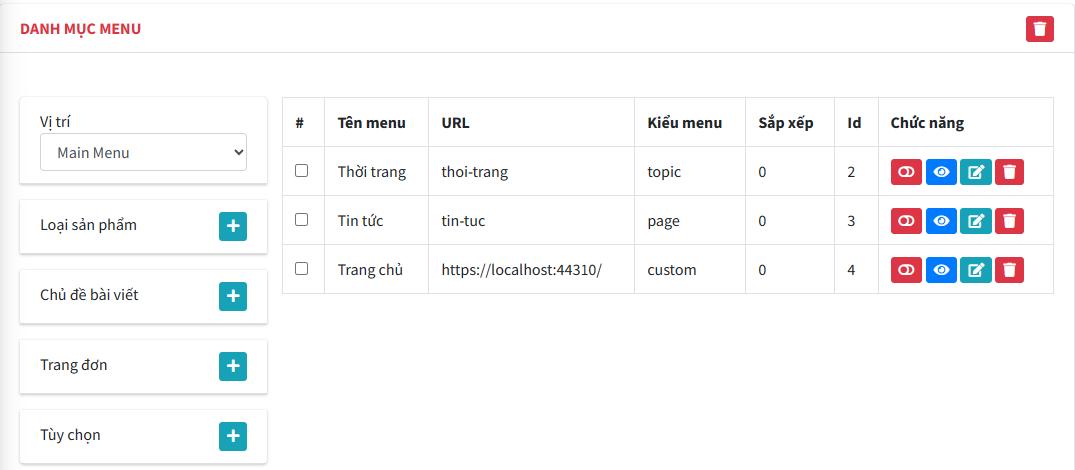
#### Ngăn không cho Edit với những menu không phải là Custom



#### Cho phép Edit với Menu Custom



## Trước edit



## Sau edit

