座號：19 姓名：柯昱廷

階段性作業五

自訂驗證器-合法身分證字號判斷

試寫一我國身分證字號合法性判斷程式，身分證字號的值須能由表單文字方塊給予，將值傳入模型中判斷，並在模型中使用自訂驗證器（CustomValidator），使其能擋下不合法的身分證字號。

* 建立Member模型。
* 建立ID、IDNumber、Name三個屬性(Property)。
* 建立Check IDNumber類別做為自訂驗證器（CustomValidator）。
* 在模型中使用內建與自訂驗證器（CustomValidator）。
* 建立HomeController與Create View測試驗證器是否能正常使用。
* 在View中須顯示該身分證子號是否正確的資訊。

結果畫面截圖(至少三張)：

圖1：

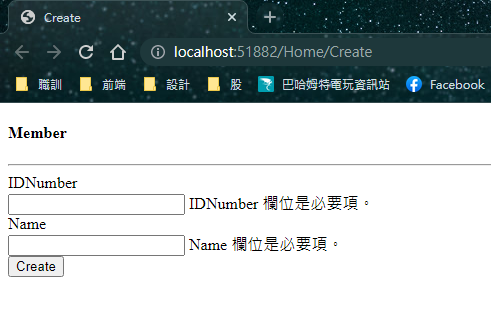


圖2：

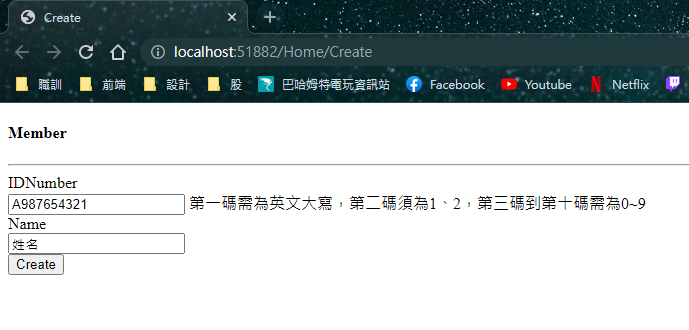
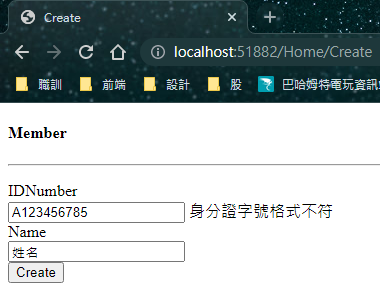


圖3：



程式碼：

Model：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations;

namespace HW5.Models

{

public class Member

{

[Key] //唯一識別值

public int ID { get; set; }

[Required] //必填

[RegularExpression("[A-Z][12][0-9]{8}", ErrorMessage = "第一碼需為英文大寫，第二碼須為1、2，第三碼到第十碼需為0~9")]

[CheckIDNumber]

public string IDNumber { get; set; }

[Required]

[StringLength(50)]

public string Name { get; set; }

public class CheckIDNumber : ValidationAttribute

{

public CheckIDNumber()

{

ErrorMessage = "身分證字號格式不符";

}

public override bool IsValid(object value)

{

//身分證字號傳入為物件所以轉字串

string id = value.ToString(); //A123456789

const string eng = "ABCDEFGHJKLMNPQRSTUVXYWZIO";

string s = id.Substring(0, 1); //s1=A

int intEng = eng.IndexOf(s) + 10; //A=10

int n1 = intEng / 10;

int n2 = intEng % 10;

//抓1~9轉為int放入陣列

int[] ary = new int[9];

for (int i = 0; i < 9; i++)

{

ary[i]= Convert.ToInt32(id.Substring(i+1, 1));

}

int sum = 0;

sum += n1 + (n2 \* 9); //A

for (int j = 0; j < 8; j++) //1~8

{

sum += ary[j] \* (8 - j); //sum += 1 \* 8

}

sum += ary[8]; //9

if (sum % 10 == 0) return true;

return false;

}

}

}

}

Controller：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using HW5.Models;

namespace HW5.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(Member member)

{

//ModelState.IsValid，如果驗證全都通過的話

if (ModelState.IsValid)

{

//把資料存入DB

return RedirectToAction("Index");

}

//錯的話把資料傳回View

return View(member);

}

}

}

View：

@model HW5.Models.Member

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Create</title>

</head>

<body>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Member</h4>

<hr />

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

<div class="form-group">

@Html.LabelFor(model => model.IDNumber, htmlAttributes: new { @class = "control-label col-md-2" })

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

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

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

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

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

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

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

</div>

</div>

<div class="form-group">

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

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

</div>

</div>

</div>

}

<script src="~/Scripts/jquery-3.6.0.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

</body>

</html>

其它(若無可免)：

**繳交作業時，請以FTP上傳程式原始檔與本文件檔各一份，原始程式碼請另放置於「完成結果檔」資料夾內。**

※功能須完全完成始得通過※