

长沙大学

ASP .NET 期末考查



October 16, 2018

周炳强

B20150304117

目录

[ASP .NET 期末考查（采用MVC模式） 2](#_Toc527475766)

[MVC代码结构图 2](#_Toc527475767)

[Model 2](#_Toc527475768)

[View 3](#_Toc527475769)

[Control 3](#_Toc527475770)

[关键代码（只贴出小部分，详细可以通过 4](#_Toc527475771)

[https://github.com/BingqiangZhou/ASP-.NET-MVC查看） 4](#_Toc527475772)

[首页、登录页、注册页与找回密码页 4](#_Toc527475773)

[关键代码 5](#_Toc527475774)

[用户管理页 10](#_Toc527475775)

[关键代码 10](#_Toc527475776)

[商品管理页 14](#_Toc527475777)

[关键代码 14](#_Toc527475778)

[销售订单管理页 19](#_Toc527475779)

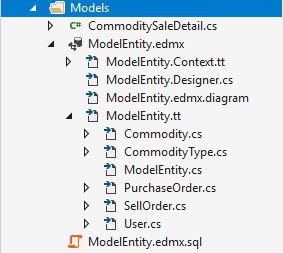
[入库订单管理页 20](#_Toc527475780)

[销量统计页 20](#_Toc527475781)

# ASP .NET 期末考查（采用MVC模式）

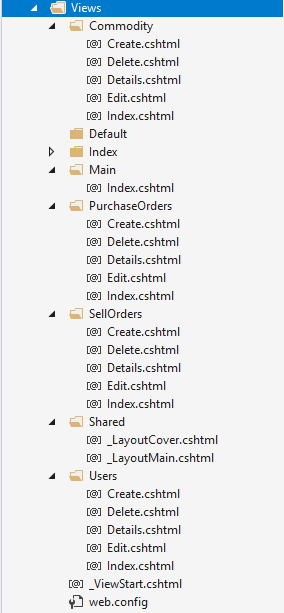
## MVC代码结构图

### Model



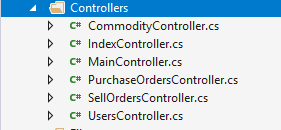
图一、Model结构图

### View



图二、View代码结构图

### Control

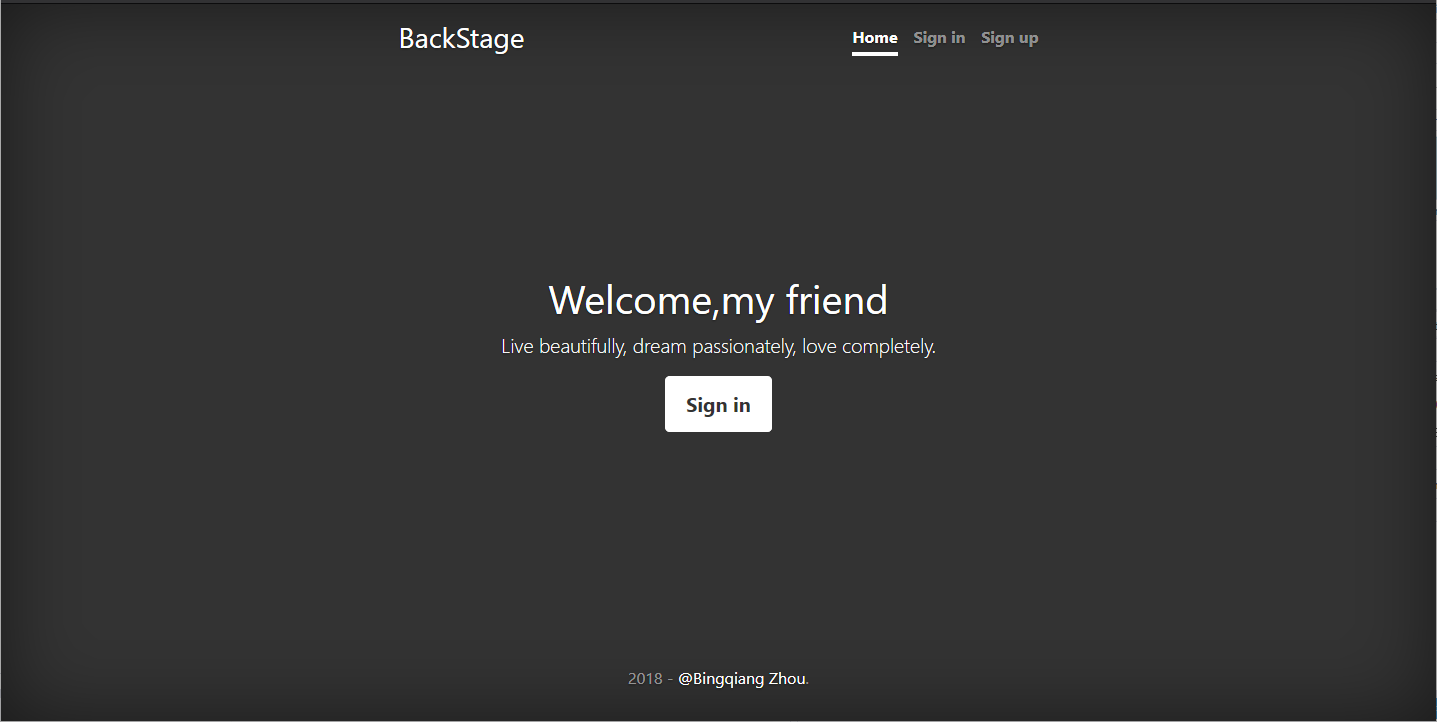


图三、Control代码结构图

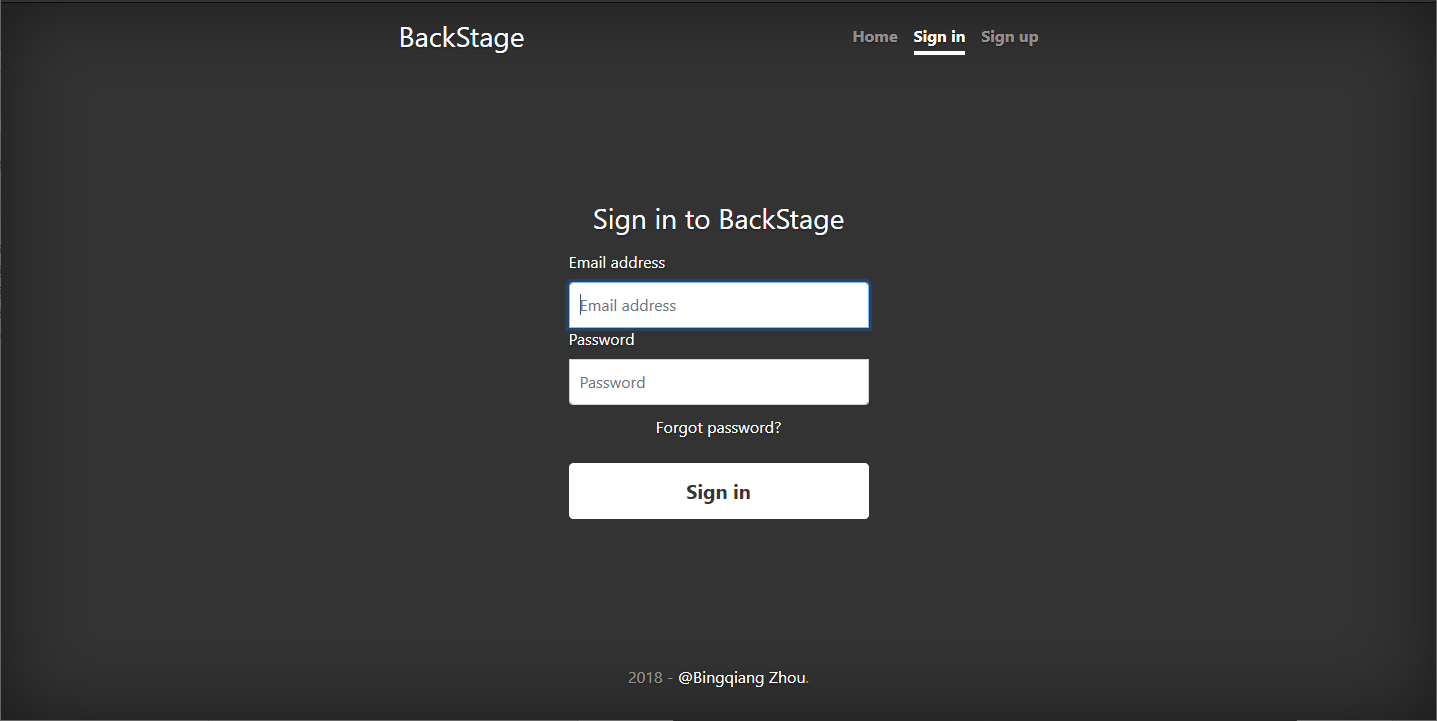
## <https://github.com/BingqiangZhou/ASP-.NET-MVC>

关键代码（只贴出小部分，详细可以通过以上GitHub查看）

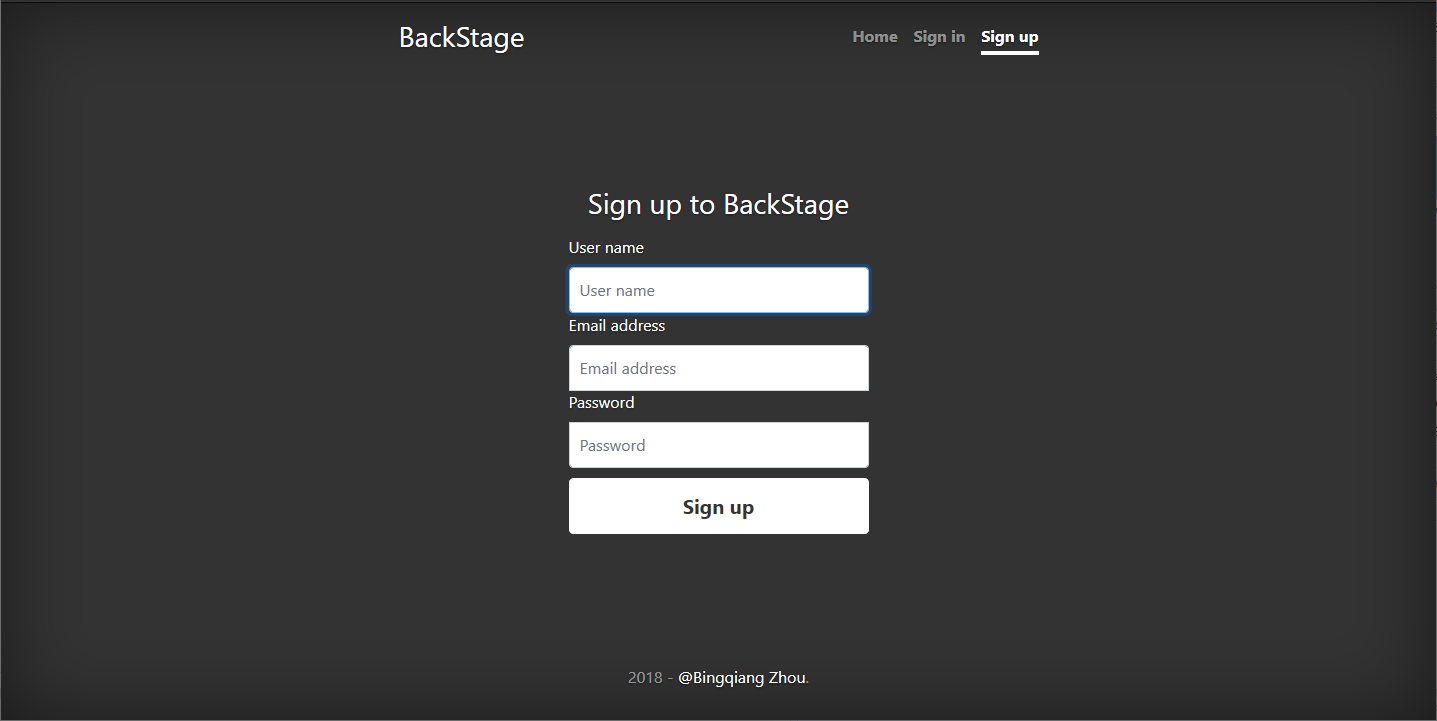
## 首页、登录页、注册页与找回密码页



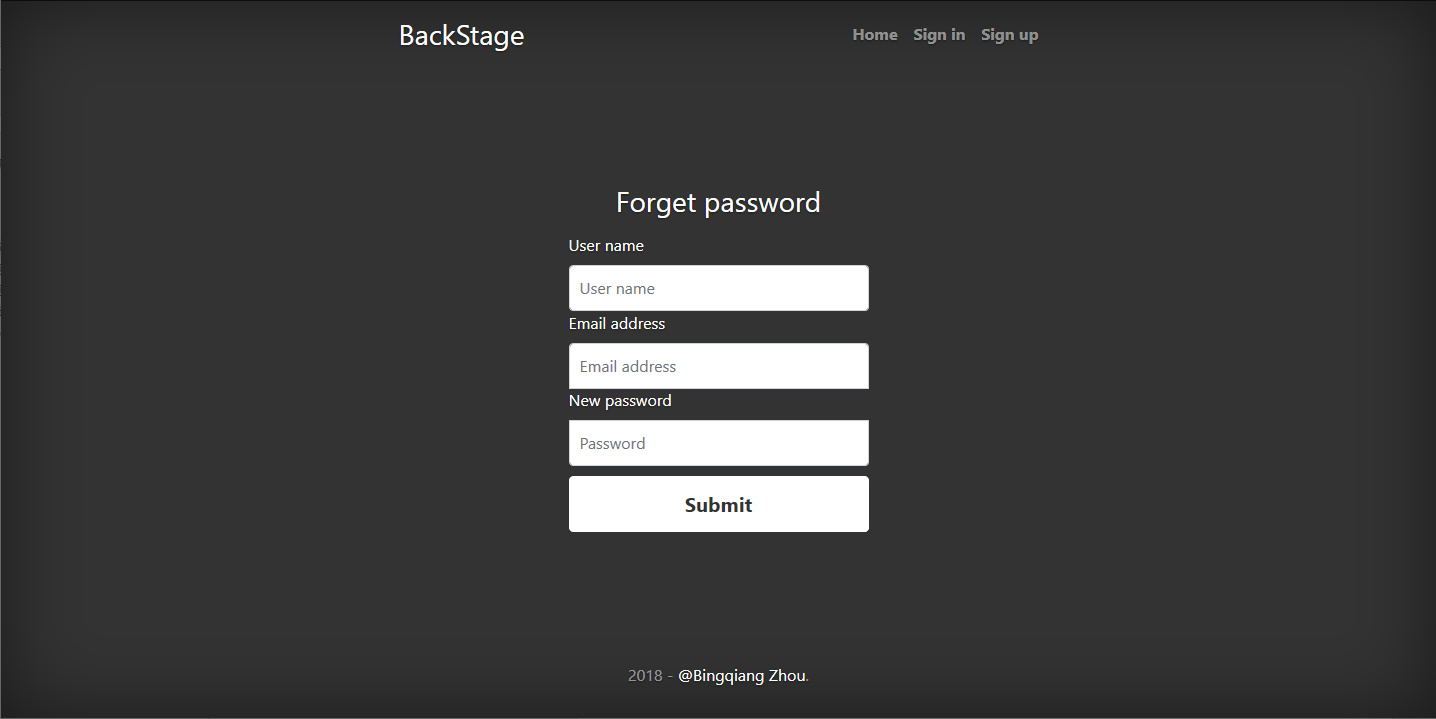
图四、首页效果图



图五、登录页效果图



图六、注册页效果图



图七、找回密码页效果图

### 关键代码

#### Model（User）

namespace Model.Models

{

using System;

using System.Collections.Generic;

public partial class User

{

public string Email { get; set; }

public string Name { get; set; }

public string Password { get; set; }

public byte Status { get; set; }

public byte Rank { get; set; }

}

}

#### View

##### \_LayoutCover.cshtml（母版页）

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

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

<meta name="author" content="">

<title>@ViewBag.Title - @ViewContext.ViewBag.Title</title>

<script src="~/Scripts/jquery-3.0.0.js"></script>

<script type="text/javascript">

$(function () {

var pageId = '#@ViewBag.Title'.replace(" ","\_");

$(".nav-link").removeClass("active");

$(pageId).addClass("active");

});

</script>

<link rel="stylesheet" href="~/Content/bootstrap.css" />

<link href="~/Content/cover.css" rel="stylesheet">

</head>

<body class="text-center">

<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

<header class="masthead mb-auto">

<div class="inner">

<h3 class="masthead-brand">@ViewContext.ViewBag.Title</h3>

<nav class="nav nav-masthead justify-content-center">

<a class="nav-link" href="/Index" id="Home">Home</a>

<a class="nav-link" href="/Index/SignIn" id="Sign\_in">Sign in</a>

<a class="nav-link" href="/Index/SignUp" id="Sign\_up">Sign up</a>

</nav>

</div>

</header>

<main role="main" class="inner cover">

@RenderBody()

</main>

<footer class="mastfoot mt-auto">

<div class="inner">

<p>@DateTime.Now.Year - <a href="mailto:bingqiang\_zhou@qq.com">@@Bingqiang Zhou</a>.</p>

</div>

</footer>

</div>

<script src="~/Scripts/jquery-3.0.0.slim.js"></script>

<script src="~/Scripts/bootstrap.bundle.js"></script>

<script src="~/Scripts/bootstrap.js"></script>

</body>

</html>

##### ForgetPassword.cshtml

@{

ViewBag.Title = "Forget password";

Layout = "../Shared/\_layoutCover.cshtml";

}

<link href="~/Content/signin.css" rel="stylesheet">

<form class="form-signin" action="./ForgetPassword" method="post">

<h1 class="h3 mb-3 font-weight-normal">Forget password</h1>

<label for="Name" class="float-left">User name</label>

<label class="float-right text-danger">@ViewBag.ErrorMessage</label>

<input id="Name" name="Name" class="form-control" placeholder="User name" required=""

oninvalid="setCustomValidity('please enter your name.')" oninput="setCustomValidity('')" autofocus="" type="text">

<label for="Email" class="float-left">Email address</label>

<input id="Email" name="Email" class="form-control" placeholder="Email address" required=""

oninvalid="setCustomValidity('please enter your email.')" oninput="setCustomValidity('')" type="email">

<label for="Password" class="float-left">New password</label>

<input id="Password" name="Password" class="form-control" placeholder="Password" required=""

oninvalid="setCustomValidity('please enter your new password.')" oninput="setCustomValidity('')" type="password">

<button class="btn btn-lg btn-secondary btn-block" type="submit">Submit</button>

</form>

##### Index.cshtml

@{

ViewBag.Title = "Home";

Layout = "../Shared/\_layoutCover.cshtml";

}

<h1 class="cover-heading">Welcome,my friend</h1>

<p class="lead">Live beautifully, dream passionately, love completely.</p>

<p class="lead">

<a href="/Index/SignIn" class="btn btn-lg btn-secondary">Sign in</a>

</p>

##### SignIn.cshtml

@{

ViewBag.Title = "Sign in";

Layout = "../Shared/\_layoutCover.cshtml";

}

<link href="~/Content/signin.css" rel="stylesheet">

<form class="form-signin" action="./SignIn" method="post">

<h1 class="h3 mb-3 font-weight-normal">Sign in to BackStage</h1>

<label for="Email" class="float-left">Email address</label>

<label class="float-right text-danger">@ViewBag.ErrorMessage</label>

<input id="Email" name="Email" class="form-control" placeholder="Email address" required=""

oninvalid="setCustomValidity('please enter your email.')" oninput="setCustomValidity('')" type="email" autofocus>

<label for="Password" class="float-left">Password</label>

<input id="Password" name="Password" class="form-control" placeholder="Password" required=""

oninvalid="setCustomValidity('please enter your new password.')" oninput="setCustomValidity('')" type="password">

<div class="checkbox mb-3">

<label>

<a href="/Index/ForgetPassword">Forgot password?</a>

</label>

</div>

<button class="btn btn-lg btn-secondary btn-block" type="submit">Sign in</button>

</form>

##### SignUp.cshtml

@{

ViewBag.Title = "Sign up";

Layout = "../Shared/\_layoutCover.cshtml";

}

<link href="~/Content/signin.css" rel="stylesheet">

<form class="form-signin" action="./SignUp" method="post">

<h1 class="h3 mb-3 font-weight-normal">Sign up to BackStage</h1>

<label for="Name" class="float-left">User name</label>

<label class="float-right text-danger">@ViewBag.ErrorMessage</label>

<input id="Name" name="Name" class="form-control" placeholder="User name" required autofocus

oninvalid="setCustomValidity('please enter your name.')" oninput="setCustomValidity('')" type="text">

<label for="Email" class="float-left">Email address</label>

<input id="Email" name="Email" class="form-control" placeholder="Email address" required

oninvalid="setCustomValidity('please enter your email.')" oninput="setCustomValidity('')" type="email">

<label for="Password" class="float-left">Password</label>

<input id="Password" name="Password" class="form-control" placeholder="Password" required

oninvalid="setCustomValidity('please enter your password.')" oninput="setCustomValidity('')" type="password">

<button class="btn btn-lg btn-secondary btn-block" type="submit">Sign up</button>

</form>

#### Control

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Model.Models;

namespace Model.Controllers

{

public class IndexController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: Index

public ActionResult Index()

{

return View();

}

public ActionResult SignIn()

{

return View("SignIn");

}

[HttpPost]

public ActionResult SignIn(string Email,string Password)

{

//System.Diagnostics.Debug.WriteLine(Email + Password);

User user = db.Users.Find(Email);

if (user != null && user.Password.Equals(Password))

{

Response.Cookies["Name"].Value = user.Name;

Response.Cookies["Name"].Expires = DateTime.Now.AddHours(1);

Response.Cookies["Email"].Value = user.Email;

Response.Cookies["Email"].Expires = DateTime.Now.AddHours(1);

return RedirectToAction("../Main");

}

ViewBag.ErrorMessage = "Verification failed!";

return View("SignIn");

}

public ActionResult SignUp()

{

return View("SignUp");

}

[HttpPost]

public ActionResult SignUp(string Email,string Name,string Password)

{

User user = db.Users.Find(Email);

if(user != null)

{

ViewBag.ErrorMessage = "User already exist!";

return View("SignUp");

}

user = new User();

user.Email = Email;

user.Name = Name;

user.Password = Password;

user.Rank = 1;

user.Status = 1;

db.Users.Add(user);

db.SaveChanges();

return View("Index");

}

public ActionResult ForgetPassword()

{

return View("ForgetPassword");

}

[HttpPost]

public ActionResult ForgetPassword(string Email, string Name, string Password)

{

User user = db.Users.Find(Email);

if (user == null)

{

ViewBag.ErrorMessage = "User is not exist!";

return View("ForgetPassword");

}

if(! Name.Equals(user.Name))

{

ViewBag.ErrorMessage = "Verification failed!";

return View("ForgetPassword");

}

user.Password = Password;

db.SaveChanges();

return View("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

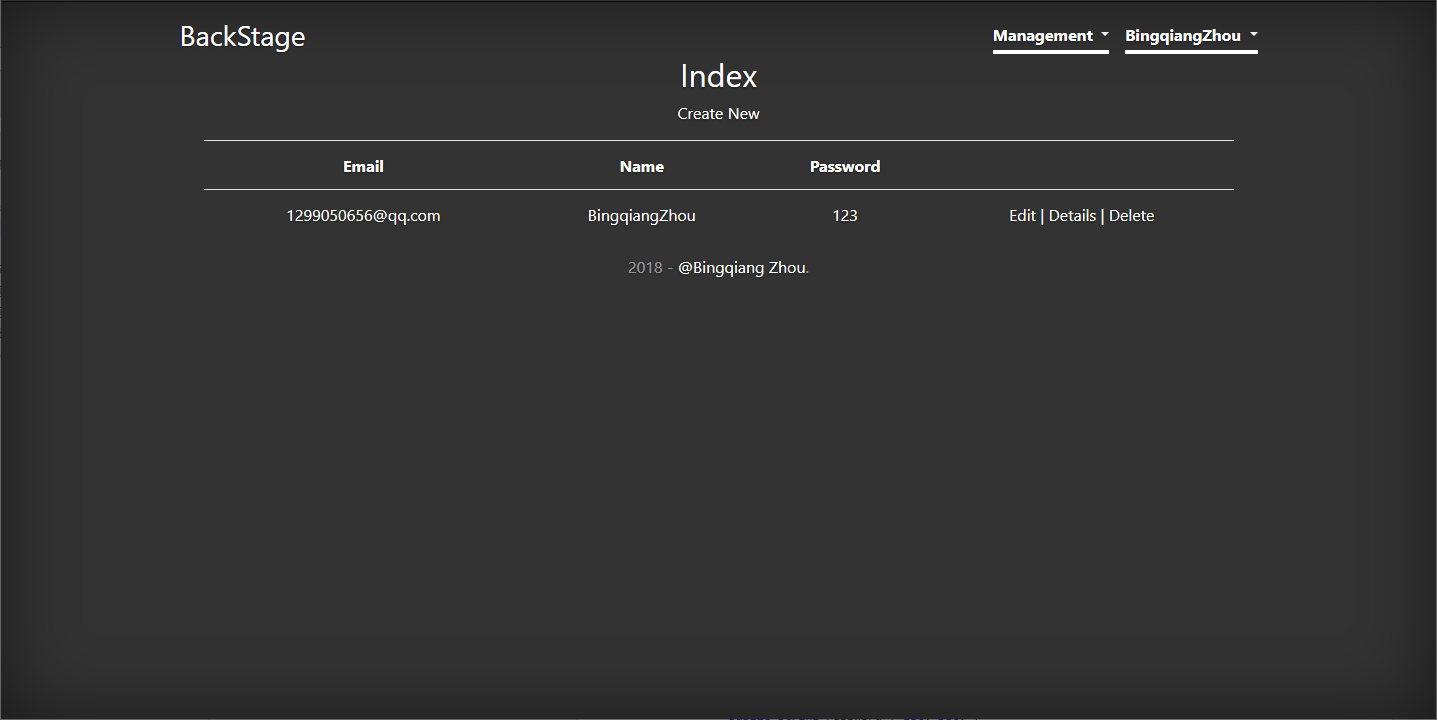
base.Dispose(disposing);

}

}

}

## 用户管理页



图八、用户管理显示信息页面效果

### 关键代码

#### Model（User）与上一部分一致

#### View（省略增删改部分）

@model IEnumerable<Model.Models.Commodity>

@{

ViewBag.Title = "Index";

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

}

<p>

@Html.ActionLink("Create New", "Create")

</p>

<form>

<nav class="navbar navbar-expand-lg bg-transparent">

<div class="btn-group" role="group" aria-label="Button group with nested dropdown">

<button type="button" class="btn btn-secondary font-weight-bold" id="Sort">Sort</button>

<div class="btn-group" role="group">

<button id="btnGroupDrop1" type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">ASC</button>

<div class="dropdown-menu" aria-labelledby="btnGroupDrop1">

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

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

</div>

</div>

</div>

<div class="navbar-nav">

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

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

</div>

</nav>

</form>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

@\*<th>

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

</th>\*@

<th>

Inventory

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

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

</td>

</tr>

}

</table>

#### Control

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Text;

using System.Web;

using System.Web.Mvc;

using Model.Models;

namespace Model.Controllers

{

public class UsersController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: Users

public ActionResult Index()

{

return View(db.Users.ToList());

}

// GET: Users/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

User user = db.Users.Find(id);

if (user == null)

{

return HttpNotFound();

}

return View(user);

}

// GET: Users/Create

public ActionResult Create()

{

return View();

}

// POST: Users/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "Email,Name,Password,Status,Rank")] User user)

{

if (ModelState.IsValid)

{

db.Users.Add(user);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(user);

}

// GET: Users/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

User user = db.Users.Find(id);

if (user == null)

{

return HttpNotFound();

}

return View(user);

}

// POST: Users/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "Email,Name,Password,Status,Rank")] User user)

{

if (ModelState.IsValid)

{

db.Entry(user).State = System.Data.Entity.EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(user);

}

// GET: Users/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

User user = db.Users.Find(id);

if (user == null)

{

return HttpNotFound();

}

return View(user);

}

// POST: Users/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

User user = db.Users.Find(id);

db.Users.Remove(user);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

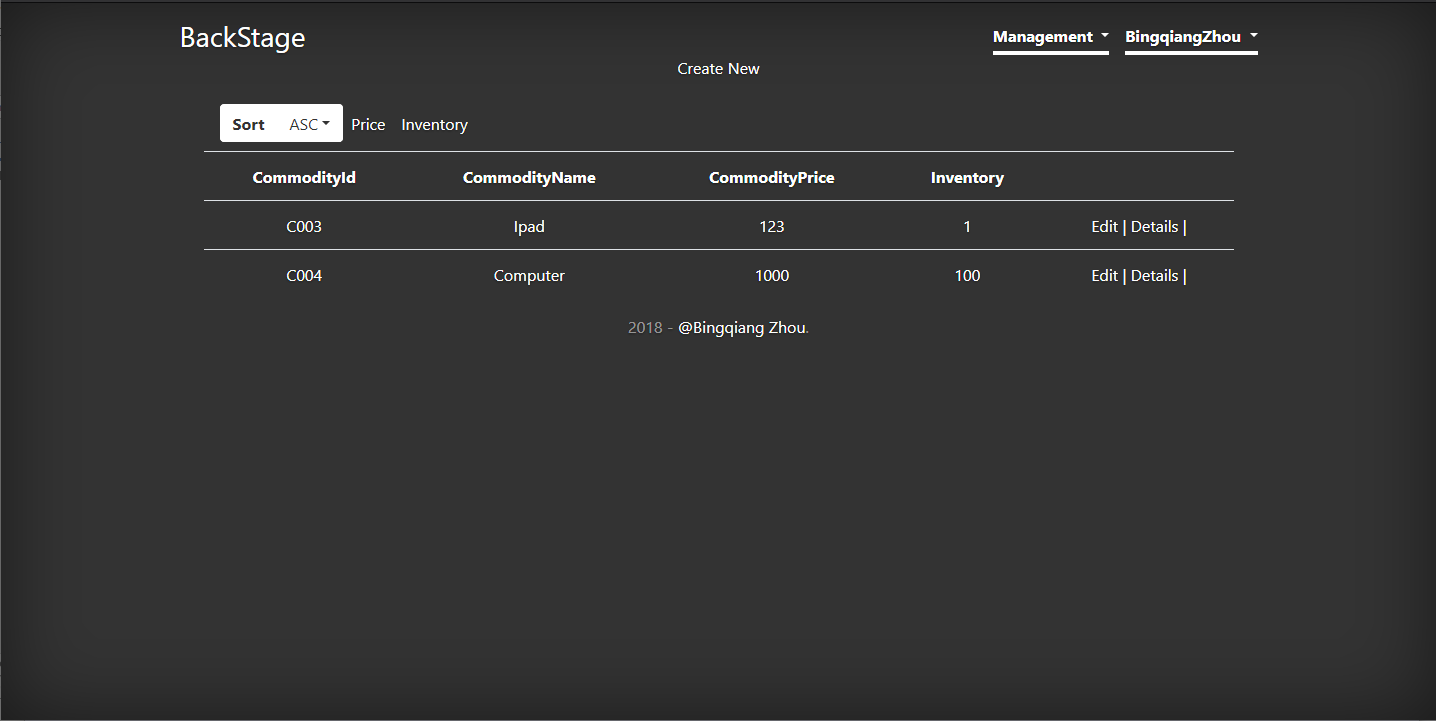
base.Dispose(disposing);

}

}

}

## 商品管理页



图九、商品管理页效果图

### 关键代码

#### Model(Commodity)

namespace Model.Models

{

using System;

using System.Collections.Generic;

public partial class Commodity

{

public string CommodityId { get; set; }

public string CommodityName { get; set; }

public int CommodityPrice { get; set; }

public int CommodityAmount { get; set; }

public string CommodityImage { get; set; }

public string CommodityType { get; set; }

public Commodity(string commodityId, string commodityName, int commodityPrice,

int commodityAmount, string commodityImage, string commodityType)

{

CommodityId = commodityId;

CommodityName = commodityName;

CommodityPrice = commodityPrice;

CommodityAmount = commodityAmount;

CommodityImage = commodityImage;

CommodityType = commodityType;

}

public Commodity()

{

}

}

}

#### View（省略增删改部分）

@model IEnumerable<Model.Models.Commodity>

@{

ViewBag.Title = "Index";

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

}

<p>

@Html.ActionLink("Create New", "Create")

</p>

<form>

<nav class="navbar navbar-expand-lg bg-transparent">

<div class="btn-group" role="group" aria-label="Button group with nested dropdown">

<button type="button" class="btn btn-secondary font-weight-bold" id="Sort">Sort</button>

<div class="btn-group" role="group">

<button id="btnGroupDrop1" type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">ASC</button>

<div class="dropdown-menu" aria-labelledby="btnGroupDrop1">

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

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

</div>

</div>

</div>

<div class="navbar-nav">

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

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

</div>

</nav>

</form>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

@\*<th>

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

</th>\*@

<th>

Inventory

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

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

</td>

</tr>

}

</table>

#### Control

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Model.Models;

using PagedList;

namespace Model.Controllers

{

public class CommodityController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: Commodity

public ActionResult Index(string Sort=" ",string SortType="ASC")

{

if (Sort.Equals("Price"))

{

if(SortType.Equals("DESC"))

return View(db.Commodities.OrderByDescending(q=>q.CommodityPrice));

else

return View(db.Commodities.OrderBy(q => q.CommodityPrice));

}

else if (Sort.Equals("Inventory"))

{

if (SortType.Equals("DESC"))

return View(db.Commodities.OrderByDescending(q => q.CommodityAmount));

else

return View(db.Commodities.OrderBy(q => q.CommodityAmount));

}

else

{

return View(db.Commodities.ToList());

}

}

// GET: Commodity/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Commodity commodity = db.Commodities.Find(id);

if (commodity == null)

{

return HttpNotFound();

}

return View(commodity);

}

// GET: Commodity/Create

public ActionResult Create()

{

ViewBag.Types = GetCommodityType();

return View();

}

[NonAction]

public List<SelectListItem> GetCommodityType()

{

var list = db.CommodityTypes.Select(p => p.TypeId);

var selectItemList = new List<SelectListItem>();

int i = 0;

foreach (var item in list)

{

if (i == 0)

selectItemList.Add(new SelectListItem() { Value = item.ToString(), Text = item.ToString(), Selected = true });

else

selectItemList.Add(new SelectListItem() { Value = item.ToString(), Text = item.ToString() });

i++;

}

return selectItemList;

}

// POST: Commodity/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "CommodityId,CommodityName,CommodityPrice,CommodityAmount,CommodityImage,CommodityType")] Commodity commodity)

{

if (ModelState.IsValid)

{

db.Commodities.Add(commodity);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(commodity);

}

// GET: Commodity/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Commodity commodity = db.Commodities.Find(id);

if (commodity == null)

{

return HttpNotFound();

}

return View(commodity);

}

// POST: Commodity/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "CommodityId,CommodityName,CommodityPrice,CommodityAmount,CommodityImage")] Commodity commodity)

{

if (ModelState.IsValid)

{

db.Entry(commodity).State = System.Data.Entity.EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(commodity);

}

// GET: Commodity/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Commodity commodity = db.Commodities.Find(id);

if (commodity == null)

{

return HttpNotFound();

}

return View(commodity);

}

// POST: Commodity/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

Commodity commodity = db.Commodities.Find(id);

db.Commodities.Remove(commodity);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

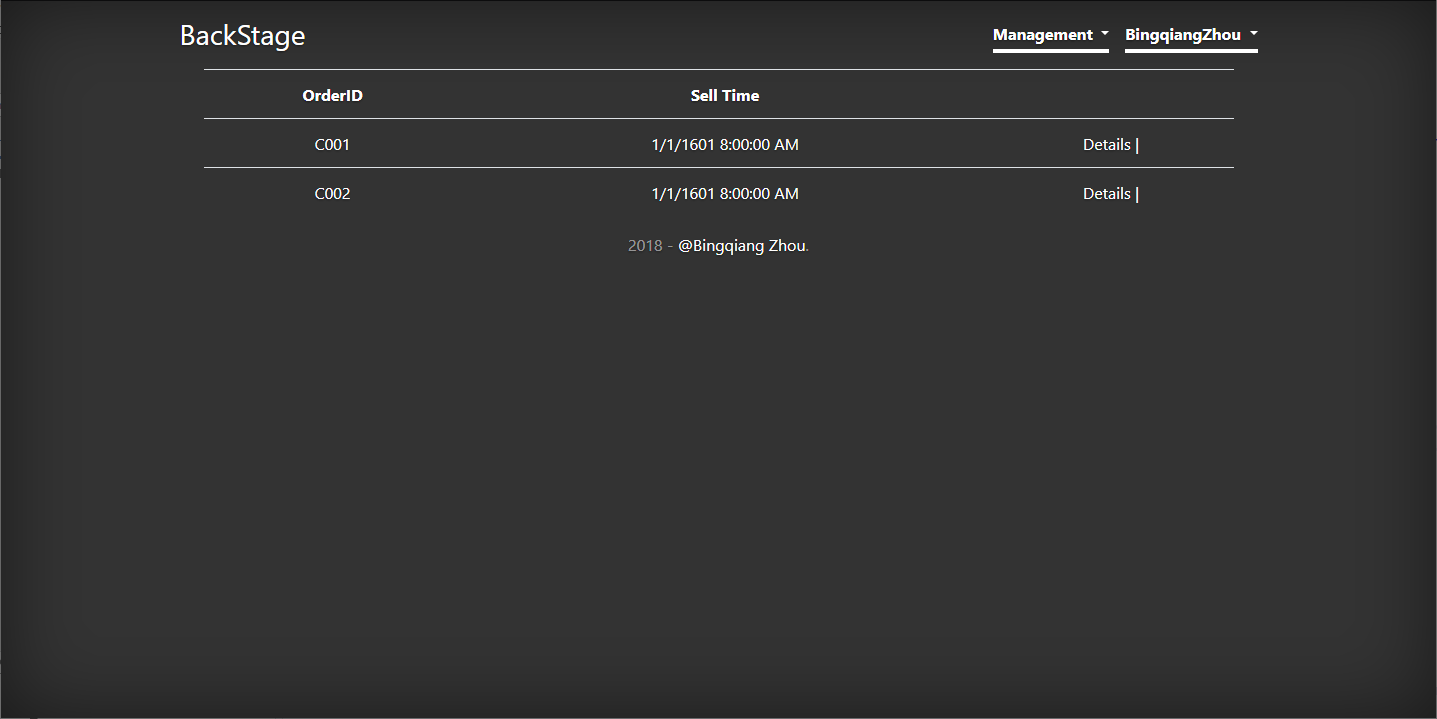
base.Dispose(disposing);

}

}

}

## 销售订单管理页



图十、销售订单管理页

### 关键代码

#### Model(SellOrder)

namespace Model.Models

{

using System;

using System.Collections.Generic;

public partial class SellOrder

{

public string OrderId { get; set; }

public string CommodityID { get; set; }

public int CommodityAmount { get; set; }

}

}

#### View(省略增删改部分)

@model List<string>

@{

ViewBag.Title = "Index";

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

}

<p></p>

<table class="table">

<tr>

<th>

OrderID

</th>

<th>

Sell Time

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item)

</td>

<td>

@{

long time = long.Parse(item.ToString().Substring(1, item.Length - 1));

string time1 = DateTime.FromFileTime(time).ToString();

}

@time1

</td>

<td>

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

</td>

</tr>

}

</table>

#### Control

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Model.Models;

namespace Model.Controllers

{

public class SellOrdersController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: SellOrders

public ActionResult Index()

{

var list = db.SellOrders.Select(p => p.OrderId).Distinct().ToList();

return View(list);

}

// GET: SellOrders/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

var order = db.SellOrders.AsEnumerable().Where(p => p.OrderId.Equals(id));

var list = order.Join(

db.Commodities, p => p.CommodityID, q => q.CommodityId,

(p, q) =>

new Commodity(q.CommodityId,

q.CommodityName,

q.CommodityPrice,

p.CommodityAmount,

q.CommodityImage,q.CommodityType)).ToList();

ViewBag.ID = id;

return View(list);

}

// GET: SellOrders/Create

public ActionResult Create()

{

return View();

}

// POST: SellOrders/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "OrderId,CommodityID,CommodityAmount")] SellOrder sellOrder)

{

if (ModelState.IsValid)

{

db.SellOrders.Add(sellOrder);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(sellOrder);

}

// GET: SellOrders/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

SellOrder sellOrder = db.SellOrders.Find(id);

if (sellOrder == null)

{

return HttpNotFound();

}

return View(sellOrder);

}

// POST: SellOrders/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "OrderId,CommodityID,CommodityAmount")] SellOrder sellOrder)

{

if (ModelState.IsValid)

{

db.Entry(sellOrder).State = System.Data.Entity.EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(sellOrder);

}

// GET: SellOrders/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

SellOrder sellOrder = db.SellOrders.Find(id);

if (sellOrder == null)

{

return HttpNotFound();

}

return View(sellOrder);

}

// POST: SellOrders/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

SellOrder sellOrder = db.SellOrders.Find(id);

db.SellOrders.Remove(sellOrder);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

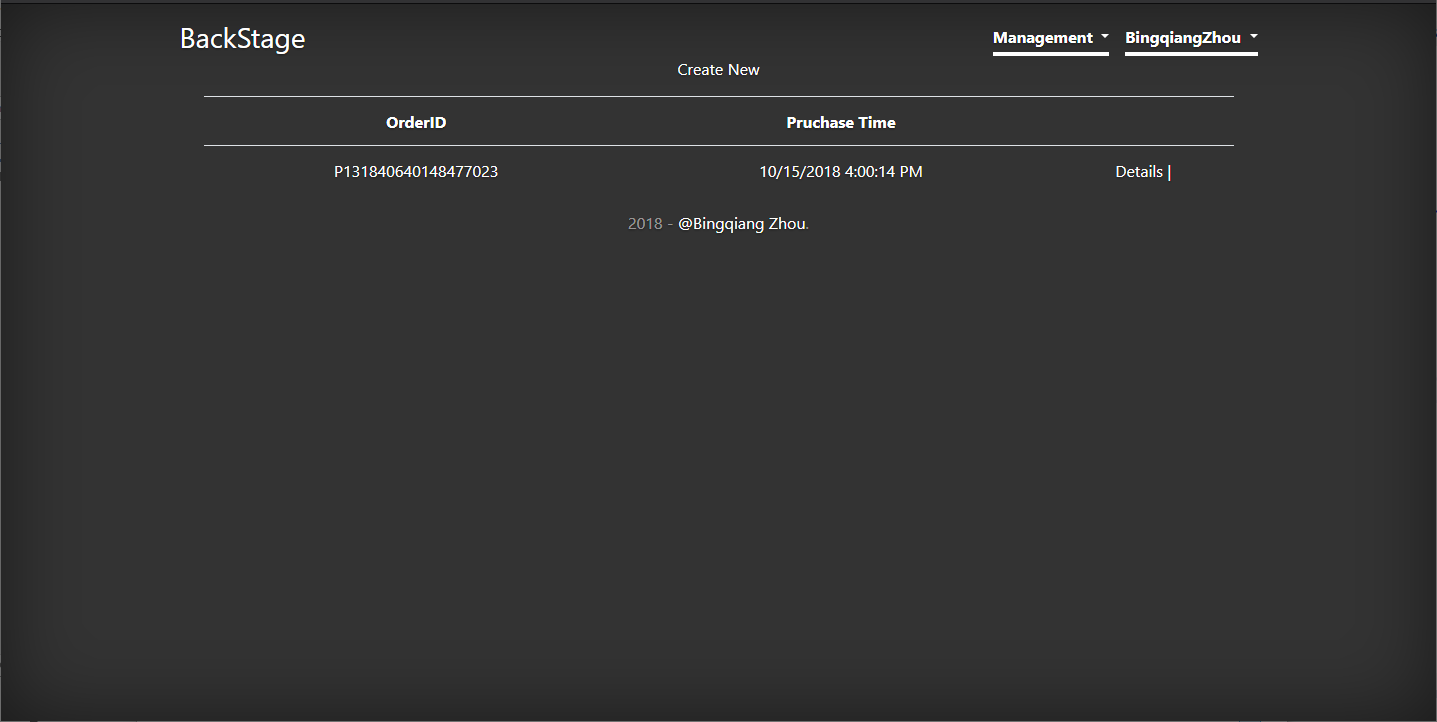
base.Dispose(disposing);

}

}

}

## 入库订单管理页



图十一、入库订单管理页

### 关键代码

#### Model（PurchaseOrder）

namespace Model.Models

{

using System;

using System.Collections.Generic;

public partial class PurchaseOrder

{

public string PurchaseOrderId { get; set; }

public string CommodityId { get; set; }

public int CommodityAmount { get; set; }

}

}

#### View（省略增删改部分）

@model List<string>

@{

ViewBag.Title = "Index";

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

}

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

OrderID

</th>

<th>

Pruchase Time

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item)

</td>

<td>

@{

long time = long.Parse(item.ToString().Substring(1, item.Length - 1));

string time1 = DateTime.FromFileTime(time).ToString();

}

@time1

</td>

<td>

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

</td>

</tr>

}

</table>

#### Control

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using Model.Models;

namespace Model.Controllers

{

public class PurchaseOrdersController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: PurchaseOrders

public ActionResult Index()

{

return View(db.PurchaseOrders.Select(p=>p.PurchaseOrderId).Distinct().ToList());

}

// GET: PurchaseOrders/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

var order = db.PurchaseOrders.AsEnumerable().Where(p => p.PurchaseOrderId.Equals(id));

var list = order.Join(

db.Commodities, p => p.CommodityId, q => q.CommodityId,

(p, q) =>

new Commodity(q.CommodityId,

q.CommodityName,

q.CommodityPrice,

p.CommodityAmount,

q.CommodityImage,q.CommodityType)).ToList();

ViewBag.ID = id;

return View(list);

}

// GET: PurchaseOrders/Create

public ActionResult Create(string id=null)

{

if(id != null)

{

ViewBag.PurchaseOrderId = id;

ViewBag.Commodity = GetCommodityFromOrder(id);

}

ViewBag.ID = GetCommodityID();

return View();

}

[NonAction]

public List<SelectListItem> GetCommodityID()

{

var list = db.Commodities.Select(p => p.CommodityId);

var selectItemList = new List<SelectListItem>();

int i = 0;

foreach (var item in list)

{

if (i == 0)

selectItemList.Add(new SelectListItem() { Value = item.ToString(), Text = item.ToString(), Selected = true });

else

selectItemList.Add(new SelectListItem() { Value = item.ToString(), Text = item.ToString() });

i++;

}

return selectItemList;

}

[NonAction]

public List<Commodity> GetCommodityFromOrder(string id)

{

var commdity = from p in db.PurchaseOrders

join c in db.Commodities on p.CommodityId equals c.CommodityId

where p.PurchaseOrderId == id

select c;

return commdity.ToList();

}

// POST: PurchaseOrders/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "PurchaseOrderId,CommodityId,CommodityAmount")] PurchaseOrder purchaseOrder)

{

if (ModelState.IsValid)

{

db.PurchaseOrders.Add(purchaseOrder);

db.SaveChanges();

}

ViewBag.PurchaseOrderId = purchaseOrder.PurchaseOrderId;

ViewBag.ID = GetCommodityID();

ViewBag.Commodity = GetCommodityFromOrder(purchaseOrder.PurchaseOrderId);

Commodity cc = db.Commodities.Find(purchaseOrder.CommodityId);

cc.CommodityAmount += purchaseOrder.CommodityAmount;

db.SaveChanges();

return View(purchaseOrder);

}

// GET: PurchaseOrders/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

PurchaseOrder purchaseOrder = db.PurchaseOrders.Find(id);

if (purchaseOrder == null)

{

return HttpNotFound();

}

return View(purchaseOrder);

}

// POST: PurchaseOrders/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "PurchaseOrderId,CommodityId,CommodityAmount")] PurchaseOrder purchaseOrder)

{

if (ModelState.IsValid)

{

db.Entry(purchaseOrder).State = System.Data.Entity.EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(purchaseOrder);

}

// GET: PurchaseOrders/Delete/5

public ActionResult Delete(string pid,string cid)

{

if (pid == null || cid == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

PurchaseOrder purchaseOrder = db.PurchaseOrders.Find(pid,cid);

ViewBag.PurchaseOrderId = purchaseOrder.PurchaseOrderId;

if (purchaseOrder == null)

{

return HttpNotFound();

}

return View(purchaseOrder);

}

// POST: PurchaseOrders/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string pid,string cid)

{

PurchaseOrder purchaseOrder = db.PurchaseOrders.Find(pid,cid);

int amount = purchaseOrder.CommodityAmount;

db.PurchaseOrders.Remove(purchaseOrder);

Commodity commodity = db.Commodities.Find(purchaseOrder.CommodityId);

commodity.CommodityAmount -= amount;

db.SaveChanges();

ViewBag.ID = GetCommodityID();

ViewBag.PurchaseOrderId = purchaseOrder.PurchaseOrderId;

return RedirectPermanent("Create/"+purchaseOrder.PurchaseOrderId);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

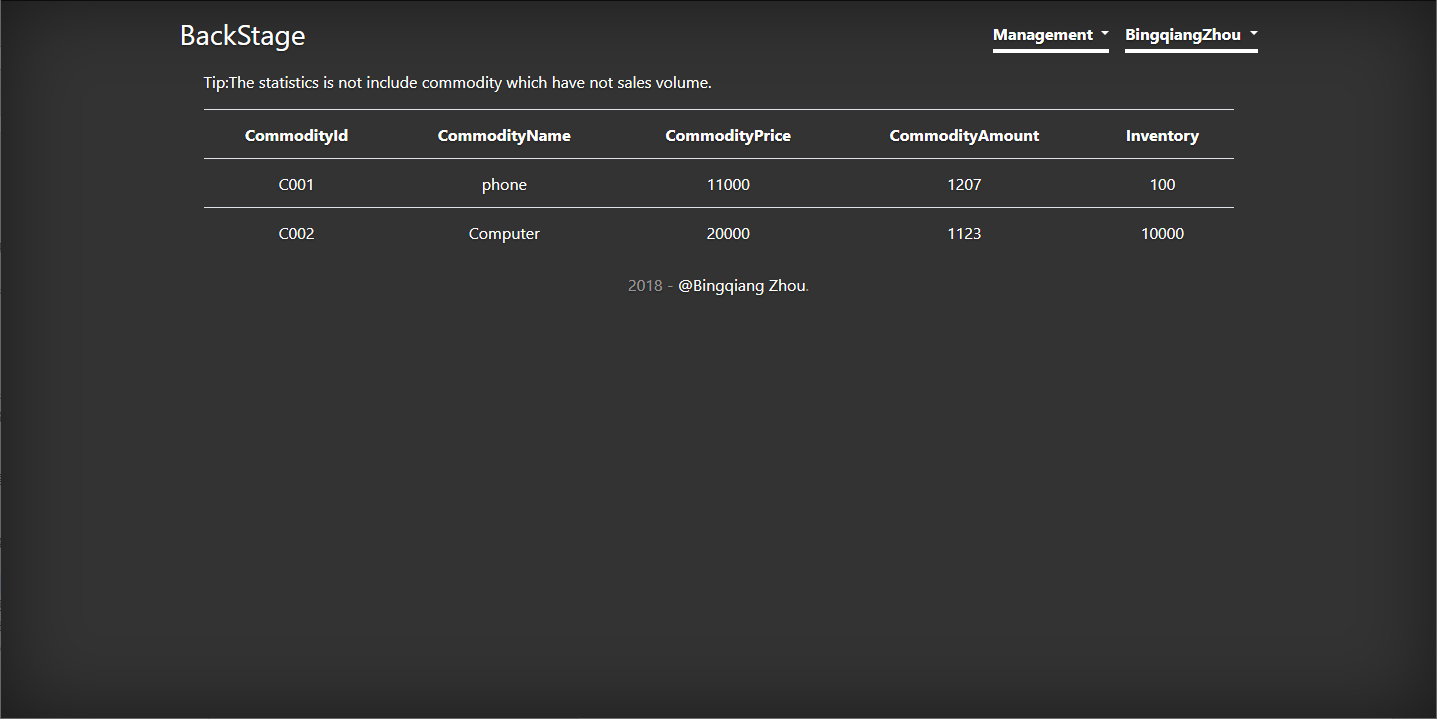
}

}

}

## 销量统计页

### 关键代码



图十二、销量统计页

#### Model(CommodityDetail)

namespace Model.Models

{

using System;

using System.Data.Entity;

using System.Linq;

public class CommodityDetail : DbContext

{

public CommodityDetail()

: base("name=CommoditySaleDetail")

{

}

public virtual DbSet<CommoditySaleDeatil> MyEntities { get; set; }

}

public class CommoditySaleDeatil

{

public string CommodityId { get; set; }

public string CommodityName { get; set; }

public int CommodityPrice { get; set; }

public int CommodityAmount { get; set; }

public string CommodityImage { get; set; }

public int? CommoditySellAmount { get; set; }

public CommoditySaleDeatil()

{

}

public CommoditySaleDeatil(string commodityId, string commodityName, int commodityPrice, int commodityAmount, string commodityImage, int? commoditySellAmount)

{

CommodityId = commodityId;

CommodityName = commodityName;

CommodityPrice = commodityPrice;

CommodityAmount = commodityAmount;

CommodityImage = commodityImage;

CommoditySellAmount = commoditySellAmount;

}

}

}

#### View（省略增删改部分）

@model IEnumerable<Model.Models.CommoditySaleDeatil>

@{

ViewBag.Title = "Index";

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

}

<p></p>

<p class="text-left">Tip:The statistics is not include commodity which have not sales volume.</p>

<table class="table">

<tr>

<th>

CommodityId

</th>

<th>

CommodityName

</th>

<th>

CommodityPrice

</th>

<th>

CommodityAmount

</th>

<th>

Inventory

</th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

</tr>

}

</table>

#### Control

using Model.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Model.Controllers

{

public class MainController : Controller

{

private MVCEntities db = new MVCEntities();

// GET: Main

public ActionResult Index()

{

if(IsLogin() == false)

{

return RedirectToAction("../Index");

}

//get Commodity's Sale Amount

var commodity = db.Commodities.AsEnumerable().Join(db.SellOrders, p => p.CommodityId, q => q.CommodityID,

(p, q) =>

new CommoditySaleDeatil(p.CommodityId, p.CommodityName, p.CommodityPrice,

p.CommodityAmount, p.CommodityImage, q.CommodityAmount));

return View(commodity.ToList());

}

public ActionResult SignOut()

{

Response.Cookies["Name"].Expires = DateTime.Now.AddDays(-1);

Response.Cookies["Email"].Expires = DateTime.Now.AddDays(-1);

return RedirectToAction("../Index");

}

[NonAction]

public bool IsLogin()

{

if(Request.Cookies["Name"] == null || Request.Cookies["Email"] == null)

{

return false;

}

else

{

ViewBag.Name = Request.Cookies["Name"].Value;

return true;

}

}

}

}