# ASP .NET 期末考查（采用MVC模式）

## 实现效果：

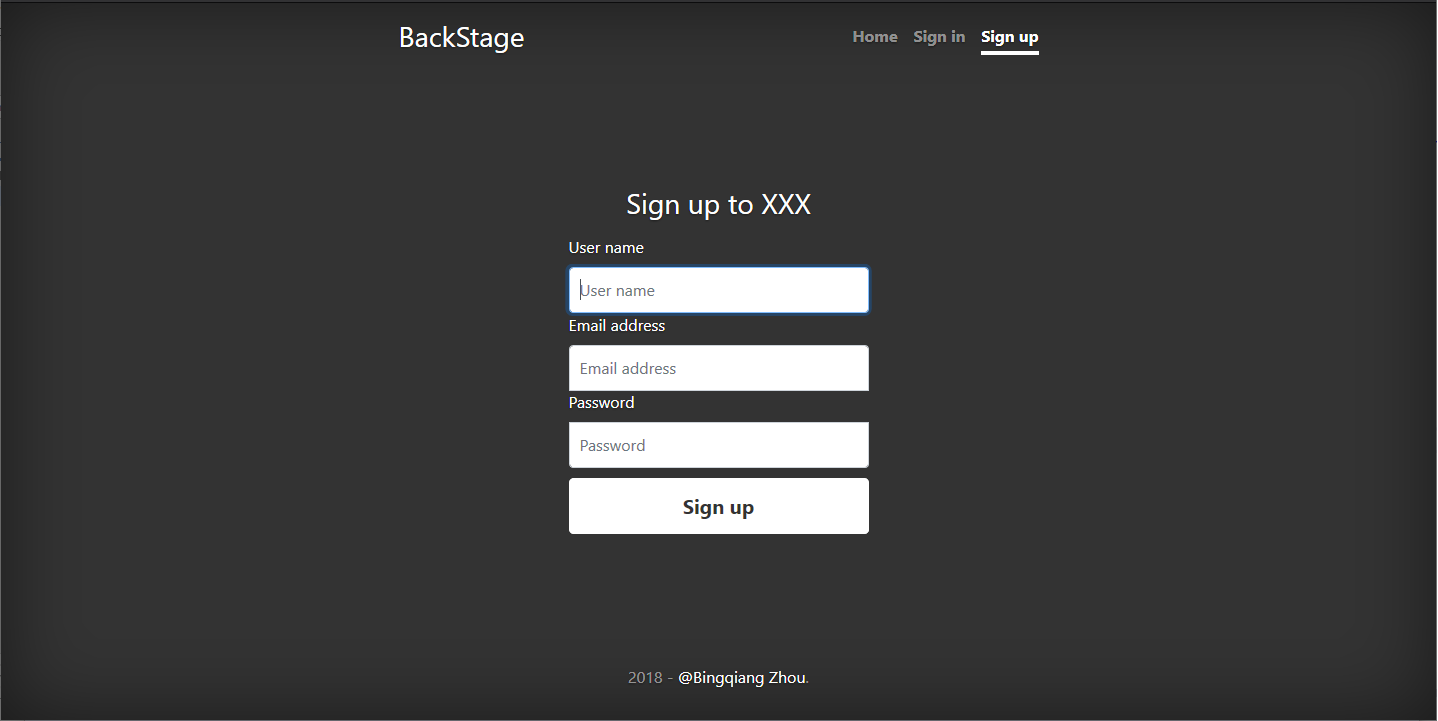
## 可通过浏览http://bingqiangzhou.cn:8888/ 查看效果；

## 测试账号：1299050656@qq.com 密码：123456

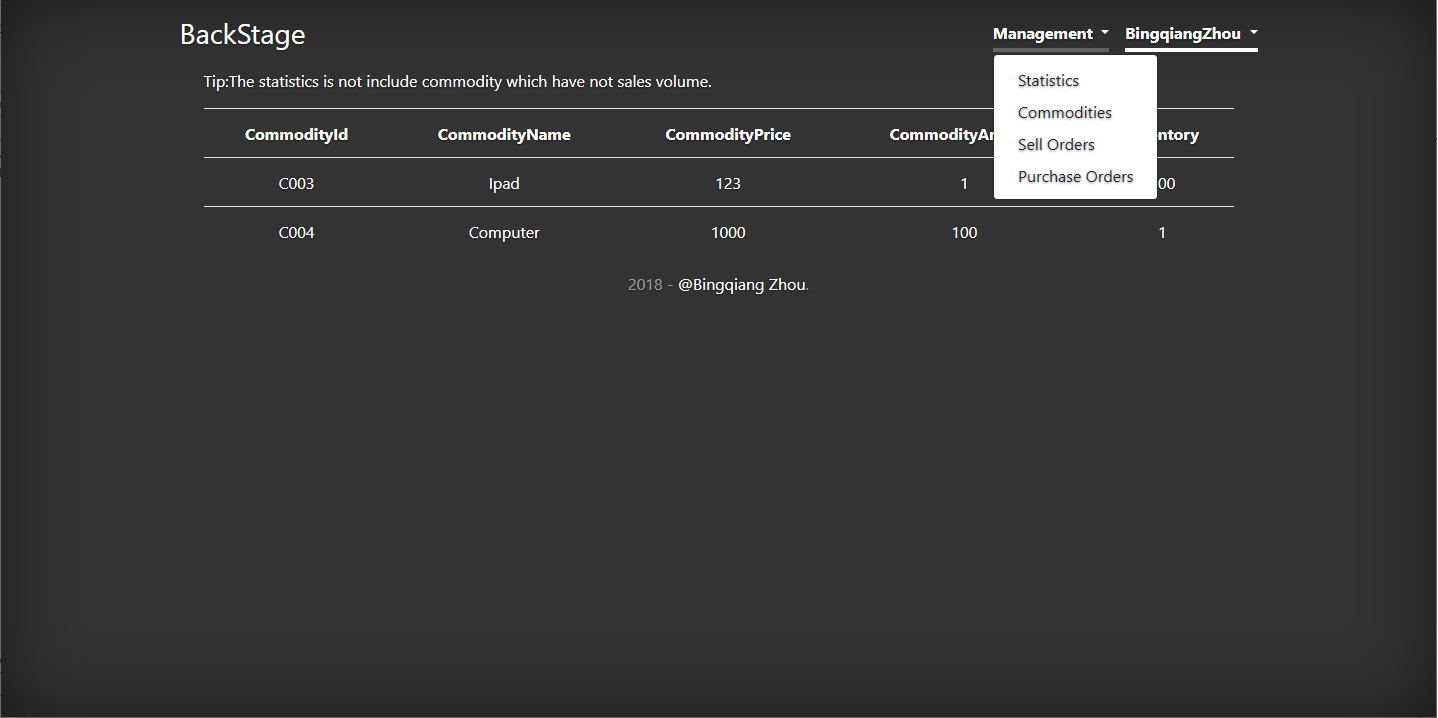
### 首页

### 登录页

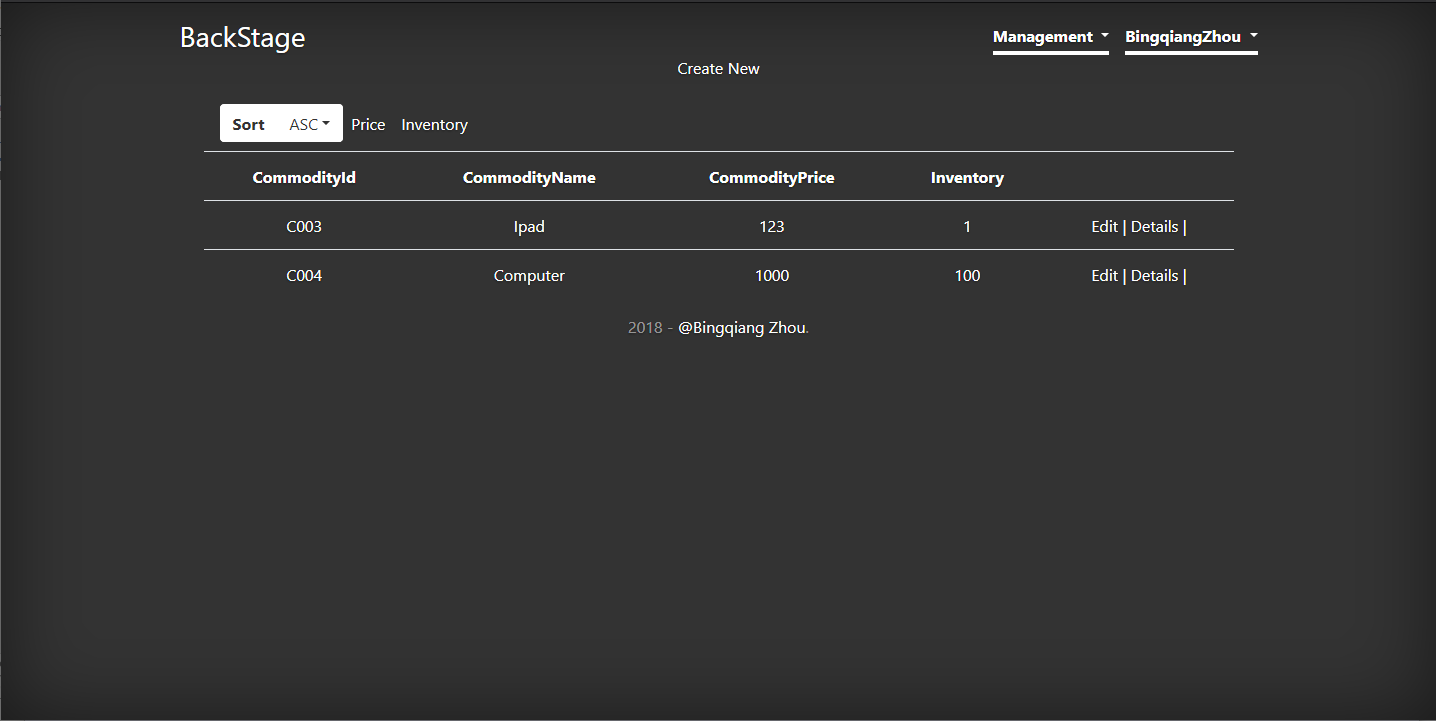
### 注册页

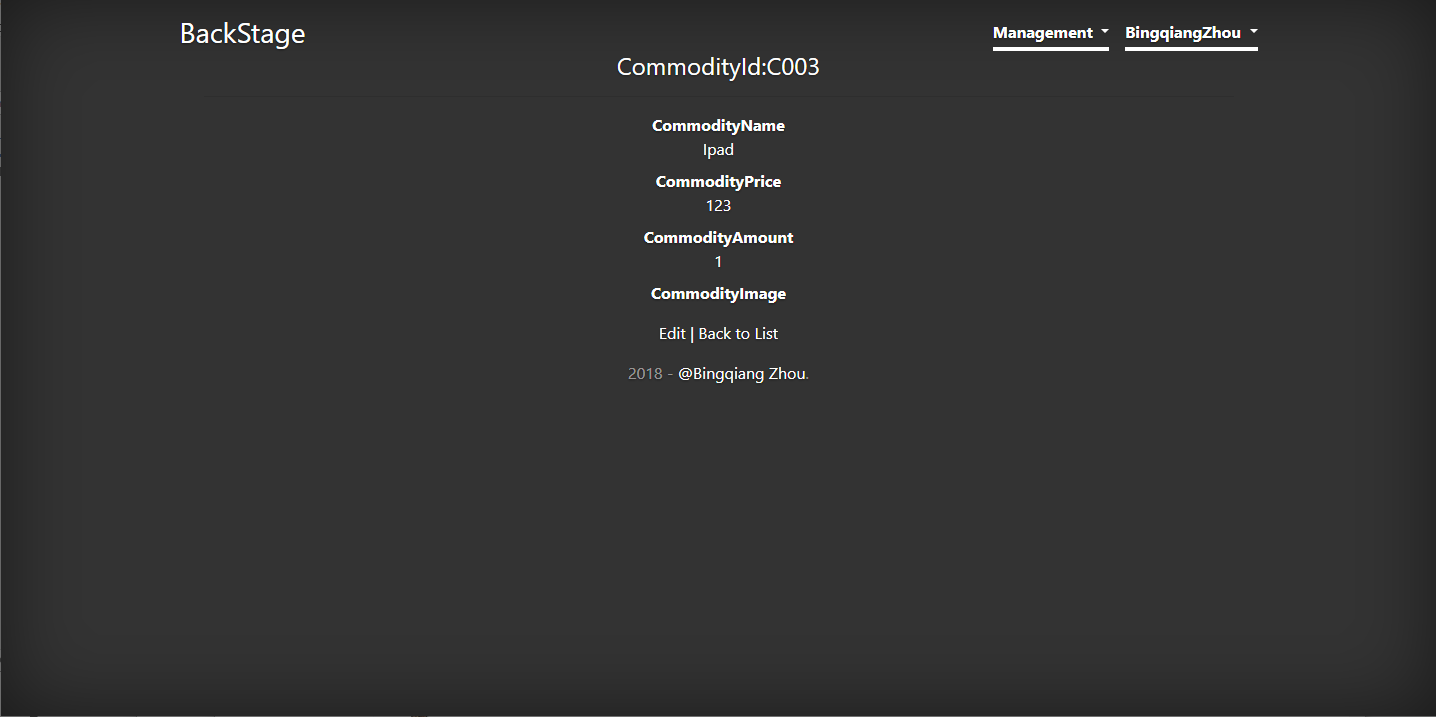
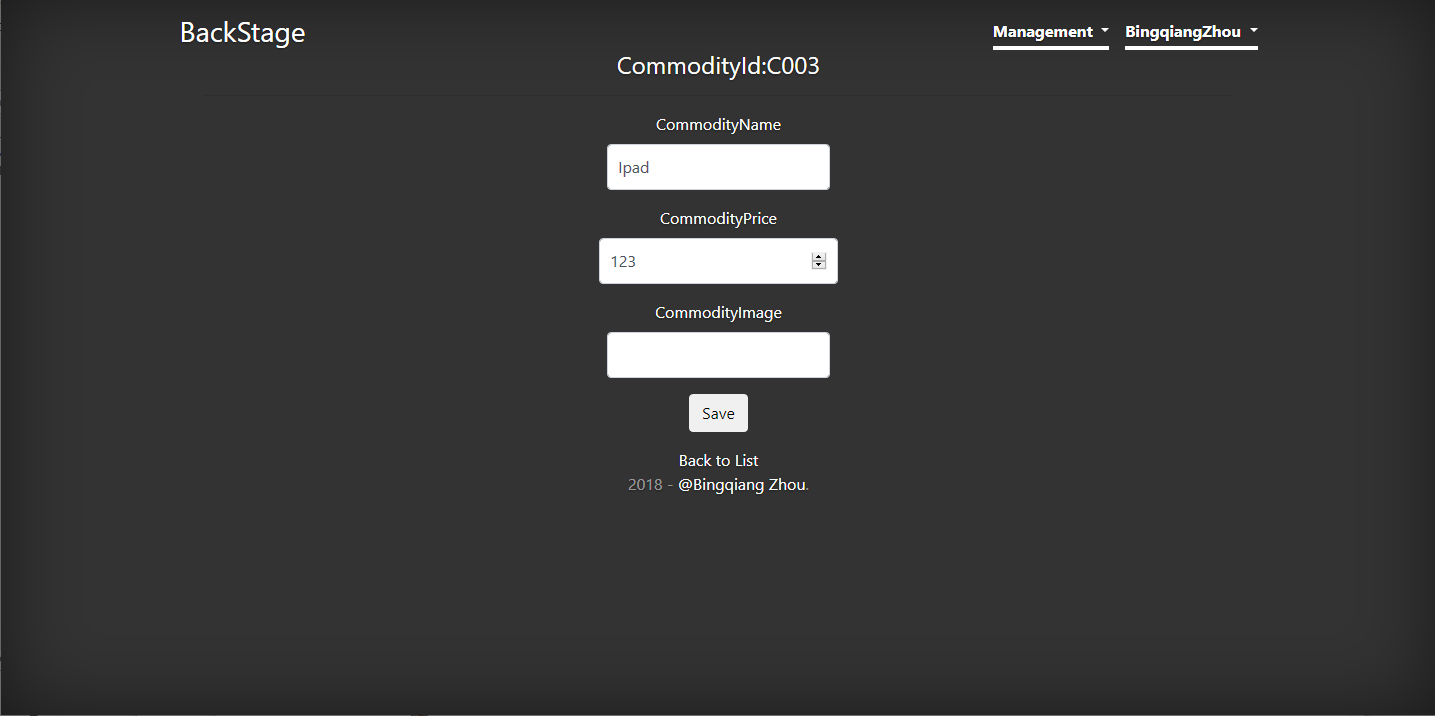
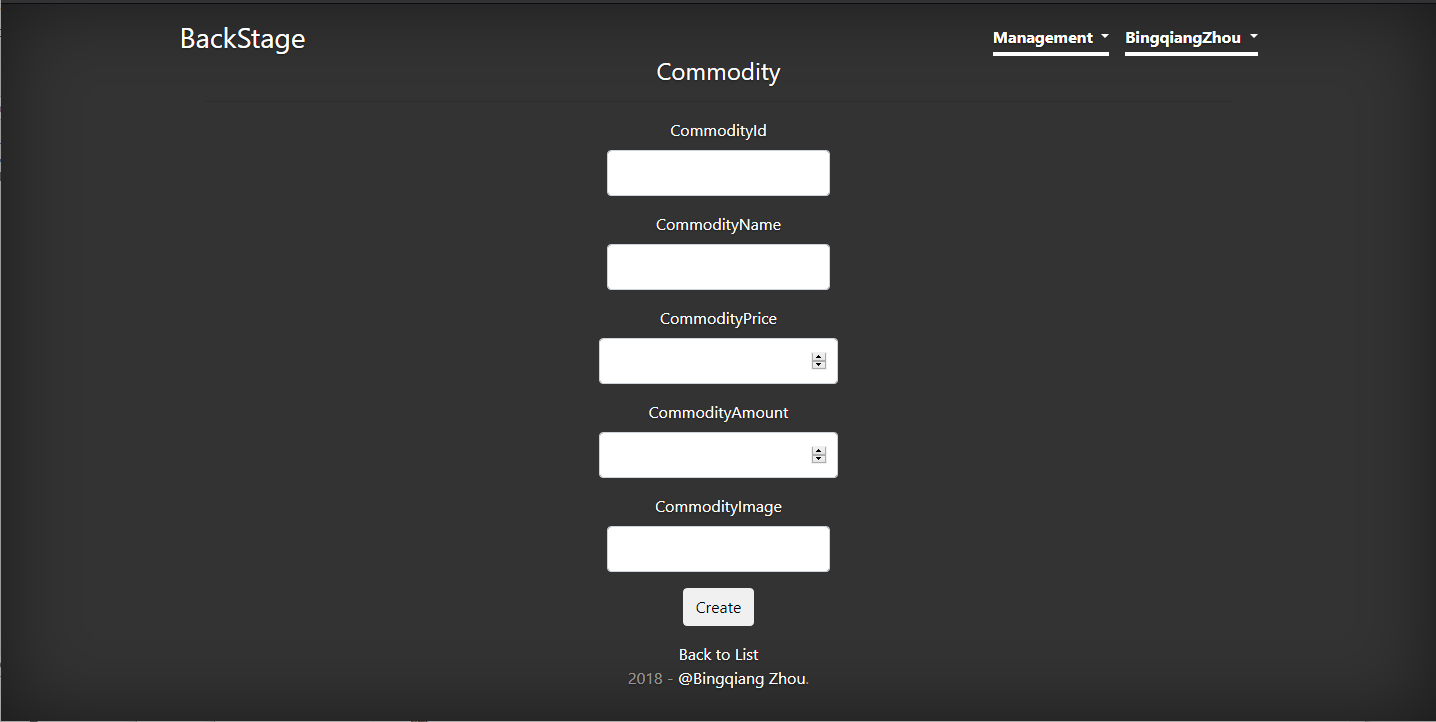


主页：（statistic）显示商品销量

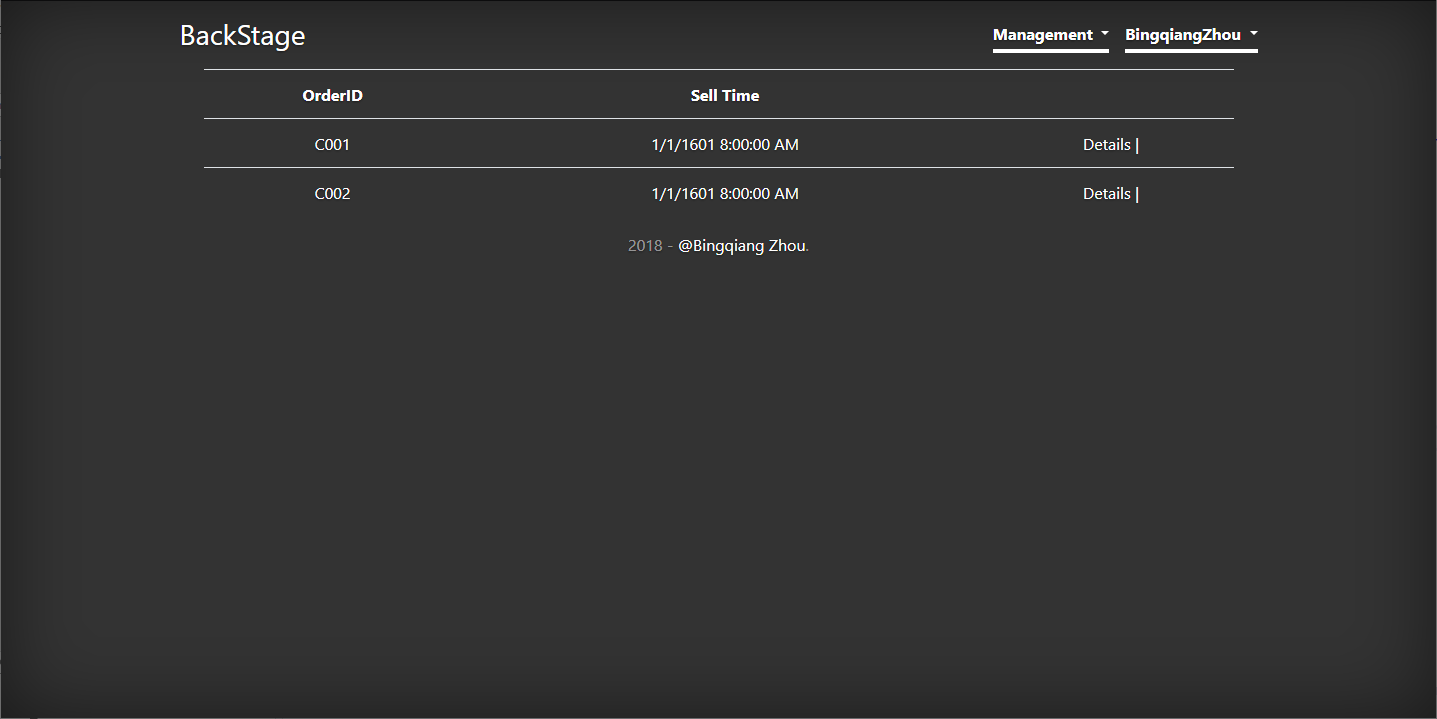


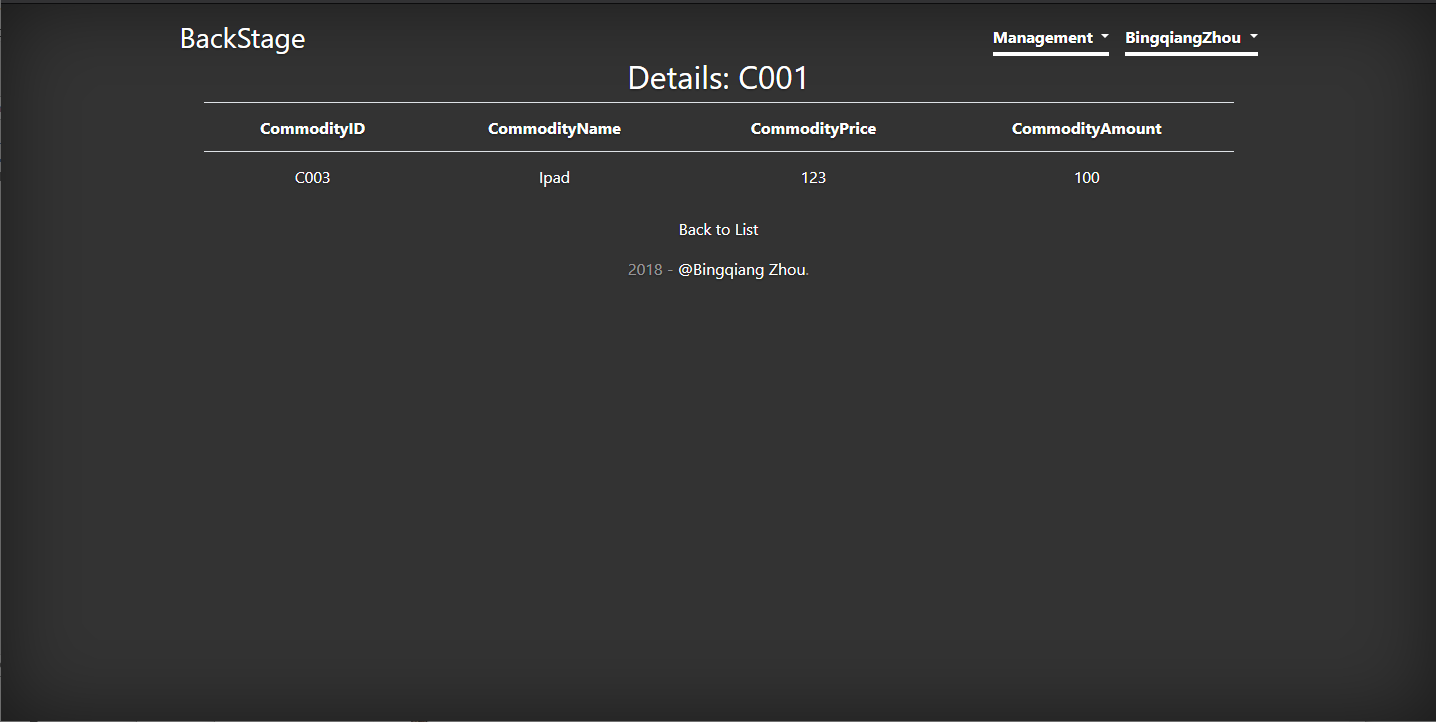
### 商品管理页：可以通过价格和库存排序，包括商品增改查的操作



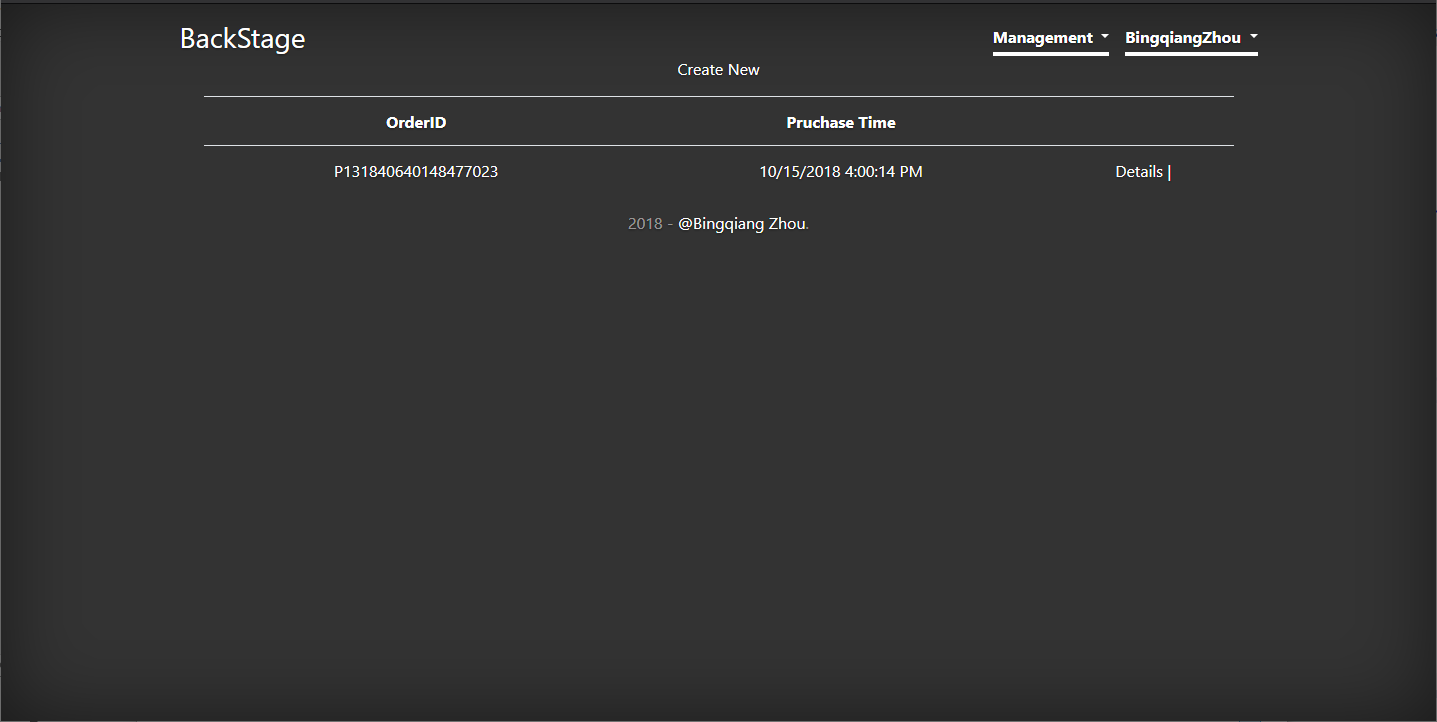


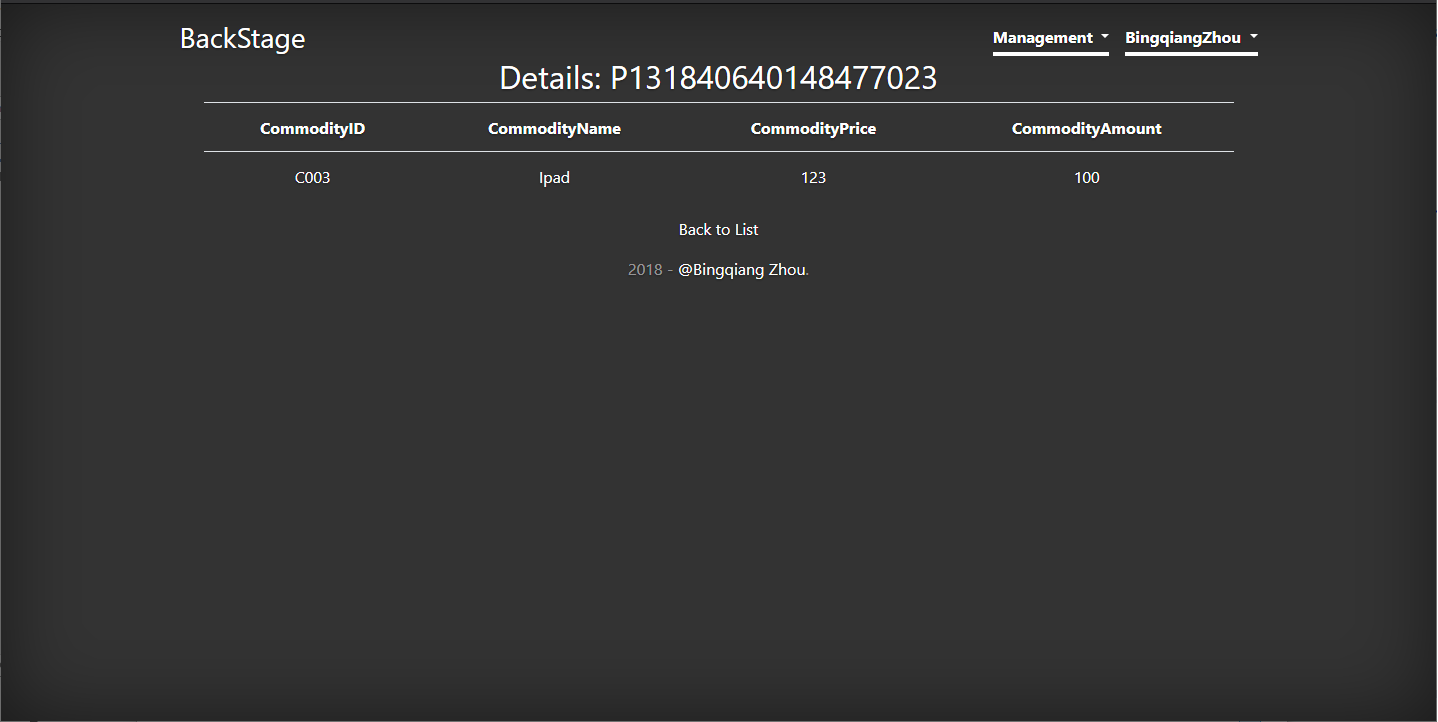
### 销售订单管理页：可查看详情

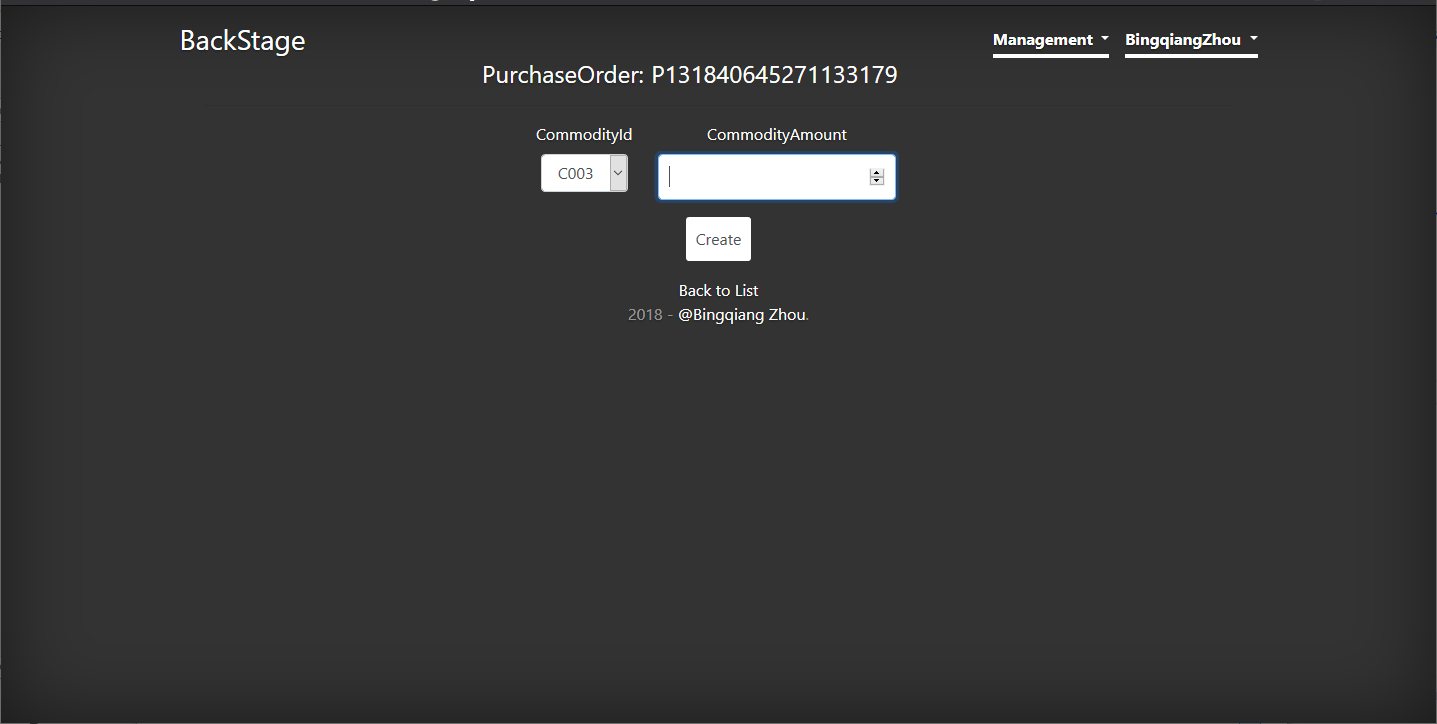




### 入库订单管理页：可以进行增删改查



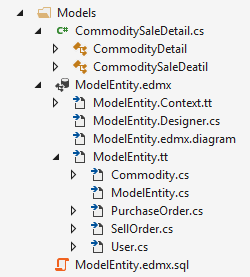




## 关键代码（只贴出小部分，详细可以上GitHub

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

### Model



#### Commodity.cs

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 Commodity(string commodityId, string commodityName, int commodityPrice, int commodityAmount, string commodityImage)

{

CommodityId = commodityId;

CommodityName = commodityName;

CommodityPrice = commodityPrice;

CommodityAmount = commodityAmount;

CommodityImage = commodityImage;

}

public Commodity()

{

}

}

}

#### User.cs

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; }

}

}

#### PurchaseOrder.cs

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; }

}

}

#### SellOrder.cs

namespace Model.Models

{

using System;

using System.Collections.Generic;

public partial class SellOrder

{

public string OrderId { get; set; }

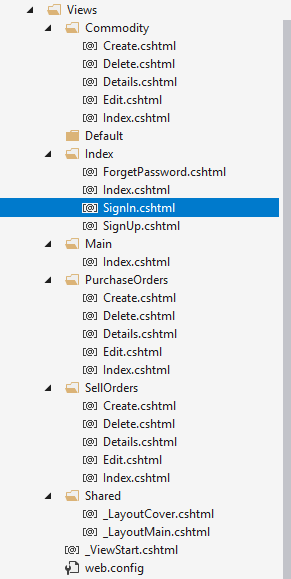
public string CommodityID { get; set; }

public int CommodityAmount { get; set; }

}

}

### View



#### Index（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>

@{

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>

#### Index（index.cshtml）

@{

ViewBag.Title = "Home";

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

}

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

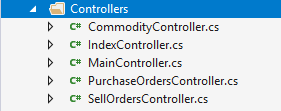
<p class="lead">Live beautifully, dream passionately, love completely.</p>

<p class="lead">

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

</p>

### Control



#### CommodityController.cs

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()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

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

}

}

}

#### IndexController.cs

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);

}

}

}