

Institute of Technology Tallaght, Department of Computing

Higher Diploma in Computing 2018

The screenshot shows a software application window titled "The Shop Application". The main title bar says "The Shop Application" and the sub-title bar says "The Shop Application is a free web framework for your product and customer management". Below the title bar is a blue header bar with a "Start" button. The main content area has three columns:

- The Shop App**
 - University
 - Aline da Silva
 - Niall McClelland
 - Rafal Kita
 - Years of Learning
 - Microsoft Visual Studio 2015
 - Java SE Development Kit
 - MySQL - MySQL Workbench 8.0
- EAD - Enterprise Applications Development**
 - The Shop Application
 - MEET code first with React
 - MEET Shopping Cart
 - MEET Authentication
 - MEET UI
 - The MEET API
 - MEET API adding to a DB
 - MEET Getting up the MEET UI Code First
 - Logout (Logout)
- Institute of Technology Tallaght**
 - The Institute of Technology Tallaght is the largest provider of higher education and knowledge creation within South Dublin County and its environs, for broader participation in higher education in the region.
 - The institute was established as Tallaght Regional Technical College in 1982 at that time the first new Regional Technical College opened in Ireland. It was built on land donated by the local government by the Commissioners in Tallaght, for Educational use.

At the bottom of the window, there is a purple footer bar with the text "© 2018 The Shop Application | This document is not intended for reproduction | Page 18 |"

STUDENTS:
Aline da Silva
Niall McClelland
Rafal Kita

EAD

CA2 Project

Value: 50%

Out: 12/03/18

In: 21/05/18

Demos: 04/05/17

Team Project – 3 people

Lecturer: Dermot Boyle

- A database containing at least 2 tables (SQL Azure or Oracle or mySQL) :

Screenshot of the AWS Amazon RDS Instances page for the shopdb database.

Summary

Engine: SQL Server Express Edition 14.00.3015.40.v1	DB instance class: db.t2.micro	DB instance status: available	Pending maintenance: none
-----------------------------------------------------	--------------------------------	-------------------------------	---------------------------

CloudWatch (58) Legend: shopdb

CPU Utilization (Percent): A line chart showing CPU utilization from 05/16 08:30 to 09:00. It shows a sharp peak around 08:35 reaching approximately 70%.

DB Connections (Count): A line chart showing DB connections from 05/16 08:30 to 09:00. It shows a small peak around 08:45 reaching approximately 0.75.

Free Storage Space (MB): A line chart showing free storage space from 05/16 08:30 to 09:00. It shows a slight dip around 08:45.

Freeable Memory (MB): A line chart showing freeable memory from 05/16 08:30 to 09:00. It shows a peak around 08:45 reaching approximately 150 MB.

Write IOPS (Count/Second): A line chart showing write IOPS from 05/16 08:30 to 09:00. It shows a peak around 08:45 reaching approximately 40 IOPS.

Read IOPS (Count/Second): A line chart showing read IOPS from 05/16 08:30 to 09:00. It shows a peak around 08:45 reaching approximately 150 IOPS.

Connect

Endpoint: shopdb.c79di6yegnby.eu-west-1.rds.amazonaws.com	Port: 1433	Publicly accessible: Yes
-----------------------------------------------------------	------------	--------------------------

Security group rules (2)

Security group	Type	Rule
rds-launch-wizard (sg-0a1639f195e8e3484)	CIDR/IP - Inbound	0.0.0.0/0
rds-launch-wizard (sg-0a1639f195e8e3484)	CIDR/IP - Outbound	0.0.0.0/0

Details

Configurations	Security and network	Instance and IOPS	Maintenance details
ARN: arn:aws:rds:eu-west-1:255981215498:db:shopdb	Availability zone: eu-west-1a	Instance Class: db.t2.micro	Auto minor version upgrade: No
Engine: SQL Server Express Edition 14.00.3015.40.v1	VPC: Default (vpc-b4b42fd3)	Storage Type: General Purpose (SSD)	Maintenance window: tue:05:08-tue:05:38 UTC (GMT)
License Model: License Included	Subnet group: default	Storage: 20 GB	Pending Modifications: None
Created Time: Wed Apr 18 09:14:52 GMT+100 2018	Subnets: subnet-f771eab6, subnet-2cc91677, subnet-f853c16f	Availability and durability: Multi AZ: N/A	Pending maintenance: none
Username: shopuser	Security groups: rds-launch-wizard (sg-0a1639f195e8e3484) (active)	DB instance status: available	Encryption details: Encryption enabled: No
Option Group: defaultsqlserver-ex-14-00	Publicly accessible: Yes	Backup and Restore: Automated backups: Enabled (1 Day)	
Parameter group: default.sqlserver-ex-14.0 (in-sync)	Endpoint: shopdb.c79di6yegnby.eu-west-1.rds.amazonaws.com	Backup window: 00:44-01:14 UTC (GMT)	
Resource ID: db-650CX06MRKBK551ZDUR0YOMCI	Certificate authority: rds-ca-2015 (Mar 5, 2020)	Latest restore time: May 16, 2018 at 9:17:01 AM UTC+1	
		Copy tags to snapshots: No	

- URI addressing scheme for web service:

```
drop table Orders;
drop table Customer;
drop table Product;
```

```
CREATE TABLE Product (
    ProductID int PRIMARY KEY IDENTITY,
    Pname      varchar(255) NOT NULL UNIQUE,
    Price      decimal(10,2),
);
```

```
INSERT INTO Product VALUES ('Product1', 5)
INSERT INTO Product VALUES ('Product2', 10)
INSERT INTO Product VALUES ('Product3', 7)
INSERT INTO Product VALUES ('Product4', 7.6)
INSERT INTO Product VALUES ('Product5', 7.6)
```

```
CREATE TABLE Customer (
    CustomerID int PRIMARY KEY IDENTITY,
    CName      varchar(255) NOT NULL,
    CAddress   varchar(255),
    Phone      varchar(255),
);
```

```
INSERT INTO Customer VALUES ('CustomerA', 'Street 12', '087 111111')
INSERT INTO Customer VALUES ('CustomerB', 'Avenue 7', '088 123123')
INSERT INTO Customer VALUES ('CustomerC', 'Street 47', '099 987987')
INSERT INTO Customer VALUES ('CustomerD', 'Street 120', '087 111111')
INSERT INTO Customer VALUES ('CustomerE', 'Avenue 500', '088 123123')
```

```
CREATE TABLE Orders (
    OrderID      int PRIMARY KEY IDENTITY,
    ProductID    int,
    CustomerID   int,
    Qty          int,
    CONSTRAINT FK_ProductID FOREIGN KEY (ProductID)
    REFERENCES Product(ProductID),
    CONSTRAINT FK_CustomerID FOREIGN KEY (CustomerID)
    REFERENCES Customer(CustomerID)
);
```

```
INSERT INTO Orders VALUES (1, 1, 8)
INSERT INTO Orders VALUES (2, 2, 2)
INSERT INTO Orders VALUES (3, 3, 15)
```

- Application in Amazon Elastic Beanstalk

The screenshot shows the AWS Elastic Beanstalk Overview page for the **shop2-19-dev** environment. The top navigation bar includes links for Services, Resource Groups, VPC, EC2, S3, and a Create New Application button. The main dashboard displays the application's status as **Green** (indicated by a green checkmark icon), the **Running Version** (v20180519120226), and configuration details (IIS 10.0 running on 64bit Windows Server 2016/1.2.0). A table of recent events shows log entries from May 19, 2018, detailing the launch of the environment, setting it to GREEN, completing an update, starting another update, and adding an EC2 instance to an Auto Scaling group.

Time	Type	Details
2018-05-19 13:10:16 UTC+0100	INFO	Successfully launched environment: shop2-19-dev
2018-05-19 13:09:20 UTC+0100	INFO	Environment health has been set to GREEN
2018-05-19 13:09:02 UTC+0100	INFO	UpdateAppVersion Completed
2018-05-19 13:08:59 UTC+0100	INFO	Started Application Update
2018-05-19 13:06:01 UTC+0100	INFO	Added EC2 instance 'i-047af725a7238c67d' to Auto Scaling Group 'awseb-e-ppx8mj9dxe-stack-AWSEBAutoScalingGroup-H1XE6I8R4OOH'.

We're moving to a new design for AWS Elastic Beanstalk. Let us know what you think! You can switch back to the previous version while we finalize the design.

[Feedback](#) | [English \(US\)](#)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) | [Terms of Use](#)

This ad is supporting your extension. [Download](#): More info | Privacy Policy | Hide on this page

- Application running in Amazon:

The screenshot shows the AWS EC2 Dashboard with the following interface elements:

- Left Sidebar:** Lists various AWS services and resources under categories like EC2, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, Systems Manager Services, and Shared Resources.
- Top Bar:** Includes the AWS logo, Services dropdown, Resource Groups dropdown, VPC, EC2, S3 icons, and user information (Rafal, Ireland, Support).
- Table Header:** Shows columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP.
- Table Data:** Two rows are present:
 - Lpic-1 (Instance ID: i-052f8a1ede63fe59e, t2.micro, eu-west-1a, stopped, 0/2 checks, None, ec2-34-247-22-253.eu..., 34.247.22.253)
 - shop2-19-dev (Instance ID: i-047af725a7238c67d, t2.micro, eu-west-1b, running, 2/2 checks, None, ec2-34-247-22-253.eu..., 34.247.22.253)
- Instance Details View:** For the shop2-19-dev instance, it shows the following details:

Description		Status Checks		Monitoring		Tags	
Instance ID	i-047af725a7238c67d	Public DNS (IPv4)	ec2-34-247-22-253.eu-west-1.compute.amazonaws.com	IPv4 Public IP	34.247.22.253	IPv6 IPs	-
Instance state	running	IPv4 IPs	-	Private DNS	ip-172-31-7-139.eu-west-1.compute.internal	Private IPs	172.31.7.139
Instance type	t2.micro	Secondary private IPs	-	VPC ID	vpc-b4b42fd3	Subnet ID	subnet-f853c09f
Elastic IPs	34.247.22.253*	Network interfaces	eth0	Source/dest. check	True	T2 Unlimited	Disabled
Availability zone	eu-west-1b	Owner	295981215498	Launch time	May 19, 2018 at 1:04:35 UTC+1 (less than one hour)	Termination protection	False
Security groups	awsweb-e-ppx8lmj9dxe-stack-AWSEBSecurityGroup-JVJ5TAUQIZY, default, view inbound rules	Lifecycle	normal	Monitoring	basic	Alarm status	None
Scheduled events	No scheduled events	RAM disk ID	-	Kernel ID	-	Placement group	-
AMI ID	aws-elasticbeanstalk-amzn-2018.04.11.x86_64-WindowsServer2016-hvm-201805152132 (ami-a74271de)	Placement group	-				
Platform	windows						
IAM role	aws-elasticbeanstalk-ec2-role						
Key pair name	privateInstanceKey						
EBS-optimized	False						
Root device type	ebs						
Root device	/dev/sda1						
Block devices	/dev/sda1						
Elastic GPU	-						
Elastic GPU type	-						
Elastic GPU status	-						
- Page Bottom:** Includes Feedback, English (US) language selector, and copyright information (© 2008 – 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use).

- Database Set in Azure:

Microsoft Azure

Home > shop2AzureDB

Shop2AzureDB SQL database

Search resources, services, and docs

Copy Restore Export Set server firewall Delete Connect with...

Resource group (change) **shopAzure**

Status **Online**

Location **Central US**

Subscription (change) **Azure for Students**

Subscription ID **d3b8f74e-dac8-4575-9b42-c19ca5f880ca**

Server name **shop2azureserver.database.windows.net**

Elastic database pool **No elastic pool**

Connection strings [Show database connection strings](#)

Pricing tier **Standard S0: 10 DTUs**

Oldest restore point **2018-05-20 11:05 UTC**

Resource

DTU PERCENTAGE 0 %

Activate Windows
Go to Settings to activate Windows.

OVERVIEW ACTIVITY LOG TAGS DIAGNOSIS AND SOLVE PROBLEMS QUICK START QUERY EDITOR (PREVIEW) SETTINGS CONFIGURE GEO-REPLICATION CONNECTION STRINGS SYNC TO OTHER DATABASES ADD AZURE SEARCH DROPSITES

- Application running in Azure:

Microsoft Azure

Home > App Services

App Services
alinecarmodasilvaoutlook (Default Directory)

+ Add Edit columns Refresh Assign Tags Start Restart Stop Delete

Subscriptions: Azure for Students

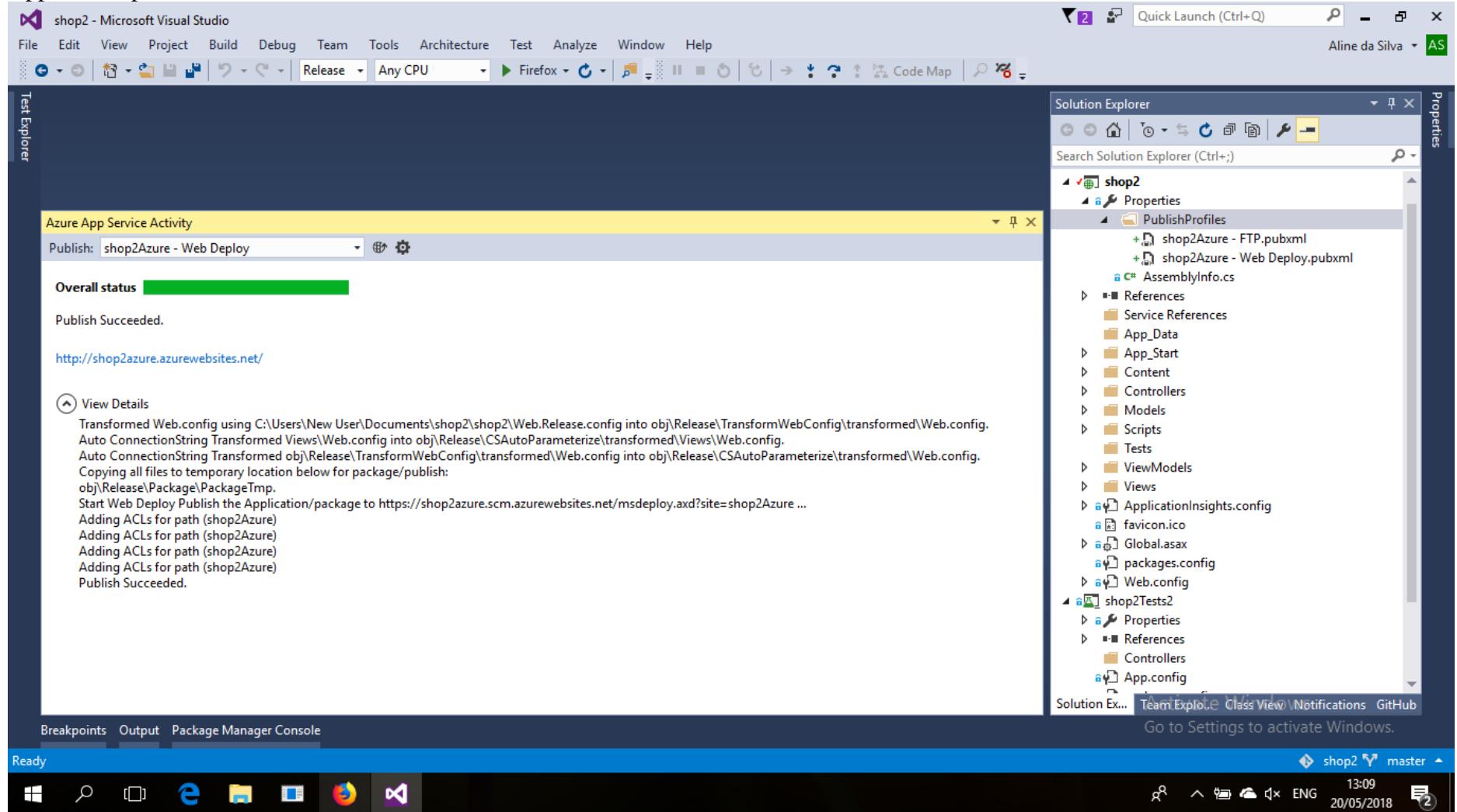
Filter by name... All resource groups All locations No grouping

2 items

NAME ↑↓	STATUS	APP TYPE	APP SERVICE PLAN	LOCATION ↑↓	SUBSCRIPTION ↑↓
shop2	Running	Web app	ServicePlan604c82c6-82a0	Central US	Azure for Students
shop2Azure	Running	Web app	ServicePlan604c82c6-82a0	Central US	Azure for Students

https://portal.azure.com/#

- Application published:



- Application information in Azure:

Microsoft Azure

Home > App Services > shop2Azure

App Services shop2Azure alinecarmodasilvaoutlook (Default Direc...)

+ Add Edit columns More

Filter by name...

NAME
shop2
shop2Azure

shop2Azure App Service

Browse Stop Swap Restart Delete Get publish profile Reset publish profile

Resource group (change) **shopAzure**

Status Running

Location Central US

Subscription (change) **Azure for Students**

Subscription ID d3b8f74e-dac8-4575-9b42-c19ca5f880ca

URL <https://shop2azure.azurewebsites.net>

App Service plan/pricing tier **ServicePlan604c82c6-82a0 (Free: 0 Small)**

FTP/deployment username No FTP/deployment user set

FTP hostname <ftp://waws-prod-dm1-087.ftp.azurewebsites.windows.net>

FTPS hostname <https://waws-prod-dm1-087.ftp.azurewebsites.windows.net>

Diagnose and solve problems
Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

Application Insights
Application Insights helps you detect and diagnose quality issues in your apps, and helps you understand what your users actually do with it.

App Service Advisor
App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.

Http 5xx

100
80
60
40
20
0

14 14:15 14:30 14:45

HTTP SERVER ERRO... !

Data In

100B
80B
60B
40B
20B
0B

14 14:15 14:30 14:45

DATAIN !

Data Out

100B
80B
60B
40B
20B
0B

14 14:15 14:30 14:45

DATA OUT !

- Source code in GitHub:

<https://github.com/rafal321/shop2>

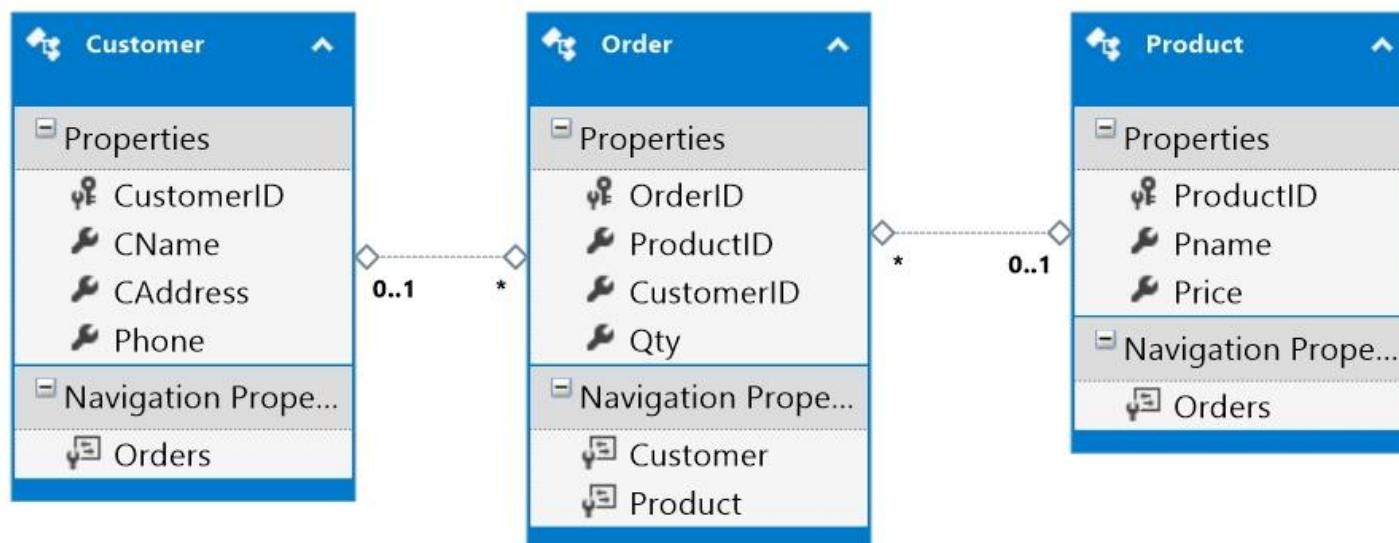
<https://github.com/NAM166/shop2-Client>

- Links to the MVC online:

<http://mvc.glowcloud.xyz/>

<http://shop2azure.azurewebsites.net/>

- Database design:



**Screen shots for:
Web and Console app**

The Shop Application

The Shop Application is a free web framework for your product and customer management.

[IT Login »](#)

The Shop App

Created By:

- [Aline da Silva](#)
- [Niall McClelland](#)
- [Rafal Kita](#)

Tools & Technology used:

[Microsoft Visual Studio »](#)[AWS Elastic Beanstalk »](#)[AWS - RDS »](#)

EAD - Enterprise Applications Development

The Blog We Like

- MVC code-first with related...
- MVC Shopping cart
- Movies 5 - Authentication
- Movies 4 - Client
- Movies 3 - The REST of the CRUD
- Movies 2 - The REST API talking to a db
- Movies 1 - Setting up the Movies db Code-First

[Learn more »](#)

Institute of Technology Tallaght

The Mission of Institute of Technology Tallaght is to be the centre of higher education and knowledge creation within South Dublin County and its environs, to broaden participation in higher education in the region...

The institute was established as Tallaght Regional Technical College, in 1992, at that time the final new Regional Technical College opened in Ireland. It was built on lands donated to the Irish government by the Dominicans in Tallaght, for Educational use.

[Explore more »](#)

Customers Section

[Create New](#)

Find by Name:

Customer Name	Name	Address	
Another one	9999 Street	1231231479	Edit Details Delete
CustomerAA	Street 1234	087 111111	Edit Details Delete
CustomerB	Avenue 7	088 123123	Edit Details Delete

Page 1 of 3

[1](#) [2](#) [3](#) [»](#)

Products Section

[Create New](#)

Find by Name:

Product Name**Price**

Product1

5.00

[Edit](#) | [Details](#) | [Delete](#)

Product22

10.00

[Edit](#) | [Details](#) | [Delete](#)

Product23

12.78

[Edit](#) | [Details](#) | [Delete](#)

Page 1 of 3

[1](#) [2](#) [3](#) [»](#)

Orders Section

[Create New](#)

Find by Name:

Quantity	Customer Name	Product Name	
10	CustomerB	Product22	Edit Details Delete
15	CustomerC	Product3	Edit Details Delete
23	CustomerAA	Product1	Edit Details Delete

Page 1 of 1

[1](#)

Orders Section

[Create New](#)

Find by Name:

 [Search](#)

Quantity	Customer Name	Product Name	
10	CustomerB	Product22	Edit Details Delete
15	CustomerC	Product3	Edit Details Delete
23	CustomerAA	Product1	Edit Details Delete

Page 1 of 1

[1](#)

About.

This is a page about our Shop.

The Shop Application - video information



Meet Our Boss



Meet the Team



Aline da Silva

CEO

Some text that describes me lorem ipsum ipsum lorem.

acarmodasilva@yahoo.com.br

[Linkedin](#)

Niall McClelland

CEO

Some text that describes me lorem ipsum ipsum lorem.

example@example.com

[Linkedin](#)

Rafal Kita

CEO

Some text that describes me lorem ipsum ipsum lorem.

example@example.com

[Linkedin](#)

Contact.

Our contact page.

Institute of Technology, Tallaght
Dublin 24, D24 FKT9
Phone: 425.555.0100

Support: Support@example.com
Marketing: Marketing@example.com



Customers Section

[Create New](#)

Find by Name:

Customer Name	Name	Address	
CustomerC	Street 47	099 987987	Edit Details Delete
CustomerD	Street 120	087 111111	Edit Details Delete
Dave	123 college street	0854541111	Edit Details Delete

Page 2 of 3

[«](#) [1](#) [2](#) [3](#) [»](#)

Customers Section

[Create New](#)

Find by Name:

Customer Name	Name	Address	
nama A	122aaa	4651651	Edit Details Delete
sadasdasd	asdads	55555	Edit Details Delete

Page 3 of 3

[«](#) [1](#) [2](#) [3](#)

Products Section

[Create New](#)

Find by Name:

Product Name	Price	
Product3	7.00	Edit Details Delete
Product4	7.60	Edit Details Delete
Product5	7.60	Edit Details Delete

Page 2 of 3

[«](#) [1](#) [2](#) [3](#) [»](#)

Products Section

[Create New](#)

Find by Name: [Search](#)

Product Name	Price	
ProductB	23.50	Edit Details Delete

Page 3 of 3

« [1](#) [2](#) [3](#)

Edit

Customer

Name

Another one

Address

9999 Street

Phone

1231231479

[Back to List](#)

Details

Customer

Name Another one
Address 9999 Street
Phone 1231231479

Orders

Product Name	Quantity
--------------	----------

[Edit](#) | [Back to List](#)

Edit

Customer

Name

Another one

Address

9999 Street

Phone

1231231479

[Back to List](#)

Edit

Customer

Name

Another one

Address

9999 Street dublin

Phone

1231231479

[Back to List](#)

Customers Section

[Create New](#)

Find by Name:

Customer Name	Name	Address	
Another one	9999 Street dublin	1231231479	Edit Details Delete
CustomerAA	Street 1234	087 111111	Edit Details Delete
CustomerB	Avenue 7	088 123123	Edit Details Delete

Page 1 of 3

[1](#) [2](#) [3](#) [»](#)

Delete

Are you sure you want to delete this?

Customer

Name	Another one
Address	9999 Street dublin
Phone	1231231479

[Delete](#) | [Back to List](#)

Customers Section

[Create New](#)

Find by Name:

Customer Name	Name	Address	
CustomerAA	Street 1234	087 111111	Edit Details Delete
CustomerB	Avenue 7	088 123123	Edit Details Delete
CustomerC	Street 47	099 987987	Edit Details Delete

Page 1 of 3

[1](#) [2](#) [3](#) [»](#)

Edit

Product

Name

Product1

Price

5.00

[Back to List](#)

Details

Product

Name	Product1
Price	5.00

Customer	Quantity
CustomerAA	23

[Edit](#) | [Back to List](#)

Delete

Are you sure you want to delete this?

Product

Name	Product1
Price	5.00

[Delete](#)[| Back to List](#)

Edit

Order

ProductID	Product22	<input type="button" value="▼"/>
CustomerID	CustomerB	<input type="button" value="▼"/>
Qty	10	<input type="button" value="▼"/>

[Back to List](#)

Edit

Order

ProductID	Product22	<input type="button" value="▼"/>
CustomerID	CustomerB	<input type="button" value="▼"/>
Qty	14	<input type="button" value="▼"/>

[Back to List](#)

Orders Section

[Create New](#)

Find by Name:

Quantity	Customer Name	Product Name	
14	CustomerB	Product22	Edit Details Delete
15	CustomerC	Product3	Edit Details Delete
23	CustomerAA	Product1	Edit Details Delete

Page 1 of 1

[1](#)

Delete

Are you sure you want to delete this?

Order

Qty	14
Name	CustomerB
Name	Product22

[Delete](#) | [Back to List](#)

shop2Client (Running) - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Quick Launch (Ctrl+Q)

Aline da Silva AS

Process All Customers in database by descending Customer ID:
Customer ID: 9 Customer Name: sadasdasd Customer Address: asdads Phone Number: 55555
Customer ID: 11 Customer Name: nama A Customer Address: 122aaa Phone Number: 4651651
Customer ID: 14 Customer Name: Dave Customer Address: 123 college street Phone Number: 0854541111
Customer ID: 4 Customer Name: CustomerD Customer Address: Street 120 Phone Number: 087 111111
Customer ID: 3 Customer Name: CustomerC Customer Address: Street 47 Phone Number: 099 987987
Customer ID: 2 Customer Name: CustomerB Customer Address: Avenue 7 Phone Number: 088 123123
Customer ID: 1 Customer Name: CustomerAA Customer Address: Street 1234 Phone Number: 087 111111
Customer 1:
Customer ID: 2 Customer Name: CustomerB Customer Address: Avenue 7 Phone Number: 088 123123
Customers containing Dave in name:
[{"CustomerID":14,"CName":"Dave "}]Create new Customer
Customer Alex Dunne added

1", Phone = "0876589995");

100 %

Watch 1

Name	Type

Immediate Window

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List

Activate Windows
Copy Settings to activate Windows.

Ready Ln 82 Col 78 Ch 78 INS shop2-Client master 23:53 20/05/2018

Windows Taskbar icons: File Explorer, Edge, File Manager, File Explorer, Word, Google Chrome, File Manager

Source Code:

MVC Controllers:

```
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.Http.Results;
using System.Web.Mvc;
using shop2.Models;
using PagedList;

namespace shop2.Controllers
{
    public class CustomerController : Controller
    {
        //private shopdbEntities db = new shopdbEntities();
        public ICustomersContext db = new CustomersContext();

        public CustomerController() { }

        public CustomerController(ICustomersContext context)
        {
            db = context;
        }

        // GET: Customer
        //public ActionResult Index() //method before sorting/filtering implemented
        //{
        //    return View(db.Customers.ToList());
        //}
        =====sorting filtering=====
        public ActionResult Index(string sortOrder, string currentFilter, string
searchString, int? page)
        {
            ViewBag.CurrentSort = sortOrder;
            ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
            ViewBag.AddressSortParm = sortOrder == "Address" ? "address_desc" :
"Address";
            //ViewBag.PhoneSortParm = sortOrder == "Phone" ? "phone_desc" : "Phone";

            if (searchString!=null)
            {
                page = 1;
            }
            else
            {
                searchString = currentFilter;
            }
            ViewBag.CurrentFilter = searchString;

            var customers = from c in db.Customers
                            select c;
            =====searchinf stuff
            if (!String.IsNullOrEmpty(searchString))
            {
                customers = customers.Where(c => c.CName.Contains(searchString));
            }
            =====
        }
    }
}
```

```

        switch (sortOrder)
    {
        case "name_desc":
            customers = customers.OrderByDescending(x => x.CName);
            break;
        case "Address":
            customers = customers.OrderBy(s => s.CAddress);
            break;
        case "address_desc":
            customers = customers.OrderByDescending(s => s.CAddress);
            break;
        //case "Phone":
        //    customers = customers.OrderBy(s => s.Phone);
        //    break;
        //case "phone_desc":
        //    customers = customers.OrderByDescending(s => s.Phone);
        //    break;
        default:
            customers = customers.OrderBy(y => y.CName);
            break;
    }
    int pageSize = 3;
    int pageNumber = (page ?? 1);
    return View(customers.ToPagedList(pageNumber, pageSize));
}

//=====
// GET: Customer/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Customer customer = db.Customers.Find(id);
    if (customer == null)
    {
        return HttpNotFound();
    }
    return View(customer);
}

// GET: Customer/Create
public ActionResult Create()
{
    return View();
}

// POST: Customer/Create
// To protect from overposting attacks, please enable the specific properties
you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]           //raf: attribute helps prevent cross-site
request forgery attacks.
public ActionResult Create([Bind(Include = "CustomerID,CName,CAddress,Phone")]
Customer customer)
{
    try
    {
        if (ModelState.IsValid)      //raf: server-side validation
        {

```

```

        db.Customers.Add(customer);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}
catch (DataException)
{
    ModelState.AddModelError("", "Unable to save changes. Try again, and
if the problem persists see your system administrator.");
}
return View(customer);
}

// GET: Customer/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Customer customer = db.Customers.Find(id);
    if (customer == null)
    {
        return HttpNotFound();
    }
    return View(customer);
}

// POST: Customer/Edit/5
// To protect from overposting attacks, please enable the specific properties
you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "CustomerID,CName,CAddress,Phone")]
Customer customer)
{
    if (ModelState.IsValid)
    {
        //db.Entry(customer).State = EntityState.Modified;
        db.MarkAsModified(customer);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(customer);
}

//This parameter is false when the HttpGet Delete method is called without a
//previous failure. When it is called by the HttpPost Delete method in response
to entry
// a database update error, the parameter is true and an error message is
passed to the view.

// GET: Customer/Delete/5
public ActionResult Delete(int? id, bool? saveChangesError = false)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
}

```

```

        }
        //raf: Updating the Delete Page
        if (saveChangesError.GetValueOrDefault())
        {
            ViewBag.ErrorMessage = "Delete failed. Try again and if the problem
persists see your system administrator.";
        }
        //-----

        Customer customer = db.Customers.Find(id);
        if (customer == null)
        {
            return HttpNotFound();
        }
        return View(customer);
    }

    // POST: Customer/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        try
        {
            //raf: delete operation catches any database update errors
            Customer customer = db.Customers.Find(id);
            db.Customers.Remove(customer);
            db.SaveChanges();
        }
        catch (DataException)
        {
            return RedirectToAction("Delete", new { id = id, saveChangesError =
true });
        }
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
        //Closing Database Connections
    }
}
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace shop2.Controllers
{
    public class HomeController : Controller
    {

        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "This is a page about our Shop.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Our contact page.";

            return View();
        }
        public ActionResult Team()
        {
            ViewBag.Message = "This is a page about our Shop.";

            return View();
        }
    }
}
```

```

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 shop2.Models;
using PagedList;

namespace shop2.Controllers
{
    public class OrdersController : Controller
    {
        private shopdbEntities db = new shopdbEntities();

        // GET: Orders

        public ActionResult Index(string sortOrder, string currentFilter, string
searchString, int? page)
        {
            ViewBag.CurrentSort = sortOrder;
            ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "qty_desc" : "";
            ViewBag.AddressSortParm = sortOrder == "Qty" ? "qty_desc" : "Qty";

            if (searchString != null)
            {
                page = 1;
            }
            else
            {
                searchString = currentFilter;
            }
            ViewBag.CurrentFilter = searchString;

            var orders = from c in db.Orders
                         select c;
            //====searchinf stuff
            // int result = int.parseInt(searchString);
            //int value;

            //int xxx = int.Parse(searchString);
            //if (!String.IsNullOrEmpty(searchString))
            //{
            //    orders = orders.Where(c => c.Qty == xxx);
            //}
            //=====
            switch (sortOrder)
            {
                case "qty_desc":
                    orders = orders.OrderByDescending(x => x.Qty);
                    break;
                default:
                    orders = orders.OrderBy(y => y.Qty);
                    break;
            }
            int pageSize = 3;
            int pageNumber = (page ?? 1);
            return View(orders.ToPagedList(pageNumber, pageSize));
    }
}

```

```

        //return View(orders.ToList());
    }

    // GET: Orders/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Order order = db.Orders.Find(id);
        if (order == null)
        {
            return HttpNotFound();
        }
        return View(order);
    }

    // GET: Orders/Create
    public ActionResult Create()
    {
        ViewBag.CustomerID = new SelectList(db.Customers, "CustomerID", "CName");
        ViewBag.ProductID = new SelectList(db.Products, "ProductID", "Pname");
        return View();
    }

    // POST: Orders/Create
    // To protect from overposting attacks, please enable the specific properties
    you want to bind to, for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "OrderID,ProductID,CustomerID,Qty")] Order order)
    {
        try
        {
            if (ModelState.IsValid)
            {
                db.Orders.Add(order);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }
        catch
        {
            ModelState.AddModelError("", "Unable to save changes. Try again, and
            if the problem persists see your system administrator.");
        }
        ViewBag.CustomerID = new SelectList(db.Customers, "CustomerID", "CName",
        order.CustomerID);
        ViewBag.ProductID = new SelectList(db.Products, "ProductID", "Pname",
        order.ProductID);
        return View(order);
    }

    // GET: Orders/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

```

```

        }
        Order order = db.Orders.Find(id);
        if (order == null)
        {
            return HttpNotFound();
        }
        ViewBag.CustomerID = new SelectList(db.Customers, "CustomerID", "CName",
order.CustomerID);
        ViewBag.ProductID = new SelectList(db.Products, "ProductID", "Pname",
order.ProductID);
        return View(order);
    }

    // POST: Orders/Edit/5
    // To protect from overposting attacks, please enable the specific properties
you want to bind to, for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include = "OrderID,ProductID,CustomerID,Qty")]
Order order)
{
    if (ModelState.IsValid)
    {
        db.Entry(order).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.CustomerID = new SelectList(db.Customers, "CustomerID", "CName",
order.CustomerID);
    ViewBag.ProductID = new SelectList(db.Products, "ProductID", "Pname",
order.ProductID);
    return View(order);
}

    // GET: Orders/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Order order = db.Orders.Find(id);
    if (order == null)
    {
        return HttpNotFound();
    }
    return View(order);
}

    // POST: Orders/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
{
    Order order = db.Orders.Find(id);
    db.Orders.Remove(order);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)

```

```

        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

```

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 shop2.Models;
using PagedList;

namespace shop2.Controllers
{
    public class ProductController : Controller
    {
        private shopdbEntities db = new shopdbEntities();

        // GET: Product
        //public ActionResult Index()
        //{
        //return View(db.Products.ToList());
        //}
        =====sorting filtering=====

        public ActionResult Index(string sortOrder, string currentFilter, string
searchString, int? page)
        {
            ViewBag.CurrentSort = sortOrder;
            ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
            ViewBag.AddressSortParm = sortOrder == "Price" ? "price_desc" : "Price";

            if(searchString !=null)
            {
                page = 1;
            }
            else
            {
                searchString = currentFilter;
            }
            ViewBag.CurrentFilter = searchString;

            var products = from c in db.Products
                          select c;
            =====searchinf stuff
            if (!String.IsNullOrEmpty(searchString))
            {
                products = products.Where(c => c.Pname.Contains(searchString));
            }
        }
}

```

```

//=====
switch (sortOrder)
{
    case "name_desc":
        products = products.OrderByDescending(x => x.Pname);
        break;
    case "Address":
        products = products.OrderBy(s => s.Price);
        break;
    case "address_desc":
        products = products.OrderByDescending(s => s.Price);
        break;
    default:
        products = products.OrderBy(y => y.Pname);
        break;
}
int pageSize = 3;
int pageNumber = (page ?? 1);
return View(products.ToPagedList(pageNumber, pageSize));
// return View(products.ToList());
}

// GET: Product/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Product product = db.Products.Find(id);
    if (product == null)
    {
        return HttpNotFound();
    }
    return View(product);
}

// GET: Product/Create
public ActionResult Create()
{
    return View();
}

// POST: Product/Create
// To protect from overposting attacks, please enable the specific properties
you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "ProductID,Pname,Price")] Product
product)
{
    try
    {
        if (ModelState.IsValid)
        {
            db.Products.Add(product);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }
}

```

```

        catch (DataException)
    {
        ModelState.AddModelError("", "Unable to save changes. Try again, and
if the problem persists see your system administrator.");
    }
    return View(product);
}

// GET: Product/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Product product = db.Products.Find(id);
    if (product == null)
    {
        return HttpNotFound();
    }
    return View(product);
}

// POST: Product/Edit/5
// To protect from overposting attacks, please enable the specific properties
you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "ProductID,Pname,Price")] Product
product)
{
    if (ModelState.IsValid)
    {
        db.Entry(product).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(product);
}

// GET: Product/Delete/5
public ActionResult Delete(int? id, bool? saveChangesError = false)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    //Aline: Updating the Delete Page
    if (saveChangesError.GetValueOrDefault())
    {
        ViewBag.ErrorMessage = "Delete failed. Try again and if the problem
persists see your system administrator.";
    }
    //-----

    Product product = db.Products.Find(id);
    if (product == null)
    {
        return HttpNotFound();
    }
}

```

```
        }
        return View(product);
    }

// POST: Product/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    try
    {
        //Aline: delete operation catches any database update errors
        Product product = db.Products.Find(id);
        db.Products.Remove(product);
        db.SaveChanges();
    }
    catch (DataException)
    {
        return RedirectToAction("Delete", new { id = id, saveChangesError =
true });
    }
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
```

Api Controllers:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Routing;
using shop2.Models;

namespace shop2.Controllers
{
    [RoutePrefix("shop2")]
    public class CustomerAPIController : ApiController
    {
        private shopdbEntities db = new shopdbEntities();

        [Route("api/Customer/all")]
        // GET: api/CustomerAPI = /shop2/api/Customer/all
        public List<Customer> Get()
        {
            db.Configuration.ProxyCreationEnabled = false;
            return db.Customers.OrderByDescending(x => x.CName).ToList();
        }

        [Route("api/customer/ById/{id}")]
        // GET: /shop2//api/customer/ById/2
        public Customer Get(int id)
        {
            db.Configuration.ProxyCreationEnabled = false;
            return db.Customers.SingleOrDefault(x=>x.CustomerID == id);
        }

        [Route("api/customer/ByKeyword/{Keyword}")]
        public IHttpActionResult GetAllcustomersByKeyword(string keyword)
        {
            db.Configuration.ProxyCreationEnabled = false;
            var result = db.Customers.Where(m => m.CName.Contains(keyword)).Select(m => new { m.CustomerID, m.CName });
            if(result.Count()==0)
            {
                return NotFound();
            }
            else
            {
                return Ok(result);
            }
        }

        [Route("api/customer/New")]
        // POST: api/Customer/New
        public HttpResponseMessage Post([FromBody] Customer value)
        {
            if (value == null)
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest, "Failed to read Input");
            }
            if (db.Customers.Count(p => p.CName.Equals(value.CName)) != 0)
            {
                return Request.CreateResponse(HttpStatusCode.OK, "Customer" + value.CName + "Already exists in database.");
            }
        }
    }
}
```

```
        }
        db.Customers.Add(value);
        db.SaveChanges();
        return Request.CreateResponse(HttpStatusCode.Created, "Customer " +
value.CName + "added to database. ");
    }

    // PUT: api/CustomerAPI/5
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace shop2.Controllers
{
    public class OrdersAPIController : ApiController
    {
        // GET: api/OrdersAPI
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET: api/OrdersAPI/5
        public string Get(int id)
        {
            return "value";
        }

        // POST: api/OrdersAPI
        public void Post([FromBody]string value)
        {
        }

        // PUT: api/OrdersAPI/5
        public void Put(int id, [FromBody]string value)
        {
        }

        // DELETE: api/OrdersAPI/5
        public void Delete(int id)
        {
        }
    }
}
```

Customer View:

```
@model shop2.Models.Customer

{@
    ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Customer</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.CName, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CName, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CName, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.CAddress, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CAddress, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CAddress, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Phone, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Phone, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Phone, "", 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>
}

<div>
    @Html.ActionLink("Back to List", "Index")
```

```
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}



---


```
@model shop2.Models.Customer

@{
 ViewBag.Title = "Delete";
}

Delete

@if(created error message)@

@ViewBag.ErrorMessage</p>

Are you sure you want to delete this?

<h4>Customer</h4>
 <hr />
 <dl class="dl-horizontal">
 <dt>
 @Html.DisplayNameFor(model => model.CName)
 </dt>

 <dd>
 @Html.DisplayFor(model => model.CName)
 </dd>

 <dt>
 @Html.DisplayNameFor(model => model.CAddress)
 </dt>

 <dd>
 @Html.DisplayFor(model => model.CAddress)
 </dd>

 <dt>
 @Html.DisplayNameFor(model => model.Phone)
 </dt>

 <dd>
 @Html.DisplayFor(model => model.Phone)
 </dd>
 </dl>
 @using (Html.BeginForm()) {
 @Html.AntiForgeryToken()

 <div class="form-actions no-color">
 <input type="submit" value="Delete" class="btn btn-default" /> |
 @Html.ActionLink("Back to List", "Index")
 </div>
 }
</div>


```


```

```

@model shop2.Models.Customer

{@
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Customer</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.CName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.CName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.CAddress)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.CAddress)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Phone)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Phone)
        </dd>

    </dl>
</div>
@* -----Raf----- *@
<div class="row">
    <div class="col-sm-6">
        <br />
        <dt>
            @Html.DisplayNameFor(model => model.Orders)
        </dt>
        <dd>
            <table class="table">
                <tr>
                    <th>Product Name</th>
                    <th>Quantity</th>
                </tr>
                @foreach (var item in Model.Orders)
                {
                    <tr>
                        <td>
                            @Html.DisplayFor(modelItem => item.Product.Pname)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem => item.Qty)
                        </td>
                    </tr>
                }
            </table>
        </dd>
    </div>
</div>

```

```

        </table>
    </dd>
</div>
<div style="text-align: right;">
    @*
    -----
    *@

<hr />
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.CustomerID }) |
    @Html.ActionLink("Back to List", "Index")
</p>

```

```

@model shop2.Models.Customer

 @{
     ViewBag.Title = "Edit";
 }

<h2>Edit</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Customer</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.CustomerID)

        <div class="form-group">
            @Html.LabelFor(model => model.CName, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CName, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CName, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.CAddress, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CAddress, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CAddress, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Phone, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Phone, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Phone, "", new { @class =
"text-danger" })
            </div>
        </div>
    </div>
}

```

```

        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

@model PagedList.IPagedList<shop2.Models.Customer>
@using PagedList.Mvc;
<link href "~/Content/PagedList.css" rel="stylesheet" type="text/css" />

 @{
    ViewBag.Title = "Customers";
}

<h2>Customers Section</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>

@*== Searching Stuff=====*/
@using (Html.BeginForm("Index", "Customer", FormMethod.Get))
{
    <p>
        Find by Name: @Html.TextBox("SearchString", ViewBag.CurrentFilter as string)
        <input type="submit" value="Search" />
    </p>
}
@*== Searching Stuff=====*/

<table class="table">
    <tr>
        <th>
            @*@Html.DisplayNameFor(model => model.CName) ===Old Code*@
            @Html.ActionLink("Customer Name", "Index", new { sortOrder = ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter })
        </th>
        <th>
            Name
        </th>
        <th>
            @*@Html.DisplayNameFor(model => model.CAddress) ===Old Code*@
            @Html.ActionLink("Address", "Index", new { sortOrder = ViewBag.AddressSortParm, currentFilter = ViewBag.CurrentFilter })
        </th>
    @*<th>
```

```

        @Html.DisplayNameFor(model => model.Phone)
    </th>
    *@
        <th></th>
    </tr>

@foreach (var item in Model)
{
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.CName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.CAddress)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Phone)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.CustomerID }) |
            @Html.ActionLink("Details", "Details", new { id = item.CustomerID }) |
            @Html.ActionLink("Delete", "Delete", new { id = item.CustomerID })
        </td>
    </tr>
}

</table>
Page @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) of @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFilter }))
```

Home Views:

```

    ViewBag.Title = "About";
}
<h2>@ViewBag.Title.</h2>
<h3>@ViewBag.Message</h3>

<div style="float:right">
    <h3>Meet Our Boss</h3>
    
</div>

<p>The Shop Application - video information</p>
<div class="video-container" style="align-content:center">
    <iframe width="560" height="315"
        src="https://www.youtube.com/embed/c3482qAzzLQ" frameborder="0" allow="autoplay,
        encrypted-media" allowfullscreen></iframe>
</div>

@{
    ViewBag.Title = "Contact";
}
<h2>@ViewBag.Title.</h2>
<h3>@ViewBag.Message</h3>
<div style="text-align:center">
    <div>
        <address>
            Institute of Technology, Tallaght<br />
            Dublin 24, D24 FKT9<br />
            <abbr title="Phone">Phone:</abbr>
            425.555.0100
        </address>
        <address>
            <strong>Support:</strong> <a
                href="mailto:Support@example.com">Support@example.com</a><br />
            <strong>Marketing:</strong> <a
                href="mailto:Marketing@example.com">Marketing@example.com</a>
        </address>
    </div>
    <div class="shadow-sm p-3 mb-5 bg-white rounded">
        <div class="google-maps w3-card-4" style="margin-top:10px">
            <iframe
                src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d9539.574048879096!2d-
                6.371330007389621!3d53.291442568108565!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2
                !1s0x48670b4b8ec071f5%3A0xb6cf7501e0d9522a!2sITT+Dublin!5e0!3m2!1sen!2sie!4v1490959511
                226"
                width="600" height="450" frameborder="0" style="border:1px
                allowfullscreen></iframe>
        </div>
        <br>
        <div class="google-maps w3-card-4" style="margin-top:10px">

```

```

        <iframe
src="https://www.google.com/maps/embed?pb=!1m0!3m2!1sen!2sie!4v1491835964991!6m8!1m7!1
sz7Pij7kQQ6u9Lerskk9yA!2m2!1d53.29063724771768!2d-
6.368101423583113!3f107.6877083337143!4f-22.012422274721686!5f0.7820865974627469"
width="600" height="450" frameborder="0" style="border:1px"
allowfullscreen></iframe>
    </div>
</div>
</div>

```

```

@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron rkb" >
    <h1 style="font-family: 'Contrail One', cursive;">The Shop Application</h1>
    <p class="lead">The Shop Application is a free web framework for your product and
customer management.</p>
    <p><a href="https://elearning.it-tallaght.ie/login/index.php" class="btn btn-
primary btn-lg">IT Login &raquo;</a></p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>The Shop App</h2>
        <p>Created By:</p>
        <ul>
            <li style="font-family: 'Lobster', cursive; font-size:large">Aline da
Silva</li>
            <li style="font-family: 'Playball', cursive; font-size:large">Niall
McClelland</li>
            <li style="font-family: 'Aldrich', sans-serif; font-size:large">Rafal
Kita</li>
        </ul>
        <p>Tools & Technology used:</p>
            <a href="https://www.visualstudio.com/downloads/" class="btn btn-light
btn-lg" target="_blank">Microsoft Visual Studio &raquo;</a>
            <a href="https://aws.amazon.com/elasticbeanstalk/" class="btn btn-light
btn-lg" target="_blank">AWS Elastic Beanstalk &raquo;</a>
            <a href="https://aws.amazon.com/rds/" class="btn btn-light btn-lg"
target="_blank">AWS - RDS &raquo;</a>
    </div>
    <div class="col-md-4">
        <h2><cite style="font-family: 'Aldrich', sans-serif">EAD - </cite>Enterprise
Applications Development</h2>
        <p>The Blog We Like</p>
        <ul>
            <li>MVC code-first with related...</li>
            <li>MVC Shopping cart</li>
            <li>Movies 5 - Authentication</li>
            <li>Movies 4 - Client</li>
            <li>Movies 3 - The REST of the CRUD</li>
            <li>Movies 2 - The REST API talking to a db</li>
            <li>Movies 1 - Setting up the Movies db Code-First</li>
        </ul>
        <p><a class="btn btn-default" href="http://dermotb.blogspot.ie/">Learn more
&raquo;</a></p>
    </div>
</div>

```

```

</div>
<div class="col-md-4">
    <h2>Institute of Technology Tallaght</h2>
    <p>
        TThe Mmission of Institute of Technology Tallaght is to be the centre of
        higher education and knowledge creation within South Dublin County
        and its environs, to broaden participation in higher education in the
        region..<br />The institute was established as Tallaght Regional
        Technical College, in 1992, at that time the final new Regional Technical
        College opened in Ireland. It was built on lands donated to the
        Irish government by the Dominicans in Tallaght, for Educational use.
    </p>
    <p><a class="btn btn-default" href="http://www.it-tallaght.ie/">Explore more
&raquo;</a></p>
</div>
</div>

```

```

{@
    ViewBag.Title = "Team";
}

<h2>Meet the Team</h2>

<div class="row">
    <div class="column2">
        <div class="card2">
            <div class="text-center">
                
            </div>
            <div class="container2">
                <h2>Aline da Silva</h2>
                <p class="title2">CEO</p>
                <p>Some text that describes me lorem ipsum ipsum lorem.</p>
                <p>acarmodasilva@yahoo.com.br</p>
                <p><a class="button2" href="https://www.linkedin.com/in/aline-da-
silva-979b8366/" target="_blank">Linkedin </a></p>
            </div>
        </div>
    </div>
</div>

<div class="column2">
    <div class="card2">
        <div class="text-center">
            
        </div>
        <div class="container2">
            <h2>Niall McClelland</h2>
            <p class="title2">CEO</p>
            <p>Some text that describes me lorem ipsum ipsum lorem.</p>
            <p>example@example.com</p>
            <p><a class="button2"
                href="https://en.wikipedia.org/wiki/Homer_Simpson" target="_blank">Linkedin </a></p>
        </div>
    </div>
</div>

```

```
</div>
<div class="column2">
  <div class="card2">
    <div class="text-center">
      
    <div class="img-circle">
    </div>
    <div class="container2">
      <h2>Rafal Kita</h2>
      <p class="title2">CEO</p>
      <p>Some text that describes me lorem ipsum ipsum lorem.</p>
      <p>example@example.com</p>
      <p><a class="button2"
        href="https://en.wikipedia.org/wiki/Bart_Simpson" target="_blank">Linkedin </a></p>
    </div>
  </div>
</div>
</div>
```

Orders View:

```
@model shop2.Models.Order

{@
    ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Order</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.ProductID, "ProductID", htmlAttributes: new
{ @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("ProductID", null, htmlAttributes: new { @class =
"form-control" })
                @Html.ValidationMessageFor(model => model.ProductID, "", new { @class =
= "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.CustomerID, "CustomerID", htmlAttributes:
new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("CustomerID", null, htmlAttributes: new { @class =
"form-control" })
                @Html.ValidationMessageFor(model => model.CustomerID, "", new { @class =
= "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Qty, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Qty, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Qty, "", 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>
}

<div>
    @Html.ActionLink("Back to List", "Index")

```

```

</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}



---


@model shop2.Models.Order

 @{
    ViewBag.Title = "Delete";
}

<h2>Delete</h2>
@*created error message*@
<p class="error">@ViewBag.ErrorMessage</p>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Order</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Qty)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Qty)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Customer.CName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Customer.CName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Product.Pname)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Product.Pname)
        </dd>
    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>

```

```

@model shop2.Models.Order

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Order</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Qty)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Qty)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Customer.CName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Customer.CName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Product.Pname)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Product.Pname)
        </dd>

    </dl>
</div>

<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.OrderID }) |
    @Html.ActionLink("Back to List", "Index")
</p>

```

```

@model shop2.Models.Order

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Order</h4>

```

```

<hr />
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model => model.OrderID)

<div class="form-group">
    @Html.LabelFor(model => model.ProductID, "ProductID", htmlAttributes: new
{ @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ProductID", null, htmlAttributes: new { @class =
"form-control" })
        @Html.ValidationMessageFor(model => model.ProductID, "", new { @class =
= "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.CustomerID, "CustomerID", htmlAttributes:
new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("CustomerID", null, htmlAttributes: new { @class =
"form-control" })
        @Html.ValidationMessageFor(model => model.CustomerID, "", new { @class =
= "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Qty, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Qty, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Qty, "", new { @class =
= "text-danger" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

@model PagedList.IPagedList<shop2.Models.Order>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
@{
    ViewBag.Title = "Orders";
}

<h2>Orders Section</h2>
<p>
    @Html.ActionLink("Create New", "Create")

    @*== Searching Stuff=====*@
    @using (Html.BeginForm("Index", "Customer", FormMethod.Get))
    {
        <p>
            Find by Name: @Html.TextBox("SearchString")
            <input type="submit" value="Search" />
        </p>
    }
    @*== Searching Stuff=====*@

</p>

<table class="table">
    <tr>
        <th>
            @* @Html.DisplayNameFor(model => model.Qty) Old Code*@
            @Html.ActionLink("Quantity", "Index", new { sortOrder =
ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter })
        </th>
        <th>
            @* @Html.DisplayNameFor(model => model.Customer.CName) Old Code*@
            @Html.ActionLink("Customer Name", "Index", new { sortOrder =
ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter })
        </th>
        <th>
            @* @Html.DisplayNameFor(model => model.Product.Pname) *@
            @Html.ActionLink("Product Name", "Index", new { sortOrder =
ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter })
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Qty)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Customer.CName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Product.Pname)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.OrderID }) |

```

```
    @Html.ActionLink("Details", "Details", new { id = item.OrderID })
}
    @Html.ActionLink("Delete", "Delete", new { id = item.OrderID })
</td>
</tr>
}

</table>
Page @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) of @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFilter
}))
```


Product Views:

```
@model shop2.Models.Product

{@
    ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Pname, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Pname, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Pname, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Price, "", 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>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

```

@model shop2.Models.Product

@{
    ViewBag.Title = "Delete";
}

<h2>Delete</h2>
<*/created error message*@
<p class="error">@ViewBag.ErrorMessage</p>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Product</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Pname)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Pname)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Price)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Price)
        </dd>
    </dl>

    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>

```

```

@model shop2.Models.Product

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Product</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Pname)
        </dt>

        <dd>

```

```

        @Html.DisplayFor(model => model.Pname)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Price)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Price)
    </dd>

    </dl>
</div>
<dd>
    [* -----Raf----- *]
    <div class="row">
        <div class="col-sm-6">
            <br />
            <table class="table">
                <tr>
                    <th>Customer</th>
                    <th>Quantity</th>
                </tr>
                @foreach (var item in Model.Orders)
                {
                    <tr>
                        <td>
                            @Html.DisplayFor(modelItem => item.Customer.CName)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem => item.Qty)
                        </td>
                    </tr>
                }
            </table>
        </div>
    </dd>
</dl>
</div>
[* ----- *]
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.ProductID }) |
    @Html.ActionLink("Back to List", "Index")
</p>

```

```

@model shop2.Models.Product

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />

```

```

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

        <div class="form-group">
            @Html.LabelFor(model => model.Pname, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Pname, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Pname, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Price, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Price, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Price, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

@model PagedList.IPagedList<shop2.Models.Product>
@using PagedList.Mvc;
<link href "~/Content/PagedList.css" rel="stylesheet" type="text/css" />

 @{
    ViewBag.Title = "Products";
}

<h2>Products Section</h2>
<p>
    @Html.ActionLink("Create New", "Create")

    @*== Searching Stuff=====*@
    @using (Html.BeginForm("Index", "Product", FormMethod.Get))
    {
        <p>
            Find by Name: @Html.TextBox("SearchString", ViewBag.CurrentFilter as string)
            <input type="submit" value="Search" />
        </p>
    }

```

```

@*== Searching Stuff=====*@



| *@Html.DisplayNameFor(model => model.Pname) Old Code*@         @Html.ActionLink("Product Name", "Index", new { sortOrder = ViewBag.NameSortParm, currentfilter = ViewBag.CurrentFilter }) | *@Html.DisplayNameFor(model => model.Price) Old Code*@         @Html.ActionLink("Price", "Index", new { sortOrder = ViewBag.AddressSortParm, currentfilter = ViewBag.CurrentFilter }) |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                           |                                                                                                                                                                                       |


```

```

</table>
Page @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) of @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFilter }))
```

```

        }

    div.rka {
        font-family: 'Contrail One', cursive;
    }

    div.rkb {
        background: rgba(183,222,237,1);
        background: -moz-linear-gradient(top, rgba(183,222,237,1) 0%, rgba(113,206,239,1) 48%, rgba(33,180,226,1) 49%, rgba(183,222,237,1) 100%);
        background: -webkit-gradient(left top, left bottom, color-stop(0%, rgba(183,222,237,1)), color-stop(48%, rgba(113,206,239,1)), color-stop(49%, rgba(33,180,226,1)), color-stop(100%, rgba(183,222,237,1)));
        background: -webkit-linear-gradient(top, rgba(183,222,237,1) 0%, rgba(113,206,239,1) 48%, rgba(33,180,226,1) 49%, rgba(183,222,237,1) 100%);
        background: -o-linear-gradient(top, rgba(183,222,237,1) 0%, rgba(113,206,239,1) 48%, rgba(33,180,226,1) 49%, rgba(183,222,237,1) 100%);
        background: -ms-linear-gradient(top, rgba(183,222,237,1) 0%, rgba(113,206,239,1) 48%, rgba(33,180,226,1) 49%, rgba(183,222,237,1) 100%);
        background: linear-gradient(to bottom, rgba(183,222,237,1) 0%, rgba(113,206,239,1) 48%, rgba(33,180,226,1) 49%, rgba(183,222,237,1) 100%);
        filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#b7deed', endColorstr='#b7deed', GradientType=0 );
        font-family: 'Contrail One', cursive;
    }

    div.rkc {
        background: rgba(182,22,147,1);
        background: -moz-linear-gradient(top, rgba(182,22,147,1) 0%, rgba(157,21,121,1) 50%, rgba(112,0,79,1) 51%, rgba(189,0,126,1) 100%);
        background: -webkit-gradient(left top, left bottom, color-stop(0%, rgba(182,22,147,1)), color-stop(50%, rgba(157,21,121,1)), color-stop(51%, rgba(112,0,79,1)), color-stop(100%, rgba(189,0,126,1)));
        background: -webkit-linear-gradient(top, rgba(182,22,147,1) 0%, rgba(157,21,121,1) 50%, rgba(112,0,79,1) 51%, rgba(189,0,126,1) 100%);
        background: -o-linear-gradient(top, rgba(182,22,147,1) 0%, rgba(157,21,121,1) 50%, rgba(112,0,79,1) 51%, rgba(189,0,126,1) 100%);
        background: -ms-linear-gradient(top, rgba(182,22,147,1) 0%, rgba(157,21,121,1) 50%, rgba(112,0,79,1) 51%, rgba(189,0,126,1) 100%);
        background: linear-gradient(to bottom, rgba(182,22,147,1) 0%, rgba(157,21,121,1) 50%, rgba(112,0,79,1) 51%, rgba(189,0,126,1) 100%);
        filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#b61693', endColorstr='#bd007e', GradientType=0 );
    }

    .footer {
        position: fixed;
        left: 0;
        bottom: 0;
        width: 100%;
        background-color: rgba(010,010,255,0.1);
        color: brown;
        text-align: center;
    }
    /*????????????????????????????????????????????????*/
}

html {
    box-sizing: border-box;
}

*, *:before, *:after {
    box-sizing: inherit;
}

```

```

.column2 {
  float: left;
  width: 33.3%;
  margin-bottom: 16px;
  padding: 0 8px;
}

@@media screen and (max-width: 650px) {
  .column2 {
    width: 100%;
    display: block;
  }
}

.card2 {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
}

.container2 {
  padding: 0 16px;
}

.container2::after, .row::after {
  content: "";
  clear: both;
  display: table;
}

.title2 {
  color: grey;
}

.button2 {
  border: 1px solid grey;
  outline: 0;
  display: inline-block;
  padding: 8px;
  color: black;
  background-color: white;
  text-align: center;
  cursor: pointer;
  width: 100%;
  border-radius: 6px;
}

.button2:hover {
  background-color: #555;
}

</style>
@*
-----*@
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top rkc">
    <div class="container">
      <div class="navbar-header" >
        <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
          <span class="icon-bar"></span>

```

```

        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <div class="rka">
        @Html.ActionLink("The Shop Application", "Index", "Home", new {
area = "" }, new { @class = "navbar-brand" })
    </div>
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("Customers", "Index", "Customer")</li>
        <li>@Html.ActionLink("Products", "Index", "Product")</li>
        <li>@Html.ActionLink("Orders", "Index", "Orders")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Team", "Team", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
    </ul>
</div>
</div>
<div class="container body-content">
    @RenderBody()
    <hr />
    <div class="footer">
        <p style="font-family: 'Lobster', cursive;">
            &copy; @DateTime.Now.Year - The Shop Application
&nbsp;&nbsp;&nbsp;&nbsp;#1045; Aline da Silva &#1045; Niall McClelland &#1045;;
Rafal Kita &nbsp;&nbsp;&nbsp;&nbsp;&#9731;
        </p>
    </div>
</div>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>

```

Web Config:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=301880
  -->
<configuration>
  <configSections>

    <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
    Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
    requirePermission="false" />
    <!-- For more information on Entity Framework configuration, visit
    http://go.microsoft.com/fwlink/?LinkId=237468 --><!-- For more information on Entity
    Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=237468 --
  </configSections>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.1" />
    <httpRuntime targetFramework="4.6.1" />
    <httpModules>
      <add name="ApplicationInsightsWebTracking"
      type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule,
      Microsoft.AI.Web" />
    </httpModules>
  </system.web>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" culture="neutral"
publicKeyToken="30ad4fe6b2a6aeed" />
        <bindingRedirect oldVersion="0.0.0.0-11.0.0.0" newVersion="11.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Optimization"
publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="0.0.0.0-1.6.5135.21930"
newVersion="1.6.5135.21930" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35"
/>
        <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35"
/>
        <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>

```

```

        <bindingRedirect oldVersion="0.0.0.0-5.2.6.0" newVersion="5.2.6.0" />
    </dependentAssembly>
    <dependentAssembly>
        <assemblyIdentity name="Antlr3.Runtime" publicKeyToken="eb42632606e9261f"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-3.5.0.2" newVersion="3.5.0.2" />
    </dependentAssembly>
</assemblyBinding>
</runtime>
<system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <modules>
        <remove name="TelemetryCorrelationHttpModule" />
        <add name="TelemetryCorrelationHttpModule"
type="Microsoft.AspNet.TelemetryCorrelation.TelemetryCorrelationHttpModule,
Microsoft.AspNet.TelemetryCorrelation" precondition="integratedMode,managedHandler" />
        <remove name="ApplicationInsightsWebTracking" />
        <add name="ApplicationInsightsWebTracking"
type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule,
Microsoft.AI.Web" precondition="managedHandler" />
    </modules>
    <handlers>
        <remove name="ExtensionlessUrlHandler-Integrated-4.0" />
        <remove name="OPTIONSVerbHandler" />
        <remove name="TRACEVerbHandler" />
        <add name="ExtensionlessUrlHandler-Integrated-4.0" path="*.*" verb="*"
type="System.Web.Handlers.TransferRequestHandler"
preCondition="integratedMode,runtimeVersionv4.0" />
    </handlers></system.webServer>
<system.codedom>
    <compilers>
        <compiler language="c#;cs;cssharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.8.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
        <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.8.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008 /define:_MYTYPE="Web";
optionInfer+" />
    </compilers>
</system.codedom>
<entityFramework>
    <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
    <providers>
        <provider invariantName="System.Data.SqlClient"
type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
</entityFramework>
<connectionStrings>
    <add name="CustomersContext"
connectionString="metadata=res://*/Models.shop2Model.csdl|res://*/Models.shop2Model.ss
dl|res://*/Models.shop2Model.msl;provider=System.Data.SqlClient;provider connection
string="data source=shopdb.c79di6yegnby.eu-west-1.rds.amazonaws.com,1433;initial
catalog=shopdb;user
id=shopuser;password=shopuser123;MultipleActiveResultSets=True;App=EntityFramework&quo
t;" providerName="System.Data.EntityClient" />
    <add name="shopdbEntities"
connectionString="metadata=res://*/Models.shop2Model.csdl|res://*/Models.shop2Model.ss

```

```
dl|res://*/Models.shop2Model.msl;provider=System.Data.SqlClient;provider connection  
string="data source=shopdb.c79di6yegnby.eu-west-1.rds.amazonaws.com,1433;initial  
catalog=shopdb;user  
id=shopuser;password=shopuser123;MultipleActiveResultSets=True;App=EntityFramework&quo  
t;" providerName="System.Data.EntityClient" />  
</connectionStrings>  
</configuration>
```

Shop2 Testing :

TestCustomerController:

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using shop2.Controllers;
using shop2.Models;
using System.Web.Mvc;
using System;
using System.Collections.Generic;

namespace shop2Tests2
{
    [TestClass]
    public class TestCustomerController
    {

        [TestMethod()]
        public void CreateCustomerTest()
        {
            TestCustomersContext C = new TestCustomersContext();
            CustomerController controller = new CustomerController(C);
            Customer customerToCreate = new Customer() { CustomerID = 13, CName =
"Bob", CAddress = "Tallaght", Phone = "08744458" };

            var result = controller.Create(customerToCreate) as RedirectToRouteResult;

            Assert.AreEqual("Index", result.RouteValues["action"]);
        }

        [TestMethod()]
        public void IndexTestSpecficWord()
        {

            TestCustomersContext c = new TestCustomersContext();

            CustomerController controller = new CustomerController(c);

            Customer Customer1 = new Customer() { CName = "Bob", CAddress = "Dublin" ,
};

            c.Customers.Add(Customer1);

            ViewResult result = CustomerController.Index("", "Dublin") as ViewResult;

            var customers = (List<Customer>)result.ViewData.Model;

            foreach (var r in customers)

            {

                StringAssert.Contains(r.CAddress, "Dublin");

            }

            Assert.AreEqual(1, customers.Count);
        }
    }
}
```

```
    }

}
```

TestCustomerDbSet

```
using shop2.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace shop2Tests2
{
    class TestCustomerDbSet : TestDbSet<Customer>
    {
        public override Customer Find(params object[] keyValues)
        {
            return this.SingleOrDefault(customer => customer.CustomerID ==
(int)keyValues.Single());
        }
    }
}
```

TestControllerContext

```
using shop2.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.Entity;

namespace shop2Tests2
{
    public class TestCustomersContext : ICustomersContext
    {
        public TestCustomersContext()
        {
            this.Customers = new TestCustomerDbSet();
        }

        public DbSet<Customer> Customers { get; set; }

        public int SaveChanges()
        {
            return 0;
        }

        public void MarkAsModified(Object item) { }
        public void Dispose() { }
    }
}
```

TestDbSet

```
using System;
```

```
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Data.Entity;
using System.Linq;

namespace shop2Tests2
{
    public class TestDbSet<T> : DbSet<T>, IQueryable, IEnumerable<T>
        where T : class
    {
        ObservableCollection<T> _data;
        IQueryable _query;

        public TestDbSet()
        {
            _data = new ObservableCollection<T>();
            _query = _data.AsQueryable();
        }

        public override T Add(T item)
        {
            _data.Add(item);
            return item;
        }

        public override T Remove(T item)
        {
            _data.Remove(item);
            return item;
        }

        public override T Attach(T item)
        {
            _data.Add(item);
            return item;
        }

        public override T Create()
        {
            return Activator.CreateInstance<T>();
        }

        public override TDerivedEntity Create<TDerivedEntity>()
        {
            return Activator.CreateInstance<TDerivedEntity>();
        }

        public override ObservableCollection<T> Local
        {
            get { return new ObservableCollection<T>(_data); }
        }

        Type IQueryable.ElementType
        {
            get { return _query.ElementType; }
        }

        System.Linq.Expressions.Expression IQueryable.Expression
        {
            get { return _query.Expression; }
        }
    }
}
```

```
IQueryProvider IQuerybable.Provider
{
    get { return _query.Provider; }
}

System.Collections.IEnumerator System.Collections.IEnumerable.GetEnumerator()
{
    return _data.GetEnumerator();
}

IEnumerator<T> IEnumerable<T>.GetEnumerator()
{
    return _data.GetEnumerator();
}
}

}
```

Models for Testing:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;

namespace shop2.Models
{
    public class CustomersContext : DbContext, ICustomersContext
    {
        public CustomersContext() : base("name=CustomersContext")
        {
        }
        public DbSet<Customer> Customers { get; set; }
        public void MarkAsModified(Object item)
        {
            Entry(item).State = EntityState.Modified;
        }
    }
}

using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;

namespace shop2.Models
{
    public interface ICustomersContext : IDisposable
    {
        DbSet<Customer> Customers { get; }
        int SaveChanges();
        void MarkAsModified(Object item);
    }
}
```

Shop2 Client Console:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net.Http;
using System.Net.Http.Headers;

namespace shop2Client
{
    class Program
    {
        static void Main(string[] args)
        {
            RunAsync().Wait();
        }
    }
}
```

```

        }

    public static async Task RunAsync() // async methods return Task or Task<T>
    {
        try
        {
            using (HttpClient client = new HttpClient())
            {
                client.BaseAddress = new Uri("http://localhost:51931/shop2/");
                //replace http://localhost:51931 with the address returned when
you run with Microsoft Edge

                client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));

                // Test get all Customers
                Console.WriteLine("All Customers in database by descending
Customer ID:");
                HttpResponseMessage response = await
client.GetAsync("api/Customer/all"); //customers/all is the route defined to reach
the method GetAllCustomers() in the Web Api project
                if (response.IsSuccessStatusCode)
                {
                    // read results
                    var customer = await
response.Content.ReadAsAsync<List<Customer>>(); //GetAllCustomers() returns a list of
Customers
                    foreach (var customers in customer)
                    {
                        Console.WriteLine(customers);
                    }
                }
                else
                {
                    Console.WriteLine(response.StatusCode + " " +
response.ReasonPhrase);
                }
            }

            // Test get customer with ID 1
            Console.WriteLine("Customer 1:");
            response = await client.GetAsync("api/customer/ById/2");
            if (response.IsSuccessStatusCode)
            {
                // read results
                var customer = await response.Content.ReadAsAsync<Customer>();
                Console.WriteLine(customer);
            }
            else
            {
                Console.WriteLine(response.StatusCode + " " +
response.ReasonPhrase);
            }
        }

        // Test get Customer IDs and Name for specified keyword
        Console.WriteLine("Customers containing Dave in name:");
        response = await client.GetAsync("api/customer/ByKeyword/Dave");
        if (response.IsSuccessStatusCode)
        {
            // read results
            var customers = await response.Content.ReadAsStringAsync();
            foreach (var m in customers)

```

```

        {
            Console.WriteLine(m);
        }
    }
else
{
    Console.WriteLine(response.StatusCode + " " +
response.ReasonPhrase);
}

// Test POST
Console.WriteLine("Create new Customer");
Customer newCustomer = new Customer() { CustomerID = 6, CName =
"Alex Dunne", CAddress = "26 Meadow View, Raheny, Dublin", Phone = "0876589995" };
response = await client.PostAsJsonAsync("api/customer/New",
newCustomer);
if (response.IsSuccessStatusCode)
{
    Console.WriteLine("Customer {0} added", newCustomer.CName);
}
else
{
    Console.WriteLine(response.StatusCode + " " +
response.ReasonPhrase);
}
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
Console.ReadLine();

}
}

```