

Assignment - Part 2

Submission Date ____Friday 29th Nov 2019____

Question – ASP.NET Core Web Application Development (50%)

An animal sanctuary called Animal Lifeline wants to make available via a web application details of animals that it currently has available for the public to foster. Typical information it might hold regarding each animal would be:

- Animal name
- Type of animal
- Breed
- Gender
- Date of birth
- Description/characteristics
- Image of the animal

You are required to develop an ASP.NET Core web-based application that facilitates creating, displaying, editing and deleting individual animal records.

You should take a software layered approach to developing your application, with a supporting database for a service layer and an ASP.NET Core MVC web layer. The services should be unit tested before being used by the MVC web layer. The MVC web layer acts as the user interface for the application. Ideally the MVC web layer should be protected by a login facility with only authenticated users being authorised to modify any of the data accessible to the user. User input should be validated.

You should present the following as part of your assignment submission:

- (a) C# class model for an Animal with a snapshot of the database (10 marks)
- (b) Code for the services developed with associated unit testing code (10 marks)
- (c) Code for the Animal *Controller* (5 marks)
- (d) Code for each of the MVC Animal views (5 marks)
- (e) Snapshots of user testing the MVC web application (10 marks)
- (f) Demonstration of your software in the laboratory. (10 marks)

Provide a rationale covering your design decisions associated with developing the application and comment on any innovative techniques or limitations in your application.

Note. This assignment should be named “Assignment Part 2” and uploaded as a PDF to Blackboard under “COM580 – Enterprise Computing”.)

END

Contents

1.C# class model for an Animal with a snapshot of the database	3
Animal Class	3
DB SQL.....	3
DbContext	4
DB Design	4
DB Content.....	5
2.Code for the services developed with associated unit testing code	5
Animal Data Service Interface.....	5
Animal Data Service	5
Animal Service Tests	6
3.Code for the Animal <i>Controller</i>	9
Animal Controller	9
4.Code for each of the MVC Animal views	11
Index View.....	11
Create View	12
Delete View	13
Details View	14
Edit View	15
5.Snapshots of user testing the MVC web application	17
Home View UI	17
Register View UI.....	18
Login View UI	19
Index View UI	20
Create View UI	21
Details View UI.....	22
Edit View UI.....	23
Delete View UI	24

1.C# class model for an Animal with a snapshot of the database

Animal Class

```
using System;
using System.Collections.Generic;
using System.Text;
using System.ComponentModel.DataAnnotations;

namespace Assignment2Data.Animal
{
    public class Animals
    {
        [Key]
        public int AniId { get; set; }

        [DataType(DataType.Text)]
        [Required(ErrorMessage = "Animal name Required")]
        public string AniName { get; set; }

        [DataType(DataType.Text)]
        [Required(ErrorMessage = "Animal type Required")]
        public string AniType { get; set; }

        [DataType(DataType.Text)]
        [Required(ErrorMessage = "Animal breed Required")]
        public string AniBreed { get; set; }

        [DataType(DataType.Date)]
        [Required(ErrorMessage = "Animal Date of Birth Required")]
        public DateTime AniDoB { get; set; }

        [DataType(DataType.Text)]
        [Required(ErrorMessage = "Animal Description Required")]
        public string AniDesc { get; set; }

        [DataType(DataType.Text)]
        [Required(ErrorMessage = "Animal Picture Required")]
        public string AniPic { get; set; }
    }
}
```

DB SQL


```
CREATE TABLE [dbo].[Animal] (
    [AniId] INT IDENTITY (1, 1) NOT NULL,
    [AniName] NVARCHAR (MAX) NOT NULL,
    [AniType] NVARCHAR (MAX) NOT NULL,
    [AniBreed] NVARCHAR (MAX) NOT NULL,
    [AniDoB] DATETIME2 (7) NOT NULL,
    [AniDesc] NVARCHAR (MAX) NOT NULL,
    [AniPic] NVARCHAR (MAX) NOT NULL,
    CONSTRAINT [PK_Animal] PRIMARY KEY CLUSTERED ([AniId] ASC)
);
```

DbContext

```
using System;
using System.Collections.Generic;
using System.Text;
using Assignment2Data.Animal;
using Microsoft.EntityFrameworkCore;

namespace Assignment2Data.Repository
{
    public class AnimalDbContext : DbContext
    {
        public AnimalDbContext()
        {
        }
        public AnimalDbContext(DbContextOptions<AnimalDbContext> options)
        : base(options)
        {
        }
        public DbSet<Animals> Animal { get; set; }
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=Assignment2;Trusted_Connection = True;");
        }
        public void Initialise()
        {
            Database.EnsureDeleted();
            Database.EnsureCreated();
        }
    }
}
```

DB Design

	Name	Data Type	Allow Nulls	Default
	AniId	int	<input type="checkbox"/>	
	AniName	nvarchar(MAX)	<input type="checkbox"/>	
	AniType	nvarchar(MAX)	<input type="checkbox"/>	
	AniBreed	nvarchar(MAX)	<input type="checkbox"/>	
	AniDoB	datetime2(7)	<input type="checkbox"/>	
	AniDesc	nvarchar(MAX)	<input type="checkbox"/>	
	AniPic	nvarchar(MAX)	<input type="checkbox"/>	
			<input type="checkbox"/>	

DB Content

	AniId	AniName	AniType	AniBreed	AniDoB	AniDesc	AniPic
▶	1	Rover	Dog	Alsatian	23/09/2016 12:0...	Brown and Blac...	https://www.p...
	2	Socks	Cat	Shorthair	23/09/2016 12:0...	Black body, Wh...	https://encrypt...
	3	Nibbles	Ferret	Black Sable	23/09/2016 12:0...	Knows several t...	https://www.co...
	5	Flopsy	Rabbit	Mini Lop	23/09/2016 12:0...	Will always co...	https://upload....
	7	Fly	Dog	German Shorth...	23/09/2016 12:0...	Very well traine...	https://i.pinim...
	8	Monty	Snake	Burmese Python	23/09/2016 12:0...	Likes to be stro...	https://www.na...
⚙	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.Code for the services developed with associated unit testing code

Animal Data Service Interface

```
using System;
using System.Collections.Generic;
using System.Text;
using Assignment2Data.Animal;

namespace Assignment2Data.Services
{
    public interface IAnimalDataService
    {
        void Initialise();
        IList<Animals> SelectAll();
        Animals SelectByID(int id);
        Animals Insert(Animals obj);
        void Update(Animals obj);
        bool Delete(int id);
    }
}
```

Animal Data Service

```
using System;
using System.Collections.Generic;
using System.Text;
using Assignment2Data.Animal;
using Assignment2Data.Repository;
using Microsoft.EntityFrameworkCore;
using System.Linq;

namespace Assignment2Data.Services
{
    public class AnimalDataService : IAnimalDataService
    {
        private AnimalDBContext db = null;
        public AnimalDataService()
        {
            this.db = new AnimalDBContext();
        }
        public AnimalDataService(AnimalDBContext db)
        {
            this.db = db;
        }
    }
}
```

```

    public void Initialise()
    {
        db.Initialise();
    }
    public IList<Animals> SelectAll()
    {
        return db.Animal.OrderBy(a => a.AnidoB).ToList();
    }
    public Animals SelectByID(int id)
    {
        return db.Animal.FirstOrDefault(c => c.Anid == id);
    }
    public Animals Insert(Animals obj)
    {
        db.Animal.Add(obj);
        db.SaveChanges();
        return obj;
    }
    public void Update(Animals obj)
    {
        db.Entry(obj).State = EntityState.Modified;
        db.SaveChanges();
    }
    public bool Delete(int id)
    {
        var existing = db.Animal.FirstOrDefault(c => c.Anid == id);
        if (existing == null)
        {
            return false;
        }
        db.Remove(existing);
        db.SaveChanges();
        return true;
    }
}
}

```

Animal Service Tests

```

using System;
using Xunit;
using System.Linq;
using Assignment2Data.Services;
using Assignment2Data.Animal;

namespace Assignment2Test
{
    public class AnimalServiceTests
    {
        private IAnimalDataService service;

        public AnimalServiceTests()
        {
            service = new AnimalDataService();
        }
    }
}

```

```

[Fact]
public void TestSelectAll()
{
    ServiceSeeder.Seed(service);

    var animals = service.SelectAll();
    var animalCount = animals.Count();

    Assert.Equal(2, animalCount);
}

[Fact]
public void TestInsert()
{
    ServiceSeeder.Seed(service);

    var animals = service.SelectAll();
    var beforeCount = animals.Count();

    var test = service.Insert(new Animals
    {
        AniName = "test",
        AniType = "test",
        AniBreed = "test",
        AniDoB = DateTime.Parse("01/01/2000"),
        AniDesc = "test",
        AniPic = "test"
    });

    animals = service.SelectAll();
    var afterCount = animals.Count();

    Assert.Equal(beforeCount+1, afterCount);
}

[Fact]
public void TestDelete()
{
    ServiceSeeder.Seed(service);

    var animals = service.SelectAll();
    var beforeCount = animals.Count();

    service.Delete(2);

    animals = service.SelectAll();
    var afterCount = animals.Count();

    Assert.Equal(beforeCount - 1, afterCount);
}

```

```

[Fact]
public void TestUpdate()
{
    ServiceSeeder.Seed(service);

    var beforeName = service.SelectByID(1);

    beforeName.AniName = "Test";

    service.Update(beforeName);

    var afterName = service.SelectByID(1);

    Assert.Equal("Test", afterName.AniName);
}

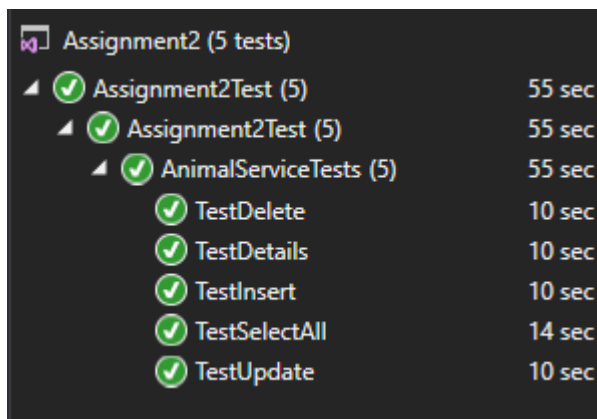
[Fact]
public void TestDetails()
{
    ServiceSeeder.Seed(service);

    var before = service.SelectByID(1);

    var after = service.SelectByID(1);

    Assert.Equal(before, after);
}
}
}

```



The screenshot shows a test runner interface with a dark background. It displays a tree view of test results for 'Assignment2 (5 tests)'. The tree is expanded, showing 'Assignment2Test (5)' with a green checkmark, which is further expanded to show 'AnimalServiceTests (5)' with a green checkmark. Under 'AnimalServiceTests', five individual tests are listed, each with a green checkmark and a duration: 'TestDelete' (10 sec), 'TestDetails' (10 sec), 'TestInsert' (10 sec), 'TestSelectAll' (14 sec), and 'TestUpdate' (10 sec).

Test Name	Duration
Assignment2 (5 tests)	55 sec
Assignment2Test (5)	55 sec
AnimalServiceTests (5)	55 sec
TestDelete	10 sec
TestDetails	10 sec
TestInsert	10 sec
TestSelectAll	14 sec
TestUpdate	10 sec

3.Code for the *Animal Controller*

Animal Controller

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Assignment2Data.Animal;
using Assignment2Data.Services;
using Microsoft.AspNetCore.Authorization;

namespace Assignment2WebMVC.Controllers
{
    [Authorize]
    public class AnimalController : Controller
    {
        private IAnimalDataService service;
        public AnimalController()
        {
            service = new AnimalDataService();
        }
        [HttpGet]
        public IActionResult Index()
        {
            List<Animals> animal = (List<Animals>)service.SelectAll();
            return View(animal);
        }
        [HttpGet]
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Create(Animals obj)
        {
            if (ModelState.IsValid)
            { // check valid state
                service.Insert(obj);
                return RedirectToAction("Index");
            }
            else // not valid so redisplay
            {
                return View(obj);
            }
        }
    }
}
```

```

[HttpGet, ActionName("Delete")]
public IActionResult Delete(int id)

{
    Animals existing = service.SelectByID(id);
    return View(existing);
}
[HttpPost, ActionName("Delete")]
public IActionResult ConfirmDelete(int id)
{
    service.Delete(id);
    return RedirectToAction("Index");
}
[HttpGet]
public ActionResult Edit(int id)
{
    Animals existing = service.SelectByID(id);
    return View(existing);
}
[HttpPost, ActionName("Edit")]
public ActionResult ConfirmEdit(Animals obj)
{
    if (ModelState.IsValid)
    { // check valid state
        service.Update(obj);
        return RedirectToAction("Index");
    }
    else // not valid so redisplay
    {
        return View(obj);
    }
}
[HttpGet]
public ActionResult Details(int id)
{
    Animals existing = service.SelectByID(id);
    return View(existing);
}
}
}

```

4.Code for each of the MVC Animal views

Index View

```
@model IEnumerable<Assignment2Data.Animal.Animals>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                Name
            </th>
            <th>
                Species
            </th>
            <th>
                Breed
            </th>
            <th>
                Date of Birth
            </th>
            <th>
                Description
            </th>
            <th>
                Picture
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model) {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.AniParam)
                </td>
            </tr>
        }
    </tbody>
</table>
```

```

        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.AniId }) |
            @Html.ActionLink("Details", "Details", new { id = item.AniId }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.AniId})
        </td>
    </tr>
}
</tbody>
</table>

```

Create View

```
@model Assignment2Data.Animal.Animals
```

```

@{
    ViewData["Title"] = "Create";
}

```

```

<h1>Create</h1>

<h4>Animals</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="AniName" class="control-label"></label>
                <input asp-for="AniName" class="form-control" />
                <span asp-validation-for="AniName" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniType" class="control-label"></label>
                <input asp-for="AniType" class="form-control" />
                <span asp-validation-for="AniType" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniBreed" class="control-label"></label>
                <input asp-for="AniBreed" class="form-control" />
                <span asp-validation-for="AniBreed" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniDoB" class="control-label"></label>
                <input asp-for="AniDoB" class="form-control" />
                <span asp-validation-for="AniDoB" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniDesc" class="control-label"></label>
                <input asp-for="AniDesc" class="form-control" />
                <span asp-validation-for="AniDesc" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniPic" class="control-label"></label>
                <input asp-for="AniPic" class="form-control" />
                <span asp-validation-for="AniPic" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>

```

```

</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

Delete View

```

@model Assignment2Data.Animal.Animals

@{
    ViewData["Title"] = "Delete";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Animals</h4>
    <hr />
    <dl class="row">
        <dt class="col-sm-2">
            Animal ID
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniId)
        </dd>
        <dt class="col-sm-2">
            Name
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniName)
        </dd>
        <dt class="col-sm-2">
            Species
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniType)
        </dd>
        <dt class="col-sm-2">
            Breed
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniBreed)
        </dd>
        <dt class="col-sm-2">
            Date Of Birth
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniDoB)
        </dd>
        <dt class="col-sm-2">
            Description
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AniDesc)
        </dd>
        <dt class="col-sm-2">

```

```

        Picture
    </dt>
    <dd class = "col-sm-10">
        
    </dd>
</dl>

<form asp-action="Delete">
    <input type="submit" value="Delete" class="btn btn-danger" /> |
    <a asp-action="Index">Back to List</a>
</form>
</div>

```

Details View

```
@model Assignment2Data.Animal.Animals
```

```

@{
    ViewData["Title"] = "Details";
}

<h1>Details</h1>

<div>
    <h4>Animals</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            Animal ID
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniId)
        </dd>
        <dt class = "col-sm-2">
            Name
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniName)
        </dd>
        <dt class = "col-sm-2">
            Species
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniType)
        </dd>
        <dt class = "col-sm-2">
            Breed
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniBreed)
        </dd>
        <dt class = "col-sm-2">
            Date Of Birth
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniDoB)
        </dd>
        <dt class = "col-sm-2">
            Description
        </dt>
    </dl>

```

```

        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.AniDesc)
        </dd>
        <dt class = "col-sm-2">
            Picture
        </dt>
        <dd class = "col-sm-10">
            
        </dd>
    </dl>
</div>
<div>
    @Html.ActionLink("Edit", "Edit", new { id = Model.AniId }) |
    <a asp-action="Index">Back to List</a>
</div>

```

Edit View

@model Assignment2Data.Animal.Animals

```

@{
    ViewData["Title"] = "Edit";
}
<h1>Edit</h1>

<h4>Animals</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="AniId" class="control-label">ID</label>
                <input asp-for="AniId" class="form-control" readonly="readonly" />
                <span asp-validation-for="AniId" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniName" class="control-label">Name</label>
                <input asp-for="AniName" class="form-control" />
                <span asp-validation-for="AniName" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniType" class="control-label">Species</label>
                <input asp-for="AniType" class="form-control" />
                <span asp-validation-for="AniType" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniBreed" class="control-label">Breed</label>
                <input asp-for="AniBreed" class="form-control" />
                <span asp-validation-for="AniBreed" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniDoB" class="control-label">Date of Birth</label>
                <input asp-for="AniDoB" class="form-control" />
                <span asp-validation-for="AniDoB" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="AniDesc" class="control-label">Description</label>
                <input asp-for="AniDesc" class="form-control" />
                <span asp-validation-for="AniDesc" class="text-danger"></span>
            </div>
        </form>
    </div>

```

```

        <div class="form-group">
            <label asp-for="AniPic" class="control-label">Picture URL</label>
            <input asp-for="AniPic" class="form-control" />
            <span asp-validation-for="AniPic" class="text-danger"></span>
        </div>
        <div class="form-group">
            <input type="submit" value="Save" class="btn btn-primary" />
        </div>
    </form>
</div>

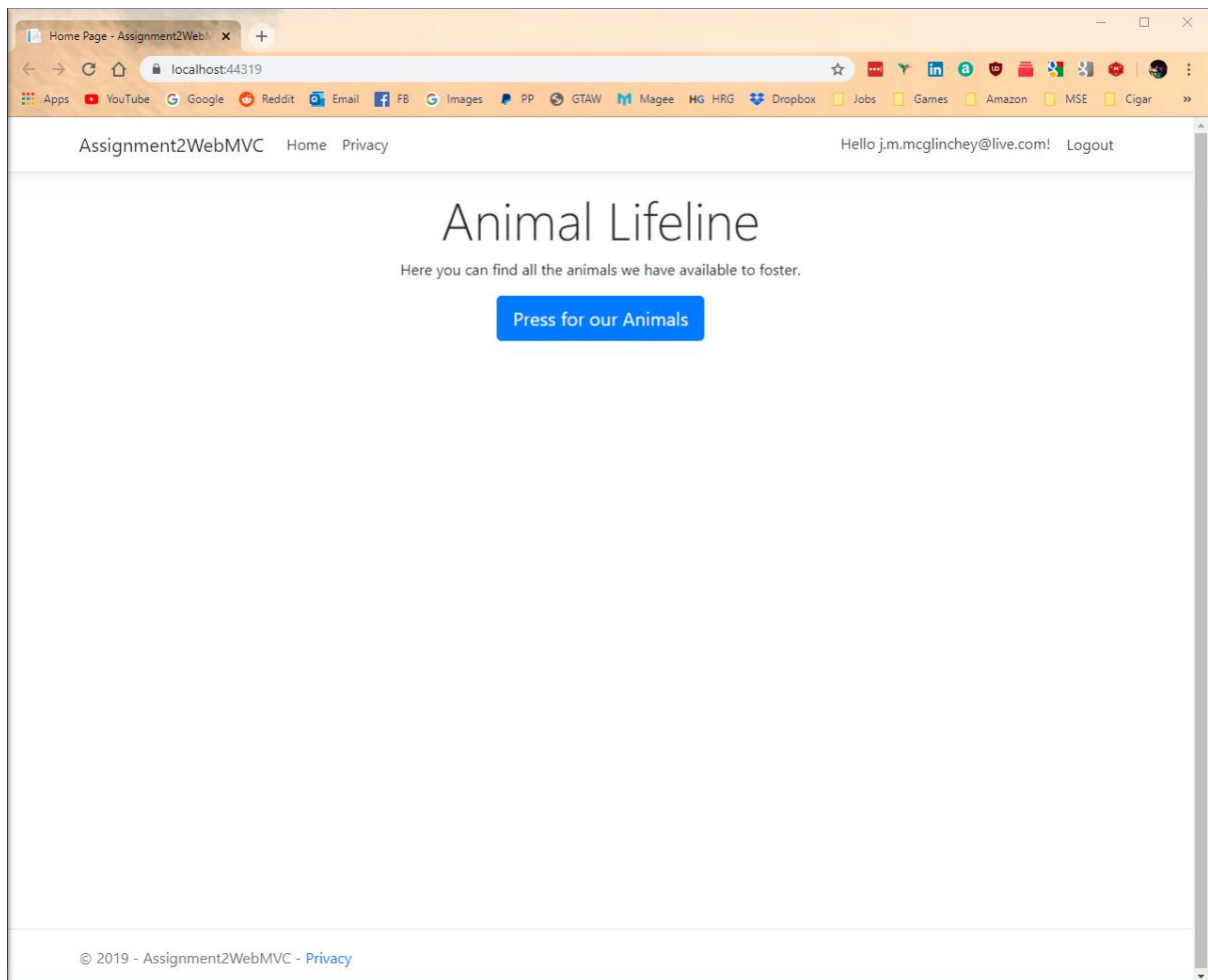
<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```


5.Snapshots of user testing the MVC web application

Home View UI



Register View UI

The screenshot shows a web browser window with the URL `localhost:44319/identity/Account/Register?returnUrl=%2FAnimal%3F`. The browser's address bar and tabs are visible at the top. The page content is as follows:

Assignment2WebMVC Home Privacy Register Login

Register

Create a new account.

Email

Password

Confirm password

[Register](#)

© 2019 - Assignment2WebMVC - [Privacy](#)

Login View UI

Log in - Assignment2WebMVC x Assignment(19-20) Part2.docx x +

localhost:44319/identity/Account/Login?ReturnUrl=%2FAnimal%3F

Apps YouTube Google Reddit Email FB Images PP GTAW Magee HG HRG Dropbox Jobs Games Amazon MSE Cigar »

Assignment2WebMVC Home Privacy Register Login

Log in

Use a local account to log in.

Email

Password

☐ Remember me?

[Log in](#)

[Forgot your password?](#)

[Register as a new user](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.



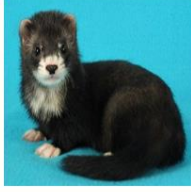
© 2019 - Assignment2WebMVC - [Privacy](#)

Index View UI

Assignment2WebMVC Home Privacy Hello j.m.mcglinchey@live.com! Logout

Index

[Create New](#)

Name	Species	Breed	Date of Birth	Description	Picture	
Rover	Dog	Alsation	23/09/2016	Brown and Black, Friendly		Edit Details Delete
Socks	Cat	Shorthair	23/09/2016	Black body, White feet		Edit Details Delete
Nibbles	Ferret	Black Sable Mitt	23/09/2016	Knows several tricks		Edit Details Delete

Create View UI

Assignment2WebMVC Home Privacy Hello j.m.mcglinchey@live.com! Logout

Create Animals

Name

Species

Breed

Date of Birth

Description

Picture URL

[Create](#)

[Back to List](#)

© 2019 - Assignment2WebMVC - [Privacy](#)

Assignment2WebMVC Home Privacy Hello j.m.mcglinchey@live.com! Logout

Create Animals

Name
Animal name Required

Species
Animal type Required

Breed
Animal breed Required

Date of Birth
Animal Date of Birth Required

Description
Animal Description Required

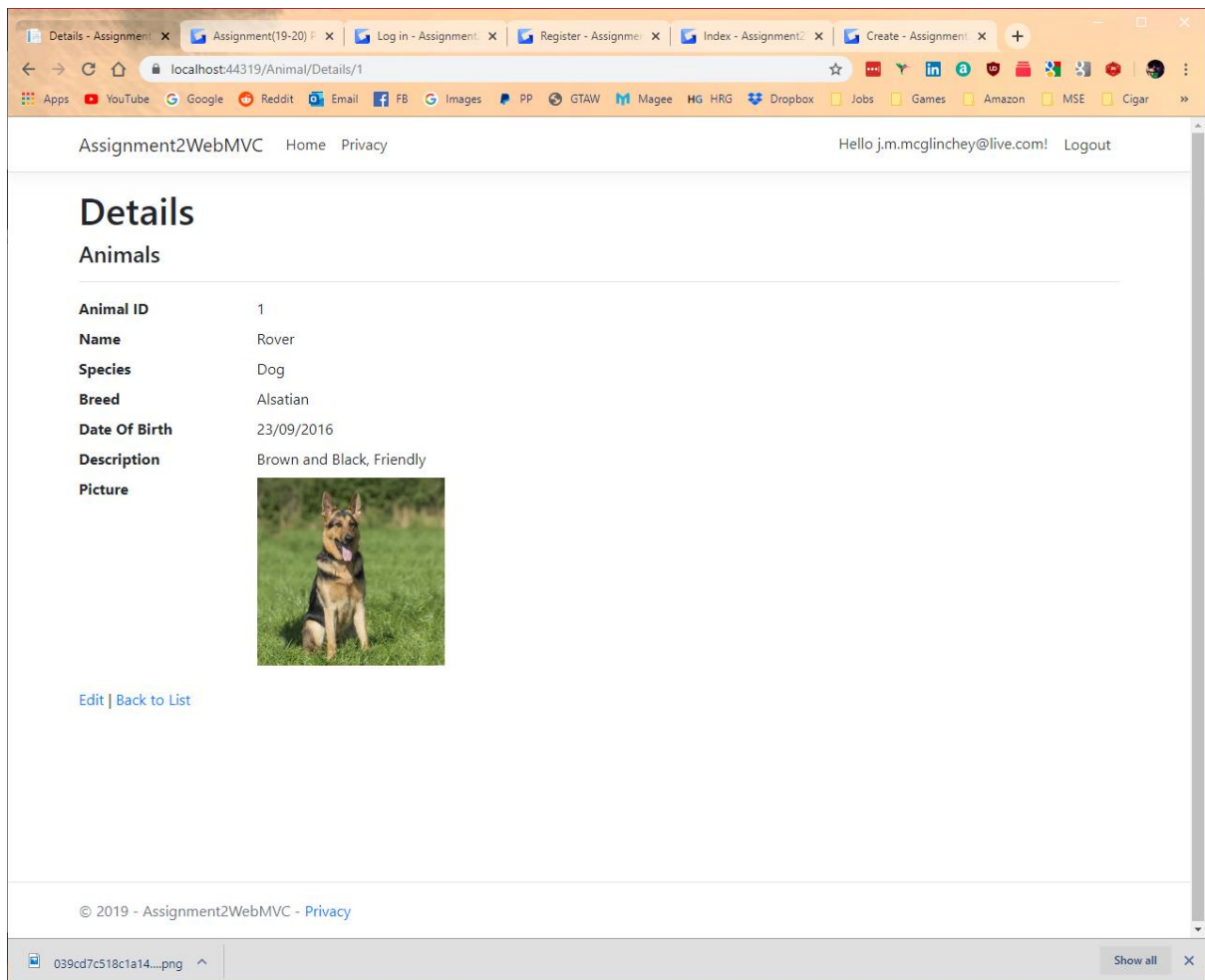
Picture URL
Animal Picture Required

[Create](#)

[Back to List](#)

Create validation

Details View UI



Edit View UI

Assignment2WebMVC Home Privacy Hello j.m.mcglinchey@live.com! Logout

Edit Animals

ID
1

Name
Rover

Species
Dog

Breed
Alsatian

Date of Birth
23/09/2016

Description
Brown and Black, Friendly

Picture URL
https://www.puppies.co.uk/wp-content/uploa

[Save](#)

[Back to List](#)

© 2019 - Assignment2WebMVC - [Privacy](#)

Delete View UI

