AdminController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using BlogApp;
using BlogApp.Repositories;
using BlogUILayer.Models;
using DataAccessLayer.Authentication;

namespace BlogUILayer.Controllers
{
    public class AdminController : Controller
    {



        private readonly IAuthService authService;
        private readonly IAdminRepository adminRepository;
        private readonly IEmpInfoRepository empRepository;

        public AdminController(IAuthService authService, IAdminRepository adminRepository
            , IEmpInfoRepository empRepository)
        {
            this.authService = authService;
            this.adminRepository = adminRepository;
            this.empRepository = empRepository;
        }


        // GET: Account
        public ActionResult Login()
        {
            return View();
        }

        //public ActionResult Login()
        //{
        //    return View();
        //}

        [HttpPost]


        public ActionResult Login(LoginViewModel model)
        {
            // Add authentication logic for Employee login
            if (ModelState.IsValid)
            {
                bool Authres = authService.Authenticate(model.EmailId, model.Password, UserRole.Admin);
                if (Authres)
                {
                    var employee = adminRepository.GetAdminInfoByEmailId(model.EmailId);
                    Session["EmailId"] = model.EmailId;
            return RedirectToAction("Home", "Admin");
        }
        else
        {
            ModelState.AddModelError("", "Invalid username or password");
            // Add a flag to indicate unsuccessful login attempt
            ModelState.AddModelError("LoginFailed", "true");
        }
    }
    return View(model);
}



//public ActionResult Login(string email, string password)
//{
//    // Validate admin credentials
//    if (email == "admin@example.com" && password == "adminpassword")
//    {
//        // Admin authentication successful
//        // Redirect to admin dashboard or perform other actions
//        return RedirectToAction("AdminDashboard");
//    }

//    // Admin authentication failed
//    ViewBag.ErrorMessage = "Invalid credentials";
//    return View();
//}

//public ActionResult AdminDashboard()
//{
//    return View();
//}


// GET: Admin/Create
public ActionResult Home()
{
    var employeeList = empRepository.GetAllEmpInfos();
    var employeeViewModelList = employeeList.Select(admin => new EmpInfoModel
    {
        EmailId = admin.EmailId,
        Name = admin.Name,
        DateOfJoining = admin.DateOfJoining
    });
    // Map ot
    return View(employeeViewModelList);
}


[HttpGet, ActionName("Create")]
public ActionResult Create()
{
    EmpInfoModel model = new EmpInfoModel
    {

        DateOfJoining = DateTime.Now,
    };
    return View(model);
}

[HttpPost, ActionName("Create")]
public ActionResult CreateEmployee(EmpInfoModel model)
{
    if (ModelState.IsValid)
    {
        var NewEmploye = new EmpInfo
        {
            Name = model.Name,
            DateOfJoining = model.DateOfJoining,
            EmailId = model.EmailId,
            PassCode = model.PassCode
        };
        empRepository.AddEmpInfo(NewEmploye);
        return RedirectToAction("Home", "Admin");
    }
    else
    {
        TempData["ErrorMessage"] = "Something went wrong ";
        return View("Create", "Admin");
    }
}

// GET: Admin/Edit/5
public ActionResult Edit(int id)
{

    var employee = empRepository.GetEmpInfoById(id);

    if (employee != null)
    {

        var employeeViewModel = new EmpInfoModel
        {
            EmpInfoId = employee.EmpInfoId,
            Name = employee.Name,
            EmailId = employee.EmailId,
            DateOfJoining = employee.DateOfJoining,
            PassCode = employee.PassCode
        };
        return View(employeeViewModel);
    }

    // Map the employee data to your view model


    return View();
}

// POST: Admin/Edit/5
[HttpPost]
public ActionResult Edit(EmpInfoModel employeeViewModel)
{
    if (ModelState.IsValid)
```

```
            {
                // Map the view model data back to your data model
                var employee = new EmpInfo
                {
                    DateOfJoining = employeeViewModel.DateOfJoining,
                    EmpInfoId = employeeViewModel.EmpInfoId,
                    Name = employeeViewModel.Name,
                    EmailId = employeeViewModel.EmailId,
                    PassCode = employeeViewModel.PassCode
                };

                empRepository.UpdateEmpInfo(employee);
                return RedirectToAction("Home", "Admin"); // Redirect to the index action or another appropriate
action
            }
            else
            {
                TempData["ErrorMessage"] = "Something went wrong ";
                return View("Edit", "Admin");
            }
        }


        [HttpPost]
        public ActionResult Delete(int Id)
        {

            var temp = empRepository.GetEmpInfoById(Id);
            if (temp != null)
            {
                empRepository.DeleteEmpInfo(temp.EmpInfoId);
                return RedirectToAction("Home", "Admin");
            }
            else
            {
                TempData["ErrorMessage"] = "Something went wrong ";
                return RedirectToAction("Home", "Admin");
            }
        }
        public ActionResult Logout()
        {
            Session.Clear();
            Session.Abandon();
            FormsAuthentication.SignOut();

            return RedirectToAction("Index", "Blog");
        }
    }
}
```

BlogController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using BlogApp;
using BlogApp.Repositories;
using BlogUILayer.Models;

namespace BlogUILayer.Controllers
{
    public class BlogController : Controller
    {
        private readonly IBlogInfoRepository _blogRepository;
        private readonly IEmpInfoRepository empRepository;

        public BlogController(IBlogInfoRepository blogReposotiry, IEmpInfoRepository empRepository)
        {
            this._blogRepository = blogReposotiry;
            this.empRepository = empRepository;
        }
        // GET: log
        public ActionResult Index()
        {

            var blogs = _blogRepository.GetAllBlogInfos();
            var blogViewModels = blogs.Select(blog => new BlogViewModel
            {

                Title = blog.Title,
                Subject = blog.Subject,
                DateOfCreation = blog.DateOfCreation,
                BlogUrl = blog.BlogUrl,
                EmployeeName = blog.Employee.Name
            }).ToList();
            return View(blogViewModels);
        }

        // GET: log/Details/5
        public ActionResult Details(int id)
        {
            return View();
        }

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

        // POST: log/Create
        [HttpPost]
        public ActionResult Create(AddBlogModel blogInfo)
        {
            try
            {
                // TODO: Add insert logic here

                if (ModelState.IsValid)
                {
                    string loggedInEmployee = (string)Session["EmailId"];
                    BlogInfo blog = new BlogInfo
                    {
                        Title = blogInfo.Title,
                        Subject = blogInfo.Subject,
                        DateOfCreation = DateTime.Now,
                        BlogUrl = "https://designyourownblog.com/blog/",
                    };

                    _blogRepository.AddBlogWithForeignKey(loggedInEmployee, blog);

                    return RedirectToAction("Home", "Employee"); // Redirect to the blog list or another action
                }

                return View(blogInfo);


            }
            catch
            {
                return View();
            }
        }

        // GET: log/Edit/5
        public ActionResult Edit(int id)
        {
            return View();
        }

        // POST: log/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add update logic here

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }

        // GET: log/Delete/5
        public ActionResult Delete(int id)
        {
            return View();
        }

        // POST: log/Delete/5
        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
```

```
            {
                // TODO: Add delete logic here

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

EmployeeController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using BlogApp.Repositories;
using BlogUILayer.Models;
using DataAccessLayer.Authentication;

namespace BlogUILayer.Controllers
{
    public class EmployeeController : Controller
    {
        private readonly IAuthService authService;
        private readonly IBlogInfoRepository blogReposotiry;
        private readonly IEmpInfoRepository empRepository;

        // GET: Employee
        public EmployeeController(IEmpInfoRepository empRepository, IAuthService authService,
IBlogInfoRepository blogReposotiry)
        {
            this.empRepository = empRepository;
            this.authService = authService;
            this.blogReposotiry = blogReposotiry;
        }


        // GET: Employee
        public ActionResult Login()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Login(LoginViewModel model)
        {
            // Add authentication logic for Employee login
            if (ModelState.IsValid)
            {
                bool Authres = authService.Authenticate(model.EmailId, model.Password, UserRole.Employee);
                if (Authres)
                {
                    var employee = empRepository.GetEmpInfoByEmialId(model.EmailId);
                    Session["EmailId"] = model.EmailId;
                    Session["UserName"] = employee.Name;
                    Session["UserId"] = employee.EmpInfoId;
                    return RedirectToAction("Home", "Employee");
                }
                else
                {
                    ModelState.AddModelError("", "Invalid username or password");
                    // Add a flag to indicate unsuccessful login attempt
                    ModelState.AddModelError("LoginFailed", "true");
                }
            }
            return View(model);
        }

        public ActionResult Home()
        {
            string email = (string)Session["EmailId"];
            if (email != null)
            {
                var blogWrtieByLoggedInEmployee = blogReposotiry.GetBlogInfoByEmployeeId(email);
                var blogViewModels = blogWrtieByLoggedInEmployee.Select(blog => new BlogViewModel
                {

                    Title = blog.Title,
                    Subject = blog.Subject,
                    DateOfCreation = blog.DateOfCreation,
                    BlogUrl = blog.BlogUrl,
                    EmployeeName = blog.Employee.Name
                }).ToList();
                return View(blogViewModels);
            }
            return View();
        }
        public ActionResult Logout()
        {
            Session.Clear();
            Session.Abandon();
            FormsAuthentication.SignOut();

            return RedirectToAction("Index", "Blog");
        }
    }
}
```

AddBlogModel.cs

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

namespace BlogUILayer.Models
{
    public class AddBlogModel
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public string Subject { get; set; }
        public DateTime DateOfCreation { get; set; }
        public string BlogUrl { get; set; }
        public string EmpEmailId { get; set; }
    }
}
```

AdminInfoModel.cs

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

namespace BlogUILayer.Models
{
    public class AdminInfoModel
    {
        public int AdminInfoId { get; set; }
        public string EmailId { get; set; }
        public string Password { get; set; }
    }
}
```

BlogInfoModel.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
using BlogApp;

namespace BlogUILayer.Models
{
    public class BlogInfoModel
    {
        [Key]
        public int BlogInfoId { get; set; }
        public string Title { get; set; }
        public string Subject { get; set; }
        public DateTime DateOfCreation { get; set; }
        public string BlogUrl { get; set; }
        public string Name  { get; set; }

    }
}
```

BlogViewModel.cs

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

```
namespace BlogUILayer.Models
{
    public class BlogViewModel
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public string Subject { get; set; }
        public DateTime DateOfCreation { get; set; }
        public string BlogUrl { get; set; }
        public string EmployeeName { get; set; }
    }
}
```

EmpInfoModel.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace BlogUILayer.Models
{
    public class EmpInfoModel
    {
        [Key]
        public int EmpInfoId { get; set; }
        [Index("UQ_EmailId", IsUnique = true)]
        [MaxLength(255)]
        public string EmailId { get; set; }
        public string Name { get; set; }
        public DateTime DateOfJoining { get; set; }
        public string PassCode { get; set; }
        public string cnfrmPassword { get; set; }
    }
}
```

LoginViewModel.cs

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

namespace BlogUILayer.Models
{

        public class LoginViewModel
        {
            public string EmailId { get; set; }
            public string Password { get; set; }
        }
    }
```