

AdminInfo.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace WebAppBlog.Models
{
    [Table("AdminInfo")]
    public class AdminInfo
    {
        [Key]
        public int AdminId { get; set; }
        public string? EmailId { get; set; }

        public string? Password { get; set; }
    }
}
```

EmpInfo.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace WebAppBlog.Models
{
    [Table("EmpInfo")]
    public class EmpInfo
    {
        [Key]
        public int EmpId { get; set; }
        public string? EmailId { get; set; }

        public string? Name { get; set; }

        public DateTime DOJ { get; set; }

        public int PassCode { get; set; }
    }
}
```

BlogInfo.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace WebAppBlog.Models
{
    [Table("BlogInfo")]
    public class BlogInfo
    {
        [Key]
        public int BlogId { 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; }
    }
}

```

BlogContext.cs

```

using Microsoft.EntityFrameworkCore;

namespace WebAppBlog.Models
{
    public class BlogContext: DbContext
    {
        public BlogContext(DbContextOptions<BlogContext> options)
            : base(options)
        {

        }

        public virtual DbSet<AdminInfo> AdminInfos { get; set; }

        public virtual DbSet<BlogInfo> BlogInfos { get; set; }

        public virtual DbSet<EmpInfo> EmpInfos { get; set; }
    }
}

```

HomeController

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using WebAppBlogMVC.Filters;
using WebAppBlogMVC.Models;

namespace WebAppBlogMVC.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

```

```

        return View();
    }
    [OurAuthFilter]
    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";

        return View();
    }

    [HttpGet]
    public ActionResult AdminLogin()
    {
        return View(new AdminInfo());
    }

    [HttpPost]
    public ActionResult AdminLogin(AdminInfo sign)
    {
        if (ModelState.IsValid)
        {
            if ((sign.EmailId == "surya@gmail.com") && (sign.Password ==
"surya@25"))
            {
                Session["UserId"] = Guid.NewGuid();
                return RedirectToAction("EmployeeList", "AdminInfoes");
            }
            else
            {
                ModelState.AddModelError("", "Either User Name or Password
Incorrect!!!");
                return View(sign);
            }
        }
        else
        { return View(sign); }
    }

    [HttpGet]
    public ActionResult EmpLogin()
    {
        return View(new EmpInfo());
    }

    [HttpPost]
    public ActionResult EmpLogin(EmpInfo sign)
    {
        if (ModelState.IsValid)
        {
            if ((sign.EmailId == "nivi@gmail.com") && (sign.PassCode == 4444))

```

```

        {
            Session["UserId"] = Guid.NewGuid();
            return RedirectToAction("SaveBlog", "EmpInfoes");
        }
        else
        {
            ModelState.AddModelError("", "Either User Name or Password
Incorrect!!!");
            return View(sign);
        }
    }
    else
    { return View(sign); }
}
}
}

```

AdminInfoController

```

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 WebAppBlogMVC.Models;

namespace WebAppBlogMVC.Controllers
{
    public class AdminInfoesController : Controller
    {
        private BlogDbEntities db = new BlogDbEntities();

        // GET: AdminInfoes
        public ActionResult Index()
        {
            return View(db.AdminInfoes.ToList());
        }

        // GET: AdminInfoes/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            AdminInfo adminInfo = db.AdminInfoes.Find(id);
            if (adminInfo == null)
            {

```

```

        return HttpNotFound();
    }
    return View(adminInfo);
}

// GET: AdminInfoes/Create
public ActionResult Create()
{
    return RedirectToAction("SaveEmployee", "AdminInfoes");
}

// POST: AdminInfoes/Create
// To protect from overposting attacks, enable the specific properties you
want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "AdminId,EmailId,Password")]
AdminInfo adminInfo)
{
    if (ModelState.IsValid)
    {
        db.AdminInfoes.Add(adminInfo);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(adminInfo);
}

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

// POST: AdminInfoes/Edit/5
// To protect from overposting attacks, 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 = "AdminId,EmailId,Password")]
AdminInfo adminInfo)
{
    if (ModelState.IsValid)
    {
        db.Entry(adminInfo).State = EntityState.Modified;
    }
}

```

```

        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(adminInfo);
}

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

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

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

[HttpGet]
public ActionResult EmployeeList()
{
    return View(db.EmpInfoes.ToList());
}

[HttpGet]
public ActionResult SaveEmployee()
{
    return View(new EmpInfo());
}
}
}

```

EmpInfoController

```
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 WebAppBlogMVC.Models;

namespace WebAppBlogMVC.Controllers
{
    public class EmpInfoesController : Controller
    {
        private BlogDbEntities db = new BlogDbEntities();

        // GET: EmpInfoes
        public ActionResult Index()
        {
            return View(db.EmpInfoes.ToList());
        }

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

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

        // POST: EmpInfoes/Create
        // To protect from overposting attacks, enable the specific properties you
        // want to bind to, for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"EmpId,EmailId,Name,DOJ,PassCode")] EmpInfo empInfo)
        {
            if (ModelState.IsValid)
            {
                db.EmpInfoes.Add(empInfo);
                db.SaveChanges();
            }
        }
    }
}
```

```

        return RedirectToAction("Index");
    }

    return View(empInfo);
}

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

// POST: EmpInfoes/Edit/5
// To protect from overposting attacks, 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 = "EmpId,EmailId,Name,DOJ,PassCode")]
EmpInfo empInfo)
{
    if (ModelState.IsValid)
    {
        db.Entry(empInfo).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(empInfo);
}

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

// POST: EmpInfoes/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)

```



```

    {
        EmpInfo empInfo = db.EmpInfoes.Find(id);
        db.EmpInfoes.Remove(empInfo);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

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

        base.Dispose(disposing);
    }

    [HttpGet]
    public ActionResult SaveBlog()
    {
        return View(new BlogInfo());
    }
}

```

BlogInfoController

```

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 WebAppBlogMVC.Models;

namespace WebAppBlogMVC.Controllers
{
    public class BlogInfoesController : Controller
    {
        private BlogDbEntities db = new BlogDbEntities();

        // GET: BlogInfoes
        public ActionResult Index()
        {
            return View(db.BlogInfoes.ToList());
        }

        // GET: BlogInfoes/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {

```

```

        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    BlogInfo blogInfo = db.BlogInfoes.Find(id);
    if (blogInfo == null)
    {
        return HttpNotFound();
    }
    return View(blogInfo);
}

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

// POST: BlogInfoes/Create
// To protect from overposting attacks, enable the specific properties you
want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"BlogId,Title,Subject,DateOfCreation,BlogUrl,EmpEmailId")] BlogInfo blogInfo)
{
    if (ModelState.IsValid)
    {
        db.BlogInfoes.Add(blogInfo);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(blogInfo);
}

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

// POST: BlogInfoes/Edit/5
// To protect from overposting attacks, 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 =
"BlogId,Title,Subject,DateOfCreation,BlogUrl,EmpEmailId")] BlogInfo blogInfo)

```

```

    {
        if (ModelState.IsValid)
        {
            db.Entry(blogInfo).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(blogInfo);
    }

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

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

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

```

LogInController

```

using BlogLib;
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 WebAppBlogMVC.Models;

namespace WebAppBlogMVC.Controllers
{
    public class LoginsController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View(new Login());
        }
        [HttpPost]
        public ActionResult Index(LogIn sign)
        {
            if (ModelState.IsValid)
            {
                if ((sign.EmailId == "surya@gmail.com") && (sign.Password ==
"surya@25"))
                {
                    Session["UserId"] = Guid.NewGuid();
                    return RedirectToAction("Index", "Home");
                }
                else
                {
                    ModelState.AddModelError("", "Either User Name or Password
Incorrect!!!");
                    return View(sign);
                }
            }
            else
            { return View(sign); }
        }
    }
}

```

App Settings

```

{
    "Logging": {
        "LogLevel": {
            "Default": "Information",
            "Microsoft.AspNetCore": "Warning"
        }
    },
    "ConnectionStrings": {
        "BlogCS": "Server=tcp:suryablogdb2401.database.windows.net,1433;Initial
Catalog=BlogDb;Persist Security Info=False;User

```

```
ID=suryaadmin;Password=Suryacse@251201;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"
    },
    "AllowedHosts": "*"
}

```

Layout

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <style>
        body {
            font-family: Arial, Helvetica, sans-serif;
        }

        .mynav_navbar {
            overflow: hidden;
        }

        .mynav_navbar a {
            font-size: 16px;
            color: white;
            text-decoration: none;
        }

        .mynav_dropdown {
            overflow: hidden;
        }

        .mynav_dropdown .mynav_dropbtn {
            font-size: 16px;
            border: none;
            outline: none;
            color: white;
            padding: 14px 16px;
            background-color: inherit;
            font-family: inherit;
            margin: 0;
        }

        .mynav_dropdown-content {
            display: none;
            position: absolute;
            background-color: #f9f9f9;
            min-width: 160px;
        }

        .mynav_dropdown-content a {

```

```

        float: none;
        padding: 12px 16px;
        text-decoration: none;
        display: block;
        margin-left: 5px;
        color: black;
    }

    .mynav_dropdown:hover .mynav_dropdown-content {
        display: block;
        color: seagreen;
    }
</style>
</head>
<body>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark">
        <div class="container">
            @Html.ActionLink("Blog Tracker", "Index", "Home", new { area = "" }, new
{ @class = "navbar-brand" })
            @Html.ActionLink("Home", "Index", "Home", new { area = "" }, new {
@class = "navbar-brand" })
            @Html.ActionLink("Logout", "Index", "Home", new { area = "" }, new {
@class = "navbar-brand" })
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target=".navbar-collapse" title="Toggle navigation" aria-
controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse d-sm-inline-flex justify-content-
between">
                <ul class="navbar-nav flex-grow-1">
                    <div class="mynav_dropdown">
                        <button class="mynav_dropbtn">
                            LogIn
                            <i class="fa fa-caret-down"></i>
                        </button>
                        <div class="mynav_dropdown-content">
                            <a>@Html.ActionLink("Admin", "AdminLogin", "Home", new {
area = "" }, new { @class = "nav-link" })</a>
                            <a>@Html.ActionLink("Employee", "EmpLogin", "Home", new
{ area = "" }, new { @class = "nav-link" })</a>
                        </div>
                    </div>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

```

```
        </footer>
    </div>

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