```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using NEAproject.Models;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Threading.Tasks;



namespace NEAproject.Controllers
{
    public class HomeController : Controller
    {

        private readonly ILogger<HomeController> _logger;



        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;


        }


        public ActionResult Index(homemodel homemodel, bool? loginsuccess = false, string Save = null)
            //ActionResult adds another task to render the page on the browser in this instance. bridge between CSS/HTML page and browser
        {
            bool valid = true;
            //to the code
            if (homemodel.selectcomplexity == "getLineardatapoints")
            {
                ViewBag.datapointlist = JsonConvert.SerializeObject(Helper.getLineardatapoints(homemodel.valueofn));
                valid = Helper.isvalidint(homemodel.valueofn.ToString());
            }
            else if (homemodel.selectcomplexity == "getNtothe2points")
            {
                ViewBag.datapointlist = JsonConvert.SerializeObject(Helper.getNtothe2points(homemodel.valueofn));
                valid = Helper.isvalidint(homemodel.valueofn.ToString());
            }
            else if (homemodel.selectcomplexity == "get2totheNpoints")
            {
                ViewBag.datapointlist = JsonConvert.SerializeObject(Helper.get2totheNpoints(homemodel.valueofn));
                valid = Helper.isvalidint(homemodel.valueofn.ToString());
            }
            else
            {
                List<datapoint> datapointlist = new List<datapoint>();
                datapointlist.Add(new datapoint("0", 0));
                datapointlist.Add(new datapoint("1", 1));
                datapointlist.Add(new datapoint("2", 2));
                datapointlist.Add(new datapoint("3", 3));
                datapointlist.Add(new datapoint("4", 4));
                datapointlist.Add(new datapoint("5", 5));
                datapointlist.Add(new datapoint("6", 6));
                datapointlist.Add(new datapoint("7", 7));
                datapointlist.Add(new datapoint("8", 8));
                datapointlist.Add(new datapoint("9", 9));
                datapointlist.Add(new datapoint("10", 10));
                datapointlist.Add(new datapoint("11", 11));
                datapointlist.Add(new datapoint("12", 12));
                datapointlist.Add(new datapoint("13", 13));
                datapointlist.Add(new datapoint("14", 14));
                datapointlist.Add(new datapoint("15", 15));



                ViewBag.datapointlist = JsonConvert.SerializeObject(datapointlist);
            }
            ViewBag.validint = valid;



            return View(homemodel);
        }



        public ActionResult Comparison(homemodel homemodel, string Save = null)
        {
            //to the code
            ViewBag.Message = "Your comparison page.";
            if (homemodel != null)
            {

                ViewBag.datapointlist = JsonConvert.SerializeObject(Helper.getLineardatapoints(homemodel.valueofn));

                ViewBag.datapointlist1 = JsonConvert.SerializeObject(Helper.getNtothe2points(homemodel.valueofn));

                ViewBag.datapointlist2 = JsonConvert.SerializeObject
```

```
                  (Helper.get2totheNpoints(homemodel.valueofn));

            }


            return View(homemodel);
        }


        public IActionResult About()
        {
            ViewBag.Message = "Your application description page.";
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None,
          NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId =
              Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }

    }
}

```