< Summary

Methods/Properties

ClientProfileDataValidation(FuelQuoteAp…

Information

Class:	FuelQuoteApp_p1.Controllers.ClientController
Assembly:	FuelQuoteApp_p1
File(s):	C:\Users\Dell\OneDrive\Desktop\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

Line coverage

100%

Covered lines:	20
Uncovered lines:	0
Coverable lines:	20
Total lines:	123
Line coverage:	100%

Branch coverage

100%

Covered branches:	18
Total branches:	18
Branch coverage:	100%

Method coverage

Method coverage is only available for sponsors.

Upgrade to PRO version

Metrics

Method	Branch coverage	Cyclomatic complexity	Line coverage
ClientProfileDataValidation(...)	100%	18	100%

File(s)

C:\Users\Dell\OneDrive\Desktop\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

# Line Line coverage

```
using FuelQuoteApp_p1.EntModels.Models;
using FuelQuoteApp_p1.Models.Client_Profile;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;
using System;
using System.Diagnostics.CodeAnalysis;

namespace FuelQuoteApp_p1.Controllers
{
    public class ClientController : Controller
    {
        [HttpGet]
        [ExcludeFromCodeCoverage]
        public IActionResult ClientDashBoard()
        {
            int usrID = 1; //HardCoded
            Client client = new Client
            {
                User_ID = new User()
                {
                    Id = usrID,
                    Email="sainarne15@gmail.com",
                    UserName="sainarne15@gmail.com"
                },
                FullName = "Narne Lakshmi Narasimha Sai",
                Address1 = "Kirby Dr",
                Address2 = "",
                City = "Houston",
                State = States.TX.ToString(),
                ZipCode = 77054.ToString()
            };
            HttpContext.Session.SetString("ClientDetails", JsonConvert.SerializeObject(client));

            return View();
        }

        [HttpGet]
        [ExcludeFromCodeCoverage]
        public IActionResult ClientProfile()
        {
            return View();
        }


        [HttpPost]
        [ExcludeFromCodeCoverage]
        public ActionResult ClientProfile(Profile clientProfile)
        {
            if (ModelState.IsValid)
            {
                Client client = new Client
                {
                    FullName = clientProfile.FullName,
                    Address1 = clientProfile.Address1,
                    Address2 = clientProfile.Address2,
                    City = clientProfile.City,
                    State = clientProfile.State.ToString(),
                        ZipCode = clientProfile.ZipCode,
                        Email = User.Identity.Name

                    };

                    User usrDetails = new User();
                    client.User_ID = usrDetails;


                    return RedirectToAction("GetQuote", "Quote");
                }
                else
                {
                    return View();
                }

            }

            [HttpGet]
            [ExcludeFromCodeCoverage]
            public IActionResult DisplayProfile()
            {
                Client client = new Client();
                client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));


                Profile cl = new Profile()
                {
                    FullName = client.FullName,
                    Address1 = client.Address1,
                    Address2 = client.Address2,
                    City = client.City,
                    State = States.TX,
                    ZipCode = client.ZipCode
                };

                return View(cl);
            }


            public bool ClientProfileDataValidation(Profile data)
8           {
8               bool flag = false;
8               if ((data.FullName.Length <= 50) && (data.FullName != String.Empty))
6               {
6                   if (((data.Address1.Length <= 100) && (data.Address1 != String.Empty)) && (data.Address2.Length <= 100))
5                   {
5                       if ((data.City.Length <= 100) && (data.City != String.Empty))
3                       {
3                           if (data.ZipCode.Length <= 9 && data.ZipCode.Length >= 5)
2                           {
2                               flag = true;
2                           }
3                       }
5                   }
6               }
                else
2               {
2                   flag = false;
2               }

8               return flag;
8           }
        }
    }
```

Generated by: ReportGenerator 5.1.9.0
10-07-2022 - 15:08:07
GitHub | www.palmmedia.de