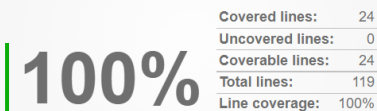


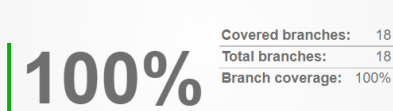
Information

Class: FuelQuoteApp_p1.Controllers.ClientController
Assembly: FuelQuoteApp_p1
File(s): E:\UH\Summer\SD\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

Line coverage



Branch coverage



Method coverage

Method coverage is only available for sponsors.

Upgrade to PRO version

.ctor(FuelQuoteApp_p1.Provider.IFuelQu...
ClientProfileDataValidation(FuelQuoteAp...

Metrics

Method	Branch coverage ⓘ	Cyclomatic complexity ⓘ	Line coverage ⓘ
.ctor(...)	100%	1	100%
ClientProfileDataValidation(...)	100%	18	100%

File(s)

E:\UH\Summer\SD\FuelQuoteApp_p1\FuelQuoteApp_p1\Controllers\ClientController.cs

#	Line	Line coverage
	1	using FuelQuoteApp_p1.EntModels.Models;
	2	using FuelQuoteApp_p1.Models.Client_Profile;
	3	using FuelQuoteApp_p1.Provider;
	4	using FuelQuoteApp_p1.Repository;
	5	using Microsoft.AspNetCore.Http;
	6	using Microsoft.AspNetCore.Mvc;
	7	using Newtonsoft.Json;
	8	using System;
	9	using System.Diagnostics.CodeAnalysis;
	10	
	11	namespace FuelQuoteApp_p1.Controllers
	12	{
	13	public class ClientController : Controller
	14	{
	15	private readonly IFuelQuoteProvider _FuelQuotePro;
	16	
1	17	public ClientController(IFuelQuoteProvider FuelQuotePro)
1	18	{
1	19	_FuelQuotePro = FuelQuotePro;
	20	
1	21	}
	22	[HttpGet]
	23	[ExcludeFromCodeCoverage]
	24	public IActionResult ClientDashBoard()
	25	{
	26	int usrID = _FuelQuotePro.GetUserID(User.Identity.Name);
	27	Client client = _FuelQuotePro.GetClient(usrID);
	28	HttpContext.Session.SetString("ClientDetails", JsonConvert.SerializeObject(client));
	29	
	30	return View();
	31	}
	32	[HttpGet]
	33	[ExcludeFromCodeCoverage]
	34	public IActionResult ClientProfile()
	35	{
	36	return View();
	37	}
	38	
	39	
	40	[HttpPost]
	41	[ExcludeFromCodeCoverage]
	42	public ActionResult ClientProfile(Profile clientProfile)
	43	{
	44	if (ModelState.IsValid)
	45	{
	46	Client client = new Client
	47	{
	48	FullName = clientProfile.FullName,
	49	Address1 = clientProfile.Address1,
	50	Address2 = clientProfile.Address2,
	51	City = clientProfile.City,
	52	State = clientProfile.State.ToString(),
	53	ZipCode = clientProfile.ZipCode,
	54	Email = User.Identity.Name
	55	};
	56	};
	57	
--		

```

58         //User usrDetails = new User();
59         // client.User_ID = usrDetails;
60         client.User_Id = _FuelQuotePro.GetUserID(User.Identity.Name);
61         Client AddedClient = _FuelQuotePro.AddClient(client);
62
63         return RedirectToAction("ClientDashBoard", "Client");
64     }
65     else
66     {
67         return View();
68     }
69 }
70
71
72 [HttpGet]
73 [ExcludeFromCodeCoverage]
74 public IActionResult DisplayProfile()
75 {
76     Client client = new Client();
77     client = JsonConvert.DeserializeObject<Client>(HttpContext.Session.GetString("ClientDetails"));
78
79
80     Profile cl = new Profile()
81     {
82         FullName = client.FullName,
83         Address1 = client.Address1,
84         Address2 = client.Address2,
85         City = client.City,
86         State = (States)Enum.Parse(typeof(States), client.State),
87         ZipCode = client.ZipCode
88     };
89
90     return View(cl);
91 }
92
93 public bool ClientProfileDataValidation(Profile data)
94 {
95     bool flag = false;
96     if ((data.FullName.Length <= 50) && (data.FullName != String.Empty))
97     {
98         if (((data.Address1.Length <= 100) && (data.Address1 != String.Empty)) && (data.Address2.Length <= 100))
99         {
100             if ((data.City.Length <= 100) && (data.City != String.Empty))
101             {
102                 if (data.ZipCode.Length <= 9 && data.ZipCode.Length >= 5)
103                 {
104                     flag = true;
105                 }
106             }
107         }
108     }
109     else
110     {
111         flag = false;
112     }
113
114     return flag;
115 }
116
117
118 }
119 }

```