Assignment #3

Dong Yan

5944970

April 8, 2021

**Part I**

**Decision table based on business requirement**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stub | Entry | | | | | | | | | |
| 3 or more accidents | T | T | T | T | T | F | F | F | F | F |
| No experience driver | T | F | F | F | F | T | F | F | F | F |
| 1-9 years driving experience | F | T | T | F | F | F | T | T | F | F |
| Over 9 years driving experience | F | F | F | T | T | F | F | F | T | T |
| Driver's age >=30 and driving experience >=2 | F | F | T | F | T | F | F | T | F | T |
| Refuse to provide insurance | X | X | X | X | X |  |  |  |  |  |
| Base rate $4000 annually |  |  |  |  |  | X |  |  |  |  |
| Base rate $2500 annually |  |  |  |  |  |  | X | X |  |  |
| Other base rate $1000 annually |  |  |  |  |  |  |  |  | X | X |
| 24% of discount on base rate |  |  |  |  |  |  |  | X |  | X |

**Decision table – Adding Don’t Cares**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stub | Entry | | | | | | | | | |
| Accidents level | 3 or more accidents | | | | | Less than 3 accidents | | | | |
| Driving experience | No experience | 1-9 years | | Others | | No experience | 1-9 years | | Others | |
| Driver's age and driving experience combination | - | - | - | - | - |  | >= 30 years old & >=2 years experience | else | >= 30 years old & >=2 years experience | else |
| Refuse to provide insurance | X | X | X | X | X |  |  |  |  |  |
| Base rate $4000 annually |  |  |  |  |  | X |  |  |  |  |
| Base rate $2500 annually |  |  |  |  |  |  | X | X |  |  |
| Other base rate $1000 annually |  |  |  |  |  |  |  |  | X | X |
| 24% of discount on base rate |  |  |  |  |  |  | X |  | X |  |

**Final decision table without redundancies**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stub | Entry |  |  |  |  |  |
| Accidents level | 3 or more accidents | Less than 3 accidents | | | | |
| Driving experience | - | No experience | 1-9 years | | Others | |
| Driver's age and driving experience combination | - |  | >= 30 years old & >=2 years experience | else | >= 30 years old & >=2 years experience | else |
| Refuse to provide insurance | X |  |  |  |  |  |
| Base rate $4000 annually |  | X |  |  |  |  |
| Base rate $2500 annually |  |  | X | X |  |  |
| Other base rate $1000 annually |  |  |  |  | X | X |
| 24% of discount on base rate |  |  | X |  | X |  |

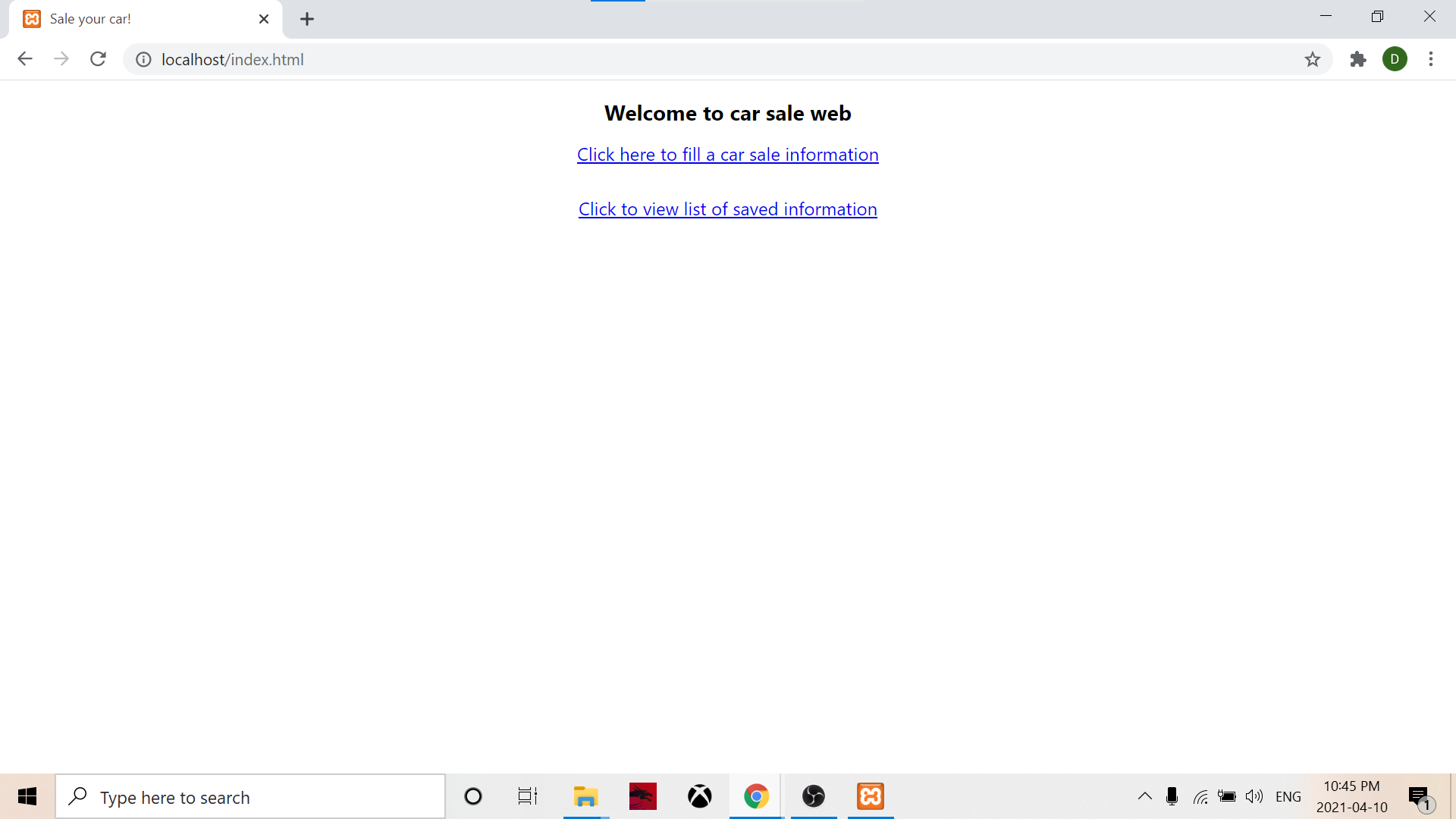
**Test case**

|  |  |  |
| --- | --- | --- |
| Test ID | Test Steps | Expected Outcome |
| 1 | As a driver has 3 or more accidents on record, regardless of driving experience | Refuse to provide this drive with insurance. |
| 2 | As a driver has less than 3 accidents on record, with no driving experience | Provide a base rate of $4000 annually. |
| 3 | As a driver has less than 3 accidents on record, with 1 to 9 years of experience | Provide a base rate of $2500 annually. |
| 4 | As a driver has less than 3 accidents on record, with 2 or more years driving experience and at age 30 or older | Provide a 24% discount on the base rate of $2500 annually. |
| 5 | As a driver has less than 3 accidents on record, with over 9 years of experience | Provide a base rate of $1000 annually. |
| 6 | As a driver has less than 3 accidents on record, with over 9 years driving experience and at age 30 or older | Provide a 24% discount on the base rate of $1000 annually. |

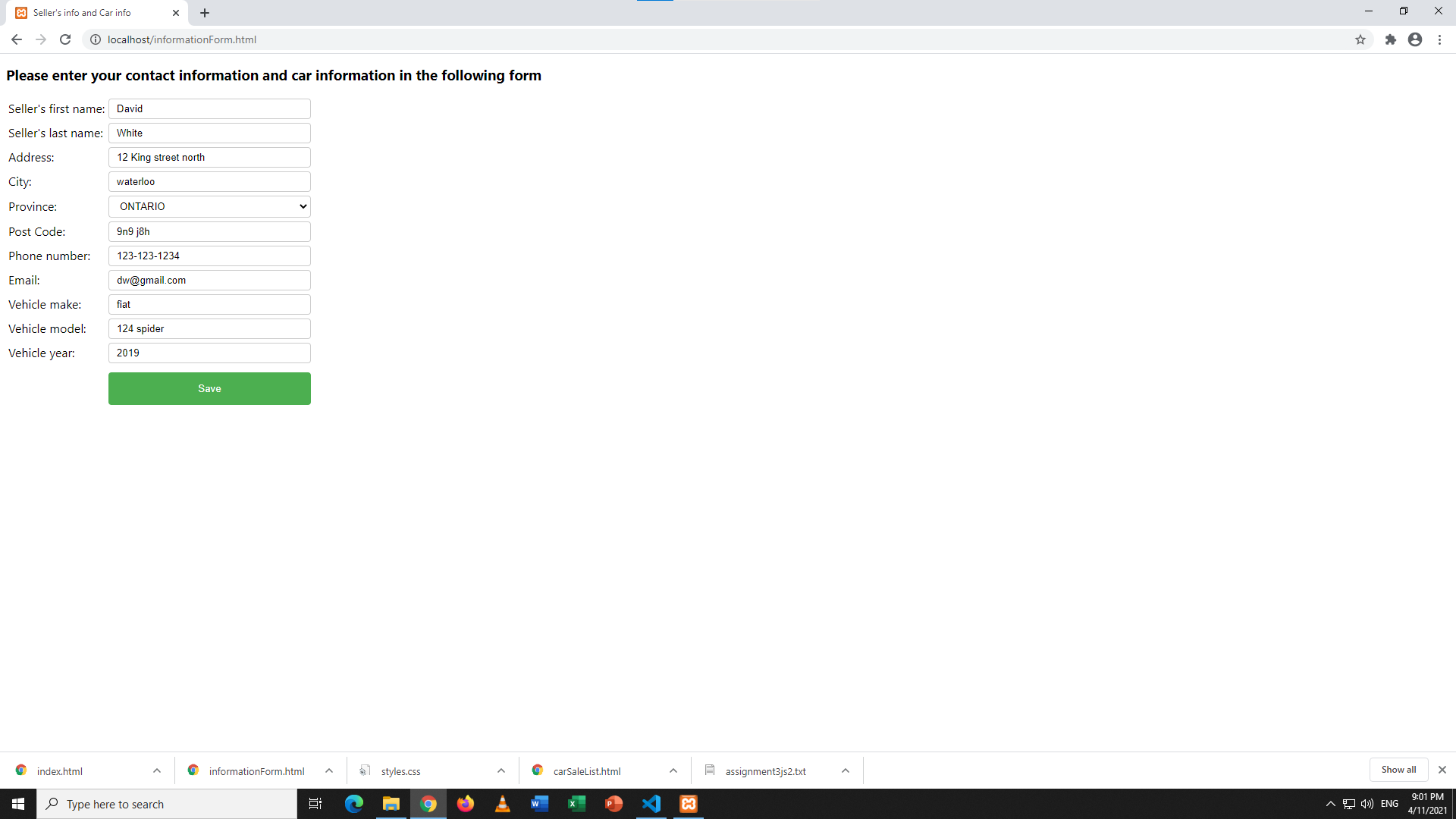
Part II

Task 1 Screenshot

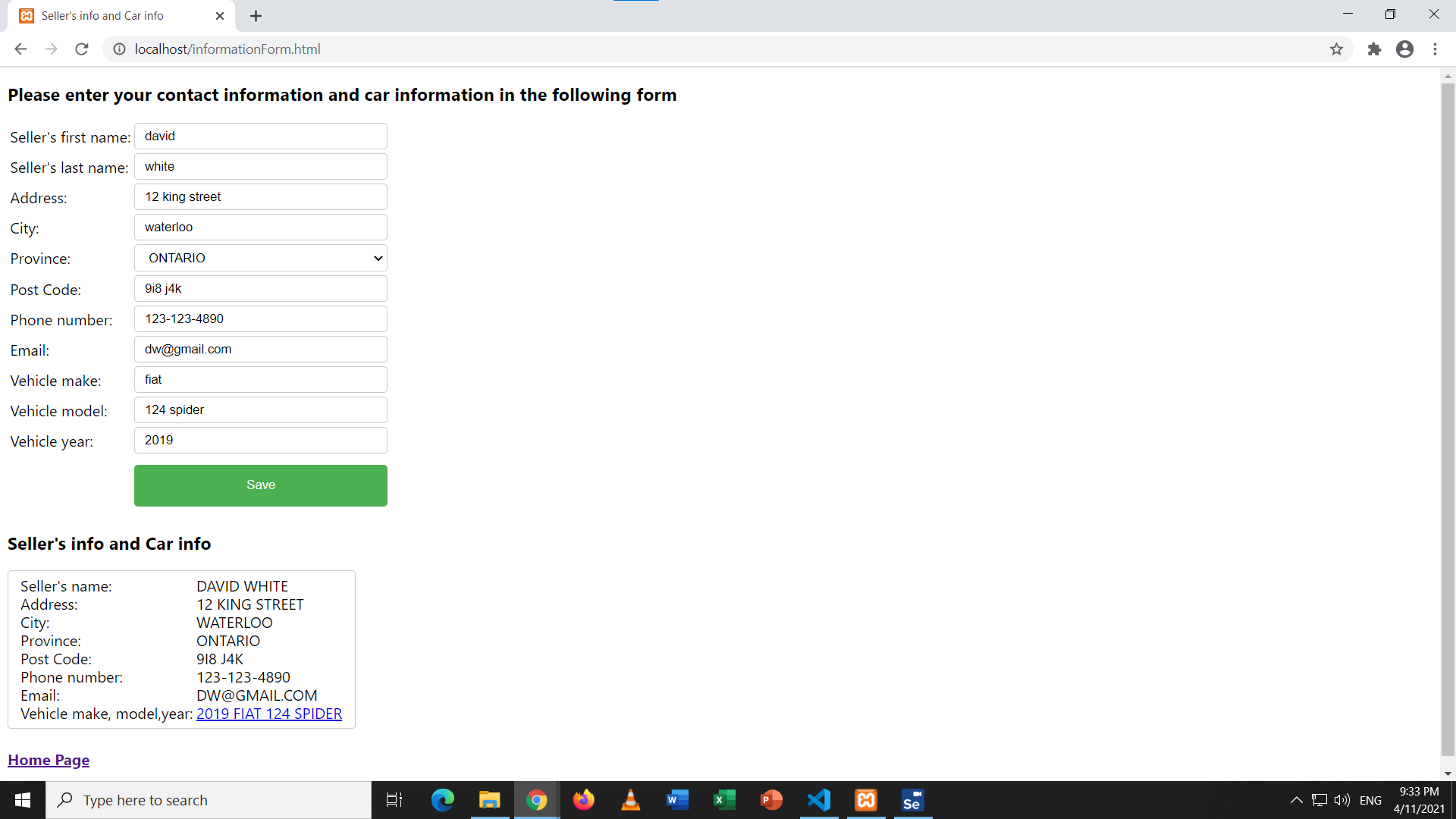
Home Page

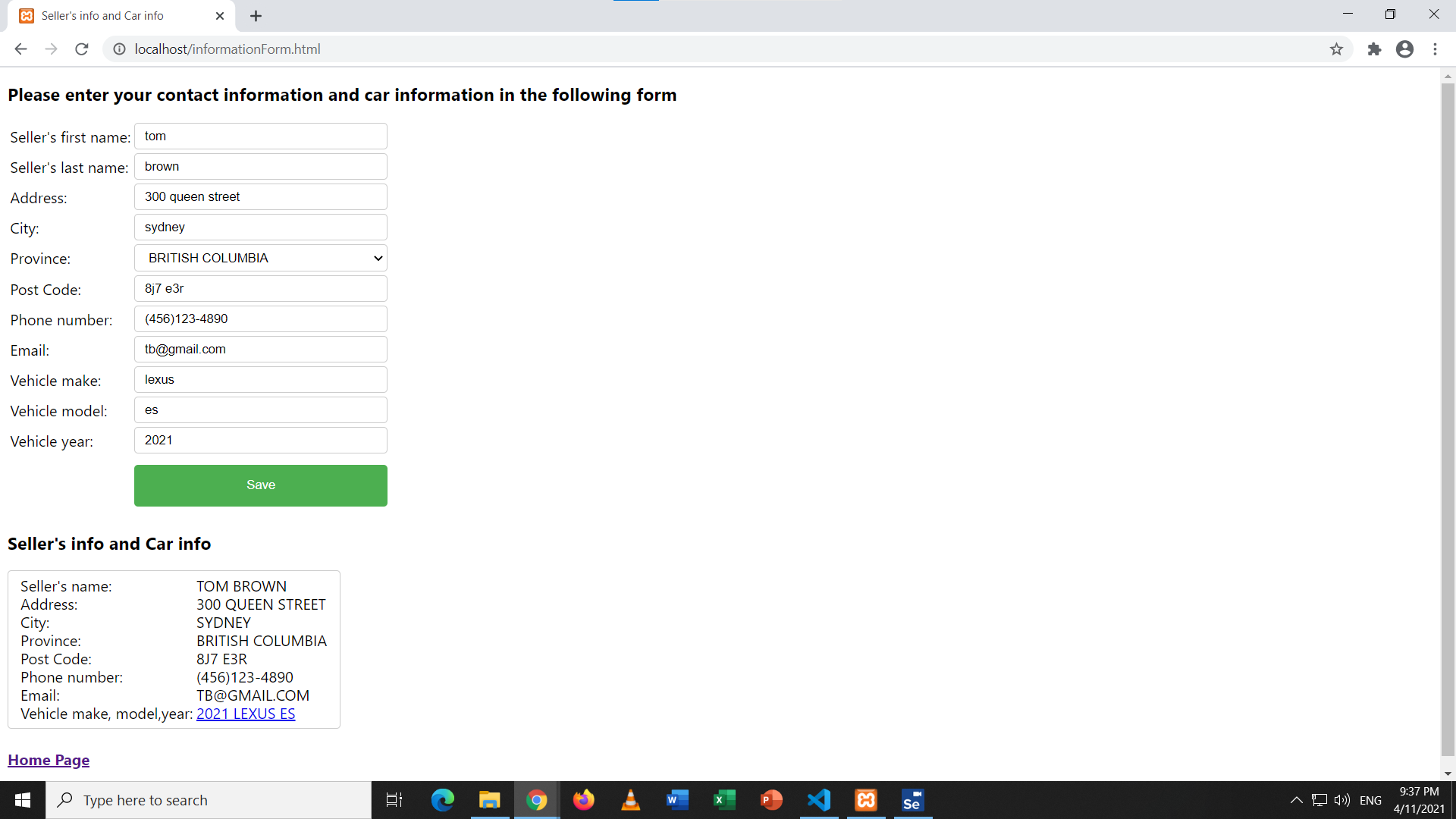


Enter info form

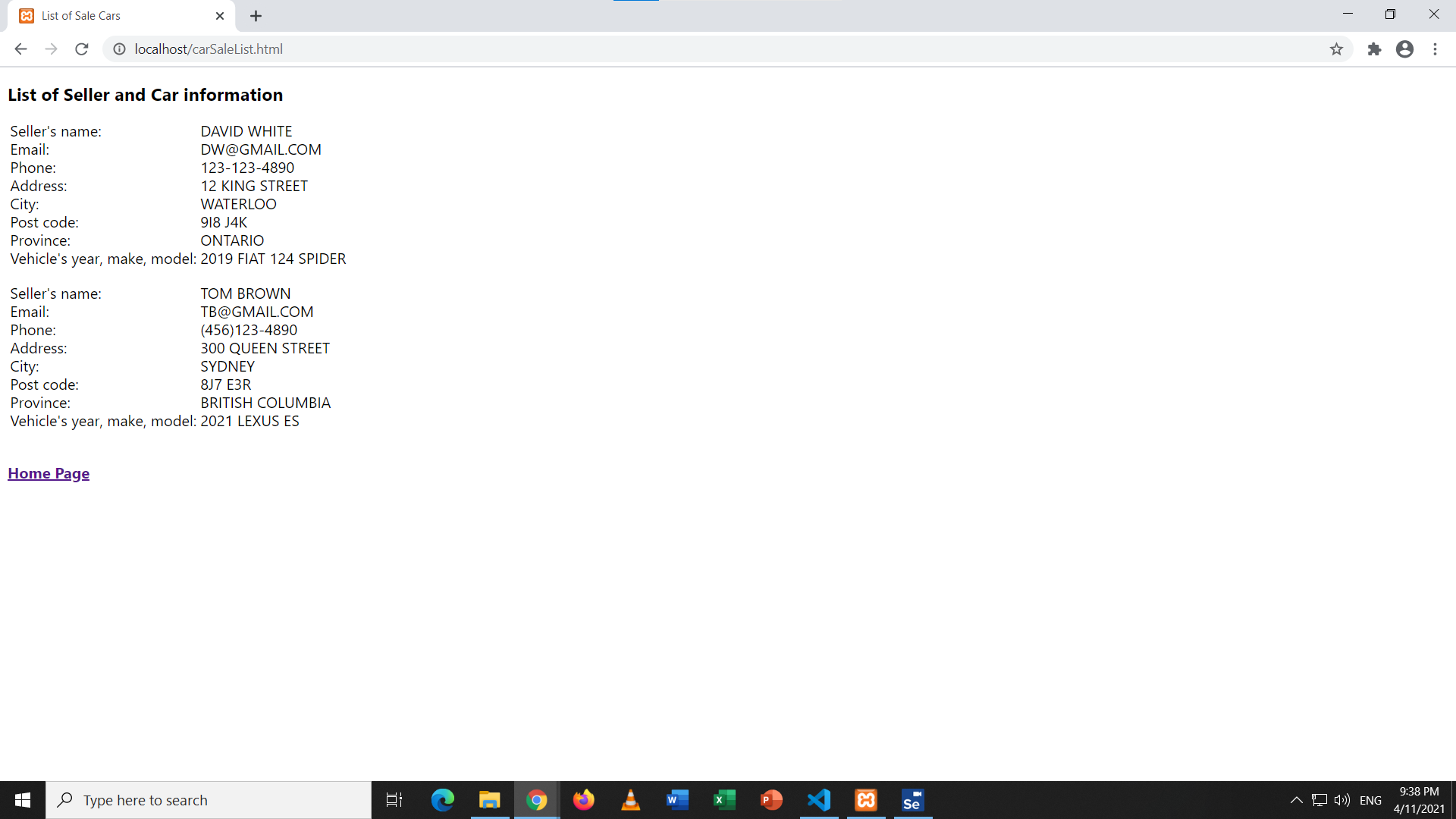


Displayed Entered Info and Link

****



Display List of All Saved Info



**C# code**

//Program: Assignment 3

//

//Purpose: Test case for web application

//

//Revised on : April 12 By Dong Yan

// Generated by Selenium IDE

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Remote;

using OpenQA.Selenium.Support.UI;

using OpenQA.Selenium.Interactions;

using NUnit.Framework;

namespace Assignment\_3

{

[TestFixture]

public class TestforAssignment3Test

{

private IWebDriver driver;

public IDictionary<string, object> vars { get; private set; }

private IJavaScriptExecutor js;

[SetUp]

public void SetUp()

{

driver = new ChromeDriver();

js = (IJavaScriptExecutor)driver;

vars = new Dictionary<string, object>();

}

[TearDown]

protected void TearDown()

{

driver.Quit();

}

[Test]

public void indexPageClickOnClickHereToFillACarSaleInformationReturnInformationForm()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/index.html");

driver.Manage().Window.Size = new System.Drawing.Size(1252, 658);

//Act

driver.FindElement(By.LinkText("Click here to fill a car sale information")).Click();

//Assert

Assert.That(driver.FindElement(By.XPath("//h3")).Text, Is.EqualTo("Please enter your contact information and car information in the following form"));

}

public void informationForm\_NoAnyInputClickSubmit\_ReturnRequiredErrorMessageForAllFields()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("firstNameError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("lastNameError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("addressError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("cityError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("provinceError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("pCodeError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("phoneError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("emailError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("makeError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("makeError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("modelError")).Text, Is.EqualTo("Required an input"));

Assert.That(driver.FindElement(By.Id("yearError")).Text, Is.EqualTo("Required an input"));

}

[Test]

public void informationForm\_Input300SSSSAsAddress\_ReturnInvalidErrorMessage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.CssSelector("#infoForm tr:nth-child(1) > td:nth-child(2)")).Click();

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("300SSSS");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("4H8 J9L");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("bb@gamil.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("bmw");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("1 series");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2009");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("addressError")).Text, Is.EqualTo("Invalid. Please start with number and at least two strings after and have spaces in between"));

}

[Test]

public void informationForm\_Input123NH0AsPostCode\_ReturnInvalidErrorMessage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 King street");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("d4f dde");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gamil.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("fiat");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("124 spider");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2019");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("pCodeError")).Text, Is.EqualTo("Invalid. Please enter e.g 9X9 X9X, X->letters, 9->digts"));

}

[Test]

public void informationForm\_Input1234567890AsPhoneNumber\_ReturnInvalidErrorMessage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 King street");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("7h8 k0p");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("1234567890");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gmail.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("bmw");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("1 series");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2009");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("phoneError")).Text, Is.EqualTo("Invalid. Please enter e.g 123-123-1234 or (123)-1123-1234"));

}

[Test]

public void informationForm\_InputDbAtgmail\_ReturnInvalidErrorMessage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 King Street");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("4f8 j9l");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gmail");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("fiat");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("spider");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2019");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("emailError")).Text, Is.EqualTo("Invalid input"));

}

[Test]

public void informationForm\_Input1889AsYear\_ReturnInvalidErrorMessage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 king street");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("4h8 k9l");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gamil.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("bmw");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("1 series");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("1889");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("yearError")).Text, Is.EqualTo("Invalid. Year start with 19 or 20"));

}

[Test]

public void informationForm\_InputValidInformationForEveryField\_ReturnOuputOfSellerAndCarInfoAndCarLink()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 king street");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("8h9 i7y");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gmail.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("fiat");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("124 spider");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2019");

//Act

driver.FindElement(By.Id("btnSave")).Click();

//Assert

Assert.That(driver.FindElement(By.Id("nameOutput")).Text, Is.EqualTo("BRANDON BROWN"));

Assert.That(driver.FindElement(By.Id("addressOutput")).Text, Is.EqualTo("12 KING STREET"));

Assert.That(driver.FindElement(By.Id("cityOutput")).Text, Is.EqualTo("WATERLOO"));

Assert.That(driver.FindElement(By.Id("provinceOutput")).Text, Is.EqualTo("ONTARIO"));

Assert.That(driver.FindElement(By.Id("pCodeOutput")).Text, Is.EqualTo("8H9 I7Y"));

Assert.That(driver.FindElement(By.Id("phoneOutput")).Text, Is.EqualTo("123-123-1234"));

Assert.That(driver.FindElement(By.Id("emailOutput")).Text, Is.EqualTo("DB@GMAIL.COM"));

Assert.That(driver.FindElement(By.Id("vehicleInfo")).Text, Is.EqualTo("2019 FIAT 124 SPIDER"));

}

[Test]

public void indexPage\_ClickOnClickToViewListOfSavedInformation\_ReturnCarSaleListPage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/index.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

//Act

driver.FindElement(By.LinkText("Click to view list of saved information")).Click();

//Assert

Assert.That(driver.FindElement(By.CssSelector("h3")).Text, Is.EqualTo("List of Seller and Car information"));

}

[Test]

public void carSaleList\_ClickOnHomePage\_ReturnHomePage()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/carSaleList.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

//Act

driver.FindElement(By.LinkText("Home Page")).Click();

//Assert

Assert.That(driver.FindElement(By.CssSelector("h3")).Text, Is.EqualTo("Welcome to car sale web"));

}

[Test]

public void informationForm\_ClickOnCarInfoLink\_ReturnCarInfoInJDPower()

{

//Arrange

driver.Navigate().GoToUrl("http://localhost/informationForm.html");

driver.Manage().Window.Size = new System.Drawing.Size(1251, 658);

driver.FindElement(By.Id("firstName")).Click();

driver.FindElement(By.Id("firstName")).SendKeys("Brandon");

driver.FindElement(By.Id("lastName")).Click();

driver.FindElement(By.Id("lastName")).SendKeys("Brown");

driver.FindElement(By.Id("address")).Click();

driver.FindElement(By.Id("address")).SendKeys("12 king street north");

driver.FindElement(By.Id("city")).Click();

driver.FindElement(By.Id("city")).SendKeys("waterloo");

driver.FindElement(By.Id("province")).Click();

{

var dropdown = driver.FindElement(By.Id("province"));

dropdown.FindElement(By.XPath("//option[. = 'ONTARIO']")).Click();

}

driver.FindElement(By.Id("pCode")).Click();

driver.FindElement(By.Id("pCode")).SendKeys("3f8 i9r");

driver.FindElement(By.Id("phone")).Click();

driver.FindElement(By.Id("phone")).SendKeys("123-123-1234");

driver.FindElement(By.Id("email")).Click();

driver.FindElement(By.Id("email")).SendKeys("db@gmail.com");

driver.FindElement(By.Id("make")).Click();

driver.FindElement(By.Id("make")).SendKeys("fiat");

driver.FindElement(By.Id("model")).Click();

driver.FindElement(By.Id("model")).SendKeys("124 spider");

driver.FindElement(By.Id("year")).Click();

driver.FindElement(By.Id("year")).SendKeys("2019");

//Act

driver.FindElement(By.Id("btnSave")).Click();

driver.FindElement(By.Id("vehicleInfo")).Click();

js.ExecuteScript("window.scrollTo(0,209)");

{

var element = driver.FindElement(By.CssSelector(".pricing-button"));

Actions builder = new Actions(driver);

builder.MoveToElement(element).Perform();

}

//Assert

Assert.That(driver.FindElement(By.CssSelector(".page-heading")).Text, Is.EqualTo("2019 FIAT 124 Spider"));

}

}

}

**HTML code**

**Index.html**

<!--

Project Name: PROG2070Assignment3

Purpose: Develop a web application that store car seller's information and car information

able to retrieve the saved data

Revised History:

By Dong Yan, April 10, 2021

-->

<!DOCTYPE html>

<head>

<title>Sale your car!</title>

<script src="assignment3js.js"></script>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h3 style="text-align: center;">Welcome to car sale web</h3>

<div style="text-align: center;"><a href="informationForm.html">Click here to fill a car sale information</a></div>

<br />

<br />

<div style="text-align: center;"><a href='carSaleList.html'>Click to view list of saved information</a></div>

</body>

</html>

**InformationForm.html**

<!DOCTYPE html>

<head>

<title>Seller's info and Car info</title>

<script src="assignment3js.js"></script>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h3>Please enter your contact information and car information in the following form</h3>

<div class="tables">

<form id="infoForm" name="infoForm">

<table>

<tr>

<td>Seller's first name:</td>

<td><input type="text" id="firstName" name="firstName"></td>

<td class="error" id="firstNameError"></td>

</tr>

<tr>

<td>Seller's last name:</td>

<td><input type="text" id="lastName" name="lastName"></td>

<td class="error" id="lastNameError"></td>

</tr>

<tr>

<td>Address:</td>

<td><input type="text" id="address" name="address" title="e.g 12 king street"></td>

<td class="error" id="addressError"></td>

</tr>

<tr>

<td>City:</td>

<td><input type="text" id="city" name="city"></td>

<td class="error" id="cityError"></td>

</tr>

<tr>

<td>Province:</td>

<td>

<select id="province" name="province">

<option value="">Please Make a Section</option>

<option value="ALBERTA">ALBERTA</option>

<option value="BRITISH COLUMBIA">BRITISH COLUMBIA</option>

<option value="MANITOBA">MANITOBA</option>

<option value="NEW BRUNSWICK">NEW BRUNSWICK</option>

<option value="NEWFOUNDLAND AND LABRADOR">NEWFOUNDLAND AND LABRADOR</option>

<option value="NOVA SCOTIA">NOVA SCOTIA</option>

<option value="ONTARIO">ONTARIO</option>

<option value="PRINCE EDWARD ISLAND">PRINCE EDWARD ISLAND</option>

<option value="QUEBEC">QUEBEC</option>

<option value="SASKATCHEWAN">SASKATCHEWAN</option>

</select>

</td>

<td class="error" id="provinceError"></td>

</tr>

<tr>

<td>Post Code:</td>

<td><input type="text" id="pCode" name="pCode" title="e.g 9X9 X9X"></td>

<td class="error" id="pCodeError"></td>

</tr>

<tr>

<td>Phone number:</td>

<td><input type="text" id="phone" name="phone" title="e.g 123-123-1234 or (123)-123-1234"></td>

<td class="error" id="phoneError"></td>

</tr>

<tr>

<td>Email:</td>

<td><input type="text" id="email" name="email"></td>

<td class="error" id="emailError"></td>

</tr>

<tr>

<td>Vehicle make:</td>

<td><input type="text" id="make" name="make"></td>

<td class="error" id="makeError"></td>

</tr>

<tr>

<td>Vehicle model:</td>

<td><input type="text" id="model" name="model"></td>

<td class="error" id="modelError"></td>

</tr>

<tr>

<td>Vehicle year:</td>

<td><input type="text" id="year" name="year"></td>

<td class="error" id="yearError"></td>

</tr>

<tr>

<td></td>

<td><input type="button" value="Save" id="btnSave" onclick="return validateForm()"></td>

</tr>

</table>

</form>

</div>

<div style="display: none;" id="displayInfo" class="tables">

<h3>Seller's info and Car info</h3>

<form id="displayForm">

<table>

<tr>

<td>Seller's name:</td>

<td id="nameOutput" name="nameOutput"></td>

</tr>

<tr>

<td>Address:</td>

<td id="addressOutput" name="addressOutput"></td>

</tr>

<tr>

<td>City:</td>

<td id="cityOutput" name="cityOutput"></td>

</tr>

<tr>

<td>Province:</td>

<td id="provinceOutput" name="provinceOutput"></td>

</tr>

<tr>

<td>Post Code:</td>

<td id="pCodeOutput" name="pCodeOutput"></td>

</tr>

<tr>

<td>Phone number:</td>

<td id="phoneOutput" name="phoneOutput"></td>

</tr>

<tr>

<td>Email:</td>

<td id="emailOutput" name="emailOutput"></td>

</tr>

<tr>

<td>Vehicle make, model,year:</td>

<td><a id="vehicleInfo" name="vehicleInfo"></td>

</tr>

</table>

</form>

<h4><a href="index.html">Home Page</a></h4>

</div>

</body>

</html>

**CarSaleList.html**

<!DOCTYPE html>

<html>

<head>

<title>List of Sale Cars</title>

<script src="assignment3js.js"></script>

<link rel="stylesheet" href="styles.css">

</head>

<body onload="view()">

<h3>List of Seller and Car information</h3>

<div id="lsOutput"></div>

<h4><a href="index.html">Home Page</a></h4>

</body>

</html>

CSS

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

font-size: 1rem;

line-height: 1;

}

.error {

color: rgb(250, 7, 7);

}

input[type=text], select {

width: 100%;

padding: 5px 10px;

margin: 0.5px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

input[type=button] {

width: 100%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

#displayForm {

width: 100%;

padding: 5px 10px;

margin: 1px 0;

display: inline-block;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

**Javascript**

//identify variables

var carObject = {};

var car = [];

function validateForm() {

// identify variables

const firstName = document.infoForm.firstName.value;

const lastName = document.infoForm.lastName.value;

const email = document.infoForm.email.value;

const phone = document.infoForm.phone.value;

const address = document.infoForm.address.value;

const city = document.infoForm.city.value;

const post = document.infoForm.pCode.value;

const province = document.infoForm.province.value;

const make = document.infoForm.make.value;

const model = document.infoForm.model.value;

const year = document.infoForm.year.value;

const a = document.getElementById("vehicleInfo");

var firstNameErr = lastNameErr = addressErr = cityErr = addressErr = phoneErr = postErr = provinceErr = emailErr = makeErr = modelErr = yearErr = true;

if (firstName == "") {

firstNameError.innerHTML = "Required an input";

} else {

firstNameError.innerHTML = "";

firstNameErr = false;

}

if (lastName == "") {

lastNameError.innerHTML = "Required an input";

} else {

lastNameError.innerHTML = "";

lastNameErr = false;

}

let emailRule = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;

if (email == "") {

emailError.innerHTML = "Required an input";

} else if (email != "" && emailRule.test(email) === false) {

emailError.innerHTML = "Invalid input";

} else {

emailError.innerHTML = "";

emailErr = false;

}

let phoneRule = /^((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}$/;

if (phone == "") {

phoneError.innerHTML = "Required an input";

} else if (phone != "" && phoneRule.test(phone) === false) {

phoneError.innerHTML = "Invalid. Please enter e.g 123-123-1234 or (123)-1123-1234";

} else {

phoneError.innerHTML = "";

phoneErr = false;

}

let addressRule = /^\d+\d\*\S+(?:\s+\S+){2,}$/;

if (address == "") {

addressError.innerHTML = "Required an input";

} else if (address != "" && addressRule.test(address) === false) {

addressError.innerHTML = "Invalid. Please start with number and at least two strings after and have spaces in between";

} else {

addressError.innerHTML = "";

addressErr = false;

}

let cityRule = /^[a-zA-Z]+(?:[\s-][a-zA-Z]+)\*$/;

if (city == "") {

cityError.innerHTML = "Required an input";

} else if (city != "" && cityRule.test(city) === false) {

cityError.innerHTML = "Invalid input";

} else {

cityError.innerHTML = "";

cityErr = false;

}

let postCodeRule = /^\d[A-Za-z]\d\s[A-Za-z]\d[A-Za-z]$/;

if (post == "") {

pCodeError.innerHTML = "Required an input";

} else if (post != "" && postCodeRule.test(post) === false) {

pCodeError.innerHTML = "Invalid. Please enter e.g 9X9 X9X, X->letters, 9->digts";

} else {

pCodeError.innerHTML = "";

postErr = false;

}

if (province == "") {

provinceError.innerHTML = "Required an input";

} else {

provinceError.innerHTML = "";

provinceErr = false;

}

if (make == "") {

makeError.innerHTML = "Required an input";

} else {

makeError.innerHTML = "";

makeErr = false;

}

if (model == "") {

modelError.innerHTML = "Required an input";

} else {

modelError.innerHTML = "";

modelErr = false;

}

let yearRule = /^(19|20)\d{2}$/;

if (year == "") {

yearError.innerHTML = "Required an input";

} else if (yearRule.test(year) === false) {

yearError.innerHTML = "Invalid. Year start with 19 or 20";

} else {

yearError.innerHTML = "";

yearErr = false;

}

if ((firstNameErr || lastNameErr || addressErr || cityErr || addressErr || phoneErr || postErr || provinceErr || emailErr || makeErr || modelErr || yearErr) == true) {

return false;

}

else {

//display output

displayInfo.style.display = "inline-block";

nameOutput.innerHTML = firstName.toUpperCase() + " " + lastName.toUpperCase();

emailOutput.innerHTML = email.toUpperCase();

phoneOutput.innerHTML = phone;

addressOutput.innerHTML = address.toUpperCase();

cityOutput.innerHTML = city.toUpperCase();

pCodeOutput.innerHTML = post.toUpperCase();

provinceOutput.innerHTML = province.toUpperCase();

vehicleInfo.innerHTML = year.toUpperCase() + " " + make.toUpperCase() + " " + model.toUpperCase();

a.setAttribute("href", "https://www.jdpower.com/cars/" + year + "/" + make + "/" + model);

carObject = {

Seller: firstName.toUpperCase() + " " + lastName.toUpperCase(),

Email: email.toUpperCase(),

Phone: phone,

Address: address.toUpperCase(),

City: city.toUpperCase(),

PostCode: post.toUpperCase(),

Province: province.toUpperCase(),

Vehicle: year.toUpperCase() + " " + make.toUpperCase() + " " + model.toUpperCase()

};

car.push(carObject);

//store in the local storage

let car\_serialized = JSON.stringify(car);

localStorage.setItem("items", car\_serialized);

//key++;

}

}

// retrieved from local storage

function view() {

var car\_deserialized = JSON.parse(localStorage.getItem("items"));

for (let i = 0; i < car\_deserialized.length; i++) {

lsOutput.innerHTML += `<table>` +

`<tr><td>Seller's name:</td>` + `<td>` + car\_deserialized[i].Seller + `</td></tr>` +

`<tr><td>Email:</td>` + `<td>` + car\_deserialized[i].Email + `</td></tr>` +

`<tr><td>Phone:</td>` + `<td>` + car\_deserialized[i].Phone + `</td></tr>` +

`<tr><td>Address:</td>` + `<td>` + car\_deserialized[i].Address + `</td></tr>` +

`<tr><td>City:</td>` + `<td>` + car\_deserialized[i].City + `</td></tr>` +

`<tr><td>Post code:</td>` + `<td>` + car\_deserialized[i].PostCode + `</td></tr>` +

`<tr><td>Province:</td>` + `<td>` + car\_deserialized[i].Province + `</td></tr>` +

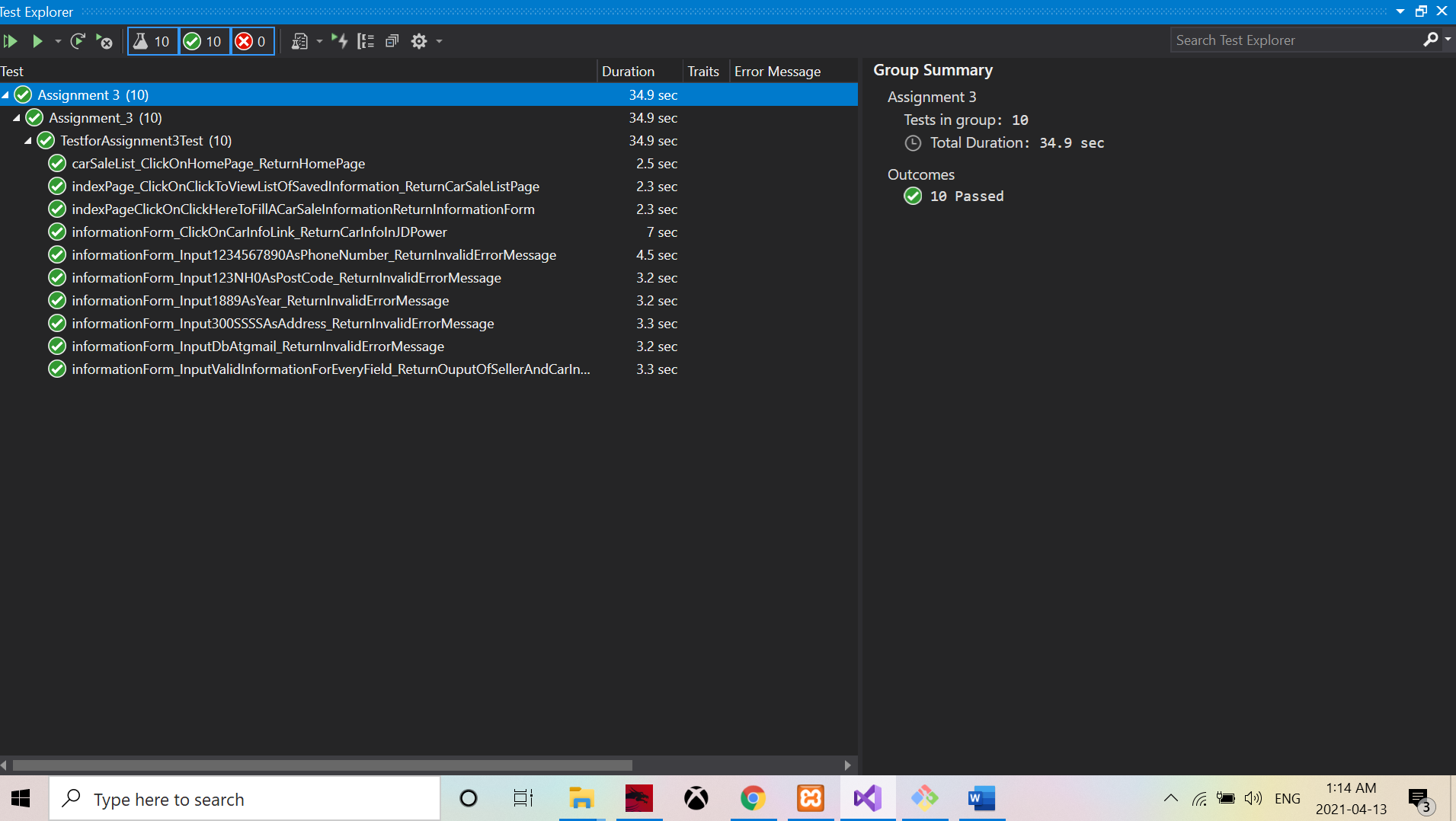
`<tr><td>Vehicle's year, make, model:</td>` + `<td>` + car\_deserialized[i].Vehicle + `</td></tr>` +

`</table><br/>`;

}

}

**ScreenShot of Test Explore**



**Screenshot of Git commits**

