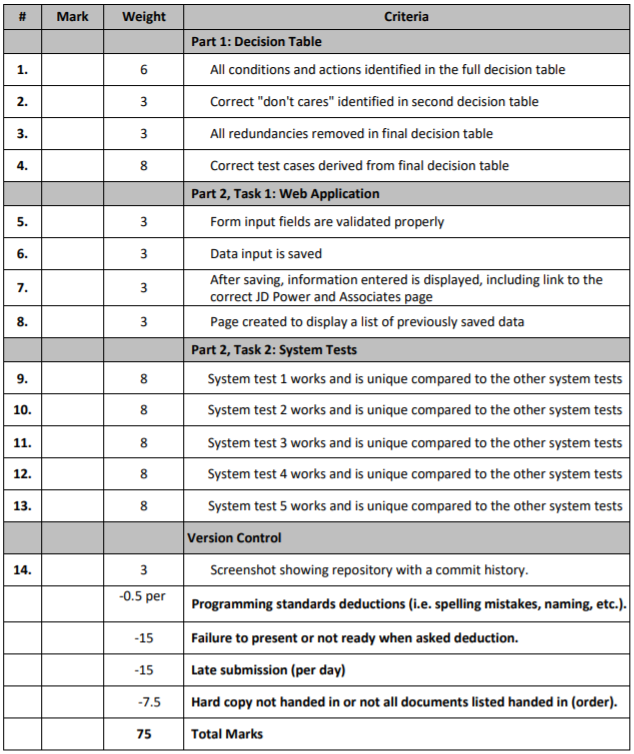
**Assignment #4**

**Tymur Koltunov**

**StudentID 8672727**

**17/11/2019**

**Rubric**



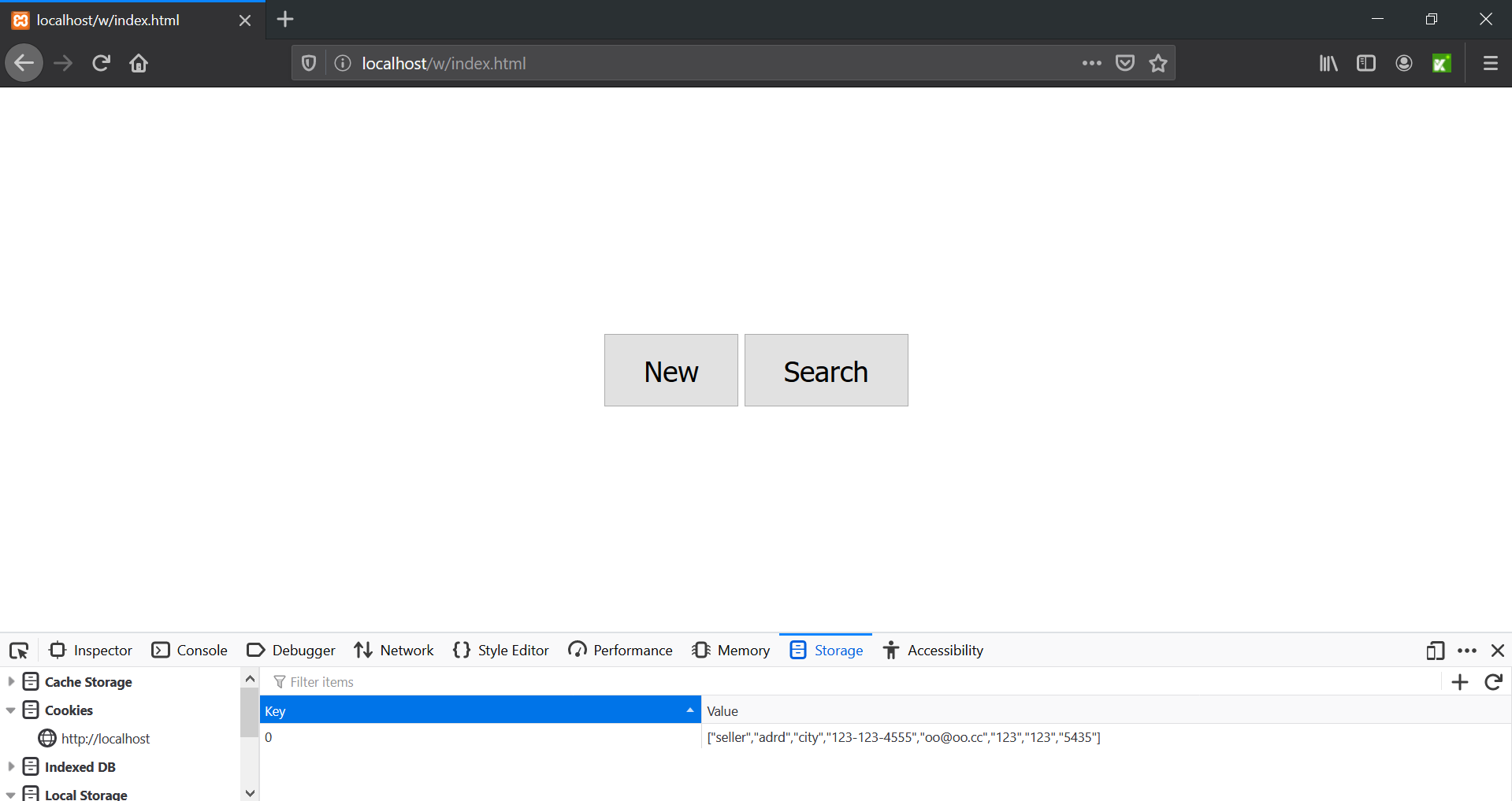
**Decision table**

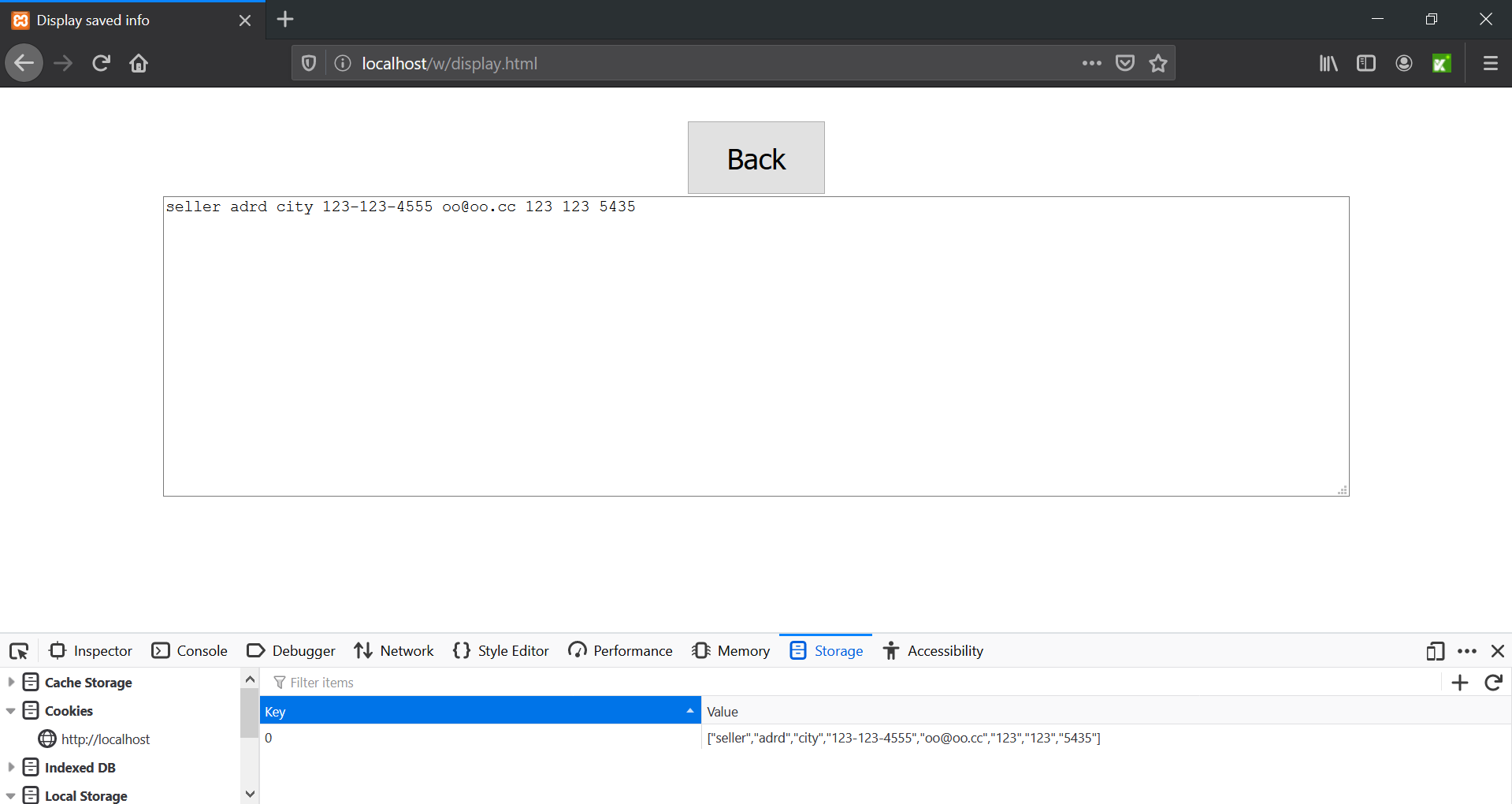
****

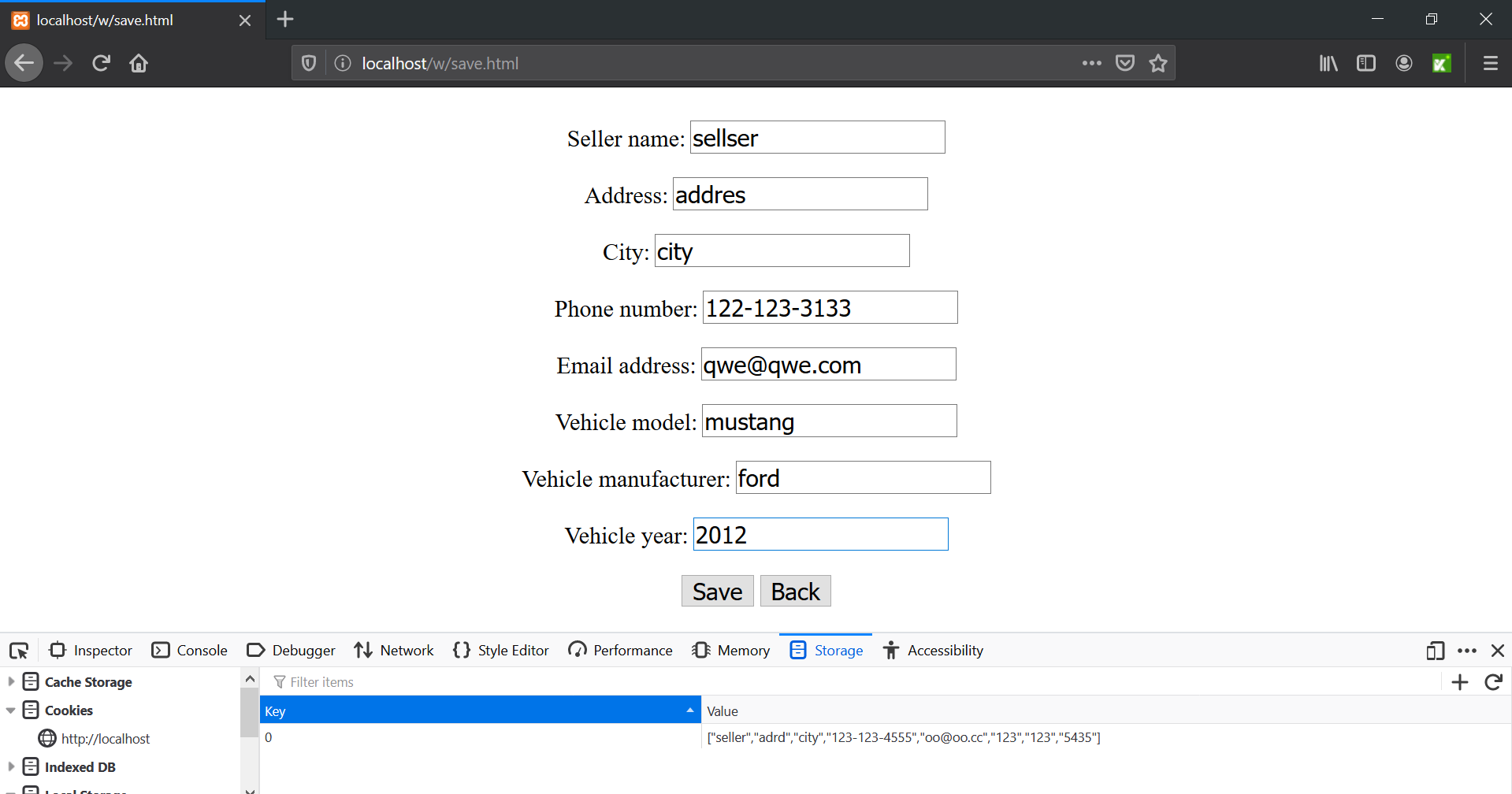
I am well aware that it is not 3 tables which I should provide, but I still do not understand the right way of doing these decision tables, so I provide my complete train of thought.

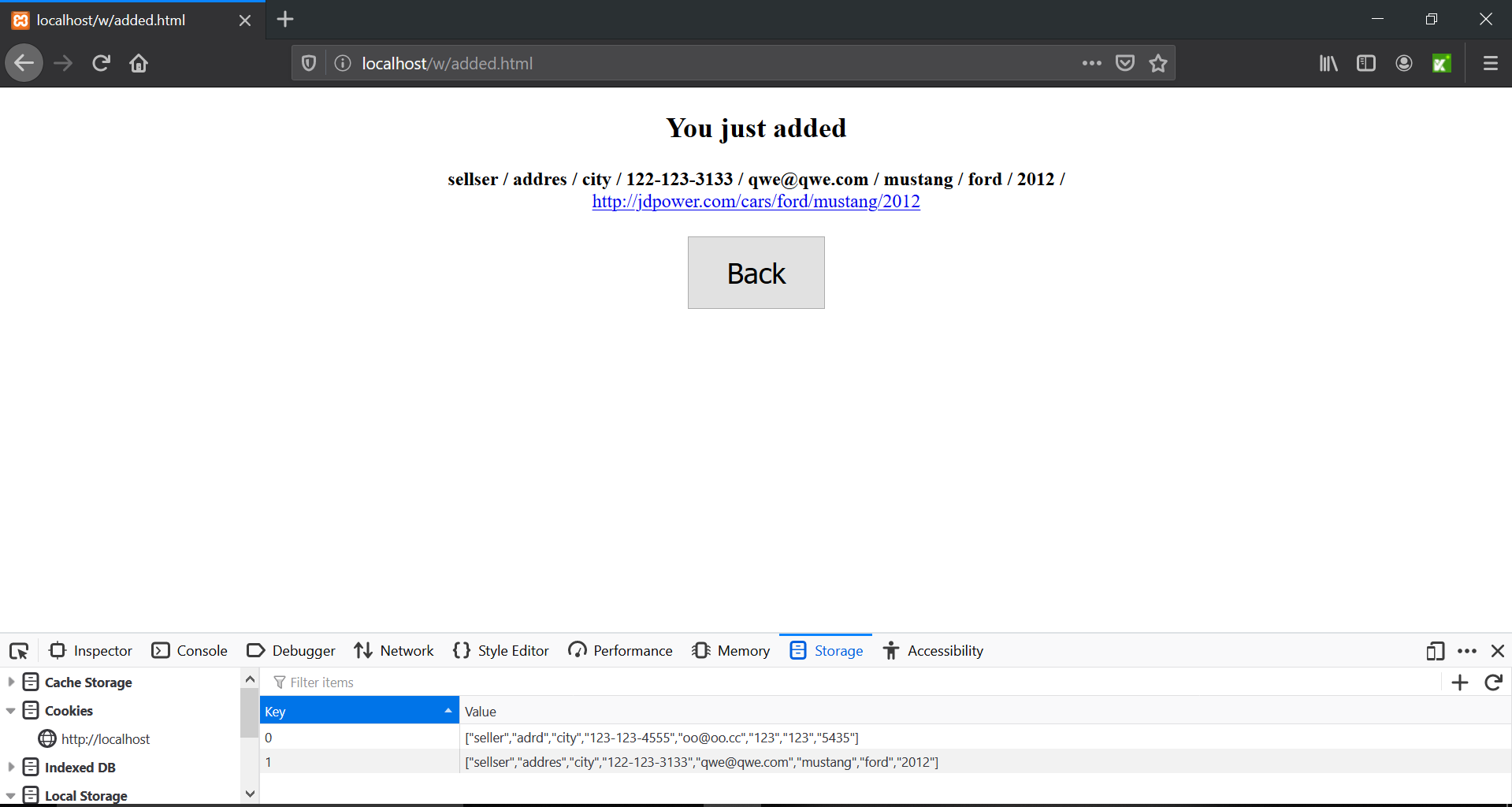
|  |  |  |
| --- | --- | --- |
| TestID | Test Steps | Expected Outcomes |
| 1 | Driver has 3 accidents on his or her record | Refuse to provide insurance |
| 2 | Driver 35 years of age with 6 months experience and 2 accidents on record | Provide insurance with base rate of 3000$ annually with no reduction |
| 3 | Driver 29 years of age with 1.5 years of experience and 1 accident on record | Provide insurance with base rate of 2000$ annually with no reduction |
| 4 | Driver 35 years of age with 7 years of experience and no accidents on record | Provide insurance with base rate of 2000$ annually with 22% reduction |
| 5 | Driver 54 years of age with 10 years of experience and 2 accidents on record | Provide insurance with base rate of 1000$ annually with 22% reduction |
| 6 | Driver 23 years of age with 2 years of experience and no accidents on record | Provide insurance with base rate of 2000$ annually with no reduction |
| 7 | Driver 28 years of age with 9 or more years of experience and no accidents on record | Provide insurance with base rate of 1000$ annually with no reduction |

**Screenshots of working application**









**C# code for system tests**

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

[TestFixture]

public class AddVehicleInputEmptyExpetedErrorMessages

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

driver = new FirefoxDriver();

baseURL = "https://www.katalon.com/";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void AddVehicle\_InputEmpty\_ExpetedErrorMessages()

{

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

driver.FindElement(By.XPath("//input[@value='New']")).Click();

driver.FindElement(By.Id("save\_button")).Click();

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("sellerError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("phoneError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("addressError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("cityError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("emailError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("modelError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("manufacturerError")).Text);

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("yearError")).Text);

}

[Test]

public void AddVehicle\_InputIncorrectPhone\_ExpectedErrorMessage()

{

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

driver.FindElement(By.XPath("//input[@value='New']")).Click();

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

driver.FindElement(By.Id("seller")).Clear();

driver.FindElement(By.Id("seller")).SendKeys("Tymur Koltunov");

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

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

driver.FindElement(By.Id("address")).SendKeys("some home");

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

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

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

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

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

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

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

driver.FindElement(By.Id("vehicleModel")).Clear();

driver.FindElement(By.Id("vehicleModel")).SendKeys("supra");

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

driver.FindElement(By.Id("vehicleManufacturer")).Clear();

driver.FindElement(By.Id("vehicleManufacturer")).SendKeys("toyota");

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

driver.FindElement(By.Id("vehicleYear")).Clear();

driver.FindElement(By.Id("vehicleYear")).SendKeys("2000");

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

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

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

driver.FindElement(By.Id("save\_button")).Click();

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("phoneError")).Text);

}

[Test]

public void AddVehicle\_InputIncorrectEmail\_ExpectedErrorMessage()

{

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

driver.FindElement(By.XPath("//input[@value='New']")).Click();

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

driver.FindElement(By.Id("seller")).Clear();

driver.FindElement(By.Id("seller")).SendKeys("Tymur Koltunov");

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

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

driver.FindElement(By.Id("address")).SendKeys("some home");

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

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

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

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

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

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

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

driver.FindElement(By.Id("vehicleModel")).Clear();

driver.FindElement(By.Id("vehicleModel")).SendKeys("supra");

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

driver.FindElement(By.Id("vehicleManufacturer")).Clear();

driver.FindElement(By.Id("vehicleManufacturer")).SendKeys("toyota");

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

driver.FindElement(By.Id("vehicleYear")).Clear();

driver.FindElement(By.Id("vehicleYear")).SendKeys("2000");

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

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

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

driver.FindElement(By.Id("save\_button")).Click();

Assert.AreEqual("Input not valid", driver.FindElement(By.Id("emailError")).Text);

}

[Test]

public void AddVehicle\_InputCorrectInfo\_ExpectedNextPageWithInfoAndLink()

{

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

driver.FindElement(By.XPath("//input[@value='New']")).Click();

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

driver.FindElement(By.Id("seller")).Clear();

driver.FindElement(By.Id("seller")).SendKeys("Some Name");

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

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

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

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

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

driver.FindElement(By.Id("phone")).SendKeys("321-321-0321");

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

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

driver.FindElement(By.Id("vehicleModel")).Clear();

driver.FindElement(By.Id("vehicleModel")).SendKeys("tundra");

driver.FindElement(By.Id("vehicleManufacturer")).Clear();

driver.FindElement(By.Id("vehicleManufacturer")).SendKeys("toyota");

driver.FindElement(By.Id("vehicleYear")).Clear();

driver.FindElement(By.Id("vehicleYear")).SendKeys("2000");

driver.FindElement(By.Id("save\_button")).Click();

Assert.AreEqual("Some Name / Some street / toronto / 321-321-0321 / valid@email.com / tundra / toyota / 2000 /", driver.FindElement(By.Id("added")).Text);

Assert.AreEqual("http://jdpower.com/cars/toyota/tundra/2000", driver.FindElement(By.Id("link")).Text);

}

[Test]

public void TheSearchInput2VehiclesExpected2VehiclesOnDisplayPageTest()

{

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

driver.FindElement(By.Id("button\_display")).Click();

driver.FindElement(By.Id("button\_clear")).Click();

driver.FindElement(By.Id("back\_from\_display")).Click();

driver.FindElement(By.XPath("//input[@value='New']")).Click();

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

driver.FindElement(By.Id("seller")).Clear();

driver.FindElement(By.Id("seller")).SendKeys("seller one");

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

driver.FindElement(By.Id("address")).SendKeys("address one");

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

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

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

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

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

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

driver.FindElement(By.Id("vehicleModel")).Clear();

driver.FindElement(By.Id("vehicleModel")).SendKeys("tundra");

driver.FindElement(By.Id("vehicleManufacturer")).Clear();

driver.FindElement(By.Id("vehicleManufacturer")).SendKeys("toyota");

driver.FindElement(By.Id("vehicleYear")).Clear();

driver.FindElement(By.Id("vehicleYear")).SendKeys("2000");

driver.FindElement(By.Id("save\_button")).Click();

driver.FindElement(By.Id("button\_back\_added")).Click();

driver.FindElement(By.XPath("//input[@value='New']")).Click();

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

driver.FindElement(By.Id("seller")).Clear();

driver.FindElement(By.Id("seller")).SendKeys("seller two");

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

driver.FindElement(By.Id("address")).SendKeys("address two");

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

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

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

driver.FindElement(By.Id("phone")).SendKeys("456-456-4567");

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

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

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

driver.FindElement(By.Id("vehicleModel")).Clear();

driver.FindElement(By.Id("vehicleModel")).SendKeys("4c");

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

driver.FindElement(By.Id("vehicleManufacturer")).Clear();

driver.FindElement(By.Id("vehicleManufacturer")).SendKeys("alpha-romeo");

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

driver.FindElement(By.Id("vehicleYear")).Clear();

driver.FindElement(By.Id("vehicleYear")).SendKeys("2015");

driver.FindElement(By.Id("save\_button")).Click();

driver.FindElement(By.Id("button\_back\_added")).Click();

driver.FindElement(By.Id("button\_display")).Click();

Assert.AreEqual("seller one address one city one 123-123-1234 email@one.com tundra toyota 2000 \r\nseller two address two city two 456-456-4567 email@two.com 4c alpha-romeo 2015 ", driver.FindElement(By.Id("display")).Text);

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText()

{

try

{

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert)

{

alert.Accept();

}

else

{

alert.Dismiss();

}

return alertText;

}

finally

{

acceptNextAlert = true;

}

}

}

}

**Index.html**

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<input class="button" type="button" onclick="location.href='save.html';" value="New" />

<input id="button\_display" class="button" type="button" onclick="location.href='display.html';" value="Search" />

</body>

</html>

**Save.html**

<HTML>

<head>

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

</head>

<body>

<label class="indent">Seller name:</label>

<input class="indent" type="text" id="seller">

<b id ="sellerError"></b><br>

<label class="indent">Address:</label>

<input class="indent" type="text" id="address">

<b id ="addressError"></b><br>

<label class="indent">City:</label>

<input class="indent" type="text" id="city">

<b id ="cityError"></b><br>

<label class="indent">Phone number:</label>

<input class="indent" type="text" id="phone">

<b id ="phoneError"></b><br>

<label class="indent">Email address:</label>

<input class="indent" type="text" id="email">

<b id ="emailError"></b><br>

<label class="indent">Vehicle model:</label>

<input class="indent" type="text" id="vehicleModel">

<b id ="modelError"></b><br>

<label class="indent">Vehicle manufacturer:</label>

<input class="indent" type="text" id="vehicleManufacturer">

<b id ="manufacturerError"></b><br>

<label class="indent">Vehicle year:</label>

<input class="indent" type="text" id="vehicleYear">

<b id ="yearError"></b><br>

<button class="indent" id="save\_button" onclick="save()">Save</button>

<input class="indent" type="button" onclick="location.href='index.html';" value="Back" />

<script type="text/javascript" src="script.js"></script>

</body>

</HTML>

**Added.html**

<!DOCTYPE html>

<html>

<head>

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

<script type="text/javascript" src="script.js"></script>

<h2>You just added</h2>

</head>

<body>

<b id="added"></b><br>

<a id="link"></a><br>

<input id="button\_back\_added" class="button\_back" type="button" onclick="location.href='index.html';" value="Back" /><br>

<script type="text/javascript">

added();

</script>

</body>

</html>

**Display.html**

<!DOCTYPE html>

<html>

<head>

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

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

<script type="text/javascript" src="script.js"></script>

<title>Display saved info</title>

</head>

<body>

<input id="back\_from\_display" class="button\_back" type="button" onclick="location.href='index.html';" value="Back" />

<input id= "button\_clear" class="button\_back" type="button" onclick="Clear()" value="Clear" /><br>

<textarea readonly id="display" style="width:1000px;height: 250px"></textarea>

<script type="text/javascript">

display();

</script>

</body>

</html>

**Script.js**

function save(){

let entity = [];

let seller = document.getElementById("seller").value;

let address = document.getElementById("address").value;

let city = document.getElementById("city").value;

let phone = document.getElementById("phone").value;

let email = document.getElementById("email").value;

let model = document.getElementById("vehicleModel").value;

let manufacturer = document.getElementById("vehicleManufacturer").value;

let year = document.getElementById("vehicleYear").value;

let i;

let str;

let result = true;

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

let phoneformat1 =/\d{3}-\d{3}-\d{4}/;

let phoneformat2 = /\(\d{3}\)\d{3}-\d{4}\)/;

if(phone.match(phoneformat2) || phone.match(phoneformat1)){

document.getElementById("phoneError").innerHTML = "";

}

else{

document.getElementById("phoneError").innerHTML = "Input not valid";

result = false;

}

if (email.match(mailformat)){

document.getElementById("emailError").innerHTML = "";

}

else{

document.getElementById("emailError").innerHTML = "Input not valid";

result = false;

}

if (city == "") {

document.getElementById("cityError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("cityError").innerHTML = "";

if (address == "") {

document.getElementById("addressError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("addressError").innerHTML = "";

if (seller == "") {

document.getElementById("sellerError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("sellerError").innerHTML = "";

if (isNaN(year) || year < 1900 || year > 2019) {

document.getElementById("yearError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("yearError").innerHTML = "";

if (model == "") {

document.getElementById("modelError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("modelError").innerHTML = "";

if (manufacturer == "") {

document.getElementById("manufacturerError").innerHTML = "Input not valid";

result = false;

}

else

document.getElementById("manufacturerError").innerHTML = "";

entity.push(seller);

entity.push(address);

entity.push(city);

entity.push(phone);

entity.push(email);

entity.push(model);

entity.push(manufacturer);

entity.push(year);

if(result){

for (i = 0; i > -1;) {

str = i.toString();

if(localStorage.getItem(str) == null){

break;

}

else

i++;

}

localStorage.setItem(str, JSON.stringify(entity));

window.location.replace("added.html");

}

}

function display(){

let entity = [];

let i;

let display = "";

let str;

for (i = 0; i > -1;) {

str = i.toString();

if(localStorage.getItem(str) == null)

break;

else{

entity.push(JSON.parse(localStorage.getItem(str)));

i++;

}

}

for (let j = 0; j < entity.length; j++) {

let obj = entity[j];

for (let property in obj) {

display += obj[property] + " ";

}

display += "\n";

}

document.getElementById("display").innerHTML = display;

}

function added(){

let entity = [];

let i;

let d = "";

let last;

let str;

let model;

let manuf;

let year;

last = JSON.parse(getLastElementFromLocalStorage());

i = 0;

for (let property in last) {

d += last[property] + " / ";

if(i == 5){

model = last[property];

}

if(i == 6){

manuf = last[property];

}

if(i == 7){

year = last[property];

}

i++;

}

let link = "http://jdpower.com/cars/" + manuf + "/" + model + "/" + year;

document.getElementById("added").innerHTML = d;

document.getElementById("link").innerHTML = link;

document.getElementById("link").href = link;

}

function getLastElementFromLocalStorage() {

let i = 0;

while(true) {

if(localStorage.getItem(i.toString()) == null) {

return localStorage.getItem((i - 1).toString());

}

i++;

}

}

function Clear(){

localStorage.clear();

}

**Style.css**

body{

text-align: center;

}

.indent{

margin-top: 20px;

font-size: 20px;

}

.button {

padding: 15px 32px;

font-size: 24px;

margin-top: 200px;

}

.button\_back {

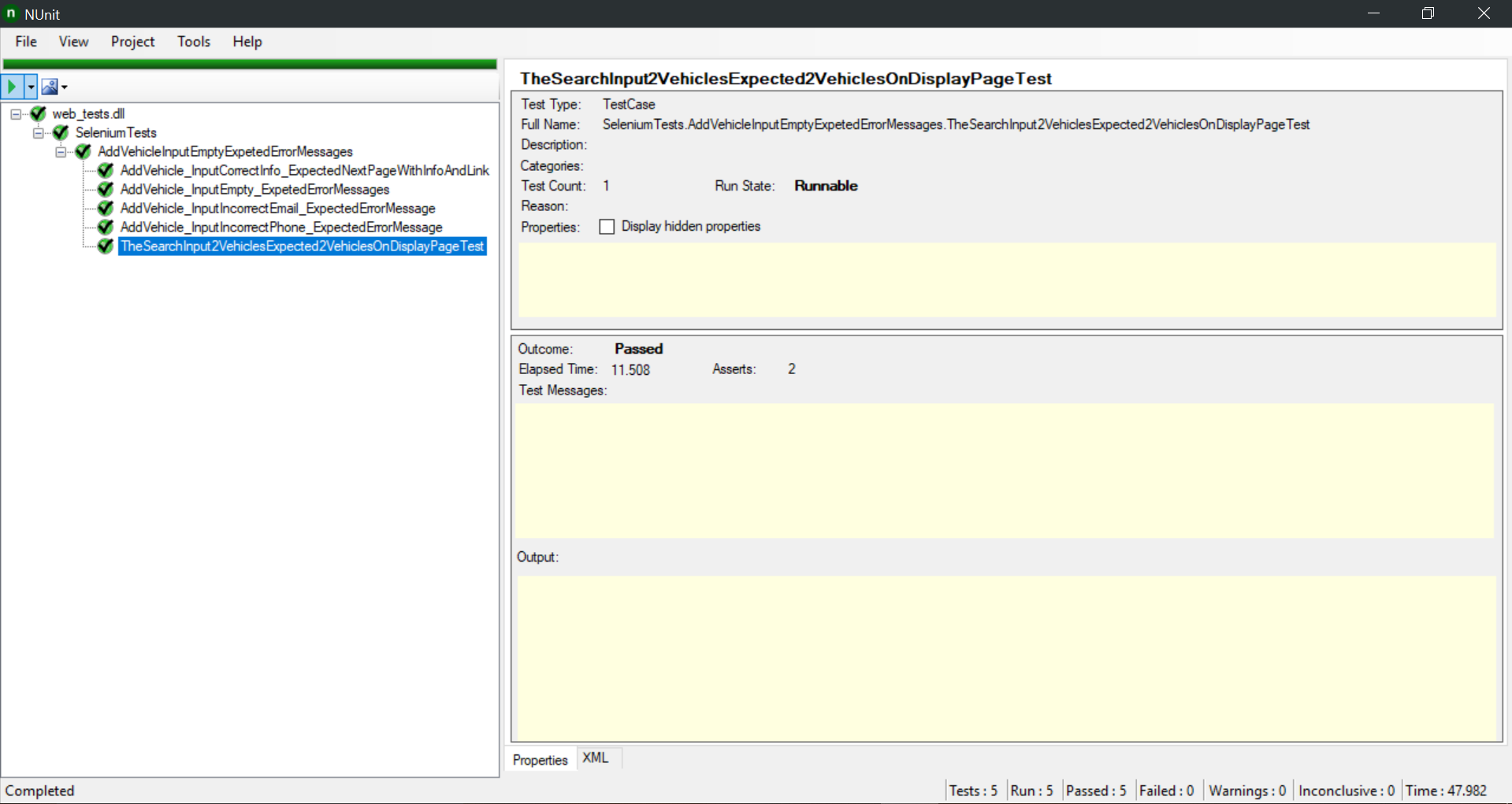
padding: 15px 32px;

font-size: 24px;

margin-top: 20px;

}

**NUnit**



**Git**

