using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace UnitTestProject

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void addSalesAlterInventory() //addSales Class

{

string linesProducts = "PD000001,Melatonin 90 Tablets(Homeopathic Formula),Y,24.5,2344";

string prodID = "PD000001";

string quantity = "30";

string[] productSplit = linesProducts.Split(',');

if (productSplit[0] == prodID)

{

if ((int.Parse(productSplit[4]) - int.Parse(quantity)) >= 0)

{

productSplit[4] = (int.Parse(productSplit[4]) - int.Parse(quantity)).ToString();

string alteredEntry = productSplit[0] + "," + productSplit[1] + "," + productSplit[2] + "," + productSplit[3] + "," + productSplit[4];

linesProducts = alteredEntry;

//writeToFile("productRecords.txt", linesProducts);

}

}

Assert.AreEqual(linesProducts, "PD000001,Melatonin 90 Tablets(Homeopathic Formula),Y,24.5,2314");

}

[TestMethod]

public void inboxStockAlert() //Inbox class

{

string strLines = "PD000001,Melatonin 90 Tablets(Homeopathic Formula),Y,24.5,9";

string[] entry = strLines.Split(',');

string message = "";

if (int.Parse(entry[4]) < 10) // replace array element with corresponding column for AMOUNT

{

message = "ALERT: " + "Low stock of " + entry[1] + " - " + entry[4] + " in stock"; // replace array elements with corresponding column for ITEM NAME and AMOUNT

//MessageBox.Show(message);

//addProductToLowStockTable(entry);

}

Assert.AreEqual(message, "ALERT: Low stock of Melatonin 90 Tablets(Homeopathic Formula) - 9 in stock");

}

[TestMethod]

public void loginTrueTest() //Login class

{

bool match = false;

string user = "Billy";

string pass = "Bob";

//initialises the new registaation data into one string array

string login\_attempt = user + "," + pass;

//Open the registeredUsers.txt file

string regUserExample = "Billy,Bob";

if (regUserExample == login\_attempt)

{

match = true;

}

Assert.AreEqual(match, true);

}

[TestMethod]

public void loginFalseTest() //Login class

{

bool match = false;

string user = "Ross";

string pass = "Bob";

//initialises the new registaation data into one string array

string login\_attempt = user + "," + pass;

//Open the registeredUsers.txt file

string regUserExample = "Billy,Bob";

if (regUserExample == login\_attempt)

{

match = true;

}

Assert.AreEqual(match, false);

}

[TestMethod]

public void exportReportTest() //Report class

{

string yearReport = "2020";

string reportFilePath;

string[,] reportGrid = new string[2,13] { {"Melatonin 90 Tablets (Homeopathic Formula)","$294","$0","$0","$0","$3234","$0","$0","$0","$735","$0","$0","$0" },

{"Astragalus 90 Vegan Capsules","$59.9","$0","$0","$0","$0","$359.4","$688.85","$688.85","$389.35","$29.95","$0","$149.75" }};

string reportFileContents;

switch (yearReport)

{

case "2018":

reportFilePath = "MonthlySalesReport2018.csv";

break;

case "2019":

reportFilePath = "MonthlySalesReport2019.csv";

break;

case "2020":

reportFilePath = "MonthlySalesReport2020.csv";

break;

default:

reportFilePath = "MonthlySalesReport2020.csv";

break;

}

Assert.AreEqual(reportFilePath, "MonthlySalesReport2020.csv");

reportFileContents = "Product,Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec\n";

for (int i = 0; i < reportGrid.GetLength(0); i++)

{

for (int j = 0; j < reportGrid.GetLength(1); j++)

{

reportFileContents += reportGrid[i,j].ToString();

if (j != 12)

{

reportFileContents += ",";

}

}

reportFileContents += "\n";

}

Assert.AreEqual(reportFileContents, "Product,Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec\nMelatonin 90 Tablets (Homeopathic Formula),$294,$0,$0,$0,$3234,$0,$0,$0,$735,$0,$0,$0\nAstragalus 90 Vegan Capsules,$59.9,$0,$0,$0,$0,$359.4,$688.85,$688.85,$389.35,$29.95,$0,$149.75\n");

}

[TestMethod]

public void saveTest() //Sales class

{

string[,] salesGrid = new string[2, 4] {{ "S000000001","10/09/2020","PD000001","30" },

{ "S000000002","10/06/2020","PD000002","12"}};

string salesEntry = "";

for (int i = 0; i < salesGrid.GetLength(0); i++)

{

for (int j = 0; j < salesGrid.GetLength(1); j++)

{

salesEntry += salesGrid[i,j].ToString();

if (j != 3) { salesEntry += ","; };

}

salesEntry += "\n";

}

Assert.AreEqual(salesEntry, "S000000001,10/09/2020,PD000001,30\nS000000002,10/06/2020,PD000002,12\n");

}

[TestMethod]

public void btn\_registerTrueTest() //Signup class

{

string[] registration\_record = {"Billy","Bob"};

string password = "Bob";

string rePassword = "Bob";

string storedUser = "";

if (password == rePassword)

{

//insert registration into the text file

for (int i = 0; i < registration\_record.GetLength(0); i++)

{

storedUser += registration\_record[i];

if (i != 1)

{

storedUser += ",";

}

else

{

storedUser += "\n";

}

}

}

Assert.AreEqual(storedUser,"Billy,Bob\n");

}

[TestMethod]

public void btn\_registerFalseTest() //Signup class

{

string[] registration\_record = { "Billy", "Bob" };

string password = "Bob";

string rePassword = "Bobo";

string storedUser = "";

if (password == rePassword)

{

//insert registration into the text file

for (int i = 0; i < registration\_record.GetLength(0); i++)

{

storedUser += registration\_record[i];

if (i != 1)

{

storedUser += ",";

}

else

{

storedUser += "\n";

}

}

}

Assert.AreNotEqual(storedUser, "Billy,Bob\n");

}

}

}