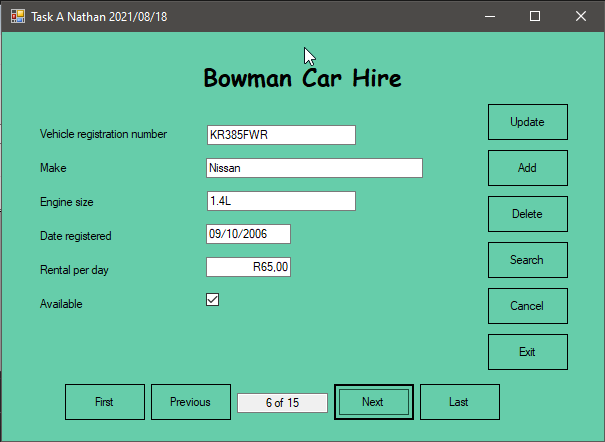
**frmCars**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CarsDatabase

{

public partial class frmCars : Form

{

public frmCars()

{

InitializeComponent();

this.Text = $"Task A Nathan {Convert.ToString(DateTime.Today).Remove(10)}";

}

static string myconnstring = ConfigurationManager.ConnectionStrings["connstring"].ConnectionString;

static int recordsCount;

static int rowNum = 1;

private void frmCars\_Load(object sender, EventArgs e)

{

toolTip1.SetToolTip(txtBoxRegNum, "Enter the vehicle registration number.");

toolTip1.SetToolTip(txtBoxMake, "Enter the make of the vehicle.");

toolTip1.SetToolTip(txtBoxEngineSize, "Enter the engine size of the vehicle in liters");

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

txtBoxRegNum.Text = dataTable.Rows[0].Field<string>(0);

txtBoxMake.Text = dataTable.Rows[0].Field<string>(1);

txtBoxEngineSize.Text = dataTable.Rows[0].Field<string>(2);

txtBoxDateReg.Text = Convert.ToString(dataTable.Rows[0].Field<DateTime>(3).ToString("dd/MM/yyyy"));

txtBoxRentalPDay.Text = Convert.ToString(dataTable.Rows[0].Field<decimal>(4).ToString("C"));

chbAvailable.Checked = dataTable.Rows[0].Field<bool>(5);

RecordNum\_RecordTotal(0);

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

private void btnUpdate\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

try

{

string regNum = txtBoxRegNum.Text;

string make = txtBoxMake.Text;

string engineSize = txtBoxEngineSize.Text;

//--- Begin date formating ---

string temp, tempDay, tempMonth, tempYear;

temp = txtBoxDateReg.Text;

tempDay = txtBoxDateReg.Text.Substring(0, 2);

tempMonth = txtBoxDateReg.Text.Substring(2, 3);

tempYear = txtBoxDateReg.Text.Substring(6, 4);

txtBoxDateReg.Text = tempYear + tempMonth + "/" + tempDay;

DateTime dateReg = Convert.ToDateTime(txtBoxDateReg.Text);

txtBoxDateReg.Text = temp;

//--- End date formating ---

decimal rentalPDay = 0;

//Bug: Will only remove if the rand symbol is present and not other currencies.

// Because the letter "R" can not be converted to a decimal.

if (txtBoxRentalPDay.Text.Contains("R"))

{

txtBoxRentalPDay.Text = txtBoxRentalPDay.Text.Remove(0, 1);

rentalPDay = Convert.ToDecimal(txtBoxRentalPDay.Text);

}

else

{

rentalPDay = Convert.ToDecimal(txtBoxRentalPDay.Text);

}

bool available = chbAvailable.Checked;

string sqlQuery = "UPDATE tblCar SET VehicleRegNo=@vehicleRegNo, Make=@make, EngineSize=@engineSize, DateRegistered=@dateRegistered, RentalPerDay=@rentalPerDay, Available=@available WHERE VehicleRegNo=@vehicleRegNo";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

sqlCon.Open();

cmd.Parameters.AddWithValue("@vehicleRegNo", regNum);

cmd.Parameters.AddWithValue("@make", make);

cmd.Parameters.AddWithValue("@engineSize", engineSize);

cmd.Parameters.AddWithValue("@dateRegistered", dateReg);

cmd.Parameters.AddWithValue("@rentalPerDay", rentalPDay);

cmd.Parameters.AddWithValue("@available", available);

int rows = cmd.ExecuteNonQuery();

//if the query runs succesfully then the value of the rows will be greater than zero else

//its value will be 0

if (rows > 0)

{

MessageBox.Show("Your record has successfully updated", "Success", MessageBoxButtons.OK);

}

else

{

MessageBox.Show("Your record failed to update", "Failure", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

/// <summary>

/// --- Add button to add new records ---

/// Bug: When a record is added it looks at the first column value of the new record and insert it

/// in alphabetical order into the table. This means the new record will not always go to

/// the bottom but in between records.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnAdd\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

string regNum = txtBoxRegNum.Text;

string make = txtBoxMake.Text;

string engineSize = txtBoxEngineSize.Text;

DateTime dateReg = Convert.ToDateTime(txtBoxDateReg.Text);

decimal rentalPDay = 0;

//Bug: Will only remove if the rand symbol is present and not other currencies.

// Because the letter "R" can not be converted to a decimal.

if (txtBoxRentalPDay.Text.Contains("R"))

{

txtBoxRentalPDay.Text = txtBoxRentalPDay.Text.Remove(0, 1);

rentalPDay = Convert.ToDecimal(txtBoxRentalPDay.Text);

}

else

{

rentalPDay = Convert.ToDecimal(txtBoxRentalPDay.Text);

}

bool available = chbAvailable.Checked;

try

{

string sqlQuery = "INSERT INTO tblCar(VehicleRegNo, Make, EngineSize, DateRegistered, RentalPerDay, Available) VALUES (@vehicleRegNo, @make, @engineSize, @dateRegistered, @rentalPerDay, @available)";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

cmd.Parameters.AddWithValue("@vehicleRegNo", regNum);

cmd.Parameters.AddWithValue("@make", make);

cmd.Parameters.AddWithValue("@engineSize", engineSize);

cmd.Parameters.AddWithValue("@dateRegistered", dateReg);

cmd.Parameters.AddWithValue("@rentalPerDay", rentalPDay);

cmd.Parameters.AddWithValue("@available", available);

sqlCon.Open();

int rows = cmd.ExecuteNonQuery();

//if the query runs succesfully then the value of the rows will be greater than zero else

//its value will be 0

if (rows > 0)

{

MessageBox.Show("Your record has successfully been added", "Success", MessageBoxButtons.OK);

}

else

{

MessageBox.Show("Your record failed to add", "Failure", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

//txtBoxRecordNum.Text = $"{rowNum} of {recordsCount + 1}";

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

/// <summary>

/// --- Deletes records ---

/// Bug: If last record is deleted and the previous button is clicked

/// the program will break.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnDelete\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

string regNum = txtBoxRegNum.Text;

try

{

string sqlQuery = "DELETE tblCar WHERE VehicleRegNo=@vehicleRegNo";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

sqlCon.Open();

cmd.Parameters.AddWithValue("@vehicleRegNo", regNum);

int rows = cmd.ExecuteNonQuery();

//if the query runs succesfully then the value of the rows will be greater than zero else

//its value will be 0

if (rows > 0)

{

MessageBox.Show("Your record has successfully been deleted", "Success", MessageBoxButtons.OK);

}

else

{

MessageBox.Show("Your record failed to delete", "Failure", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

private void btnCancel\_Click(object sender, EventArgs e)

{

txtBoxRegNum.Text = "";

txtBoxMake.Text = "";

txtBoxEngineSize.Text = "";

txtBoxDateReg.Text = "";

txtBoxRentalPDay.Text = "";

chbAvailable.Checked = false;

txtBoxRecordNum.Text = "";

}

/// <summary>

///--- Skips to the next records ---

/// Bug: The button must be clicked twice after the previous button was clicked.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnNext\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

if (rowNum < recordsCount)

{

txtBoxRegNum.Text = dataTable.Rows[rowNum].Field<string>(0);

txtBoxMake.Text = dataTable.Rows[rowNum].Field<string>(1);

txtBoxEngineSize.Text = dataTable.Rows[rowNum].Field<string>(2);

txtBoxDateReg.Text = Convert.ToString(dataTable.Rows[rowNum].Field<DateTime>(3).ToString("dd/MM/yyyy"));

txtBoxRentalPDay.Text = Convert.ToString(dataTable.Rows[rowNum].Field<decimal>(4).ToString("C"));

chbAvailable.Checked = dataTable.Rows[rowNum].Field<bool>(5);

rowNum++;

}

else

{

MessageBox.Show("You have reach the final record!");

}

RecordNum\_RecordTotal(0);

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

/// <summary>

/// --- Go's to the previous record ---

/// Bug: The button must be clicked twice after the next button was clicked.

/// Bug: After last record is deleted when previous button is clicked the program

/// will break.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void btnPrevious\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

if (rowNum > 0)

{

rowNum--;

txtBoxRegNum.Text = dataTable.Rows[rowNum].Field<string>(0);

txtBoxMake.Text = dataTable.Rows[rowNum].Field<string>(1);

txtBoxEngineSize.Text = dataTable.Rows[rowNum].Field<string>(2);

txtBoxDateReg.Text = Convert.ToString(dataTable.Rows[rowNum].Field<DateTime>(3).ToString("dd/MM/yyyy"));

txtBoxRentalPDay.Text = Convert.ToString(dataTable.Rows[rowNum].Field<decimal>(4).ToString("C"));

chbAvailable.Checked = dataTable.Rows[rowNum].Field<bool>(5);

}

else

{

MessageBox.Show("You have reach the first record!");

}

RecordNum\_RecordTotal(1);

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

private void btnFirst\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

txtBoxRegNum.Text = dataTable.Rows[0].Field<string>(0);

txtBoxMake.Text = dataTable.Rows[0].Field<string>(1);

txtBoxEngineSize.Text = dataTable.Rows[0].Field<string>(2);

txtBoxDateReg.Text = Convert.ToString(dataTable.Rows[0].Field<DateTime>(3).ToString("dd/MM/yyyy"));

txtBoxRentalPDay.Text = Convert.ToString(dataTable.Rows[0].Field<decimal>(4).ToString("C"));

chbAvailable.Checked = dataTable.Rows[0].Field<bool>(5);

rowNum = 1;

RecordNum\_RecordTotal(0);

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

private void btnLast\_Click(object sender, EventArgs e)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

txtBoxRegNum.Text = dataTable.Rows[recordsCount - 1].Field<string>(0);

txtBoxMake.Text = dataTable.Rows[recordsCount - 1].Field<string>(1);

txtBoxEngineSize.Text = dataTable.Rows[recordsCount - 1].Field<string>(2);

txtBoxDateReg.Text = Convert.ToString(dataTable.Rows[recordsCount - 1].Field<DateTime>(3).ToString("dd/MM/yyyy"));

txtBoxRentalPDay.Text = Convert.ToString(dataTable.Rows[recordsCount - 1].Field<decimal>(4).ToString("C"));

chbAvailable.Checked = dataTable.Rows[recordsCount - 1].Field<bool>(5);

rowNum = recordsCount;

RecordNum\_RecordTotal(0);

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

#region Methods

/// <summary>

/// Method to display the record number and records total at the bottom textbox.

/// </summary>

/// <param name="add">some buttons needs to add a amount to the record number</param>

public void RecordNum\_RecordTotal(int add)

{

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

string sqlQuery = "SELECT \* FROM tblCar";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

adapter.Fill(dataTable);

recordsCount = dataTable.Rows.Count;

txtBoxRecordNum.Text = $"{rowNum + add} of {recordsCount}";

}

#endregion

private void btnExit\_Click(object sender, EventArgs e)

{

const string message = "Do you want to Exit?";

const string caption = "Exit App";

var result = MessageBox.Show(message, caption, MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

Application.Exit();

}

}

private void btnSearch\_Click(object sender, EventArgs e)

{

frmSearch search = new frmSearch();

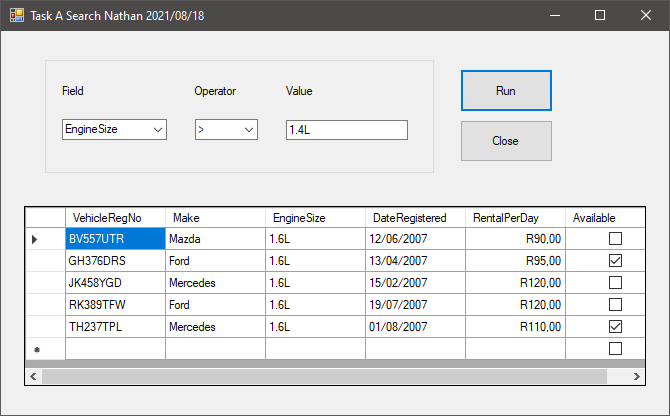
search.Show();

this.Hide();

}

}

}

**frmSearch**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CarsDatabase

{

public partial class frmSearch : Form

{

public frmSearch()

{

InitializeComponent();

this.Text = $"Task A Search Nathan {Convert.ToString(DateTime.Today).Remove(10)}";

}

static string myconnstring = ConfigurationManager.ConnectionStrings["connstring"].ConnectionString;

private void frmSearch\_Load(object sender, EventArgs e)

{

cboField.Items.Add("VehicleRegNo");

cboField.Items.Add("Make");

cboField.Items.Add("EngineSize");

cboField.Items.Add("RentalPerDay ");

cboField.Items.Add("Available");

cboOperator.Items.Add("=");

cboOperator.Items.Add("<");

cboOperator.Items.Add(">");

cboOperator.Items.Add("<=");

cboOperator.Items.Add(">=");

}

private void btnRun\_Click(object sender, EventArgs e)

{

string field, sOperator, value;

field = cboField.Text;

sOperator = cboOperator.Text;

value = txtBoxValue.Text;

//Changing the available value field to accept Yes and No values

if (field == "Available" && value == "Yes")

{

value = true.ToString();

}

else if (field == "Available" && value == "No")

{

value = false.ToString();

}

SqlConnection sqlCon = new SqlConnection(myconnstring);

DataTable dataTable = new DataTable();

try

{

string sqlQuery = "SELECT \* FROM tblCar WHERE " + field + sOperator + "@Value";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

sqlCon.Open();

cmd.Parameters.AddWithValue("@Value", value);

adapter.Fill(dataTable);

//if satement for preventing unnecessary result errors from displaying

if (field == "VehicleRegNo" && sOperator != "=" || field == "Make" && sOperator != "=" || field == "Available" && sOperator != "=")

{

MessageBox.Show($"You can't use the {field} field with the {sOperator} operator","Invalid Input",MessageBoxButtons.OK);

dataTable.Clear();

}

dgvCars.DataSource = dataTable;

//Changing the display format of the date registered and rental per day displayed

dgvCars.Columns[3].DefaultCellStyle.Format = "dd/MM/yyyy";

dgvCars.Columns[4].DefaultCellStyle.Format = "C";

dgvCars.Columns[4].DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;

}

catch (SqlException sqlEx)

{

MessageBox.Show(sqlEx.Message);

}

finally

{

sqlCon.Close();

}

}

private void btnClose\_Click(object sender, EventArgs e)

{

frmCars cars = new frmCars();

cars.Show();

this.Hide();

}

}

}