# Home Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmHomePage : Form

{

public frmHomePage()

{

InitializeComponent();

}

//Declaring high contrast - automatically set to false

bool highContrast = false;

private void btnSignUp\_MouseEnter(object sender, EventArgs e)

{

//Buttons change colour when hovered over, only if high contrast mode is turned off

if (highContrast == false)

{

btnSignUp.BackColor = Color.FromArgb(103, 173, 136);

btnSignUp.ForeColor = Color.FromArgb(179, 214, 196);

}

}

private void btnSignUp\_MouseLeave(object sender, EventArgs e)

{

// When mouse moves away, sign up button will revert back to original colour

if (highContrast == false)

{

btnSignUp.BackColor = Color.FromArgb(179, 214, 196);

btnSignUp.ForeColor = Color.FromArgb(103, 173, 136);

}

}

private void btnLogIn\_MouseEnter(object sender, EventArgs e)

{

if (highContrast == false)

{

btnLogIn.BackColor = Color.FromArgb(179, 214, 196);

btnLogIn.ForeColor = Color.FromArgb(103, 173, 136);

}

}

private void btnLogIn\_MouseLeave(object sender, EventArgs e)

{

if (highContrast == false)

{

btnLogIn.BackColor = Color.FromArgb(103, 173, 136);

btnLogIn.ForeColor = Color.FromArgb(179, 214, 196);

}

}

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

{

//Declaring sign up page

frmSignUpPage frmSignUp = new frmSignUpPage();

//Opening sign up, closing home page

frmSignUp.Show();

this.Close();

}

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

{

//Declaring log in page

frmLogIn frmLogInPage = new frmLogIn();

//Opening log in, closing home page

frmLogInPage.Show();

this.Close();

}

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

{

//Declaring green energy page

frmGreenEnergyProducts frmGreenEnergyProductsPage = new frmGreenEnergyProducts();

//Opening green energy products, closing home page

frmGreenEnergyProductsPage.Show();

this.Close();

}

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

{

//Declaring carbon footprint page

frmCarbonFootprintCalculator frmCarbonFootprintPage = new frmCarbonFootprintCalculator();

//Opening carbon footprint, closing home page

frmCarbonFootprintPage.Show();

this.Close();

}

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

{

//Declaring booking page

frmBookingPage frmToBookingPage = new frmBookingPage();

//Opening booking, closing home page

frmToBookingPage.Show();

this.Close();

}

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

{

//Declaring energy usage page

frmEnergyUsageCalculator frmToEnergyUsageCalculator = new frmEnergyUsageCalculator();

//opening energy usage, closing home page

frmToEnergyUsageCalculator.Show();

this.Close();

}

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

{

//Declaring settings page

var frmToSettings = new frmSettingsRedo();

//Settings page will open in same location

frmToSettings.Location = this.Location;

frmToSettings.StartPosition = FormStartPosition.Manual;

//Home page will show again when settings page is closed

frmToSettings.FormClosing += delegate { this.Show(); };

//settings page is shown and home page is hidden - not completely closed

frmToSettings.Show();

this.Hide();

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = Color.FromArgb(158, 202, 179);

pbNavbar.BackColor = Color.FromArgb(103, 173, 136);

btnGreenEnergyProducts.BackColor = Color.FromArgb(103, 173, 136);

btnCarbonFootprint.BackColor = Color.FromArgb(103, 173, 136);

btnBooking.BackColor = Color.FromArgb(103, 173, 136);

btnEnergyUsage.BackColor = Color.FromArgb(103, 173, 136);

btnSettings.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.ForeColor = Color.FromArgb(179, 214, 196);

pbLogo.BackColor = Color.FromArgb(103, 173, 136);

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

lblRolsaTechnologies.BackColor = Color.FromArgb(103, 173, 136);

lblHeader1.ForeColor = Color.FromArgb(65, 119, 91);

lblHeader2.ForeColor = Color.FromArgb(103, 173, 136);

lblHeader3.ForeColor = Color.White;

btnSignUp.BackColor = Color.FromArgb(179, 214, 196);

btnSignUp.ForeColor = Color.FromArgb(103, 173, 136);

btnLogIn.BackColor = Color.FromArgb(103, 173, 136);

btnLogIn.ForeColor = Color.FromArgb(179, 214, 196);

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

pbNavbar.BackColor = Color.Blue;

btnGreenEnergyProducts.BackColor = Color.Blue;

btnCarbonFootprint.BackColor = Color.Blue;

btnBooking.BackColor = Color.Blue;

btnEnergyUsage.BackColor = Color.Blue;

btnSettings.BackColor = Color.Blue;

lblSlogan.BackColor = Color.Blue;

lblSlogan.ForeColor = Color.White;

pbLogo.BackColor = Color.Blue;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

lblRolsaTechnologies.BackColor = Color.Blue;

lblHeader1.ForeColor = Color.Black;

lblHeader2.ForeColor = Color.Black;

lblHeader3.ForeColor = Color.Black;

btnSignUp.BackColor = Color.Black;

btnSignUp.ForeColor = Color.Yellow;

btnLogIn.BackColor = Color.Black;

btnLogIn.ForeColor = Color.Yellow;

highContrast = true;

}

}

}

}

# Sign Up Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Text;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmSignUpPage : Form

{

public frmSignUpPage()

{

InitializeComponent();

}

bool highContrast = false;

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

{

//Declaring database

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//Declaring username and passwords

string username = txtUsername.Text;

string password1 = txtPassword.Text;

string password2 = txtConfirmPassword.Text;

//Creating new command to search through stored usernames

SqlCommand checkUsernameCommand = new SqlCommand("SELECT COUNT(\*) FROM Users WHERE Username = @Username", sqlConnection);

checkUsernameCommand.Parameters.AddWithValue("@Username", username);

//Opening database connection

sqlConnection.Open();

int usernameCount = (int)checkUsernameCommand.ExecuteScalar();

//Closing database connection

sqlConnection.Close();

//Ensuring inputs can not be null

if (string.IsNullOrEmpty(txtUsername.Text) || string.IsNullOrEmpty(txtPassword.Text) || string.IsNullOrEmpty(txtConfirmPassword.Text))

{

txtErrorMessage1.Text = "Username and password can not be empty, please try again!";

txtErrorMessage1.Visible = true;

grbCorrectCredentials.Visible = false;

}

else

{

if (usernameCount > 0) //If username >0, means username already exists

{

txtErrorMessage1.Text = "Username has already been taken, please try again!";

txtErrorMessage1.Visible = true;

grbCorrectCredentials.Visible = false;

}

else

{

if (password1 != password2) //If password1 != password2, means passwords do not match

{

//Makes sets error message and makes it visible

txtErrorMessage1.Text = "Both passwords do not match, please try again!";

txtErrorMessage1.Visible = true;

grbCorrectCredentials.Enabled = false;

grbCorrectCredentials.Visible = false;

}

else

{

//Makes group box visible

string password = password1;

txtErrorMessage1.Visible = false;

grbCorrectCredentials.Visible = true;

txtUsername.ReadOnly = true;

txtPassword.ReadOnly = true;

txtConfirmPassword.ReadOnly = true;

}

}

}

}

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

{

if (string.IsNullOrEmpty(txtEmail.Text) || string.IsNullOrEmpty(txtPhoneNum.Text) || string.IsNullOrEmpty(txtAddress.Text))

{

txtErrorMessage2.Text = "An account can not be created without these details, please try again!";

txtErrorMessage2.Visible = true;

}

else

{

//Declaring database again

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//Calling the 'CreateNewUserRecord' procedure

SqlCommand command = new SqlCommand("CreateNewUserRecord", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

string username = txtUsername.Text;

string password = txtPassword.Text;

string DOB = dtpDOB.Text;

string email = txtEmail.Text;

string phoneNum = txtPhoneNum.Text;

string address = txtAddress.Text;

//validating email

int countEmail = 0;

//Checking if email ends in ".com"

string hasDotCom = email.Substring(email.Length - 4);

//Checking to see if the email has an @ symbol in it

foreach (char c in email)

{

if (c == '@')

{

countEmail = countEmail + 1;

}

}

//Checking to see if phone number is made up of integers

int countPhoneNumDigits = 0;

foreach (char character in phoneNum)

{

if (character <= 0 || character >= 9)

{

countPhoneNumDigits = countPhoneNumDigits + 1;

}

}

bool allDigits = false;

if (countPhoneNumDigits == phoneNum.Length)

{

//if amount of digits counted in foreach loop is equal to number of characters in the string

allDigits = true;

}

else

{

//if amount of digits counted in foreach loop is not equal to number of characters in the string

allDigits = false;

}

if (countEmail == 1 && hasDotCom == ".com")

{

if (phoneNum.Length == 11 && allDigits == true)

{

//Adding all values to database

command.Parameters.AddWithValue("@Username", username);

command.Parameters.AddWithValue("@Password", password);

command.Parameters.AddWithValue("@DOB", DOB);

command.Parameters.AddWithValue("@Email", email);

command.Parameters.AddWithValue("@PhoneNumber", phoneNum);

command.Parameters.AddWithValue("@Address", address);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

txtErrorMessage2.Visible = false;

MessageBox.Show("Account has been created successfully");

frmHomePage frmToHome = new frmHomePage();

frmToHome.Show();

this.Close();

}

else

{

//Show if phone num is not valid

txtErrorMessage2.Text = "Phone number is not valid, please try again!";

txtErrorMessage2.Visible = true;

}

}

else

{

//Show if email is not valid

txtErrorMessage2.Text = "Email is not valid, please try again!";

txtErrorMessage2.Visible = true;

}

}

}

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

{

//Declaring home page

frmHomePage frmToHomePage = new frmHomePage();

//Opening home page, closing sign up page

frmToHomePage.Show();

this.Close();

}

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

{

//Declaring log in page

frmLogIn frmToLogIn = new frmLogIn();

//Opening home page, closing sign up page

frmToLogIn.Show();

this.Close();

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = SystemColors.Control;

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

btnToLogIn.BackColor = SystemColors.Window;

btnToLogIn.ForeColor = SystemColors.WindowText;

pbBackground1.BackColor = Color.FromArgb(158, 202, 179);

pbBackground2.BackColor = Color.FromArgb(158, 202, 179);

btnToHome.ForeColor = Color.FromArgb(103, 173, 136);

lblSignUp.ForeColor = Color.FromArgb(158, 202, 179);

lblUsername.ForeColor = Color.FromArgb(103, 173, 136);

txtUsername.BackColor = SystemColors.Window;

txtUsername.ForeColor = SystemColors.WindowText;

lblPassword.ForeColor = Color.FromArgb(103, 173, 136);

txtPassword.BackColor = SystemColors.Window;

txtPassword.ForeColor = SystemColors.WindowText;

lblConfirmPassword.ForeColor = Color.FromArgb(103, 173, 136);

txtConfirmPassword.BackColor = SystemColors.Window;

txtConfirmPassword.ForeColor = SystemColors.WindowText;

btnCheckCredentials.BackColor = SystemColors.Window;

btnCheckCredentials.ForeColor = SystemColors.WindowText;

txtErrorMessage1.BackColor = SystemColors.Window;

txtErrorMessage1.ForeColor = SystemColors.WindowText;

//Group box

lblDOB.ForeColor = Color.FromArgb(103, 173, 136);

dtpDOB.CalendarMonthBackground = SystemColors.Window;

dtpDOB.CalendarForeColor = SystemColors.WindowText;

lblEmail.ForeColor = Color.FromArgb(103, 173, 136);

txtEmail.BackColor = SystemColors.Window;

txtEmail.ForeColor = SystemColors.WindowText;

lblPhoneNum.ForeColor = Color.FromArgb(103, 173, 136);

txtPhoneNum.BackColor = SystemColors.Window;

txtPhoneNum.ForeColor = SystemColors.WindowText;

lblAddress.ForeColor = Color.FromArgb(103, 173, 136);

txtAddress.BackColor = SystemColors.Window;

txtAddress.ForeColor = SystemColors.WindowText;

btnCreateAccount.BackColor = SystemColors.Window;

btnCreateAccount.ForeColor = SystemColors.WindowText;

txtErrorMessage2.BackColor = SystemColors.Window;

txtErrorMessage2.ForeColor = SystemColors.WindowText;

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

btnToLogIn.BackColor = Color.Black;

btnToLogIn.ForeColor = Color.Yellow;

pbBackground1.BackColor = Color.Blue;

pbBackground2.BackColor = Color.Blue;

btnToHome.ForeColor = Color.Black;

lblSignUp.ForeColor = Color.Black;

lblUsername.ForeColor = Color.Black;

txtUsername.BackColor = Color.Black;

txtUsername.ForeColor = Color.Yellow;

lblPassword.ForeColor = Color.Black;

txtPassword.BackColor = Color.Black;

txtPassword.ForeColor = Color.Yellow;

lblConfirmPassword.ForeColor = Color.Black;

txtConfirmPassword.BackColor = Color.Black;

txtConfirmPassword.ForeColor = Color.Yellow;

btnCheckCredentials.BackColor = Color.Black;

btnCheckCredentials.ForeColor = Color.Yellow;

txtErrorMessage1.BackColor = Color.Black;

txtErrorMessage1.ForeColor = Color.Yellow;

//Group Box

lblDOB.ForeColor = Color.Black;

dtpDOB.CalendarMonthBackground = Color.Black;

dtpDOB.CalendarForeColor = Color.Yellow;

lblEmail.ForeColor = Color.Black;

txtEmail.BackColor = Color.Black;

txtEmail.ForeColor = Color.Yellow;

lblPhoneNum.ForeColor = Color.Black;

txtPhoneNum.BackColor = Color.Black;

txtPhoneNum.ForeColor = Color.Yellow;

lblAddress.ForeColor = Color.Black;

txtAddress.BackColor = Color.Black;

txtAddress.ForeColor = Color.Yellow;

btnCreateAccount.BackColor = Color.Black;

btnCreateAccount.ForeColor = Color.Yellow;

txtErrorMessage2.BackColor = Color.Black;

txtErrorMessage2.ForeColor = Color.Yellow;

highContrast = true;

}

}

}

}

# Log In Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

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 RolsaRechnologiesSolutionV1

{

public partial class frmLogIn : Form

{

public frmLogIn()

{

InitializeComponent();

}

//declaring high contrast

bool highContrast = false;

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

{

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//Declaring username and password

string username = txtUsername.Text;

string password = txtPassword.Text;

SqlCommand checkUsernameAndPasswordCommand = new SqlCommand("SELECT COUNT(\*) FROM Users WHERE Username = @Username AND Password = @Password", sqlConnection);

//Comparing username and password to fields in the Users database

checkUsernameAndPasswordCommand.Parameters.AddWithValue("@Username", username);

checkUsernameAndPasswordCommand.Parameters.AddWithValue("@Password", password);

sqlConnection.Open();

int matchingCount = (int)checkUsernameAndPasswordCommand.ExecuteScalar();

sqlConnection.Close();

if (matchingCount == 1) //Means the system has found a match

{

txtErrorMessage1.Visible = false;

MessageBox.Show("Logged in");

//Declaring home page

frmHomePage frmLoggedInHomepage = new frmHomePage();

//Opening home page, closing log in page

frmLoggedInHomepage.Show();

this.Close();

}

else //Means the system has not found a match (or multiple which implies an error within the sign up system)

{

txtErrorMessage1.Text = "Invalid credentials entered";

txtErrorMessage1.Show();

}

}

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

{

//Declaring sign up page

frmSignUpPage frmToSignUp = new frmSignUpPage();

//Opening sign up page, closing log in page

frmToSignUp.Show();

this.Close();

}

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

{

frmHomePage frmToHomePage = new frmHomePage();

frmToHomePage.Show();

this.Close();

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = SystemColors.Control;

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

btnToSignUp.BackColor = SystemColors.Window;

btnToSignUp.ForeColor = SystemColors.WindowText;

pbBackground1.BackColor = Color.FromArgb(158, 202, 179);

pbBackground2.BackColor = Color.FromArgb(158, 202, 179);

btnToHome.ForeColor = Color.FromArgb(103, 173, 136);

lblLogIn.ForeColor = Color.FromArgb(158, 202, 179);

lblUsername.ForeColor = Color.FromArgb(103, 173, 136);

txtUsername.BackColor = SystemColors.Window;

txtUsername.ForeColor = SystemColors.WindowText;

lblPassword.ForeColor = Color.FromArgb(103, 173, 136);

txtPassword.BackColor = SystemColors.Window;

txtPassword.ForeColor = SystemColors.WindowText;

txtErrorMessage1.BackColor = SystemColors.Window;

txtErrorMessage1.ForeColor = SystemColors.WindowText;

btnLogIn.BackColor = SystemColors.Window;

btnLogIn.ForeColor = SystemColors.WindowText;

lblNoAccount.ForeColor = Color.FromArgb(158, 202, 179);

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

btnToSignUp.BackColor = Color.Black;

btnToSignUp.ForeColor = Color.Yellow;

pbBackground1.BackColor = Color.Blue;

pbBackground2.BackColor = Color.Blue;

btnToHome.ForeColor = Color.Black;

lblLogIn.ForeColor = Color.Black;

lblUsername.ForeColor = Color.Black;

txtUsername.BackColor = Color.Black;

txtUsername.ForeColor = Color.Yellow;

lblPassword.ForeColor = Color.Black;

txtPassword.BackColor = Color.Black;

txtPassword.ForeColor = Color.Yellow;

txtErrorMessage1.BackColor = Color.Black;

txtErrorMessage1.ForeColor = Color.Yellow;

btnLogIn.BackColor = Color.Black;

btnLogIn.ForeColor = Color.Yellow;

lblNoAccount.ForeColor = Color.Black;

highContrast = true;

}

}

}

}

# Green Energy Products Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmGreenEnergyProducts : Form

{

public frmGreenEnergyProducts()

{

InitializeComponent();

//Combo box variable automatically set to sans serif

cmbFonts.SelectedItem = "Microsoft Sans Serif";

//font size automatically displayed in text box when page is opened

txtTextSize.Text = fontSize.ToString();

}

//setting default font size to 8

int fontSize = 8;

//declaring high contrast

bool highContrast = false;

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

{

//Declaring home page

frmHomePage frmToHomePage = new frmHomePage();

//Opening home, closing green energy page

frmToHomePage.Show();

this.Close();

}

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

{

//Declaring carbon footprint page

frmCarbonFootprintCalculator frmToCarbonFootprint = new frmCarbonFootprintCalculator();

//opening carbon footprint page, closing green energy page

frmToCarbonFootprint.Show();

this.Close();

}

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

{

//Declaring booking page

frmBookingPage frmToBookingPage = new frmBookingPage();

//opening booking page, closing green energy page

frmToBookingPage.Show();

this.Close();

}

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

{

//Declaring energy usage page

frmEnergyUsageCalculator frmToEnergyUsageCalculator = new frmEnergyUsageCalculator();

//opening energy usage page, closing green energy page

frmToEnergyUsageCalculator.Show();

this.Close();

}

//Happens as soon as combo box value is changed

private void cmbFonts\_SelectedIndexChanged(object sender, EventArgs e)

{

//setting user chosen font to font chosen in combo box

string userChosenFont = cmbFonts.Text;

if (userChosenFont == "Arial")

{

txtGreenEnergyInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtGreenEnergyInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtGreenEnergyInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtGreenEnergyInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtGreenEnergyInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtGreenEnergyInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtGreenEnergyInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

//Error checking

MessageBox.Show("Error");

}

}

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

{

//setting user chosen font to font chosen in combo box - "Font()" requires 2 parameteres, so font is required to change text size

string userChosenFont = cmbFonts.Text;

//Ensuring font size does not go below 1

if (fontSize <= 1)

{

MessageBox.Show("Your font size can not be below 1");

}

else

{

//Subtracts 1 from font size

fontSize = fontSize - 1;

//displaying font size in text box

txtTextSize.Text = fontSize.ToString();

if (userChosenFont == "Arial")

{

txtGreenEnergyInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtGreenEnergyInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtGreenEnergyInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtGreenEnergyInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtGreenEnergyInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtGreenEnergyInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtGreenEnergyInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

//Error checking

MessageBox.Show("Error");

}

}

}

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

{

string userChosenFont = cmbFonts.Text;

//Adding 1 to font size

fontSize = fontSize + 1;

//displaying font size in text box

txtTextSize.Text = fontSize.ToString();

if (userChosenFont == "Arial")

{

txtGreenEnergyInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtGreenEnergyInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtGreenEnergyInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtGreenEnergyInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtGreenEnergyInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtGreenEnergyInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtGreenEnergyInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

MessageBox.Show("Error");

}

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = Color.FromArgb(158, 202, 179);

txtGreenEnergyInformation.BackColor = SystemColors.Window;

txtGreenEnergyInformation.ForeColor = SystemColors.WindowText;

pbNavbar.BackColor = Color.FromArgb(103, 173, 136);

pbGreenEnergyProducts.BackColor = Color.FromArgb(103, 173, 136);

btnCarbonFootprint.BackColor = Color.FromArgb(103, 173, 136);

btnBooking.BackColor = Color.FromArgb(103, 173, 136);

btnEnergyUsage.BackColor = Color.FromArgb(103, 173, 136);

btnSettings.BackColor = Color.FromArgb(103, 173, 136);

btnToHome.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.ForeColor = Color.FromArgb(179, 214, 196);

pbLogo.BackColor = Color.FromArgb(103, 173, 136);

lblGreenEnergyProducts.ForeColor = Color.White;

cmbFonts.BackColor = SystemColors.Window;

cmbFonts.ForeColor = SystemColors.WindowText;

btnRaiseFont.BackColor = SystemColors.Window;

btnRaiseFont.ForeColor = SystemColors.WindowText;

btnLowerFont.BackColor = SystemColors.Window;

btnLowerFont.ForeColor = SystemColors.WindowText;

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

txtTextSize.BackColor = SystemColors.Window;

txtTextSize.ForeColor = SystemColors.WindowText;

lblChangeFont.BackColor = Color.FromArgb(158, 202, 179);

lblChangeFont.ForeColor = Color.White;

lblTextSize.BackColor = Color.FromArgb(158, 202, 179);

lblTextSize.ForeColor = Color.White;

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

txtGreenEnergyInformation.BackColor = Color.Black;

txtGreenEnergyInformation.ForeColor = Color.Yellow;

pbNavbar.BackColor = Color.Blue;

pbGreenEnergyProducts.BackColor = Color.Blue;

btnCarbonFootprint.BackColor = Color.Blue;

btnBooking.BackColor = Color.Blue;

btnEnergyUsage.BackColor = Color.Blue;

btnSettings.BackColor = Color.Blue;

btnToHome.BackColor = Color.Blue;

lblSlogan.BackColor = Color.Blue;

lblSlogan.ForeColor = Color.White;

pbLogo.BackColor = Color.Blue;

lblGreenEnergyProducts.ForeColor = Color.Black;

cmbFonts.BackColor = Color.Black;

cmbFonts.ForeColor = Color.Yellow;

btnRaiseFont.BackColor = Color.Black;

btnRaiseFont.ForeColor = Color.Yellow;

btnLowerFont.BackColor = Color.Black;

btnLowerFont.ForeColor = Color.Yellow;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

txtTextSize.BackColor = Color.Black;

txtTextSize.ForeColor = Color.Yellow;

lblChangeFont.BackColor = Color.Black;

lblChangeFont.ForeColor = Color.Yellow;

lblTextSize.BackColor = Color.Black;

lblTextSize.ForeColor = Color.Yellow;

highContrast = true;

}

}

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

{

//Declaring settings page

var frmToSettings = new frmSettingsRedo();

//Settings page will open in same location

frmToSettings.Location = this.Location;

frmToSettings.StartPosition = FormStartPosition.Manual;

//Green energy page will show again when settings page is closed

frmToSettings.FormClosing += delegate { this.Show(); };

//settings page is shown and green energy page is hidden - not completely closed

frmToSettings.Show();

this.Hide();

}

}

}

# Carbon Footprint Calculator Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmCarbonFootprintCalculator : Form

{

public frmCarbonFootprintCalculator()

{

InitializeComponent();

//Combo box variable automatically set to sans serif

cmbFonts.SelectedItem = "Microsoft Sans Serif";

//font size automatically displayed in text box when page is opened

txtTextSize.Text = fontSize.ToString();

}

//setting default font size to 8

int fontSize = 8;

//declaring high contrast

bool highContrast = false;

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

{

//Declaring green energy page

frmGreenEnergyProducts frmToGreenEnergyProducts = new frmGreenEnergyProducts();

//Opening green energy page, closing carbon footprint page

frmToGreenEnergyProducts.Show();

this.Close();

}

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

{

//declaring following variables

string electricBill, gasBill, oilBill, mileage, numFlightsUnderFour, numFlightsOverFour, recycleNewspaper, recycleAluminumTin;

decimal calculation = 0;

bool validInputs = true;

//assigning all variables corresponding inputs

electricBill = txtElectricBill.Text;

gasBill = txtGasBill.Text;

oilBill = txtOilBill.Text;

mileage = txtMileage.Text;

numFlightsUnderFour = txtNumOfFlightsUnderFour.Text;

numFlightsOverFour = txtNumOfFlightsOverFour.Text;

recycleNewspaper = cmbRecycleNewspaper.Text;

recycleAluminumTin = cmbRecycleAluminumTin.Text;

//declaring new variables used for calculations

decimal newElectricBill, newGasBill, newOilBill;

int newMileage, newFlightsUnderFour, newFlightsOverFour;

//Checking if electric bill is number

if (decimal.TryParse(electricBill, out decimal electricValue))

{

newElectricBill = decimal.Parse(electricBill);

//Checking if gas bill is number

if (decimal.TryParse(gasBill, out decimal gasValue))

{

newGasBill = decimal.Parse(gasBill);

//Checking if oil bill is number

if (decimal.TryParse(oilBill, out decimal oilValue))

{

newOilBill = decimal.Parse(oilBill);

//Checking if mileage is whole number

if (int.TryParse(mileage, out int mileageValue))

{

newMileage = int.Parse(mileage);

//Checking if number of flights under four hours is a whole number

if (int.TryParse(numFlightsUnderFour, out int numFlightsUnderFourValue))

{

newFlightsUnderFour = int.Parse(numFlightsUnderFour);

//Checking if number of flights over four hours is a whole number

if (int.TryParse(numFlightsOverFour, out int numFlightsOverFourValue))

{

newFlightsOverFour = int.Parse(numFlightsOverFour);

//Checking if recycle newspaper combo box is empty

if (recycleNewspaper != string.Empty)

{

//Checking if recycle aluminum and tin combo box is empty

if (recycleAluminumTin != string.Empty)

{

txtErrorMessage.Hide();

validInputs = true;

if (validInputs == true)

{

//Starts calculation only if all inputs are valid

newElectricBill = newElectricBill \* 105;

calculation = calculation + newElectricBill;

newGasBill = newGasBill \* 105;

calculation = calculation + newGasBill;

newOilBill = newOilBill \* 113;

calculation = calculation + newOilBill;

newMileage = (int)(newMileage \* 0.79);

calculation = calculation + newMileage;

newFlightsUnderFour = newFlightsUnderFour \* 1100;

calculation = calculation + newFlightsUnderFour;

newFlightsOverFour = newFlightsOverFour \* 4400;

calculation = calculation + newFlightsOverFour;

if (recycleNewspaper == "Yes")

{

//If newspaper is recycled, nothing is added to the calculation

if (recycleAluminumTin == "Yes")

{

//If aluminum and tin is recycled, nothing is added to the calculation

if (calculation >= 6000 && calculation <= 15999)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered ideal";

}

else if (calculation >= 16000 && calculation <= 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered average";

}

else if (calculation >= 0 && calculation < 6000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very low";

}

else if (calculation > 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very high";

}

else

{

txtErrorMessage.Text = "Error in calculation process";

}

}

else

{

//166 added if aluminum ans tin is not recycled

calculation = calculation + 166;

//Output for ideal carbon footprint

if (calculation >= 6000 && calculation <= 15999)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered ideal";

}

//Output for average carbon footprint

else if (calculation >= 16000 && calculation <= 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered average";

}

//Output for very low carbon footprint

else if (calculation >= 0 && calculation < 6000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very low";

}

//Output for very high carbon footprint

else if (calculation > 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very high";

}

else

{

//Error checking

txtErrorMessage.Text = "Error in calculation process";

}

}

}

else

{

//184 added to calculation if newspaper is not recycled

calculation = calculation + 184;

if (recycleAluminumTin == "Yes")

{

//Output for ideal carbon footprint

if (calculation >= 6000 && calculation <= 15999)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered ideal";

}

//Output for average carbon footprint

else if (calculation >= 16000 && calculation <= 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered average";

}

//Output for very low carbon footprint

else if (calculation >= 0 && calculation < 6000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very low";

}

//Output for very high carbon footprint

else if (calculation > 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very high";

}

else

{

//Error checking

txtErrorMessage.Text = "Error in calculation process";

}

}

else

{

//166 added if aluminum ans tin is not recycled

calculation = calculation + 166;

//Output for ideal carbon footprint

if (calculation >= 6000 && calculation <= 15999)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered ideal";

}

//Output for average carbon footprint

else if (calculation >= 16000 && calculation <= 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered average";

}

//Output for very low carbon footprint

else if (calculation >= 0 && calculation < 6000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very low";

}

//Output for very high carbon footprint

else if (calculation > 22000)

{

txtResult.Text = "Your carbon footprint = " + calculation + ", this is considered very high";

}

else

{

//Error checking

txtErrorMessage.Text = "Error in calculation process";

}

}

}

}

}

//Error message if combo box is empty is empty

else

{

txtErrorMessage.Text = "You can not leave boxes empty!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if combo box is empty is empty

else

{

txtErrorMessage.Text = "You can not leave boxes empty!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if number of flights over four hours is wrong

else

{

txtErrorMessage.Text = "Number of flights over four hours long must be a whole number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if number of flights under four hours is wrong

else

{

txtErrorMessage.Text = "Number of flights under four hours long must be a whole number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if mileage is wrong

else

{

txtErrorMessage.Text = "Mileage must be a whole number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if oil bill is wrong

else

{

txtErrorMessage.Text = "Oil bill must be a number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if gas bill is wrong

else

{

txtErrorMessage.Text = "Gas bill must be a number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

//Error message if electric bill is wrong

else

{

txtErrorMessage.Text = "Electric bill must be a number, please try again!";

txtErrorMessage.Visible = true;

validInputs = false;

}

}

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

{

//Declaring home page

frmHomePage frmToHomePage = new frmHomePage();

//Opening home page, closing carbon footprint page

frmToHomePage.Show();

this.Close();

}

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

{

//Declaring booking page

frmBookingPage frmToBookingPage = new frmBookingPage();

//Opening booking, closing carbon footprint page

frmToBookingPage.Show();

this.Close();

}

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

{

//Declaring energy usage page

frmEnergyUsageCalculator frmToEnergyUsageCalculator = new frmEnergyUsageCalculator();

//Opening energy usage page, closing carbon footprint page

frmToEnergyUsageCalculator.Show();

this.Close();

}

//Happens as soon as combo box value is changed

private void cmbFonts\_SelectedIndexChanged(object sender, EventArgs e)

{

//setting user chosen font to font chosen in combo box

string userChosenFont = cmbFonts.Text;

if (userChosenFont == "Arial")

{

txtCarbonFootprintInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtCarbonFootprintInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtCarbonFootprintInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtCarbonFootprintInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtCarbonFootprintInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtCarbonFootprintInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtCarbonFootprintInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

//Error checking

MessageBox.Show("Error");

}

}

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

{

//setting user chosen font to font chosen in combo box - "Font()" requires 2 parameteres, so font is required to change text size

string userChosenFont = cmbFonts.Text;

//Ensuring font size does not go below 1

if (fontSize <= 1)

{

MessageBox.Show("Your font size can not be below 1");

}

else

{

//Subtracts 1 from font size

fontSize = fontSize - 1;

//displaying font size in text box

txtTextSize.Text = fontSize.ToString();

if (userChosenFont == "Arial")

{

txtCarbonFootprintInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtCarbonFootprintInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtCarbonFootprintInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtCarbonFootprintInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtCarbonFootprintInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtCarbonFootprintInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtCarbonFootprintInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

//Error checking

MessageBox.Show("Error");

}

}

}

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

{

string userChosenFont = cmbFonts.Text;

//Adding 1 to font size

fontSize = fontSize + 1;

//displaying font size in text box

txtTextSize.Text = fontSize.ToString();

if (userChosenFont == "Arial")

{

txtCarbonFootprintInformation.Font = new Font("Arial", fontSize); //set font to Arial

}

else if (userChosenFont == "Tahoma")

{

txtCarbonFootprintInformation.Font = new Font("Tahoma", fontSize); //set font to Tahoma

}

else if (userChosenFont == "Calibri")

{

txtCarbonFootprintInformation.Font = new Font("Calibri", fontSize); //set font to Calibri

}

else if (userChosenFont == "Verdana")

{

txtCarbonFootprintInformation.Font = new Font("Verdana", fontSize); //set font to Verdana

}

else if (userChosenFont == "Times New Roman")

{

txtCarbonFootprintInformation.Font = new Font("Times New Roman", fontSize); //set font to Times New Roman

}

else if (userChosenFont == "Century Gothic")

{

txtCarbonFootprintInformation.Font = new Font("Century Gothic", fontSize); //set font to Century Gothic

}

else if (userChosenFont == "Microsoft Sans Serif")

{

txtCarbonFootprintInformation.Font = new Font("Microsoft Sans Serif", fontSize); //set font to Microsoft Sans Serif

}

else

{

MessageBox.Show("Error");

}

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = Color.FromArgb(158, 202, 179);

txtCarbonFootprintInformation.BackColor = SystemColors.Window;

txtCarbonFootprintInformation.ForeColor = SystemColors.WindowText;

pbNavbar.BackColor = Color.FromArgb(103, 173, 136);

btnGreenEnergyProducts.BackColor = Color.FromArgb(103, 173, 136);

pbCarbonFootprint.BackColor = Color.FromArgb(103, 173, 136);

btnBooking.BackColor = Color.FromArgb(103, 173, 136);

btnEnergyUsage.BackColor = Color.FromArgb(103, 173, 136);

btnSettings.BackColor = Color.FromArgb(103, 173, 136);

btnToHome.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.ForeColor = Color.FromArgb(179, 214, 196);

pbLogo.BackColor = Color.FromArgb(103, 173, 136);

lblHowToReduceCarbonFootprint.ForeColor = Color.White;

cmbFonts.BackColor = SystemColors.Window;

cmbFonts.ForeColor = SystemColors.WindowText;

btnRaiseFont.BackColor = SystemColors.Window;

btnRaiseFont.ForeColor = SystemColors.WindowText;

btnLowerFont.BackColor = SystemColors.Window;

btnLowerFont.ForeColor = SystemColors.WindowText;

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

txtTextSize.BackColor = SystemColors.Window;

txtTextSize.ForeColor = SystemColors.WindowText;

lblChangeFont.BackColor = Color.FromArgb(158, 202, 179);

lblChangeFont.ForeColor = Color.White;

lblTextSize.BackColor = Color.FromArgb(158, 202, 179);

lblTextSize.ForeColor = Color.White;

//Group Box

grbCarbonFootprintCalculator.BackColor = SystemColors.Window;

grbCarbonFootprintCalculator.ForeColor = Color.FromArgb(103, 173, 136);

lblElectricBill.ForeColor = Color.FromArgb(103, 173, 136);

txtElectricBill.BackColor = SystemColors.Window;

txtElectricBill.ForeColor = SystemColors.WindowText;

lblGasBill.ForeColor = Color.FromArgb(103, 173, 136);

txtGasBill.BackColor = SystemColors.Window;

txtGasBill.ForeColor = SystemColors.WindowText;

lblOilBill.ForeColor = Color.FromArgb(103, 173, 136);

txtOilBill.BackColor = SystemColors.Window;

txtOilBill.ForeColor = SystemColors.WindowText;

lblNumOfFlightsUnderFour.ForeColor = Color.FromArgb(103, 173, 136);

txtNumOfFlightsUnderFour.BackColor = SystemColors.Window;

txtNumOfFlightsUnderFour.ForeColor = SystemColors.WindowText;

lblNumOfFlightsOverFour.ForeColor = Color.FromArgb(103, 173, 136);

txtNumOfFlightsOverFour.BackColor = SystemColors.Window;

txtNumOfFlightsOverFour.ForeColor = SystemColors.WindowText;

lblRecycleNewspaper.ForeColor = Color.FromArgb(103, 173, 136);

cmbRecycleNewspaper.BackColor = SystemColors.Window;

cmbRecycleNewspaper.ForeColor = SystemColors.WindowText;

lblRecycleAluminumTin.ForeColor = Color.FromArgb(103, 173, 136);

cmbRecycleAluminumTin.BackColor = SystemColors.Window;

cmbRecycleAluminumTin.ForeColor = SystemColors.WindowText;

btnGetResult.BackColor = SystemColors.Window;

btnGetResult.ForeColor = Color.FromArgb(103, 173, 136);

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

txtCarbonFootprintInformation.BackColor = Color.Black;

txtCarbonFootprintInformation.ForeColor = Color.Yellow;

pbNavbar.BackColor = Color.Blue;

btnGreenEnergyProducts.BackColor = Color.Blue;

pbCarbonFootprint.BackColor = Color.Blue;

btnBooking.BackColor = Color.Blue;

btnEnergyUsage.BackColor = Color.Blue;

btnSettings.BackColor = Color.Blue;

btnToHome.BackColor = Color.Blue;

lblSlogan.BackColor = Color.Blue;

lblSlogan.ForeColor = Color.White;

pbLogo.BackColor = Color.Blue;

lblHowToReduceCarbonFootprint.ForeColor = Color.Black;

cmbFonts.BackColor = Color.Black;

cmbFonts.ForeColor = Color.Yellow;

btnRaiseFont.BackColor = Color.Black;

btnRaiseFont.ForeColor = Color.Yellow;

btnLowerFont.BackColor = Color.Black;

btnLowerFont.ForeColor = Color.Yellow;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

txtTextSize.BackColor = Color.Black;

txtTextSize.ForeColor = Color.Yellow;

lblChangeFont.BackColor = Color.Black;

lblChangeFont.ForeColor = Color.Yellow;

lblTextSize.BackColor = Color.Black;

lblTextSize.ForeColor = Color.Yellow;

//Group Box

grbCarbonFootprintCalculator.BackColor = Color.Blue;

grbCarbonFootprintCalculator.ForeColor = Color.White;

lblElectricBill.ForeColor = Color.White;

txtElectricBill.BackColor = Color.White;

txtElectricBill.ForeColor = Color.Black;

lblGasBill.ForeColor = Color.White;

txtGasBill.BackColor = Color.White;

txtGasBill.ForeColor = Color.Black;

lblOilBill.ForeColor = Color.White;

txtOilBill.BackColor = Color.White;

txtOilBill.ForeColor = Color.Black;

lblNumOfFlightsUnderFour.ForeColor = Color.White;

txtNumOfFlightsUnderFour.BackColor = Color.White;

txtNumOfFlightsUnderFour.ForeColor = Color.Black;

lblNumOfFlightsOverFour.ForeColor = Color.White;

txtNumOfFlightsOverFour.BackColor = Color.White;

txtNumOfFlightsOverFour.ForeColor = Color.Black;

lblRecycleNewspaper.ForeColor = Color.White;

cmbRecycleNewspaper.BackColor = Color.White;

cmbRecycleNewspaper.ForeColor = Color.Black;

lblRecycleAluminumTin.ForeColor = Color.White;

cmbRecycleAluminumTin.BackColor = Color.White;

cmbRecycleAluminumTin.ForeColor = Color.Black;

btnGetResult.BackColor = Color.White;

btnGetResult.ForeColor = Color.Black;

highContrast = true;

}

}

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

{

//Declaring settings page

var frmToSettings = new frmSettingsRedo();

//Settings page will open in same location

frmToSettings.Location = this.Location;

frmToSettings.StartPosition = FormStartPosition.Manual;

//Carbon footprint page will show again when settings page is closed

frmToSettings.FormClosing += delegate { this.Show(); };

//settings page is shown and carbon footprint page is hidden - not completely closed

frmToSettings.Show();

this.Hide();

}

}

}

# Booking Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmBookingPage : Form

{

public frmBookingPage()

{

InitializeComponent();

}

//declaring high contrast

bool highContrast = false;

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

{

//Declaring home page

frmHomePage frmToHome = new frmHomePage();

//Opening home page, closing booking page

frmToHome.Show();

this.Close();

}

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

{

//Declaring green energy page

frmGreenEnergyProducts frmToGreenEnergyProducts = new frmGreenEnergyProducts();

//Opening green energy page, closing booking page

frmToGreenEnergyProducts.Show();

this.Close();

}

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

{

//Declaring carbon footprint page

frmCarbonFootprintCalculator frmToCarbonFootprint = new frmCarbonFootprintCalculator();

//Opening carbon footprint page, closing booking page

frmToCarbonFootprint.Show();

this.Close();

}

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

{

//Declaring TOS page

var frmTerms = new frmTOS();

//TOS page will open in same location

frmTerms.Location = this.Location;

frmTerms.StartPosition = FormStartPosition.Manual;

//Booking page will show again when TOS page is closed

frmTerms.FormClosing += delegate { this.Show(); };

//TOS page is shown and booking page is hidden - not completely closed

frmTerms.Show();

this.Hide();

}

//Executed as soon as value in combo box is changed

private void cmbService\_SelectedIndexChanged(object sender, EventArgs e)

{

//Index 0 in combo box is consultation

if (cmbService.SelectedIndex == 0)

{

//will display chosen service as consultation in the booking info section

txtService.Text = "Consultation";

}

//Index 1 in combo box is installation

else

{

//will display chosen service as installation in the booking info section

txtService.Text = "Installation";

}

}

private void dtpDateToBook\_ValueChanged\_1(object sender, EventArgs e)

{

//Only date is stored from date time picker

string date = dtpDateToBook.Value.ToShortDateString();

//Displayig chosen date in text box in booking info section

txtDate.Text = date.ToString();

}

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

{

//Declaring database

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//Declaring all variables that will be stored in database

string fullName = txtFullName.Text;

string service = txtService.Text;

DateTime date = DateTime.Parse(txtDate.Text);

string address = txtAddress.Text;

string email = txtEmail.Text;

string phoneNumber = txtPhoneNumber.Text;

//Creating new command to search through stored dates

SqlCommand checkDateCommand = new SqlCommand("SELECT COUNT(\*) FROM BookedAppointments WHERE date = @Date", sqlConnection);

checkDateCommand.Parameters.AddWithValue("@Date", date);

//Opening database connection

sqlConnection.Open();

int dateCount = (int)checkDateCommand.ExecuteScalar();

//Closing database connection

sqlConnection.Close();

if (dateCount > 0) //If date >0, means date already exists in database

{

txtError.Text = "Appointment slot has already been taken, please try again!";

txtError.Visible = true;

}

else

{

//Calling the 'CreateNewBookingRecord' procedure

SqlCommand command = new SqlCommand("CreateNewBookingRecord", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

//validating email

int countEmail = 0;

//Checking if email ends in ".com"

string hasDotCom = email.Substring(email.Length - 4);

//Checking to see if the email has an @ symbol in it

foreach (char c in email)

{

if (c == '@')

{

countEmail = countEmail + 1;

}

}

//Checking to see if phone number is made up of integers

int countPhoneNumDigits = 0;

foreach (char character in phoneNumber)

{

//Getting amount of digits in string

if (character <= 0 || character >= 9)

{

countPhoneNumDigits = countPhoneNumDigits + 1;

}

}

bool allDigits = false;

if (countPhoneNumDigits == phoneNumber.Length)

{

//if amount of digits counted in foreach loop is equal to number of characters in the string

allDigits = true;

}

else

{

//if amount of digits counted in foreach loop is not equal to number of characters in the string

allDigits = false;

}

if (countEmail == 1 && hasDotCom == ".com")

{

if (phoneNumber.Length == 11 && allDigits == true)

{

if (cbConfirmInfo.Checked == true && cbTermsOfService.Checked == true)

{

//Adding all values to database

command.Parameters.AddWithValue("@FullName", fullName);

command.Parameters.AddWithValue("@Service", service);

command.Parameters.AddWithValue("@Date", date);

command.Parameters.AddWithValue("@Address", address);

command.Parameters.AddWithValue("@Email", email);

command.Parameters.AddWithValue("@PhoneNumber", phoneNumber);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

MessageBox.Show("Successfully booked");

txtError.Visible = false;

}

else

{

//Shows if TOS checkbox not ticked

txtError.Text = "Please read our terms of service and confirm your details!";

txtError.Visible = true;

}

}

else

{

//Shows if phone number is not valid

txtError.Text = "Phone number is not valid, please try again!";

txtError.Visible = true;

}

}

else

{

//Shows if email address is not valid

txtError.Text = "Email is not valid, please try again!";

txtError.Visible = true;

}

}

}

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

{

//Declaring energy usage page

frmEnergyUsageCalculator frmToEnergyUsageCalculator = new frmEnergyUsageCalculator();

//Opening energy usage page, closing booking page

frmToEnergyUsageCalculator.Show();

this.Close();

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = Color.FromArgb(158, 202, 179);

pbNavbar.BackColor = Color.FromArgb(103, 173, 136);

btnGreenEnergyProducts.BackColor = Color.FromArgb(103, 173, 136);

btnCarbonFootprint.BackColor = Color.FromArgb(103, 173, 136);

pbCalender.BackColor = Color.FromArgb(103, 173, 136);

btnEnergyUsage.BackColor = Color.FromArgb(103, 173, 136);

btnSettings.BackColor = Color.FromArgb(103, 173, 136);

btnToHome.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.ForeColor = Color.FromArgb(179, 214, 196);

pbLogo.BackColor = Color.FromArgb(103, 173, 136);

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

lblBook.ForeColor = Color.FromArgb(65, 119, 91);

lblInfo.ForeColor = Color.White;

pbLine.BackColor = Color.FromArgb(65, 119, 91);

lblService.ForeColor = Color.White;

grbBookingInfo.BackColor = SystemColors.Window;

lblBookingInfo.ForeColor = Color.FromArgb(65, 119, 91);

lblFullName.ForeColor = Color.FromArgb(65, 119, 91);

lblChosenService.ForeColor = Color.FromArgb(65, 119, 91);

lblDate.ForeColor = Color.FromArgb(65, 119, 91);

lblAddress.ForeColor = Color.FromArgb(65, 119, 91);

lblEmail.ForeColor = Color.FromArgb(65, 119, 91);

lblPhoneNum.ForeColor = Color.FromArgb(65, 119, 91);

cbConfirmInfo.ForeColor = Color.Black;

cbTermsOfService.ForeColor = Color.Black;

btnCheckTOS.BackColor = Color.FromArgb(103, 173, 136);

btnCheckTOS.ForeColor = Color.White;

btnConfirmBooking.BackColor = Color.FromArgb(103, 173, 136);

btnConfirmBooking.ForeColor = Color.White;

txtError.BackColor = SystemColors.Window;

txtError.ForeColor = SystemColors.WindowText;

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

pbNavbar.BackColor = Color.Blue;

btnGreenEnergyProducts.BackColor = Color.Blue;

btnCarbonFootprint.BackColor = Color.Blue;

pbCalender.BackColor = Color.Blue;

btnEnergyUsage.BackColor = Color.Blue;

btnSettings.BackColor = Color.Blue;

btnToHome.BackColor = Color.Blue;

lblSlogan.BackColor = Color.Blue;

lblSlogan.ForeColor = Color.White;

pbLogo.BackColor = Color.Blue;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

lblBook.ForeColor = Color.Black;

lblInfo.ForeColor = Color.Black;

pbLine.BackColor = Color.Blue;

lblService.ForeColor = Color.Black;

grbBookingInfo.BackColor = Color.Blue;

lblBookingInfo.ForeColor = Color.White;

lblFullName.ForeColor = Color.White;

lblChosenService.ForeColor = Color.White;

lblDate.ForeColor = Color.White;

lblAddress.ForeColor = Color.White;

lblEmail.ForeColor = Color.White;

lblPhoneNum.ForeColor = Color.White;

cbConfirmInfo.ForeColor = Color.White;

cbTermsOfService.ForeColor = Color.White;

btnCheckTOS.BackColor = Color.White;

btnCheckTOS.ForeColor = Color.Black;

btnConfirmBooking.BackColor = Color.White;

btnConfirmBooking.ForeColor = Color.Black;

txtError.BackColor = Color.Black;

txtError.ForeColor = Color.White;

highContrast = true;

}

}

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

{

//Declaring settings page

var frmToSettings = new frmSettingsRedo();

//Settings page will open in same location

frmToSettings.Location = this.Location;

frmToSettings.StartPosition = FormStartPosition.Manual;

//Booking page will show again when settings page is closed

frmToSettings.FormClosing += delegate { this.Show(); };

//settings page is shown and booking page is hidden - not completely closed

frmToSettings.Show();

this.Hide();

}

}

}

# Energy Usage Calculator Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Net;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.ListView;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmEnergyUsageCalculator : Form

{

public frmEnergyUsageCalculator()

{

InitializeComponent();

}

//declaring high contrast

bool highContrast = false;

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

{

//Declaring home page

frmHomePage frmToHomePage = new frmHomePage();

//Opening home page, closing energy usage page

frmToHomePage.Show();

this.Close();

}

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

{

//Declaring green energy page

frmGreenEnergyProducts frmToGreenEnergyProducts = new frmGreenEnergyProducts();

//Opening green energy page, closing energy usage page

frmToGreenEnergyProducts.Show();

this.Close();

}

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

{

//Declaring carbon footprint page

frmCarbonFootprintCalculator frmToCarbonFootprintCalculator = new frmCarbonFootprintCalculator();

//Opening carbon footprint page, closing energy usage page

frmToCarbonFootprintCalculator.Show();

this.Close();

}

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

{

//Declaring booking page

frmBookingPage frmToBookingPage = new frmBookingPage();

//Opening booking page, closing energy usage page

frmToBookingPage.Show();

this.Close();

}

//declaring variables used for calculation

double calculation = 0;

double tempCalculation = 0;

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

{

//declaring variables and assigning corresponding inputs

string appliance = txtAppliance.Text;

string power = txtPower.Text;

string hours = txtHoursUsed.Text;

//making list of appliances visible and adding input

txtListedAppliances.Visible = true;

txtListedAppliances.AppendText(appliance + ", ");

//Checking if power is a double

if (double.TryParse(power, out double powerValue))

{

//declaring newPower and making it equal to power, converted to a double

double newPower = double.Parse(power);

//Checking if hours is a double

if (double.TryParse(hours, out double hoursValue))

{

//declaring newHours and making it equal to hours, converted to a double

double newHours = double.Parse(hours);

//multiplying power by hours

tempCalculation = newPower \* newHours;

//adding the calculation into the overall calculation variable

calculation = calculation + tempCalculation;

//Clearing inputs

txtAppliance.Clear();

txtPower.Clear();

txtHoursUsed.Clear();

}

else

{

//Error if hours isn't an integer/double

txtError.Text = "Please ensure that hours is a number";

}

}

else

{

//Error if power isn't an integer/double

txtError.Text = "Please ensure that power is a number";

}

}

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

{

//Executes same code as btnAddAnother to ensure final input has been added to calculation

string appliance = txtAppliance.Text;

string power = txtPower.Text;

string hours = txtHoursUsed.Text;

txtListedAppliances.Visible = true;

txtListedAppliances.AppendText(appliance + ", ");

if (double.TryParse(power, out double powerValue))

{

double newPower = double.Parse(power);

if (double.TryParse(hours, out double hoursValue))

{

double newHours = double.Parse(hours);

tempCalculation = newPower \* newHours;

calculation = calculation + tempCalculation;

double overallCalculation = calculation;

//Outputs final result in result text box

txtResult.Text = "Your overall energy consumption is, " + calculation + "kWh";

//Declaring database

string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30";

SqlConnection sqlConnection = new SqlConnection(connectionString);

//Calling the 'CreateNewUserRecord' procedure

SqlCommand command = new SqlCommand("CreateNewEnergyUsageRecord", sqlConnection);

command.CommandType = CommandType.StoredProcedure;

//Ensures only date is being stored from DateTime

string date = DateTime.Now.ToShortDateString();

//Get's the users username - would be carried over from log in but ran out of time

string username = txtUsername.Text;

//Adds parameters to the EnergyUsage database

command.Parameters.AddWithValue("@Username", username);

command.Parameters.AddWithValue("@Result", overallCalculation);

command.Parameters.AddWithValue("@Date", date);

sqlConnection.Open();

command.ExecuteNonQuery();

sqlConnection.Close();

}

else

{

//Error if hours isn't an integer/double

txtError.Text = "Please ensure that hours is a number";

}

}

else

{

//Error if power isn't an integer/double

txtError.Text = "Please ensure that power is a number";

}

}

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

{

using (var connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30"))

using (var command = connection.CreateCommand())

{

//Getting usernames from EnergyUsage database

command.CommandText = "SELECT Username FROM EnergyUsage";

connection.Open();

using (var reader = command.ExecuteReader())

{

while (reader.Read())

txtUsernames.Text += reader["Username"].ToString() + Environment.NewLine; //Outputting all usernames on their own line in the usernames text box

}

connection.Close();

}

using (var connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30"))

using (var command = connection.CreateCommand())

{

//Getting results from EnergyUsage database

command.CommandText = "SELECT Result FROM EnergyUsage";

connection.Open();

using (var reader = command.ExecuteReader())

{

while (reader.Read())

txtResults.Text += reader["Result"].ToString() + Environment.NewLine; //Outputting all results on their own line in the results text box

}

connection.Close();

}

using (var connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=\"C:\\Users\\EXAM1011\\OneDrive - Middlesbrough College\\Exams\\Task 2\\RolsaRechnologiesSolutionV1\\RolsaTechDBRedo.mdf\";Integrated Security=True;Connect Timeout=30"))

using (var command = connection.CreateCommand())

{

//Getting dates from EnergyUsage database

command.CommandText = "SELECT Date FROM EnergyUsage";

connection.Open();

using (var reader = command.ExecuteReader())

{

while (reader.Read())

txtDates.Text += reader["Date"].ToString() + Environment.NewLine; //Outputting all dates on their own line in the results text box

}

connection.Close();

}

}

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

{

//if statement allows user to toggle between normal and high contrast

if (highContrast == true)

{

//Setting all assets back to their original colours

BackColor = Color.FromArgb(158, 202, 179);

pbNavbar.BackColor = Color.FromArgb(103, 173, 136);

btnGreenEnergyProducts.BackColor = Color.FromArgb(103, 173, 136);

btnCarbonFootprint.BackColor = Color.FromArgb(103, 173, 136);

btnBooking.BackColor = Color.FromArgb(103, 173, 136);

pbEnergyUsageCalculator.BackColor = Color.FromArgb(103, 173, 136);

btnSettings.BackColor = Color.FromArgb(103, 173, 136);

btnToHome.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.BackColor = Color.FromArgb(103, 173, 136);

lblSlogan.ForeColor = Color.FromArgb(179, 214, 196);

pbLogo.BackColor = Color.FromArgb(103, 173, 136);

btnHighContrast.BackColor = SystemColors.Window;

btnHighContrast.ForeColor = SystemColors.WindowText;

txtError.BackColor = SystemColors.Window;

txtError.ForeColor = SystemColors.WindowText;

lblEnergyUsageCalculator.ForeColor = Color.White;

txtError.BackColor = SystemColors.Window;

txtError.ForeColor = SystemColors.WindowText;

grbCalculation.BackColor = SystemColors.Window;

lblUsername.ForeColor = Color.Black;

lblAppliance.ForeColor = Color.Black;

lblPower.ForeColor = Color.Black;

lblHoursUsed.ForeColor = Color.Black;

lblOr.ForeColor = Color.Black;

txtListedAppliances.BackColor = SystemColors.Window;

txtListedAppliances.ForeColor = SystemColors.WindowText;

btnUpdate.BackColor = SystemColors.Window;

btnUpdate.ForeColor = SystemColors.WindowText;

txtUsernames.BackColor = SystemColors.Window;

txtUsernames.ForeColor = SystemColors.WindowText;

txtResults.BackColor = SystemColors.Window;

txtResults.ForeColor = SystemColors.WindowText;

txtDates.BackColor = SystemColors.Window;

txtDates.ForeColor = SystemColors.WindowText;

highContrast = false;

}

else

{

//Changing the colours of all assets to brighter colours (blue, yellow, black, white)

BackColor = Color.Yellow;

pbNavbar.BackColor = Color.Blue;

btnGreenEnergyProducts.BackColor = Color.Blue;

btnCarbonFootprint.BackColor = Color.Blue;

btnBooking.BackColor = Color.Blue;

pbEnergyUsageCalculator.BackColor = Color.Blue;

btnSettings.BackColor = Color.Blue;

btnToHome.BackColor = Color.Blue;

lblSlogan.BackColor = Color.Blue;

lblSlogan.ForeColor = Color.White;

pbLogo.BackColor = Color.Blue;

btnHighContrast.BackColor = Color.Black;

btnHighContrast.ForeColor = Color.Yellow;

lblEnergyUsageCalculator.ForeColor = Color.Black;

txtError.BackColor = Color.Black;

txtError.ForeColor = Color.White;

grbCalculation.BackColor = Color.Blue;

lblUsername.ForeColor = Color.White;

lblAppliance.ForeColor = Color.White;

lblPower.ForeColor = Color.White;

lblHoursUsed.ForeColor = Color.White;

lblOr.ForeColor = Color.White;

txtListedAppliances.BackColor = Color.Black;

txtListedAppliances.ForeColor = Color.Yellow;

btnUpdate.BackColor = Color.Black;

btnUpdate.ForeColor = Color.Yellow;

txtUsernames.BackColor = Color.Black;

txtUsernames.ForeColor = Color.Yellow;

txtResults.BackColor = Color.Black;

txtResults.ForeColor = Color.Yellow;

txtDates.BackColor = Color.Black;

txtDates.ForeColor = Color.Yellow;

highContrast = true;

}

}

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

{

//Declaring settings page

var frmToSettings = new frmSettingsRedo();

//Settings page will open in same location

frmToSettings.Location = this.Location;

frmToSettings.StartPosition = FormStartPosition.Manual;

//Energy usage page will show again when settings page is closed

frmToSettings.FormClosing += delegate { this.Show(); };

//Energy usage page is shown and booking page is hidden - not completely closed

frmToSettings.Show();

this.Hide();

}

}

}

# Settings Page

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace RolsaRechnologiesSolutionV1

{

public partial class frmSettings : Form

{

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

{

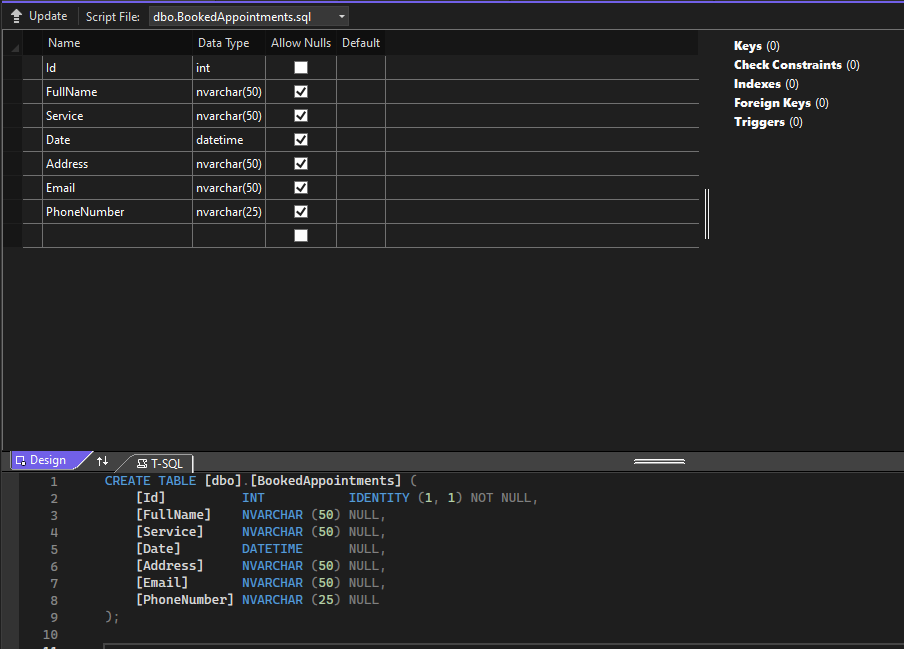
Application.Exit();

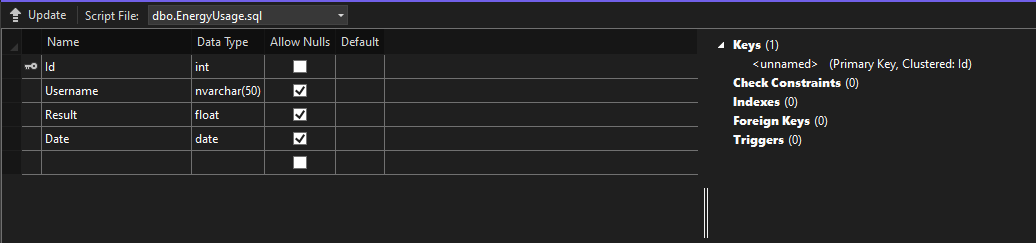
}

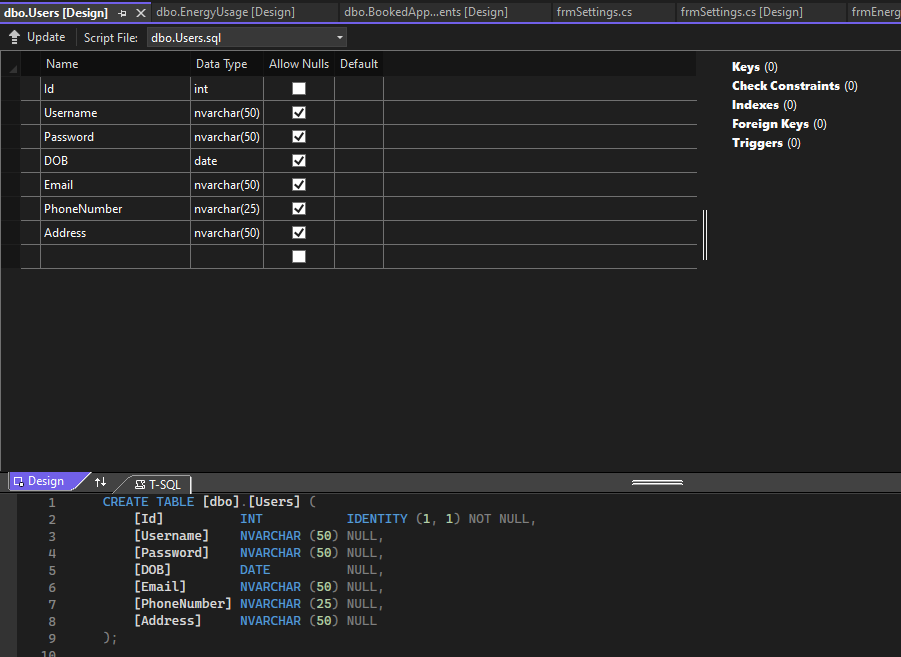
}

}

# SQL







## Stored Procedures

### CreateNewBookingRecord

CREATE PROCEDURE [dbo].[CreateNewBookingRecord]

(

@FullName nvarchar(50),

@Service nvarchar(50),

@Date date,

@Address nvarchar(50),

@Email nvarchar(25),

@PhoneNumber nvarchar(50)

)

as

begin

Insert into BookedAppointments values(@FullName, @Service, @Date, @Address, @Email, @PhoneNumber)

End

### CreateNewEnergyUsageRecord

CREATE PROCEDURE [dbo].[CreateNewEnergyUsageRecord]

(

@Username nvarchar(50),

@Result float,

@Date date

)

as

begin

Insert into EnergyUsage values(@Username, @Result, @Date)

End

### CreateNewUserRecord

CREATE PROCEDURE [dbo].[CreateNewUserRecord]

(

@Username nvarchar(50),

@Password nvarchar(50),

@DOB date,

@Email nvarchar(50),

@PhoneNumber nvarchar(25),

@Address nvarchar(50)

)

as

begin

Insert into Users values(@Username, @Password, @DOB, @Email, @PhoneNumber, @Address)

End