LoginForm.cs

namespace \_101062225\_eventManagement\_project

{

public partial class LoginForm : Form

{

public LoginForm()

{

InitializeComponent();

}

private void btnLogin\_Click(object sender, EventArgs e){

String Username = "admin";

String Password = "admin";

if(txtUser.Text == Username && txtPassword.Text == Password)

{

MessageBox.Show("Login Successful", "Login", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

this.Hide();

Main\_Form f2 = new Main\_Form();

f2.ShowDialog(); //

}

else

{

MessageBox.Show("Please Check your Username and Password", "Login", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void TxtUser\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtUser.Text))

{

e.Cancel = true;

txtUser.Focus();

errorProviderApp.SetError(txtUser, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtUser, "");

}

}

private void TxtPassword\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtPassword.Text))

{

e.Cancel = true;

txtPassword.Focus();

errorProviderApp.SetError(txtPassword, "Name should not be left blank!");

}

else

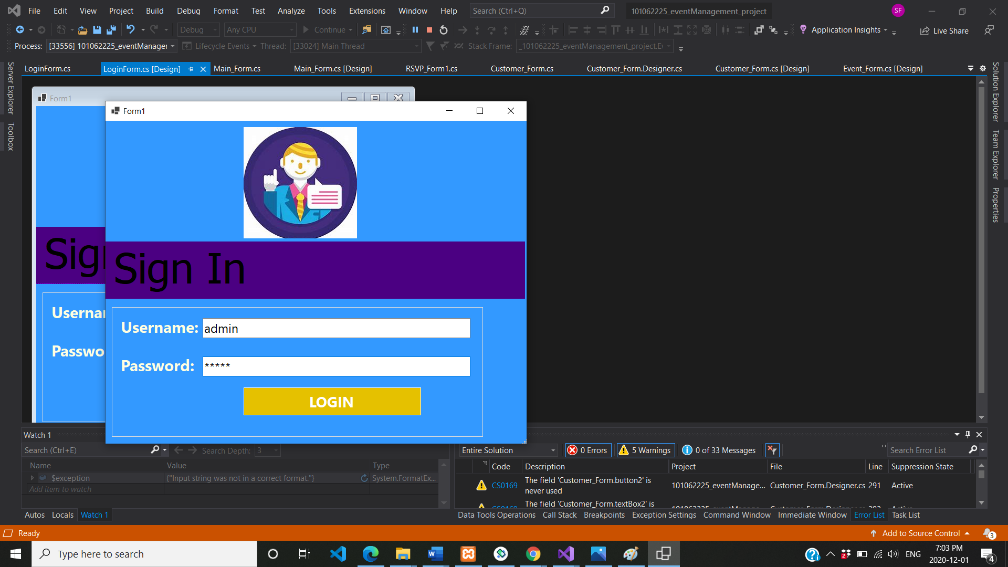
{

e.Cancel = false;

errorProviderApp.SetError(txtPassword, "");

}

} }}



A picture containing text, screenshot, monitor, computer

Description automatically generated

Main\_Form.cs

namespace \_101062225\_eventManagement\_project

{

public partial class Main\_Form : Form

{

public Main\_Form()

{

InitializeComponent();

}

private void Main\_Form\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

Customer\_Form f = new Customer\_Form();

f.ShowDialog();

}

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

{

this.Hide();

LoginForm f1 = new LoginForm();

f1.ShowDialog();

}

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

{

this.Hide();

Event\_Form f2 = new Event\_Form();

f2.ShowDialog();

}

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

{

this.Hide();

RSVP\_Form1 f3 = new RSVP\_Form1();

f3.ShowDialog();

}}}

A screenshot of a computer

Description automatically generated

Customer\_Form.cs

using System.Windows.Forms;

using System.Data.SqlClient;

namespace \_101062225\_eventManagement\_project

{

public partial class Customer\_Form : Form

{

SqlConnection sqlCon = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\reyaf\source\repos\101062225\_eventManagement\_project\DB\ContactDB.mdf;Integrated Security=True;Connect Timeout=30");

int CustomerId = 0;

public Customer\_Form()

{

InitializeComponent();

}

private void btnSave\_Click\_1(object sender, EventArgs e)

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

if (btnSave.Text == "Save")

{

SqlCommand sqlCmd = new SqlCommand("ContactAddorEdit", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Add");

sqlCmd.Parameters.AddWithValue("@CustomerId", 0);

sqlCmd.Parameters.AddWithValue("@FirstName", txtFname.Text.Trim());

sqlCmd.Parameters.AddWithValue("@LastName", txtLname.Text.Trim());

sqlCmd.Parameters.AddWithValue("@PhoneNumber", txtPhone.Text.Trim());

sqlCmd.Parameters.AddWithValue("@BookingId", txtBooking.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Saved Successfully");

}

else

{

SqlCommand sqlCmd = new SqlCommand("ContactAddorEdit", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Edit");

sqlCmd.Parameters.AddWithValue("@CustomerId", CustomerId);

sqlCmd.Parameters.AddWithValue("@FirstName", txtFname.Text.Trim());

sqlCmd.Parameters.AddWithValue("@LastName", txtLname.Text.Trim());

sqlCmd.Parameters.AddWithValue("@PhoneNumber", txtPhone.Text.Trim());

sqlCmd.Parameters.AddWithValue("@BookingId", txtBooking.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Updated Successfully");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error Message");

}

finally

{

sqlCon.Close();

}

}

void FillDataGridView()

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlDataAdapter sqlDa = new SqlDataAdapter("ContactViewOrSearch", sqlCon);

sqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

sqlDa.SelectCommand.Parameters.AddWithValue("@ContactName", txtSearch.Text.Trim());

DataTable dtbl = new DataTable();

sqlDa.Fill(dtbl);

dgvContacts.DataSource = dtbl;

sqlCon.Close();

}

private void btnReset\_Click\_1(object sender, EventArgs e)

{

Reset();

}

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

{

try

{

FillDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

private void dgvContacts\_DoubleClick(object sender, EventArgs e)

{

if (dgvContacts.CurrentRow.Index != -1)

{

CustomerId = Convert.ToInt32(dgvContacts.CurrentRow.Cells[0].Value.ToString());

txtFname.Text = dgvContacts.CurrentRow.Cells[1].Value.ToString();

txtLname.Text = dgvContacts.CurrentRow.Cells[2].Value.ToString();

txtPhone.Text = dgvContacts.CurrentRow.Cells[3].Value.ToString();

txtBooking.Text = dgvContacts.CurrentRow.Cells[4].Value.ToString();

btnSave.Text = "Update";

btnDelete.Enabled = true;

}

}

void Reset()

{

txtFname.Text = txtLname.Text = txtPhone.Text = txtBooking.Text = "";

btnSave.Text = "Save";

CustomerId = 0;

btnDelete.Enabled = false;

}

private void Customer\_Form\_Load(object sender, EventArgs e)

{

Reset();

FillDataGridView();

}

private void btnDelete\_Click\_1(object sender, EventArgs e)

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlCommand sqlCmd = new SqlCommand("ContactDeletion", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@CustomerId", CustomerId);

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Deleted Successfully");

Reset();

FillDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ERROR MESSAGE");

}

}

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

{

this.Hide();

Event\_Form f1 = new Event\_Form();

f1.ShowDialog();

}

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

{

this.Hide();

Main\_Form f2 = new Main\_Form();

f2.ShowDialog();

}

private void txtFname\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtFname.Text))

{

e.Cancel = true;

txtFname.Focus();

errorProviderApp.SetError(txtFname, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtFname, "");

}

}

private void txtLname\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtLname.Text))

{

e.Cancel = true;

txtFname.Focus();

errorProviderApp.SetError(txtLname, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtLname, "");

}

}

private void TxtPhone\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtPhone.Text))

{

e.Cancel = true;

txtPhone.Focus();

errorProviderApp.SetError(txtPhone, " should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtPhone, "");

}

}

private void TxtBooking\_Validating(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtBooking.Text))

{

txtBooking.Focus();

errorProviderApp.SetError(txtBooking, "Booking should not be left blank!");

}

else

{

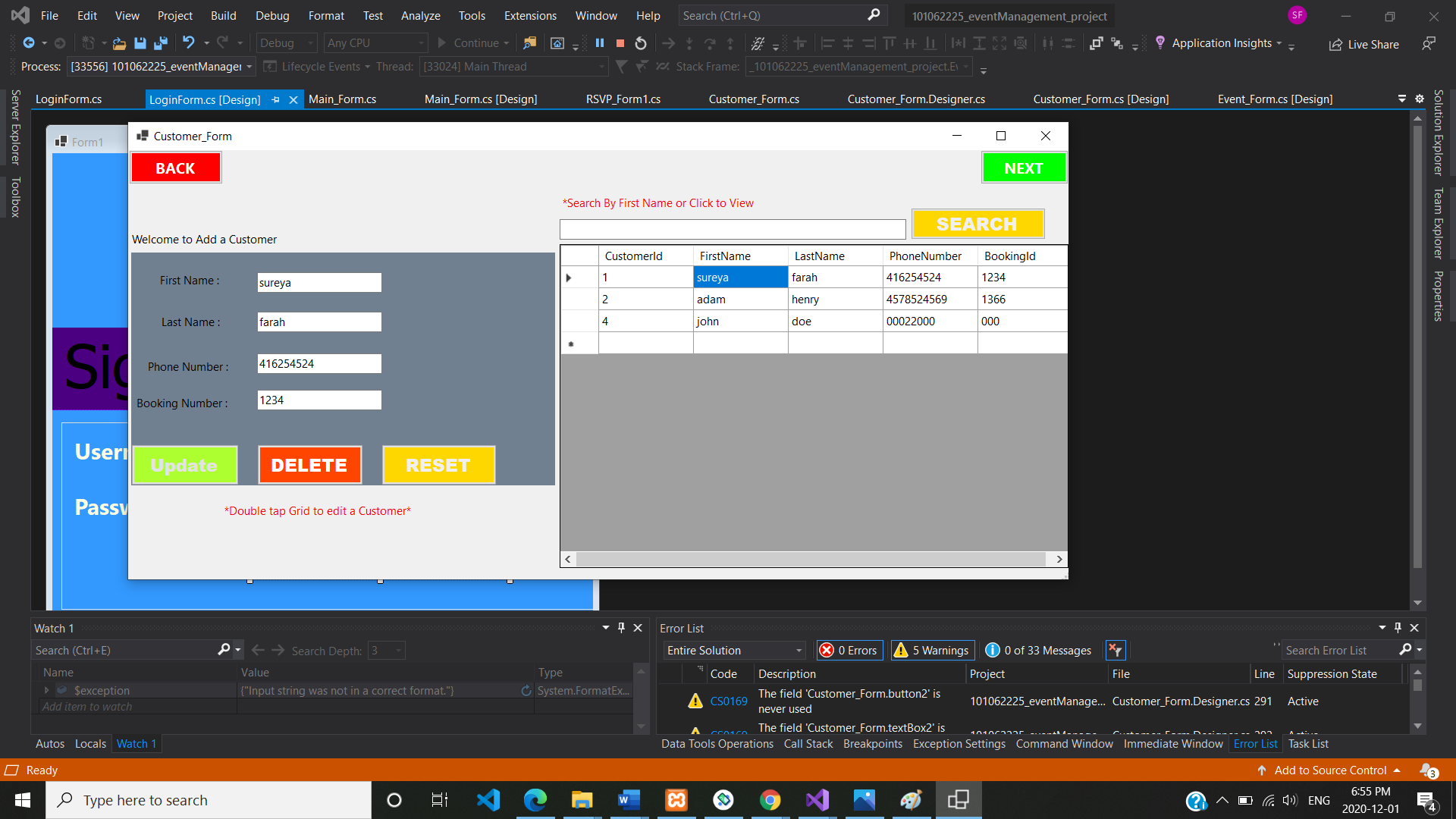
errorProviderApp.SetError(txtBooking, "");

}

}

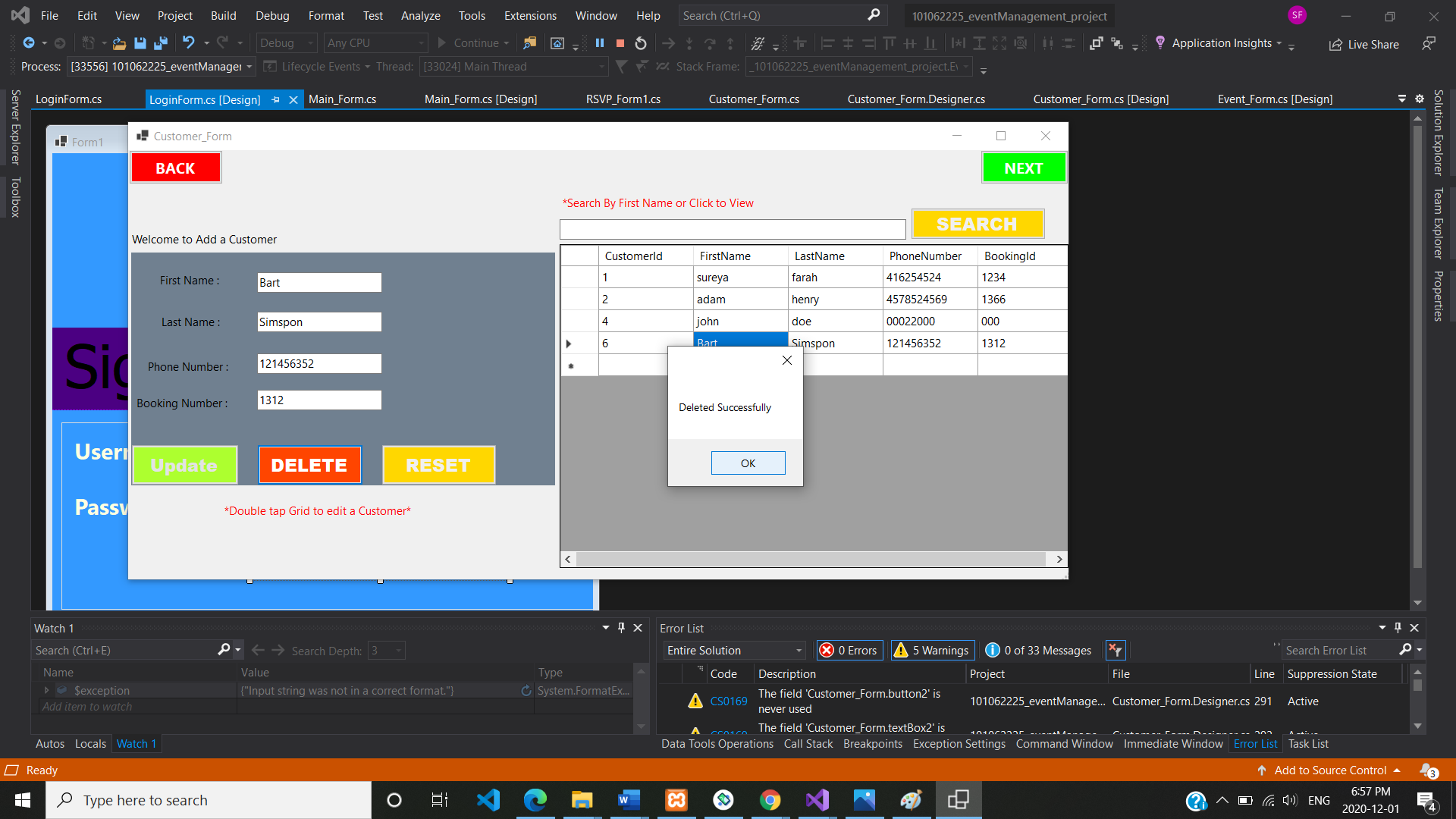
}

}

A picture containing text, screenshot, indoor, computer

Description automatically generatedA picture containing text, screenshot, computer, monitor

Description automatically generatedA picture containing text, screenshot, computer, monitor

Description automatically generated

A screenshot of a computer

Description automatically generated

Event\_Form.cs

namespace \_101062225\_eventManagement\_project

{

public partial class Event\_Form : Form

{

SqlConnection sqlCon = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\reyaf\source\repos\101062225\_eventManagement\_project\DB\ContactDB.mdf;Integrated Security=True;Connect Timeout=30");

int EventId = 0;

public Event\_Form()

{

InitializeComponent();

}

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

{

this.Hide();

Customer\_Form f1 = new Customer\_Form();

f1.ShowDialog();

}

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

{

this.Hide();

RSVP\_Form1 f2 = new RSVP\_Form1();

f2.ShowDialog();

}

private void Event\_Form\_Load(object sender, EventArgs e)

{

EventReset();

FillSearchDataGridView();

// FillEventDataGridView();

}

private void DgvEvents\_DoubleClick(object sender, EventArgs e)

{

if (dgvEvents.CurrentRow.Index != -1)

{

EventId = Convert.ToInt32(dgvEvents.CurrentRow.Cells[0].Value.ToString());

txtEventName.Text = dgvEvents.CurrentRow.Cells[1].Value.ToString();

txtEventDate.Text = dgvEvents.CurrentRow.Cells[2].Value.ToString();

txtPeople.Text = dgvEvents.CurrentRow.Cells[3].Value.ToString();

txtVenue.Text = dgvEvents.CurrentRow.Cells[4].Value.ToString();

btnEventSave.Text = "Update";

btnEventDelete.Enabled = true;

}

}

/\* void FillEventDataGridView()

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlDataAdapter sqlDa = new SqlDataAdapter("EventByNameViewOrSearch", sqlCon);

sqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

sqlDa.SelectCommand.Parameters.AddWithValue("@EventName", txtEventSearch.Text.Trim());

DataTable dtbl = new DataTable();

sqlDa.Fill(dtbl);

dgvEvents.DataSource = dtbl;

sqlCon.Close();

}\*/

void FillSearchDataGridView()

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlDataAdapter sqlDa = new SqlDataAdapter("EventByNameViewOrSearch", sqlCon);

sqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

sqlDa.SelectCommand.Parameters.AddWithValue("@EventName", txtEventSearch.Text.Trim());

DataTable dtbl = new DataTable();

sqlDa.Fill(dtbl);

dgvEvents.DataSource = dtbl;

sqlCon.Close();

}

void EventReset()

{

txtEventName.Text = txtEventDate.Text = txtPeople.Text = txtVenue.Text= "";

btnEventSave.Text = "Save";

EventId = 0;

btnEventDelete.Enabled = false;

}

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

{

try

{

FillSearchDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

private void BtnEventDelete\_Click\_1(object sender, EventArgs e)

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlCommand sqlCmd = new SqlCommand("EventDeletion", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@EventId", EventId);

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Deleted Successfully");

EventReset();

FillSearchDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ERROR MESSAGE");

}

}

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

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

if (btnEventSave.Text == "Save")

{

SqlCommand sqlCmd = new SqlCommand("EventAddorEdit", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Add");

sqlCmd.Parameters.AddWithValue("@EventId", 0);

sqlCmd.Parameters.AddWithValue("@EventName", txtEventName.Text.Trim());

sqlCmd.Parameters.AddWithValue("@EventDate", txtEventDate.Text.Trim());

sqlCmd.Parameters.AddWithValue("@PeopleAttending", txtPeople.Text.Trim());

sqlCmd.Parameters.AddWithValue("@EventVenue", txtVenue.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Saved Successfully");

}

else

{

SqlCommand sqlCmd = new SqlCommand("EventAddorEdit", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Edit");

sqlCmd.Parameters.AddWithValue("@EventId", EventId);

sqlCmd.Parameters.AddWithValue("@EventName", txtEventName.Text.Trim());

sqlCmd.Parameters.AddWithValue("@EventDate", txtEventDate.Text.Trim());

sqlCmd.Parameters.AddWithValue("@PeopleAttending", txtPeople.Text.Trim());

sqlCmd.Parameters.AddWithValue("@EventVenue", txtVenue.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Updated Successfully");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error Message");

}

finally

{

sqlCon.Close();

}

}

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

{

EventReset();

FillSearchDataGridView();

}

private void txtEventName\_Validatin(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtEventName.Text)) {

e.Cancel = true;

txtEventName.Focus();

errorProvider1.SetError(txtEventName, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProvider1.SetError(txtEventName, "");

}

}

private void TxtVenue\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtVenue.Text))

{

e.Cancel = true;

txtEventDate.Focus();

errorProvider1.SetError(txtVenue, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProvider1.SetError(txtVenue, "");

}

}

private void TxtPeople\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

private void TxtDate\_Validating(object sender, CancelEventArgs e)

{

Regex reg = new Regex(@"^(\d{1,2})/(\d{1,2})/(\d{4})$");

Match m = reg.Match(txtEventDate.Text);

if (m.Success)

{

int dd = int.Parse(m.Groups[1].Value);

int mm = int.Parse(m.Groups[2].Value);

int yyyy = int.Parse(m.Groups[3].Value);

e.Cancel = dd < 1 || dd > 31 || mm < 1 || mm > 12 || yyyy > 2020;

}

else e.Cancel = true;

if (e.Cancel)

{

if (MessageBox.Show("Wrong date format. The correct format is dd/mm/yyyy\n+ dd should be between 1 and 31.\n+ mm should be between 1 and 12.\n+ yyyy should be before 2013", "Invalid date", MessageBoxButtons.OKCancel, MessageBoxIcon.Error) == DialogResult.Cancel)

e.Cancel = false;

}

}

}

}

A picture containing text, screenshot, indoor, monitor

Description automatically generatedA picture containing text, screenshot, indoor, monitor

Description automatically generatedA picture containing text, screenshot, indoor, monitor

Description automatically generatedA picture containing text, screenshot, monitor, indoor

Description automatically generatedA picture containing text, screenshot, monitor, indoor

Description automatically generated

A picture containing text, screenshot, indoor, monitor

Description automatically generatedA picture containing text, screenshot, indoor, monitor

Description automatically generated

RSVP\_Form1.cs

namespace \_101062225\_eventManagement\_project

{

public partial class RSVP\_Form1 : Form

{

SqlConnection sqlCon = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\reyaf\source\repos\101062225\_eventManagement\_project\DB\ContactDB.mdf;Integrated Security=True;Connect Timeout=30");

int RSVPId = 0;

public RSVP\_Form1()

{

InitializeComponent();

}

private void Btn\_GetEvent\_Click(object sender, EventArgs e)

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

/\*

\* SqlDataAdapter sqlDa = new SqlDataAdapter("EventViewOrSearch", sqlCon);

DataTable dt = new DataTable();

sqlDa.Fill(dt);

\*/

SqlCommand cmd = new SqlCommand("EventViewOrSearch", sqlCon);

cmd.CommandType = CommandType.StoredProcedure;

//Add your parameters

cmd.Parameters.AddWithValue("@EventName", txtEvent.Text.Trim());

//Declare your data adapter

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

SqlDataAdapter sqlDa1 = new SqlDataAdapter("CreateRSVPevent", sqlCon);

DataTable dt1 = new DataTable();

sqlDa1.Fill(dt1);

for (int i = 0; i < dt.Rows.Count; i++)

{

cbxEvt.Items.Add(dt.Rows[i]["EventName"]);

}

for (int j = 0; j < dt1.Rows.Count; j++)

{

cbxFirst.Items.Add(dt1.Rows[j]["FirstName"]);

cbxBook.Items.Add(dt1.Rows[j]["BookingId"]);

cbxLast.Items.Add(dt1.Rows[j]["LastName"]);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

if (btnSave.Text == "Save")

{

SqlCommand sqlCmd = new SqlCommand("AddRSVPevent", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Add");

sqlCmd.Parameters.AddWithValue("@RSVPId", 0);

sqlCmd.Parameters.AddWithValue("@EventName", txtEvent.Text.Trim());

sqlCmd.Parameters.AddWithValue("@FirstName", txtFirst.Text.Trim());

sqlCmd.Parameters.AddWithValue("@BookingId", txtbook.Text.Trim());

sqlCmd.Parameters.AddWithValue("@LastName", txtlast.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Saved Successfully");

}

else

{

SqlCommand sqlCmd = new SqlCommand("AddRSVPevent", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@mode", "Edit");

sqlCmd.Parameters.AddWithValue("@RSVPId", RSVPId);

sqlCmd.Parameters.AddWithValue("@EventName", txtEvent.Text.Trim());

sqlCmd.Parameters.AddWithValue("@FirstName", txtFirst.Text.Trim());

sqlCmd.Parameters.AddWithValue("@BookingId", txtbook.Text.Trim());

sqlCmd.Parameters.AddWithValue("@LastName", txtlast.Text.Trim());

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Updated Successfully");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error Message");

}

finally

{

sqlCon.Close();

}

}

void FillRSVPDataGridView()

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlCommand cmd = new SqlCommand("RSVPViewOrSearch", sqlCon);

cmd.CommandType = CommandType.StoredProcedure;

//Add your parameters

cmd.Parameters.AddWithValue("@FirstName", txtRSVPsearch.Text.Trim());

SqlDataAdapter sda = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

sda.Fill(dt);

dgvRSVP.DataSource = dt.DefaultView;

sqlCon.Close();

}

void RSVPReset()

{

txtEvent.Text = txtFirst.Text = txtbook.Text = txtlast.Text = "";

btnSave.Text = "Save";

RSVPId = 0;

btnRdelete.Enabled = false;

}

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

{

try

{

if (sqlCon.State == ConnectionState.Closed)

sqlCon.Open();

SqlCommand sqlCmd = new SqlCommand("RSVPDeletion", sqlCon);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@RSVPId", RSVPId);

sqlCmd.ExecuteNonQuery();

MessageBox.Show("Deleted Successfully");

RSVPReset();

FillRSVPDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "ERROR MESSAGE");

}

}

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

{

try

{

FillRSVPDataGridView();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

RSVPReset();

}

private void Btn\_GetEvent\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

txtEvent.Text = cbxEvt.SelectedItem.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

try

{

txtFirst.Text = cbxFirst.SelectedItem.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

try

{

txtbook.Text = cbxBook.SelectedItem.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

try

{

txtlast.Text = cbxLast.SelectedItem.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Invalid Entry");

}

}

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

{

this.Hide();

Event\_Form f1 = new Event\_Form();

f1.ShowDialog();

}

private void RSVP\_Form1\_Load(object sender, EventArgs e)

{

RSVPReset();

FillRSVPDataGridView();

}

private void DgvRSVP\_DoubleClick(object sender, EventArgs e)

{

if (dgvRSVP.CurrentRow.Index != -1)

{

RSVPId = Convert.ToInt32(dgvRSVP.CurrentRow.Cells[0].Value.ToString());

txtEvent.Text = dgvRSVP.CurrentRow.Cells[1].Value.ToString();

txtFirst.Text = dgvRSVP.CurrentRow.Cells[2].Value.ToString();

txtbook.Text = dgvRSVP.CurrentRow.Cells[3].Value.ToString();

txtlast.Text = dgvRSVP.CurrentRow.Cells[4].Value.ToString();

btnSave.Text = "Update";

btnRdelete.Enabled = true;

}

}

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

{

this.Hide();

Event\_Form f1 = new Event\_Form();

f1.ShowDialog();

}

private void btnNext\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Main\_Form f2 = new Main\_Form();

f2.ShowDialog();

}

private void TxteEvent\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtEvent.Text))

{

e.Cancel = true;

txtEvent.Focus();

errorProviderApp.SetError(txtEvent, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtEvent, "");

}

}

private void TxtFirst\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrWhiteSpace(txtFirst.Text))

{

e.Cancel = true;

txtFirst.Focus();

errorProviderApp.SetError(txtFirst, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtFirst, "");

}

}

private void TxtBook\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 8)

{

e.Handled = true;

}

}

private void Txtlast\_Validating(object sender, CancelEventArgs e)

{

if(string.IsNullOrWhiteSpace(txtlast.Text))

{

e.Cancel = true;

txtlast.Focus();

errorProviderApp.SetError(txtlast, "Name should not be left blank!");

}

else

{

e.Cancel = false;

errorProviderApp.SetError(txtlast, "");

}

}

}

}

A screenshot of a computer

Description automatically generated

A picture containing text, screenshot, indoor, monitor

Description automatically generated

A picture containing text, screenshot, indoor, monitor

Description automatically generated

A picture containing text, indoor, screenshot, black

Description automatically generated

A picture containing text, screenshot, monitor, indoor

Description automatically generated