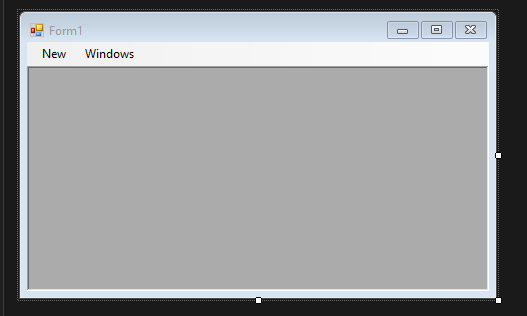
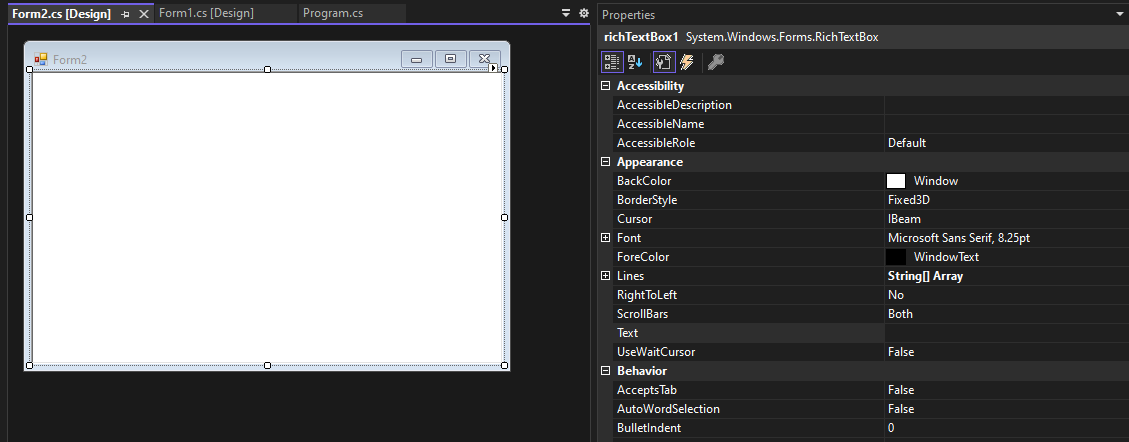
LAB6

Bài 1





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 Lab6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Form2 newMDIChild = new Form2();

// Set the Parent Form of the Child window.

newMDIChild.MdiParent = this;

// Display the new form.

newMDIChild.Show();

}

}

}

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 Lab6

{

public partial class Form2 : Form

{

public Form2()

{

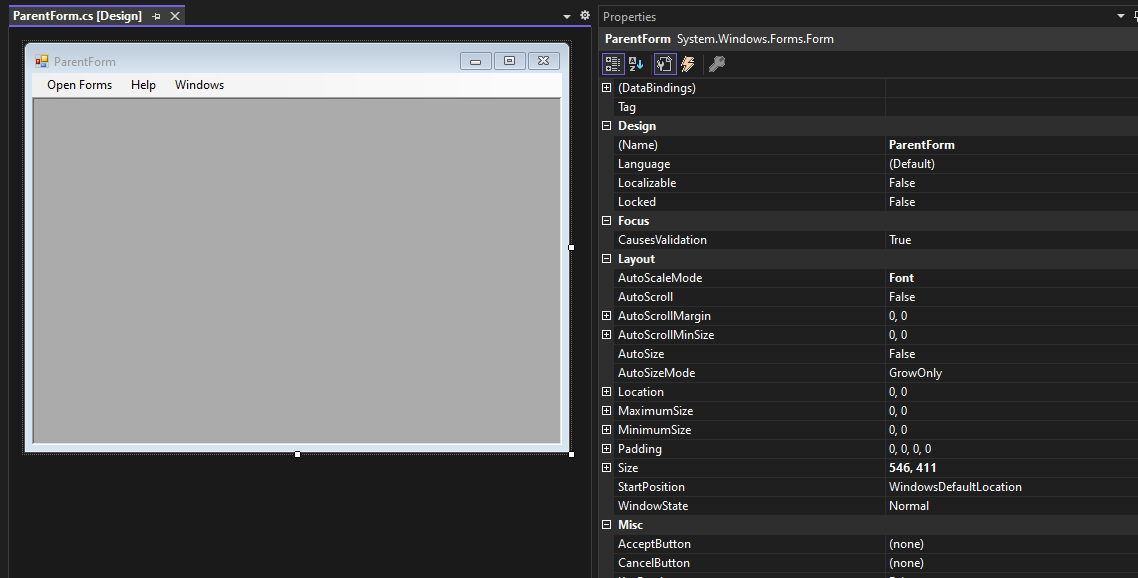
InitializeComponent();

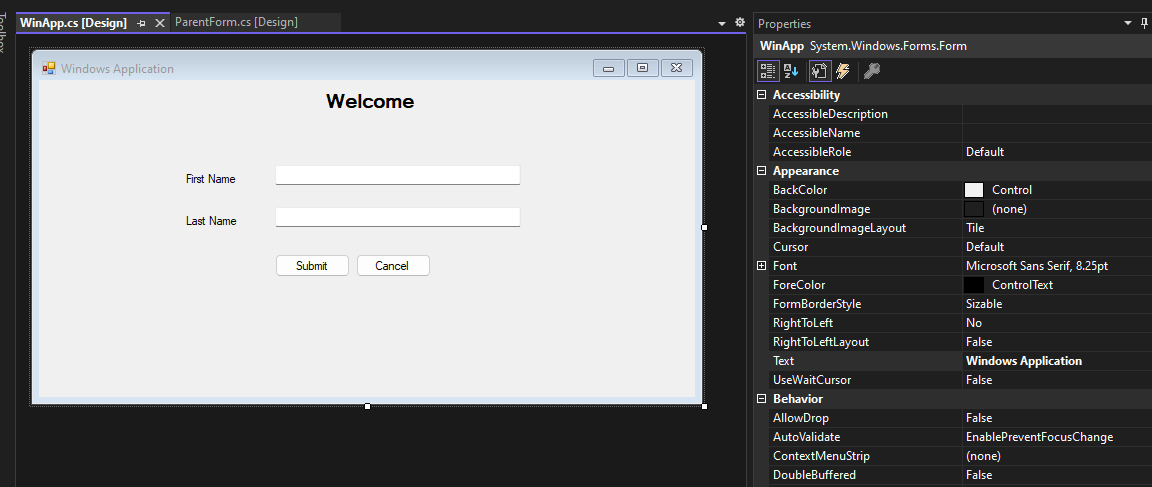
}

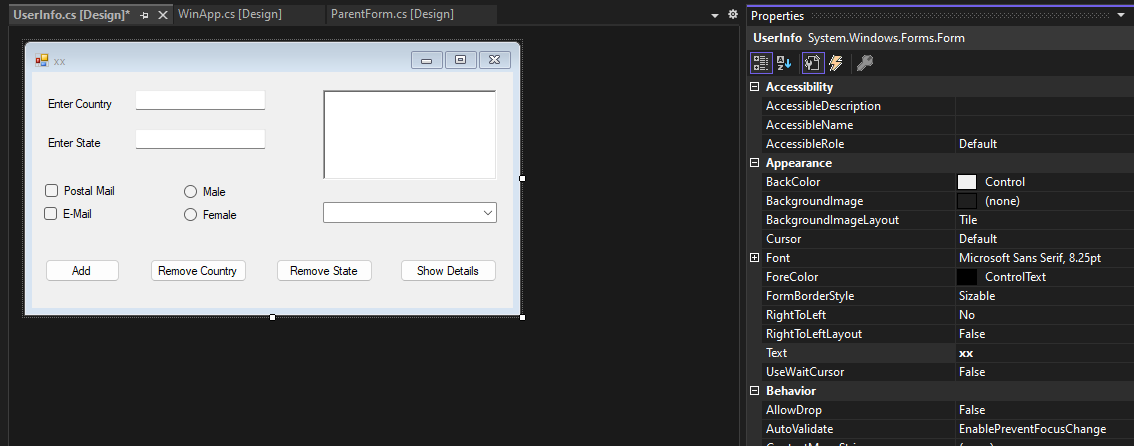
}

}

Bài 2







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 Ex2

{

public partial class ParentForm : Form

{

public ParentForm()

{

InitializeComponent();

}

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

{

WinApp objWA = new WinApp();

objWA.Show();

}

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

{

UserInfo objUI = new UserInfo();

objUI.Show();

}

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

{

Application.Exit();

}

}

}

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 Ex2

{

public partial class WinApp : Form

{

public WinApp()

{

InitializeComponent();

}

}

}

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 Ex2

{

public partial class UserInfo : Form

{

public UserInfo()

{

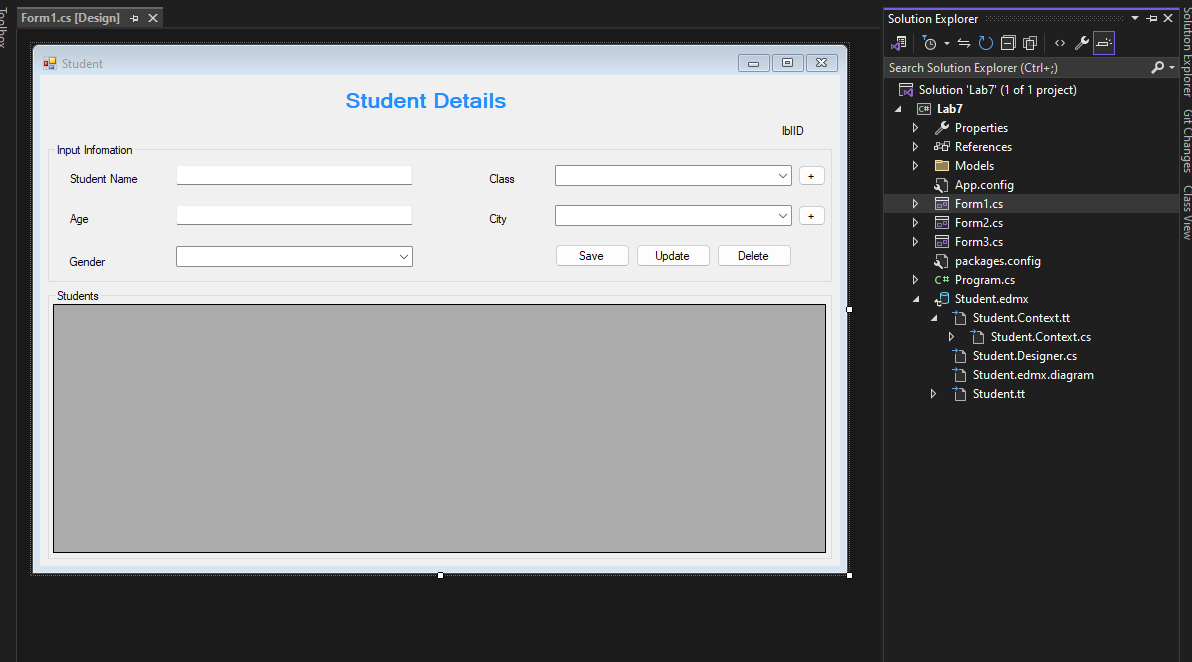
InitializeComponent();

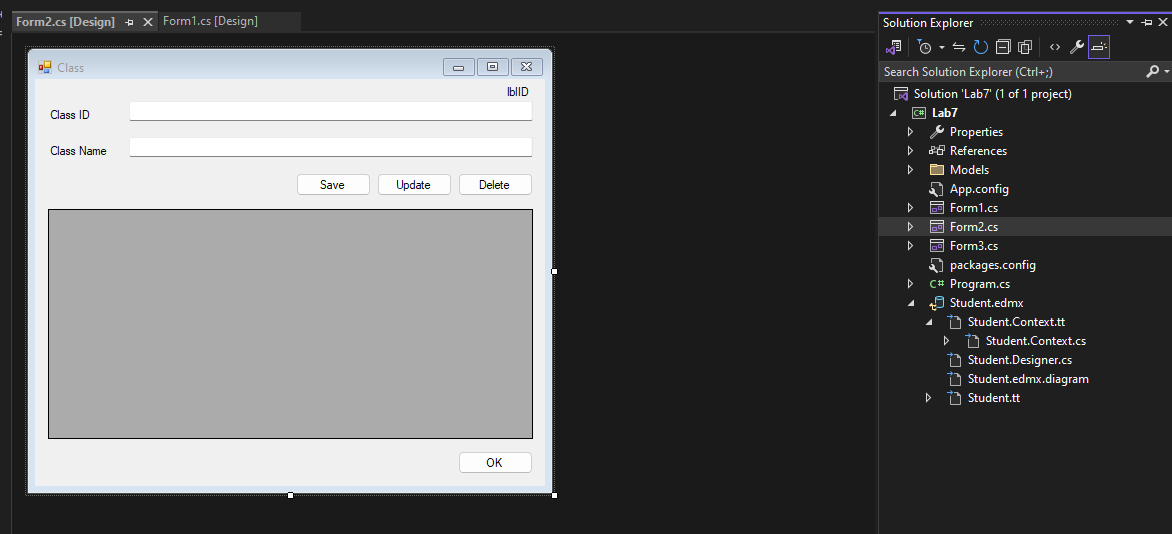
}

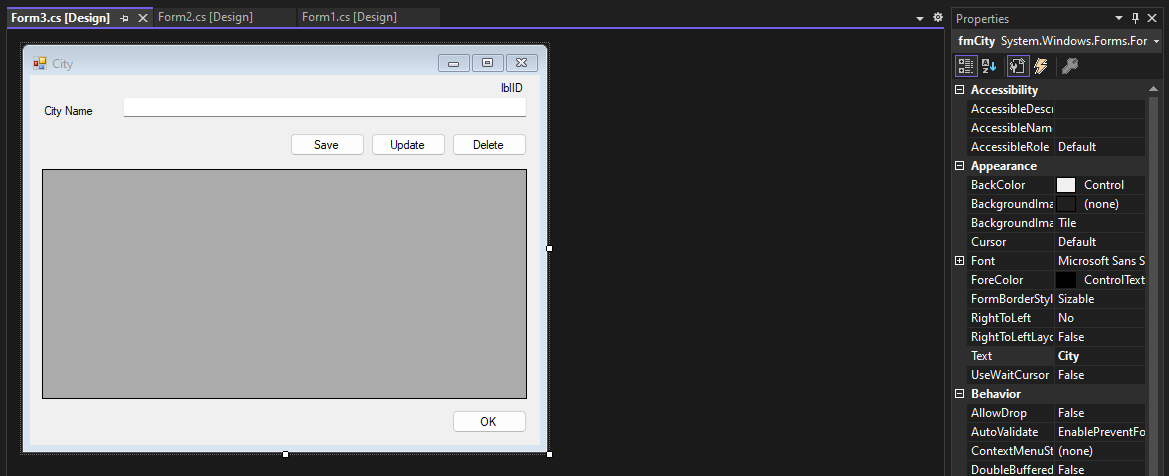
}

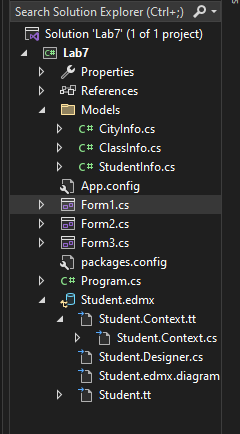
}

LAB7









Form1

using Lab7.Models;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.Entity.Core.Common.CommandTrees.ExpressionBuilder;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Serialization;

namespace Lab7

{

public partial class fmStudent : Form

{

public fmStudent()

{

InitializeComponent();

}

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

{

SetValueGender();

SetValueCity();

SetValueClass();

Display();

}

public void SetValueGender()

{

cmbGender.Items.Add("Male");

cmbGender.Items.Add("Female");

cmbGender.Items.Add("Other");

}

public void SetValueCity()

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

List<City> \_city = new List<City>();

\_city = \_stEntity.Cities.ToList();

cmbCity.DataSource = \_city;

cmbCity.DisplayMember = "CityName";

cmbCity.ValueMember = "Id";

}

}

public void SetValueClass()

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

List<Class> \_class = new List<Class>();

\_class = \_stEntity.Classes.ToList();

cmbClass.DataSource = \_class;

cmbClass.DisplayMember = "ClassID";

cmbClass.ValueMember = "Id";

}

}

public void Display()

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

List<StudentInfo> \_studentList = new List<StudentInfo>();

\_studentList = \_stEntity.StudentDetails.Select(x => new StudentInfo

{

Id = x.Id,

Name = x.Name,

Age = x.Age,

Gender = x.Gender,

ClassName = \_stEntity.Classes.Where(y => y.Id == x.ClassID).Select(y => y.ClassID).FirstOrDefault(),

CityName = \_stEntity.Cities.Where(y => y.Id == x.CityID).Select(y => y.CityName).FirstOrDefault()

}).ToList();

dgvStudents.DataSource = \_studentList;

}

}

public void ClearFields()

{

txtName.Text = "";

txtAge.Text = "";

cmbCity.SelectedIndex = -1;

cmbClass.SelectedIndex = -1;

cmbGender.SelectedIndex = -1;

}

public bool SaveStudentDetails(StudentDetail studentDetail)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

\_stEntity.StudentDetails.Add(studentDetail);

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool UpdateStudentDetails(StudentDetail studentDetail)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

StudentDetail \_st = \_stEntity.StudentDetails.Where(x => x.Id == studentDetail.Id).Select(x => x).FirstOrDefault();

\_st.Name = studentDetail.Name;

\_st.Age = studentDetail.Age;

\_st.CityID = studentDetail.CityID;

\_st.ClassID = studentDetail.ClassID;

\_st.Gender = studentDetail.Gender;

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool DeleteStudentDetails(int id)

{

bool result = false;

try

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

StudentDetail \_st = \_stEntity.StudentDetails.Find(id);

if (\_st != null)

{

\_stEntity.StudentDetails.Remove(\_st);

\_stEntity.SaveChanges();

result = true;

}

else

{

result = false;

}

}

}

catch

{

result = false;

}

return result;

}

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

{

StudentDetail st = new StudentDetail();

st.Name = txtName.Text;

st.Age = Convert.ToInt32(txtAge.Text);

st.CityID = Convert.ToInt32(cmbCity.SelectedValue);

st.Gender = cmbGender.SelectedItem.ToString();

st.ClassID = Convert.ToInt32(cmbClass.SelectedValue);

bool result = SaveStudentDetails(st);

if (result)

{

MessageBox.Show("Saved successfully!", "Save", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void dgvStudents\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgvStudents.Rows.Count > 0)

{

foreach (DataGridViewRow row in dgvStudents.SelectedRows)

{

lblID.Text = row.Cells[0].Value.ToString();

txtName.Text = row.Cells[1].Value.ToString();

txtAge.Text = row.Cells[2].Value.ToString();

cmbGender.SelectedItem = row.Cells[3].Value.ToString();

cmbClass.SelectedIndex = cmbClass.FindStringExact(row.Cells[4].Value.ToString());

cmbCity.SelectedIndex = cmbCity.FindStringExact(row.Cells[5].Value.ToString());

}

}

}

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

{

try

{

StudentDetail st = new StudentDetail();

st.Id = Convert.ToInt32(lblID.Text);

st.Name = txtName.Text;

st.Age = Convert.ToInt32(txtAge.Text);

st.CityID = Convert.ToInt32(cmbCity.SelectedValue);

st.ClassID = Convert.ToInt32(cmbClass.SelectedValue);

st.Gender = cmbGender.SelectedItem.ToString();

bool result = UpdateStudentDetails(st);

if (result)

{

MessageBox.Show("Update successfully!", "Update", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

try

{

int id = Convert.ToInt32(lblID.Text);

bool result = DeleteStudentDetails(id);

if (result)

{

MessageBox.Show("Delete successfully!", "Delete", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

fmClass fm = new fmClass();

fm.Show();

}

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

{

fmCity fm = new fmCity();

fm.Show();

}

}

}

Form2

using Lab7.Models;

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;

using System.Xml.Linq;

namespace Lab7

{

public partial class fmClass : Form

{

public fmClass()

{

InitializeComponent();

}

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

{

fmStudent fm = new fmStudent();

fm.SetValueClass();

this.Close();

}

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

{

Display();

}

public void Display()

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

List<ClassInfo> \_classList = new List<ClassInfo>();

\_classList = \_stEntity.Classes.Select(x => new ClassInfo

{

Id = x.Id,

ClassID = x.ClassID,

ClassName = x.ClassName

}).ToList();

dgvClass.DataSource = \_classList;

}

}

public void ClearFields()

{

txtClassID.Text = "";

txtClassName.Text = "";

}

public bool SaveClass(Class cl)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

\_stEntity.Classes.Add(cl);

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool UpdateClass(Class cl)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

Class \_cl = \_stEntity.Classes.Where(x => x.Id == cl.Id).Select(x => x).FirstOrDefault();

\_cl.ClassID = cl.ClassID;

\_cl.ClassName = cl.ClassName;

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool DeleteClass(int id)

{

bool result = false;

try

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

Class \_cl = \_stEntity.Classes.Find(id);

if (\_cl != null)

{

\_stEntity.Classes.Remove(\_cl);

\_stEntity.SaveChanges();

result = true;

}

else

{

result = false;

}

}

}

catch

{

result = false;

}

return result;

}

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

{

Class cl = new Class();

cl.ClassID = txtClassID.Text;

cl.ClassName = txtClassName.Text;

bool result = SaveClass(cl);

if (result)

{

MessageBox.Show("Saved successfully!", "Save", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

try

{

Class cl = new Class();

cl.Id = Convert.ToInt32(lblID.Text);

cl.ClassID = txtClassID.Text;

cl.ClassName = txtClassName.Text;

bool result = UpdateClass(cl);

if (result)

{

MessageBox.Show("Update successfully!", "Update", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

try

{

int id = Convert.ToInt32(lblID.Text);

bool result = DeleteClass(id);

if (result)

{

MessageBox.Show("Delete successfully!", "Delete", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void dgvClass\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgvClass.Rows.Count > 0)

{

foreach (DataGridViewRow row in dgvClass.SelectedRows)

{

lblID.Text = row.Cells[0].Value.ToString();

txtClassID.Text = row.Cells[1].Value.ToString();

txtClassName.Text = row.Cells[2].Value.ToString();

}

}

}

}

}

Form3

using Lab7.Models;

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 Lab7

{

public partial class fmCity : Form

{

public fmCity()

{

InitializeComponent();

}

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

{

fmStudent fm = new fmStudent();

fm.SetValueCity();

this.Close();

}

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

{

Display();

}

public void Display()

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

List<CityInfo> \_cityList = new List<CityInfo>();

\_cityList = \_stEntity.Cities.Select(x => new CityInfo

{

Id = x.Id,

CityName = x.CityName

}).ToList();

dgvCity.DataSource = \_cityList;

}

}

public void ClearFields()

{

txtCityName.Text = "";

}

public bool SaveCity(City ct)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

\_stEntity.Cities.Add(ct);

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool UpdateCity(City ct)

{

bool result = false;

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

City \_ct = \_stEntity.Cities.Where(x => x.Id == ct.Id).Select(x => x).FirstOrDefault();

\_ct.CityName = ct.CityName;

\_stEntity.SaveChanges();

result = true;

}

return result;

}

public bool DeleteCity(int id)

{

bool result = false;

try

{

using (StudentInformationEntities \_stEntity = new StudentInformationEntities())

{

City \_ct = \_stEntity.Cities.Find(id);

if (\_ct != null)

{

\_stEntity.Cities.Remove(\_ct);

\_stEntity.SaveChanges();

result = true;

}

else

{

result = false;

}

}

}

catch

{

result = false;

}

return result;

}

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

{

City ct = new City();

ct.CityName = txtCityName.Text;

bool result = SaveCity(ct);

if (result)

{

MessageBox.Show("Saved successfully!", "Save", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

try

{

City ct = new City();

ct.Id = Convert.ToInt32(lblID.Text);

ct.CityName = txtCityName.Text;

bool result = UpdateCity(ct);

if (result)

{

MessageBox.Show("Update successfully!", "Update", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

try

{

int id = Convert.ToInt32(lblID.Text);

bool result = DeleteCity(id);

if (result)

{

MessageBox.Show("Delete successfully!", "Delete", MessageBoxButtons.OK, MessageBoxIcon.Information);

ClearFields();

Display();

}

else

{

MessageBox.Show("Something went wrong! Please try again!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Something went wrong! Please try again!\n" + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void dgvCity\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgvCity.Rows.Count > 0)

{

foreach (DataGridViewRow row in dgvCity.SelectedRows)

{

lblID.Text = row.Cells[0].Value.ToString();

txtCityName.Text = row.Cells[1].Value.ToString();

}

}

}

}

}

CityInfo.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab7.Models

{

public class CityInfo

{

public int Id { get; set; }

public string CityName { get; set; }

}

}

ClassInfo.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab7.Models

{

public class ClassInfo

{

public int Id { get; set; }

public string ClassID { get; set; }

public string ClassName { get; set; }

}

}

StudentInfo.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab7.Models

{

public class StudentInfo

{

public int Id { get; set; }

public string Name { get; set; }

public int? Age { get; set; }

public string Gender { get; set; }

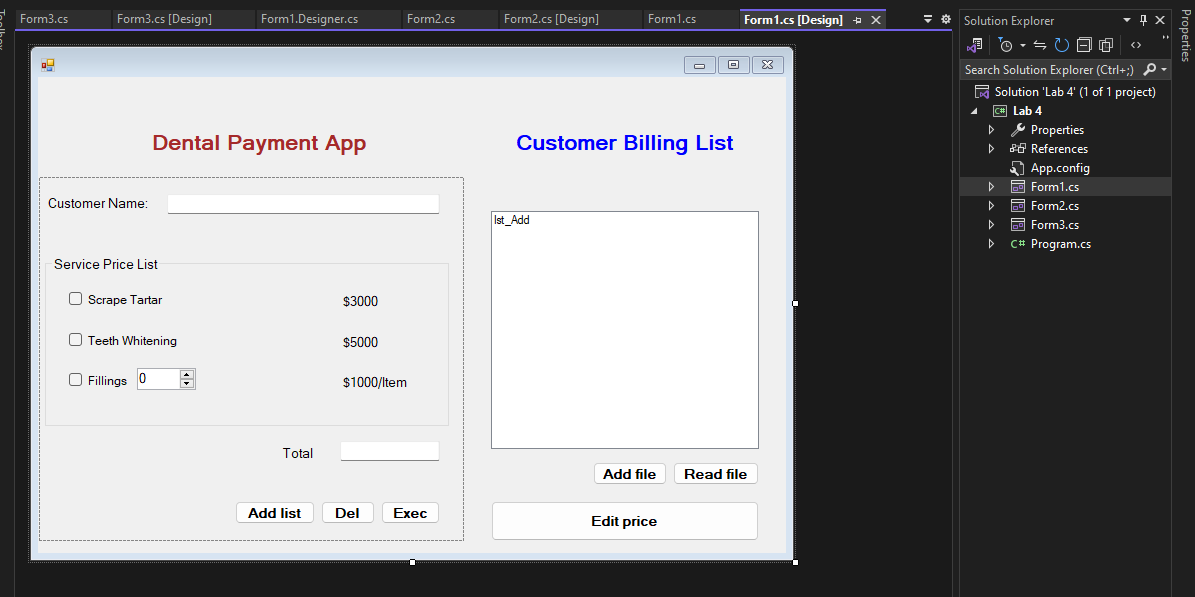
public string ClassName { get; set; }

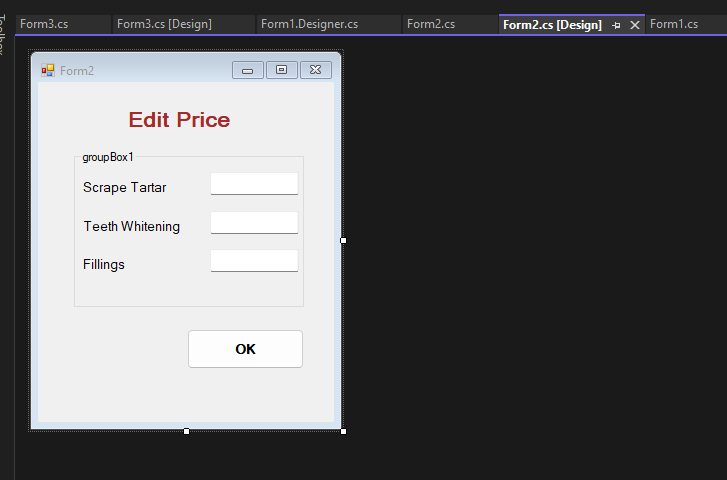
public string CityName { get; set; }

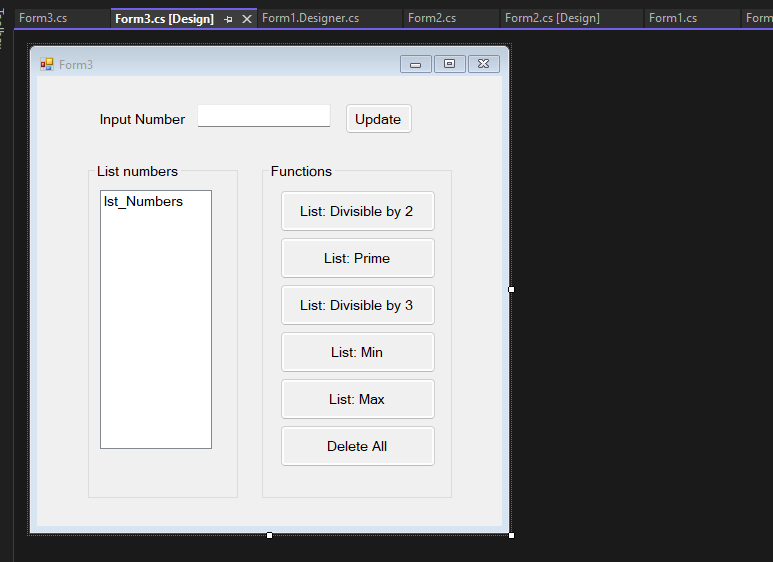
}

}

LAB4







LAB5

