

## Form Code Goes Here

```

namespace SampleDataSetProject2
{
    public partial class frmCourseDetails : Form
    {
        clsCourseDetails obj=new clsCourseDetails();
        public frmCourseDetails()
        {
            InitializeComponent();
        }

        private void frmCourseDetails_Load(object sender, EventArgs e)
        {
            try
            {
                GetCourseDetails();
                SetControlValues();
                toolTipCourseID.SetToolTip(txtCourseID,"Check the checkbox to enable for searhing ...");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void SetControlValues()
        {
            try
            {
                txtCourseID.Text = obj.CourseID.ToString();
                txtCourseName.Text = obj.CourseName.ToString();
                txtCourseAbbreviation.Text = obj.CourseAbbreviation.ToString();
                txtDescription.Text = obj.CourseDescription.ToString();
                txtNoOfSems.Text = obj.CourseNoOfSems.ToString();
            }
            catch { }
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void ClearControlValues()
{
    try
    {
        txtCourseID.Text = "";
        txtCourseName.Text = "";
        txtCourseAbbreviation.Text = "";
        txtDescription.Text = "";
        txtNoOfSems.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void GetControlValues()
{
    try
    {
        obj.CourseName = txtCourseName.Text;
        obj.CourseAbbreviation = txtCourseAbbreviation.Text;
        obj.CourseDescription = txtDescription.Text;
        obj.CourseNoOfSems = Byte.Parse(txtNoOfSems.Text);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void RefreshRecords()
{
    try
    {
        GetCourseDetails();
        SetControlValues();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void GetCourseDetails()
{
    try
    {
        obj.GetAllCourseDetails();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnAddNew_Click(object sender, EventArgs e)
{
    try
    {
        ClearControlValues();
    }
    catch (Exception ex)
    {

```

```

        MessageBox.Show(ex.Message);
    }
}

private void btnSave_Click(object sender, EventArgs e)
{
    try
    {
        GetControlValues();
        obj.SaveCourseDetails();
        MessageBox.Show("Record Saved");
        RefreshRecords();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnEdit_Click(object sender, EventArgs e)
{
    try
    {
        GetControlValues();
        obj.UpdateCourseDetails();
        MessageBox.Show("Record Updated");
        RefreshRecords();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnDelete_Click(object sender, EventArgs e)
{
    try
    {
        obj.DeleteCourseDetails();
        MessageBox.Show("Record Deleted");
        RefreshRecords();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnFirst_Click(object sender, EventArgs e)
{
    try
    {
        obj.FirstRecord();
        SetControlValues();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnNext_Click(object sender, EventArgs e)
{
    try
    {
        obj.NextRecord();
        SetControlValues();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

    }
}

private void btnPrevious_Click(object sender, EventArgs e)
{
    try
    {
        obj.PreviousRecord();
        SetControlValues();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnLast_Click(object sender, EventArgs e)
{
    try
    {
        obj.LastRecord();
        SetControlValues();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnClose_Click(object sender, EventArgs e)
{
    try
    {
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void chkSearch_CheckedChanged(object sender, EventArgs e)
{
    try
    {
        if (chkSearch.Checked == true)
        {
            txtCourseID.ReadOnly = false;
            txtCourseID.Focus();
        }
        else
            txtCourseID.ReadOnly = true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void btnSearch_Click(object sender, EventArgs e)
{
    try
    {
        obj.CourseID = Int32.Parse(txtCourseID.Text);
        obj.GetAllCourseDetailsByCourseID();
        SetControlValues();
        chkSearch.Checked = false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```
    }  
  }  
}
```

---

### (1) clsConnectionString.cs

```
public class clsConnectionString  
{  
    protected string StrCon = "server=.;database=pubs;user id=sa;pwd=";  
}
```

---

### 2) clsCourseDetails.cs

```
public class clsCourseDetails:clsConnectionString // BL  
{  
    private int _CourseID;  
    private string _CourseName;  
    private string _CourseAbbreviation;  
    private string _CourseDescription;  
    private Byte _CourseNoOFSEms;  
    private DataSet _DS;  
    private string _tname;  
    private int _RecPtr;  
  
    public clsCourseDetails()  
    {  
        _CourseID = 0;  
        _CourseName = "";  
        _CourseAbbreviation = "";  
        _CourseDescription = "";  
        _CourseNoOFSEms = 0;  
        _DS = null;  
        _tname = "tblCourseMaster";  
        _RecPtr = 0;  
    }  
  
    public clsCourseDetails(int CourseID, string CourseName, string CourseAbbreviation, string  
CourseDescription, Byte CourseNoOFSEms, DataSet ds,string tnm,int R)  
    {  
        try  
        {  
            this._CourseID = CourseID;  
            this._CourseName = CourseName;  
            this._CourseAbbreviation = CourseAbbreviation;  
            this._CourseDescription = CourseDescription;  
            this._CourseNoOFSEms = CourseNoOFSEms;  
            this._DS = ds;  
            this._tname = tnm;  
            this._RecPtr = R;  
        }  
        catch (Exception ex)  
        {  
            throw new ArgumentException(ex.Message);  
        }  
    }  
  
    public int CourseID  
    {  
        set  
        {  
            try  
            {  
                checked  
                {  
                    this._CourseID = value;  
                }  
            }  
        }  
    }  
}
```

```
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
get
{
    return _CourseID;
}
}

public string CourseName
{
    set
    {
        this._CourseName = value;
    }
    get
    {
        return _CourseName;
    }
}

public string CourseAbbreviation
{
    set
    {
        this._CourseAbbreviation = value;
    }
    get
    {
        return _CourseAbbreviation;
    }
}

public string CourseDescription
{
    set
    {
        this._CourseDescription = value;
    }
    get
    {
        return _CourseDescription;
    }
}

public string TableName
{
    set
    {
        this._tname = value;
    }
    get
    {
        return _tname;
    }
}

public Byte CourseNoOfSems
{
    set
    {
        try
        {
            checked
            {
                this._CourseNoOfSems = value;
            }
        }
    }
}
```

```

        catch (Exception ex)
        {
            throw new ArgumentException(ex.Message);
        }
    }
    get
    {
        return this._CourseNoOFSEms;
    }
}

public void BindDataToDataSet(DataRow drow)
{
    try
    {
        drow[0] = this._CourseID;
        drow[1] = this._CourseName;
        drow[2] = this._CourseAbbreviation;
        drow[3] = this._CourseDescription;
        drow[4] = this._CourseNoOFSEms;
    } catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void BindDataToObject()
{
    try
    {
        DataRow drow = this._DS.Tables[0].Rows[this._RecPtr];
        this._CourseID = Int32.Parse(drow[0].ToString());
        this._CourseName = drow[1].ToString();
        this._CourseAbbreviation = drow[2].ToString();
        this._CourseDescription = drow[3].ToString();
        this._CourseNoOFSEms = Byte.Parse(drow[4].ToString());
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void SaveCourseDetails()
{
    try
    {
        DataRow drow = this._DS.Tables[0].NewRow();
        BindDataToDataSet(drow);
        this._DS.Tables[0].Rows.Add(drow);
        clsDBActions.UpdateDataSet(this._DS, this._tname);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void PrepareForUpdate()
{
    DataRow drow = this._DS.Tables[0].Rows[this._RecPtr];
    drow.BeginEdit();
    BindDataToDataSet(drow);
    drow.EndEdit();
}

public void UpdateCourseDetails()
{
    try
    {
        PrepareForUpdate();
    }
}

```

```

        clsDBActions.UpdateDataSet(this._DS, this._tname);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void DeleteCourseDetails()
{
    try
    {
        DataRow drow = this._DS.Tables[0].Rows[this._RecPtr];
        drow.Delete();
        clsDBActions.UpdateDataSet(this._DS, this._tname);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void FirstRecord()
{
    try
    {
        this._RecPtr = 0;
        BindDataToObject();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void LastRecord()
{
    try
    {
        this._RecPtr = this._DS.Tables[0].Rows.Count - 1;
        BindDataToObject();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void NextRecord()
{
    try
    {
        this._RecPtr++;
        if (this._RecPtr == this._DS.Tables[0].Rows.Count)
        {
            this._RecPtr--;
            throw new ArgumentException("This is the last record");
        }
        BindDataToObject();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void PreviousRecord()
{
    try
    {
        if (this._RecPtr > 0)

```

```

        this._RecPtr--;
    else
        throw new ArgumentException("This is the First Record");
    BindDataToObject();
}
catch (Exception ex)
{
    throw new ArgumentException(ex.Message);
}
}

public void GetAllCourseDetails()
{
    try
    {
        string SelectStatement = "Select * from tblCourseMaster";
        this._DS = clsDBActions.GetDataBaseRecords(StrCon, SelectStatement, "tblCourseMaster");
        this._RecPtr = 0;
        BindDataToObject();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public void GetAllCourseDetailsByCourseID()
{
    try
    {
        string SelectStatement = "Select * from tblCourseMaster where CourseID=" + this._CourseID;
        this._DS = clsDBActions.GetDataBaseRecords(StrCon, SelectStatement, "tblCourseMaster");
        this._RecPtr = 0;
        BindDataToObject();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}

```

---

#### 4) clsDBActions.cs

```

public sealed class clsDBActions // DAL
{
    private static SqlConnection SQLCon;
    private static SqlCommand SQLCmd;
    private static SqlDataAdapter SQLAdpt;
    private static SqlCommandBuilder SQLCb;
    private static SqlDataReader SQLDR;
    private static DataSet DS;

    static clsDBActions()
    {
        try
        {
            SQLCon = null;
            SQLCmd = null;
            SQLAdpt = null;
            SQLDR = null;
            SQLCb = null;
            DS = null;
        }
        catch (SqlException ex)
        {
            throw new ArgumentException(ex.Message);
        }
        catch (Exception ex)
        {
            throw new ArgumentException(ex.Message);
        }
    }
}

```

```

}

public static DataSet GetDataBaseRecords(string ConnectionString, string SelectQuery, string TableName)
{
    try
    {
        SQLAdpt = new SqlDataAdapter(SelectQuery, ConnectionString);
        SQLCb = new SqlCommandBuilder(SQLAdpt);
        DS = new DataSet();
        SQLAdpt.Fill(DS, TableName);
        return DS;
    }
    catch (SqlException ex)
    {
        throw new ArgumentException(ex.Message);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void RunSQLCommand(string ConnectionString, string DMLStatement)
{
    try
    {
        SQLCon = new SqlConnection(ConnectionString);
        SQLCon.Open();
        SQLCmd = new SqlCommand(DMLStatement, SQLCon);
        SQLCmd.ExecuteNonQuery();
    }
    catch (SqlException ex)
    {
        throw new ArgumentException(ex.Message);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
    finally
    {
        if (SQLCon != null)
            if (SQLCon.State == ConnectionState.Open)
                SQLCon.Close();
    }
}

public static void UpdateDataSet(DataSet ds, string tname)
{
    try
    {
        SQLAdpt.Update(ds, tname);
    }
    catch (SqlException ex)
    {
        throw new ArgumentException(ex.Message);
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}

```

---