Student Management System

**Project Overview**

**Features to Learn (CRUD)**:

* **C**reate → Add a new student
* **R**ead → View all students
* **U**pdate → Edit student details
* **D**elete → Remove a student

**Tech Stack:**

* **C# WinForms**
* **SQL Server LocalDB** (built into Visual Studio)
* **ADO.NET** for database operations

**Step 1: Setup the Database**

1. Open **SQL Server Management Studio** or **Visual Studio SQL Server Object Explorer**
2. Create a new **Database**: StudentDB
3. Create a **Students Table**:

sql

CREATE TABLE Students (

Id INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100),

Age INT,

Course NVARCHAR(50)

);

**Step 2: Create a WinForms Project**

1. Open **Visual Studio** → **Create New Project** → **WinForms App (.NET Framework)**
2. Name it: StudentManagementSystem

**Step 3: Design the Form**

* Controls needed:
  + **TextBox**: txtName, txtAge, txtCourse
  + **Buttons**: btnAdd, btnUpdate, btnDelete, btnLoad
  + **DataGridView**: dgvStudents (to display students)

**Step 4: Connect to Database**

**App.config** connection string (for LocalDB):

xml

<configuration>

<connectionStrings>

<add name="StudentDB"

connectionString="Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\StudentDB.mdf;Integrated Security=True"

providerName="System.Data.SqlClient"/>

</connectionStrings>

</configuration>

**Step 5: Code the CRUD Operations**

**Add Namespaces**:

csharp

using System.Data;

using System.Data.SqlClient;

**Global Connection String**:

csharp

CopyEdit

string connStr = ConfigurationManager.ConnectionStrings["StudentDB"].ConnectionString;

**1. Load Students**

csharp

private void LoadStudents()

{

using (SqlConnection con = new SqlConnection(connStr))

{

SqlDataAdapter da = new SqlDataAdapter("SELECT \* FROM Students", con);

DataTable dt = new DataTable();

da.Fill(dt);

dgvStudents.DataSource = dt;

}

}

Call LoadStudents(); in **Form Load** event and after each CRUD operation.

**2. Add Student**

csharp

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

{

using (SqlConnection con = new SqlConnection(connStr))

{

string query = "INSERT INTO Students (Name, Age, Course) VALUES (@Name, @Age, @Course)";

SqlCommand cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@Name", txtName.Text);

cmd.Parameters.AddWithValue("@Age", int.Parse(txtAge.Text));

cmd.Parameters.AddWithValue("@Course", txtCourse.Text);

con.Open();

cmd.ExecuteNonQuery();

}

LoadStudents();

}

**3. Update Student**

csharp

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

{

if (dgvStudents.SelectedRows.Count > 0)

{

int id = Convert.ToInt32(dgvStudents.SelectedRows[0].Cells["Id"].Value);

using (SqlConnection con = new SqlConnection(connStr))

{

string query = "UPDATE Students SET Name=@Name, Age=@Age, Course=@Course WHERE Id=@Id";

SqlCommand cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@Name", txtName.Text);

cmd.Parameters.AddWithValue("@Age", int.Parse(txtAge.Text));

cmd.Parameters.AddWithValue("@Course", txtCourse.Text);

cmd.Parameters.AddWithValue("@Id", id);

con.Open();

cmd.ExecuteNonQuery();

}

LoadStudents();

}

}

**4. Delete Student**

csharp

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

{

if (dgvStudents.SelectedRows.Count > 0)

{

int id = Convert.ToInt32(dgvStudents.SelectedRows[0].Cells["Id"].Value);

using (SqlConnection con = new SqlConnection(connStr))

{

string query = "DELETE FROM Students WHERE Id=@Id";

SqlCommand cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@Id", id);

con.Open();

cmd.ExecuteNonQuery();

}

LoadStudents();

}

}

**6. Select Row to Edit**

To populate TextBoxes when a row is clicked:

csharp

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

{

if (e.RowIndex >= 0)

{

DataGridViewRow row = dgvStudents.Rows[e.RowIndex];

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

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

txtCourse.Text = row.Cells["Course"].Value.ToString();

}

}