FE Ado.Net BE

Entity Framework

**BE PART**

create database StudentDB

use StudentDb

create table Student(rollNo int primary key,

name varchar(20) not null,

batchCode varchar(10) not null,

marks int)

Select \* from Student

create proc GetStudents

As

Begin

Select \* from Student

End

create proc InsertStudent(@rollno int,

@name varchar(20),

@batchCode varchar(10),

@marks int)

As

Begin

insert into Student values(@rollno , @name, @batchCode, @marks)

end

**FE PART**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace ADODemo

{

class Program

{

static SqlConnection connection;

static SqlCommand command;

static SqlConnection Getconnection()

{

string connectionString = "data source=admivm\\SQLEXPRESS;initial catalog=StudentDb;user id=sa;password=pass@123";

SqlConnection connection = new SqlConnection(connectionString);

return connection;

}

static void Main(string[] args)

{

string choice = "y";

while (choice == "y")

{

Console.WriteLine("1. List all Records");

Console.WriteLine("2. Insert Record");

Console.WriteLine("3. Update Record");

Console.WriteLine("4. Delete Records");

Console.WriteLine("5. Search Record");

Console.WriteLine("Enter Choice");

int ch = int.Parse(Console.ReadLine());

switch (ch)

{

case 1: GetStudents(); break;

case 2:

{

Console.WriteLine("Enter RollNo");

int rollno = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

string name = Console.ReadLine();

Console.WriteLine("Enter Batch Code");

string batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

int marks = int.Parse(Console.ReadLine());

InsertRecord(rollno, name, batch, marks); break;

}

case 3:

{

Console.WriteLine("Enter RollNo for which to edit Record");

int rollno = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Batch Code");

string batch = Console.ReadLine();

Console.WriteLine("Enter Marks");

int marks = int.Parse(Console.ReadLine());

UpdateRecord(rollno, batch, marks); break;

}

case 4:

{

Console.WriteLine("Enter RollNo for which to delete Record");

int rollno = int.Parse(Console.ReadLine());

DeleteRecord(rollno); break;

}

case 5:

{

Console.WriteLine("Enter RollNo for which to find Record");

int rollno = int.Parse(Console.ReadLine());

GetStudent(rollno); break;

}

}

Console.WriteLine("Do you want to repeat any process");

choice = Console.ReadLine();

}

}

//public static void GetStudents()

//{

// connection = Getconnection();

// command = new SqlCommand("Select \* from Student", connection);

// connection.Open();

// SqlDataReader reader = command.ExecuteReader();

// if (reader.HasRows)

// {

// while (reader.Read())

// {

// Console.WriteLine(reader["rollNo"].ToString() + " " + reader[1]);

// }

// }

// else

// Console.WriteLine("No Record");

// connection.Close();

//}

public static void GetStudents()

{

connection = Getconnection();

command = new SqlCommand("GetStudents", connection);

command.CommandType = System.Data.CommandType.StoredProcedure;

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

Console.WriteLine(reader["rollNo"].ToString() + " " + reader[1]);

}

}

else

Console.WriteLine("No Record");

connection.Close();

}

//public static void InsertRecord(int rollno, string name, string batchCode, int marks)

//{

// connection = Getconnection();

// command = new SqlCommand("Inse into Student values(@rollno, @name , @batchcode, @marks)", connection);

// command.Parameters.AddWithValue("@rollno", rollno);

// command.Parameters.AddWithValue("@name", name);

// command.Parameters.AddWithValue("@batchcode", batchCode);

// command.Parameters.AddWithValue("@marks", marks);

// connection.Open();

// command.ExecuteNonQuery();

// connection.Close();

//}

public static void InsertRecord(int rollno, string name, string batchCode, int marks)

{

connection = Getconnection();

command = new SqlCommand("InsertStudent", connection);

command.CommandType = System.Data.CommandType.StoredProcedure;

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

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

command.Parameters.AddWithValue("@batchcode", batchCode);

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

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

public static void UpdateRecord(int rollno, string batchCode, int marks)

{

connection = Getconnection();

command = new SqlCommand("Update Student set batchCode= @batchcode, marks = @marks where rollno=@rollno", connection);

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

command.Parameters.AddWithValue("@batchcode", batchCode);

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

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

public static void DeleteRecord(int rollno)

{

connection = Getconnection();

command = new SqlCommand("Delete Student where rollno=@rollno", connection);

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

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

public static void GetStudent(int rollno)

{

connection = Getconnection();

command = new SqlCommand("Select \* from Student where rollno=@rollno", connection);

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

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

Console.WriteLine(reader["rollNo"].ToString() + " " + reader[1]);

}

}

else

Console.WriteLine("No Record");

connection.Close();

}

}

}