| **Name of Student:** Ajay Karthikesan | | | |
| --- | --- | --- | --- |
| **Roll Number:** 57 | | **Assignment Number:** 7 | |
| **Aim of Assignment:**  Design a webpage to display the use of LINQ | | | |
| **DOP:** | | **DOS:** 3.6.23 | |
| **CO Mapped:**  CO2 | **PO Mapped:**  PO3, PO5, PSO1, PSO2 | **Faculty Signature:** | **Marks:** |

## 

## Practical No. 7

**Aim:** Design a webpage to display the use of LINQ

**Theory:**

LINQ to SQL Classes:

LINQ to SQL (also known as DLINQ) is an electrifying part of

Language Integrated Query as this allows querying data in SQL

server database by using usual LINQ expressions. It also allows

you to update, delete and insert data, but the only drawback

from which it suffers is its limitation to the SQL server database.

However, there are many benefits of LINQ to SQL over

ADO.NET like reduced complexity, few lines of coding and

many more. Below is a diagram showing the execution

architecture of LINQ to SQL.

Entity Data Model/Framework:

The Entity Data Model (EDM) is a set of concepts that describe

the structure of data, regardless of its stored form. The EDM

borrows from the EntityRelationship Model described by Peter

Chen in 1976, but it also builds on the Entity-Relationship

Model and extends its traditional uses.

EDM supports a set of primitive data types that define properties

in a conceptual model. We need to consider 3 core parts which

form the basis for Entity Framework and collectively it is known

as Entity Data Model. Following are the three core parts of

EDM.

The Storage Schema Model:

• The Storage Model also called as Storage Schema Definition

Layer (SSDL) represents the schematic representation of the

backend data store.

The Conceptual Model:

• The Conceptual Model also called as Conceptual Schema

Definition Layer (CSDL) is the real entity model, against which

we write our queries.

The Mapping Model:

• Mapping Layer is just a mapping between the Conceptual

model and the Storage model.

• The logical schema and its mapping with the physical schema

is represented as an EDM.

➢ Visual Studio also provides Entity Designer, for visual

creation of the EDM and the mapping specification.

➢ The output of the tool is the XML file (\*.edmx) specifying

the schema and mapping.

➢ The Edmx file contains Entity Framework metadata artifacts.

**Code:**

File: WebForm1.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Practical7.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="IDTxt" runat="server"></asp:TextBox>&nbsp;&nbsp;

<asp:Button ID="FindIDBtn" runat="server" Text="Find ID"

onclick="FindIDBtn\_Click" />

&nbsp;&nbsp;

<asp:Button ID="AllRecBtn" runat="server" Text="Show Every Record"

onclick="AllRecBtn\_Click" />

<br />

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"

DataKeyNames="id">

<Columns>

<asp:BoundField DataField="id" HeaderText="id" ReadOnly="True" SortExpression="id" />

<asp:BoundField DataField="name" HeaderText="name" SortExpression="name" />

<asp:CheckBoxField DataField="is\_outdoor" HeaderText="is\_outdoor" SortExpression="is\_outdoor" />

<asp:BoundField DataField="pcount" HeaderText="pcount" SortExpression="pcount" />

</Columns>

</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:collConnectionString %>"

SelectCommand="SELECT \* FROM [ajayk57\_games]"></asp:SqlDataSource>

</div>

</form>

</body>

</html>

File: WebForm1.aspx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical7

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void FindIDBtn\_Click(object sender, EventArgs e)

{

try

{

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

DataClasses1DataContext dbCon = new

DataClasses1DataContext();

GridView1.DataSource = (from game in dbCon.ajayk57\_games

where

game.id == id

select game);

GridView1.DataBind();

}catch(Exception){

Response.Write("Please enter a valid, integer");

}

}

protected void AllRecBtn\_Click(object sender, EventArgs e)

{

DataClasses1DataContext dbCon=new DataClasses1DataContext();

GridView1.DataSource = (from game in dbCon.ajayk57\_games select game);

GridView1.DataBind();

}

}

}

File: WebForm2.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="Practical7.WebForm2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="IDTxt" runat="server"></asp:TextBox>&nbsp;&nbsp;

<asp:Button ID="FindIDBtn" runat="server" Text="Find ID"

onclick="FindIDBtn\_Click" />

&nbsp;&nbsp;

<asp:Button ID="AllRecBtn" runat="server" Text="Show Every Record"

onclick="AllRecBtn\_Click" />

<br />

<asp:GridView ID="GridView1" runat="server">

</asp:GridView>

</div>

</form>

</body>

</html>

File: WebForm2.aspx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical7

{

public partial class WebForm2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void FindIDBtn\_Click(object sender, EventArgs e)

{

try

{

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

DataClasses1DataContext dbCon = new

DataClasses1DataContext();

GridView1.DataSource = (from book in dbCon.ajayk57\_libraries

where

book.id == id

select book);

GridView1.DataBind();

}

catch (Exception)

{

Response.Write("Please enter a valid, integer");

}

}

protected void AllRecBtn\_Click(object sender, EventArgs e)

{

DataClasses1DataContext dbCon = new DataClasses1DataContext();

GridView1.DataSource = (from book in dbCon.ajayk57\_libraries select book);

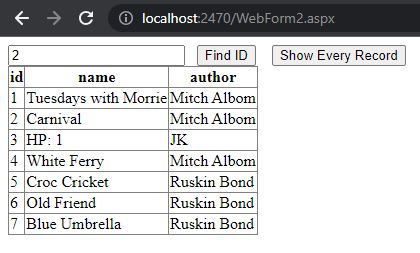
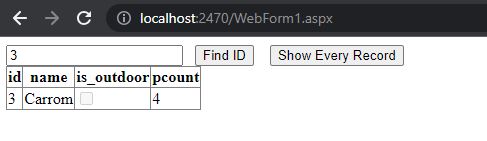
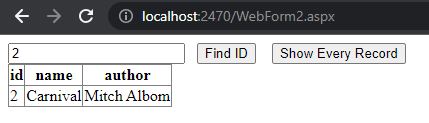
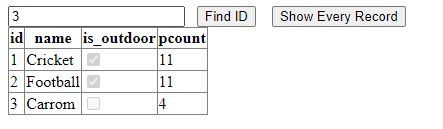
GridView1.DataBind();

}

}

}

**Output:**



**Conclusion:**

I learnt how to use LINQ to make web applications.