Laboratory Practical Report

of

**Visual Programming with C#**

**(ICT ED 465)**

Submitted To

**TRIBHUVAN UNIVERSITY**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

Under the guidance of

**Er. Santosh Dahal**

Lecturer

Sukuna Multiple Campus

**SUKUNA MULTIPLE CAMPUS**

Sundarharaincha-12, Morang, Nepal

2080

**CERTIFICATE**

This is to certify that the Laboratory Practical Report

of

**Visual Programming with C#**

**(ICT ED 465)**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

is a bonafide record of experiments carried out by him/her under by guidance.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Er. Santosh Dahal

Lecturer

Sukuna Multiple Campus

Sundarharaincha-12, Morang

(Internal Examiner)

Submitted for the Final Examination on: 2080/08/20

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lecturer

(External Examiner

**Table of Contents**

[**Introduction** 1](#_Toc152546744)

[**Step involve in database connection** 1](#_Toc152546745)

[**Example 1: Using a SQL command and a data reader** 1](#_Toc152546746)

[**Example 2: Using a SQL command and a data table** 2](#_Toc152546748)

[**Example 3: Using a LINQ to SQL data context and a lambda expression** 3](#_Toc152546749)

[**Example 4: Using Entity Framework Core and a LINQ query** 4](#_Toc152546750)

[References 5](#_Toc152546751)

# **Introduction**

Accessing Data from a database is one of the important aspects of any programming language.

It can work with different types of databases. It can work with the most common databases such as Oracle and Microsoft SQL Server. Most of the C# programs work with Microsoft SQL Server. It also can work with new forms of databases such as MongoDB.

# **Steps involve in database connection**

1. Import the necessary namespaces for working with databases, such as System.Data and System.Data.SqlClient.
2. Create a connection string that specifies the data source, database name, credentials, and other optional parameters for the database you want to connect to.
3. Create an instance of the SqlConnection class and pass the connection string as a parameter to its constructor.
4. Open the connection by calling the Open method on the SqlConnection object.
5. Perform database operations using SQL commands, data adapters, data readers, data tables, or other classes that interact with the database.
6. Close the connection by calling the Close method on the SqlConnection object.

# **Example 1: Using a SQL command and a data reader**

using System;

using System.Data;

using System.Data.SqlClient;

namespace Example1{

class Program{

static void Main(string[] args){

string connectionString = "server=(local)\\SQLExpress;database=Northwind;integrated Security=SSPI;";

using (SqlConnection connection = new SqlConnection(connectionString)){

string query = "SELECT TOP 5 \* FROM dbo.Customers ORDER BY CustomerID";

using (SqlCommand command = new SqlCommand(query, connection)){

connection.Open();

using (SqlDataReader reader = command.ExecuteReader()){

while (reader.Read())

{

Console.WriteLine("{0}\t{1}\t{2}", reader[0], reader[1], reader[2]);

}

}

connection.Close();

}

}

}

}

}

# **Example 2: Using a SQL command and a data table**

using System;

using System.Data;

using System.Data.SqlClient;

namespace Example2

{

class Program

{

static void Main(string[] args)

{

string connectionString = "server=(local)\\SQLExpress;database=Northwind;integrated Security=SSPI;";

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT TOP 5 \* FROM dbo.Customers ORDER BY CustomerID";

using (SqlCommand command = new SqlCommand(query, connection))

{

DataTable table = new DataTable("Customers");

using (SqlDataAdapter adapter = new SqlDataAdapter(command))

{

connection.Open();

adapter.Fill(table);

connection.Close();

}

foreach (DataRow row in table.Rows)

{

Console.WriteLine("{0}\t{1}\t{2}", row[0], row[1], row[2]);

}

}

}

}

}

}

# **Example 3: Using a LINQ to SQL data context and a lambda expression**

using System;

using System.Data.Linq;

using System.Linq;

namespace Example3

{

class Program

{

static void Main(string[] args)

{ string connectionString = "server=(local)\\SQLExpress;database=Northwind;integrated Security=SSPI;";

using (DataContext context = new DataContext(connectionString))

{

Table<Customer> customers = context.GetTable<Customer>();

var query = customers.OrderBy(c => c.CustomerID).Take(5);

foreach (var customer in query)

{ Console.WriteLine("{0}\t{1}\t{2}", customer.CustomerID, customer.CompanyName, customer.ContactName);

}

}

}

}

// Define a class to represent a customer entity

[Table(Name = "dbo.Customers")]

public class Customer

{

[Column(IsPrimaryKey = true)]

public string CustomerID { get; set; }

[Column]

public string CompanyName { get; set; }

[Column]

public string ContactName { get; set; }

}

}

# **Example 4: Using Entity Framework Core and a LINQ query**

using System;

using System.Linq;

using Microsoft.EntityFrameworkCore;

namespace Example4

{

class Program

{

static void Main(string[] args)

{

string connectionString = "server=(local)\\SQLExpress;database=Northwind;integrated Security=SSPI;";

using (var context = new NorthwindContext(connectionString))

{

var query = from c in context.Customers

orderby c.CustomerId

select c;

foreach (var customer in query.Take(5))

{

Console.WriteLine("{0}\t{1}\t{2}", customer.CustomerId, customer.CompanyName, customer.ContactName);

}

}

}

}

public class NorthwindContext : DbContext

{

public NorthwindContext(string connectionString) : base(GetOptions(connectionString))

{

}

private static DbContextOptions GetOptions(string connectionString)

{

return SqlServerDbContextOptionsExtensions.UseSqlServer(new DbContextOptionsBuilder(), connectionString).Options;

}

public DbSet<Customer> Customers { get; set; }

}

public class Customer

{

public string CustomerId { get; set; }

public string CompanyName { get; set; }

public string ContactName { get; set; }

}

}

# References

(n.d.). Retrieved from https://www.guru99.com/c-sharp-access-database.html

(n.d.). Retrieved from https://learn.microsoft.com/en-us/azure/azure-sql/database/connect-query-dotnet-core?view=azuresql

(n.d.). Retrieved from https://stackoverflow.com/questions/1345508/how-do-i-connect-to-a-sql-database-from-c

(n.d.). Retrieved from http://blogs.msdn.com/adonet/default.aspx%29

(n.d.). Retrieved from https://www.c-sharpcorner.com/blogs/c-sharp-database-connection