ASSIGNMENT

EAPEN THOMAS
30

INT MCA – S6

VISUAL PROGRAMMING

1) Create a console app demonstrating CRUD operations using ADO.NET Dataset with a sample database.

Program

```
using System;
using System.Data;
using System.Data.SqlClient;


namespace AdoNetCRUDExample
{
   class Program
   {
      static string connectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=SampleDB;Integrated Security=True";


      static void Main(string[] args)
      {
         while (true)
         {
            Console.WriteLine("1. Create");
            Console.WriteLine("2. Read");
            Console.WriteLine("3. Update");
            Console.WriteLine("4. Delete");
            Console.WriteLine("5. Exit");
            Console.Write("Enter your choice: ");
            int choice = int.Parse(Console.ReadLine());


            switch (choice)
```

```
        {
            case 1:
                CreateProduct();
                break;
            case 2:
                ReadProducts();
                break;
            case 3:
                UpdateProduct();
                break;
            case 4:
                DeleteProduct();
                break;
            case 5:
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine("Invalid choice!");
                break;
        }
    }
}

static void CreateProduct()
{
    Console.Write("Enter product name: ");
    string name = Console.ReadLine();
```

```
        Console.Write("Enter product price: ");
        decimal price = decimal.Parse(Console.ReadLine());
        Console.Write("Enter product quantity: ");
        int quantity = int.Parse(Console.ReadLine());

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "INSERT INTO Products (Name, Price, Quantity) VALUES (@Name,
@Price, @Quantity)";
            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.AddWithValue("@Name", name);
            command.Parameters.AddWithValue("@Price", price);
            command.Parameters.AddWithValue("@Quantity", quantity);

            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected > 0)
                Console.WriteLine("Product created successfully.");
            else
                Console.WriteLine("Failed to create product.");
        }
    }

    static void ReadProducts()
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
```

```
            string query = "SELECT * FROM Products";
            SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
            DataSet dataSet = new DataSet();

            adapter.Fill(dataSet, "Products");

            DataTable productsTable = dataSet.Tables["Products"];
            foreach (DataRow row in productsTable.Rows)
            {
                Console.WriteLine($"ID: {row["ID"]}, Name: {row["Name"]}, Price: {row["Price"]}, Quantity: {row["Quantity"]}");
            }
        }
    }

    static void UpdateProduct()
    {
        Console.Write("Enter product ID to update: ");
        int id = int.Parse(Console.ReadLine());
        Console.Write("Enter new name: ");
        string name = Console.ReadLine();
        Console.Write("Enter new price: ");
        decimal price = decimal.Parse(Console.ReadLine());
        Console.Write("Enter new quantity: ");
        int quantity = int.Parse(Console.ReadLine());

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

```
        {
            string query = "UPDATE Products SET Name = @Name, Price = @Price, Quantity = @Quantity WHERE ID = @ID";
            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.AddWithValue("@Name", name);
            command.Parameters.AddWithValue("@Price", price);
            command.Parameters.AddWithValue("@Quantity", quantity);
            command.Parameters.AddWithValue("@ID", id);

            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected > 0)
                Console.WriteLine("Product updated successfully.");
            else
                Console.WriteLine("Failed to update product.");
        }
    }

    static void DeleteProduct()
    {
        Console.Write("Enter product ID to delete: ");
        int id = int.Parse(Console.ReadLine());

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "DELETE FROM Products WHERE ID = @ID";
            SqlCommand command = new SqlCommand(query, connection);
```

```
            command.Parameters.AddWithValue("@ID", id);

            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();
            if (rowsAffected > 0)
                Console.WriteLine("Product deleted successfully.");
            else
                Console.WriteLine("Failed to delete product.");
        }
    }
  }
}
```