TASK 2

APPLY THE CRUD OPERATIONS FOR THE DOCTOR PRESCRIPTIONS USING SQL SERVER.USE ARRAY LIST

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProgramingFundamentalsProject
{
    internal class Program
    {
        static void Main(string[] args)
        {

            DoctorPrescriptionMenu.Menu();
            Console.ReadKey();
        }
    }

}
```

DoctorPrescription.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProgramingFundamentalsProject
{
    internal class DoctorPrescription
    {
        public int doctorid;
        public string patientname;
        public string medication;
        public double dosage;

        public DoctorPrescription(int _doctorid, string _patientname, string
_medication, double _dosage)
        {
            doctorid = _doctorid;
            patientname = _patientname;
            medication = _medication;
            dosage = _dosage;
        }

        public override string ToString()
        {
            return $"[
doctorid={doctorid},patientname={patientname},medication={medication},
dosage={dosage}]";
        }
    }
}
```

DoctorPrescriptionDAO.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProgramingFundamentalsProject
{
    internal class DoctorPrescriptionDAO
    {
        private string connectionString = "Data 
Source=(localdb)\\MSSQLLocalDB;Initial Catalog=Week4AssessmentDb1;Integrated 
Security=True;";
        public void Create(DoctorPrescription doctorPres)
        {
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "INSERT INTO DoctorPrescription (DoctorID,PatientN
ame,Medication,Dosage) VALUES (@DoctorID,@PatientName,@Medication,@Dosage)";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@DoctorID", doctorPres.doctorid);
                cmd.Parameters.AddWithValue("@PatientName", 
doctorPres.patientname);
                cmd.Parameters.AddWithValue("@Medication", 
doctorPres.medication);
                cmd.Parameters.AddWithValue("@Dosage", doctorPres.dosage);

                conn.Open();
                cmd.ExecuteNonQuery();
            }
        }

        // Read a DoctorPrscription by ID
        public DoctorPrescription Read(int DoctorID)
        {
            DoctorPrescription doctorPres = null;
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "SELECT DoctorID,PatientName,Medication,Dosage 
FROM DoctorPrescription WHERE DoctorID = @DoctorID";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@DoctorID", DoctorID);

                conn.Open();
                SqlDataReader reader = cmd.ExecuteReader();
                if (reader.Read())
                {
                    doctorPres = new DoctorPrescription((int)reader["DoctorID"], 
reader["PatientName"].ToString(), reader["Medication"].ToString(), 
                         (double)reader["Dosage"]);
                }
            }
            return doctorPres;
        }
        // Update a DoctorPrescription
        public void Update(DoctorPrescription doctorPres)
        {
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "UPDATE DoctorPrescription SET DoctorID =
@DoctorID, PatientName = @PatientName, Medication = @Medication,  Dosage =
@Dosage WHERE DoctorID= @DoctorID";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@DoctorID", doctorPres.doctorid);
                cmd.Parameters.AddWithValue("@PatientName",
doctorPres.patientname);
                cmd.Parameters.AddWithValue("@Medication",
doctorPres.medication);
                cmd.Parameters.AddWithValue("@Dosage", doctorPres.dosage);

                conn.Open();
                cmd.ExecuteNonQuery();
            }
        }

        // Delete a DoctorPrescription by ID
        public void Delete(int DoctorID)
        {
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "DELETE FROM DoctorPrescription WHERE DoctorID =
@DoctorID";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@DoctorID", DoctorID);

                conn.Open();
                cmd.ExecuteNonQuery();
            }
        }

        // List all Trainers
        public List<DoctorPrescription> ListAll()
        {
            List<DoctorPrescription> doctorPress = new
List<DoctorPrescription>();
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "SELECT DoctorID, PatientName, Medication, Dosage
FROM DoctorPrescription";
                SqlCommand cmd = new SqlCommand(query, conn);

                conn.Open();
                SqlDataReader reader = cmd.ExecuteReader();
                while (reader.Read())
                {
                    DoctorPrescription doctorPres = new
DoctorPrescription((int)reader["DoctorID"], reader["PatientName"].ToString(),
reader["Medication"].ToString(),
                         (double)reader["Dosage"]);
                    doctorPress.Add(doctorPres);
                }
            }
            return doctorPress;
        }
    }

}
```

DoctorPrescriptionMenu.cs

```
namespace ProgramingFundamentalsProject
{
    internal class DoctorPrescriptionMenu
    {
        public static void Menu()
        {
            DoctorPrescriptionUI ui = new DoctorPrescriptionUI();
            bool running = true;

            while (running)
            {
                Console.WriteLine("\nDoctorPrescription Management System");
                Console.WriteLine("1. Create DoctorPrescription");
                Console.WriteLine("2. Read DoctorPrescription");
                Console.WriteLine("3. Update DoctorPrescription");
                Console.WriteLine("4. Delete DoctorPrescription");
                Console.WriteLine("5. List All DoctorPrescription");
                Console.WriteLine("6. Exit");

                Console.Write("Choose an option: ");
                string choice = Console.ReadLine();

                switch (choice)
                {
                    case "1":
                        ui.CreateDoctorPrescription();
                        break;
                    case "2":
                        ui.ReadDoctorPrescription();
                        break;
                    case "3":
                        ui.UpdateDoctorPrescription();
                        break;
                    case "4":
                        ui.DeleteDoctorPrescription();
                        break;
                    case "5":
                        ui.ListAllDoctorPrescription();
                        break;
                    case "6":
                        running = false;
                        Console.WriteLine("Exiting...");
                        break;
                    default:
                        Console.WriteLine("Invalid choice. Please try again.");
                        break;
                }
            }
        }
    }

}
```

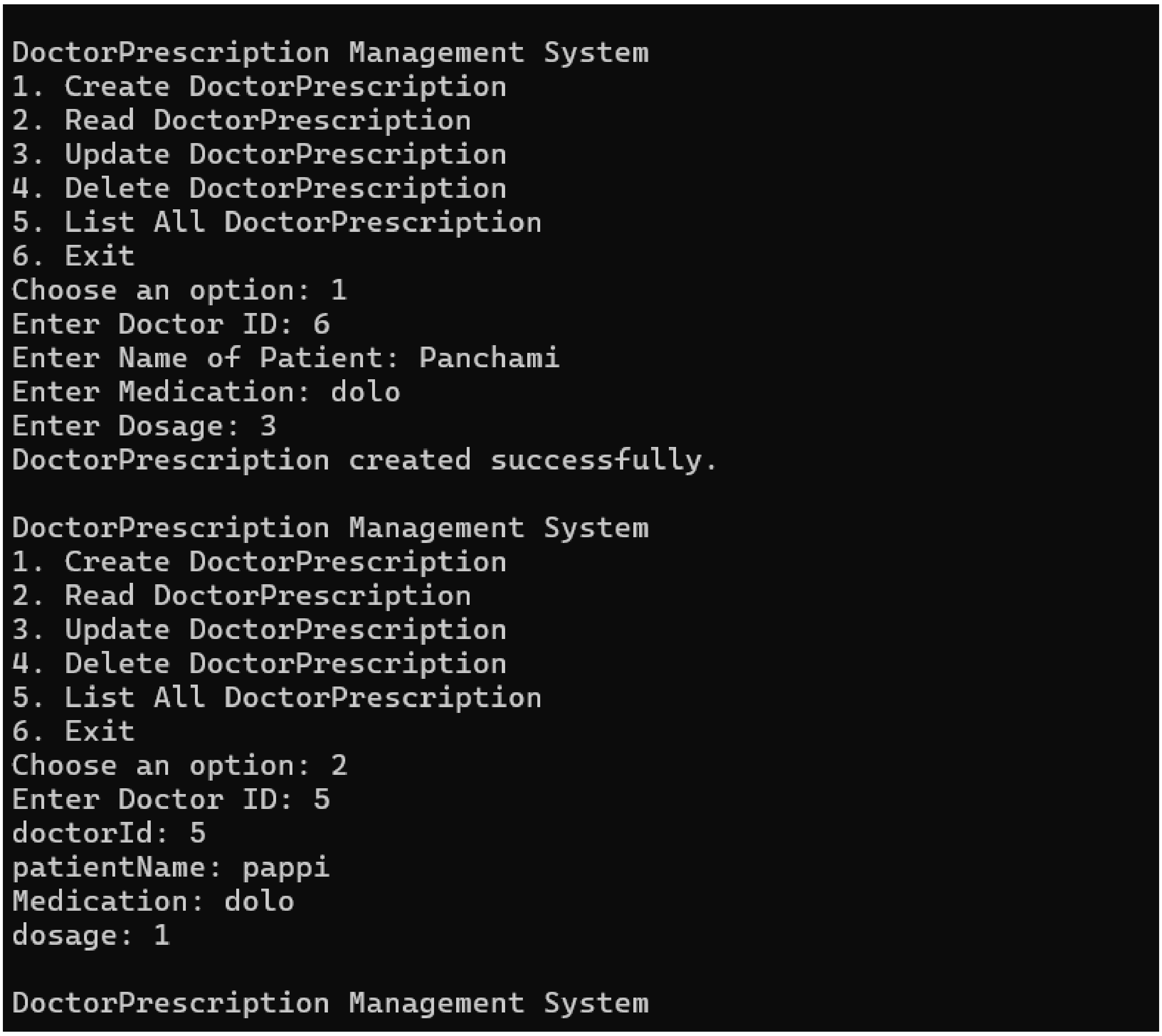
DoctorPrescriptionUI.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

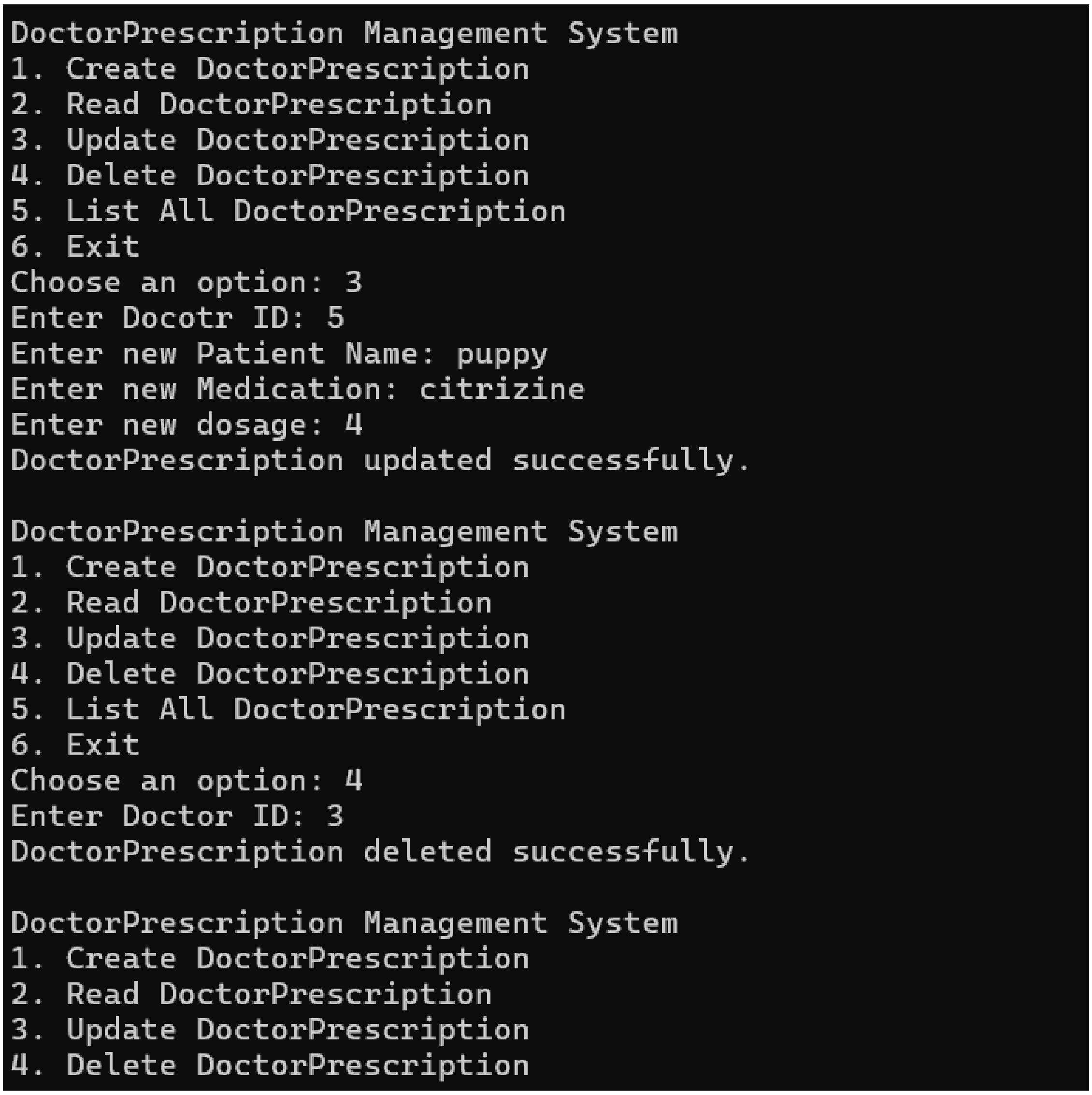
```
namespace ProgramingFundamentalsProject
{
    internal class DoctorPrescriptionUI
    {
        private DoctorPrescriptionDAO doctorPresDAO = new
DoctorPrescriptionDAO();

        public void CreateDoctorPrescription()
        {
            Console.Write("Enter Doctor ID: ");
            int doctorId = int.Parse(Console.ReadLine());
            Console.Write("Enter Name of Patient: ");
            string patientName = Console.ReadLine();
            Console.Write("Enter Medication: ");
            string Medication = Console.ReadLine();
            Console.Write("Enter Dosage: ");
            int dosage = int.Parse(Console.ReadLine());
            DoctorPrescription doctorPres = new DoctorPrescription( doctorId,
patientName, Medication, dosage);

            doctorPresDAO.Create(doctorPres);
            Console.WriteLine("DoctorPrescription created successfully.");
        }

        public void ReadDoctorPrescription()
        {
            Console.Write("Enter Doctor ID: ");
            int doctorId = int.Parse(Console.ReadLine());

            DoctorPrescription doctorPres = doctorPresDAO.Read(doctorId);
            if (doctorPres != null)
            {
                Console.WriteLine($"doctorId: {doctorPres.doctorid}");
                Console.WriteLine($"patientName: {doctorPres.patientname}");
                Console.WriteLine($"Medication: {doctorPres.medication}");
                Console.WriteLine($"dosage: {doctorPres.dosage}");
            }
            else
            {
                Console.WriteLine(" DoctorPrescription not found.");
            }
        }

        public void UpdateDoctorPrescription()
        {
            Console.Write("Enter Docotr ID: ");
            int doctorId = int.Parse(Console.ReadLine());

            DoctorPrescription doctorPres = doctorPresDAO.Read(doctorId);
            if (doctorPres != null)
            {
                Console.Write("Enter new Patient Name: ");
                doctorPres.patientname = Console.ReadLine();
                Console.Write("Enter new Medication: ");
                doctorPres.medication = Console.ReadLine();
                Console.Write("Enter new dosage: ");
                doctorPres.dosage = double.Parse(Console.ReadLine());

                doctorPresDAO.Update(doctorPres);
                Console.WriteLine("DoctorPrescription updated successfully.");
            }
            else
            {
                Console.WriteLine("DoctorPrescription not found.");
            }
        }

        public void DeleteDoctorPrescription()
        {
            Console.Write("Enter Doctor ID: ");
            int doctorid = int.Parse(Console.ReadLine());

            doctorPresDAO.Delete(doctorid);
            Console.WriteLine("DoctorPrescription deleted successfully.");
        }

        public void ListAllDoctorPrescription()
        {
            List<DoctorPrescription> doctorPress = doctorPresDAO.ListAll();
            foreach (DoctorPrescription doctorPres in doctorPress)
            {
                Console.WriteLine($"ID: {doctorPres.doctorid}, PatientName:
{doctorPres.patientname}, Medication: {doctorPres.medication}, Dosage:
{doctorPres.dosage}");
            }
        }
    }

}
```

Week4AssessmentDb

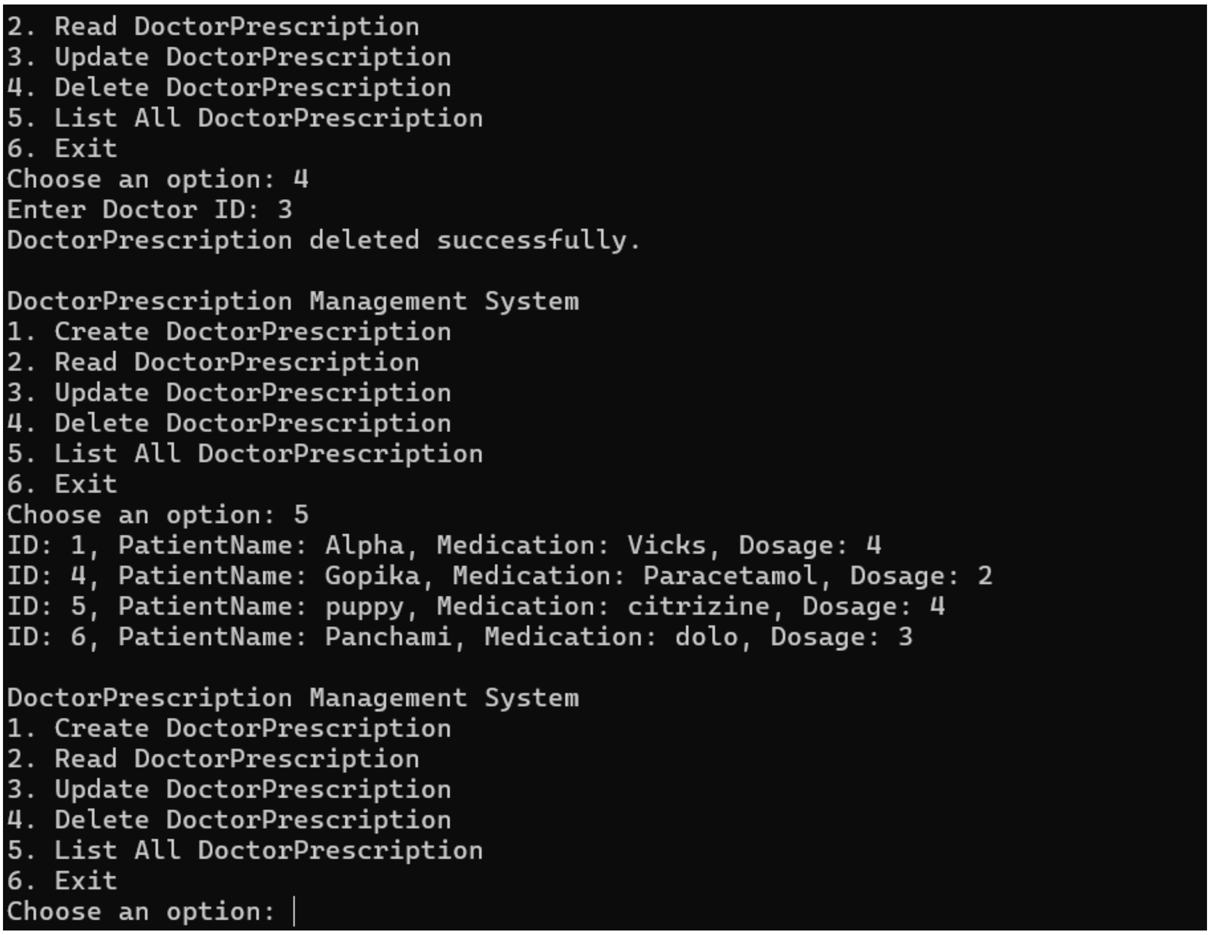
```
CREATE DATABASE Week4AssessmentDb;
USE Week4AssessmentDb;
CREATE TABLE DoctorPrescription( DoctorID INT PRIMARY KEY, PatientName VARCHAR(225),
Medication NVARCHAR(100), Dosage FLOAT );
INSERT INTO DoctorPrescription(DoctorID,PatientName,Medication,Dosage)VALUES (1,'Fidha','Dolo
650',2),(2,'Sarika','Para',1),(3,'Athuliya','Vicks',3);
```



```
DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 1
Enter Doctor ID: 6
Enter Name of Patient: Panchami
Enter Medication: dolo
Enter Dosage: 3
DoctorPrescription created successfully.

DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 2
Enter Doctor ID: 5
doctorId: 5
patientName: pappi
Medication: dolo
dosage: 1

DoctorPrescription Management System
```



```
DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 3
Enter Docotr ID: 5
Enter new Patient Name: puppy
Enter new Medication: citrizine
Enter new dosage: 4
DoctorPrescription updated successfully.

DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 4
Enter Doctor ID: 3
DoctorPrescription deleted successfully.

DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
```



```
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 4
Enter Doctor ID: 3
DoctorPrescription deleted successfully.

DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 5
ID: 1, PatientName: Alpha, Medication: Vicks, Dosage: 4
ID: 4, PatientName: Gopika, Medication: Paracetamol, Dosage: 2
ID: 5, PatientName: puppy, Medication: citrizine, Dosage: 4
ID: 6, PatientName: Panchami, Medication: dolo, Dosage: 3

DoctorPrescription Management System
1. Create DoctorPrescription
2. Read DoctorPrescription
3. Update DoctorPrescription
4. Delete DoctorPrescription
5. List All DoctorPrescription
6. Exit
Choose an option: 
```