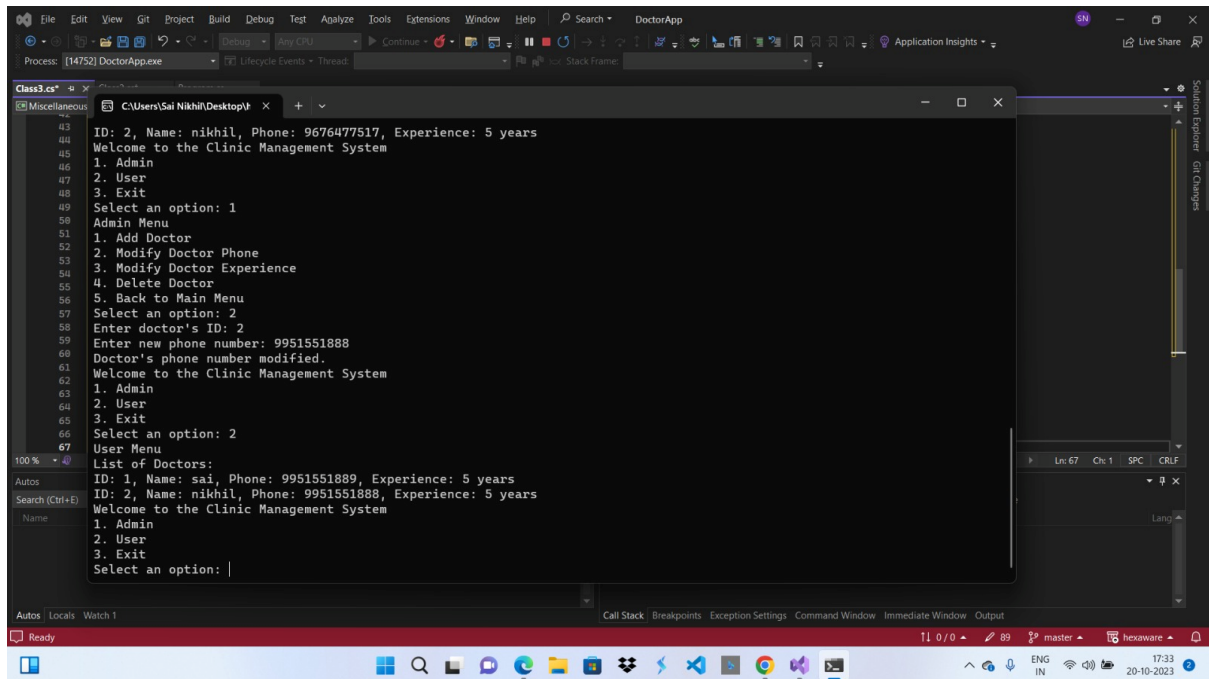


OUTPUT:



```
43 ID: 2, Name: nikhil, Phone: 9676477517, Experience: 5 years
44 Welcome to the Clinic Management System
45 1. Admin
46 2. User
47 3. Exit
48 Select an option: 1
49 Admin Menu
50 1. Add Doctor
51 2. Modify Doctor Phone
52 3. Modify Doctor Experience
53 4. Delete Doctor
54 5. Back to Main Menu
55 Select an option: 2
56 Enter doctor's ID: 2
57 Enter new phone number: 9951551888
58 Doctor's phone number modified.
59 Welcome to the Clinic Management System
60 1. Admin
61 2. User
62 3. Exit
63 Select an option: 2
64 User Menu
65 List of Doctors:
66 ID: 1, Name: sai, Phone: 9951551889, Experience: 5 years
67 ID: 2, Name: nikhil, Phone: 9951551888, Experience: 5 years
68 Welcome to the Clinic Management System
69 1. Admin
70 2. User
71 3. Exit
72 Select an option: |
```

Doctor.CS

namespace ClinicManagement

```
{
    public class Doctor
    {
        public int DoctorId { get; set; }
        public string Name { get; set; }
        public string PhoneNumber { get; set; }
        public int YearsOfExperience { get; set; }
    }
}
```

Program.CS

using System;

namespace ClinicManagement

```
{
```

```
class Program
{
    static void Main(string[] args)
    {
        ClinicManagementApp.Run();
    }
}
```

Clinic.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
namespace ClinicManagement
{
    public partial class ClinicManagementApp
    {
        static List<Doctor> doctorList = new List<Doctor>();
        static int doctorIdCounter = 1;

        public static void Run()
        {
            while (true)
            {
                Console.WriteLine("Welcome to the Clinic Management System");
                Console.WriteLine("1. Admin");
                Console.WriteLine("2. User");
                Console.WriteLine("3. Exit");
                Console.Write("Select an option: ");
            }
        }
    }
}
```

```
if (int.TryParse(Console.ReadLine(), out int choice))
{
    switch (choice)
    {
        case 1:
            AdminMenu();
            break;
        case 2:
            UserMenu();
            break;
        case 3:
            Console.WriteLine("Exiting the application.");
            return;
        default:
            Console.WriteLine("Invalid choice. Please select a valid option.");
            break;
    }
}
else
{
    Console.WriteLine("Invalid input. Please enter a number.");
}
}
```

```
public static void AdminMenu()
{
    Console.WriteLine("Admin Menu");
    Console.WriteLine("1. Add Doctor");
    Console.WriteLine("2. Modify Doctor Phone");
    Console.WriteLine("3. Modify Doctor Experience");
}
```

```
Console.WriteLine("4. Delete Doctor");

Console.WriteLine("5. Back to Main Menu");

Console.Write("Select an option: ");

if (int.TryParse(Console.ReadLine(), out int choice))
{
    switch (choice)
    {
        case 1:
            AddDoctor();
            break;
        case 2:
            ModifyDoctorPhone();
            break;
        case 3:
            ModifyDoctorExperience();
            break;
        case 4:
            DeleteDoctor();
            break;
        case 5:
            return;
        default:
            Console.WriteLine("Invalid choice. Please select a valid option.");
            break;
    }
}

else
{
    Console.WriteLine("Invalid input. Please enter a number.");
}
```

```
}
```

```
public static void AddDoctor()
```

```
{
```

```
    Doctor doctor = new Doctor
```

```
    {
```

```
        DoctorId = doctorIdCounter++,
```

```
        Name = ReadString("Enter doctor's name: "),
```

```
        PhoneNumber = ReadString("Enter doctor's phone: "),
```

```
        YearsOfExperience = ReadInt("Enter doctor's experience (in years): ")
```

```
    };
```

```
    doctorList.Add(doctor);
```

```
    Console.WriteLine("Doctor added successfully.");
```

```
}
```

```
public static void ModifyDoctorPhone()
```

```
{
```

```
    int doctorId = ReadInt("Enter doctor's ID: ");
```

```
    Doctor doctor = doctorList.FirstOrDefault(d => d.DoctorId == doctorId);
```

```
    if (doctor != null)
```

```
    {
```

```
        doctor.PhoneNumber = ReadString("Enter new phone number: ");
```

```
        Console.WriteLine("Doctor's phone number modified.");
```

```
    }
```

```
    else
```

```
    {
```

```
        Console.WriteLine("Doctor not found with the given ID.");
```

```
    }
```

```
}
```

```
public static void ModifyDoctorExperience()
```

```

{
    int doctorId = ReadInt("Enter doctor's ID: ");

    Doctor doctor = doctorList.FirstOrDefault(d => d.DoctorId == doctorId);

    if (doctor != null)
    {
        doctor.YearsOfExperience = ReadInt("Enter new experience (in years): ");

        Console.WriteLine("Doctor's experience modified.");
    }
    else
    {
        Console.WriteLine("Doctor not found with the given ID.");
    }
}

```

```

public static void DeleteDoctor()
{
    int doctorId = ReadInt("Enter doctor's ID: ");

    Doctor doctor = doctorList.FirstOrDefault(d => d.DoctorId == doctorId);

    if (doctor != null)
    {
        doctorList.Remove(doctor);

        Console.WriteLine("Doctor deleted.");
    }
    else
    {
        Console.WriteLine("Doctor not found with the given ID.");
    }
}

```

```

public static void UserMenu()
{

```

```
    Console.WriteLine("User Menu");

    Console.WriteLine("List of Doctors:");

    foreach (var doctor in doctorList)
    {
        Console.WriteLine($"ID: {doctor.DoctorId}, Name: {doctor.Name}, Phone:
{doctor.PhoneNumber}, Experience: {doctor.YearsOfExperience} years");
    }
}
```

```
public static string ReadString(string prompt)
{
    Console.Write(prompt);

    return Console.ReadLine();
}
```

```
public static int ReadInt(string prompt)
{
    int result;

    do
    {
        Console.Write(prompt);
    } while (!int.TryParse(Console.ReadLine(), out result));

    return result;
}
}
```