1) Add User Details

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Xml;

using System.IO;


  class DBConnection

  {

    public static SqlConnection GetConnection()

    {

      String xmlString = System.IO.File.ReadAllText("mssql.xml");


      string username;

      string password;

      string schema;

      string host;


      using (XmlReader reader = XmlReader.Create(new StringReader(xmlString)))

      {

        reader.ReadToFollowing("username");

        username = reader.ReadElementContentAsString();

        reader.ReadToFollowing("password");

        password = reader.ReadElementContentAsString();
```

```
            reader.ReadToFollowing("host");

            host = reader.ReadElementContentAsString();

            reader.ReadToFollowing("schema");

            schema = reader.ReadElementContentAsString();

          }


          string connection_string = "Data Source=" + host + ";Initial Catalog=" + schema + ";User id=" + 
username + ";Password=" + password + ";";

          //Console.WriteLine(connection_string);

          SqlConnection conn = new SqlConnection(connection_string);

          return conn;


        }

    }


using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;


namespace ADO.NETP1EBOX

{

    class Program

    {

        static void Main(string[] args)
        {
            UserBO userBO = new UserBO();
            Console.WriteLine("Enter total number of users");
            int n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter user details");
            for (int i = 0; i < n; i++)
            {
                string details = Console.ReadLine();
                string[] str = details.Split(',');
                string name = str[0];
                string username = str[1];
                string password = str[2];
                string contactNo = str[3];
                User user = new User(name, username, password, contactNo);
                int j = userBO.InsertUser(user);
                if (j > 0)
                    Console.WriteLine("Record Inserted Successfully");
            }
            List<User> users = userBO.GetUserList();

            Console.WriteLine("{0,-15}{1,-15}{2,-15}{3,-15}", "Name", "Username", "Password", "ContactNo");
            foreach(User u1 in users)
            {
                Console.WriteLine("{0,-15}{1,-15}{2,-15}{3,-15}", u1.Name, u1.UserName, u1.Password, u1.ContactNo);
            }
            Console.ReadLine();
        }
    }
}

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

namespace ADO.NETP1EBOX
{
    class UserBO
    {
        public List<User> GetUserList()
        {
            UserDAO userDAO = new UserDAO();
            List<User> users = userDAO.GetUserList();
            return users;
        }
        public int InsertUser(User e)
        {
            UserDAO userDAO = new UserDAO();
            int i = userDAO.InsertUser(e);
            return i;
        }
    }
}

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

namespace ADO.NETP1EBOX
{
    class UserDAO
    {
        public int InsertUser(User user)
        {
            SqlConnection con = DBConnection.GetConnection();
            con.Open();


            string insert = "insert into person values(@name,@username,@password,@contactNo)";

            SqlCommand cmd = new SqlCommand(insert, con);
            cmd.Parameters.AddWithValue("@name", user.Name);
            cmd.Parameters.AddWithValue("@username", user.UserName);
            cmd.Parameters.AddWithValue("@password", user.Password);

            cmd.Parameters.AddWithValue("@contactNo", user.ContactNo);

            int i = cmd.ExecuteNonQuery();

            con.Close();

            return i;

        }

        public List<User> GetUserList()

        {

            List<User> users = new List<User>();

            SqlConnection con = DBConnection.GetConnection();

            con.Open();

            string data = "select * from person";

            SqlCommand cmd = new SqlCommand(data, con);

            SqlDataReader dr = cmd.ExecuteReader();

            while (dr.Read())

            {

                User user = new User(dr[0].ToString(), dr[1].ToString(), dr[2].ToString(), dr[3].ToString());

                users.Add(user);

            }

            return users;

        }

    }

}


    class User

    {

        private string _name;
private string _username;

private string _password;

private string _contactNo;

public User()

{


}

public User(string _name, string _username, string _password, string _contactNo)

{

   this._name = _name;

   this._username = _username;

   this._password = _password;

   this._contactNo = _contactNo;

}

public string Name

{

   get

   {

      return _name;

   }

   set

   {

      _name = value;

   }

}

public string UserName

{
    get
    {
        return _username;
    }
    set
    {
        _username = value;
    }
}
public string Password
{
    get
    {
        return _password;
    }
    set
    {
        _password = value;
    }
}
public string ContactNo
{
    get
    {
        return _contactNo;
    }
    set
```

```
        {
            _contactNo = value;
        }
    }
}
```

2) Update Person

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;



class Program

{

    static void Main(string[] args)

    {

        PersonBO personBO = new PersonBO();

        Console.WriteLine("Enter the name");

        string Name = Console.ReadLine();

        Console.WriteLine("Enter the mobile number");

        string Mobilenumber = Console.ReadLine();

        bool j = personBO.UpdatePerson(Name,Mobilenumber);

        if (j == true)

        {

            Console.WriteLine("Update successfully");

        }

        else

        {

            Console.WriteLine("Person not found");
        }
        Console.ReadLine();
    }
}

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

class Person
{
    private string _name;
    private string _mobileNumber;
    private string _username;
    private string _password;

    public string MobileNumber
    {
        get { return this._mobileNumber; }
        set { this._mobileNumber = value; }
    }
    public string Name
    {
        get { return this._name; }
        set { this._name = value; }
    }
    public string Username
    {
        get { return this._username; }
        set { this._username = value; }
    }
    public string Password
    {
        get { return this._password; }
        set { this._password = value; }
    }


    public Person(string _name, string _mobileNumber, string _username, string _password)
    {
        this._name = _name;
        this._mobileNumber = _mobileNumber;
        this._username = _username;
        this._password = _password;
    }
}

using System;
using System.Data.SqlClient;
using System.Xml;
using System.IO;


class DBConnection

{

   public static SqlConnection GetConnection()

   {

      String xmlString = System.IO.File.ReadAllText("mssql.xml");


      string username;

      string password;

      string schema;

      string host;


      using (XmlReader reader = XmlReader.Create(new StringReader(xmlString)))

      {

         reader.ReadToFollowing("username");

         username = reader.ReadElementContentAsString();

         reader.ReadToFollowing("password");

         password = reader.ReadElementContentAsString();

         reader.ReadToFollowing("host");

         host = reader.ReadElementContentAsString();

         reader.ReadToFollowing("schema");

         schema = reader.ReadElementContentAsString();

      }



      string connection_string = "Data Source=" + host + ";Initial Catalog=" + schema + ";User id=" +
username + ";Password=" + password + ";";
```

```
        //Console.WriteLine(connection_string);

        SqlConnection conn = new SqlConnection(connection_string);

        return conn;

    }

}


using System;

using System.Collections.Generic;

using System.Data.SqlClient;


class PersonDAO

{

    public bool UpdatePerson(string _name, string _mobileNumber)

    {

        SqlConnection con =  DBConnection.GetConnection();

        con.Open();

        string str = "update person SET mobile_number=@_mobileNumber where name=@_name";

        SqlCommand cmd = new SqlCommand(str, con);

        cmd.Parameters.AddWithValue("@_name", _name);

        cmd.Parameters.AddWithValue("@_mobileNumber", _mobileNumber);

        int i=cmd.ExecuteNonQuery();

        con.Close();

        if(i>0)

        {

            return true;

        }

        else
        {
        return false;
        }
    }
}

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

class PersonBO
{
    public bool UpdatePerson(string _name, string _mobileNumber)
    {
        PersonDAO personDAO = new PersonDAO();
        bool i = personDAO.UpdatePerson(_name, _mobileNumber);
        return i;
    }
}
```

3) Search and Delete Item Type

```
using System;

using System.Collections.Generic;

using System.Text;

using System.IO;

using System.Xml;

using System.Data.SqlClient;


class DBConnection

{

    public static SqlConnection GetConnection()

    {

        String xmlString = System.IO.File.ReadAllText("mssql.xml");

        string username;

        string password;

        string schema;

        string host;

        using (XmlReader reader = XmlReader.Create(new StringReader(xmlString)))

        {

            reader.ReadToFollowing("username");

            username = reader.ReadElementContentAsString();

            reader.ReadToFollowing("password");

            password = reader.ReadElementContentAsString();

            reader.ReadToFollowing("host");

            host = reader.ReadElementContentAsString();

            reader.ReadToFollowing("schema");

            schema = reader.ReadElementContentAsString();
        }
        string connection_string = "Data Source=" + host + ";Initial Catalog=" + schema + ";User id=" +
username + ";Password=" + password + ";";
        //Console.WriteLine(connection_string);
        SqlConnection conn = new SqlConnection(connection_string);
        return conn;
    }
}


using System;
using System.Collections.Generic;
using System.Data.SqlClient;


class Program
{
    static void Main(string[] args)
    {
        ItemTypeBO itbo=new ItemTypeBO();
        Console.WriteLine("Menu");
        Console.WriteLine("1.Search item type");
        Console.WriteLine("2.Delete item type");
        int choice = Convert.ToInt32(Console.ReadLine());
        if(choice == 1)
        {
            Console.WriteLine("Enter the item type name");
            string searchName = Console.ReadLine();
            Console.WriteLine("Item type Details");
            ItemType it = itbo.GetItemTypeByName(searchName);
```

```
      Console.WriteLine("Id: "+it.Id);

      Console.WriteLine("Name: "+it.Name);

      Console.WriteLine("Deposit: "+it.Deposit);

      Console.WriteLine("Cost per day: "+it.CostPerDay);

    }

    else if(choice == 2)

    {

      Console.WriteLine("Enter the item type name");

      string name = Console.ReadLine();

      bool flag;

      flag = itbo.DeleteItemType(name);

      if (flag)

      {

        Console.WriteLine("Deleted successfully");

        Console.WriteLine("Item type details");

        Console.WriteLine(String.Format("{0,-5}{1,-15} {2,-15} {3}", "Id", "Name", "Deposit",
"CostPerDay"));

        List<ItemType> item = itbo.GetAllItemType();

        foreach (ItemType u in item)

        {

          Console.WriteLine("{0,-5}{1,-15} {2,-15} {3}",u.Id,u.Name,u.Deposit,u.CostPerDay);

        }

      }

      else

        Console.WriteLine("Invalid input");

    }

    else

    {
            Console.WriteLine("Invalid input");
        }
    }
}

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Text;

class ItemTypeDAO
{
    public List<ItemType> GetAllItemType()
    {
        List<ItemType> list = new List<ItemType>();
        SqlConnection con = DBConnection.GetConnection();
        con.Open();
        string str = "select * from item_type";
        SqlCommand cmd = new SqlCommand(str, con);
        SqlDataReader sdr = cmd.ExecuteReader();
        while (sdr.Read())
        {
            ItemType it = new ItemType(Convert.ToInt32(sdr[0]), sdr[1].ToString(), Convert.ToDouble(sdr[2]), Convert.ToDouble(sdr[3]));
            list.Add(it);
        }
        con.Close();
        return list;
```

```
}
public bool DeleteItemType(string name)
{
    SqlConnection con = DBConnection.GetConnection();
    con.Open();
    string str = "delete from item_type where name='" + name + "'";
    SqlCommand cmd = new SqlCommand(str, con);
    int r = cmd.ExecuteNonQuery();
    if (r > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
    con.Close();
}
public ItemType GetItemTypeByName(string name)
{
    SqlConnection con = DBConnection.GetConnection();
    con.Open();
    string str = "select * from item_type where name='" + name + "'";
    SqlCommand cmd = new SqlCommand(str, con);
    SqlDataReader sdr = cmd.ExecuteReader();
    if (sdr.HasRows)
    {
            sdr.Read();
            ItemType it = new ItemType(Convert.ToInt32(sdr[0]), sdr[1].ToString(),
Convert.ToDouble(sdr[2]), Convert.ToDouble(sdr[3]));
            return it;
        }
        else
        {
            return null;
        }
        con.Close();
    }
}

using System;
using System.Collections.Generic;
using System.Data.SqlClient;

class ItemTypeBO
{
    public List<ItemType> GetAllItemType()
    {
        return new ItemTypeDAO().GetAllItemType();
    }
    public bool DeleteItemType(string name)
    {
        return new ItemTypeDAO().DeleteItemType(name);
    }
    public ItemType GetItemTypeByName(string name)
    {
        return new ItemTypeDAO().GetItemTypeByName(name);
    }
}


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



class ItemType
{
    private int _id;
    private string _name;
    private double _deposit;
    private double _costPerDay;
    public ItemType(int _id, string _name, double _deposit, double _costPerDay)
    {
        this._id = _id;
        this._name = _name;
        this._deposit = _deposit;
        this._costPerDay = _costPerDay;
    }
```

```
public int Id
{
   get
   {
      return _id;
   }
   set
   {
      _id = value;
   }
}
public string Name
{
   get
   {
      return _name;
   }
   set
   {
      _name = value;
   }
}
public double Deposit
{
   get
   {
      return _deposit;
```

```
        }
        set
        {
            _deposit = value;
        }
    }
    public double CostPerDay
    {
        get
        {
            return _costPerDay;
        }
        set
        {
            _costPerDay = value;
        }
    }
}
```

4) Event Details – Stored Procedure

```
using System.Data;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

class EventDAO

{

   public List<Event> GetAllEvents()

   {

      List<Event> events = new List<Event>();

      SqlConnection con = DBConnection.GetConnection();

      con.Open();

      SqlCommand cmd = new SqlCommand("stpGetAllEvents", con);

      cmd.CommandType = CommandType.StoredProcedure;

      SqlDataReader dr = cmd.ExecuteReader();

      while (dr.Read())

      {

         Event e = new Event(dr[0].ToString(),dr[1].ToString(),dr[2].ToString());

         events.Add(e);

      }

      con.Close();

      return events;

   }

}
```

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
class Program
{
    static void Main(string[] args)
    {
        List<Event> events = new List<Event>();
        Console.WriteLine("Event details");
        Console.WriteLine(String.Format("{0,-25} {1,-15} {2}", "Name", "Type", "Organizer"));
        EventBO eb = new EventBO();
        events = eb.GetAllEvents();
        foreach (Event e in events)
        {
            Console.WriteLine(String.Format("{0,-25} {1,-15} {2}", e.Name, e.Type, e.Organizer));
        }
    }
}

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

class Event
{
    private string _name;
    private string _type;

    private string _organizer;


    public string Name

    {

        get { return this._name; }

        set { this._name = value; }

    }


    public string Type

    {

        get { return this._type; }

        set { this._type = value; }

    }


    public string Organizer

    {

        get { return this._organizer; }

        set { this._organizer = value; }

    }


    public Event() { }


    public Event(string _name, string _type, string _organizer)

    {

        this._name = _name;

        this._type = _type;
```

```
        this._organizer = _organizer;
    }
}

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;

class EventBO
{
    public List<Event> GetAllEvents()
    {
        return new EventDAO().GetAllEvents();
    }
}



using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Xml;
using System.Data.SqlClient;

class DBConnection
```

```
{
    public static SqlConnection GetConnection()
    {
        String xmlString = System.IO.File.ReadAllText("mssql.xml");
        string username;
        string password;
        string schema;
        string host;
        using (XmlReader reader = XmlReader.Create(new StringReader(xmlString)))
        {
            reader.ReadToFollowing("username");
            username = reader.ReadElementContentAsString();
            reader.ReadToFollowing("password");
            password = reader.ReadElementContentAsString();
            reader.ReadToFollowing("host");
            host = reader.ReadElementContentAsString();
            reader.ReadToFollowing("schema");
            schema = reader.ReadElementContentAsString();
        }
        string connection_string = "Data Source=" + host + ";Initial Catalog=" + schema + ";User id=" + username + ";Password=" + password + ";";
        //Console.WriteLine(connection_string);
        SqlConnection conn = new SqlConnection(connection_string);
        return conn;
    }
}
```