# WCF Assignment Guide

❖ **Introduction** : Developing a Client-Server application with WCF

❖ **Requirement** :

- Visual Studio .Net 2010/ 2012 (.Net Framework 4.0/4.5 )
- In SQL Server 2008 / 2012 , create a database named Manager and create a table named Books as the following :

T400\SQL2K8.Manager - dbo.Books

| Column Name | Data Type | Allow Nulls |
|---|---|---|
| BookID | int | ☐ |
| BookTitle | varchar(50) | ☐ |
| BookPrice | float | ☐ |
| | | ☐ |

➢ **Step 1 : Create WCFService Application**

1.1 From menu File|New WebSite, In Project Type | Visual C# & Templates | WCF Service and named *WCFBookService*

1.2 Write code in App_Code/**IService.cs** file as the following ":

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

[ServiceContract]
public interface IService
{
    [OperationContract]
    List<Book> GetAllBooks();
    [OperationContract]
    void AddBook(Book newBook);
    [OperationContract]
    void UpdateBook(Book updateBook);
    [OperationContract]
    void DeleteBook(Book deleteBook) ;
}
[DataContract]
public class Book{
    [DataMember]
    public int BookID { get; set; }
    [DataMember]
     public string BookTitle { get; set; }
    [DataMember]
    public float BookPrice { get; set; }
}
```

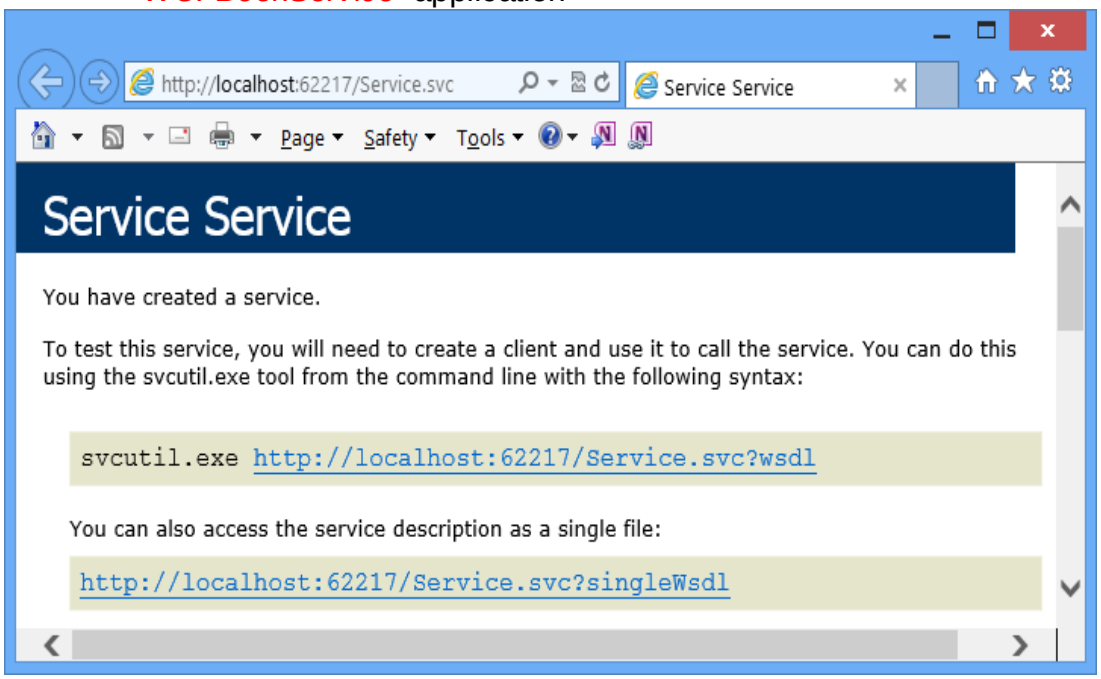
**1.3** Write code in App_Code/**Service.cs** file as the following :

```
using System.Data;
using System.Data.SqlClient;
public class Service : IService
{
    string strConnection =
       "server=.\\SQL2K8;database=Manager;uid=sa;pwd=123";
    public List<Book> GetAllBooks() {
        List<Book> bookList = new List<Book>();
        string SQL = "select * from Books";
        SqlConnection cnn = new SqlConnection(strConnection);
        SqlCommand cmd = new SqlCommand(SQL, cnn);
        cnn.Open();
        SqlDataReader rd = cmd.ExecuteReader
            (CommandBehavior.CloseConnection);
        if (rd.HasRows)  {
            while (rd.Read())
            {
                Book b = new Book() {
                    BookID = int.Parse(rd["BookID"].ToString()),
                    BookTitle = rd["BookTitle"].ToString(),
                    BookPrice = float.Parse(rd["BookPrice"].ToString())
                };
                bookList.Add(b);
            }
        }
        return bookList.OrderByDescending(book => book.BookPrice).ToList();
    }

     public void AddBook(Book newBook)
     {
         SqlConnection cnn = new SqlConnection(strConnection);
         string SQLInsert = "Insert Books values(@ID,@Title,@Price)";
         SqlCommand cmd = new SqlCommand(SQLInsert, cnn);
         cmd.Parameters.AddWithValue("@ID", newBook.BookID);
         cmd.Parameters.AddWithValue("@Title", newBook.BookTitle);
         cmd.Parameters.AddWithValue("@Price", newBook.BookPrice);
         try {
             cnn.Open();
             cmd.ExecuteNonQuery();
         }
         catch{
             throw new Exception("Insert Error");
         }
         finally {
             cnn.Close();
         }
     }
```

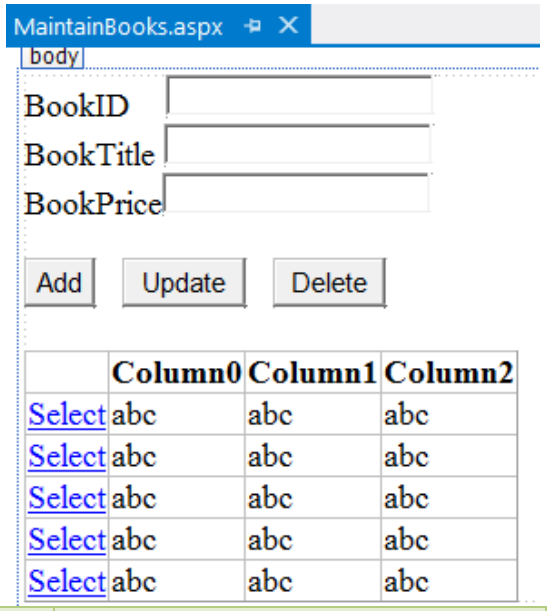
```
    public void UpdateBook(Book updateBook)
    {
        SqlConnection cnn = new SqlConnection(strConnection);
        string SQLUpdate =
            "Update Books set BookTitle = @Title , BookPrice = @Price Where BookID=@ID";
        SqlCommand cmd = new SqlCommand(SQLUpdate, cnn);
        cmd.Parameters.AddWithValue("@ID", updateBook.BookID);
        cmd.Parameters.AddWithValue("@Title", updateBook.BookTitle);
        cmd.Parameters.AddWithValue("@Price", updateBook.BookPrice);
        try {
            cnn.Open();
            cmd.ExecuteNonQuery();
        }
        catch {
            throw new Exception("Update Error");
        }
        finally{
            cnn.Close();
        }
    }
    public void DeleteBook(Book deleteBook)
    {
         SqlConnection cnn = new SqlConnection(strConnection);
         string SQLDelete = "Delete Books where BookID=@ID";
         SqlCommand cmd = new SqlCommand(SQLDelete, cnn);
         cmd.Parameters.AddWithValue("@ID", deleteBook.BookID);
         try
         {
             cnn.Open();
             cmd.ExecuteNonQuery();
         }
         catch
         {
             throw new Exception("Delete Error");
         }
         finally
         {
             cnn.Close();
         }
    }
}
```

1.4 Right-Click on Service.svc file , select *View In Browser* to run *WCFBookService* application




➢ **Step 2 : Create Web Application and Reference to *WCFBookService***

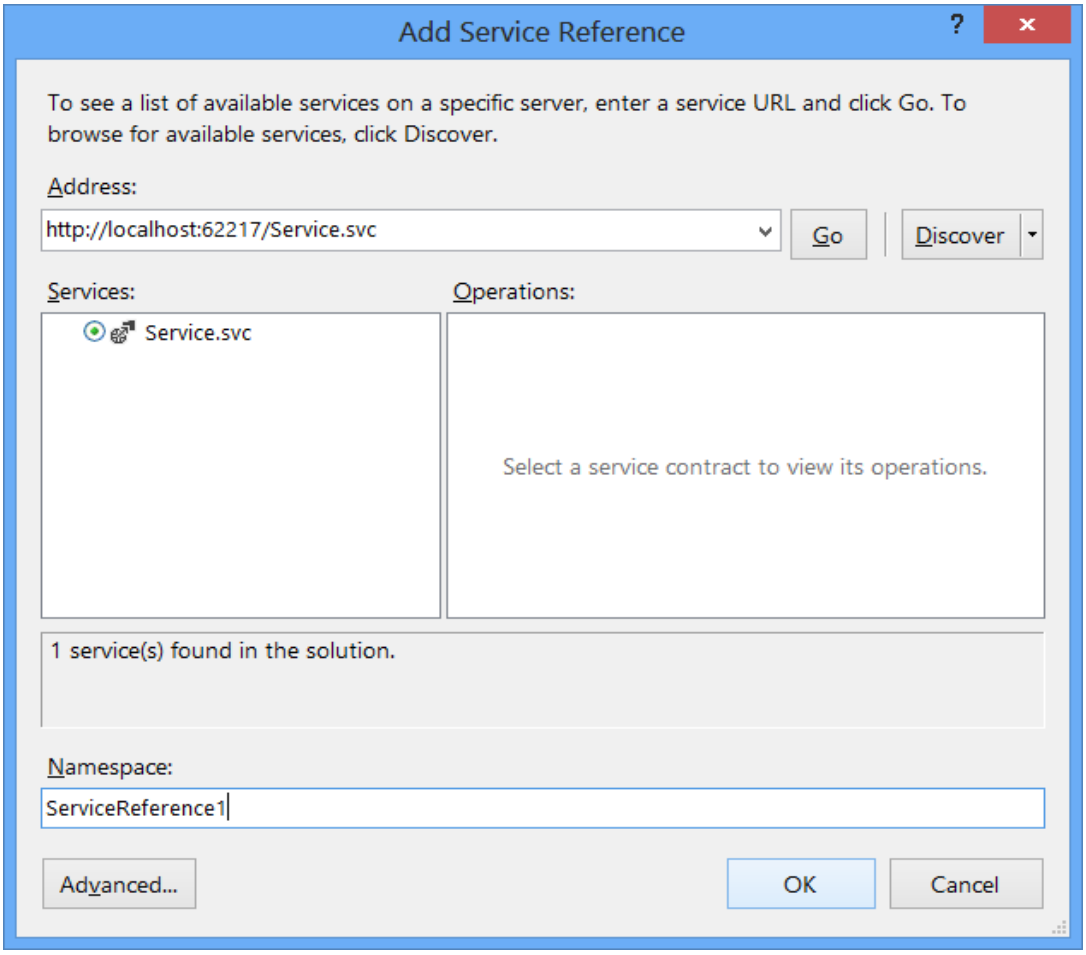
2.1 From menu File|Add New WebSite, In Project Type | Visual C# Templates| **ASP.NET Empty Web Site** , named MyWebApp , and add new a ASP.NET named MaintainBooks.aspx with user interface as the below table:




| Control Type | Properties/ Event | Value |
|---|---|---|
| TextBox | ID | txtBookID |
| TextBox | ID | txtBookTitle |
| TextBox | ID | txtBookPrice |
| Button | ID | btnAdd |
| | Text | Add |
| | Event : Click | btnAdd_Click |
| Button | ID | btnUpdate |

| | | |
|---|---|---|
| | Text | Update |
| | Event: Click | btnUpdate_Click |
| Button | ID | btnDelete |
| | Text | Delete |
| | Event: Click | btnDelete_Click |
| GridView | ID | gvBookList |
| | Event: SelectedIndexChanging | gvBookList_SelectedIndexChanging |
| | AutoGenerateSelectButton | True |

2.2 Right – click on project ,select *Add Service Reference*, dialog showed, click *Discover* button or enter address as the above figure : *http://localhost:62217/Service.svc* , click **Go** to find services and click **OK** button to finish.




2.3 Double –click on the MaintainBooks.aspx page and write code for MaintainBooks.aspx.cs file as the following :

```
using ServiceReference1;
public partial class MaintainBooks : System.Web.UI.Page{
   protected void Page_Load(object sender, EventArgs e) {
       if (!IsPostBack){
           LoadData();
       }
   }
   public void LoadData(){
       ServiceClient BookData = new ServiceClient();
       gvBookList.DataSource = BookData.GetAllBooks();
       gvBookList.DataBind();
   }
   protected void btnAdd_Click(object sender, EventArgs e){
       ServiceClient BookData = new ServiceClient();
       int ID = int.Parse(txtBookID.Text);
       string Title = txtBookTitle.Text;
       float Price = float.Parse(txtBookPrice.Text);
       Book b = new Book { BookID = ID, BookPrice = Price, BookTitle = Title };
       BookData.AddBook(b);
       LoadData();
   }
   protected void btnDelete_Click(object sender, EventArgs e){
       ServiceClient BookData = new ServiceClient();
       int ID = int.Parse(txtBookID.Text);
       Book b = new Book { BookID = ID };
       BookData.DeleteBook(b);
       LoadData();
   }
  protected void btnUpdate_Click(object sender, EventArgs e){
      ServiceClient BookData = new ServiceClient();
      int ID = int.Parse(txtBookID.Text);
      string Title = txtBookTitle.Text;
      float Price = float.Parse(txtBookPrice.Text);
      Book b = new Book {
          BookID = ID,
          BookPrice = Price,
          BookTitle = Title
      };
      BookData.UpdateBook(b);
      LoadData();
  }

  protected void gvBookList_SelectedIndexChanging
      (object sender, GridViewSelectEventArgs e)
  {
      GridViewRow row = gvBookList.Rows[e.NewSelectedIndex];
      txtBookID.Text = row.Cells[1].Text;
      txtBookTitle.Text = row.Cells[3].Text;
      txtBookPrice.Text = row.Cells[2].Text;
  }
}
```

➢ **Step 3 : Run MyWebApp application**

On the **MyWebApp** project , select MaintainBooks.aspx , right-click | View in Browser , result as the below figure .