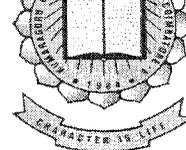




P-9108



ON DEMAND DATA CONSOLIDATION USING WEB SERVICES

A PROJECT REPORT

Submitted by

ARULAMUDHAN.R.S
PRASANTH.N

71206104005
71206104035

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2010



BONAFIDE CERTIFICATE

Certified that this project report “**ON DEMAND DATA CONSOLIDATION USING WEB SERVICES**” is the bonafide work of “**ARULAMUDHAN.R.S and PRASANTH.N**” who carried out the project work under my supervision.



SIGNATURE

Dr. S.THANGASAMY, Ph.D

DEAN OF THE DEPARTMENT

Computer Science and Engineering
Kumaraguru College of Technology,
Coimbatore-641 006.

SIGNATURE

Mr.M.NAGESWARA GUPTHA M.E



SUPERVISOR

SENIOR LECTURER/CSE Department
Kumaraguru College of Technology,
Coimbatore-641 006.

Submitted for the University Examination held on 15.04.2010



Internal Examiner



External Examiner

DECLARATION

We,

ARULAMUDHAN.R.S

71206104005

PRASANTH.N

71206104035

hereby declare that the project entitled ” **ON DEMAND DATA CONSOLIDATION USING WEB SERVICES**” is a record of original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any Institutions, for fulfillment of the requirement of the course study.

The report is submitted in partial fulfillment for the award of the degree of bachelor of computer science and engineering of Anna University ,Chennai.

Place: Coimbatore

Date: 15.4.2010


[ARULAMUDHAN.R.S]


[PRASANTH.N]

ACKNOWLEDGEMENT

We extend our sincere thanks to our principal **Dr.Ramachandran** ,Kumaraguru college of technology, Coimbatore,for being a constant source of inspiration and providing us with necessary facility to work on this project.

We would like to make a special acknowledgement and thanks to **Dr.S.THANGASAMY,Ph.D.**,Dean,Professor and Head of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

We express deep gratitude and gratefulness to our guide, **Mr.M.NAGESWARA GUPTHA,M.E.**, Department of Computer Science and Engineering for her supervision, enduring patience, active involvement and guidance.

We would also like to thank our project coordinator, **Mrs.P.DEVAKI, M.E** Assistant Professor, Department of Computer Science and Engineering, for her support during the course of our project.

We would like to convey our honest thanks to all members of staff of the Department and wealth of experience from which we greatly benefited.

ABSTRACT

Data consolidation is one of the type of data integration. Data Consolidation captures data from multiple source systems and integrates it into a single persistent data store. This data store may be used for reporting and analysis .It can act as a source of data for downstream different data environments . Metadata Data is a data of the data which contains a manual entry about the fields and field types of the data present at heterogeneous databases. We present a system that allows data consolidation applications through Webservice. Isolated tasks are performed as Web services. The intent behind a Web service is to drive the Internet as a transactional tool rather than simply a visual tool. A simple and flexible Web service workflow language is defined that composes the isolated tasks into a workflow which models the user's interactions. As all this is done encapsulated away from the user, the user effort is kept minimal.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGENO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
	1.1 DATA INTEGRATION	1
	1.2 EXAMPLE	3
	1.3 WEBSERVICES	4
	1.4 WEBSERVICECOMPONENTS	6
2.	LITERATURE SURVEY	7
	2.1 EXISTING SYSTEM	7
	2.2PROPOSED SYSTEM	7
	2.3 PROBLEM DEFINITION	8
	2.4 LAYOUT OF CONSOLIDATION PROCESS	9
3.	REQUIREMENT SPECIFICATION	10
	3.1 HARDWARE REQUIREMENTS	10
	3.2 SOFTWARE REQUIREMENTS	10
4.	IMPLEMENTATION	11
	4.1 MODULES	11
	4.2 MODULE DESCRIPTION	13
	4.3 SAMPLECODING	13

5	SCREENSHOTS	26
6.	CONCLUSION AND FUTURE WORK	36
	6.1 CONCLUSION	36
	6.2 FUTURE WORK	36
7.	REFERENCES	37

S.No	Figure Name	Page No
OUTPUT SCREENS		
1	Fig.5.1.LOGIN	27
2	Fig.5.2. ACCESS	28
3	Fig.5.3. XML	29
4	Fig.5.4.SQL	30
5	Fig.5.5 METADATA	31
6	Fig5.6.VERIFICATION OF METADATA	32
7	Fig.5.7 PROVIDING AS A SERVICE	33
8	Fig.5.8 ACQUIRING QUERY FROM THE USER	34
9	Fig5.9 FINAL CONSOLIDATED OUTPUT	35

CHAPTER 1

INTRODUCTION

1.1 Data integration involves combining data residing in different sources and providing users with a unified view of these data. Data integration appears with increasing frequency as the volume and the need to share existing data explodes. It has become the focus of extensive theoretical work, and numerous open problems remain unsolved.

Classification of Data Integration:

- Data consolidation
- Data federation
- Data propagation

Data Federation :

Data Federation provides a single virtual view of one or more source data files. When a business application issues a query against this virtual view, a data federation engine retrieves data from the appropriate source data stores, integrates it to match the virtual view and query definition, and sends the results to the requesting business application.

Advantage:

- * Provides access to current data
- * Eliminates the need to consolidate source data into another data store
- * Can be used when data security policies and license restrictions prevent source data being copied
- * Less costly than a physical data store
- * Good short term data integration solution following a company merger or acquisition.

Disadvantage:

- * Not well suited for retrieving and reconciling large amounts of data
- * Not well suited for applications where there are significant data quality problems in the source data
- * Potential performance impact and overhead of accessing multiple data sources at run time

Data Propagation

Data Propagation applications copy data from one location to another. These applications usually operate online and push data to the target location; i.e., they are event-driven. Updates to a source system may be propagated asynchronously or synchronously to the target system. Synchronous propagation requires that updates to both source and target systems occur in the same physical transaction.

Advantage:

- * Propagation guarantees the delivery of the data to the target
- * Two way exchange of data supported
- * Can be used for the real-time or near-real-time movement of data
- * Can be used for workload balancing, backup and recovery, and disaster recovery

Disadvantage:

- * Limited bulk data movement and data restructuring / cleansing capabilities

Data Consolidation

Data Consolidation captures data from multiple source systems and integrates it into a single persistent data store. This data store may be used for reporting and analysis as in data warehousing, or it can act as a source of data for downstream applications as in an operational data store.

Advantage:

- * Can be used to push or pull (or both) data from source to destination
- * Allows for large volumes of data to be transformed
- * Can handle complex transformations
- * Able to deliver complex queries without negative impact on operational systems

1.2 Example

Consider a web application where a user can query a variety of information about cities (such as crime statistics, weather, hotels, demographics, etc). Traditionally, the information must exist in a single database with a single schema. But any single enterprise would find information of this breadth somewhat difficult and expensive to collect. Even if the resources exist to gather the data, it would likely duplicate data in existing crime databases, weather websites, and census data. When an application-user queries the mediated schema, the data-integration solution transforms this query into appropriate queries over the respective data sources. Finally, the virtual database combines the results of these queries into answers into the users query.

1.3 Web service:

Data consolidation without webservice leads to a problem of lack of structure and semantic annotation of individual data items pose a severe restrictions on its usefulness to the end user. A Web service is an application that is written to meet Internet and Extensible Markup Language (XML) technology standards. It performs a specific task and is made available to other users through a network. By expanding on this, according to the World Wide Web Consortium . Web service is a software system identified by a Universal Resource Identifier (URI), is defined, described, and found using XML, Uses XML-based messages that are sent over Internet-based protocols to support direct interaction with other software applications. Web services are typically application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

In common usage the term refers to clients and servers that communicate over the Hypertext Transfer Protocol (HTTP) protocol used on the web. Such services tend to fall into one of two camps: Big Web Services^[1] and Restful Web Services.

"Big Web Services" use Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard and have been popular with traditional enterprise. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL). The latter is not a requirement of a SOAP *endpoint*, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as spring, Apache Axis2 and Apache

CXF being notable exceptions). Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a web service.

When running composite web services, each sub service can be considered autonomous. The user has no control over these services. Also the web services themselves are not reliable; the service provider may remove, change or update their services without giving notice to users. The reliability and fault tolerance is not well supported; faults may happen during the execution. Exception handling in the context of web services is still an open research issue.

The W3C defines a "web service" as "a software system designed to support interoperable machine-to-machine interaction over a network

Service-oriented architecture

Web services can also be used to implement an architecture according to service oriented (**SOA**) concepts, where the basic unit of communication is a message, rather than an operation. This is often referred to as message oriented services.

SOA Web services are supported by most major software vendors and industry analysts.

1.4 Web services: Components

To leverage a Web service, a network, the ability to do XML-based messaging, and a description of the service are the significant protocols.

Network

A network is the foundation needed to access a Web service or to make a Web service available. Generally, this network is based on Hypertext Transfer Protocol (HTTP), but it can also be based on other protocols.

XML-based messaging

XML-based messaging enables the communication between Web services and clients. SOAP is an XML protocol that implements this communication, including the publishing, finding, binding, and calling of a Web service.

Service description

The service description is written in the WSDL specification. WSDL is a series of XML documents that describe the service itself, the programming interface for the service, and the location of the service.

CHAPTER 2

LITERATURE REVIEW

2.1 Existing system:

Airline operational data has to be combined with multiple government, airport and proprietary sources to present a complete picture. Delivering consistent, reliable, and timely flight status information through a variety of outlets is a foundation for quality service and a satisfied customer. Customers expect convenient access to up-to-date status information from airport monitors, automated phone systems, websites, and mobile phone messaging. For most carriers there are at least three categories of data consumers like reservation, status enquiry, current status. Combining data from multiple sources helps improve the accuracy and timeliness of the final result. Algorithms for combining the disparate data sources must be intelligent and adaptable to meet changing conditions.

2.2 Proposed system:

The proposed system preserves that the data consolidation process is provided as a webservice. In this implementation data of different types located at different data source is clustered into a single by appropriate function. When the user post a query as a fieldname and fieldtype, the service searches whether the data is available at existing database and if it matches the consolidation process is being carried out. In case of nonavailability of information warning message will be displayed.

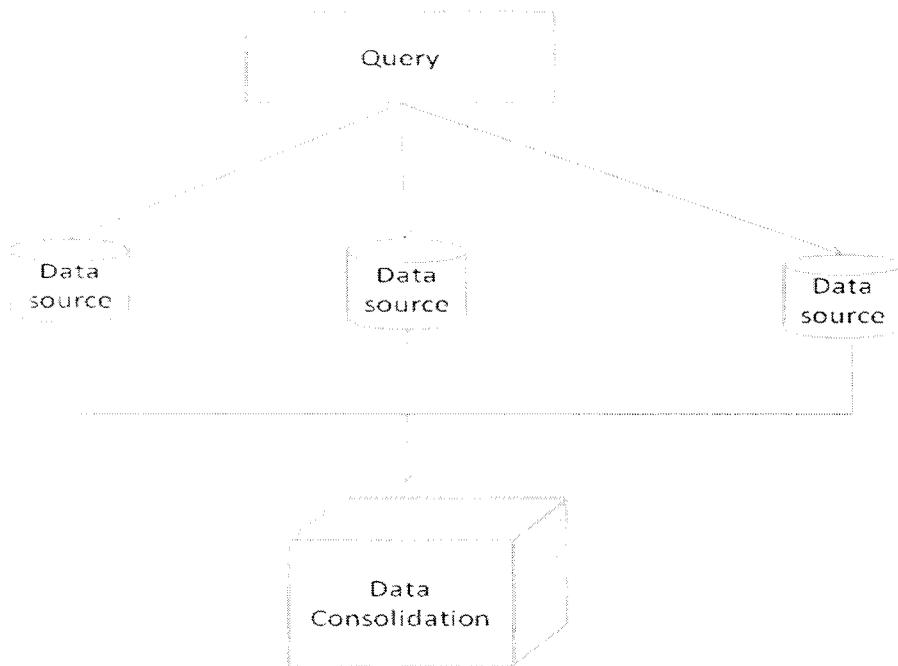
2.3 Problem Definition:

Relative types of data are floated at different data sources in today's cyber world in order to combine all relative data, data consolidation process provides the way. Data consolidation is an emerging technique which involves assembling information located at various database. Data consolidation is provided as a web service where Web services are typically application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

In common usage the term refers to clients and servers that communicate over the Hypertext Transfer Protocol (HTTP) protocol used on the web

Data consolidation is for more better than the other two techniques such as data propagation and data federation. This technique is applicable for many uses in various organisations. This implementation is for student information system where users expectations will get satisfied.

2.4 Layout of Data consolidation process:



ETL PROCESS

Steps involved in data consolidation:

- 1.Application starts operating by perfect registration,registration involves getting user name and password from the user.
- 2.Query from the user is acquired in string format.
- 3.Verification of posted query with the meta file where it contains field names and its field types of different data sources.
- 4.The process begins by storing all the values of databases as a dataset.
- 5.Consolidation starts if there is availability of field name and field type in meta file.
- 6.Warning message will be displayed in case of absence in field name.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 Hardware requirement:

Processor : Pentium IV
Hard disk drive : 120GB
Motherboard : Intel

3.2 Software requirement:

Operating System : Windowsxp
Front end : Asp.net
Back end : Sql,Access,Xml

CHAPTER 4

IMPLEMENTATION

4.1 Modules

1. Log in
2. Database Maintenance,
3. Meta descriptor,
4. Data consolidation.



4.2 Modular Description

Log in

This allows the user to access the application by providing the appropriate username and password of the applicant.

Database Maintenance

Admin have authority to create and modify the database. Distributed environment contains different types of databases, from which the user can retrieve information that he requires.

P-3108

For example:

1. xml db

```
<xmldb type="field">  
<name>Name</name>  
<type>String</type>  
</xmldb>
```

2. Access db

```
<access type="field">  
<name>Name</name>  
<type>String</type>  
</access>
```

3. sql

```
<sql type="field">  
<name>Name</name>  
<type>String</type>  
</sql>
```

Meta Descriptor:

Meta descriptor contains the collection of information which is retrieved from different data sources as per the user's desired query. The information such as the field name and their appropriate field type of the corresponding databases are being stored.

Dataconsolidation

When the field name and field type of the query posted by the user matches with the content of metadata then data values of different databases which is stored as datasets gets consolidated

4.3 Sample coding

```
Imports System.Web.Services
```

```
Imports System.Web.Services.Protocols
```

```
Imports System.ComponentModel
```

```
Imports System.Xml
```

```
Imports System.Data.Sql
```

```
Imports System.Data.SqlClient
```

```
Imports System.IO
```

```
Imports System.Xml.XPath
```

To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

```
<System.Web.Script.Services.ScriptService()> _  
<System.Web.Services.WebService(Namespace:="http://tempuri.org/")> _  
<System.Web.Services.WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
```

<ToolboxItem(False)> _

Public Class Service1

Inherits System.Web.Services.WebService

Public accessconnection As New

OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\school.mdb")

Public a As New ArrayList()

Public b As New ArrayList()

Public c As New ArrayList()

Public sqlconnection As New SqlConnection("Data

Source=.\SQLEXPRESS;AttachDbFilename=F:\final data

consol\sql.mdf;Integrated Security=True;User Instance=True")

Dim AccArray() As Integer

Dim AccArray() As Integer = {0, 0}

Dim sqlArray() As Integer = {0, 0}

Dim objCommand As OleDb.OleDbCommand

Dim objDataReader As OleDb.OleDbDataReader

Dim Dataset1 As DataSet

Dim Dataset2 As DataSet

Dim Dataset3 As DataSet

Dim Dataset4 As DataSet

Dim flag1 As Integer = 0

Dim flag2 As Integer = 0

Dim flag3 As Integer = 0

Dim flag4 As Integer = 0

Dim flag5 As Integer = 0

Dim flag6 As Integer = 0

<WebMethod(> _

Public Function Data(ByVal Data1 As String, ByVal Data2 As String,
ByVal Data3 As String, ByVal Data4 As String, ByVal Data5 As String,
ByVal Data6 As String)

 accessconnection.Open()

 createDataset(Data1, Data3, Data5)

 MessageBox.Show(accessconnection.State)

 Dim m_xmlr As XmlTextReader

 Create the XML Reader

 m_xmlr = New XmlTextReader("C:\meta.xml")

 Disable whitespace so that you don't have to read over whitespaces

 m_xmlr.WhitespaceHandling = WhitespaceHandling.None

 read the xml declaration and advance to family tag

 m_xmlr.Read()

 read the family tag

 m_xmlr.Read()

 Load the Loop

 While Not m_xmlr.EOF

 Go to the name tag

 m_xmlr.Read()

 if not start element exit while loop

 If Not m_xmlr.IsStartElement() Then

 Exit While

```

End If
Get the Gender Attribute Value
Dim genderAttribute = m_xmlr.GetAttribute("field")
Dim Databasename = m_xmlr.Name
m_xmlr.Read()
Dim firstNameValue = m_xmlr.ReadElementString("name")
Dim lastNameValue = m_xmlr.ReadElementString("type")
If firstNameValue = Data1 Then
    If Databasename = "xmldb" Then
        Dim expression As String = "/school/student/" & Data1
        MessageBox.Show("found in xml")
        a = getElements(expression)
    End If
End If
If firstNameValue = Data3 Then
    If Databasename = "xmldb" Then
        Dim expression As String = "/school/student/" & Data3
        MessageBox.Show("found in xml")
        b = getElements(expression)
    End If
End If
If firstNameValue = Data5 Then
    If Databasename = "xmldb" Then
        Dim expression As String = "/school/student/" & Data5
        MessageBox.Show("found in xml")
    End If
End If

```

```
c = getElements(expression)
```

```
End If
```

```
End If
```

```
If firstNameValue = Data1 And lastNameValue = Data2 Then
```

```
    If Databasename = "access" Then
```

```
        AccArray(0) = 0
```

```
        Dataset1 = acess(firstNameValue, lastNameValue)
```

```
        flag1 = 1
```

```
        '   MessageBox.Show("falg1 added")
```

```
    End If
```

```
    If Databasename = "sql" Then
```

```
        Dataset2 = acess(firstNameValue, lastNameValue)
```

```
        flag2 = 1
```

```
        MessageBox.Show("falg2 added")
```

```
    End If
```

```
End If
```

```
If firstNameValue = Data3 And lastNameValue = Data4 Then
```

```
    If Databasename = "access" Then
```

```
        ' AccArray(0) = 0
```

```
        flag3 = 1
```

```
        MessageBox.Show("falg3 added")
```

```
        Dataset3 = acess(firstNameValue, lastNameValue)
```

```
    End If
```

```
    If Databasename = "sql" Then
```

```
        flag4 = 1
```

```

        MessageBox.Show("falg4 added")
    sqlArray(0) = 0
    Dataset4 = acces(firstNameValue, lastNameValue)
End If
'MessageBox.Show(" Found in database: " & Databasename _
' & " Name: " & firstNameValue & " Type: " _
' & lastNameValue)
'MessageBox.Show(vbCrLf)
End If
If firstNameValue = TextBox5.Text And lastNameValue =
TextBox6.Text Then
    ' MessageBox.Show(" Found in database: " & Databasename _
    ' & " Name: " & firstNameValue & " Type: " _
    ' & lastNameValue)
'MessageBox.Show(vbCrLf)
End If
If firstNameValue = Data5 And lastNameValue = Data6 Then
    If Databasename = "access" Then
        AccArray(0) = 0
        flag5 = 1
        '        MessageBox.Show("falg5 added")
        Dataset3 = acces(firstNameValue, lastNameValue)
    End If
    If Databasename = "sql" Then
        flag6 = 1

```

```

        '        MessageBox.Show("falg6 added")
        sqlArray(0) = 0
        Dataset4 = acess(firstNameValue, lastNameValue)
    End If
    'MessageBox.Show(" Found in database: " & Databasename _
    ' & " Name: " & firstNameValue & " Type: " _
    ' & lastNameValue)
    'MessageBox.Show(vbCrLf)
End If
End While
'MessageBox.Show(a.Count)
insertData(a, b, c)
close the reader
Console.Read()
Dataset1 = acess(Data1, Data3, Data5)
Dataset2 = Sql(Data1, Data3, Data5)
Dataset3 = xmldata()
Dim dSet As New DataSet("student")
Dim fs As System.IO.FileStream
open the xml file so we can use it to fill the dataset
'Try
fs = New System.IO.FileStream("C:\meta.xml", FileMode.Open)
    Catch e As Exception
Debug.WriteLine(e)
End Try

```

'fill the dataset

Try

 dSet.ReadXml("C:\school.xml")

Catch ex As Exception

 Debug.WriteLine(ex)

End Try

Dataset1.Merge(Dataset2)

Dataset1.Merge(Dataset3)

m_xmlr.Close()

Return Dataset1

End Function

Public Function getElements(ByVal exp As String)

 Dim arr As New ArrayList()

 Dim file As String = "C:/school.xml"

 Dim doc As New XPathDocument(file)

 Dim nav = doc.CreateNavigator()

 Dim expr As XPathExpression

 expr = nav.Compile(exp)

 Dim itre As XPathNodeIterator = nav.Select(expr)

 MessageBox.Show("Lement function")

Try

 While itre.MoveNext()

 Dim nav2 As XPathNavigator = itre.Current.Clone()

 arr.Add(nav2.Value)

 ' MessageBox.Show(nav2.Value)

End While

Catch ex As Exception

End Try

Return arr

End Function

Public Function Sql(ByVal Data As String, ByVal Data1 As String,
ByVal Data2 As String)

' sqlconnection.Open()

Dim Query As String = ""

If ((flag2 = 1) And (flag4 = 1) And (flag6 = 1)) Then

Query = "Select " & Data & "," & Data1 & "," & Data2 & " from
school "

ElseIf ((flag2 = 1) And (flag4 = 1)) Then

Query = "Select " & Data & "," & Data1 & " from school "

ElseIf ((flag2 = 1) And (flag6 = 1)) Then

Query = "Select " & Data & "," & Data2 & " from school "

ElseIf ((flag4 = 1) And (flag6 = 1)) Then

Query = "Select " & Data1 & "," & Data2 & " from school "

ElseIf (flag2 = 1) Then

Query = "Select " & Data & " from school "

'Query = "Select " & Data & "," & Data1 & " from school "

ElseIf (flag6 = 1) Then

Query = "Select " & Data2 & " from school "

ElseIf (flag4 = 1) Then

Query = "Select " & Data1 & " from school "

```
'Query = "Select " & Data & "," & Data1 & " from school "
```

```
End If
```

```
'MessageBox.Show(Query)
```

```
Dim sqladapter As New SqlDataAdapter(Query, sqlconnection)
```

```
Dim dataset As New DataSet()
```

```
sqladapter.Fill(dataset, "school")
```

```
' sqlconnection.Close()
```

```
Return dataset
```

```
End Function
```

```
Public Function dis(ByVal ad As ArrayList)
```

```
'Dim s As String
```

```
'For i = 0 To ad.Count - 1
```

```
'MessageBox.Show(ad.Count)
```

```
Next
```

```
End Function
```

```
Public Function xmldata()
```

```
Dim query As String = "select * from temp"
```

```
'MessageBox.Show(query)
```

```
Dim sqladapter As New SqlDataAdapter(query, sqlconnection)
```

```
Dim dataset As New DataSet()
```

```
sqladapter.Fill(dataset, "school")
```

```
' sqlconnection.Close()
```

```
Return dataset
```

```
End Function
```

Public Function access(ByVal Data As String, ByVal Data1 As String,
ByVal Data2 As String)

Try

' accessconnection.Open()

'Dim Query As String = "Select " & Data & " from school "

Dim Query As String = ""

If ((flag1 = 1) And (flag3 = 1) And (flag5 = 1)) Then

 Query = "Select " & Data & ", " & Data1 & ", " & Data2 & " from
school "

ElseIf ((flag1 = 1) And (flag3 = 1)) Then

 Query = "Select " & Data & ", " & Data1 & " from school "

ElseIf ((flag1 = 1) And (flag5 = 1)) Then

 Query = "Select " & Data & ", " & Data2 & " from school "

ElseIf ((flag3 = 1) And (flag5 = 1)) Then

 Query = "Select " & Data1 & ", " & Data2 & " from school "

ElseIf (flag1 = 1) Then

 Query = "Select " & Data & " from school "

 'Query = "Select " & Data & ", " & Data1 & " from school "

ElseIf (flag5 = 1) Then

 Query = "Select " & Data2 & " from school "

ElseIf (flag3 = 1) Then

 Query = "Select " & Data1 & " from school "

 'Query = "Select " & Data & ", " & Data1 & " from school "

End If

'MessageBox.Show(Query)

```

Dim oleadapter As New OleDb.OleDbDataAdapter(Query,
accessconnection)

Dim dataset As New DataSet()
oleadapter.Fill(dataset, "school")
'accessconnection.Close()

Return dataset

'SqlDataAdapter da = new SqlDataAdapter(select,sqlConnection1);
'// Create a New DataSet
'DataSet ds = new DataSet();
'// Fill The DataSet With the Contents of the Stock Table
'da.Fill(ds , "stock");
'// Now Return ds which is a DataSet
'return(ds);

'Dim objCommand As New OleDb.OleDbCommand(Query,
accessconnection)

'Dim objDataReader As OleDb.OleDbDataReader
'objDataReader = objCommand.ExecuteReader()
'MessageBox.Show(objDataReader.HasRows)
'objDataReader.Read()
'MessageBox.Show(objDataReader.GetString(0))

Catch

'MessageBox.Show("eXceptiop in connection")

End Try

End Function

```

```

Public Function createDataset(ByVal field1 As String, ByVal field2 As
String, ByVal field3 As String)
    sqlconnection.Open()
    Dim Drop As String = "Drop table temp"
    Dim sqlCommand As New SqlCommand(Drop, sqlconnection)
    sqlCommand.ExecuteNonQuery()
    Dim Query As String = "create table temp (" & field1 & " varchar(30)"
& "," & field2 & " varchar(30)" & "," & field3 & " varchar(30))"
    Dim sqlCommand1 As New SqlCommand(Query, sqlconnection)
    sqlCommand1.ExecuteNonQuery()
    'MessageBox.Show(Query)
    sqlconnection.Close()
End Function

```

```

Public Function insertData(ByVal A1 As ArrayList, ByVal A2 As
ArrayList, ByVal A3 As ArrayList)
    sqlconnection.Open()
    Dim i As Integer
    For i = 0 To A1.Count - 1
        Dim Query As String = "insert into temp values (" & A1.Item(i) &
",," & A2.Item(i) & ",," & A3.Item(i) & ",," )"
        Dim sqlCommand1 As New SqlCommand(Query, sqlconnection)
        sqlCommand1.ExecuteNonQuery()
        'MessageBox.Show(Query)
    Next
    sqlconnection.Close()

```

```

    End Function
End Class

Public Partial Class WebForm1
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        End Sub

        Protected Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
            Dim ser As New Service1()
            'ser.Data("Name", "String", "rollno", "int")
            GridView1.DataSource = ser.Data(TextBox1.Text, TextBox2.Text,
TextBox3.Text, TextBox4.Text, TextBox5.Text, TextBox6.Text)
            GridView1.DataBind()
        End Sub
    End Class

```

CHAPTER 5

SCREENSHOTS

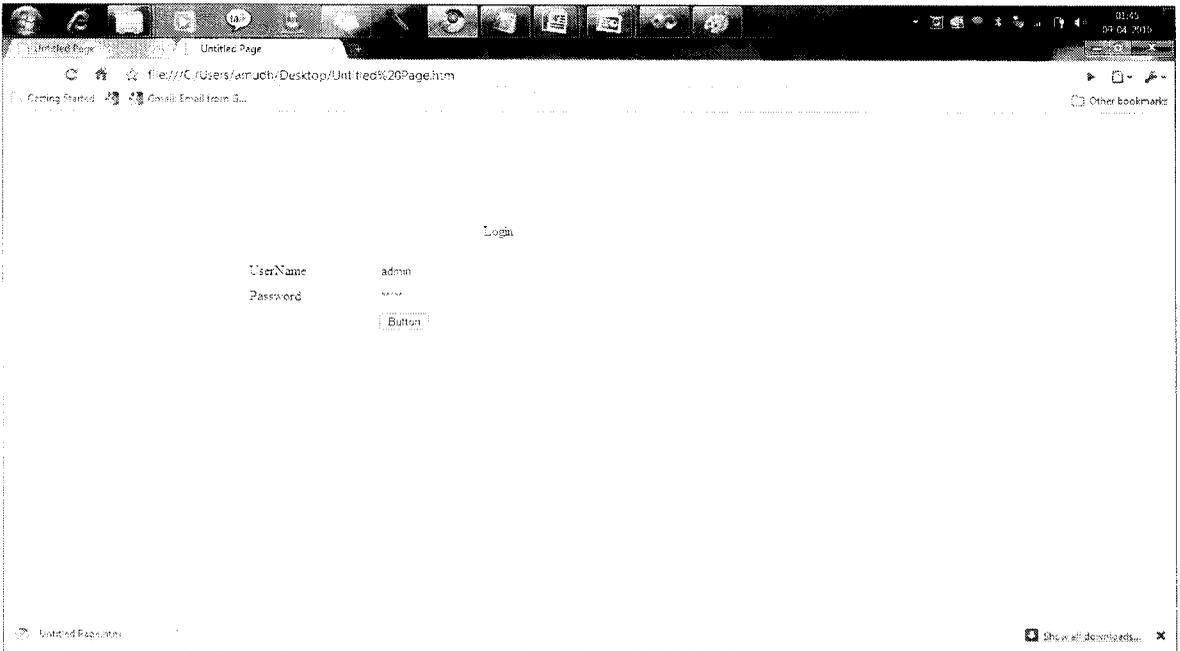


Fig.5.1.LoginForm

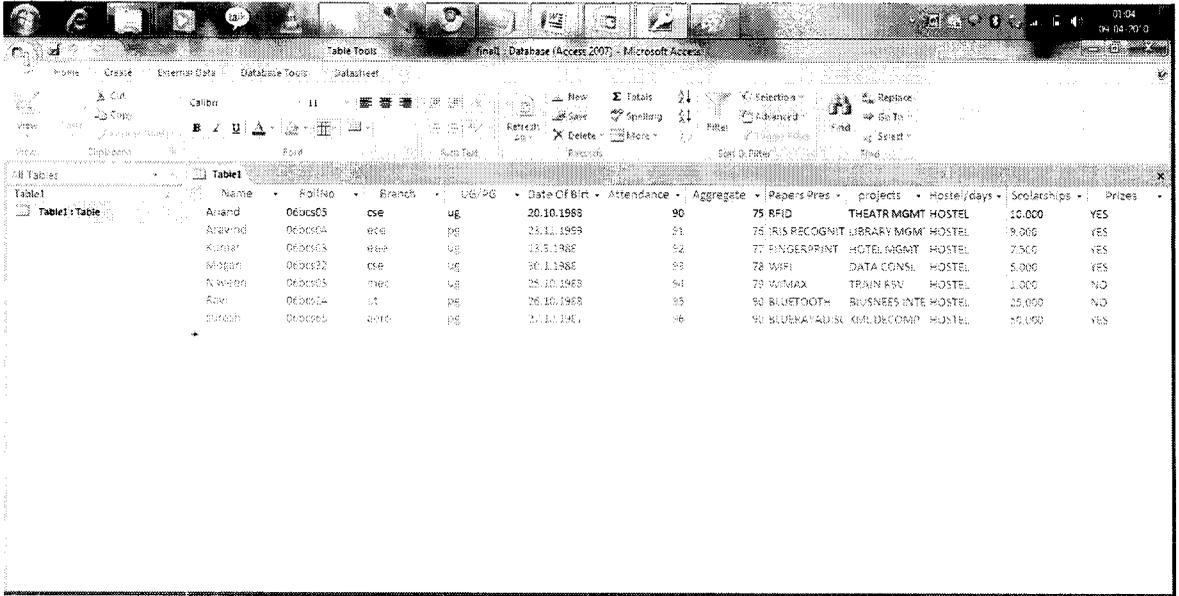


Fig.5.2. Access

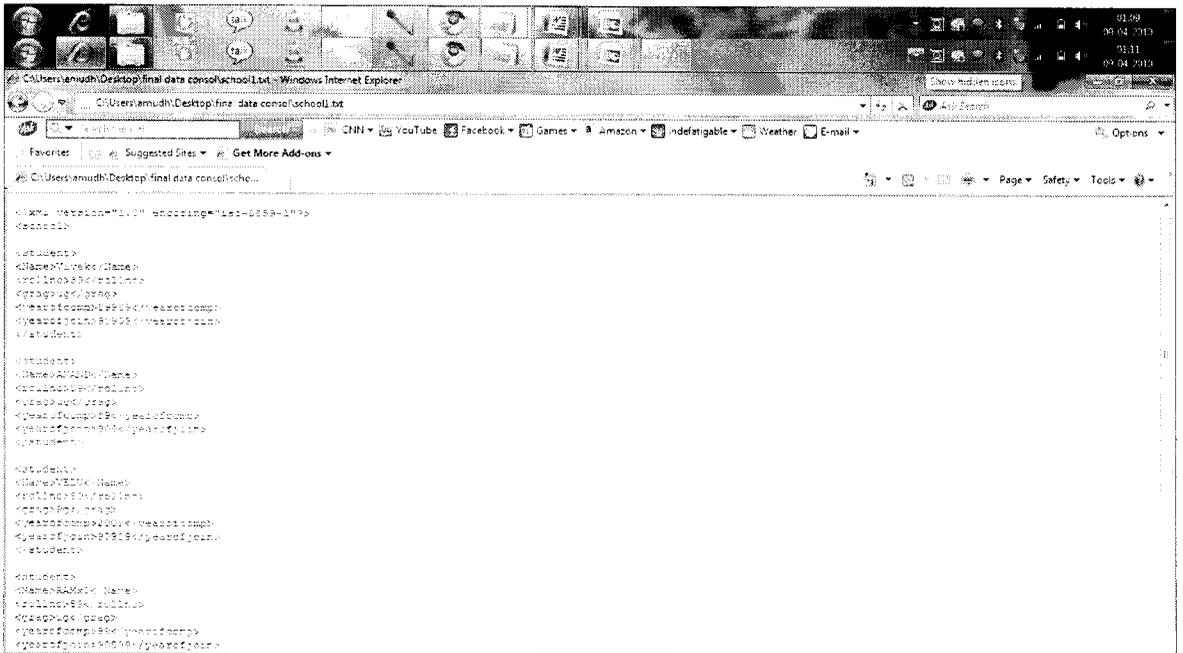


Fig.5.3. XML

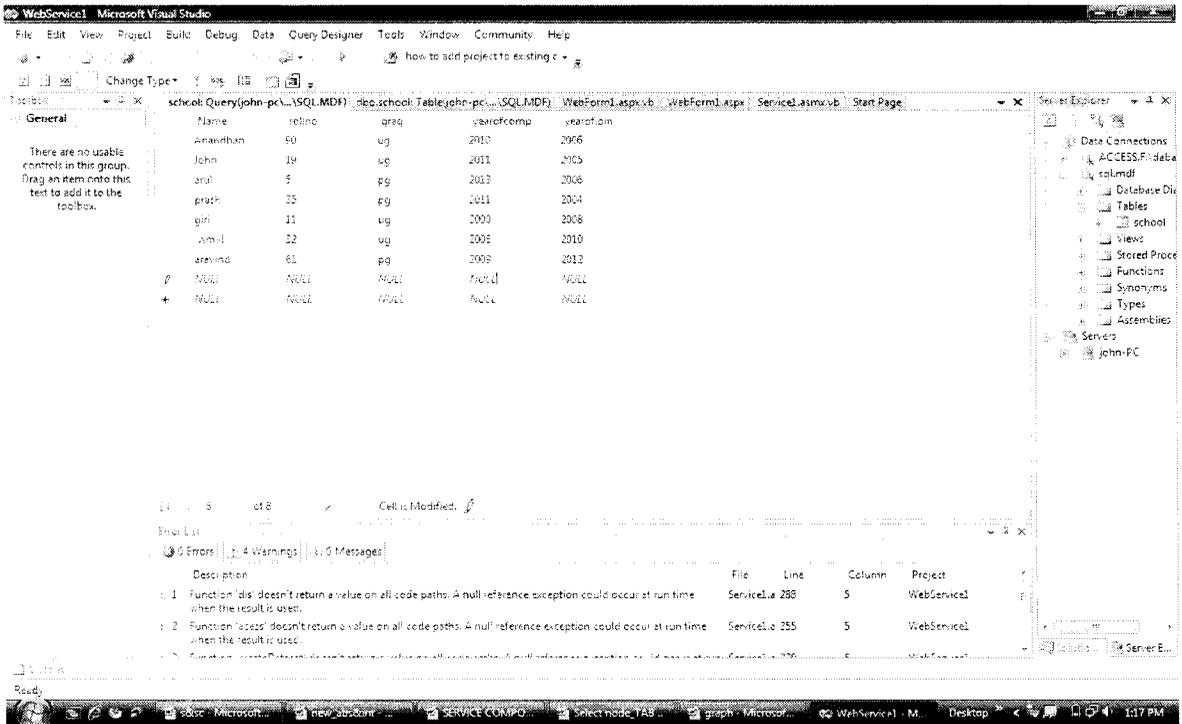


Fig 5.4.SQL

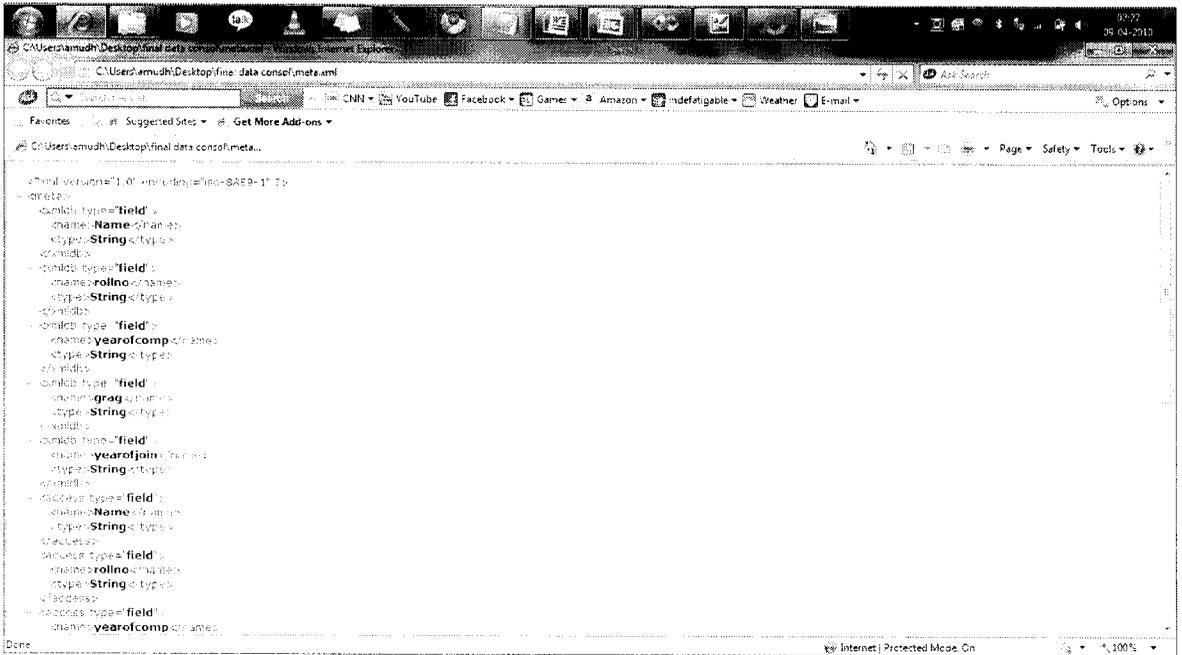


Fig.5.5 METADATA

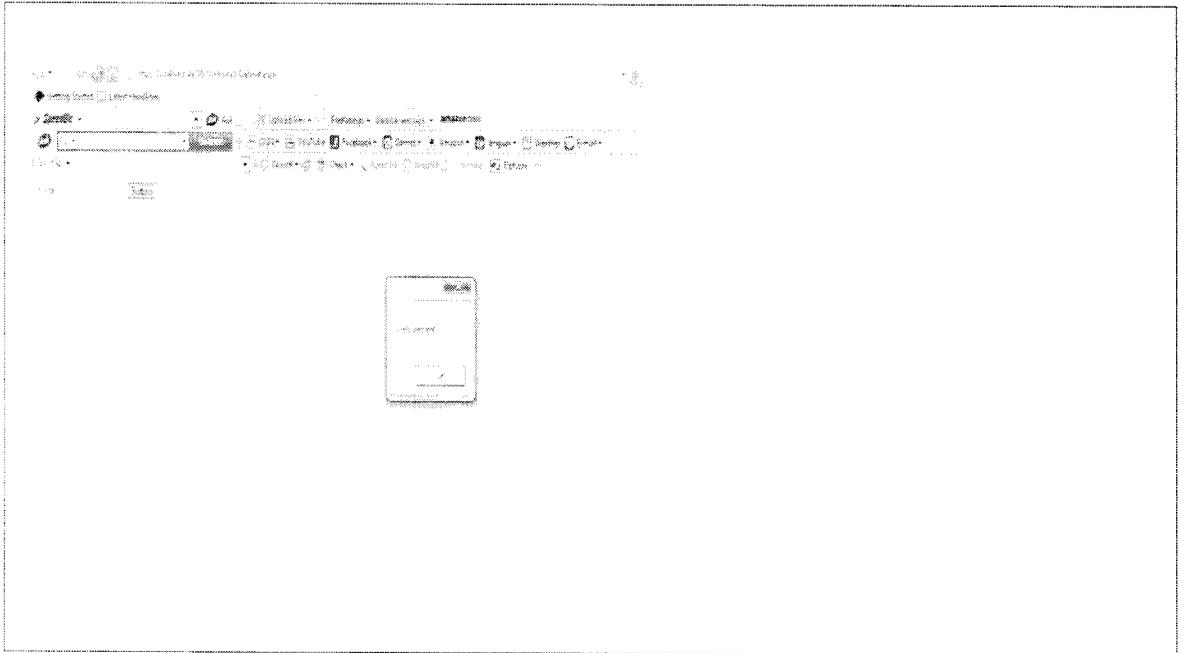


Fig5.6. VERIFICATION OF METADATA FOR POSTED QUERY

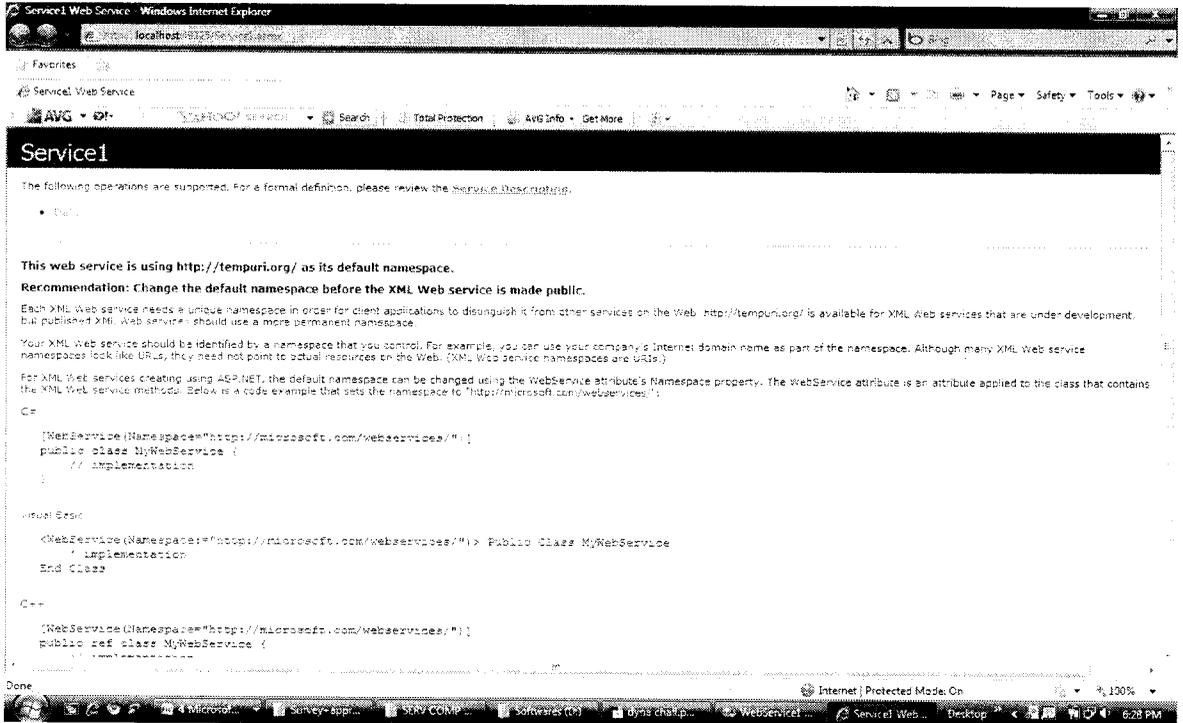


Fig.5.7 PROVIDING AS A SERVICE



Fig.5.8 ACQUIRING QUERY FROM THE USER

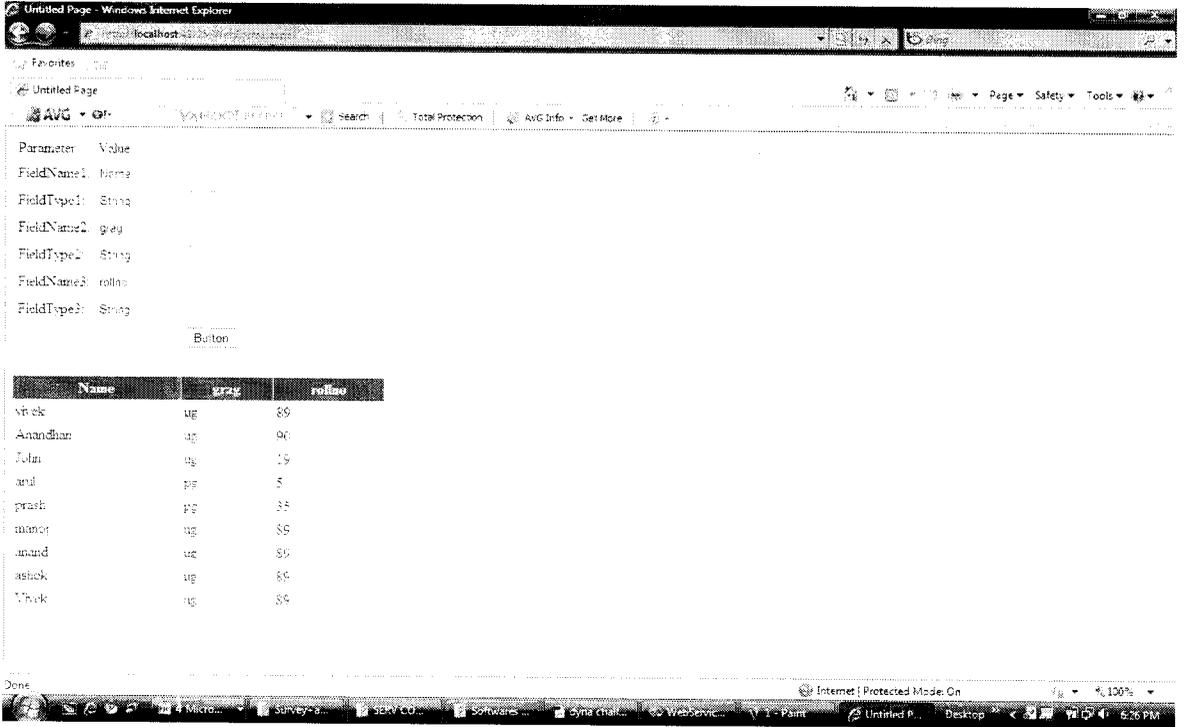


Fig5.9 FINAL CONSOLIDATED OUTPUT

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Data is an important source for almost all the applications. Web services provides that opportunities for user to utilize the services from anywhere, any environment, platform even at any time. By consuming the data consolidation as an web service, user can access the data through on demand needs.

6.2 Future Work

In future the project can be enhanced by applying visualization technique for the current implementation. By having visualization, data can be analyzed and report can be generated based on user constraints and hence more comprehensible form of data can be presented to the users.

REFERENCES:

- [1] D.D. Duong and V. Wuwongse, “Database Schema Integration Using XDD,” Proceedings of Advances in Web-Age Information Management Conference, China: Lecture Notes in Computer Science, Springer Verlag, vol. 2762, 2003, pp. 92-103.

- [2] P. Gianolli and J. Mylopoulos, “A semantic approach to data integration “Proceedings of the 20th. International Conference on Conceptual Modeling (ER), Yokohama, Japan, 2001.

- [3] G. Lausen and P.J. Marron, “Adaptive evaluation techniques for querying XML-based E-Catalogs,” DBLP, 2002, pp. 19-28.

- [4] Ludascher, R. Himmeroder, G. Lausen, W. May and C. Schleppehorst “Managing semistructured data with FLORID: a Deductive object- oriented perspective,” Journal of Information Systems , vol. 23, no. 8, 1998, pp. 1-25.

- [5] W. May, “Logic-based data integration: a semi-materializing approach,” Journal of Applied Logic , vol. 3, No. 1, 2005, pp. 271–307.