INS PRACTICALS

Write programs to implement the following Substitution Cipher Techniques: Caesar Cipher, Monoalphabetic Cipher

Caesar Cipher:

```
import java.util.Scanner;
/**
* @author 0x4427
 */
public class CeaserCipher {
   String message;
   static int key;
   //for enc
   static String encryptCeaser(String message1, int key1)
   {
      char ch;
      String encryptedMessage = "";
      for (int i=0; i< message1.length();++i) {
         ch = message1.charAt(i);
         System.out.println((int)ch);

         if (ch>= 'a' && ch <= 'z') {
            ch = (char) (ch + key1);
            if (ch>= 'z') {
               ch = (char) (ch -  'z' + 'a' - 1);
            }
            encryptedMessage += ch;
      }
         else if (ch >='A' && ch <= 'Z') {
```

```
            ch = (char) (ch + key1);
            if (ch > 'Z') {
               ch = (char) (ch -  'Z' + 'A' - 1);
            }
            encryptedMessage += ch;
        }
        else {
              encryptedMessage += ch;
        }
        }
    return encryptedMessage;
}
 //for dec
static String decryptCeaser(String message1, int key1)
{
    char ch;
    String decryptedMessage = "";
    for (int i=0; i< message1.length();++i) {
        ch = message1.charAt(i);
        System.out.println((int)ch);

        if (ch>= 'a' && ch <= 'z') {
            ch = (char) (ch - key1);
            if (ch>= 'z') {
              ch = (char) (ch -  'z' + 'a' - 1);
            }
            decryptedMessage += ch;
    }
        else if (ch >='A' && ch <= 'Z') {
```

```
            ch = (char) (ch - key1);
            if (ch > 'Z') {
                ch = (char) (ch -  'Z' + 'A' - 1);
            }
            decryptedMessage += ch;
        }
        else {
                decryptedMessage += ch;
        }
        }
    return decryptedMessage;
}
public static void main(String args[]) {
String plainText;
int key;
String CipherText;

Scanner sc = new Scanner(System.in);

System.out.println("Enter a message to encrypt: ");
plainText = sc.nextLine();

System.out.println("Enter key: ");
key = sc.nextInt();

CipherText = encryptCeaser(plainText, key);
System.out.println("Cipher Text=" + CipherText);

plainText = decryptCeaser(CipherText, key);
```

```
   System.out.println("Plaint Text=" + plainText);
   System.out.println("Input Key=" + key);
       }
  }
```

Monoalphabetic Cipher:

```
import java.util.Scanner;
/**
 * @author comp101
 */
public class MonoAlpha {
   public static void main(String[] args) {
     final char RALPHABETS[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
     final char MALPHABETS[] = {'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'Z', 'X', 'C', 'V', 'B', 'N', 'M'};

      char citext[] = new char[20];
      char detext[] = new char[20];
      int i, len;
      String pt;

      Scanner s = new Scanner(System.in);
      System.out.println("Enter Plain Text:");
      pt = s.nextLine();
      len = (pt.length());
      for (i = 0; i < len; i++) {
         for (int j = 0; j < 26; j++) {
            if (RALPHABETS[j] == pt.charAt(i)) {
               citext[i] = MALPHABETS[j];
               break;
```

```
            }
          }
        }
        System.out.println("Cipher Text:");

        for (i = 0; i < len; i++) {
           System.out.println(citext[i]);
        }
        String b = new String(citext);
        for (i = 0; i < len; i++) {
           for (int j = 0; j < 26; j++) {
              if (MALPHABETS[j] == b.charAt(i)) {
                 detext[i] = RALPHABETS[j];
                 break;
              }
           }
        }

        System.out.println("\nDecipher text:");
         for (i = 0; i < len; i++)
         {
          System.out.println(detext[i]);
     }
   }
}
```

Write programs to implement the following Substitution Cipher Techniques:

Vernam Cipher:

```
import java.lang.Math;
/**
* @author 0x4427
 * without user input
 */
public class VernamCip
   public static void main(String args[]) {
      String pt = new String("computerscien");
      char[] arText = pt.toCharArray();

      String key = new String("ABCDEFGXYZHIJ");
      char[] arKey = key.toCharArray();

      char[] ct = new char[13];
      System.out.println("Encoded " + pt + " to be... ");

      for (int i = 0; i < arText.length;  i++ )
      {
         ct[i] = (char) (arText[i] ^ arKey[i]);
          System.out.print(ct[i]);
      }
       System.out.println("\n Decoded to be... " );
       for (int i = 0; i < ct.length;  i++ ) {

          char temp = (char) (ct[i] ^ arKey[i]);
           System.out.print(temp);
    }      }    }
```

Write programs to implement the following Transposition Cipher Techniques: - Rail Fence Cipher, Simple Columnar Technique

Rail Fence Cipher:

```
import java.util.*;
/**
 * @author 0x4427
 */
public class P3_RailFence {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      int i = 0, j, l, r, s, c = 0, t = 0;
      String plainText = " ", cipherText = " ";
      System.out.println("RAIL FENCE CIPHER");
       System.out.println("Enter the plaintext:");
       plainText = in.nextLine();
        System.out.println("Enter the Rails:");
        r = in.nextInt();

        s = plainText.length();
        System.out.println("r = " +r);
        System.out.println("s = " +s);

        char ct[][] = new char[r][s];
        while(i<r){
          ct[i][c] = plainText.charAt(t);
          t++;
          if (i == r - 1 && t <= plainText.length() - 1){
             c = c +1;
             for(j=r-2; j>=0; j--){
```

```
                ct[j][c] = plainText.charAt(t);
                t++;
                if(j == 0 && t<= plainText.length() - 1){
                   c++;
                   i = 0;
                }
              }
            }
            i = i +1;
          }
          for (i = 0; i < r; i++){
             for (j = 0; j < ct[i].length; j++){
                char d = ct[i][j];
                if (d == '\0'){
                   continue;
                } else {
                   cipherText = cipherText + ct[i][j];
                }
             }
          }
          System.out.println("The Cipher text is:\n" + cipherText);
          System.out.println(cipherText.length());
       }
    }
```

Simple Columnar Technique:

```
import java.util.Scanner;
/**
 * @author 0x4427
 * with key
 */
public class SCT {
    public static void main(String args[]) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your plain text");

         String ptext = sc.nextLine();
          System.out.println("Enter key");
         String key = sc.nextLine();

         String pt = key.concat(ptext);

        System.out.println("Enter the no. of rows ");
        int r = Integer.parseInt(sc.nextLine());
       System.out.println("Enter the no. of columns ");
          int c = Integer.parseInt(sc.nextLine());

       int count = 0;
       char matrix[][] = new char [r] [c];
        for (int i = 0; i < r;  i++ ) {
            for (int j = 0; j < c;  j++ ) {
           if (count >= pt.length()) {
              matrix[i][j] = ' ';
              count++;
```

```
            } else {
                matrix[i][j] = pt.charAt(count);
                  count++;
               }
             }
           }
          for (int i = 0; i < r;  i++ ) {
                  for (int j = 0; j < c;  j++ ) {
                     System.out.print("\t" + matrix[i][j]) ;
                  }
                   System.out.println("\n") ;
          }
            System.out.println("\tEnter the order of columns you want to view them in") ;
            int choice[] = new int[c];
               for (int k = 0; k < c;  k++ ) {
                  System.out.println("\nChoice " + k + "~>") ;
                  choice[k] = Integer.parseInt(sc.nextLine());
               }
                System.out.println("\nCipher text in matrix is ~> " ) ;
                String cipher = "";
                for (int j = 0; j < c;  j++ ) {
                      int k = choice[j];
                        for (int i = 0; i < r;  i++ ) {
                           cipher +=   matrix[i][j] ;
                        }
                }
                  System.out.print(cipher ) ;
    }
}
```

Write program to encrypt and decrypt strings using DES Algorithm:

```
import java.math.BigInteger;
import java.security.*;
import javax.crypto.*;
import java.util.*;
import javax.crypto.spec.*;
import javax.swing.JOptionPane;
public class DES {
byte[] skey = new byte[1000];
String skeyString;
static byte [] raw;
String inputMessage, encryptedData, decryptedDdata;
//Constructor
public DES() throws Exception {
generatedSymmetricKey();//1
inputMessage = JOptionPane.showInputDialog(null, "Enter Message to encrypt"); //2 Input
byte[] ibyte = inputMessage.getBytes();
byte[] ebyte = encrypt(raw, ibyte);
System.out.println("Encrypted Message" + toHexString(ebyte));
byte[] dbyte = decrypt(raw, ebyte);
String DecryptedMessage = new String(dbyte);
System.out.println ("Decrpyted Message" + DecryptedMessage);
}
//Method : to generate symmetric keys
void generatedSymmetricKey() throws Exception {
Random r = new Random();
int num = r.nextInt(1000);
String knum = String.valueOf(num);
byte[] knumb = knum.getBytes();
```

```
System.out.println("Sym Keys :" + toHexString(knumb));
skey = getRawKey(knumb);
}
private static byte[] getRawKey(byte[] seed) throws Exception {
KeyGenerator kgen = KeyGenerator.getInstance("DES");
SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
sr.setSeed(seed);
kgen.init(56, sr);
SecretKey skey = kgen.generateKey();
raw = skey.getEncoded();
return raw;
}
//Method 3: Encryption
public static byte[] encrypt(byte[] raw, byte[] clear) throws Exception {
SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
byte[] encrypted = cipher.doFinal(clear);
return encrypted;
}
//method 4: Decryption
public static byte[] decrypt(byte[] rav, byte[] encrypted) throws Exception {
SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.DECRYPT_MODE, skeySpec);
byte[] decrypted = cipher.doFinal(encrypted);
return decrypted;
}
//Method 5: To convert byte into hex string
```

```
private String toHexString(byte[] hash) {
//Convert byte array into signum representation
BigInteger number = new BigInteger(1, hash);
//Convert message digest into hex value
StringBuilder hexString = new StringBuilder(number.toString(16));
//Fad with leading zerors
while (hexString.length()<32){
hexString.insert(0, '0');
}
return hexString.toString();
}
public static void main(String args[]) throws Exception{
DES des = new DES();
}
}
```

Write a program to implement RSA algorithm to perform encryption / decryption of a given string.

RSA algorithm:

```
// Java Program to Implement the RSA Algorithm
import java.math.*;
import java.util.*;
public class RSA {
public static void main(String args[]) {
int p, q, n, z, d = 0, e, i;
//The number to be encrypted and decrypted
int msg = 12;//plaintext
double c;
BigInteger msgback;
// 1st prime number p and q
```

```
p = 3; q = 11;
n = p + q;
z = (p-1) * (q-1);
System.out.println("The value of z = " + z);
for (e = 2; e< z; e++) {
//e is for public key exponent
if (gcd(e, z) == 1) {
break;
}
}
System.out.println("The value of e = " + e);
for (i = 0; i <= 9; i++) {
int x = 1 + (i * z);
//d is for private key exponent
if (x % e == 0) {
d = x / e;
break;
}
}
System.out.println("The value of d = " + d);
//cipher text
c = (Math.pow(msg, e)) % n;
System.out.println("Encrypted message is : " + c);
//converting int value of n to BigInteger
BigInteger N = BigInteger.valueOf(n);
//coverting float value of c to BigInteger
BigInteger C = BigDecimal.valueOf(c).toBigInteger();
msgback = (C.pow(d)).mod(N);
System.out.println("Decrypted message is : " +msgback);
```

```
}
static int gcd(int e, int z) {
if (e == 0) {
return z;
} else {
return gcd(z % e, e);
}
}
}
```

Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.

Diffie-Hellman:

```
import java.util.*;
/**
 * @author 0x4427
 */
public class DH {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
      System.out.println("Enter Prime Number 1 p: ");
      int p = sc.nextInt();
      System.out.println("Enter Prime Number 2 g: ");
      int g = sc.nextInt();

       System.out.println("Choose 1st secrest no(Alice) 'a': ");
       int a = sc.nextInt();
       System.out.println("Choose 2nd secrest no(Bob) 'b': ");
       int b = sc.nextInt();
```

```
int A = (int) Math.pow(g,a) % p;
int B = (int) Math.pow(g,b) % p;

 int S_A = (int) Math.pow(B,a) % p;
 int S_B = (int) Math.pow(A,b) % p;

 if(S_A == S_B){
       System.out.println("Alice and Bob can communicate with each other (^_^)");
          System.out.println("They share a secret no = " + S_A);
 } else {
       System.out.println("Alice and Bob cannot communicate with each other :/ :O :P :B _ ");
 }
  }
  }
```

Write a program to implement the MD5 algorithm compute the message digests.

MD5 algorithm:

```
import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class MD5 {
   public static void main(String[] args) throws NoSuchAlgorithmException {
      try {
         String s1 = "Information and Security";
         System.out.println("HashCode Gnerated for MD5 for: "+s1);
         MessageDigest md = MessageDigest.getInstance("MD5");
```

```
            byte[] hash = md.digest(s1.getBytes(StandardCharsets.UTF_8));
            System.out.println("\n" + s1 + " : " + toHexString(hash));
        } catch (NoSuchAlgorithmException e) {
            System.err.println("Exception throws for incorrect algorithm: " + e);
        }
    }
    private static String toHexString(byte[] hash) {
        BigInteger number = new BigInteger(1, hash);
        StringBuilder hexString = new StringBuilder(number.toString(16));

        while (hexString.length() < 32) {
            hexString.insert(0, '0');
        }
        return hexString.toString();
    }
}
```

MD5 algorithm:

```
import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class MD5 {
    public static void main(String[] args) throws NoSuchAlgorithmException {
        try {
            String s1 = "Information and Security";
            System.out.println("HashCode Gnerated for MD5 for: "+s1);
            MessageDigest md = MessageDigest.getInstance("MD5");
```

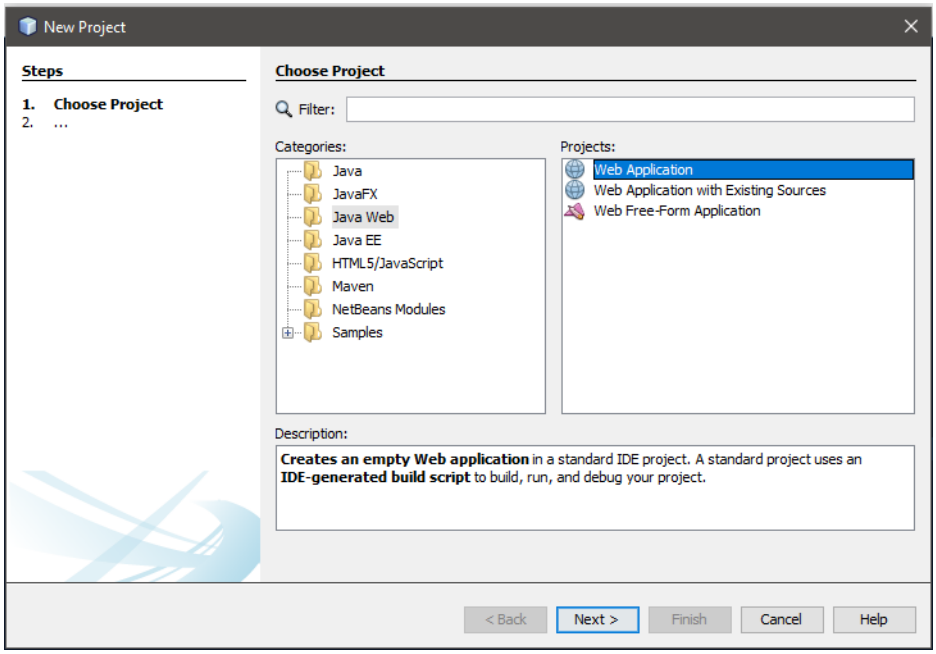
```
            byte[] hash = md.digest(s1.getBytes(StandardCharsets.UTF_8));
            System.out.println("\n" + s1 + " : " + toHexString(hash));
        } catch (NoSuchAlgorithmException e) {
            System.err.println("Exception throws for incorrect algorithm: " + e);
        }
    }
    private static String toHexString(byte[] hash) {
        BigInteger number = new BigInteger(1, hash);
        StringBuilder hexString = new StringBuilder(number.toString(16));

        while (hexString.length() < 32) {
            hexString.insert(0, '0');
        }
        return hexString.toString();
    }
}
```

Practical – 01

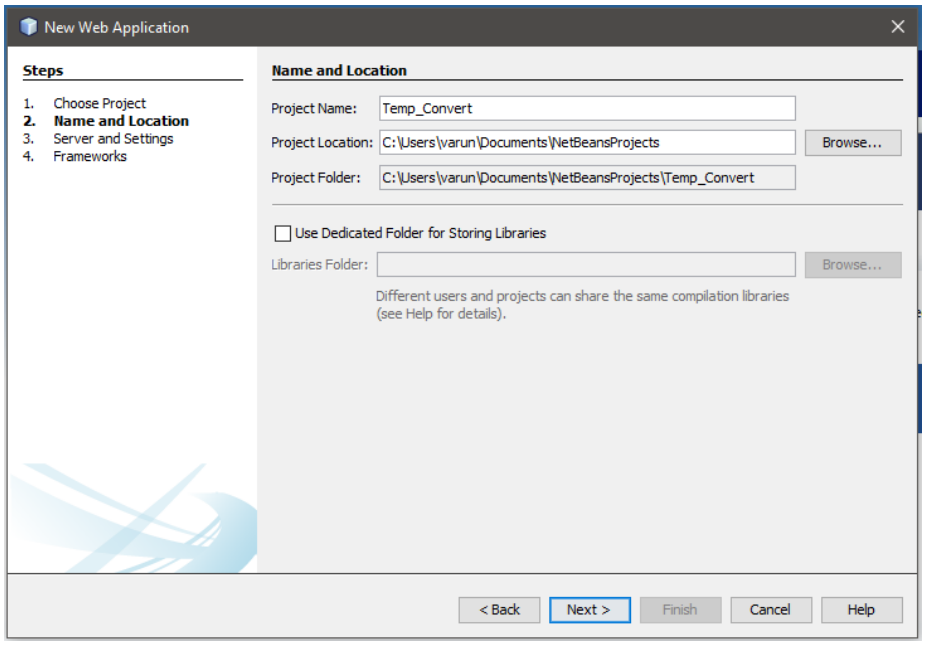
Aim: Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice a versa.

Output:

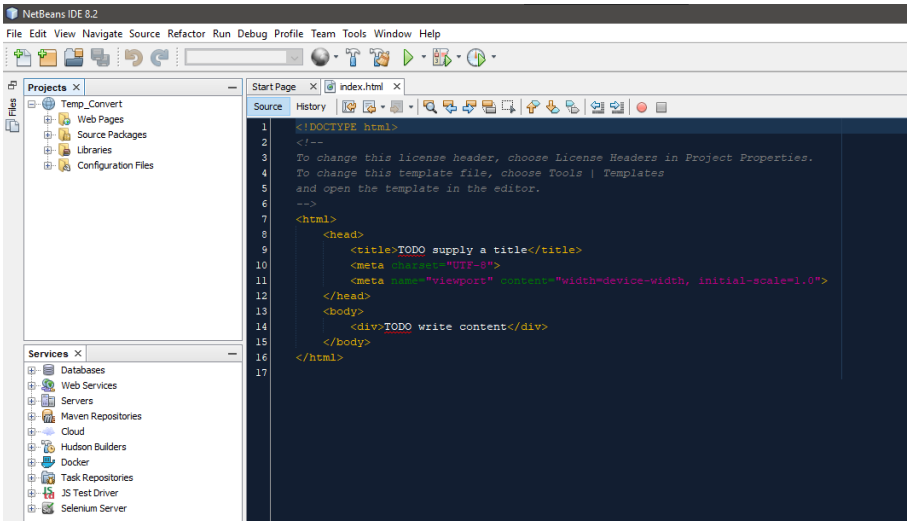
Step 1: Go to File and select New Project. Select Java Web in categories and Web Application in Projects. Click on Next to create web-based project.



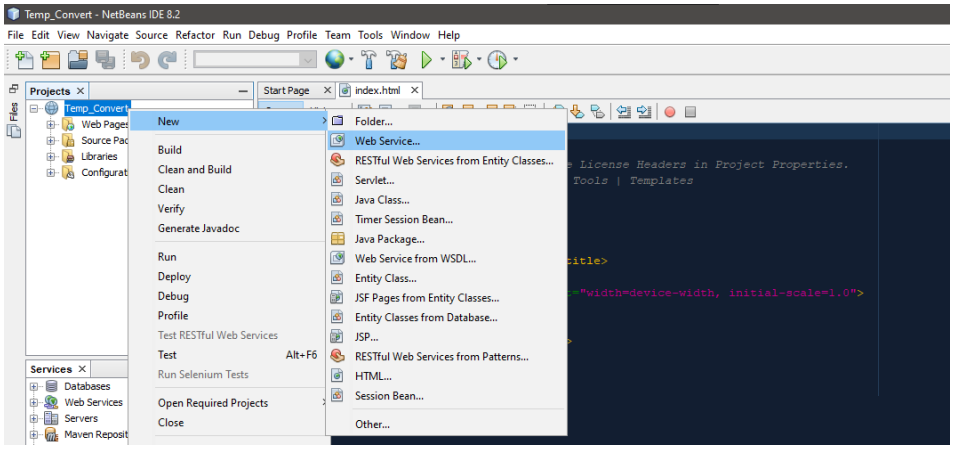
Step 2: Enter a project name whatever you want and then click on Next. On next page click Finish.



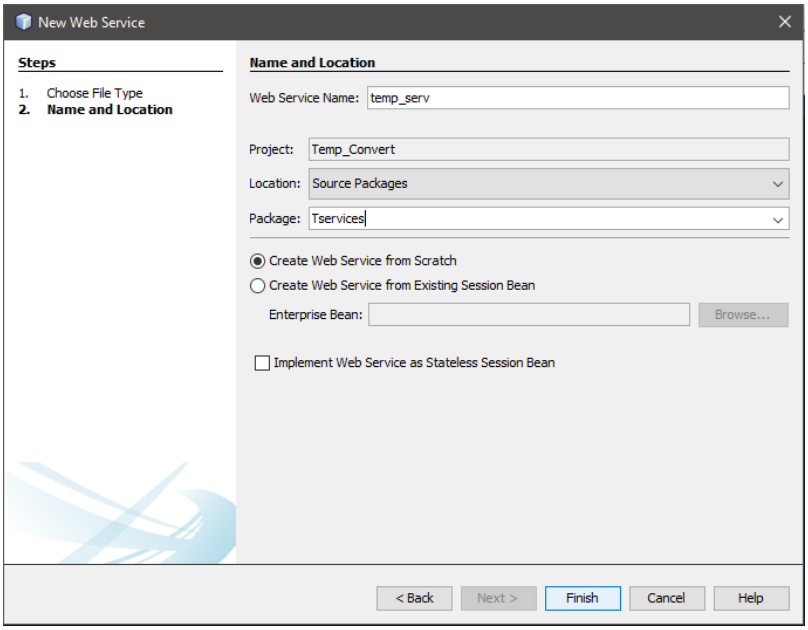
Step 3: After completion of project creation process a window will appear like below.



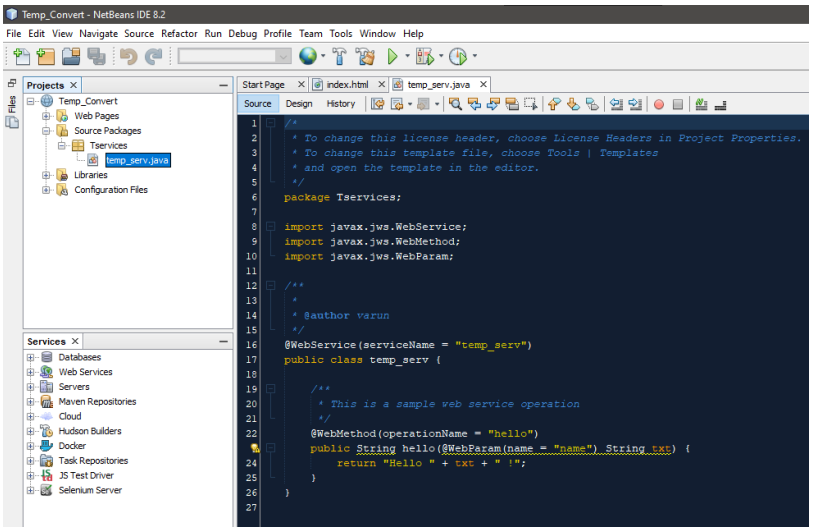
Step 4: Create web service. Right click on Project -> New -> Web Service



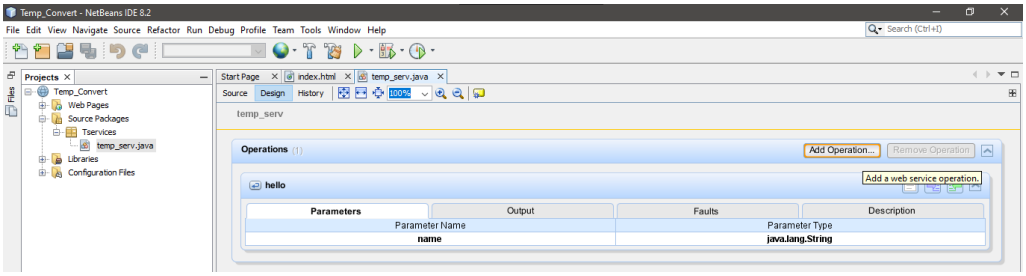
Step 5: Enter a Web Service Name and package name and then click on Finish to create a Web Service.



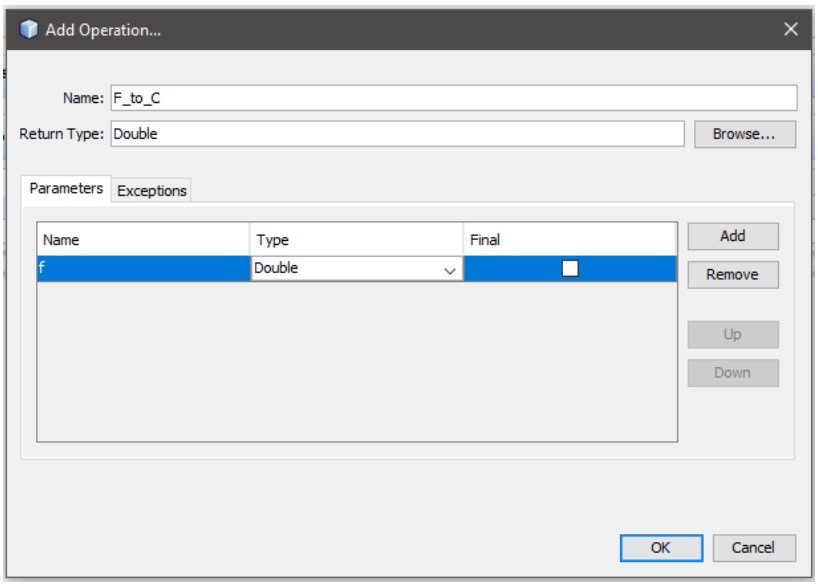
Step 6: As you can see in following pic; In Source Packages there is a package Service which contains the service file temp_serv.java. Open this file by double clicking on it, so that we can add two operation that will convert temperature from Celsius to Fahrenheit and vice-versa. Go to design mode by clicking on Design.



Step 7: Click on Add Operation to add operation.



Step 8: Give Operation name F_to_C and return type as Double. So, this method will return value in Double data type. After that click on Add button to give parameters for method. Give its name as f and type as Double and then click on OK button. Your one operation is now successfully created.



Step 9: Repeat step 7 & 8 to create second operation. But this time replace some above entered data with following data: F_to_C -> C_to_F f -> c

Step 10: Now go to source mode by clicking on Source view and delete the hello segment of the code. Because it is default operation and we don't need this.

Step 11: Now replace the selected area with following code to convert Fahrenheit to Celsius.

Double c = (f – 32) * 1.8;

return c;

```
12  /**
13   *
14   * @author varun
15   */
16  @WebService(serviceName = "temp_serv")
17  public class temp_serv {
18
19
20      /**
21       * Web service operation
22       */
23      @WebMethod(operationName = "F_to_C")
        public Double F_to_C(@WebParam(name = "f") Double f) {
            Double c = (f-32)*1.8;
26          return c;
27      }
28
```

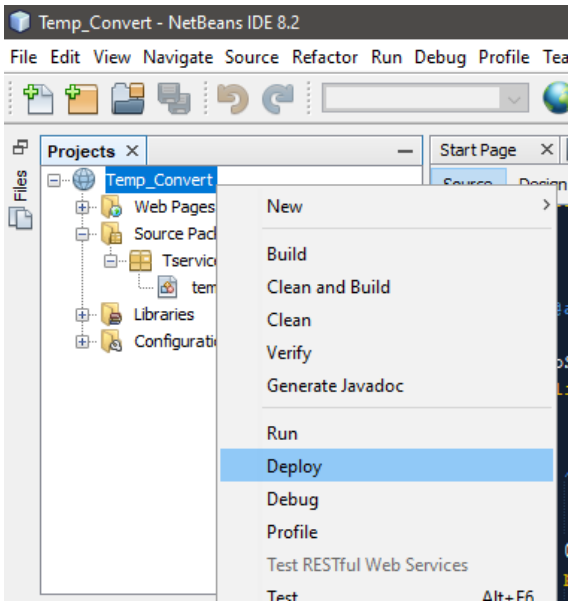
Step 12: Now replace the selected area with following code to convert Celsius to Fahrenheit and then press Ctrl+S to save.

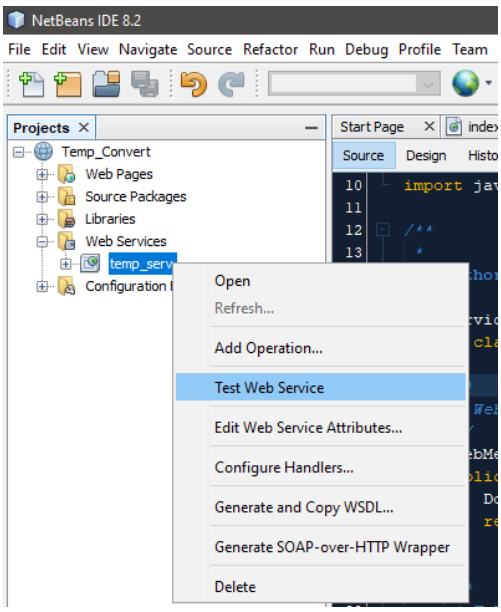
Double f = (c*1.8)+32;

return f;

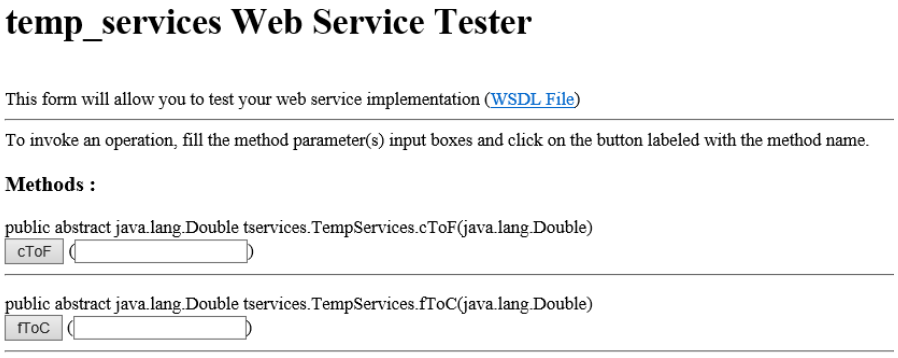
```
20      /**
21       * Web service operation
22       */
23      @WebMethod(operationName = "F_to_C")
        public Double F_to_C(@WebParam(name = "f") Double f) {
25          Double c = (f-32)*1.8;
26          return c;
27      }
28
29      /**
30       * Web service operation
31       */
32      @WebMethod(operationName = "C_to_F")
        public Double C_to_F(@WebParam(name = "c") Double c) {
            //TODO write your implementation code here:
35          return null;
36      }
37  }
38
```

Step 13: Now right click on project name and click on Deploy to deploy your project. To test your web service, do the following process as shown in the second picture.

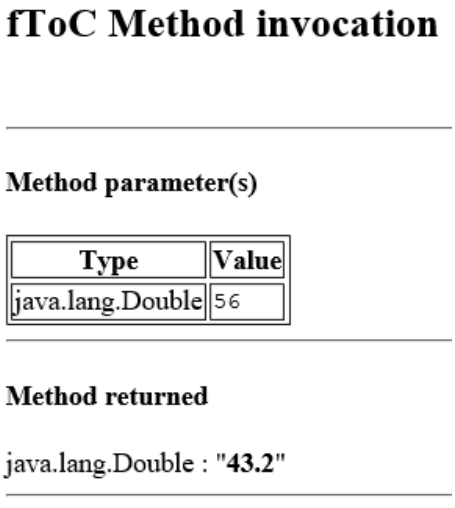




Step 14: Following window will be opened in the browser. Now if you will enter a numeric data into first box and you will click on first button it will convert the entered data into Celsius.



Step 15: Selected value is in Celsius of 56.

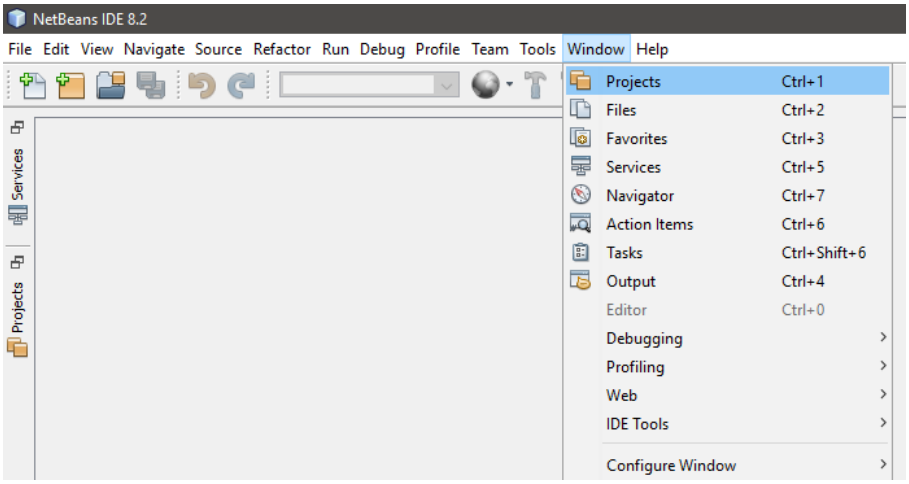


Practical – 02

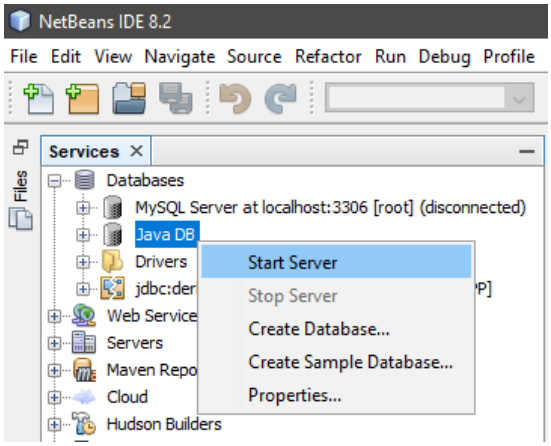
Aim: Write a program to implement the operation can receive request and will return a response in two ways. a) One - Way operation b) Request –Response

Output:

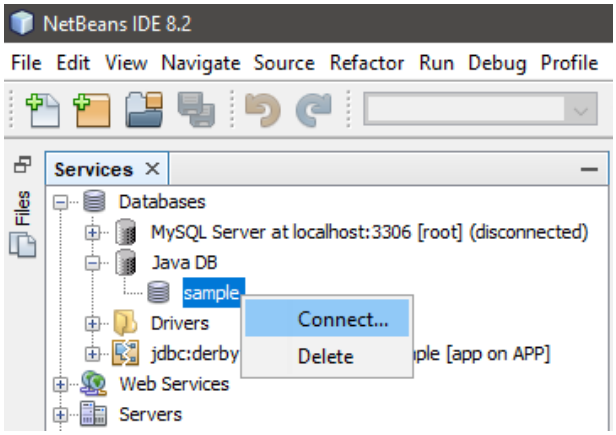
Step 1: Click on Window menu and click on Projects, Files & Services to open it.



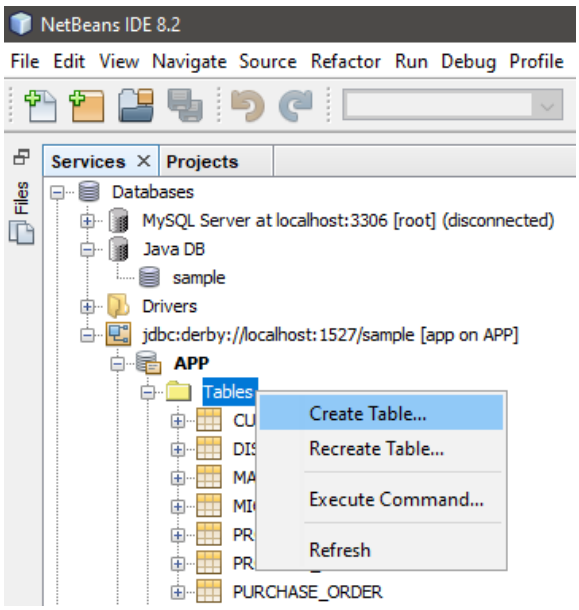
Step 2: Right click on Java DB and then click on Start Server to start the server.



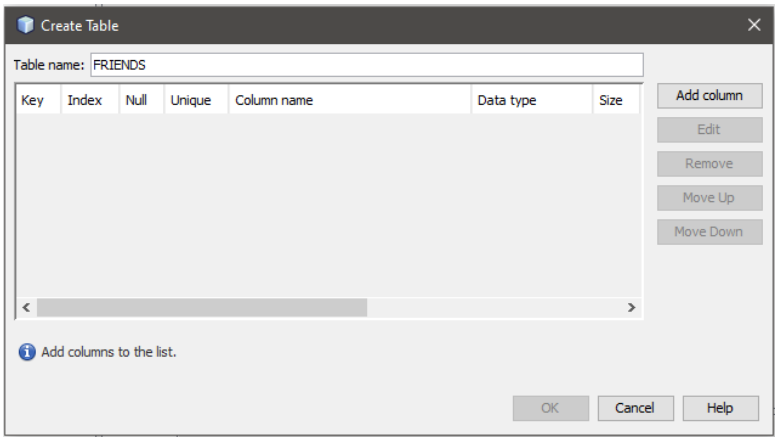
Step 3: Now expand Java DB and right click on sample and then click on connect to connect the sample database with server.



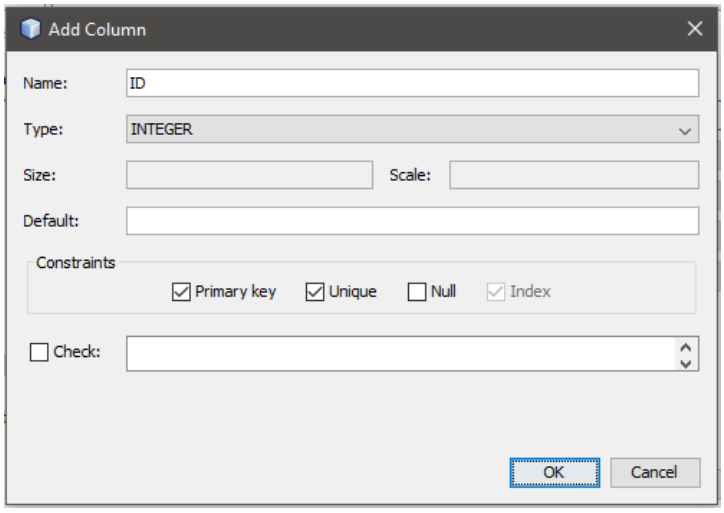
Step 4: Now we are going to create a table in default database sample.
Right click on Table -> Create Table



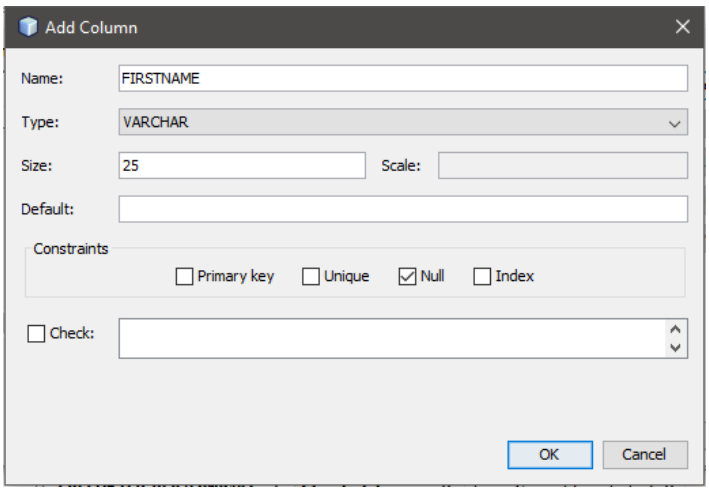
Step 5: Give table name as FRIENDS.



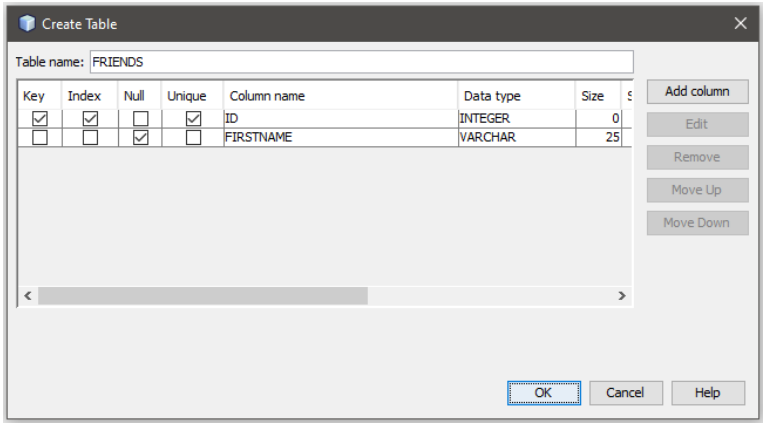
Step 6: Now click on Add column button to add columns in table. Enter details as shown in the picture below and select Primary key. After that click on OK button.



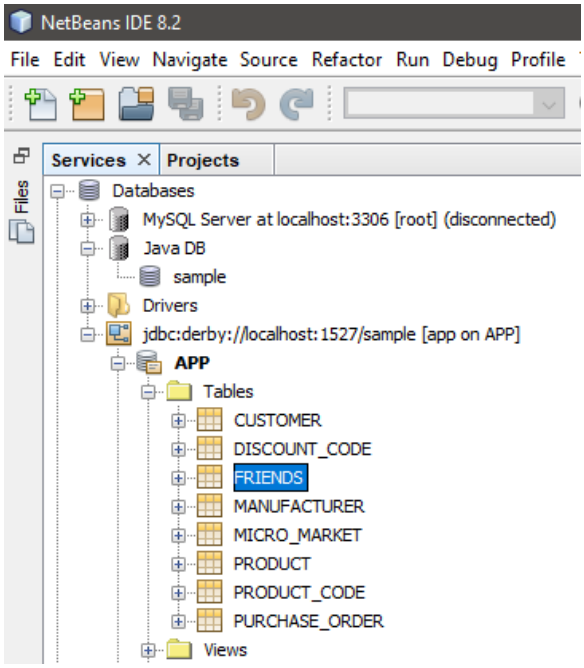
Step 7: Now add second column with following detail. But don't select primary and click on OK button.



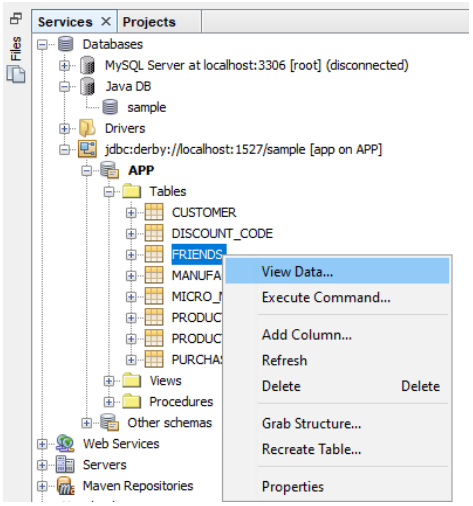
Step 8: Now click on OK button.



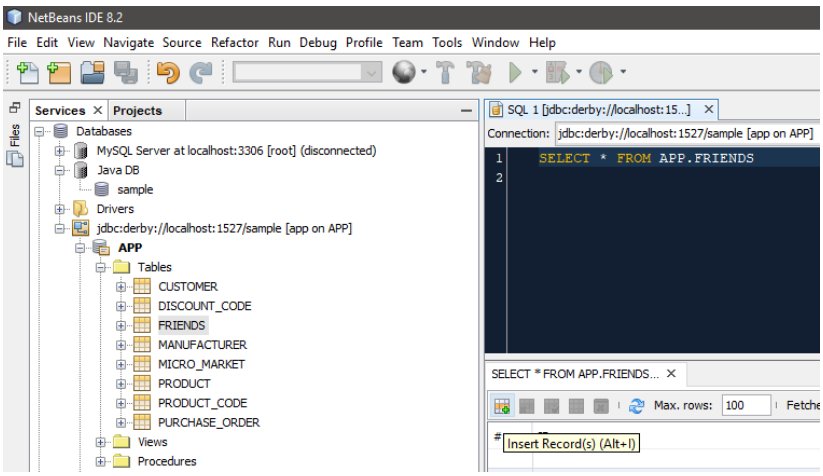
Step 9: Now you can see a table with name FRIENDS in the table.



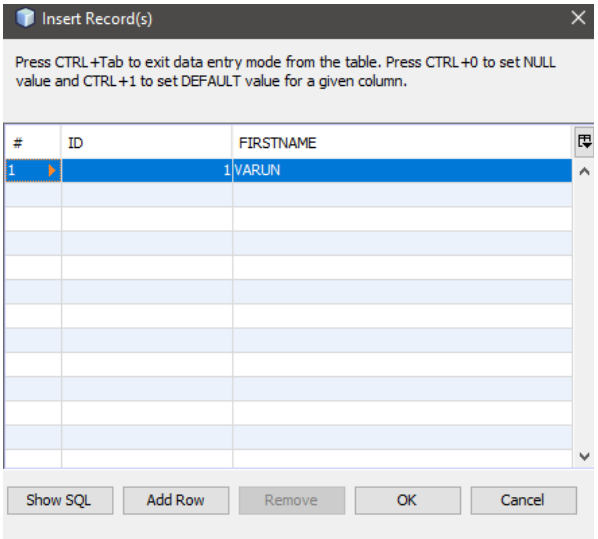
Step 10: Right click on FRIENDS to view and add records into it.



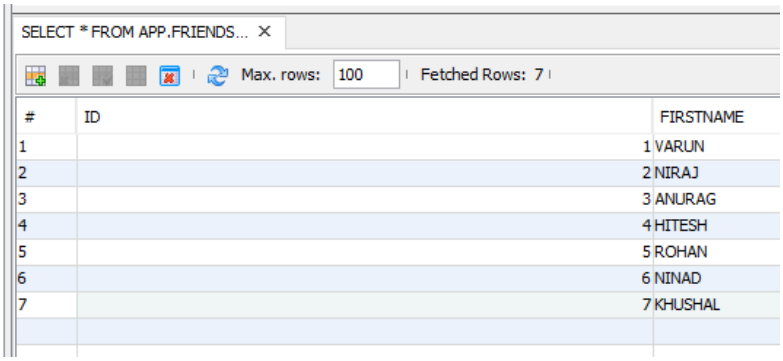
Step 11: Now click on the leftmost icon in second panel to insert some record.



Step 12: Insert a record and then click on Add Row button to insert more record. After that click on OK button to finish.

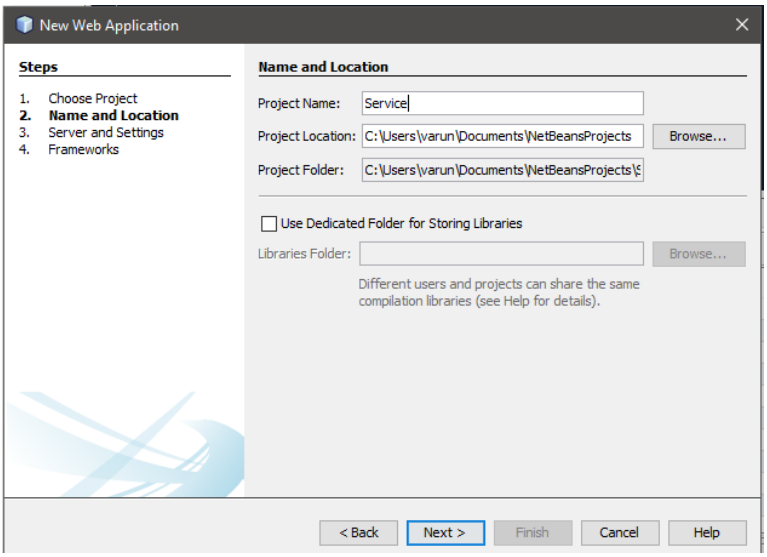


Step 13: As you can see, I have entered 7 records.

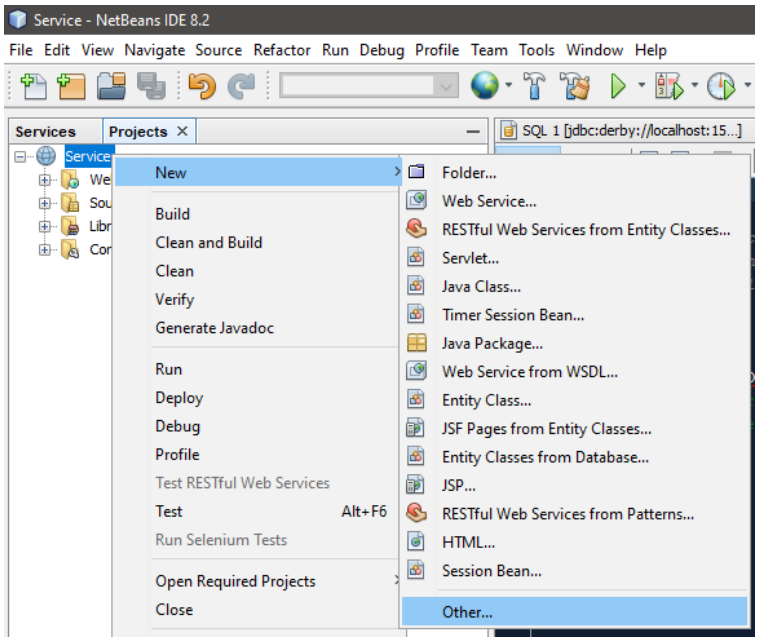


#	ID	FIRSTNAME
1	1	VARUN
2	2	NIRAJ
3	3	ANURAG
4	4	HITESH
5	5	ROHAN
6	6	NINAD
7	7	KHUSHAL

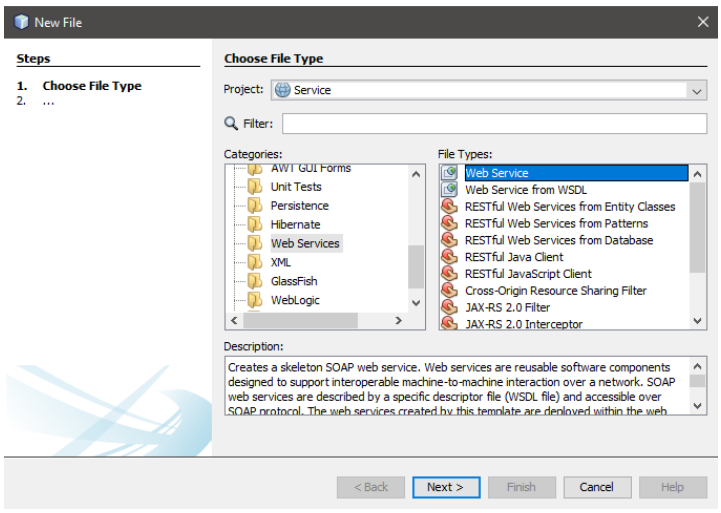
Step 14. Create a Web Application with name Service**.**



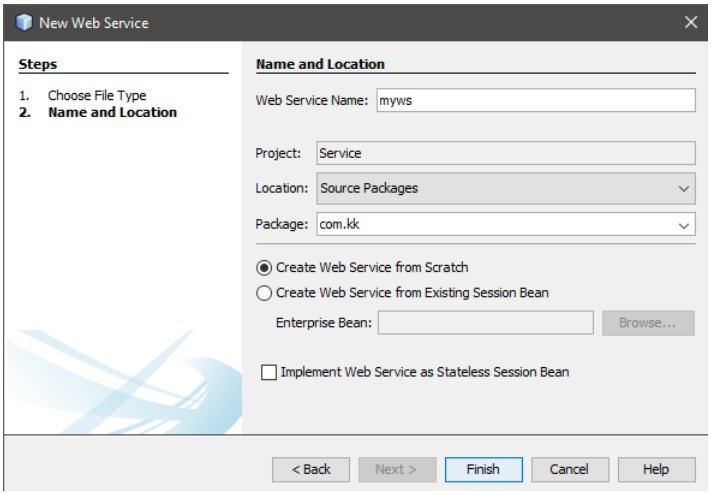
Step 15: Creating a web service. Right click on Service web application -> New -> Other



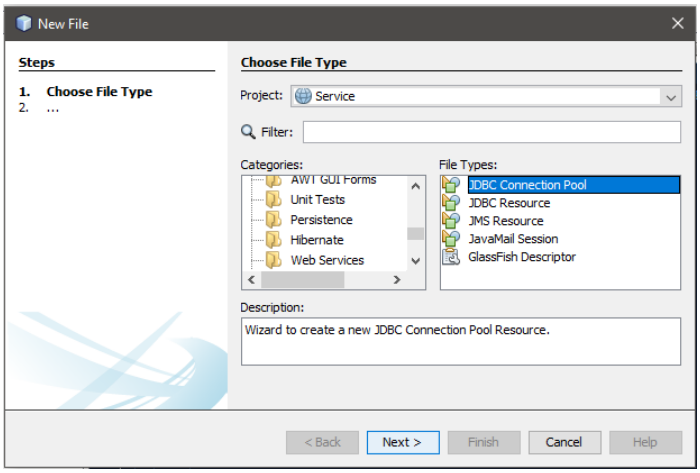
Step 16: In Categories select Web Services and in File Types select Web Service and then click on Next button.



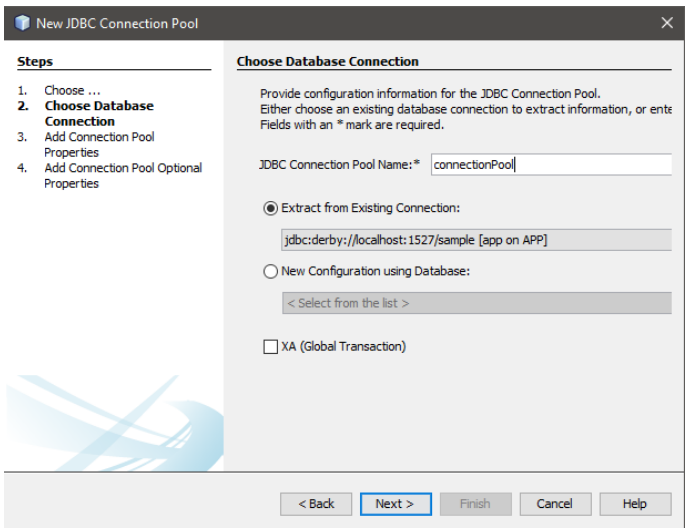
Step 17: In Web Service Name and Package, enter myws and com.kk respectively. After that click on Finish button.

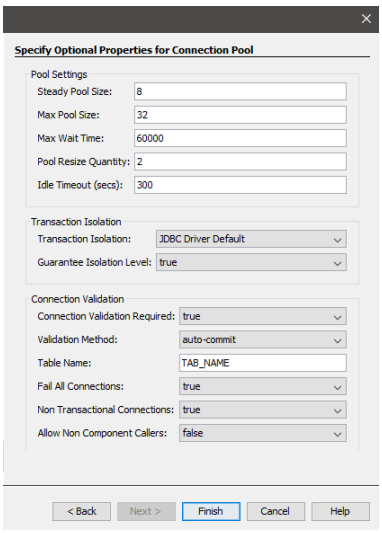


Step 18: Creating connection for database. Right click on Service web application -> New -> Select Other (Repeat Step 15). In Categories select GlassFish and in File Types select JDBC Connection Pool. After that click on Next button.

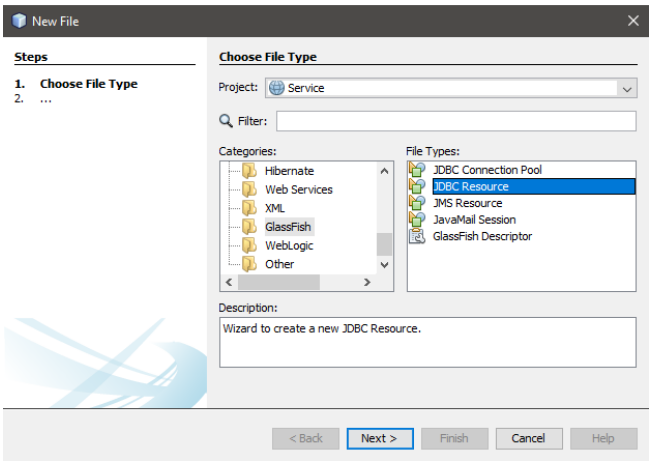


Step 19: Leave everything as it is and click on Next button. Follow same instructions for next step and finally click on Finish button.

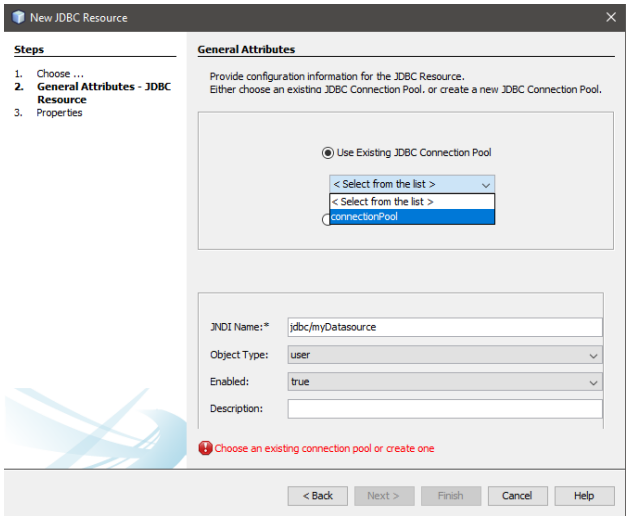




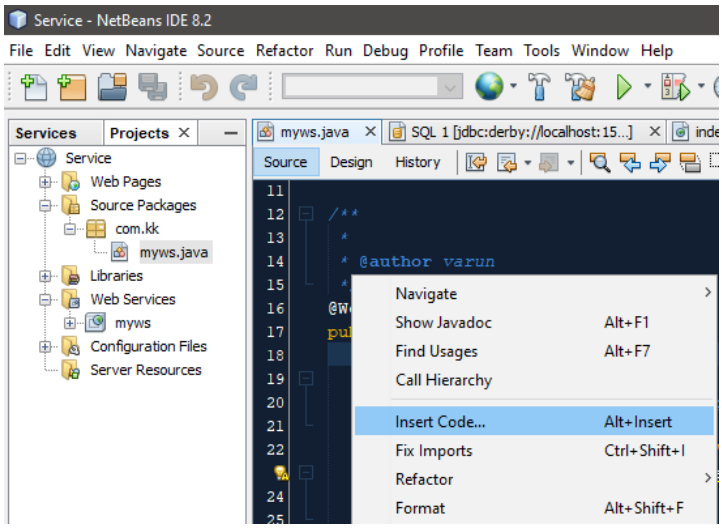
Step 20: Right click on Service web application -> New -> Select Other (Repeat Step 15). In categories select GlassFish and in File Types select JDBC resource. Click on Next.

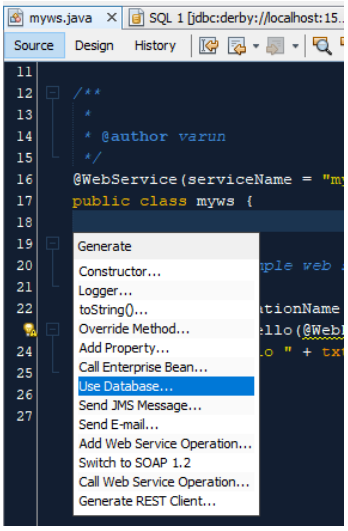


Step 21: Select connectionPool and in JNDI Name enter jdbc/mydb. After that click on Finish.

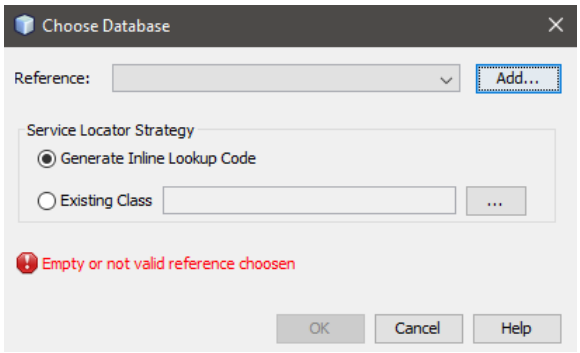


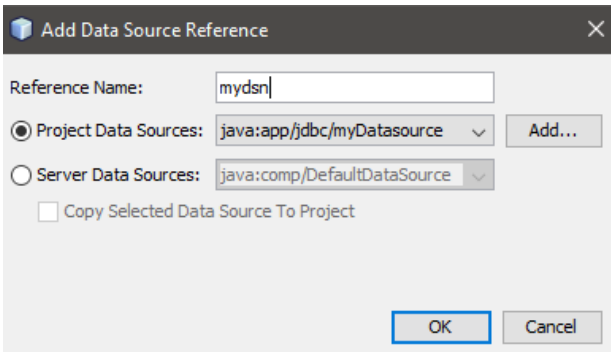
Step 22: Now open myws.java file by double clicking on it**.** In myws.java file right click at line number 18 and select Insert Code and select Data Source.



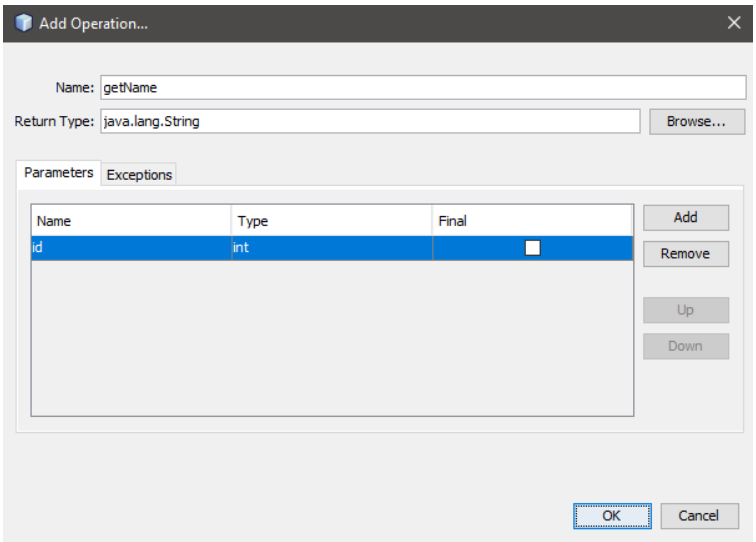


Step 23: Click on Add button. In Reference Name enter mydsn and then click OK in both the panels.

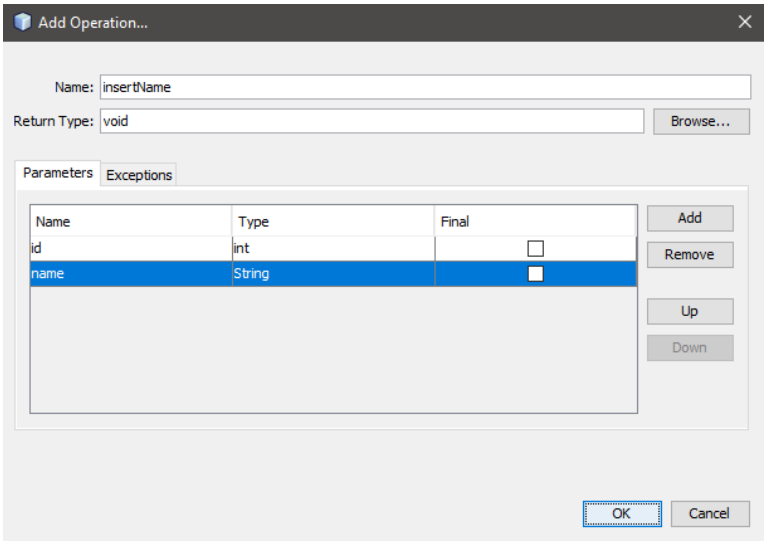




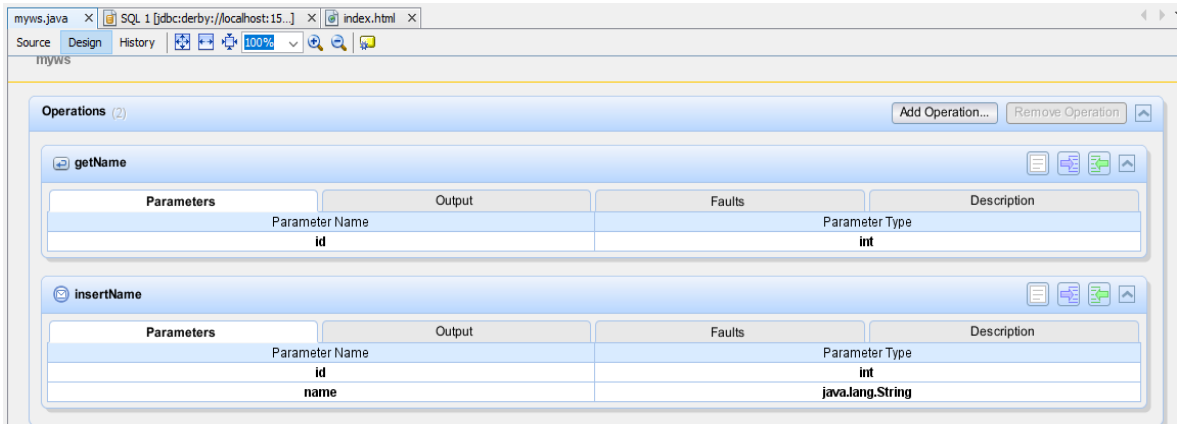
Step 24: Now go in Design mode. Select hello window and then click on Remove Operation to remove this method as we don't need this. Click on Add Operation. Enter Name as getName and click on Add to add a parameter. In parameters give Name as id and Type as int. After that click on OK.



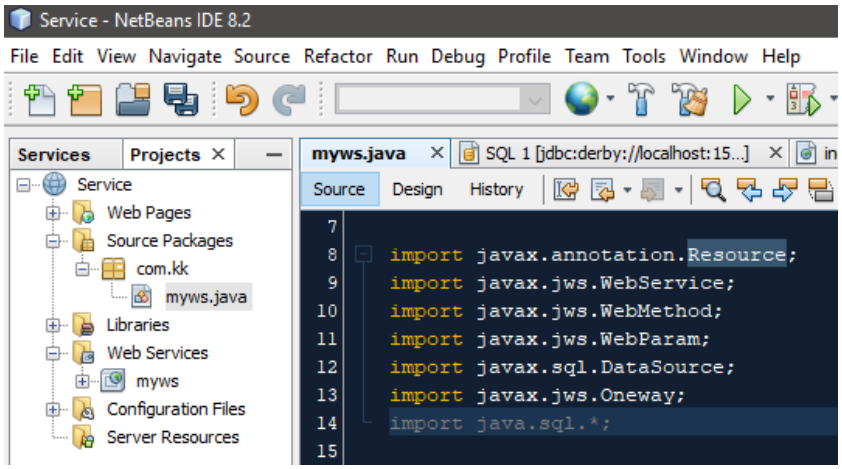
Step 25: Now add one more operation by following step number 24. But this time give Name as insertName and Return Type void. Add two parameters by clicking two times on Add button and change the Name and Type as below. After that click on OK button.



Step 26: As you can see, two operations are created with name getName and insertName.



Step 27: Now go into the Source mode of myws.java and add the selected statement to import everything from SQL.



Step 28: Now replace the contents of getName method with following code:

try { Connection c=mydsn.getConnection();

PreparedStatement ps=c.prepareStatement("select * from friends where id=?");

ps.setInt(1, id);

ResultSet r=ps.executeQuery();

if(r.next())

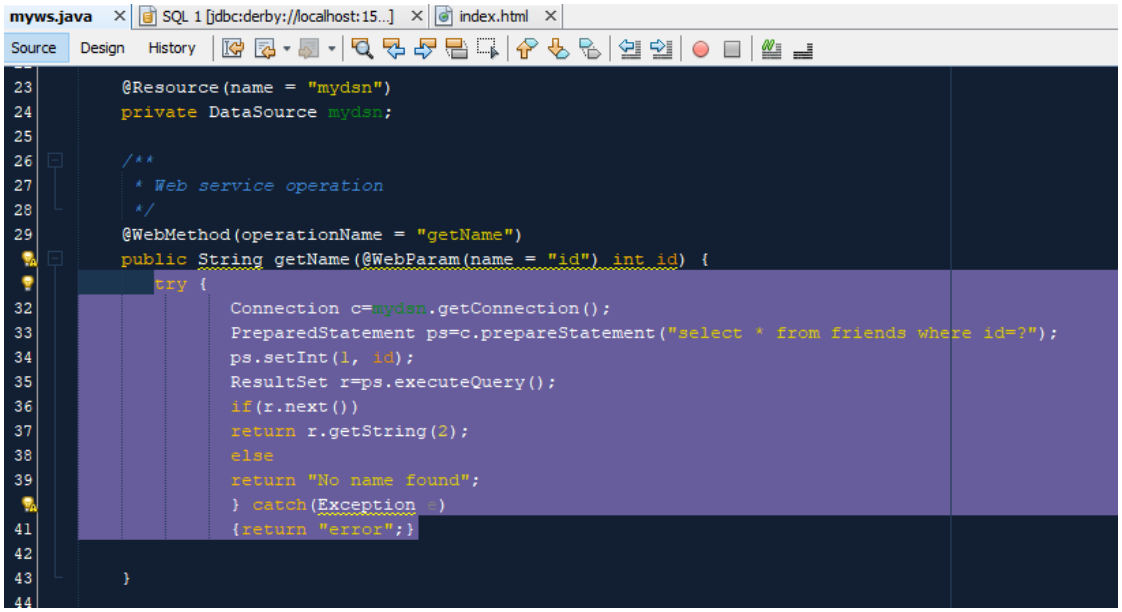
return r.getString(2);

else

return "No name found";

} catch(Exception e)

{return "error";}



Step 29: Now replace the contents of insertName method with following code and after that press Ctrl+S to save the changes.

try { Connection c=mydsn.getConnection();

PreparedStatement ps=c.prepareStatement("insert into friends(id,firstname)values(?,?)");

ps.setInt(1, id);

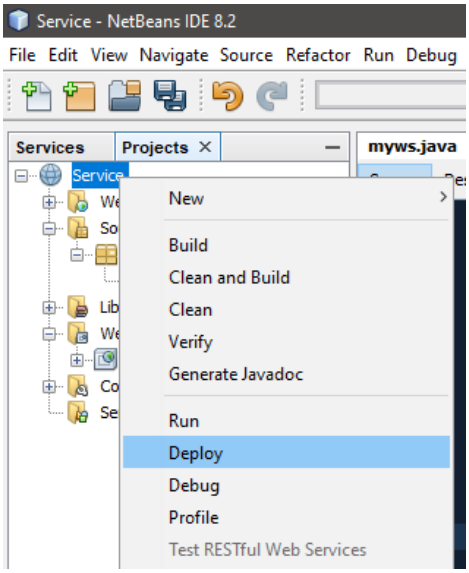
ps.setString(2, name);

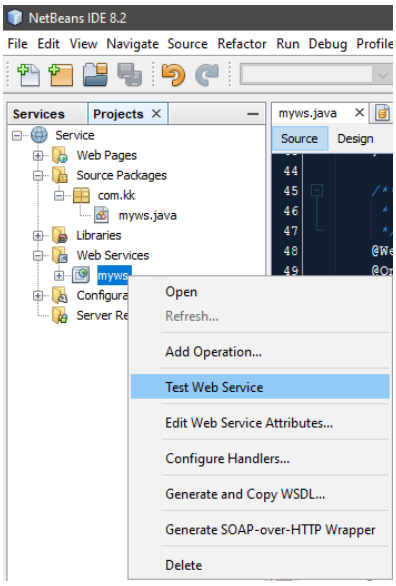
ps.executeUpdate();

}catch(Exception e){}

```
/**
 * Web service operation
 */
@WebMethod(operationName = "insertName")
@Oneway
public void insertName(@WebParam(name = "id") int id, @WebParam(name = "name") String name) {

    try {
            Connection c=mydsn.getConnection();
            PreparedStatement ps=c.prepareStatement("insert into friends(id,firstname)values(?,?)");
            ps.setInt(1, id);
            ps.setString(2, name);
            ps.executeUpdate();
            }catch(Exception e){}

    }
}
```

Step 30: Now deploy the Service web application by right clicking on it. After deploying test your web service through the following process as shown in the second picture.





Step 31: A window will open in browser like below. Now enter an ID in textbox which you have inserted in your database table and click on getName button. It will give you the first name of particular entered ID.

myws Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String com.kk.Myws.getName(int)

getName (1)

public abstract void com.kk.Myws.insertName(int,java.lang.String)

insertName (,)

getName Method invocation

Method parameter(s)

Type	Value
int	1

Method returned

java.lang.String : "**VARUN**"

Step 32: To do one-way operation, Enter ID and Firstname in the textboxes and then click on insertName button. It will insert the data in table. But make sure, you are not entering the ID which is already created in the table.

myws Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String com.kk.Myws.getName(int)
getName ( )

public abstract void com.kk.Myws.insertName(int,java.lang.String)
insertName (8 , SHIV)

Step 33: Now you can check the table record to ensure that the operation is successfully done. Click on Refresh Records to update the inserted ID and FIRSTNAME.

SELECT * FROM APP.FRIENDS... ×

Max. rows: 100 | Fetched Rows: 8 | Refresh Records

#	ID	FIRSTNAME
1	1	VARUN
2	2	NIRAJ
3	3	ANURAG
4	4	HITESH
5	5	ROHAN
6	6	NINAD
7	7	KHUSHAL
8	8	SHIV

Practical – 03

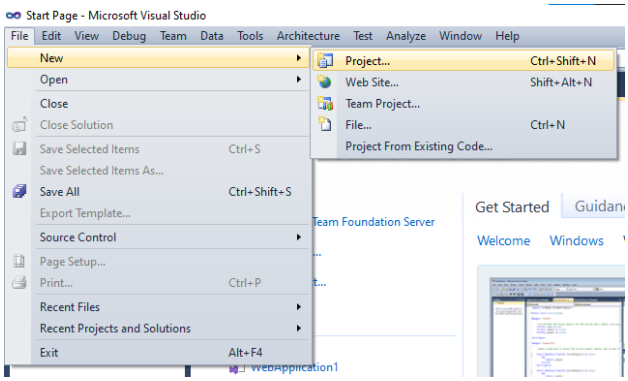
Aim: Develop client which consumes web services developed in different platform.

Requirement: Visual Studio Community 2017 or 2010 Ultimate, JDK Version 8 or latest.

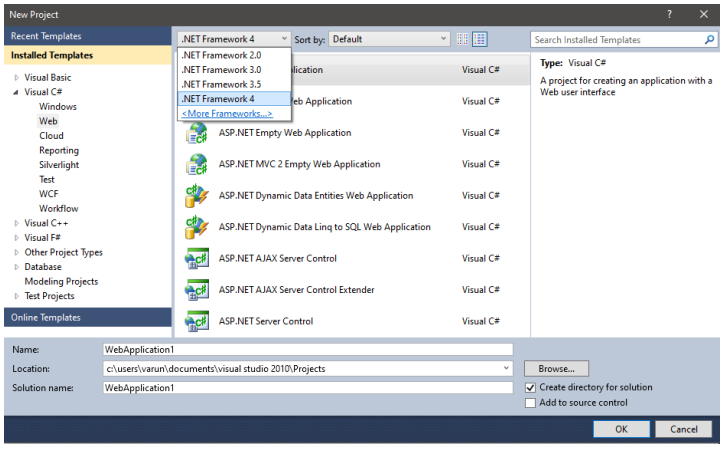
In this practical we are creating Web Service in Visual Studio ant then we will consume it in NetBeans.

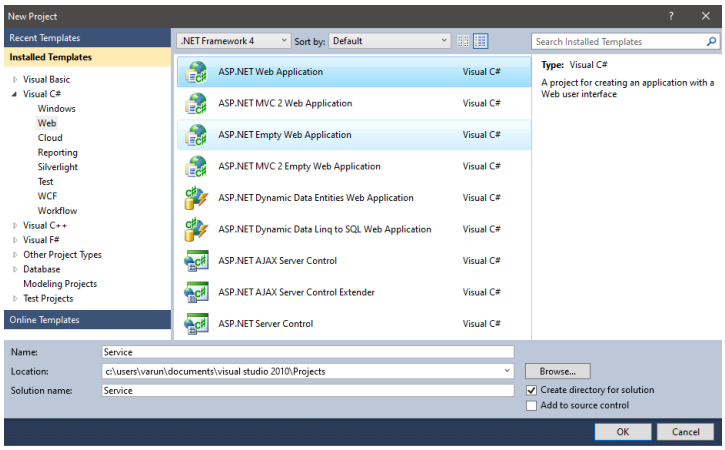
Output:

Step 1: Open Visual Studio IDE and click on File. File -> New -> Project

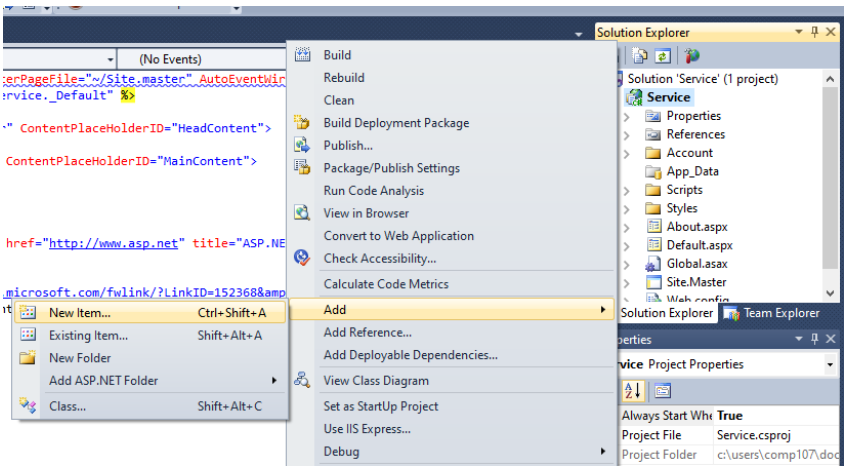


Step 2: Select .NET Framework 4.0. Then go to Visual C# -> Web -> ASP.NET Web Application. Give name as Service and click on OK button.

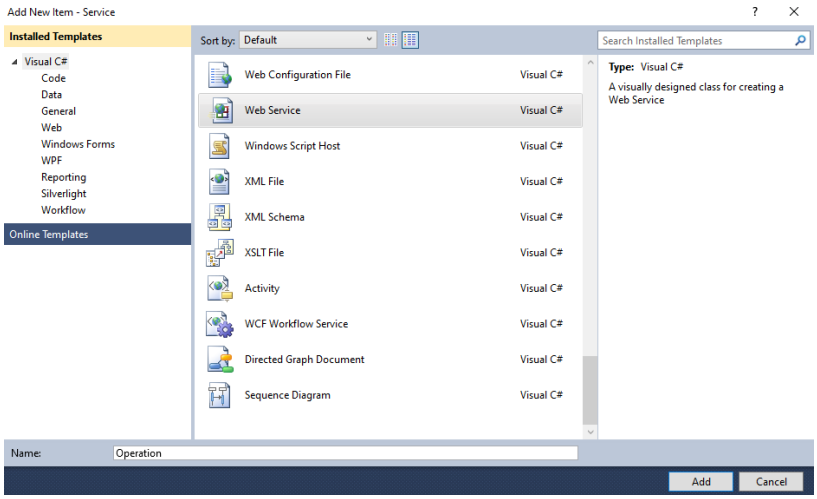




Step 3: Now you can see, on right side in Solution Explorer Service project is created. Right click on Service -> Add -> New Item.



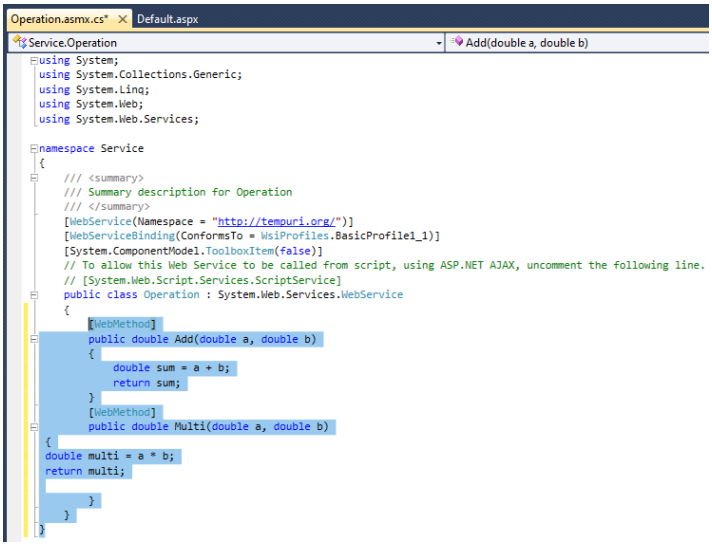
Step 4: Select Web Service and give Name as Operation. Then click on Add button.



Step 5: After clicking on Add button, Operation.asmx.cs file will open automatically otherwise open it from Solution Explorer and add the following code into class Operation.

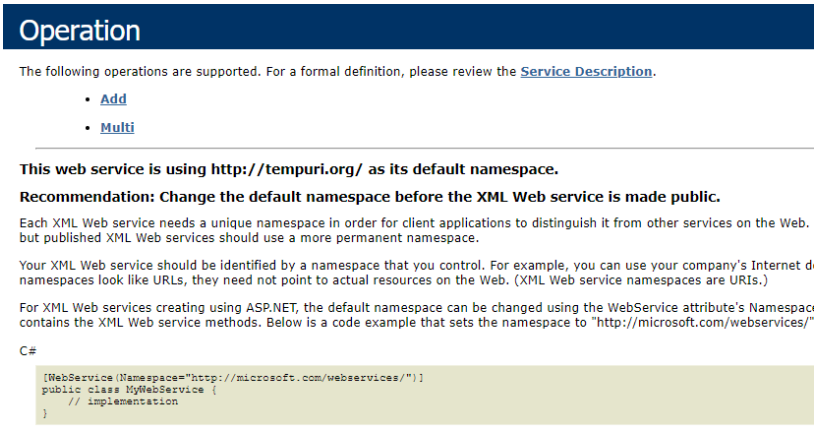
```
[WebMethod]
public double Add(double a, double b)
{ double sum = a + b;
return sum; }
[WebMethod]
public double Multi(double a, double b)
{ double multi = a * b;
return multi; }
```

After that press Ctrl+S to save the methods. Actually, we are creating two methods for Web Service. One is for addition of two numbers and second one is for multiplication of two numbers.

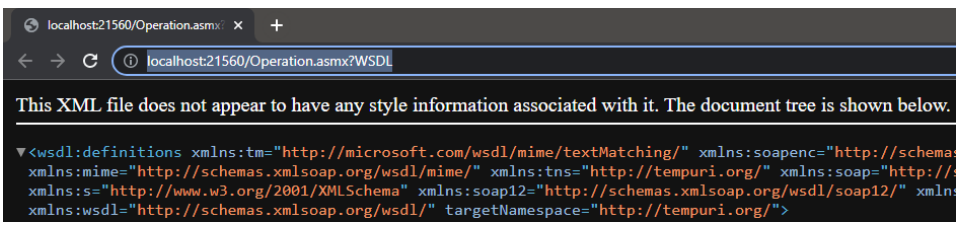


Step 6: Now run the project by clicking on green arrow button below the Window menu.

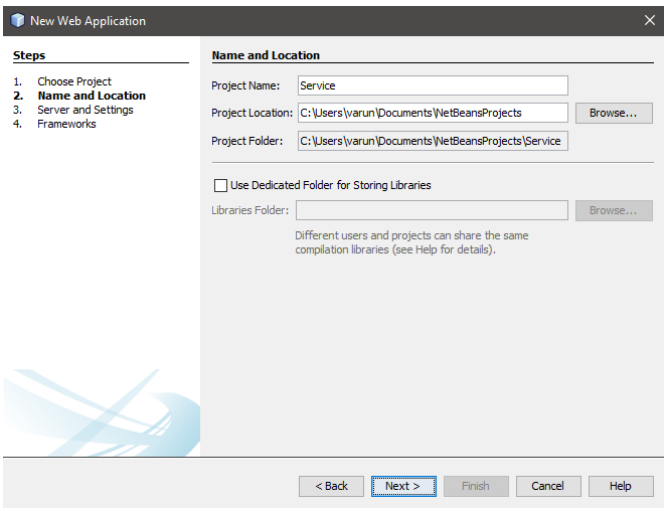
Step 7: A window will open in browser and that is our web service. You can check services by clicking on Add or Multi option. But we don't need this. Now click on Service Description option.



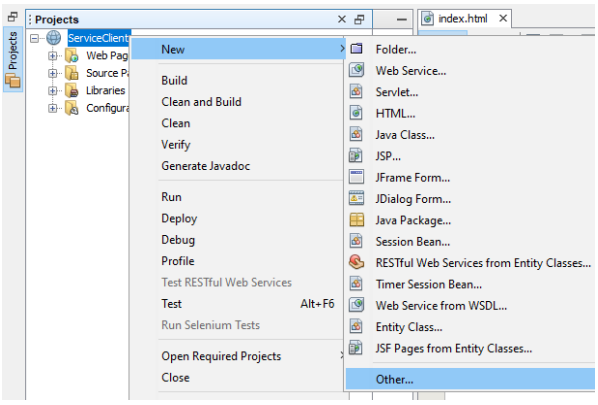
Step 8: Now a new window will open. Select the link and copy it and save it anywhere so that we can use it later. We will use this link in NetBeans to consume these services. Don't close Visual Studio and browser, just minimize it otherwise the server will stop.

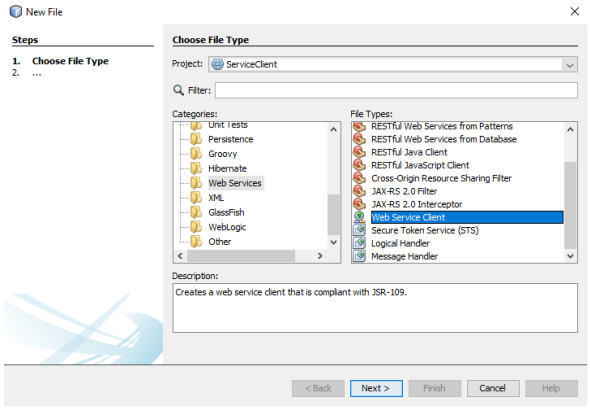


Step 9: Now open NetBeans. Create a Web Application from File -> New Project with the name ServiceClient. Then click on Next -> Finish.

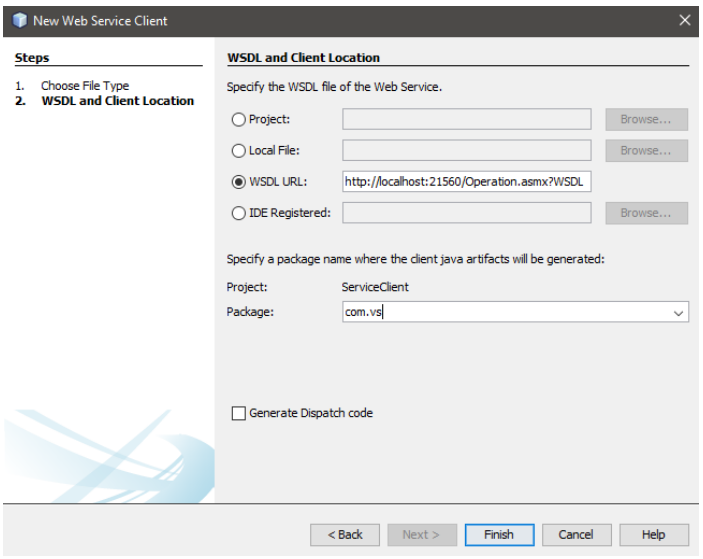


Step 10: Now create a Web Service Client. Right click on ServiceClient -> New -> Other. Select Web Services in Categories section and Web Service Client in File Types and click on Next button.

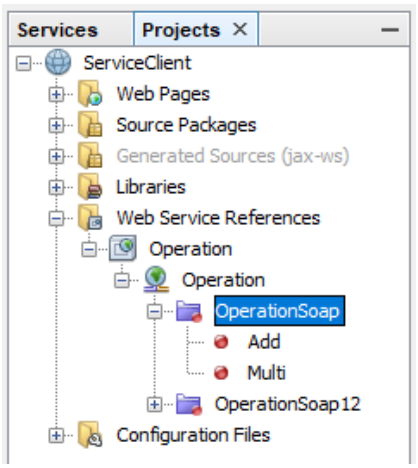


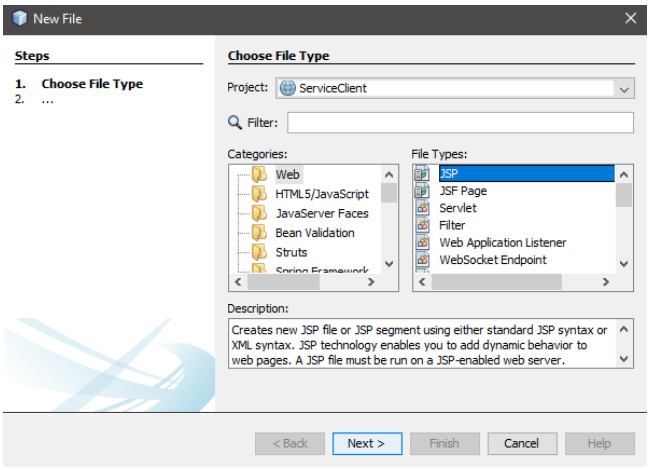


Step 11: Select WSDL URL and paste the link that you have copied from browser on run of Visual Studio. Enter package name as com.vs and click on Finish button.

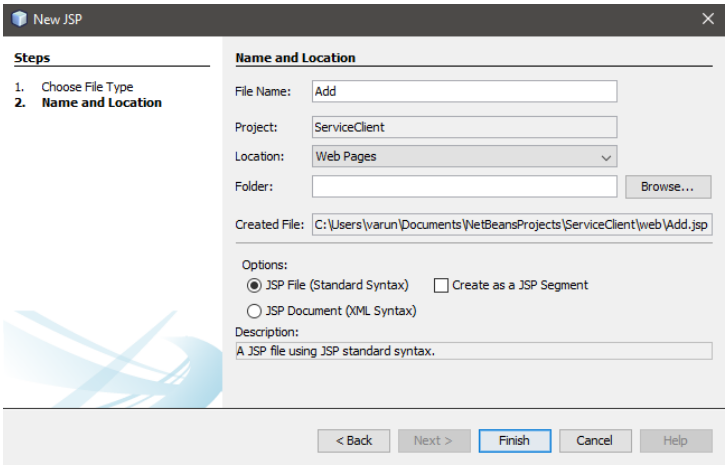


Step 12: As you can see in the left picture, we got both the service methods i.e., Add & Multi. Now create a JSP page by right clicking on ServiceClient -> New -> Other. Then Select Web in Categories section -> Select JSP and click on Next button.

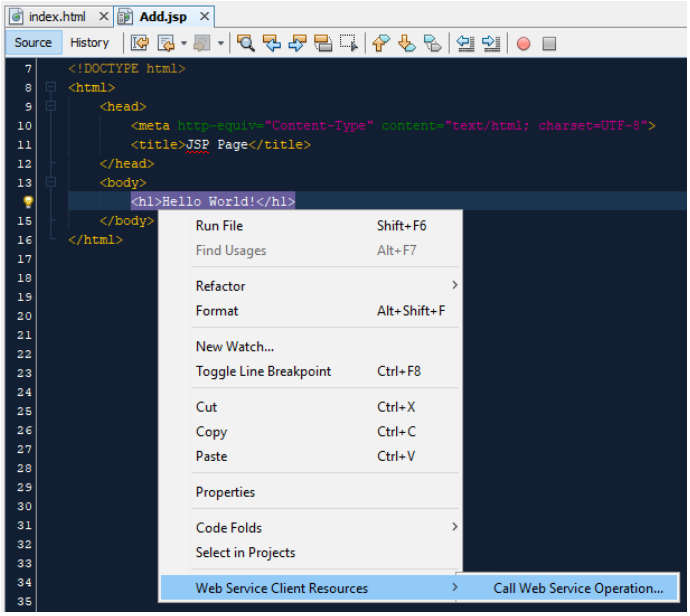




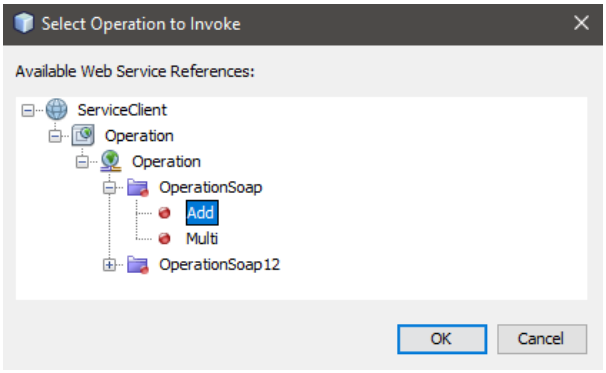
Step 13: Enter File Name as Add and click on Finish button.



Step 14: In Add.jsp file, Right click between body tag and select Call Web Service Operation.



Step 15: Expand and select Add. After selecting, click on OK button.

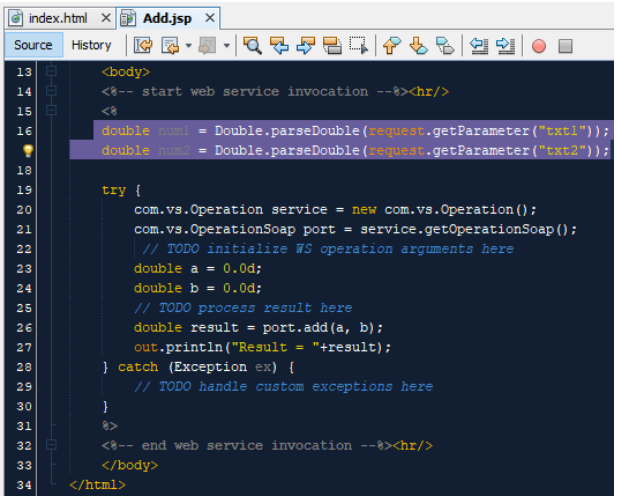


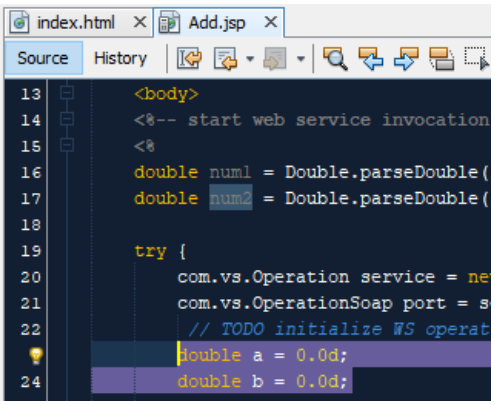
Step 16: Now add the following code outside of try block:

double num1 = Double.parseDouble(request.getParameter("txt1"));

double num2 = Double.parseDouble(request.getParameter("txt2"));

And pass num1 & num2 to a & b variables respectively. Then press Ctrl+S to save this code.





Step 17: Now open index.html file of ServiceClient project and replace the contents of body tag with following code. After that press Ctrl+S to save it.

<form>

<input type="text" name="txt1" placeholder="Enter First

Number">

<input type="text" name="txt2" placeholder="Enter Second

Number">

<input type="submit" formaction="Add.jsp" value="Add Numbers">

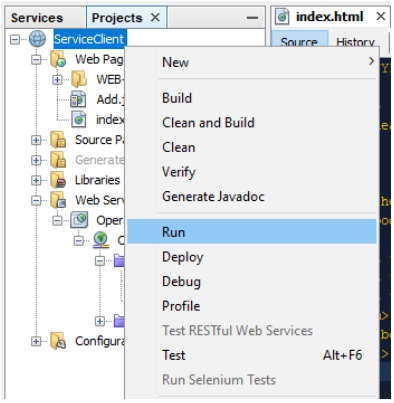
<input type="submit" formaction="Multi.jsp" value="Mulitply Numbers">

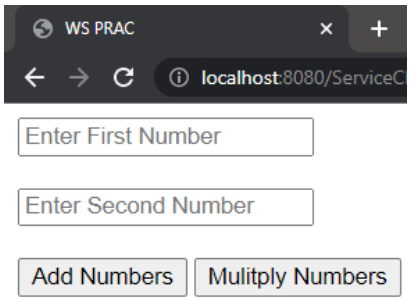
</form>

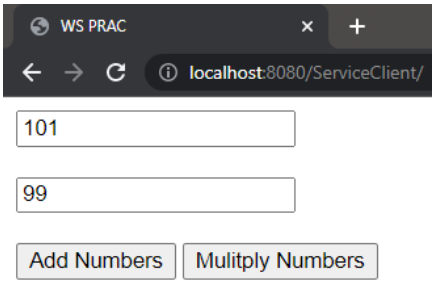
```
<!DOCTYPE html>

<html>
    <head>
        <title>WS PRAC</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form>
<input type="text" name="txt1" placeholder="Enter First Number"><br><br>
<input type="text" name="txt2" placeholder="Enter Second Number"><br><br>
<input type="submit" formaction="Add.jsp" value="Add Numbers">
<input type="submit" formaction="Multi.jsp" value="Mulitply Numbers">
        </form>
    </body>
</html>
```

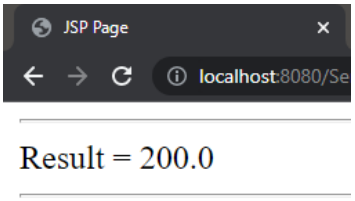
Step 18: Now for multiplying numbers, Repeat everything from Step 12 to Step 16. Give the JSP File Name as Multi. In Select Operation to Invoke select Multi. Paste the code given in Step 16 outside the try block and pass num1 & num2 to a & b variables respectively.

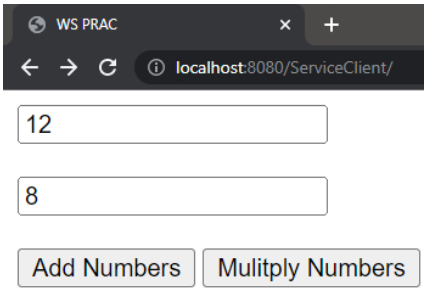
Step 19: Now run the ServerClient web application. A window will open in browser as below. Now enter two numbers and click on Add Numbers or Multiply Numbers button. Wait to get result. You will get result on a new page.

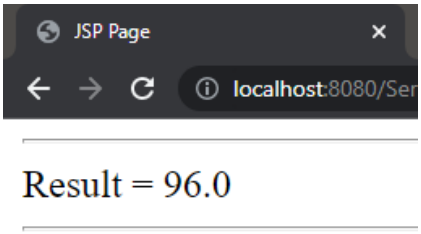










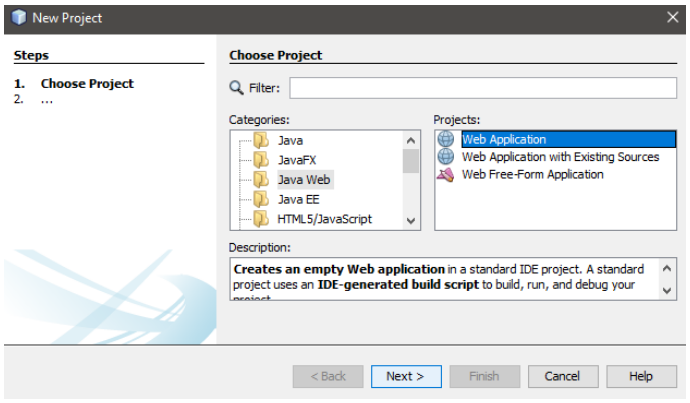


Practical – 04

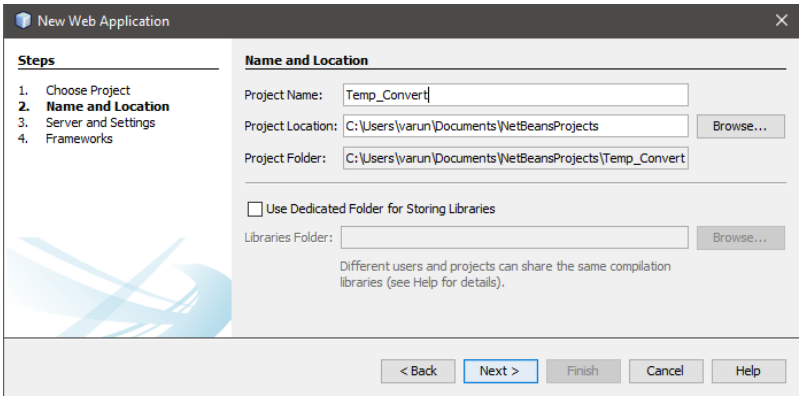
Aim: Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.

Output:

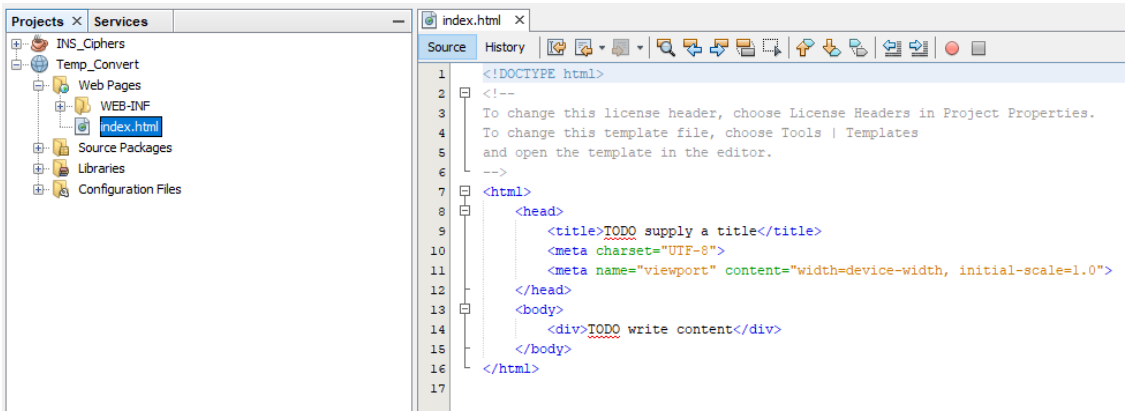
Step 1: Go to File and select New Project. Select Java Web in categories and Web Application in Projects. Click on Next to create web-based project.



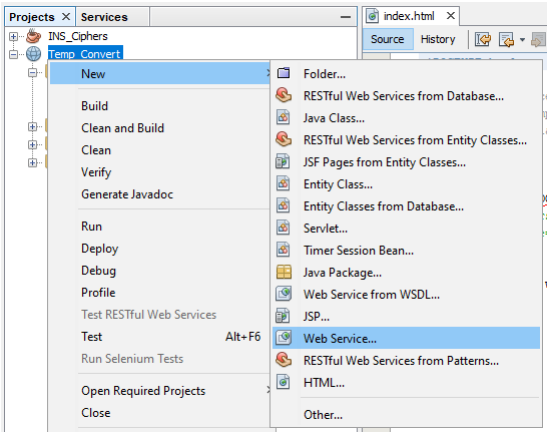
Step 2: Enter a project name whatever you want and then click on Next. On next page click Finish.



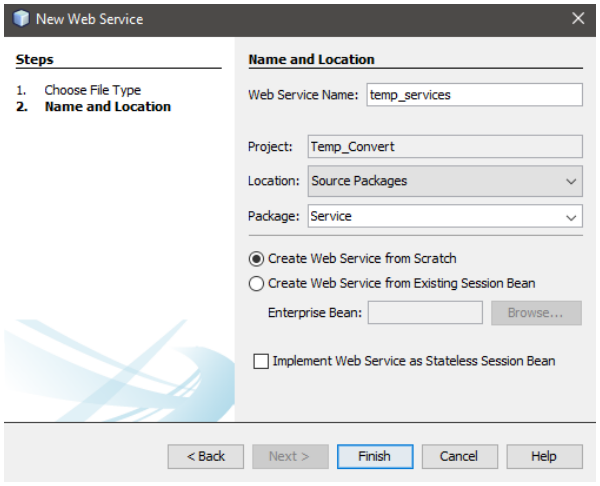
Step 3: After completion of project creation process a window will appear like below.



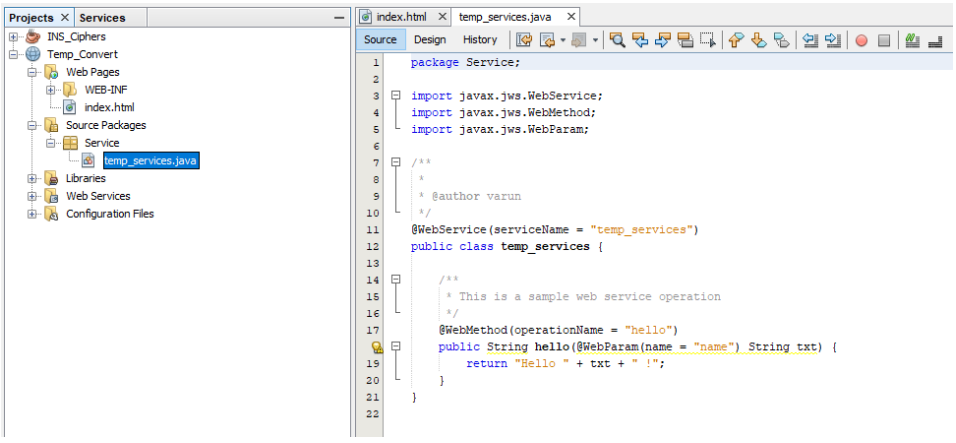
Step 4: Create web service. Right click on Project -> New -> Web Service



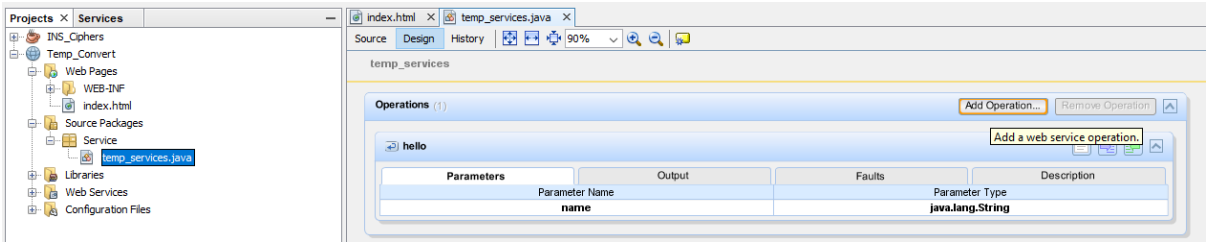
Step 5: Enter a Web Service Name and package name and then click on Finish to create a Web Service.



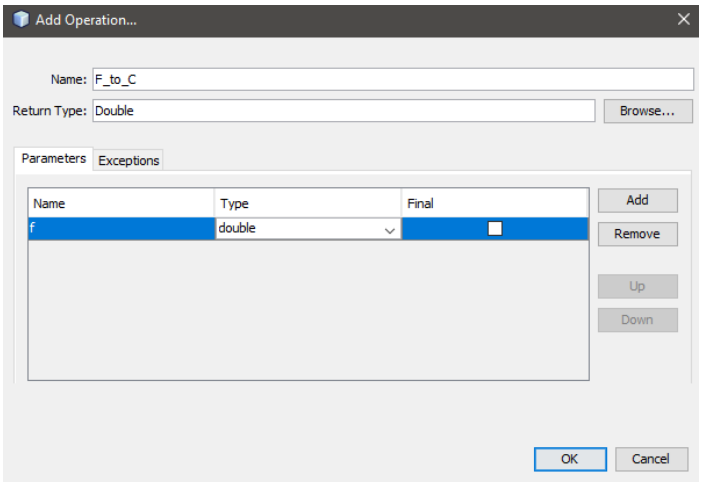
Step 6: As you can see in following pic; In Source Packages there is a package Service which contains the service file temp_services.java. Open this file by double clicking on it, so that we can add two operation that will convert temperature from Celsius to Fahrenheit and vice-versa. Go to design mode by clicking on Design.



Step 7: Click on Add Operation to add operation.



Step 8: Give Operation name F_to_C and return type as Double. So, this method will return value in Double data type. After that click on Add button to give parameters for method. Give its name as f and type as Double and then click on OK button. Your one operation is now successfully created.



Step 9: Repeat step 7 & 8 to create second operation. But this time replace some above entered data with following data: F_to_C -> C_to_F f -> c

Step 10: Now go to source mode by clicking on Source view and delete the hello segment of the code. Because it is default operation and we don't need this.

Step 11: Now replace the selected area with following code to convert Fahrenheit to Celsius.

Double c = (f – 32) * 1.8;

return c;

```
7   /**
8    *
9    * @author varun
10   */
11  @WebService(serviceName = "temp_services")
12  public class temp_services {
13      /**
14       * Web service operation
15       */
16      @WebMethod(operationName = "F_to_C")
        public Double F_to_C(@WebParam(name = "f") double f) {
            double c = (f-32)*1.8;
19          return c;
20
21      }
```

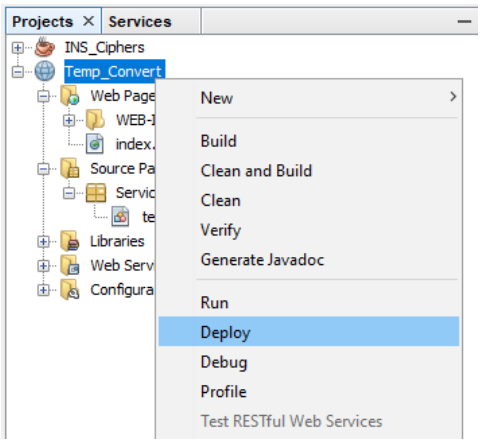
Step 12: Now replace the selected area with following code to convert Celsius to Fahrenheit and then press Ctrl+S to save.

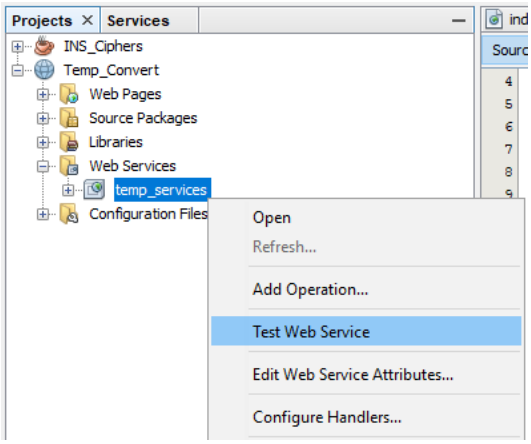
Double f = (c*1.8)+32;

return f;

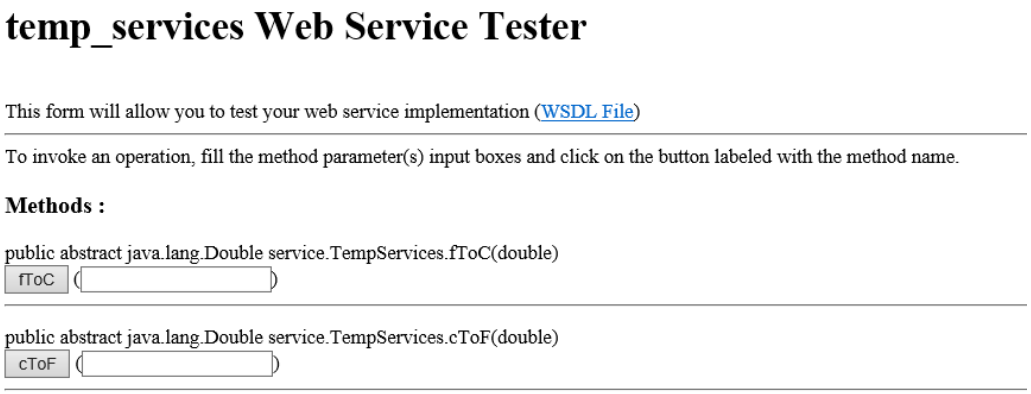
```
16      @WebMethod(operationName = "F_to_C")
        public Double F_to_C(@WebParam(name = "f") double f) {
18          double c = (f-32)*1.8;
19          return c;
20
21      }
22      /**
23       * Web service operation
24       */
25      @WebMethod(operationName = "C_to_F")
        public Double C_to_F(@WebParam(name = "c") double c) {
            //TODO write your implementation code here:
28          return null;
29      }
30  }
31
```

Step 13: Now right click on project name and click on Deploy to deploy your project. To test your web service, do the following process as shown in the second picture.

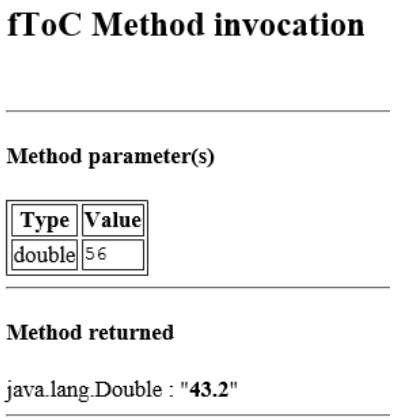




Step 14: Following window will be opened in the browser. Now if you will enter a numeric data into first box and you will click on first button it will convert the entered data into Celsius.



Step 15: Selected value is in Celsius of 56.



fToC Method invocation

Method parameter(s)

Type	Value
double	56

Method returned

java.lang.Double : "**43.2**"

Step 15: Similarly, second textbox and button will convert the numeric value into Fahrenheit. Now to consume this web service we are creating a client.

Step 16: Now create a new web application project with name as **Temp_client** and open the index.html page of Temp_client and write the following code into that.

<html> <head> <title>Temperature converter</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0"> </head>

<body> <form>

**
<input type="text" name="data">
**

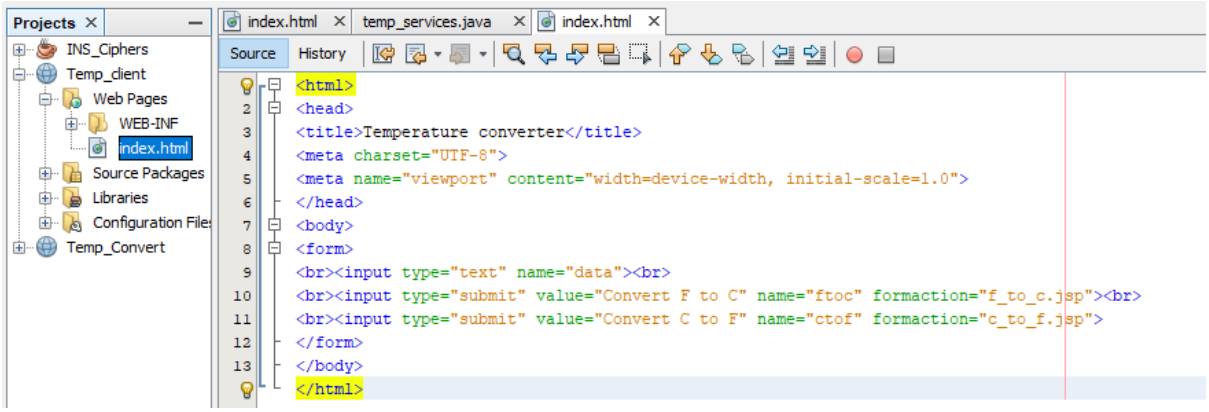
**
<input type="submit" value="Convert F to C" name="ftoc" formaction="f_to_c.jsp">
**

**
<input type="submit" value="Convert C to F" name="ctof" formaction="c_to_f.jsp">**

</form>

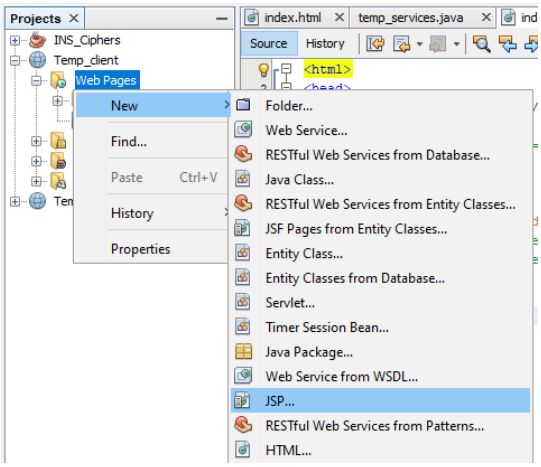
</body>

</html>

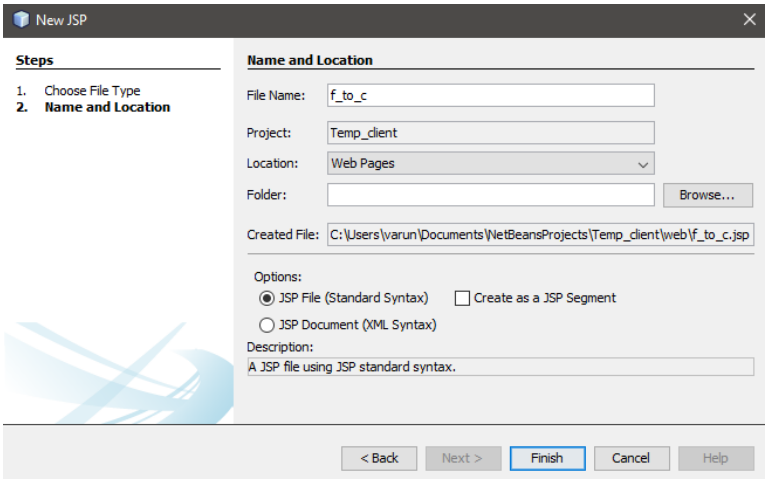


Step 17: Now create two JSP pages for both submit button.

Right click on Web Pages -> New -> JSP

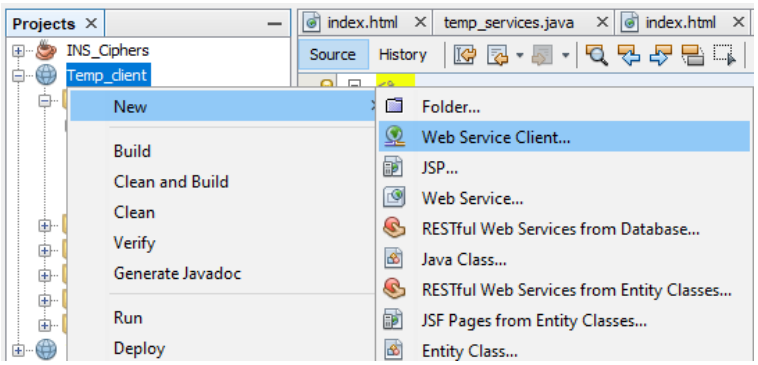


Step 18: Enter file name f_to_c and then click on Finish.

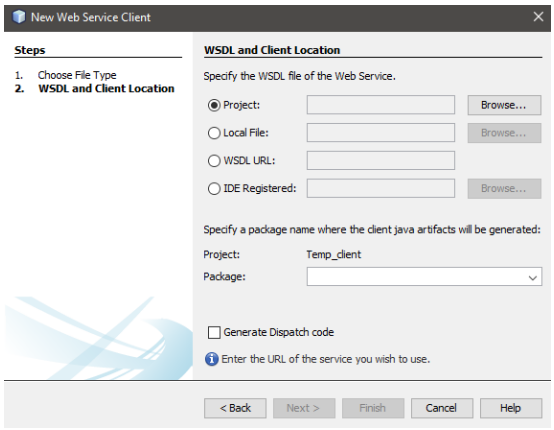


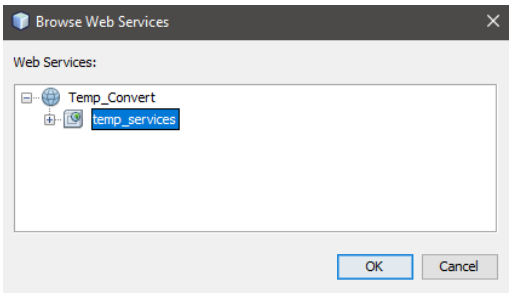
Step 19: Now repeat the step number 17 & 18. But give the File Name as c_to_f.

Step 20: Now you have created two jsp files. Right click on Temp_client and select Web Service Client as below.

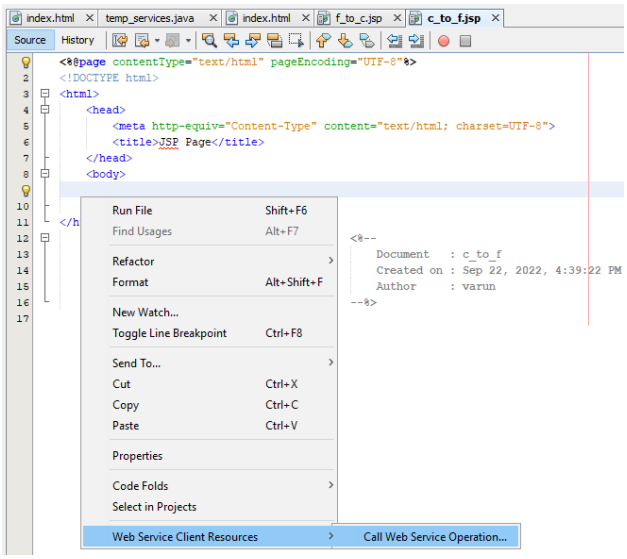


Step 21: In New Web Service Client window, click on Browse, a new window will appear and then select temp_services and click on OK. On the next window click on Finish.

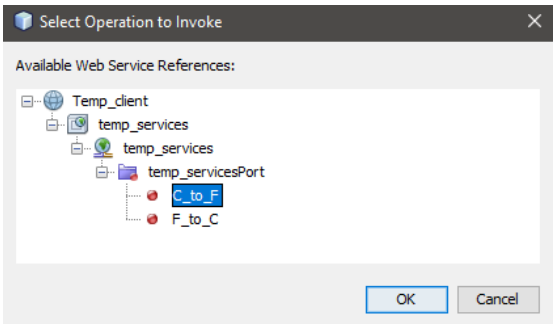


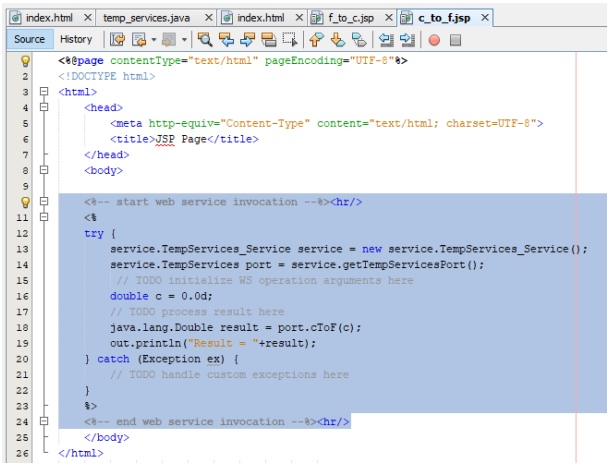


Step 22: Delete the content of the body tag then right click in between the body section and select Call Web Service Operation as below.

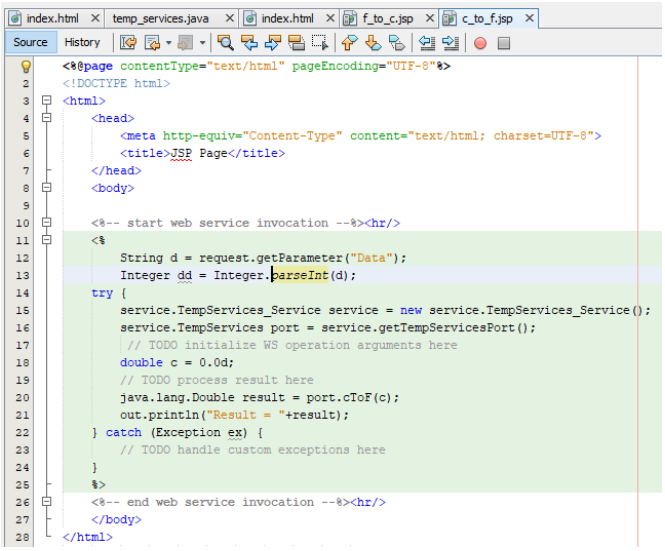


Step 23: New window will appear, select the C_to_F by expanding it and click on OK. So that selected code in second pic will be automatically generated.



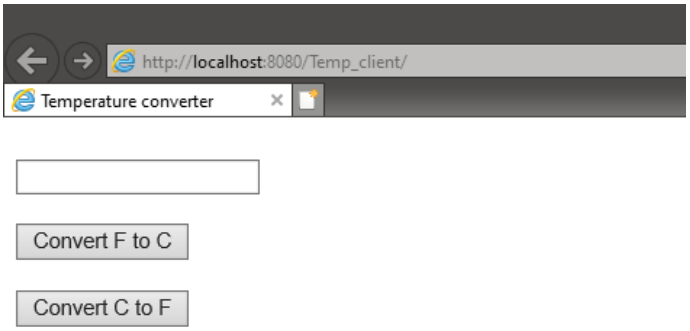


Step 24: Now make the selected area in step 23 as like selected area in below pic by adding some line of code.



Step 25: Now Open the f_to_c.jsp file and follow the steps from 22 to 24. Only the change is in 23 number step and i.e., instead of C_to_F, you have to select F_to_C.

Step 26: Now run the Temp_client project. A window will be open like below.



Step 27: Now you can to enter any numeric data into textbox and if you will click the first button it will convert the numeric value into Celsius and vice-versa for the second button.

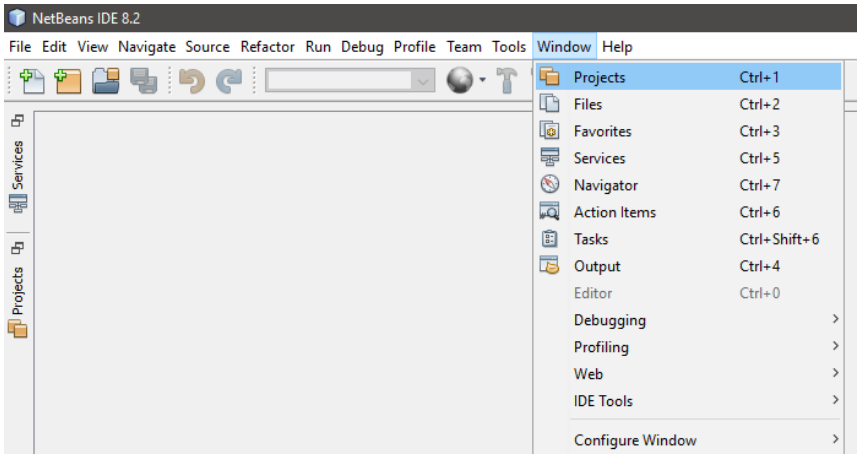
Step 28: There are so many methods to consume the web service. But I found it easy.

Practical – 05

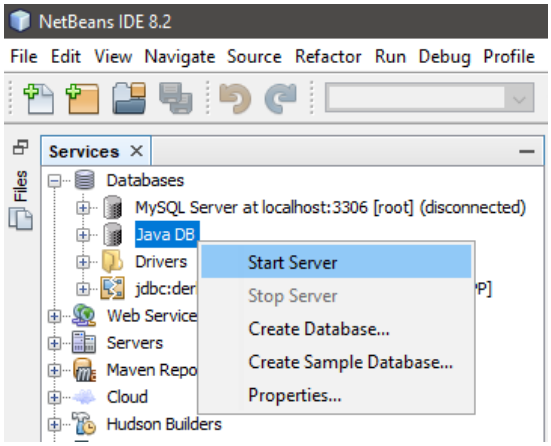
Aim: Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

Output:

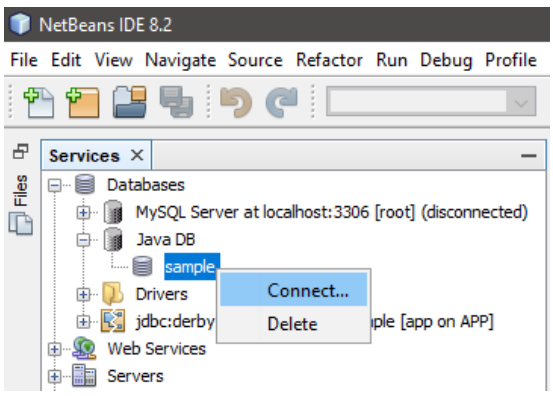
Step 1: Click on Window menu and click on Projects, Files & Services to open it.



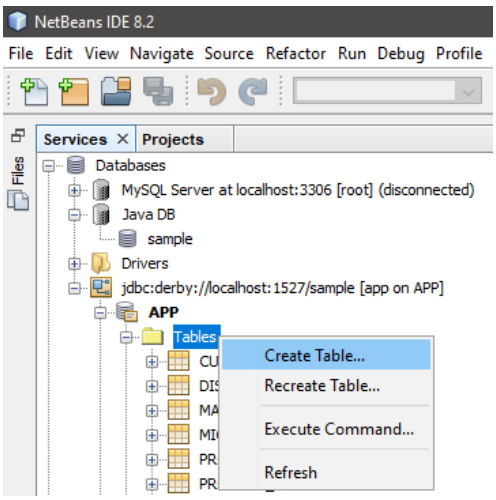
Step 2: Right click on Java DB and then click on Start Server to start the server.



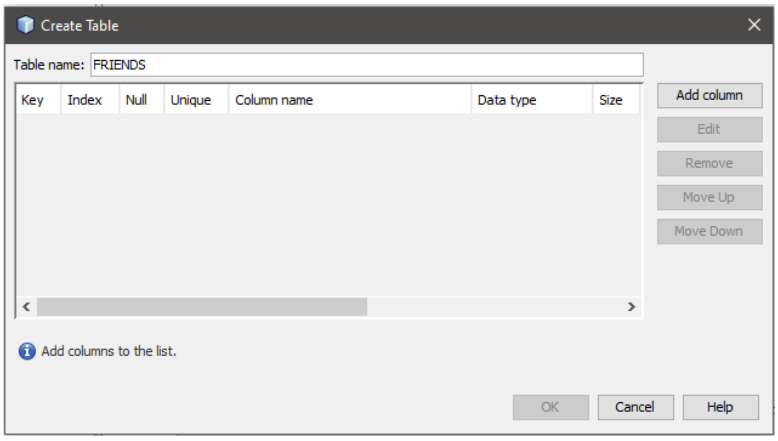
Step 3: Now expand Java DB and right click on sample and then click on connect to connect the sample database with server.



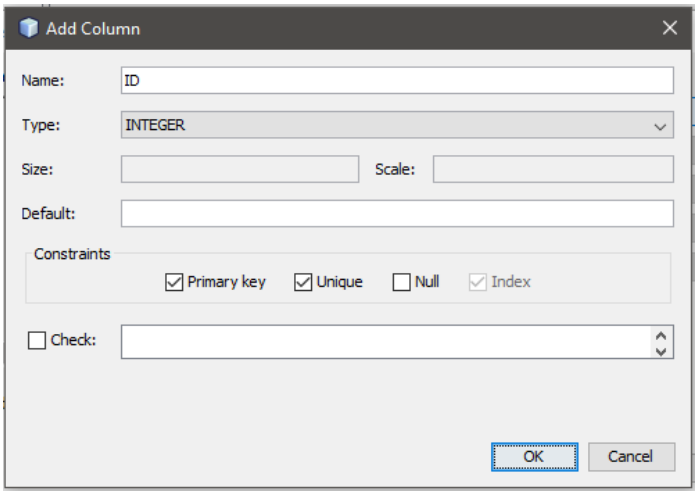
Step 4: Now we are going to create a table in default database sample.
Right click on Table -> Create Table



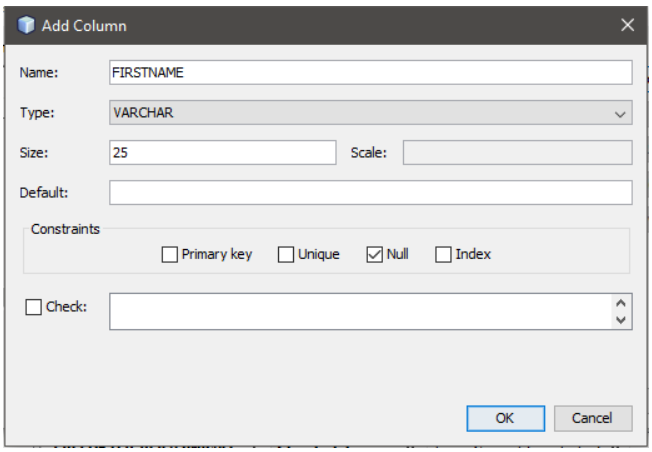
Step 5: Give table name as FRIENDS.



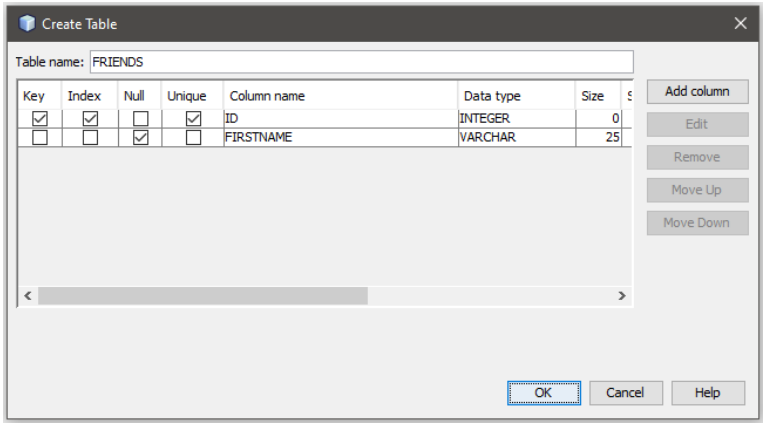
Step 6: Now click on Add column button to add columns in table. Enter details as shown in the picture below and select Primary key. After that click on OK button.



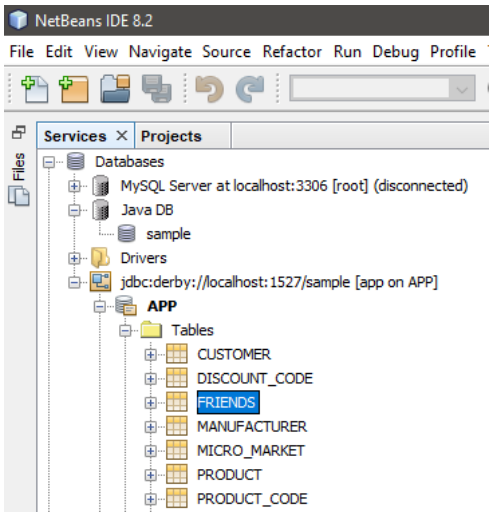
Step 7: Now add second column with following detail. But don't select primary and click on OK button.



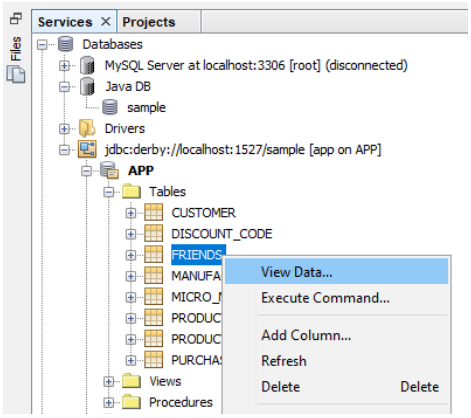
Step 8: Now click on OK button.



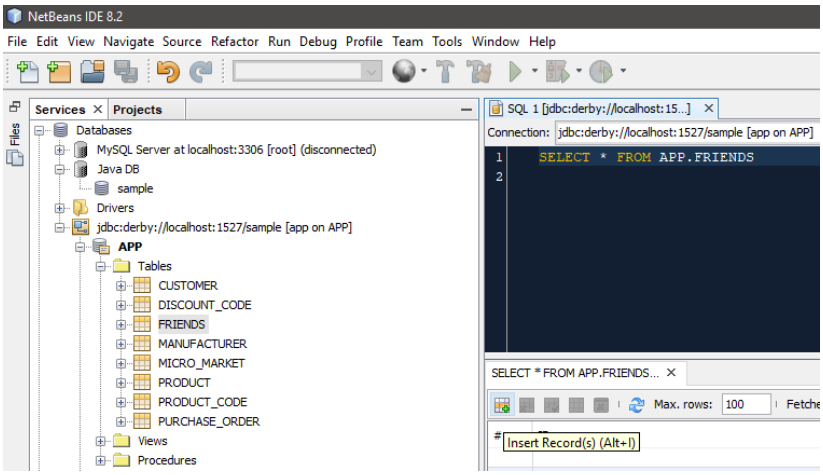
Step 9: Now you can see a table with name FRIENDS in the table.



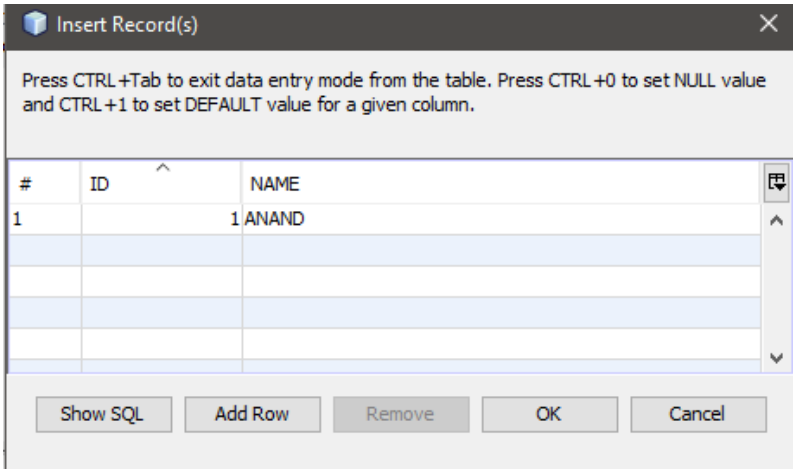
Step 10: Right click on FRIENDS to view and add records into it.



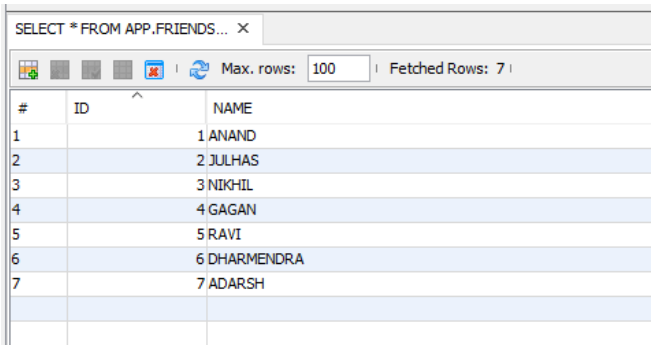
Step 11: Now click on the leftmost icon in second panel to insert some record.



Step 12: Insert a record and then click on Add Row button to insert more record. After that click on OK button to finish.

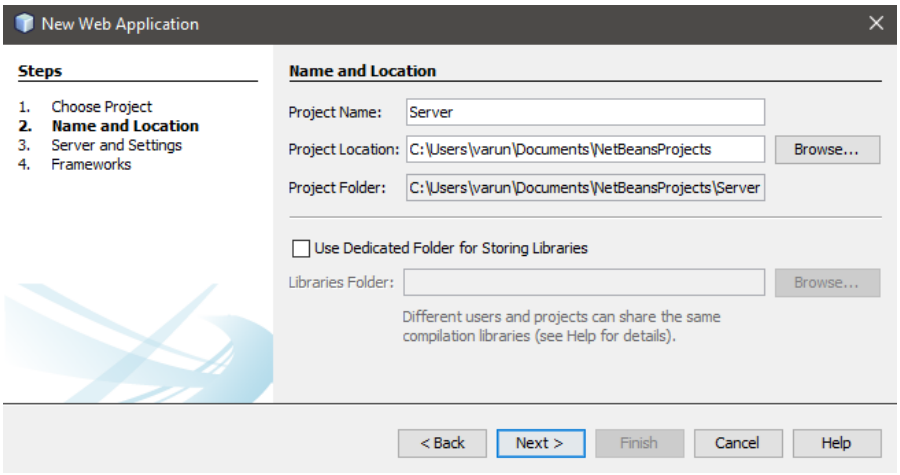


Step 13: As you can see, I have entered 7 records.

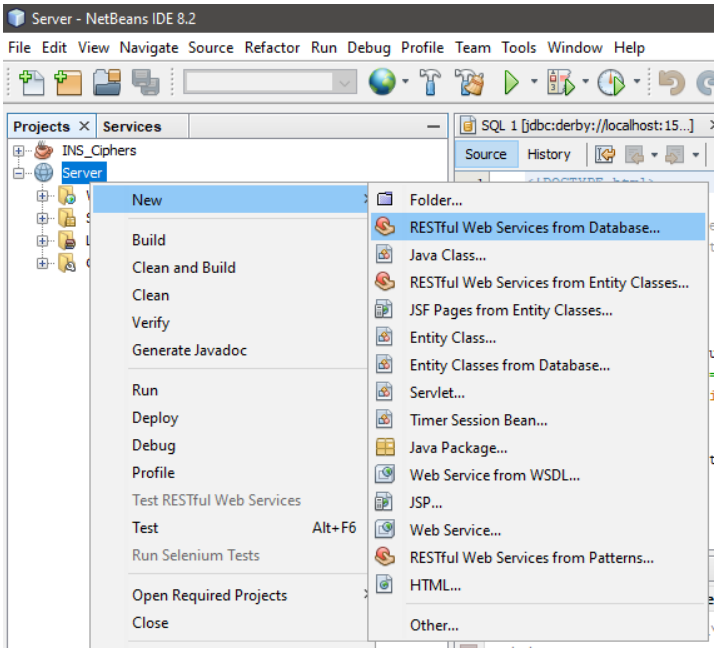


#	ID	NAME
1	1	ANAND
2	2	JULHAS
3	3	NIKHIL
4	4	GAGAN
5	5	RAVI
6	6	DHARMENDRA
7	7	ADARSH

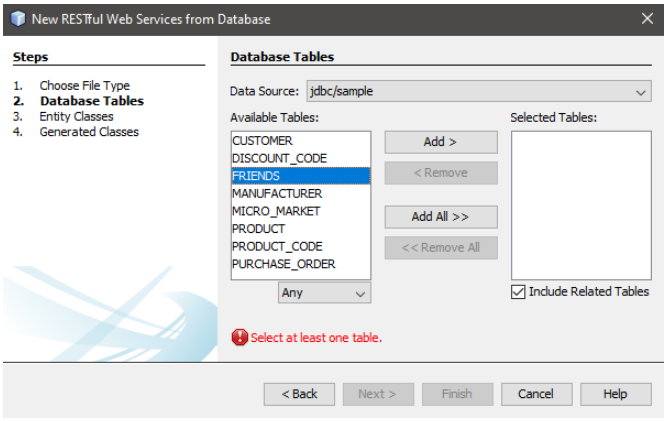
Step 14: Now create a web application with the name Server. After that click on Next and then Finish button.



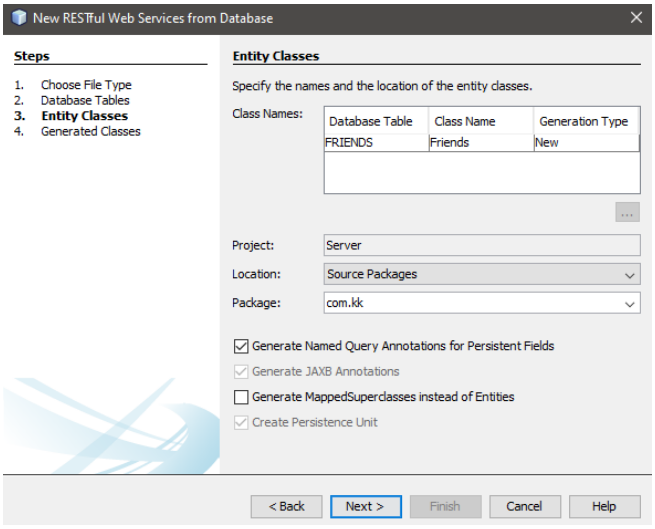
Step 15: Now create a RESTful Web Service from Database by right clicking on the project name.



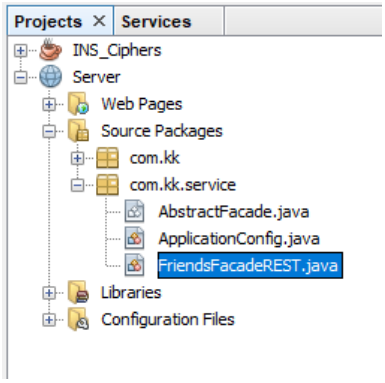
Step 16: Choose Data Source jdbc/sample and select FRIENDS and click on Add button. After that click on Next button

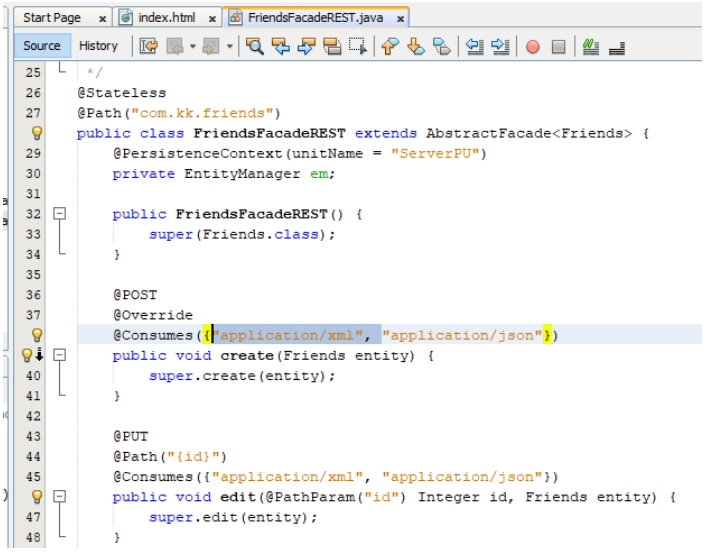


Step 17: Enter Package name as com.kk and click on Next button and then Finish.

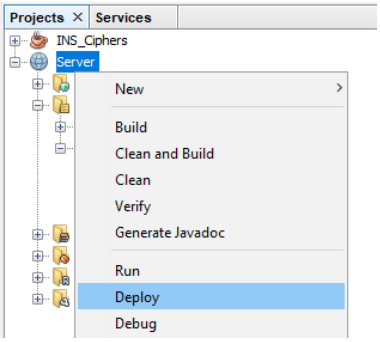


Step 18: Now open the selected file by double clicking on it and remove the selected part from every method in this file. So that it will communicate only in JSON format. You can also use methods to convert it. But this is easiest method.

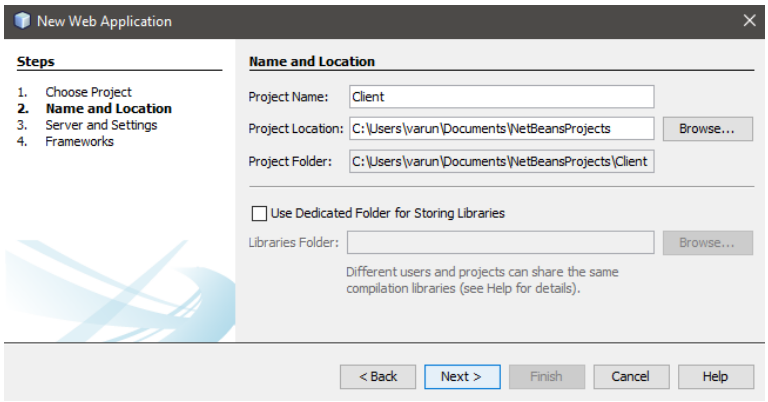




Step 19: After that right click on project name and Deploy it.



Step 20: Now create one more web application as Client. After that click on Next and then Finish button.



Step 21: Now open the index.html file of Client project and add the following code in between HEAD tag.

```
<style>
table { font-family: arial, sans-serif;
border-collapse: collapse;
} td, th { border: 1px solid #000000;
text-align: center;
padding: 8px; }
</style> <script>
var request = new XMLHttpRequest();
request.open('GET', 'http://localhost:8080/Server/webresources/com.kk.friends/', true);
request.onload = function () {
// begin accessing JSON data here
var data = JSON.parse(this.response);
for (var i = 0; i < data.length; i++) {
var table = document.getElementById("myTable");
var row = table.insertRow();
```

var cell1 = row.insertCell(0);

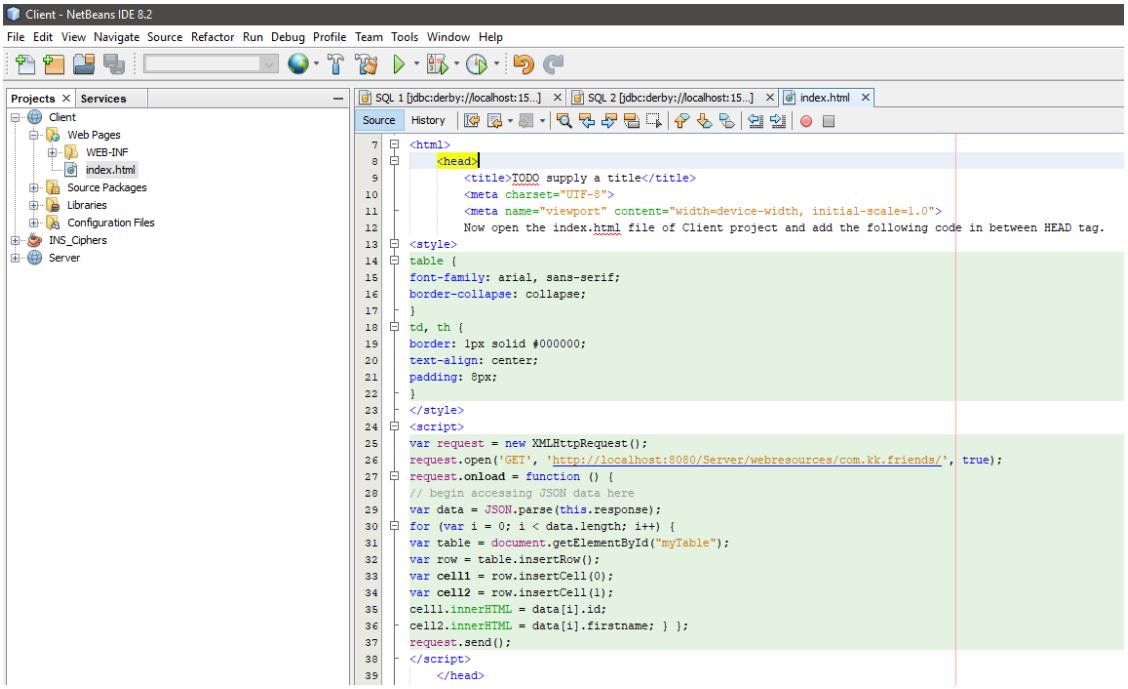
var cell2 = row.insertCell(1);

cell1.innerHTML = data[i].id;

cell2.innerHTML = data[i].firstname; } };

request.send();

</script>



Server URL is case sensitive. It will change accordingly if you have not given the project and file names as of mine.

Step 22: Replace the content of body tag with the following code.

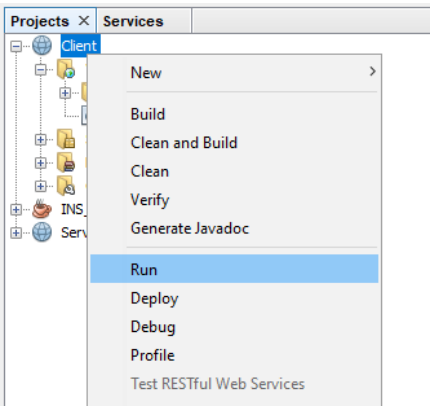
<table id="myTable">

<tr> <th>ID</th>

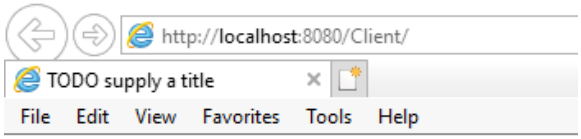
<th>NAME</th> </tr> </table>

```
</script>
    </head>
    <body>
        <table id="myTable">
        <tr>
        <th>ID</th>
        <th>NAME</th>
        </tr>
        </table>
    </body>
</html>
```

Step 23: Now Run the Client Web Application.



Step 24: A window will open in browser which represents a data in tabular format. These data are the records entered in FRIEND table.



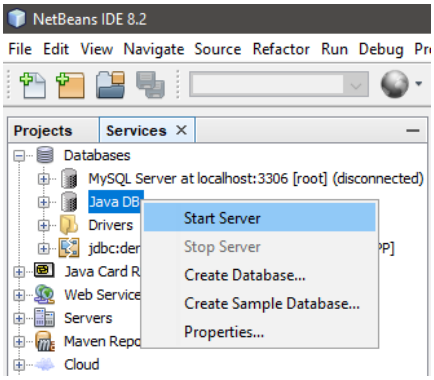
ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	GAGAN
5	RAVI
6	DHARMENDRA
7	ADARSH

Practical – 06

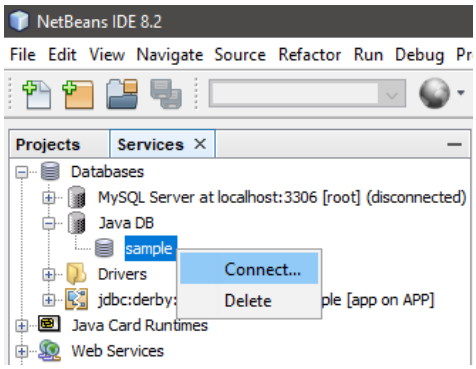
Aim: Define a RESTful web service that accepts the details to be stored in a database and performs CRUD operation.

Output:

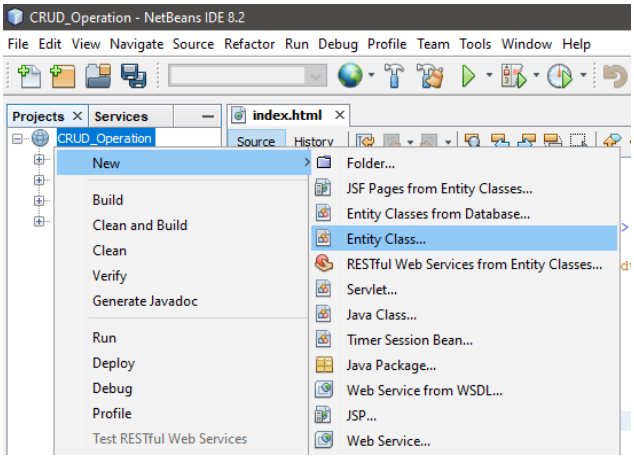
1. Right click on Java DB and then click on Start Server to start the server. If services window is not visible then click on Window menu and click on Projects, Files & Services to open it.



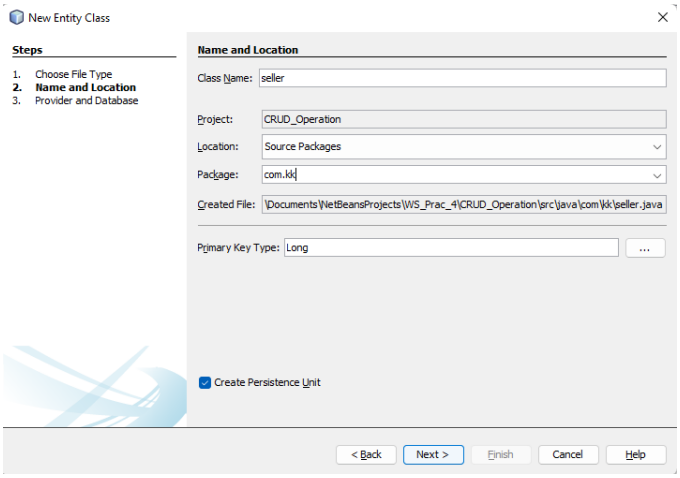
2. Now expand Java DB and right click on sample and then click on connect to connect the sample database with server.



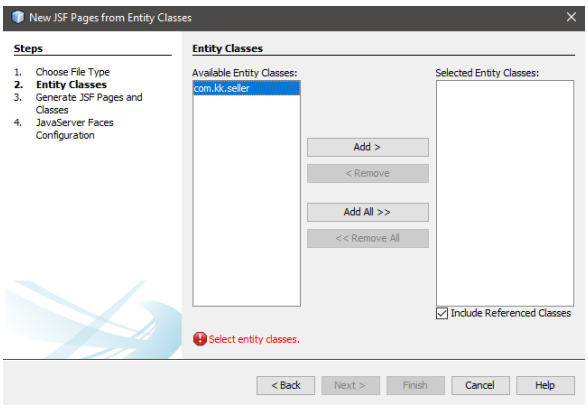
3. Now create a web application with the name **CRUD_Operation**. Then create an entity class. Right click on project name -> New -> Entity Class.

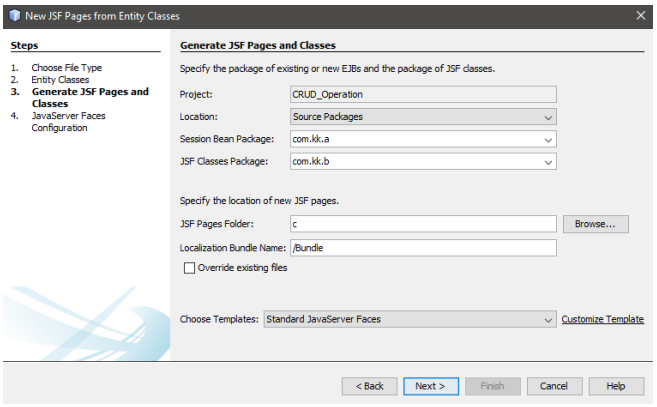


4. A window will appear like below pic. Enter following data and click on Next and Finish.
Class Name -> seller
Package Name -> com.kk

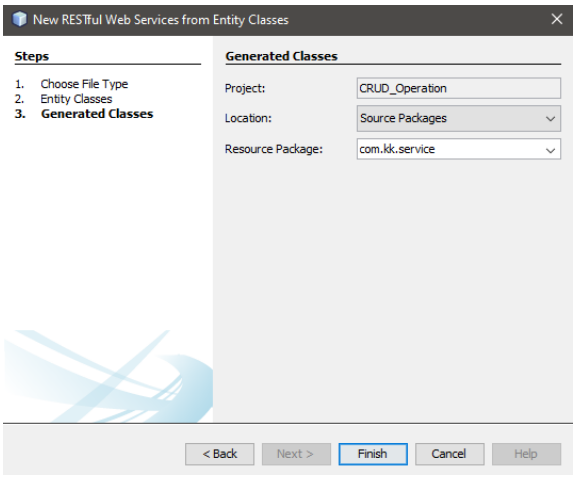


5. Now create JSF Pages from Entity Classes. Right click on project name -> New -> JSF Pages from Entity Classes. Select com.kk.seller and click on Add button and then Next button. A window like below will appear on the screen. Enter the data into that window as entered in second pic and click on Next and Finish on next window.

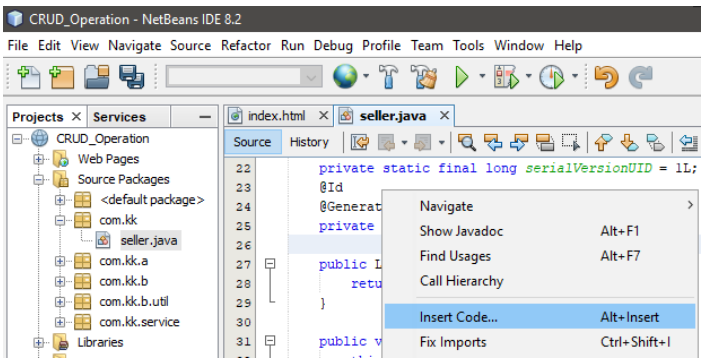


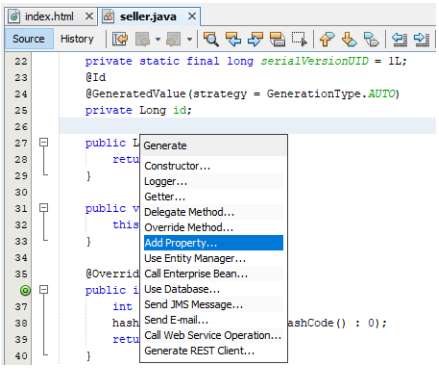


6. Create RESTful Web Services from Entity Classes. Right click on project name -> New -> RESTful Web Services from Entity Classes. Repeat step 5 for selecting com.kk.seller entity class and then on next page and enter the **com.kk.service** in Resource Package and click on Finish button.

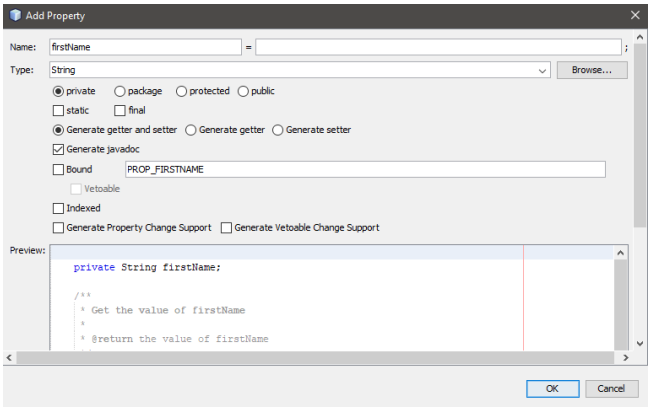


7. Now open seller.java file under com.kk package. In this file at line number 26, do right click and select Insert Code. A new list will appear. Click on Add Property.

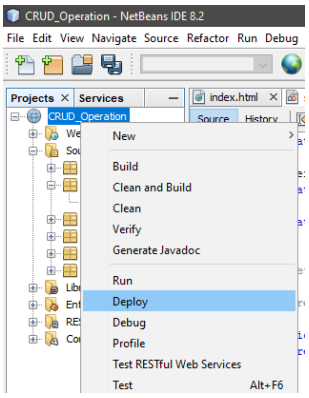


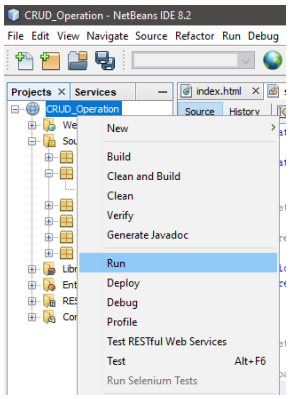


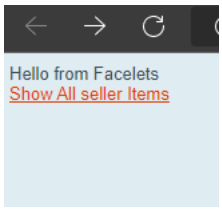
8. A new window will open. Enter name as firstName. Make sure name should be exact same as of mine and then click on OK button. Actually, we are setting getter and setter method for firstName.

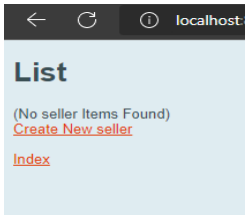


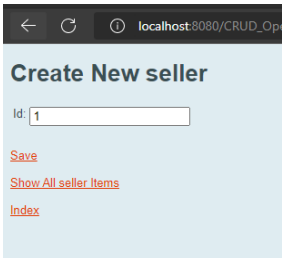
9. Now right click on web application name - > Deploy then Run it. A window will open in browser like below. Now click on Show All sellers Items for CRUD operation. You can add data by clicking on Create New seller and can view, edit and delete by clicking on View, Edit and Destroy options. Click on Create New seller. Enter id into Id. Now click on Save option to save the data. Click on Show All seller Items to view all records whether our data is entered or not. We can see that 1 Id is appearing.

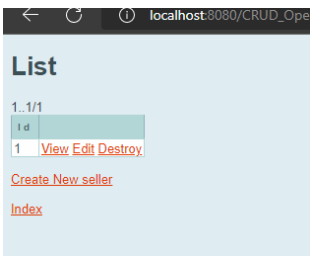










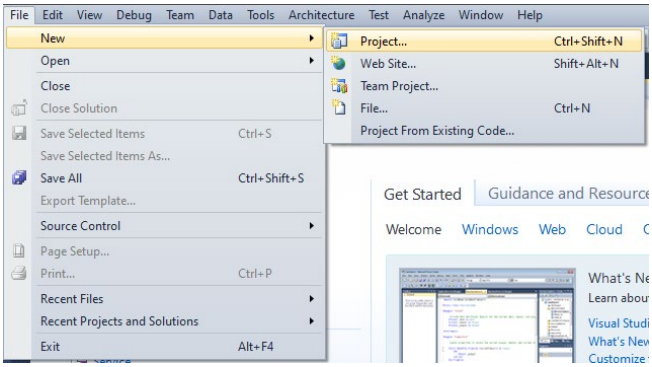


Practical – 07

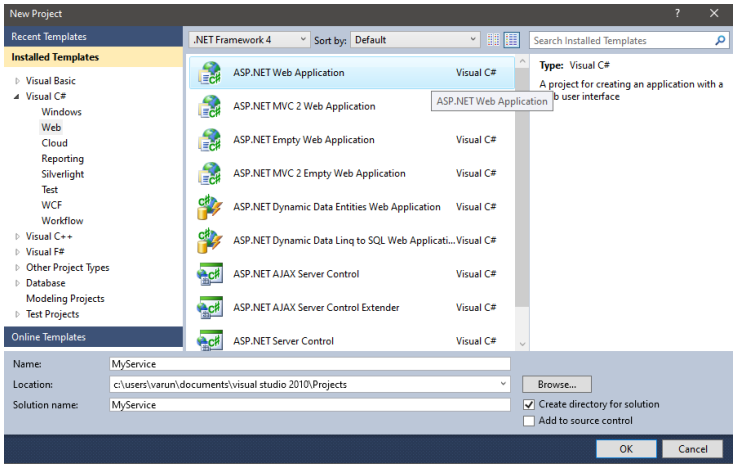
Aim: Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.

Output:

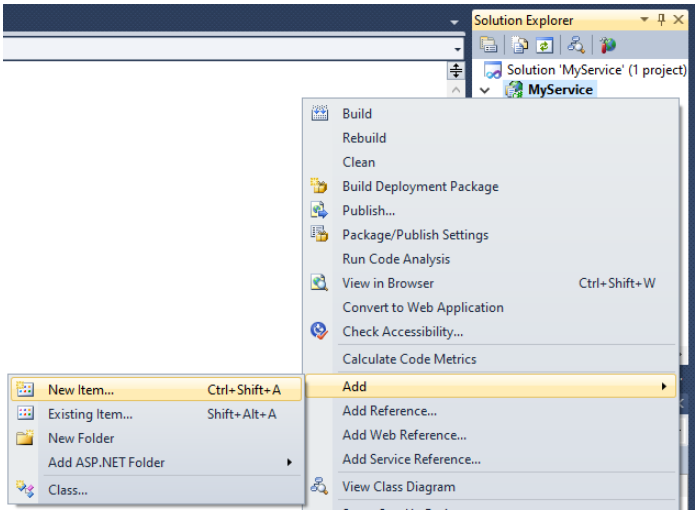
1. Open Visual Studio to create a Web Application. File -> New -> Project



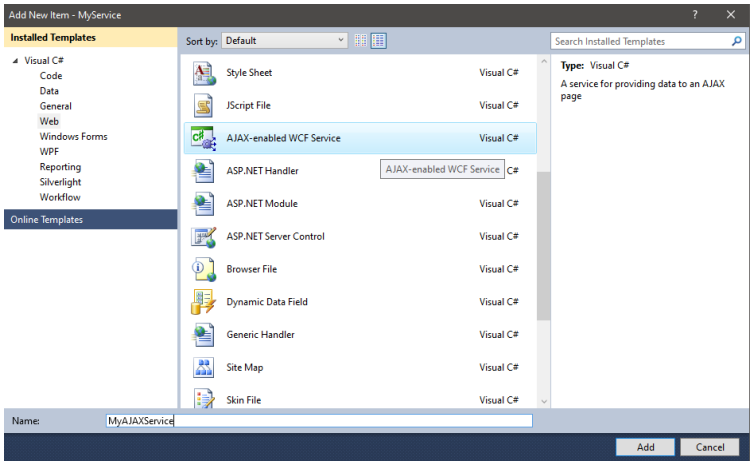
2. Select the latest .NET Framework available. Then go to Visual C# -> Web -> ASP.NET Web Application. Give name as **Service** and click on OK button.



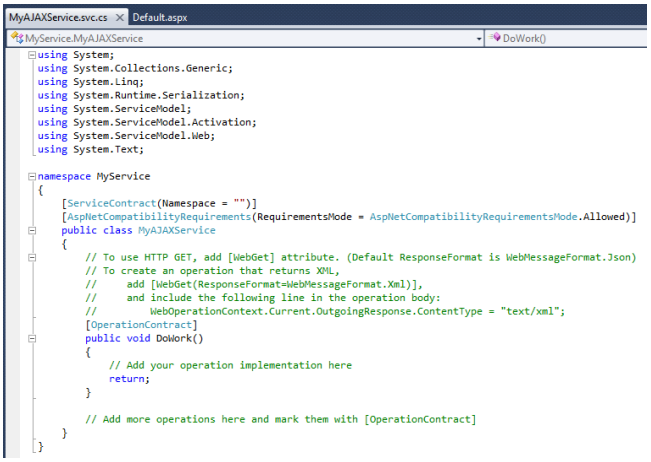
3. Now we are going to add AJAX-enabled WCF Service. Go to solution Explorer. Right click on MyService -> Add -> New Item.



4. In Templates select AJAX-enabled WCF Service and give **MyAJAXService** name to it. After that click on Add button.

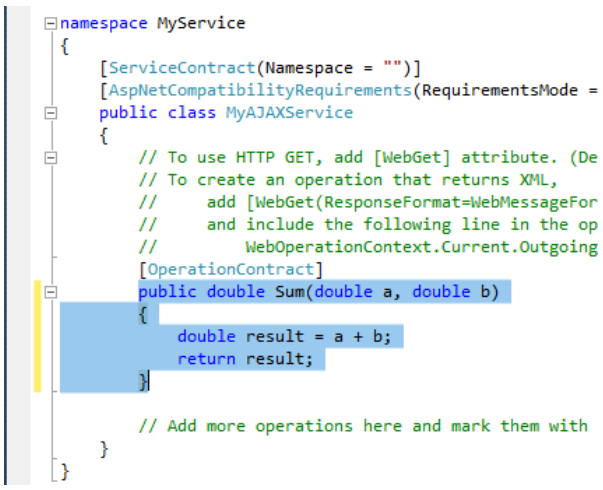


5. In solution explorer, service is added to the web application. It will be opened automatically. If not then open it by double clicking on it. It will look like below.

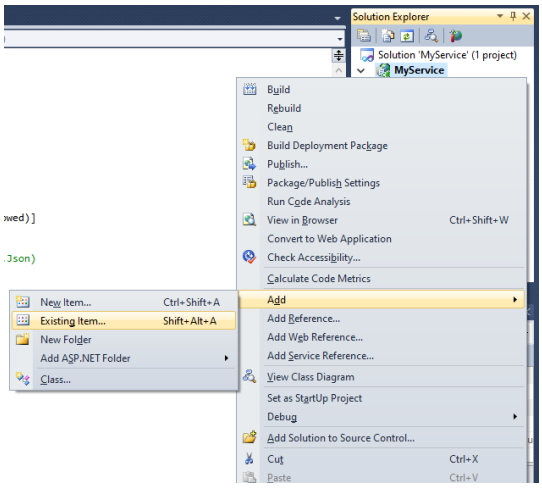


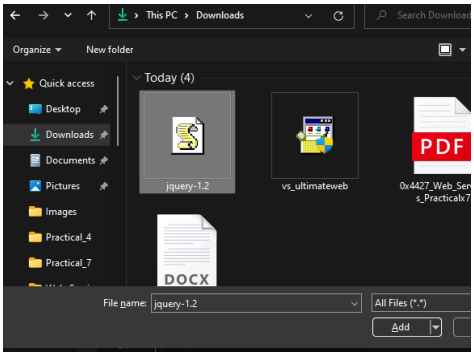
6. Replace the DoWork service method with another method. As I'm going to add a service method Sum for addition of two numbers. You can add any method. But accordingly you have to change the calling of this service. After adding the below code, press Ctrl+S to save the changes.

```
public double Sum(double a, double b)
{
double result = a + b;
return result;
 }
```

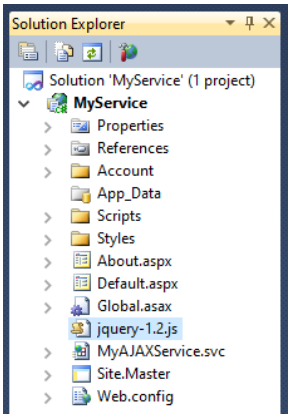


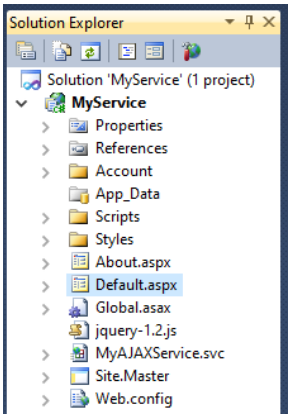
7. As I will use jQuery to call above WCF Service. So I need to add library for jQuery. I have downloaded it and I'm going to add it to my Web Application. Right click on MyService -> Add -> Existing Item or use Shift+Alt+A. Then locate and select it and press Add button.





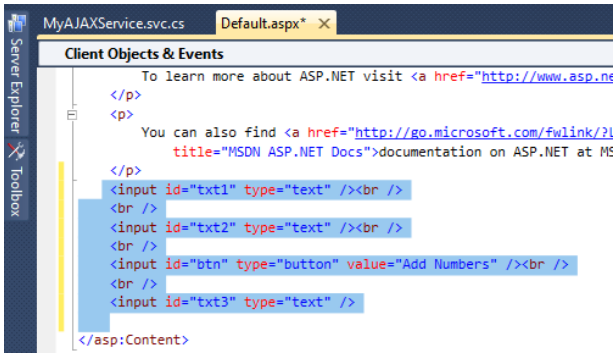
8. As you can see library is added to application. Now I'm going to consume that service. Open Default.aspx page from Solution Explorer by double clicking on it.

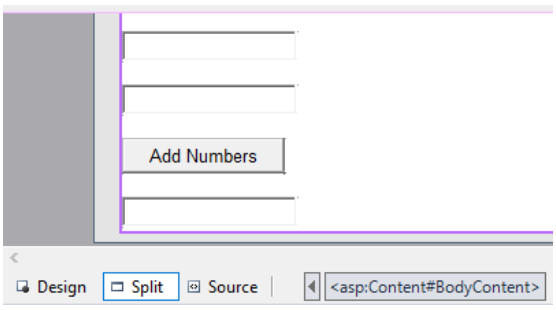




10. Now add the following code after closing </p> tag. It will create three textboxes and one button. You can see it in Design mode.

```
<input id="txt1" type="text" /><br />

<br />

<input id="txt2" type="text" /><br />

<br />

<input id="btn" type="button" value="Add Numbers" /><br />

<br />

<input id="txt3" type="text" />
```





11. We have given following ID's to above components. We will use these ID's in jQuery . 1st Textbox = txt1, 2nd Textbox = txt2, Button = btn, 3rd Textbox = txt3.

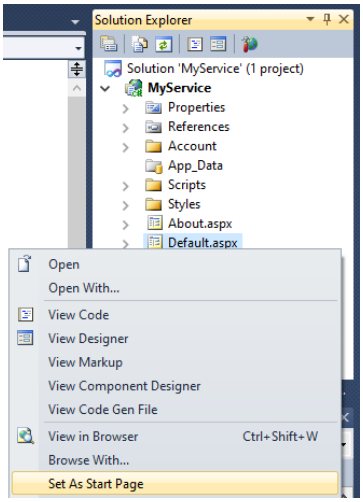
1st and 2nd textboxes are for enter two numbers. When you will click on button then jQuery will get triggered and by calling the service we will get output in 3rd textbox. Now again go into Source mode.

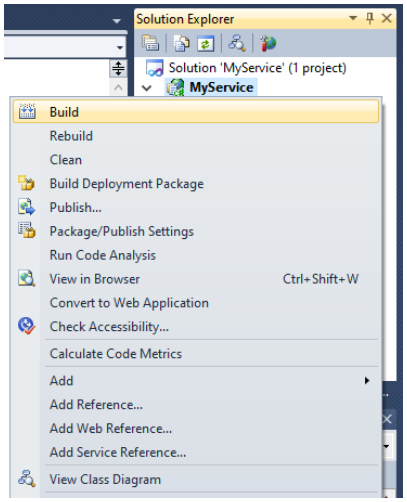
12. Add the following jQuery code in HEAD tag section and then press Ctrl+S to save the changes.

```
<script type="text/javascript" src="jQuery -1.2.js"></script>

<script type="text/javascript" >

$(document).ready(function (){

$("#btn").click(function (){

var num1 = $("#txt1").val();

var num2 = $("#txt2").val();

$.ajax({ url: "MyAJAXService.svc/Sum",

type: "POST",

contentType: "application/json; charset=utf-8",

data:JSON.stringify({a: num1, b: num2}),

dataType: "json",

success : function(data){

$("#txt3").val(data.d); },

error : function(err){

alert(err); } }); }); });

</script>
```

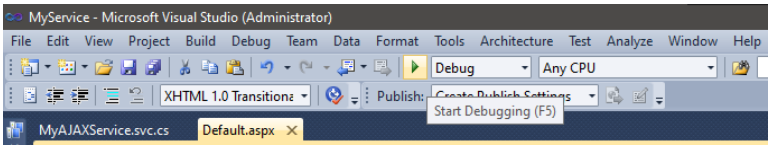
As you can see in first line of code, in src I have given the path of library that I have added using step 8 and 9. Without this library our code will not work. So it is mondatory.

13. Now set Default.aspx page as start page. Right click on Default.aspx -> Set As Start Page. Then Build the MyService application.

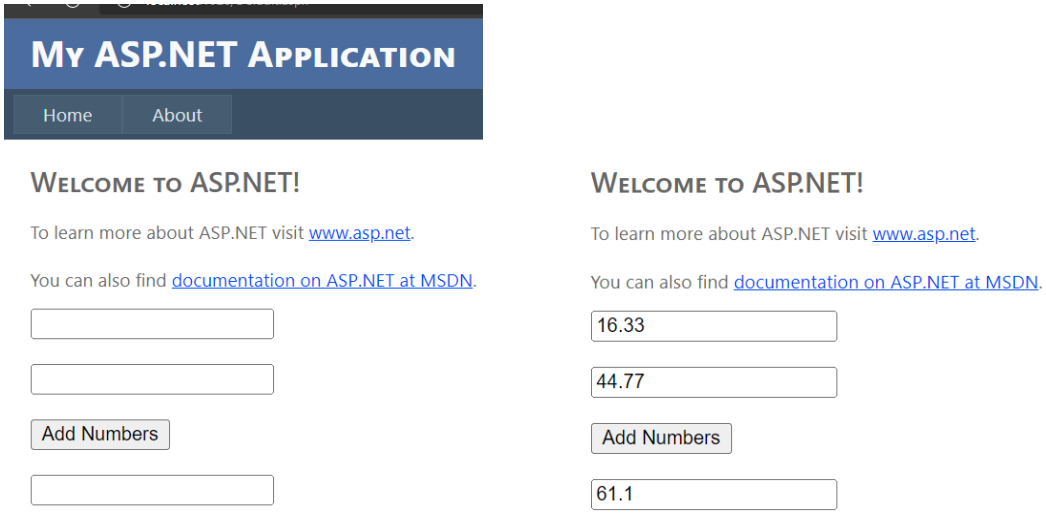




14. Press Run button in Toolbar.



15. A window will open in the browser like below. Enter two numbers and click on Add Numbers button to see the result.

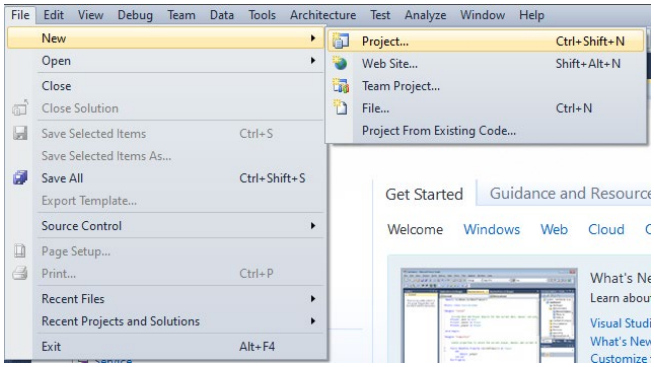


Practical – 08

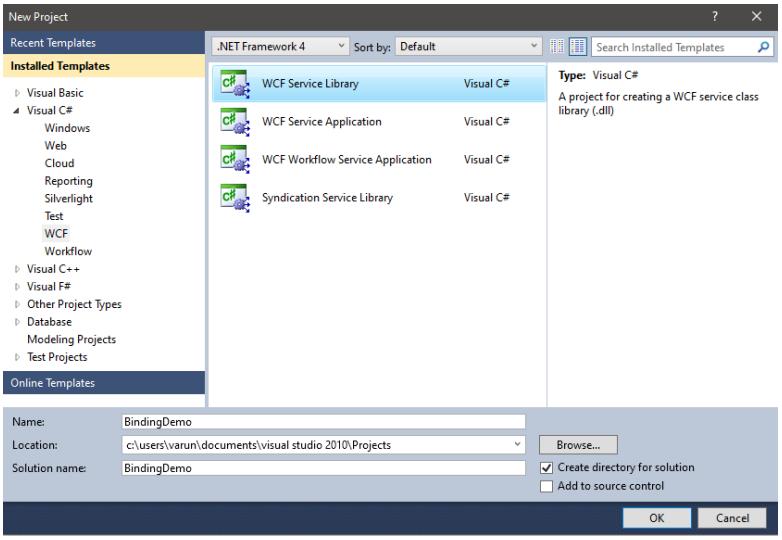
Aim: Demonstrate using the binding attribute of an endpoint element in WCF.

Output:

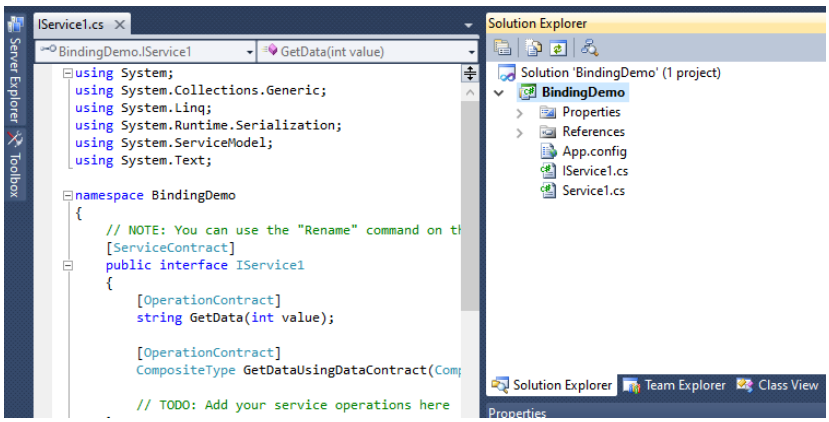
1. Open Visual Studio. Then go to File -> New -> Project



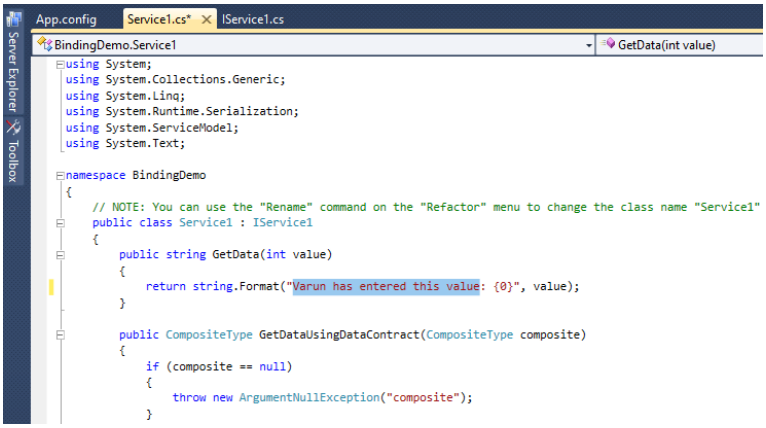
2. Select the latest .NET Framework available. Then go to Visual C# -> WCF -> WCF Service Library. Give name as **BindingDemo** and click on OK button.



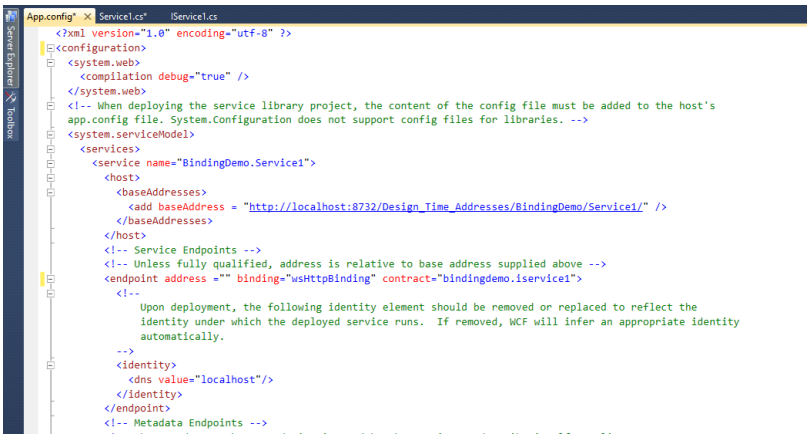
3. Now you can see, On right side in Solution Explorer BindingDemo project is created.



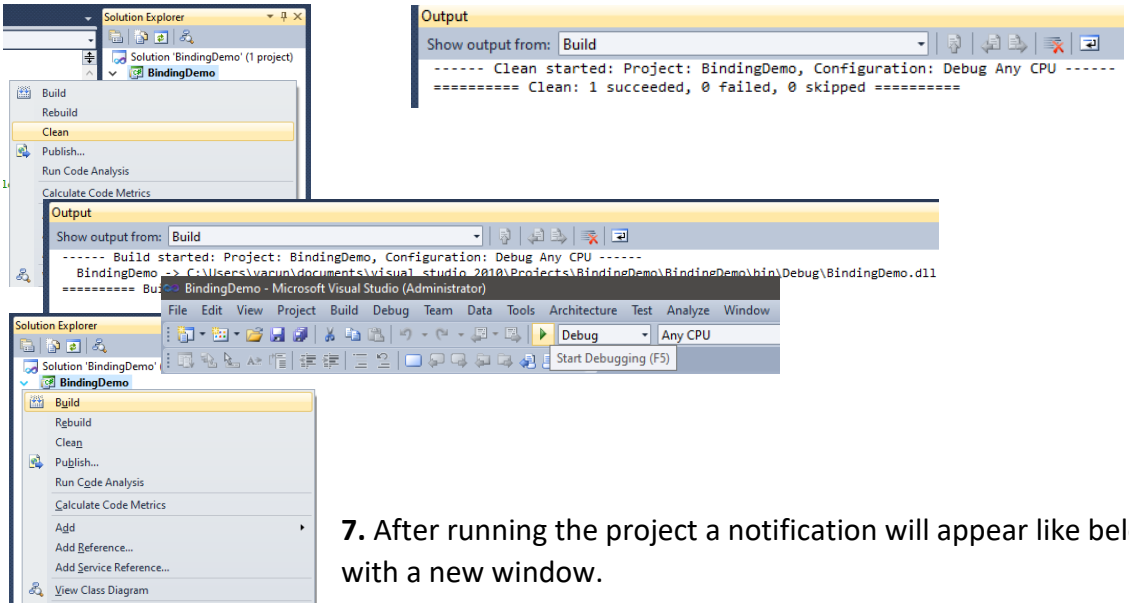
4. Open App.config and Service1.cs file by double clicking on them. In Service1.cs file, change the following code as shown in the below picture and press Ctrl+S to save it.



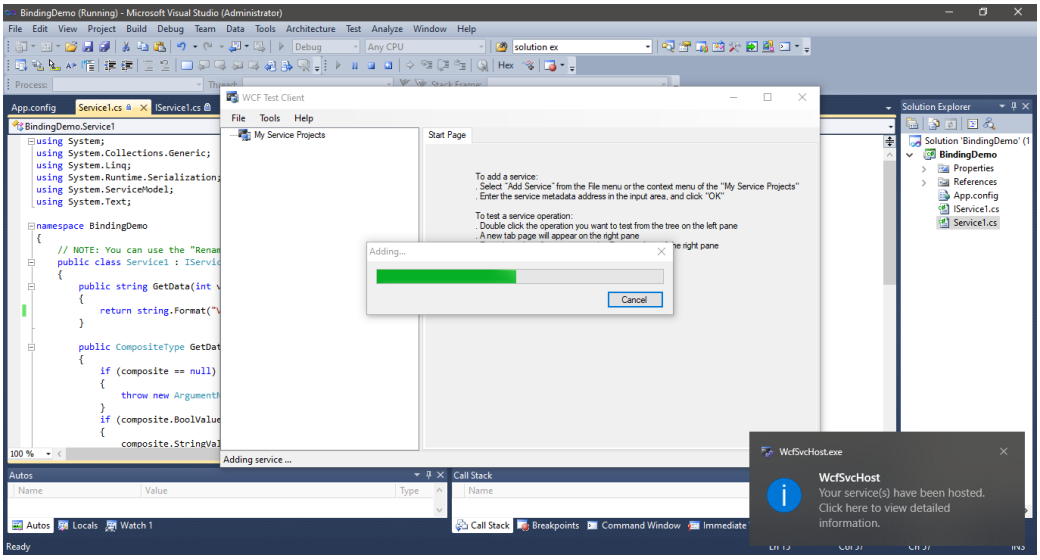
5. In App.config file, name and contract attribute of service and endpoint will contain your WCF application name **BindingDemo**.



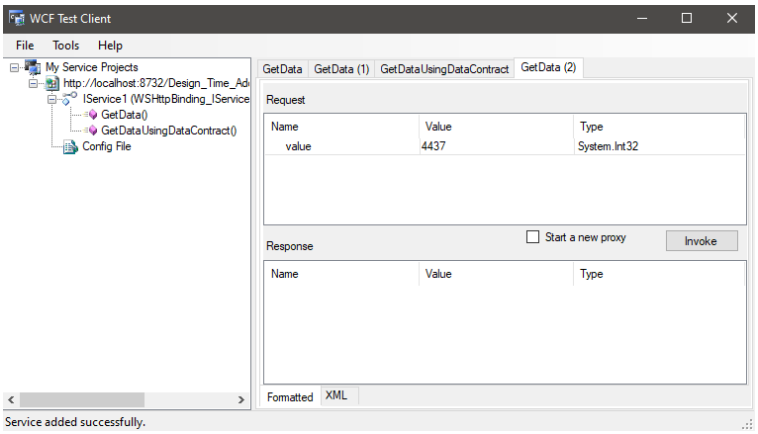
6. Now right click on BindingDemo project -> Clean. Again right click on project -> Build. And finally click on Run.

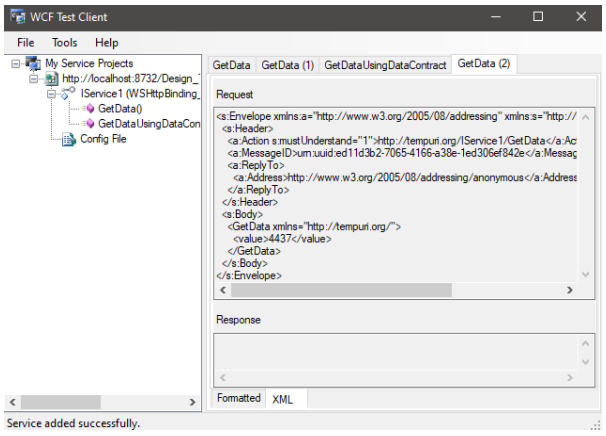


7. After running the project a notification will appear like below with a new window.

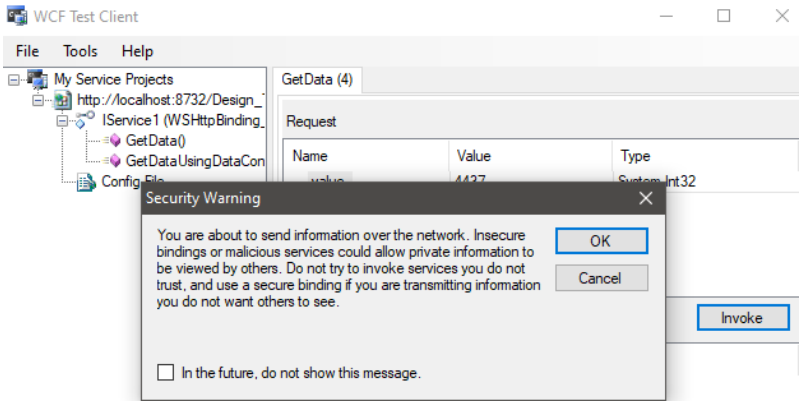


8. In WCF Test Client window, double click on GetData() and enter a value in the Request window. Click on XML to see the XML Request. Go back to Formatted and click on Invoke.





9. After clicking on Invoke, A security warning will appear, Click on OK.



10. As you can see**,** Response is generated with the given value in both Formatted and XML format.