EX NO:14 **WEB SERVICES**

DATE:

AIM:

To Create a web service application that allows users to update customer information in a database,

and returns a list of customers. Use Java, JAX-RS, and JDBC to connect to the database. Create

a web service client in Java to invoke the web service.

Note: JAX-RS stands for Java API for RESTful Web Services. It is a Java programming

language API that provides support for creating web services that follow the REST

architectural style..

ALGORITHM:

Step 1: Start: Create a Java class named db annotated with @WebService.

Step 2: Define a method insert annotated with @WebMethod. It accepts parameters id, name, and amount for inserting records into the database.

Step 3: Within the insert method, establish a connection to the database using JDBC.

Step 4: Execute an SQL insert query to add the new record to the database.

Step 5: Execute a select query to retrieve all records from the customer table and construct a string representation of the records.

Step 6: Define a method edit annotated with @WebMethod to update existing records in the database.

Step 7: Inside the edit method, execute an SQL update query to modify the record with the provided id.

Step 8: Return a string containing the result of the operation (success or failure) along with the updated list of records.

Step 9: Stop.

SOURCE CODE:

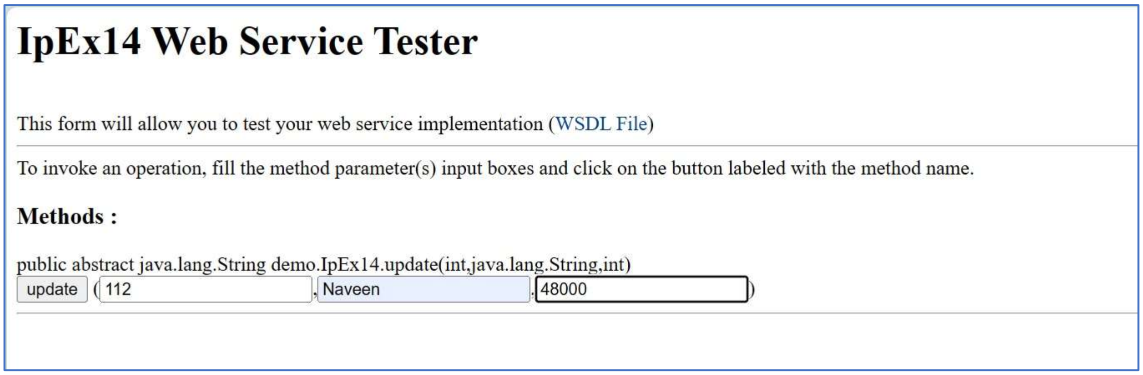
--db.java

```
package demo;
import java.sql.*;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "db")
public class db
{
   @WebMethod(operationName = "insert")
   public String insert(@WebParam(name = "id") int id, @WebParam(name = "name") String name, @WebParam(name = "amount") int amount)
   {
       StringBuilder result = new StringBuilder();
       result.append("The new record for "+name+" is inserted successfully\n\n");
       result.append("ID\t").append("Name\t").append("Total\n");
       result.append("---\t").append("-----\t").append("------\n");
       try{
          Connection c=DriverManager.getConnection("jdbc:derby://localhost:1527/vishwa");
          Statement st=c.createStatement();
          st.executeUpdate("INSERT INTO customer VALUES("+id+",'"+name+"',"+amount+")");
          ResultSet rs=st.executeQuery("Select * from customer");
          while(rs.next())
          {
             int cid=rs.getInt("id");
             String nam=rs.getString("name");
             int tot=rs.getInt("total");
             result.append(cid+"\t").append(nam+"\t").append(tot+"\n");
```

```
            }
            st.close();
            c.close();
        }
        catch(SQLException e)
        {
            e.printStackTrace();
        }
        return result.toString();
    }


    @WebMethod(operationName = "edit")
    public String edit(@WebParam(name = "id") int id, @WebParam(name = "name") String
name, @WebParam(name = "amount") int amount)
    {
        StringBuilder result = new StringBuilder();
        try {
            Connection c = DriverManager.getConnection("jdbc:derby://localhost:1527/vishwa");
            Statement st=c.createStatement();
            String updateQuery = "UPDATE customer SET name='" + name + "', total=" + amount
+ " WHERE id=" + id;
            int rowsAffected = st.executeUpdate(updateQuery);
            if (rowsAffected > 0) {
                result.append("The customer record with ID ").append(id).append(" is updated
successfully.\n\n");
                result.append("ID\t").append("Name\t").append("Total\n");
                result.append("---\t").append("-----\t").append("------\n");
                ResultSet rs=st.executeQuery("Select * from customer");
                while(rs.next())
                {
                    int cid=rs.getInt("id");
```

```
          String nam=rs.getString("name");
          int tot=rs.getInt("total");
          result.append(cid+"\t").append(nam+"\t").append(tot+"\n");
        }
      }
      else {
        result.append("Failed to update the customer record with ID ").append(id).append(".\n\n");
      }
      st.close();
      c.close();
    }
    catch (SQLException e) {
      e.printStackTrace();
      result.append("Error occurred while updating the customer record.\n\n");
    }
    return result.toString();
  }
}
```

OUTPUT:



IpEx14 Web Service Tester

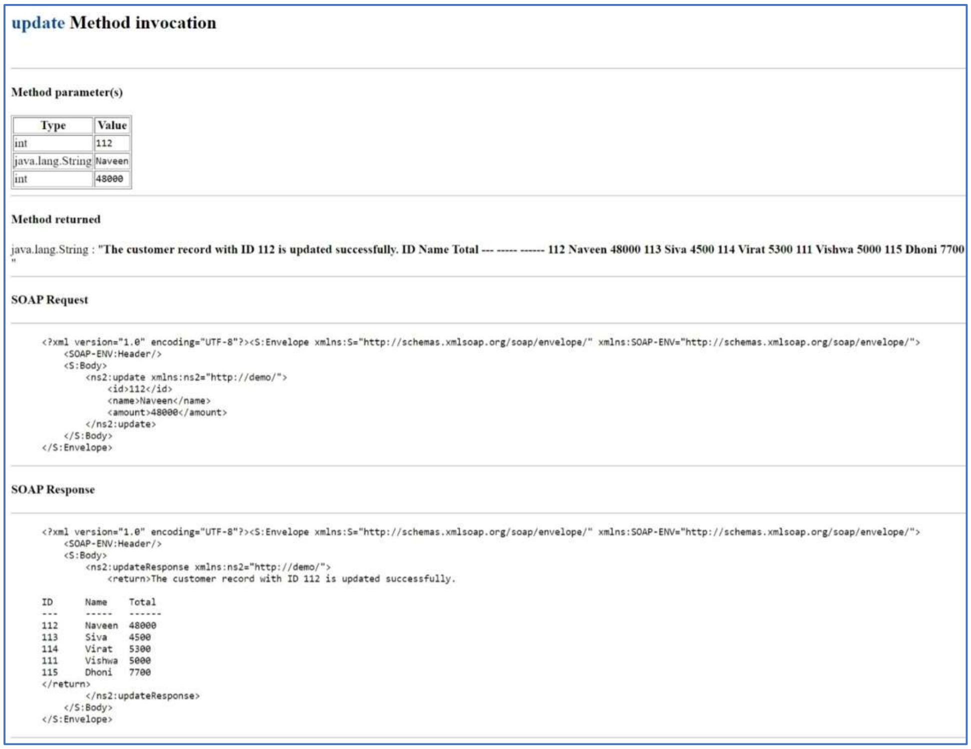
This form will allow you to test your web service implementation (WSDL File)

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 demo.IpEx14.update(int,java.lang.String,int)

update (112 , Naveen , 48000)



update Method invocation

Method parameter(s)

Type	Value
int	112
java.lang.String	Naveen
int	48000

Method returned

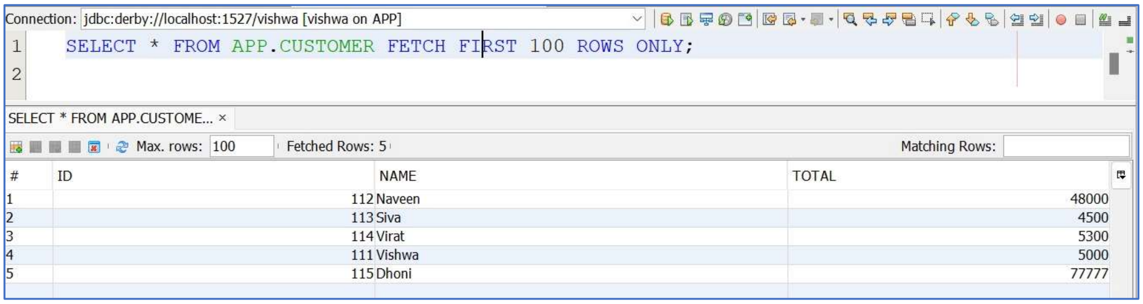
java.lang.String : **"The customer record with ID 112 is updated successfully. ID Name Total --- ----- ------ 112 Naveen 48000 113 Siva 4500 114 Virat 5300 111 Vishwa 5000 115 Dhoni 7700 "**

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:update xmlns:ns2="http://demo/">
            <id>112</id>
            <name>Naveen</name>
            <amount>48000</amount>
        </ns2:update>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:updateResponse xmlns:ns2="http://demo/">
            <return>The customer record with ID 112 is updated successfully.

ID      Name    Total
---     -----   ------
112     Naveen  48000
113     Siva    4500
114     Virat   5300
111     Vishwa  5000
115     Dhoni   7700
</return>
        </ns2:updateResponse>
    </S:Body>
</S:Envelope>
```



Connection: jdbc:derby://localhost:1527/vishwa [vishwa on APP]

```
SELECT * FROM APP.CUSTOMER FETCH FIRST 100 ROWS ONLY;
```

SELECT * FROM APP.CUSTOME... ×

Max. rows: 100 Fetched Rows: 5 Matching Rows:

#	ID	NAME	TOTAL
1	112	Naveen	48000
2	113	Siva	4500
3	114	Virat	5300
4	111	Vishwa	5000
5	115	Dhoni	77777

Code & Output (20)	
Quiz (5)	
Timely Submission (5)	
Total (30)	
Initial	

RESULT:

Thus the execution of code for a given problem statement of Webservices by using java has been executed and the output is verified successfully.