

PRACTICAL 1

AIM: Develop Time Server service that returns current time in JAVA and call it from clients developed in JAVA, PHP, Android and .NET

INPUT:

```
//TimeServer.java
package MSCCS.Shiv21;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
@WebService
@SOAPBinding(style=Style.RPC)
public interface TimeServer
{@WebMethod String getTimeAsString();
@WebMethod long getTimeAsElapsed();
}
```

```
//TimeServerImpl.java
package MSCCS.Shiv21;
import java.util.Date;
import javax.jws.WebService;
@WebService(endpointInterface="MSCCS.Shiv21.TimeServer")
public class TimeServerImpl
{public String getTimeAsString()
{return new Date().toString();}
public long getTimeAsElapsed()
{return new Date().getTime();}}
```

```
//TimeServerPublisher.java
package MSCCS.Shiv21;
import javax.xml.ws.Endpoint;
public class TimeServerPublisher
{public static void main(String[] args)
{Endpoint.publish("http://127.0.0.1:9876/one?wsdl",new TimeServerImpl());}}
```

```
//TimeClient.java
package MSCCS.Shiv21;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.URL;
class TimeClient
{public static void main(String[] args) throws Exception
```

```
{URL url=new URL("http://127.0.0.1:9876/one?wsdl");
QName qname=new QName("http://Shiv21.MSCCS/","TimeServerImplService");
Service service=Service.create(url,qname);
TimeServer eif=service.getPort(TimeServer.class);
System.out.println(eif.getTimeAsString());
System.out.println(eif.getTimeAsElapsed());}
```

OUTPUT:

Publishing Time Server:

```
C:\> C:\Windows\system32\cmd.exe - java MSCCS.Shiv21.TimeServerPublisher
```

```
D:\>javac MSCCS/Shiv21/*.java
```

```
D:\>java MSCCS.Shiv21.TimeServerPublisher
```

Executing Time Client:

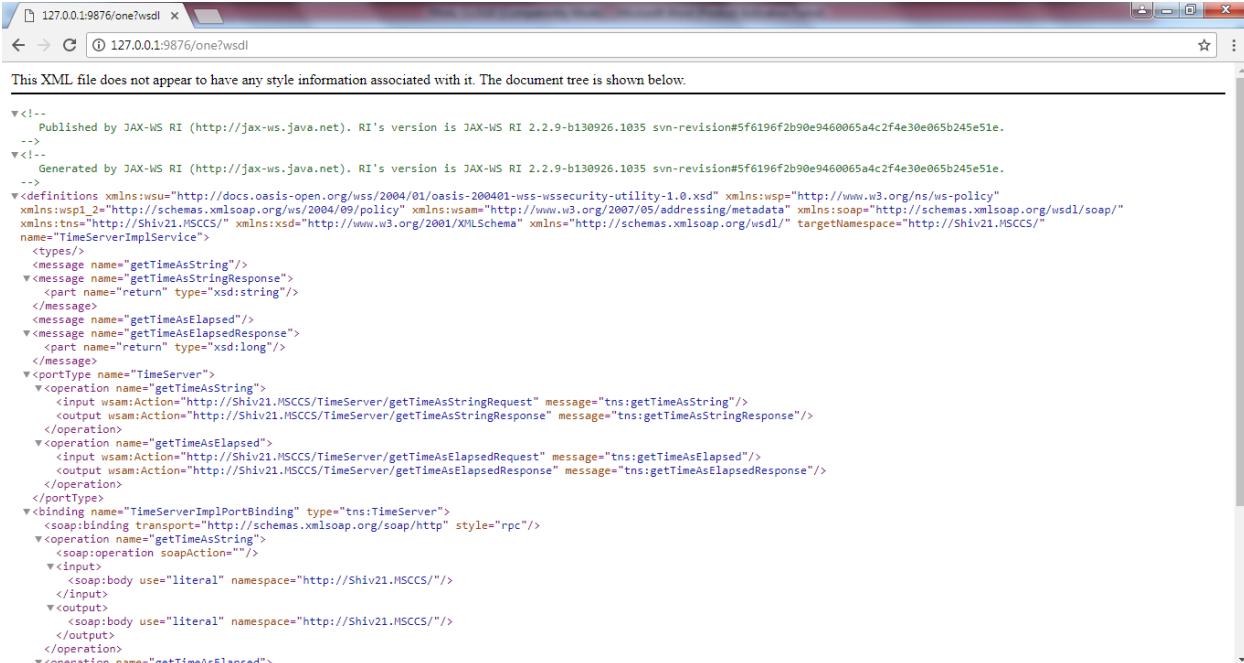
```
C:\> C:\Windows\system32\cmd.exe
```

```
D:\>java MSCCS.Shiv21.TimeClient
Mon Mar 05 13:07:23 EST 2018
1520273243356
```

```
D:\>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
<!--
  Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsps="http://www.w3.org/ns/ws-policy"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://Shiv21.MSCCS/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://Shiv21.MSCCS/" name="TimeServerImplService">
  <types/>
  <message name="getTimeAsString"/>
  <message name="getTimeAsStringResponse">
    <part name="return" type="xsd:string"/>
  </message>
  <message name="getTimeAsElapsed"/>
  <message name="getTimeAsElapsedResponse">
    <part name="return" type="xsd:long"/>
  </message>
  <operation name="TimeServer">
    <operation name="getTimeAsString">
      <input wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsStringRequest" message="tns:getTimeAsString"/>
      <output wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsStringResponse" message="tns:getTimeAsStringResponse"/>
    </operation>
    <operation name="getTimeAsElapsed">
      <input wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsElapsedRequest" message="tns:getTimeAsElapsed"/>
      <output wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsElapsedResponse" message="tns:getTimeAsElapsedResponse"/>
    </operation>
  </portType>
  <binding name="TimeServerImplPortBinding" type="tns:TimeServer">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
    <operation name="getTimeAsString">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="literal" namespace="http://Shiv21.MSCCS"/>
      </input>
      <output>
        <soap:body use="literal" namespace="http://Shiv21.MSCCS"/>
      </output>
    </operation>
    <operation name="getTimeAsElapsed">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="literal" namespace="http://Shiv21.MSCCS"/>
      </input>
      <output>
        <soap:body use="literal" namespace="http://Shiv21.MSCCS"/>
      </output>
    </operation>
  </binding>
  <service name="TimeServerImplService">
    <port name="TimeServerImplPort" binding="tns:TimeServerImplPortBinding">
      <soap:address location="http://127.0.0.1:9876/one"/>
    </port>
  </service>
</definitions>
```



```
127.0.0.1:9876/one?wsdl x
127.0.0.1:9876/one?wsdl

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="TimeServerImplService" targetNamespace="http://Shiv21.MSCCS/TimeServer" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://Shiv21.MSCCS/TimeServer" xmlns:soap="http://schemas.xmlsoap.org/soap/http" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/soap/http" xmlns:wsam="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing/endpoint-reference" xmlns:wsa11="http://schemas.xmlsoap.org/ws/2004/08/addressing/endpoint-reference/policy" xmlns:wsa4="http://schemas.xmlsoap.org/ws/2004/08/addressing/header">

    <message name="getTimeAsStringResponse">
        <part name="return" type="xsd:string"/>
    </message>
    <message name="getTimeAsElapsedResponse">
        <part name="return" type="xsd:long"/>
    </message>
    <portType name="TimeServer">
        <operation name="getTimeAsString">
            <input wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsStringRequest" message="tns:getTimeAsString"/>
            <output wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsStringResponse" message="tns:getTimeAsStringResponse"/>
        </operation>
        <operation name="getTimeAsElapsed">
            <input wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsElapsedRequest" message="tns:getTimeAsElapsed"/>
            <output wsam:Action="http://Shiv21.MSCCS/TimeServer/getTimeAsElapsedResponse" message="tns:getTimeAsElapsedResponse"/>
        </operation>
    </portType>
    <binding name="TimeServerImplPortBinding" type="tns:TimeServer">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
        <operation name="getTimeAsString">
            <soap:operation soapAction="" />
            <input>
                <soap:body use="literal" namespace="http://Shiv21.MSCCS/"/>
            </input>
            <output>
                <soap:body use="literal" namespace="http://Shiv21.MSCCS/"/>
            </output>
        </operation>
        <operation name="getTimeAsElapsed">
            <soap:operation soapAction="" />
            <input>
                <soap:body use="literal" namespace="http://Shiv21.MSCCS/"/>
            </input>
            <output>
                <soap:body use="literal" namespace="http://Shiv21.MSCCS/"/>
            </output>
        </operation>
    </binding>
    <service name="TimeServerImplService">
        <port name="TimeServerImplPort" binding="tns:TimeServerImplPortBinding">
            <soap:address location="http://127.0.0.1:9876/one"/>
        </port>
    </service>
</definitions>
```

PRACTICAL 2

AIM: Develop web service in java that returns complex data types (e.g. as List of friends).

INPUT:

```
//Player.java
package Shiv21.team;
public class Player {
    private String name;
    private String nickname;
    public Player() { }
    public Player(String name, String nickname) {
        setName(name);
        setNickname(nickname);
    }
    public void setName(String name) { this.name = name; }
    public String getName() { return name; }
    public void setNickname(String nickname) { this.nickname = nickname; }
    public String getNickname() { return nickname; }
}
```

```
//Team.java
package Shiv21.team;
import java.util.List;
public class Team
{
    private List<Player> players;
    private String name;
    public Team() { }
    public Team(String name, List<Player> players)
    {
        setName(name);
        setPlayers(players);
    }
    public void setName(String name)
    {
        this.name = name;
    }
    public String getName()
    {
        return name;
    }
    public void setPlayers(List<Player> players)
    {
        this.players = players;
```

```
        }
        public List<Player> getPlayers()
        {
            return players;
        }
        public void setRosterCount(int n) { }
        public int getRosterCount()
        {
            return (players == null) ? 0 : players.size();
        }
    }
```

```
//TeamsPublisher.java
package Shiv21.team;
import javax.xml.ws.Endpoint;
class TeamsPublisher {
    public static void main(String[ ] args) {
        int port = 8888;
        String url = "http://localhost:" + port + "/teams";
        System.out.println("Publishing Teams on port " + port);
        Endpoint.publish(url, new Teams());
    }
}
```

```
//Teams.java
package Shiv21.team;
import java.util.List;
import javax.jws.WebService;
import javax.jws.WebMethod;
@WebService
public class Teams {
    private TeamsUtility utils;
    public Teams() {
        utils = new TeamsUtility();
        utils.make_test_teams();
    }
    @WebMethod
    public Team getTeam(String name) { return utils.getTeam(name); }
    @WebMethod
    public List<Team> getTeams() { return utils.getTeams(); }
}
```

```
//TeamsUtility.java
package Shiv21.team;
import java.util.Set;
import java.util.List;
```

```

import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;
public class TeamsUtility {
    private Map<String, Team> team_map;
    public TeamsUtility() {
        team_map = new HashMap<String, Team>();
    }
    public Team getTeam(String name) { return team_map.get(name); }
    public List<Team> getTeams() {
        List<Team> list = new ArrayList<Team>();
        Set<String> keys = team_map.keySet();
        for (String key : keys)
            list.add(team_map.get(key));
        return list;
    }
    public void make_test_teams() {
        List<Team> teams = new ArrayList<Team>();
        Player chico = new Player("Leonard Marx", "Chico");
        Player groucho = new Player("Julius Marx", "Groucho");
        Player harpo = new Player("Adolph Marx", "Harpo");
        List<Player> mb = new ArrayList<Player>();
        mb.add(chico); mb.add(groucho); mb.add(harpo);
        Team marx_brothers = new Team("Marx Brothers", mb);
        teams.add(marx_brothers);
        store_teams(teams);
    }
    private void store_teams(List<Team> teams) {
        for (Team team : teams)
            team_map.put(team.getName(), team);
    }
}

```

```

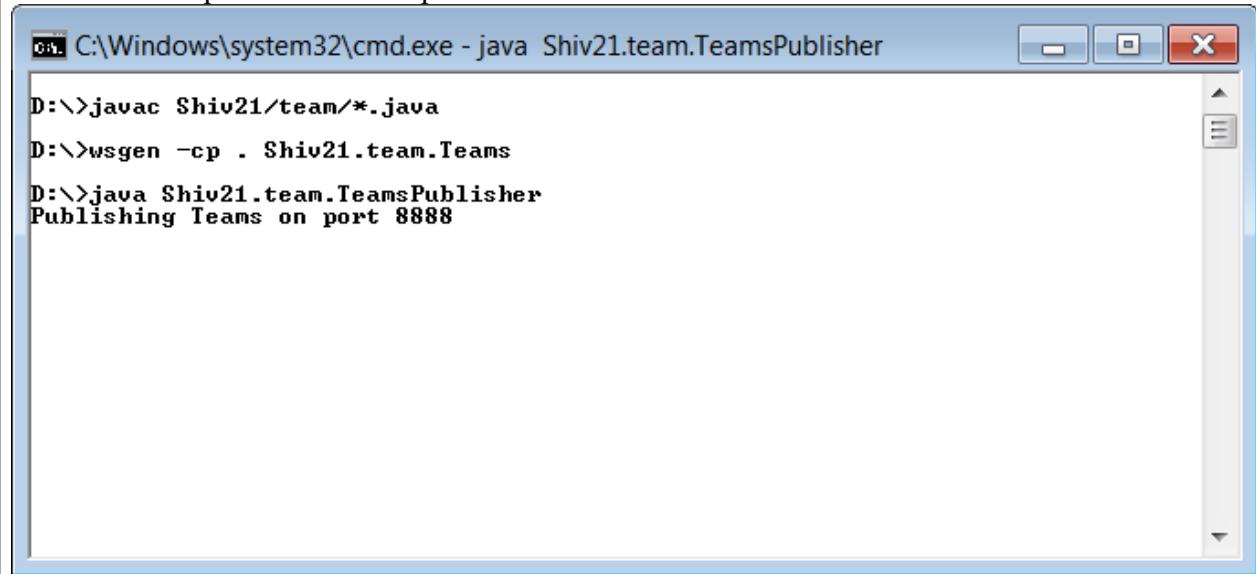
//TeamClient.java
import teamsC.TeamsService;
import teamsC.Teams;
import teamsC.Team;
import teamsC.Player;
import java.util.List;
class TeamClient {
    public static void main(String[ ] args) {
        TeamsService service = new TeamsService();
        Teams port = service.getTeamsPort();
        List<Team> teams = port.getTeams();
        for (Team team : teams) {
            System.out.println("Team name: " + team.getName() +

```

```
" (roster count: " + team.getRosterCount() + ")");
for (Player player : team.getPlayers())
System.out.println(" Player: " + player.getNickname());
}
}
}System.out.println(" Player: " + player.getNickname()); }
```

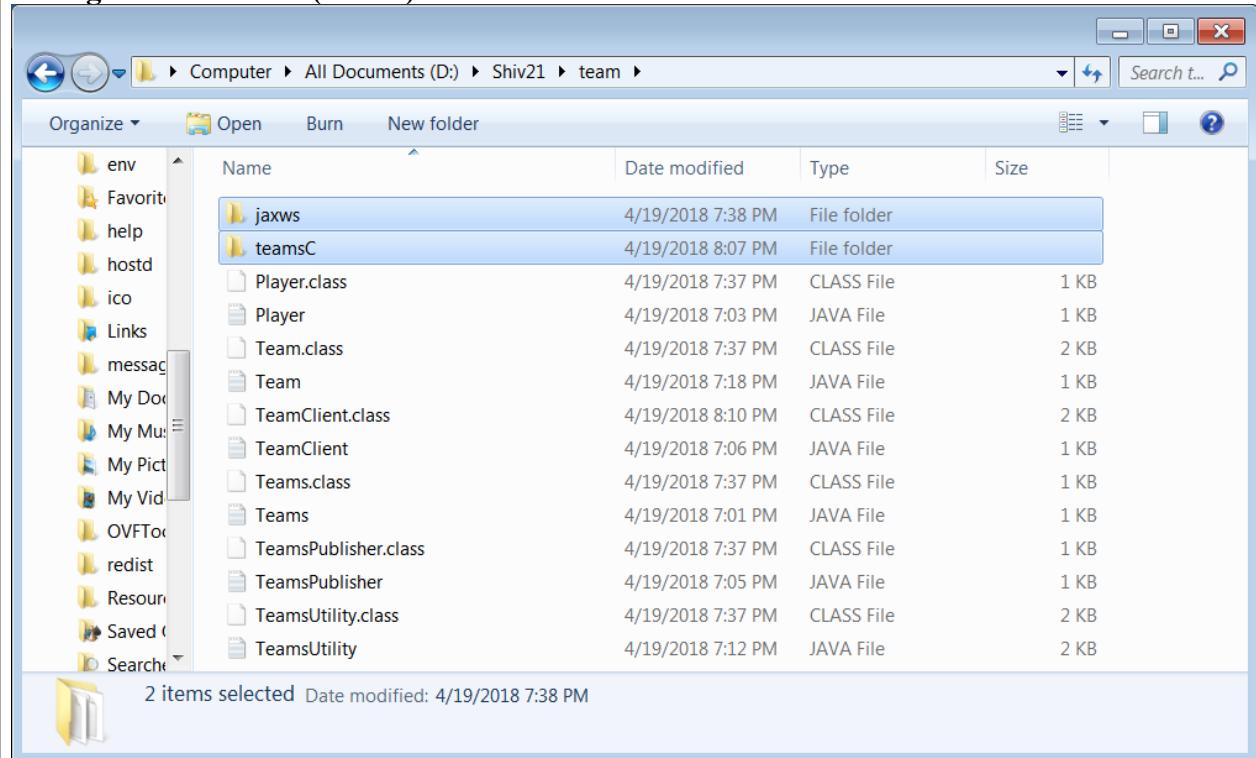
OUTPUT:

1st CMD: Compile all files except TeamClient

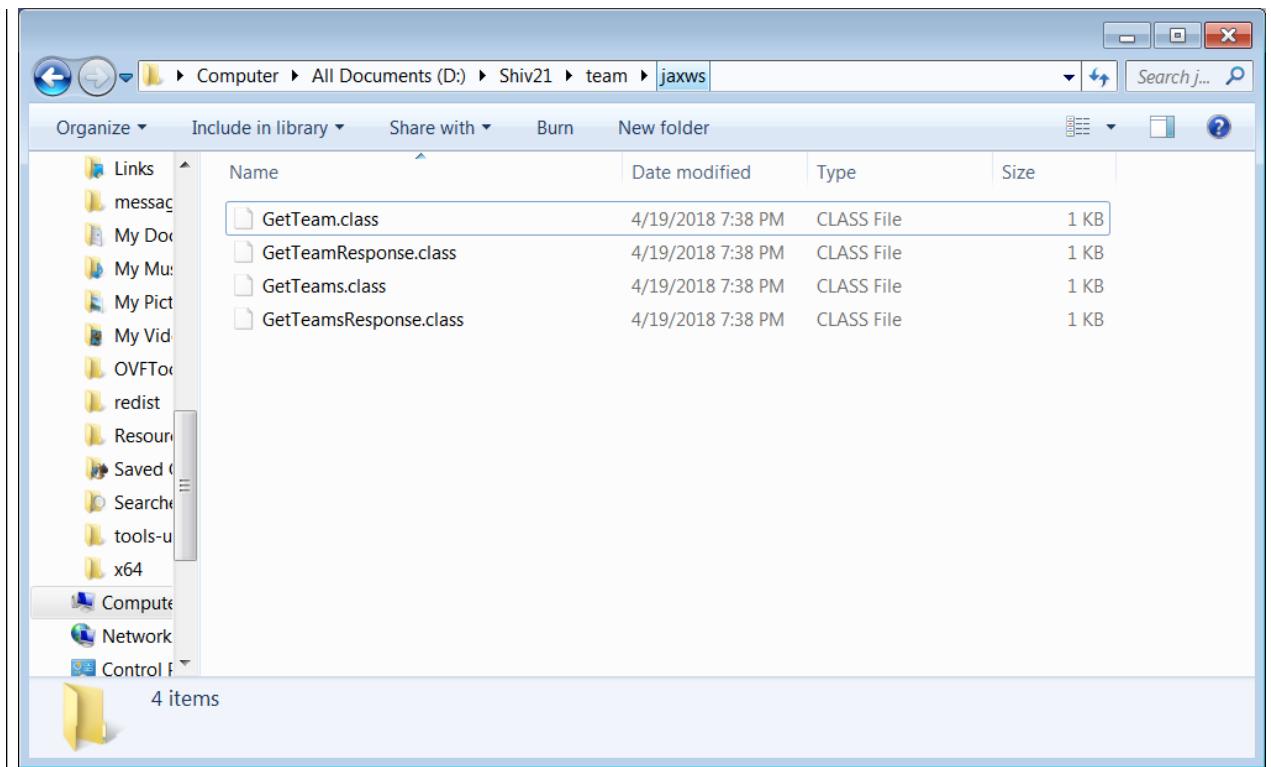


```
C:\Windows\system32\cmd.exe - java Shiv21.team.TeamsPublisher
D:\>javac Shiv21/team/*.java
D:\>wsgen -cp . Shiv21.team.Teams
D:\>java Shiv21.team.TeamsPublisher
Publishing Teams on port 8888
```

Changes in the folder (added)



Name	Date modified	Type	Size
jaxws	4/19/2018 7:38 PM	File folder	
teamsC	4/19/2018 8:07 PM	File folder	
Player.class	4/19/2018 7:37 PM	CLASS File	1 KB
Player	4/19/2018 7:03 PM	JAVA File	1 KB
Team.class	4/19/2018 7:37 PM	CLASS File	2 KB
Team	4/19/2018 7:18 PM	JAVA File	1 KB
TeamClient.class	4/19/2018 8:10 PM	CLASS File	2 KB
TeamClient	4/19/2018 7:06 PM	JAVA File	1 KB
Teams.class	4/19/2018 7:37 PM	CLASS File	1 KB
Teams	4/19/2018 7:01 PM	JAVA File	1 KB
TeamsPublisher.class	4/19/2018 7:37 PM	CLASS File	1 KB
TeamsPublisher	4/19/2018 7:05 PM	JAVA File	1 KB
TeamsUtility.class	4/19/2018 7:37 PM	CLASS File	2 KB
TeamsUtility	4/19/2018 7:12 PM	JAVA File	2 KB



2nd CMD:

A screenshot of a Windows Command Prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The command entered is 'D:\Shiv21\team>wsimport -p teamsC -keep http://localhost:8888/teams?wsdl'. The output shows the process of generating and compiling code:

```
D:\Shiv21\team>wsimport -p teamsC -keep http://localhost:8888/teams?wsdl
parsing WSDL...
Generating code...
Compiling code...
D:\Shiv21\team>_
```

Changes in the folder (added)

A screenshot of a Windows File Explorer window. The address bar shows the path: Computer > All Documents (D:) > Shiv21 > team > teamsC. The main area displays a list of files with the following details:

Name	Date modified	Type	Size
GetTeam.class	4/19/2018 8:07 PM	CLASS File	1 KB
GetTeam	4/19/2018 8:07 PM	JAVA File	2 KB
GetTeamResponse.class	4/19/2018 8:07 PM	CLASS File	1 KB
GetTeamResponse	4/19/2018 8:07 PM	JAVA File	2 KB
GetTeams.class	4/19/2018 8:07 PM	CLASS File	1 KB
GetTeams	4/19/2018 8:07 PM	JAVA File	1 KB
GetTeamsResponse.class	4/19/2018 8:07 PM	CLASS File	1 KB
GetTeamsResponse	4/19/2018 8:07 PM	JAVA File	2 KB
ObjectFactory.class	4/19/2018 8:07 PM	CLASS File	3 KB
ObjectFactory	4/19/2018 8:07 PM	JAVA File	4 KB
package-info.class	4/19/2018 8:07 PM	CLASS File	1 KB
package-info	4/19/2018 8:07 PM	JAVA File	1 KB
Player.class	4/19/2018 8:07 PM	CLASS File	1 KB
Player	4/19/2018 8:07 PM	JAVA File	2 KB
Team.class	4/19/2018 8:07 PM	CLASS File	2 KB
Team	4/19/2018 8:07 PM	JAVA File	3 KB
Teams.class	4/19/2018 8:07 PM	CLASS File	2 KB
Teams	4/19/2018 8:07 PM	JAVA File	2 KB

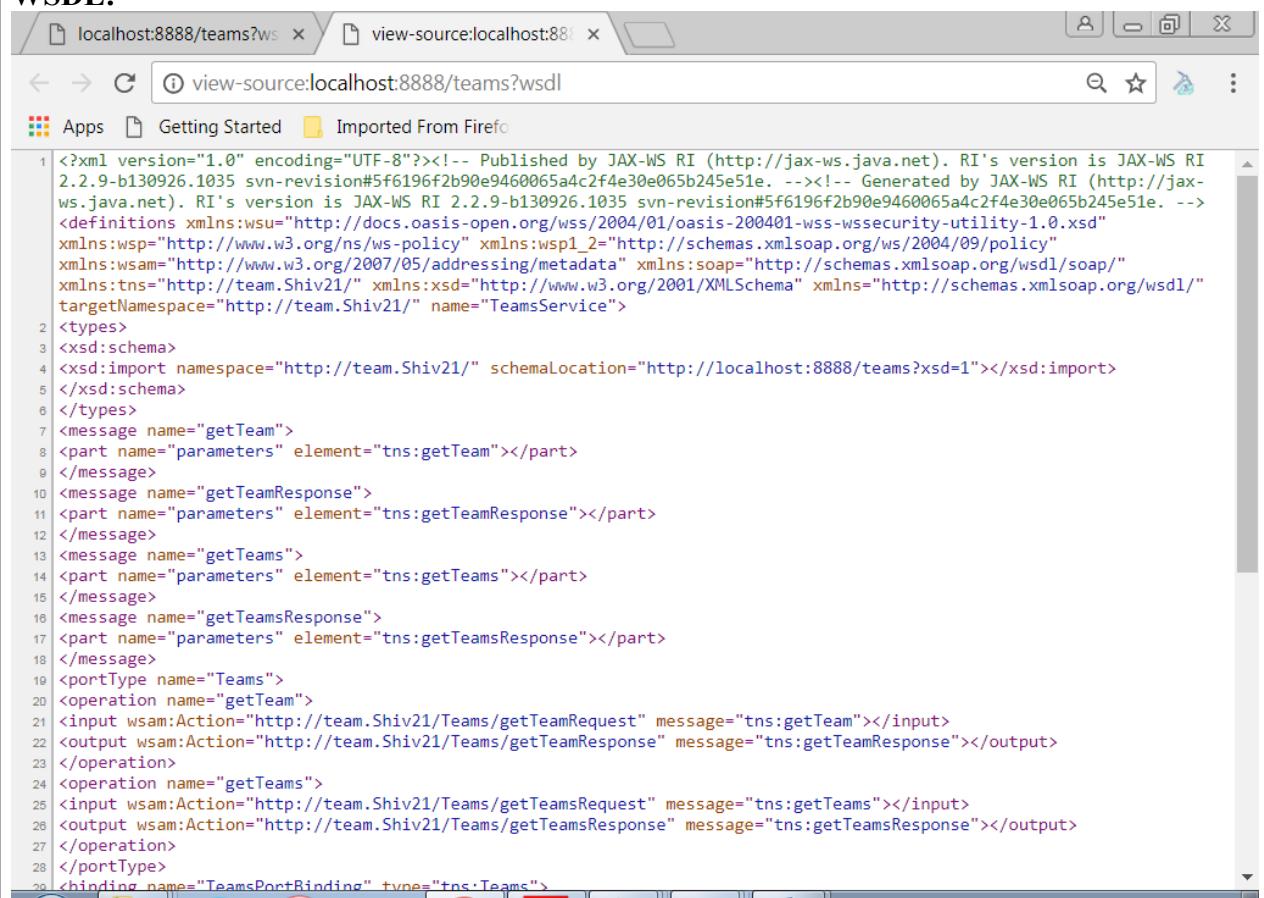
20 items

3rd CMD: Now compile and run TeamClient.java

A screenshot of a Windows Command Prompt window. The command prompt shows the following session:

```
D:\Shiv21\team>javac TeamClient.java
D:\Shiv21\team>java TeamClient
Team name: Marx Brothers (roster count: 3)
  Player: Chico
  Player: Groucho
  Player: Harpo
D:\Shiv21\team>
```

WSDL:



The screenshot shows a web browser window with two tabs: 'localhost:8888/teams?wsdl' and 'view-source:localhost:8888/teams?wsdl'. The main content area displays the WSDL XML code for a service named 'TeamsService'. The code includes definitions for types, messages, and port types, along with their corresponding operations and message details.

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. --><!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns: wsp="http://www.w3.org/ns/ws-policy" xmlns: wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns: wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns: soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns: tns="http://team.Shiv21/" xmlns: xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://team.Shiv21/" name="TeamsService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://team.Shiv21/" schemaLocation="http://localhost:8888/teams?xsd=1"></xsd:import>
    </xsd:schema>
  </types>
  <message name="getTeam">
    <part name="parameters" element="tns:getTeam"></part>
  </message>
  <message name="getTeamResponse">
    <part name="parameters" element="tns:getTeamResponse"></part>
  </message>
  <message name="getTeams">
    <part name="parameters" element="tns:getTeams"></part>
  </message>
  <message name="getTeamsResponse">
    <part name="parameters" element="tns:getTeamsResponse"></part>
  </message>
  <portType name="Teams">
    <operation name="getTeam">
      <input wsam:Action="http://team.Shiv21/Teams/getTeamRequest" message="tns:getTeam"></input>
      <output wsam:Action="http://team.Shiv21/Teams/getTeamResponse" message="tns:getTeamResponse"></output>
    </operation>
    <operation name="getTeams">
      <input wsam:Action="http://team.Shiv21/Teams/getTeamsRequest" message="tns:getTeams"></input>
      <output wsam:Action="http://team.Shiv21/Teams/getTeamsResponse" message="tns:getTeamsResponse"></output>
    </operation>
  </portType>
  <binding name="TeamsPortBinding" type="tns:Teams">
```

The screenshot shows a web browser window with two tabs open. The left tab is titled 'localhost:8888/teams?ws' and the right tab is titled 'view-source:localhost:8888/teams?wsdl'. The main content area displays the WSDL (Web Services Description Language) XML code for the 'TeamsService'.

```
20 <operation name="getTeam">
21   <input wsam:Action="http://team.Shiv21/Teams/getTeamRequest" message="tns:getTeam"></input>
22   <output wsam:Action="http://team.Shiv21/Teams/getTeamResponse" message="tns:getTeamResponse"></output>
23 </operation>
24 <operation name="getTeams">
25   <input wsam:Action="http://team.Shiv21/Teams/getTeamsRequest" message="tns:getTeams"></input>
26   <output wsam:Action="http://team.Shiv21/Teams/getTeamsResponse" message="tns:getTeamsResponse"></output>
27 </operation>
28 </portType>
29 <binding name="TeamsPortBinding" type="tns:Teams">
30   <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"></soap:binding>
31   <operation name="getTeam">
32     <soap:operation soapAction=""></soap:operation>
33     <input>
34       <soap:body use="literal"></soap:body>
35     </input>
36     <output>
37       <soap:body use="literal"></soap:body>
38     </output>
39   </operation>
40   <operation name="getTeams">
41     <soap:operation soapAction=""></soap:operation>
42     <input>
43       <soap:body use="literal"></soap:body>
44     </input>
45     <output>
46       <soap:body use="literal"></soap:body>
47     </output>
48   </operation>
49 </binding>
50 <service name="TeamsService">
51   <port name="TeamsPort" binding="tns:TeamsPortBinding">
52     <soap:address location="http://localhost:8888/teams"></soap:address>
53   </port>
54 </service>
55 </definitions>
```

Document2 - Word (Product Activation Failed)

C:\Windows\system32\cmd.exe - java Shiv21.team.TeamsPublisher

```
D:\>javac Shiv21/team/*.java
D:\>wsgen -cp . Shiv21.team.Teams
D:\>java Shiv21.team.TeamsPublisher
Publishing Teams on port 8888
```

C:\Windows\system32\cmd.exe

```
D:\>Shiv21\team>wsimport -p teamsC -keep http://localhost:8888/teams?wsdl
parsing WSDL...
```

Generating code...

Compiling code...

```
D:\>Shiv21\team>
```

Command Prompt

```
D:\>Shiv21\team>javac TeamClient.java
D:\>Shiv21\team>java TeamClient
Team name: Marx Brothers (roster count: 3)
Player: Chico
Player: Groucho
Player: Harpo
D:\>Shiv21\team>
```

PAGE 1 OF 1 1 WORD

9:01 PM
4/19/2018

PRACTICAL 3

AIM: Develop Web service in Java that returns matrix multiplication by Strassen's algorithm. Two matrices will be entered at run time by client. Server does the matrix multiplication and returns answer to client.

INPUT:

```
//StrassenServerImpl.java
package Ex01.strassen;
import javax.jws.WebService;
@WebService(endpointInterface = "Ex01.strassen.StrassenServer")
public class StrassenServerImpl implements StrassenServer {
    public int[][] multiply(int[][] A, int[][] B)
    { int n = A.length;
        int[][] R = new int[n][n];
        if (n == 1)
            R[0][0] = A[0][0] * B[0][0];
        else
        { int[][] A11 = new int[n/2][n/2];
            int[][] A12 = new int[n/2][n/2];
            int[][] A21 = new int[n/2][n/2];
            int[][] A22 = new int[n/2][n/2];
            int[][] B11 = new int[n/2][n/2];
            int[][] B12 = new int[n/2][n/2];
            int[][] B21 = new int[n/2][n/2];
            int[][] B22 = new int[n/2][n/2];
            split(A, A11, 0 , 0);
            split(A, A12, 0 , n/2);
            split(A, A21, n/2, 0);
            split(A, A22, n/2, n/2);
            split(B, B11, 0 , 0);
            split(B, B12, 0 , n/2);
            split(B, B21, n/2, 0);
            split(B, B22, n/2, n/2);
            int [][] M1 = multiply(add(A11, A22), add(B11, B22));
            int [][] M2 = multiply(add(A21, A22), B11);
            int [][] M3 = multiply(A11, sub(B12, B22));
            int [][] M4 = multiply(A22, sub(B21, B11));
            int [][] M5 = multiply(add(A11, A12), B22);
            int [][] M6 = multiply(sub(A21, A11), add(B11, B12));
            int [][] M7 = multiply(sub(A12, A22), add(B21, B22));
            int [][] C11 = add(sub(add(M1, M4), M5), M7);
            int [][] C12 = add(M3, M5);
            int [][] C21 = add(M2, M4);
            int [][] C22 = add(sub(add(M1, M3), M2), M6);
            join(C11, R, 0 , 0);
        }
    }
}
```

```

join(C12, R, 0 , n/2);
join(C21, R, n/2, 0);
join(C22, R, n/2, n/2);      }
return R;  }

public int[][] sub(int[][] A, int[][] B)
{ int n = A.length;
int[][] C = new int[n][n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        C[i][j] = A[i][j] - B[i][j];
return C;  }

public int[] add(int[][] A, int[][] B)
{   int n = A.length;
int[][] C = new int[n][n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        C[i][j] = A[i][j] + B[i][j];
return C;  }

public void split(int[][] P, int[][] C, int iB, int jB)
{   for(int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++)
    for(int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++)
        C[i1][j1] = P[i2][j2];  }

public void join(int[][] C, int[][] P, int iB, int jB)
{   for(int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++)
    for(int j1 = 0, j2 = jB; j1 < C.length; j1++, j2++)
        P[i2][j2] = C[i1][j1];  }

public String StrassenMessage(String sreq) {
    String MatrixC="";
    System.out.println("Server: StrassenMessage() invoked... ");
    System.out.println("Server: Message > " + sreq);
    int idx = sreq.indexOf(",");
    int N = Integer.parseInt(sreq.substring(0, idx));
    System.out.println("N=" + N);
    int t=0;
    int[][] A = new int[N][N];
    int[][] B = new int[N][N];
    int[][] C = new int[N][N];
    for (int i = 0; i < N; i++)
        {for (int j = 0; j < N; j++)
            {int from = sreq.indexOf(',',t);
             int to = sreq.indexOf(',', from+1);
             A[i][j] = Integer.parseInt(sreq.substring(from+1,to));
             t=to;  }  }
    for (int i = 0; i < N; i++)
        {      for (int j = 0; j < N; j++)
            {      int from = sreq.indexOf(',',t);

```

```

int to = sreq.indexOf(',', from+1);
        B[i][j] = Integer.parseInt(sreq.substring(from+1,to));
        t=to; } }
int[][] MAT = multiply(A, B);
StringBuilder MatC = new StringBuilder();
MatC.append(N+",");
for (int i = 0; i < N; i++)
{ for (int j = 0; j < N; j++)
    {MatC.append(MAT[i][j]);
if (i==N-1 && j==N-1)
    MatC.append("");
    else
    MatC.append(","); }
    System.out.println(); }
MatrixC=MatC.toString();
return(MatrixC); }}
```

```

//StrassenServerPublisher.java
package Ex01.strassen;
import javax.xml.ws.Endpoint;
public class StrassenServerPublisher {
    public static void main(String[ ] args) {
        Endpoint.publish("http://localhost:9876/strassen", new StrassenServerImpl()); }}
```

```

//StrassenServer.java
package Ex01.strassen;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
@WebService // This signals that this is a Service Endpoint Interface (SEI)
@SOAPBinding(style = Style.RPC)
public interface StrassenServer {
    @WebMethod // This signals that this method is a service operation
    String StrassenMessage(String strMsg); }}
```

```

//StrassenClient.java
package Ex01.strassen;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.URL;
class StrassenClient { public static void main(String argv[ ]) throws Exception {
    long start,finish, difference;
    if (argv.length < 1) {
System.out.println("Usage: java StrassenClient \\"N,matrix elements separated by ,\"");
        System.exit(1); }}
```

```

String strMsg = argv[0];
URL url = new URL("http://localhost:9876/strassen?wsdl");
QName qname = new QName("http://strassen.Ex01/", "StrassenServerImplService");
Service service = Service.create(url, qname);
StrassenServer eif = service.getPort(StrassenServer.class);
start=System.currentTimeMillis();
String sreq=eif.StrassenMessage(strMsg);
finish=System.currentTimeMillis();
sreq=sreq+ ",";
int idx = sreq.indexOf(",");
int N = Integer.parseInt(sreq.substring(0,idx));
int t=0;
int[][] C = new int[N][N];
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < N; j++)
    {
        int from = sreq.indexOf(',',t);
        int to = sreq.indexOf(',', from+1);
        C[i][j] = Integer.parseInt(sreq.substring(from+1,to));
        t=to;
    }
}
System.out.println("\nMatrix Multiplication is ...\\n");
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < N; j++)
        System.out.print(C[i][j] + " ");
    System.out.println();
}
difference=finish-start;
System.out.println("Time required for matrix multiplication (Using Strassen algorithm) :");
System.out.println(difference + " milli seconds");
    }
}

```

OUTPUT:

Server Side:

```
C:\Windows\system32>D:  
D:\MSCCS\CLOUD\Ex01>javac Ex01\strassen/*.java  
javac: file not found: Ex01\strassen\*.java  
Usage: javac <options> <source files>  
use -help for a list of possible options  
D:\MSCCS\CLOUD\Ex01>cd ..  
D:\MSCCS\CLOUD>javac Ex01\strassen/*.java  
D:\MSCCS\CLOUD>java Ex01.strassen.StrassenServerPublisher  
Server: StrassenMessage() invoked...  
Server: Message > 2,5,2,3,4,3,3,1,5,  
N=2
```

Client Side:

```
C:\Windows\system32>d:  
D:\MSCCS\CLOUD\Ex01>javac Ex01\strassen/StrassenClient.java  
javac: file not found: Ex01\strassen\StrassenClient.java  
Usage: javac <options> <source files>  
use -help for a list of possible options  
D:\MSCCS\CLOUD\Ex01>cd ..  
D:\MSCCS\CLOUD>javac Ex01\strassen/StrassenClient.java  
D:\MSCCS\CLOUD>java Ex01.strassen.StrassenClient 2,5,2,3,4,3,3,1,5,  
Matrix Multiplication is ...  
17 25  
13 29  
Time required for matrix multiplication (Using Strassen algorithm) :  
153 milli seconds  
D:\MSCCS\CLOUD>
```

PRACTICAL 4

AIM: Demonstrate CRUD operations with suitable database using SOAP or RESTful Web service.

INPUT:

```
//Server.java

package webservice.prac4;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
import java.util.*;

@WebService
@SOAPBinding(style = Style.RPC)
public interface Server {
    @WebMethod int insertFriend(String name, String addr);
    @WebMethod String[] getFriend(int id);
    @WebMethod boolean deleteFriend(int id);
}
```

```
//ServerImplement.java

package webservice.prac4;
import java.sql.*;
import java.util.*;
import javax.jws.WebService;
@WebService(endpointInterface = "webservice.prac4.Server")

public class ServerImplement implements Server {

    private static final String url = "jdbc:mysql://localhost:3306/library";
    private static final String user = "root";
    private static final String password = "";

    public String[] getFriend(int id) {
        String[] data = new String[2];
        try {
            Connection con = DriverManager.getConnection(url, user, password);
            Statement stmt = null;
            stmt = con.createStatement();

            String sql = "SELECT * FROM `friends` WHERE `id`=" + id;
            ResultSet rs = stmt.executeQuery(sql);
```

```

        while(rs.next()){

            data[0] = rs.getString("name");
            data[1] = rs.getString("addr");
        }
        rs.close();

    } catch (Exception e) {
        e.printStackTrace();
    }

    return data;
}

public int insertFriend(String name,String addr){

    int id = 0;
    try {
        Connection con = DriverManager.getConnection(url, user, password);

        String query = "Insert into friends (name,addr)" + " values (?, ?)";
        PreparedStatement preparedStmt =
con.prepareStatement(query,Statement.RETURN_GENERATED_KEYS);

        preparedStmt.setString (1, name);
        preparedStmt.setString (2, addr);

        preparedStmt.executeUpdate();
        ResultSet rs=preparedStmt.getGeneratedKeys();

        if(rs.next()){
            id=rs.getInt(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return id;
}

public boolean deleteFriend(int id){

    boolean stat = false;
    try {
        Connection con = DriverManager.getConnection(url, user, password);

        String query = "Delete from friends where id = ?";
        PreparedStatement preparedStmt = con.prepareStatement(query);

        preparedStmt.setInt (1, id);

```

```

        preparedStmt.execute();
        stat = true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return stat;
}
}

```

//ServerPublisher.java

```

package webservice.prac4;
import javax.xml.ws.Endpoint;
public class ServerPublisher {
    public static void main(String[ ] args) {
        Endpoint.publish("http://127.0.0.1:9876/prac4",new ServerImplement());
    }
}

```

//Client.java

```

package webservice.prac4;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.URL;
import java.util.*;

class Client {
    public static void main(String args[ ]) throws Exception {
        URL url = new URL("http://127.0.0.1:9876/prac4?wsdl");

        QName qname = new QName("http://prac4.webservice/", "ServerImplementService");
        Service service = Service.create(url, qname);
        Server eif = service.getPort(Server.class);

        Scanner s = new Scanner(System.in);

        System.out.println("1 > Add Friend");
        System.out.println("2 > Get Friend By Id");
        System.out.println("3 > Delete Friend");
        System.out.print("Enter Your Option:");

        int op = s.nextInt();
        int id = 0;
        String name="",addr="";
        switch(op){

            case 1:
                System.out.print("Enter Friends Name:");

```

```
name=s.next();
System.out.print("\nEnter Friends Address:");
addr=s.next();
int a = eif.insertFriend(name,addr);
if(a>0){
    System.out.println("Your Friend has been Added with Id:"+a);
}
break;
case 2:
System.out.print("Enter Friends ID:");
id=s.nextInt();
String[] data = eif.getFriend(id);
System.out.println("Name:"+data[0]+"\nAddress:"+data[1]);
break;
case 3:
System.out.print("Enter Friends ID To Delete:");
id=s.nextInt();
if(eif.deleteFriend(id)){
    System.out.println("Friend has been Deleted");
}
break;
default:
System.out.print("No Such Options!!!!\n Try pressing Another Key:");
break;
}
}
}
```

OUTPUT:

Server Side:

```
devil@devil-Inspiron-3542 ~/Desktop
File Edit View Search Terminal Help
devil@devil-Inspiron-3542 ~/Desktop $ javac webservice/prac4/*.java
devil@devil-Inspiron-3542 ~/Desktop $ java webservice.prac4.ServerPublisher
```

Client Side:

```
devil@devil-Inspiron-3542 ~/Desktop
File Edit View Search Terminal Help
devil@devil-Inspiron-3542 ~/Desktop $ java webservice.prac4.Client
1 > Add Friend
2 > Get Friend By Id
3 > Delete Friend
Enter Your Option:1Enter Friends Name:");
Enter Friends Name:Shubham
System.out.print("Enter Friends Address:");
Enter Friends Address:Malad
Your Friend has been Added with Id:8
devil@devil-Inspiron-3542 ~/Desktop $ java webservice.prac4.Client
1 > Add Friend
2 > Get Friend By Id
3 > Delete Friend
Enter Your Option:2
Enter Friends ID:8 Enter Friends ID:");
Name:Shubham ();
Address:Malad = eif.getFriend(id);
devil@devil-Inspiron-3542 ~/Desktop $ java webservice.prac4.Client
1 > Add Friend
2 > Get Friend By Id
3 > Delete Friend
Enter Your Option:3
Enter Friends ID To Delete:8
Friend has been Deleted("Friend has been Deleted");
```

Practical 5

Aim: Create a Virtual Private Cloud in AWS.

Steps:

1) Create VPC

The screenshot shows the AWS VPC console interface. At the top, there's a navigation bar with links like 'Most Visited', 'Web Development Co...', 'Sean Halpin - a web d...', 'Free Vector Icons and ...', and 'Getting Started'. Below the navigation is a search bar and a 'Create VPC' button. The main area displays a table titled 'Your VPCs (1/1) Info' with one row. The row details a VPC named 'vpc-053939be11f7fd0e' with a status of 'Available', IPv4 CIDR '172.31.0.0/16', and a DHCP options set 'dopt-0aa8467e199...'. On the left, a sidebar titled 'VIRTUAL PRIVATE CLOUD' lists various services: Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The 'Your VPCs' section is highlighted. At the bottom, a breadcrumb navigation shows 'VPC > Your VPCs > Create VPC'. A large modal window titled 'Create VPC' is open, showing the 'VPC settings' tab. It includes fields for 'Resources to create' (set to 'VPC only'), 'Name tag - optional' (set to 'my-demo-vpc'), 'IPv4 CIDR block' (set to '10.0.0.0/16'), 'IPv6 CIDR block' (set to 'No IPv6 CIDR block'), and 'Tenancy' (set to 'Shared').

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> X	<input type="text" value="my-demo-vpc"/> X

Add new tag

You can add 49 more tags.

Cancel **Create VPC**

Internet gateways (1/1) Info

igw-0031d09380fae79fd

Name	Internet gateway ID	State	VPC ID	Owner
igw-0031d09380fae79fd	Attached	vpc-053939be11f7fdc0e	091350768449	

igw-0031d09380fae79fd

Details **Tags**

Internet gateway ID	State	VPC ID	Owner
igw-0031d09380fae79fd	Attached	vpc-053939be11f7fdc0e	091350768449

2) Create Internet Gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key



Value - optional

RemoveAdd new tag

You can add 49 more tags.

CancelCreate internet gateway

Actions ▲

[Attach to VPC](#)[Detach from VPC](#)[Manage tags](#)[Delete](#)

Screenshot of the AWS VPC Attach to VPC interface. The search bar shows "Search for services, features, blogs, docs, and more [Alt+S]". The breadcrumb navigation shows "VPC > Internet gateways > Attach to VPC (igw-05aeb0bb77a61ff7e)". The main title is "Attach to VPC (igw-05aeb0bb77a61ff7e) Info". A sub-section titled "VPC" contains the instruction "Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below." A search bar displays "vpc-0e597d60bccf35177". A link "▶ AWS Command Line Interface command" is present. At the bottom are "Cancel" and "Attach internet gateway" buttons.

Screenshot of the AWS Subnets list interface. The search bar shows "Search for services, features, blogs, docs, and more [Alt+S]". The left sidebar shows "Subnets (6) Info" under the "Subnets" section. The main table lists six subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Available IPv6 addresses
-	subnet-0e010d8ad173f804c	Available	vpc-053939be11f7fdc0e	172.31.64.0/20	-	4091	us-east-1
-	subnet-03461d08bb9bec...	Available	vpc-053939be11f7fdc0e	172.31.80.0/20	-	4091	us-east-1
-	subnet-0215ce0e5969b4c...	Available	vpc-053939be11f7fdc0e	172.31.16.0/20	-	4091	us-east-1
-	subnet-0e8c6616b50cd99c0	Available	vpc-053939be11f7fdc0e	172.31.32.0/20	-	4091	us-east-1
-	subnet-0cf09282d899c51d	Available	vpc-053939be11f7fdc0e	172.31.0.0/20	-	4091	us-east-1
-	subnet-0f0971b000071df2d	Available	vpc-053939be11f7fdc0e	172.31.48.0/20	-	4091	us-east-1

3) Create Subnet.

4) Configure Subnet for public and private instance.

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

Associated VPC CIDRs

IPv4 CIDRs

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block Info

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Public-Subnet"/>

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel **Create subnet**

https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#subnets:Subnet

You have successfully created 1 subnet: subnet-0e68b26aaafc5fcc2a

Subnets (1) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Available IPv6 addresses
Public-Subnet	subnet-0e68b26aaafc5fcc2a	Available	vpc-0e597d60bccf35177 ...	10.0.1.0/24	-	251	us-east-1

Select a subnet

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

[X](#)

▼ Tags - optional

Key	Value - optional	Remove
<input type="text" value="Name"/>	<input type="text" value="Private-Subnet"/> X	Remove

[Add new tag](#)

You can add 49 more tags.

[Remove](#)

[Add new subnet](#)

[Cancel](#) [Create subnet](#)

5) Create Routes in routing Table.

[AWS services](#) [Search for services, features, blogs, docs, and more](#) [Alt+G]

[New VPC Experience](#) Tell us what you think

VPC Dashboard [EC2 Global View](#)

Filter by VPC [Select a VPC](#)

VIRTUAL PRIVATE CLOUD

- Your VPCs
- Subnets
- Route Tables**
- Internet Gateways
- Egress Only Internet Gateways
- Carrier Gateways
- DHCP Options Sets
- Elastic IPs
- Managed Prefix Lists

Route tables (1/2) [Info](#)

[Actions](#) [Create route table](#)

Name	Route table ID	Explicit subnet assoc...	Edge associat...	Min...	VPC	Owner ID
rtb-0fa7df3b374198...	3...	-	-	Yes	vpc-053939be11f7fd0e	091350768449
rtb-0fa7df3b374198...	3...	-	-	Yes	vpc-0e597d60bccf5177	091350768449

[Edit Name](#) [Cancel](#) [Save](#)

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	<input type="text" value="local"/> X	Active	No
0.0.0.0/0	<input type="text" value="igw-05ae0bb77a61#7d"/> X	-	No

[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

VPC > Route tables > rtb-0711c1c65605f831d > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
<input type="checkbox"/> Private-Subnet	subnet-0fd567343db4df186	10.0.2.0/24	-	Main (rtb-0711c1c65605f831d / Public-RT)	
<input checked="" type="checkbox"/> Public-Subnet	subnet-0e68b26aafc5fc2a	10.0.1.0/24	-	Main (rtb-0711c1c65605f831d / Public-RT)	

Selected subnets

- subnet-0e68b26aafc5fc2a / Public-Subnet

VPC > Route tables > rtb-0ghe739e5a400145 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
<input checked="" type="checkbox"/> Private-Subnet	subnet-0fd567343db4df186	10.0.2.0/24	-	Main (rtb-0711c1c65605f831d / Public-RT)	
<input type="checkbox"/> Public-Subnet	subnet-0e68b26aafc5fc2a	10.0.1.0/24	-	rtb-0711c1c65605f831d / Public-RT	

Selected subnets

- subnet-0fd567343db4df186 / Private-Subnet

6) Create Security groups.

New VPC Experience Tell us what you think

VPC Dashboard EC2 Global View Filter by VPC:

VIRTUAL PRIVATE CLOUD Your VPCs

Security Groups (1/2) Info

Name	Security group ID	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
<input checked="" type="checkbox"/> default-sg	sg-08ba9205ba0529325	vpc-0e597d60bcff5177	default VPC secur...	091350768449	1 Permission entry	1 Permission entry
-	sg-0c9a8fe12c941...	vpc-053939be11f7fd0e	default VPC secur...	091350768449	1 Permission entry	1 Permission entry

VPC > Security Groups > sg-08ba9205ba0529325 - default > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-03f688ff35971d1ea	All traffic	All	All	Custom	<input type="text" value="sg-08ba9205ba0529325"/> <input type="button" value="Delete"/>
-	All traffic	All	All	My IP	<input type="text" value="103.58.152.185/32"/> <input type="button" value="Delete"/>

7) Create a public instance.

i We noticed that you didn't select a key pair. If you want to be able to connect to your instance it is recommended that you create one.

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Create new key pair

Proceed without key pair

Key pair name

new-pair-key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

.pem

For use with OpenSSH

.ppk

8) Create a private instance.

We are replacing this launch experience with a new launch experience, which we will continue to improve based on your feedback. Opt-in to the new experience by selecting the button on the right and give us feedback. For now you can still opt out once you have tried it. [Opt-in to the new experience](#)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review Cancel and Exit

Step 1: Choose an Amazon Machine Image (AMI)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

My AMIs	Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-03edeff12e34e59e (64-bit x86) / ami-0482730ee38e3f893 (64-bit Arm)
AWS Marketplace	Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Community AMIs	Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-0a3c14e1ddbe7f23c (64-bit x86) / ami-04ee85818705ec002 (64-bit Arm) Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
<input type="checkbox"/> Free tier only	macOS Monterey 12.2.1 - ami-03f795d99e0a6256e

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~1 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-0e597d60bccf35177 | my-demo-vpc Create new VPC

Subnet: subnet-0e68b26aaafc5fc2a | Public-Subnet | us-east-1 Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Hostname type: Use subnet setting (IP name)

DNS Hostname:

- Enable IP name IPv4 (A record) DNS requests
- Enable resource-based IPv4 (A record) DNS requests
- Enable resource-based IPv6 (AAAA record) DNS requests

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory Create new directory

IAM role: None Create new IAM role

Cancel Previous Review and Launch Next: Add Storage

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-08ba9205ba0529325	default	default VPC security group	Copy to new

Inbound rules for sg-08ba9205ba0529325 (Selected security groups: sg-08ba9205ba0529325)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	103.58.152.185/32	sg-08ba9205ba0529325 (default)
All traffic	All	All	sg-08ba9205ba0529325	

Cancel Previous Review and Launch

9) Ping public-ec2 instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and a detailed Instances section. The Instances section is expanded, showing various filters and lists. In the main content area, an instance named 'i-03049d297c3ea3f29' is selected. To the right of the instance details, a terminal window is open with the command 'ping 18.234.134.72'. The terminal output shows four successful replies from the instance's public IP address.

10) Ping private-ec2 instance.

This screenshot is similar to the previous one but shows a private EC2 instance named 'i-001ab0f111d6056f5'. The terminal window on the right shows a ping command to the instance's private IP address, 54.234.148.104. The terminal output shows four successful replies from the instance's private IP address.

Conclusion: Virtual private cloud implemented successfully.

PRACTICAL 6

AIM: Develop WCF service returning response in JSON type.

INPUT:

```
//MyService.svc.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace MyWCFService1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "MyService1" in code,
    svc and config file together.
    // NOTE: In order to launch WCF Test Client for testing this service, please select MyService1.svc or
    MyService1.svc.cs at the Solution Explorer and start debugging.

    public class MyService1 : IMyService1
    {
        public string Message(string name)
        {
            return "Hello" + " " + name;
        }

        public string Prod_id(string id)
        {
            return "Your product id is" + " " + id;
        }
    }
}
```

```
//IMyService1.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace MyWCFService1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "MyService1" in code,
    svc and config file together.
    // NOTE: In order to launch WCF Test Client for testing this service, please select MyService1.svc or
    MyService1.svc.cs at the Solution Explorer and start debugging.

    public class MyService1 : IMyService1
    {
        public string Message(string name)
        {
            return "Hello" + " " + name;
        }

        public string Prod_id(string id)
        {
            return "Your product id is" + " " + id;
        }
    }
}
```

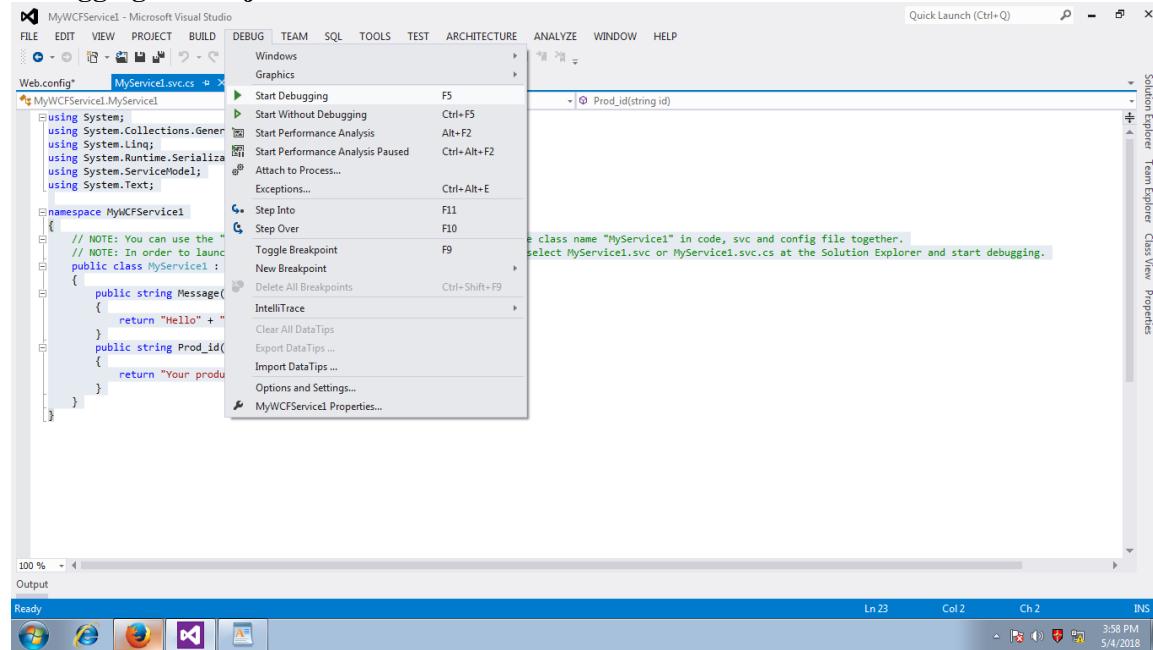
```

//Web.config
<?xml version="1.0"?>
<configuration>
<appSettings>
<add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
</appSettings>
<system.web>
<compilation debug="true" targetFramework="4.5" />
<httpRuntime targetFramework="4.5"/>
</system.web>
<system.serviceModel>
<behaviors>
<serviceBehaviors>
<behavior name ="ServiceBehavior">
    <!-- To avoid disclosing metadata information, set the values below to false before deployment -->
    <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"/>
    <!-- To receive exception details in faults for debugging purposes, set the value below to true. Set to false before deployment to avoid disclosing exception information -->
    <serviceDebug includeExceptionDetailInFaults="false"/>
</behavior>
</serviceBehaviors>
<endpointBehaviors >
<behavior name="web">
    <webHttp />
</behavior>
</endpointBehaviors>
</behaviors>
<services>
<service name="MyWCFService1.MyService1" behaviorConfiguration ="ServiceBehavior">
    <endpoint binding ="webHttpBinding" contract ="MyWCFService1.IMyService1" behaviorConfiguration ="web">
        </endpoint>
    </service>
</services>
<protocolMapping>
    <add binding="basicHttpsBinding" scheme="https" />
</protocolMapping>
<serviceHostingEnvironment aspNetCompatibilityEnabled="true" multipleSiteBindingsEnabled="true" />
</system.serviceModel>
<system.webServer>
<modules runAllManagedModulesForAllRequests="true"/>
<!--
    To browse web app root directory during debugging, set the value below to true.
    Set to false before deployment to avoid disclosing web app folder information.
-->
<directoryBrowse enabled="true"/>
</system.webServer>
</configuration>

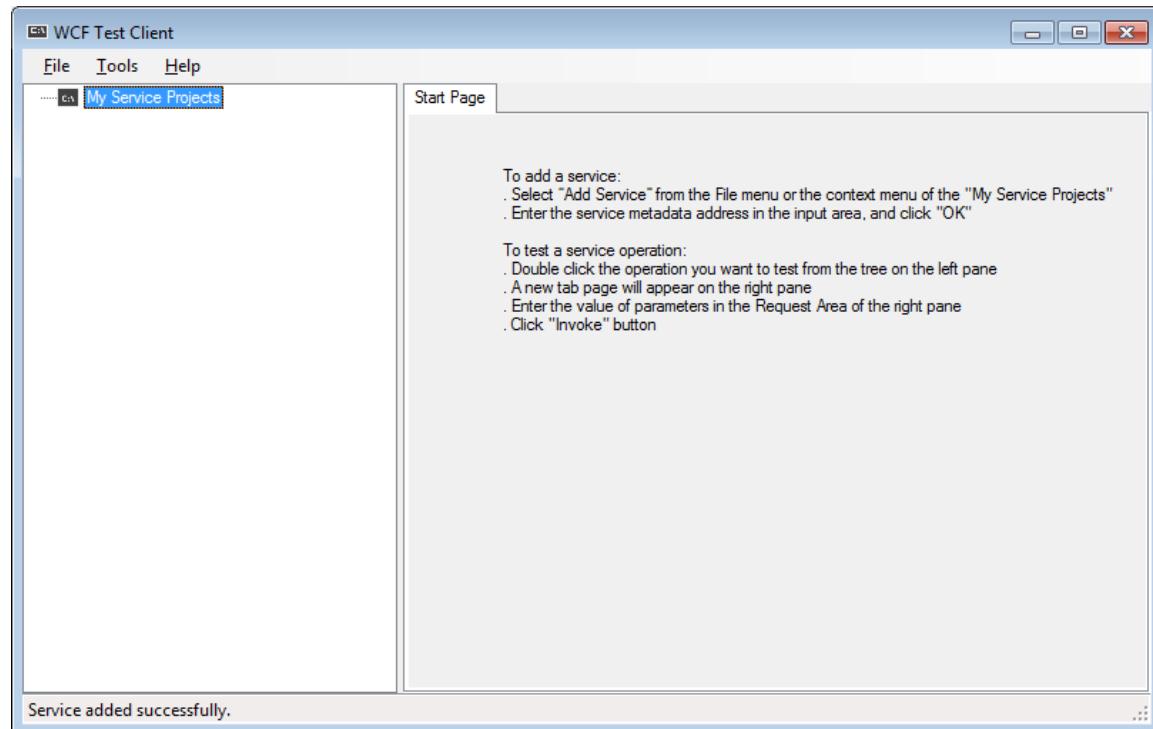
```

OUTPUT:

Debugging the Project:



Adding The Service:





MyService1 Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:7549/MyService1.svc?wsdl
```

You can also access the service description as a single file:

```
http://localhost:7549/MyService1.svc?singleWSDL
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        HelloClient client = new HelloClient();

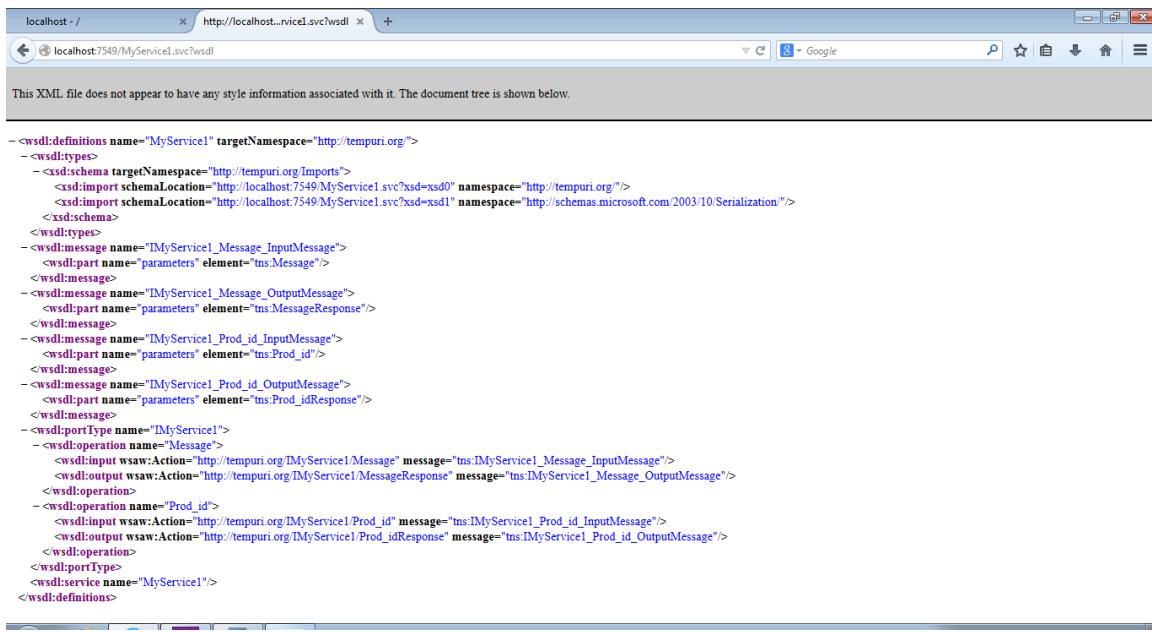
        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

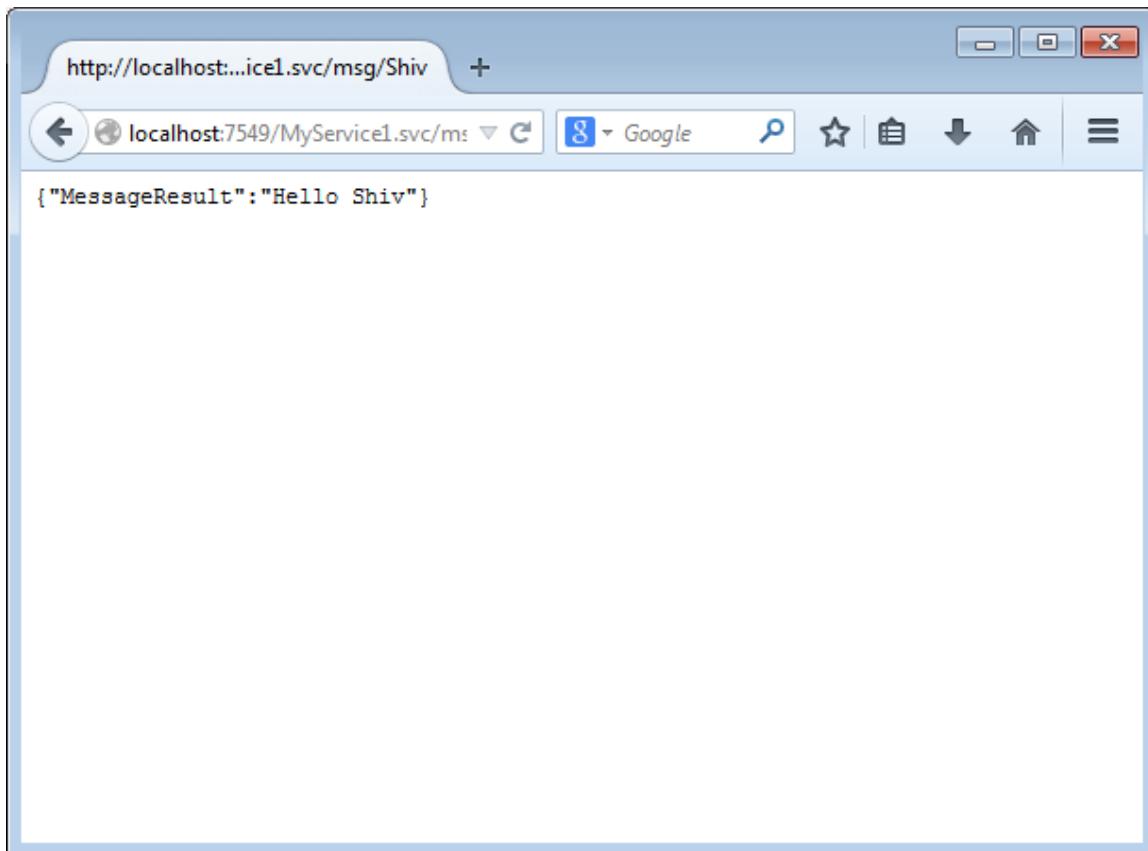
```
Class Test
    Shared Sub Main()
        Dim client As HelloClient = New HelloClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions name="MyService1" targetNamespace="http://tempuri.org">
  <wsdl:types>
    <xsd:schema targetNamespace="http://tempuri.org/Imports">
      <xsd:import schemaLocation="http://localhost:7549/MyService1.svc?xsd=xsd0" namespace="http://tempuri.org"/>
      <xsd:import schemaLocation="http://localhost:7549/MyService1.svc?xsd=xsd1" namespace="http://schemas.microsoft.com/2003/10/Serialization"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="IMyService1_Message_InputMessage">
    <wsdl:part name="parameters" element="tns:Message"/>
  </wsdl:message>
  <wsdl:message name="IMyService1_Message_OutputMessage">
    <wsdl:part name="parameters" element="tns:MessageResponse"/>
  </wsdl:message>
  <wsdl:message name="IMyService1_Prod_id_InputMessage">
    <wsdl:part name="parameters" element="tns:Prod_id"/>
  </wsdl:message>
  <wsdl:message name="IMyService1_Prod_id_OutputMessage">
    <wsdl:part name="parameters" element="tns:Prod_idResponse"/>
  </wsdl:message>
  <wsdl:portType name="IMyService1">
    <wsdl:operation name="Message">
      <wsdl:input wsaw:Action="http://tempuri.org/IMyService1/Message" message="tns:IMyService1_Message_InputMessage"/>
      <wsdl:output wsaw:Action="http://tempuri.org/IMyService1/MessageResponse" message="tns:IMyService1_Message_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="Prod_id">
      <wsdl:input wsaw:Action="http://tempuri.org/IMyService1/Prod_id" message="tns:IMyService1_Prod_id_InputMessage"/>
      <wsdl:output wsaw:Action="http://tempuri.org/IMyService1/Prod_idResponse" message="tns:IMyService1_Prod_id_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:service name="MyService1">
  </wsdl:service>
</wsdl:definitions>
```



Practical 7

Aim: Perform IAM (Identity and Access Management) Operations in AWS.

A) MFA

Steps:

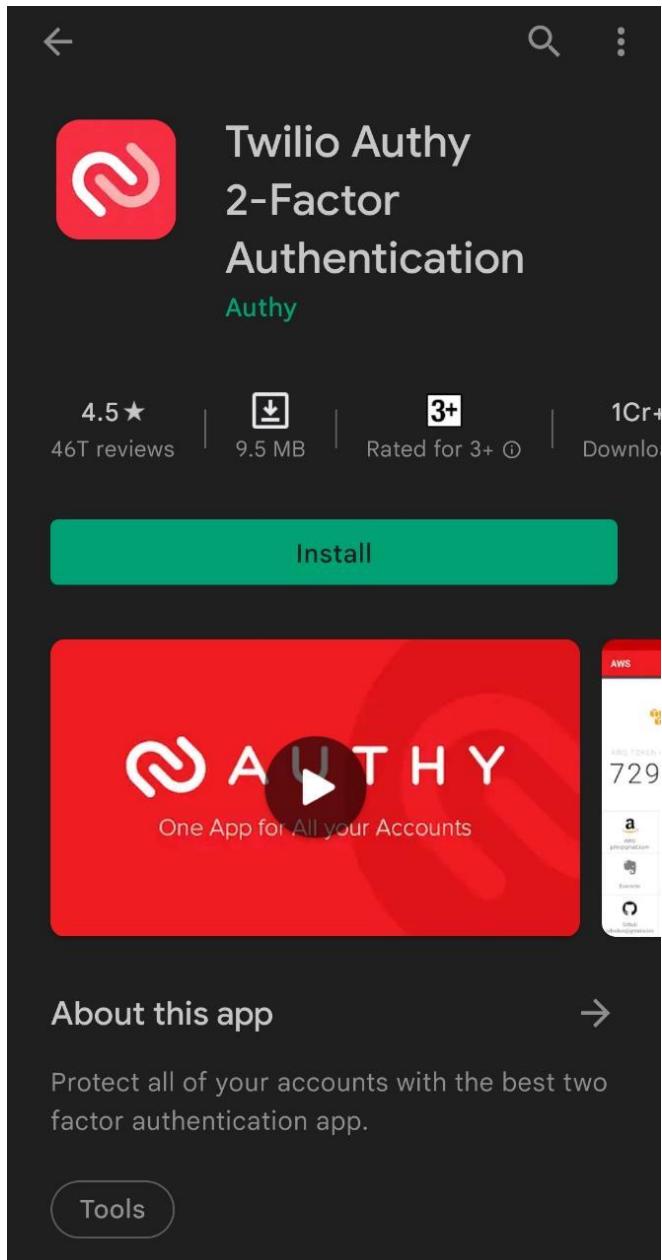
- 1) Log in to your Aws Console.
- 2) Search AWS inside search bar.
- 3) On IAM Dashboard Click-On Add MFA

The screenshot shows the AWS IAM Dashboard. At the top, there is a blue banner with the text "Introducing the new IAM dashboard experience. We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think." Below the banner, the main title is "IAM dashboard". Under "Security recommendations", there is a red notification box with the heading "Add MFA for root user" and a link to "Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.". Another green notification box says "Root user has no active access keys" with a link to "Using access keys attached to an IAM user instead of the root user improves security.". In the "AWS Account" section, there are fields for "Account ID" and "Account Alias", both of which are highlighted with blue circles. A "Create" button is also visible. The "IAM resources" section shows counts for User groups (0), Users (0), Roles (3), Policies (0), and Identity providers (0). The "What's new" section has a "View all" link. On the right side, there is a "Quick Links" sidebar with a "My security credentials" link, a "0" badge, and links for "Privacy", "Terms", and "Cookie preferences".

- 4) Click on Activate MFA

The screenshot shows the "Your Security Credentials" page. The left sidebar lists navigation options: Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, and Service control policies (SCPs). The main content area is titled "Your Security Credentials" and contains the text: "Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#). To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference." Below this, there are sections for "Password" (which is collapsed), "Multi-factor authentication (MFA)" (which is expanded), "Access keys (access key ID and secret access key)", "CloudFront key pairs", "X.509 certificate", and "Account identifiers". A prominent blue "Activate MFA" button is located within the expanded "Multi-factor authentication (MFA)" section.

5) Install Authy From Android Play Store



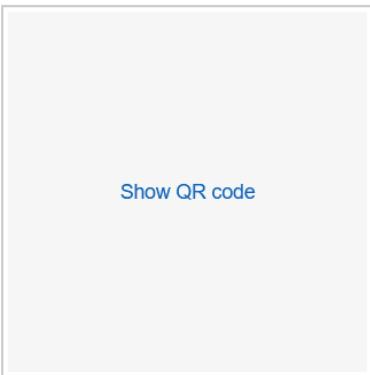
6) Scan QR Code from Authy.

7) Enter 2 consecutive MFA codes.

Set up virtual MFA device ×

1. Install a compatible app on your mobile device or computer
See a [list of compatible applications](#)

2. Use your virtual MFA app and your device's camera to scan the QR code


[Show QR code](#)

Alternatively, you can type the secret key. [Show secret key](#)

3. Type two consecutive MFA codes below

[Cancel](#) [Previous](#) [Assign MFA](#)

8) Click on Assign MFA

Set up virtual MFA device ×



Alternatively, you can type the secret key. [Show secret key](#)

3. Type two consecutive MFA codes below

MFA code 1	769518
------------	--------

MFA code 2	794041
------------	--------

[Cancel](#) [Previous](#) [Assign MFA](#)

Screenshot of the AWS IAM Security Credentials page.

Identity and Access Management (IAM)

- Dashboard**
- Access management**
 - User groups
 - Users
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Access reports**
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

Multi-factor authentication (MFA)

Use MFA to increase the security of your AWS environments. Signing in to MFA-protected accounts requires a user name, password, and an authentication code from an MFA device.

Device type	Serial number	Actions
Virtual	arn:aws:iam::[REDACTED]ot-account-mfa-device	Manage

Access keys (access key ID and secret access key)

CloudFront key pairs

X.509 certificate

Account identifiers

9) Multifactor Authentication has been successfully implemented.

Screenshot of the AWS Multi-factor authentication sign-in page.

aws

Multi-factor authentication

Your account is secured using multi-factor authentication (MFA). To finish signing in, turn on or view your MFA device and type the authentication code below.

Email address: zeeWITHZ@gmail.com

MFA code

Submit

Troubleshoot MFA

Cancel

Amazon Lightsail

Lightsail is the easiest way to get started on AWS

Learn more »

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

English ▾

Screenshot of the AWS IAM Dashboard.

Identity and Access Management (IAM)

IAM dashboard

Security recommendations

- Root user has MFA** Having multi-factor authentication (MFA) for the root user improves security for this account.
- Root user has no active access keys** Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	3	0	0

What's new Updates for features in IAM

AWS Account

- Account ID
- Account Alias
- Create
- Sign-in URL for IAM users in this account https://[REDACTED].signin.aws.amazon.com/console

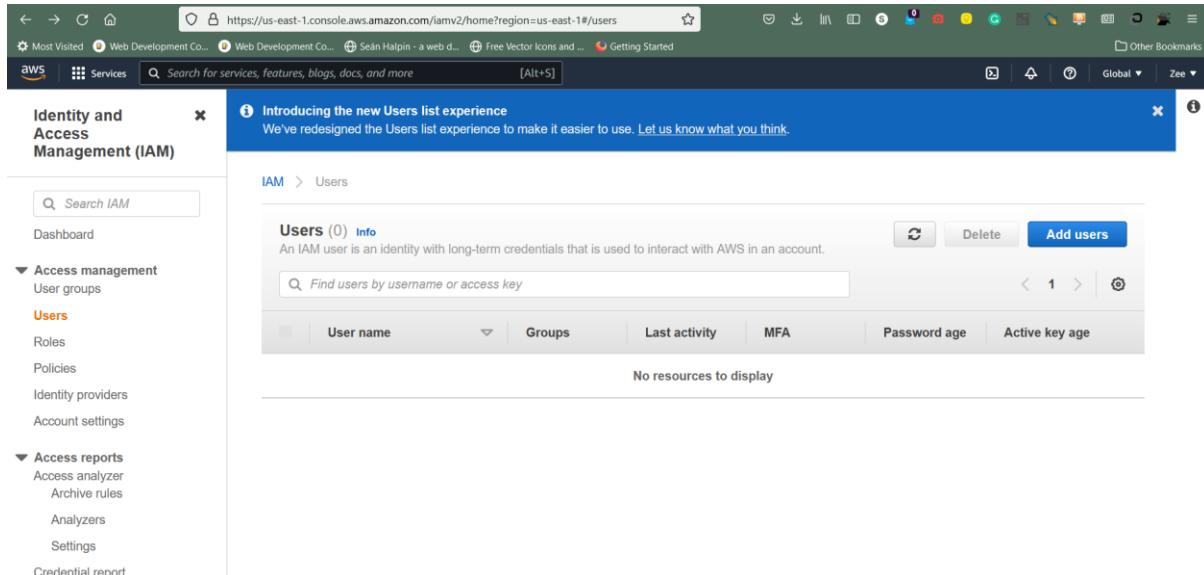
Quick Links

My security credentials Manage your access keys, multi-factor authentication (MFA) and other credentials.

B) Create users

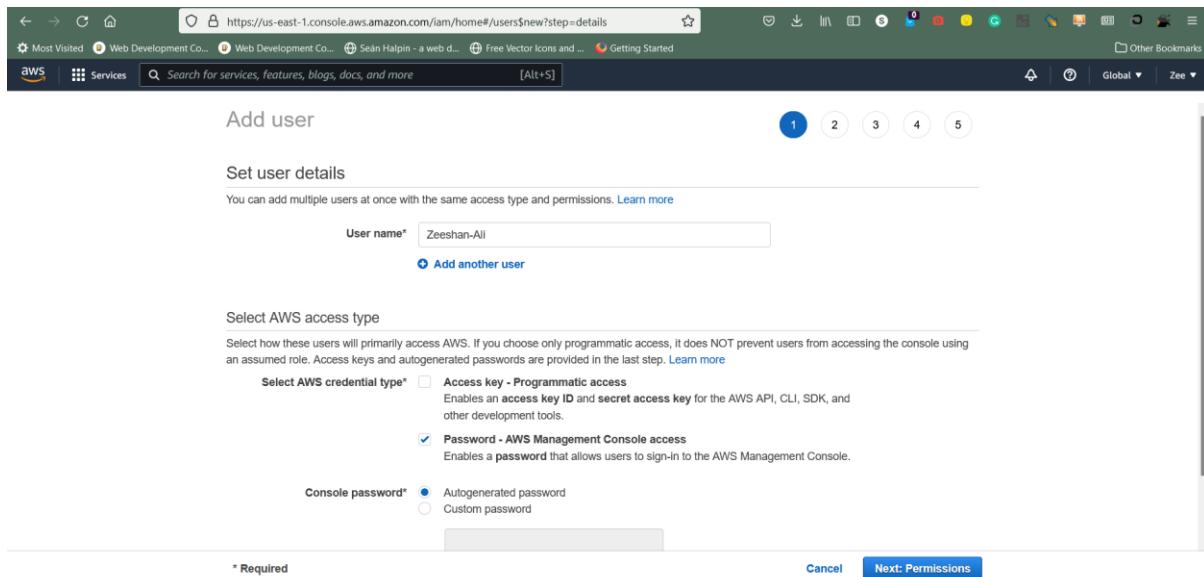
Steps:

1) Click-On Add Users



The screenshot shows the AWS IAM service in the AWS Management Console. The left sidebar is titled 'Identity and Access Management (IAM)' and includes sections for Dashboard, Access management (with 'Users' selected), Policies, Identity providers, Account settings, and Access reports. The main content area is titled 'Users (0) Info' and contains a message about the redesigned user list experience. Below this is a search bar labeled 'Find users by username or access key'. A table header row shows columns for User name, Groups, Last activity, MFA, Password age, and Active key age. A message at the bottom states 'No resources to display'.

2) Add User Name



The screenshot shows the 'Add user' wizard, step 1: Set user details. The title is 'Add user' and there are five steps indicated at the top right. The first step is highlighted. The form has a section for 'Set user details' with a note that you can add multiple users at once. It includes a 'User name*' field containing 'Zeeshan-All' and a link to 'Add another user'. Below this is a section for 'Select AWS access type' with a note about programmmatic access. It shows two options: 'Access key - Programmatic access' (unchecked) and 'Password - AWS Management Console access' (checked). The checked option is described as enabling a password for sign-in. There is also a 'Console password*' field with radio buttons for 'Autogenerated password' (selected) and 'Custom password'. At the bottom are 'Required' notes, a 'Cancel' button, and a 'Next: Permissions' button.

3) Set Permissions

Add user

1 2 3 4 5

Set permissions

Add user to group (selected)

Copy permissions from existing user

Attach existing policies directly

Get started with groups

You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

Create group

Cancel Previous Next: Tags

▼ Set permissions boundary

Set a permissions boundary to control the maximum permissions this user can have. This is an advanced feature used to delegate permission management to others. [Learn more](#)

- Create user without a permissions boundary
 Use a permissions boundary to control the maximum user permissions

Cancel Previous Next: Tags

4) Add Tags

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
Web-Development	Fullstack	x
Add new key		

You can add 49 more tags.

Cancel Previous Next: Tags

5) Click-On Create User

The screenshot shows the 'Add user' review step in the AWS IAM console. At the top, there are five numbered tabs: 1, 2, 3, 4 (highlighted in blue), and 5. The main content area is titled 'Review' with the sub-section 'User details'. It lists the following configuration:

User name	Zeeshan-Ali
AWS access type	AWS Management Console access - with a password
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Below this is the 'Permissions summary' section, which states: 'The user shown above will be added to the following groups.' A table shows one group entry:

Type	Name
Managed policy	IAMUserChangePassword

At the bottom right are 'Cancel', 'Previous', and 'Create user' buttons.

6) User has been created successfully

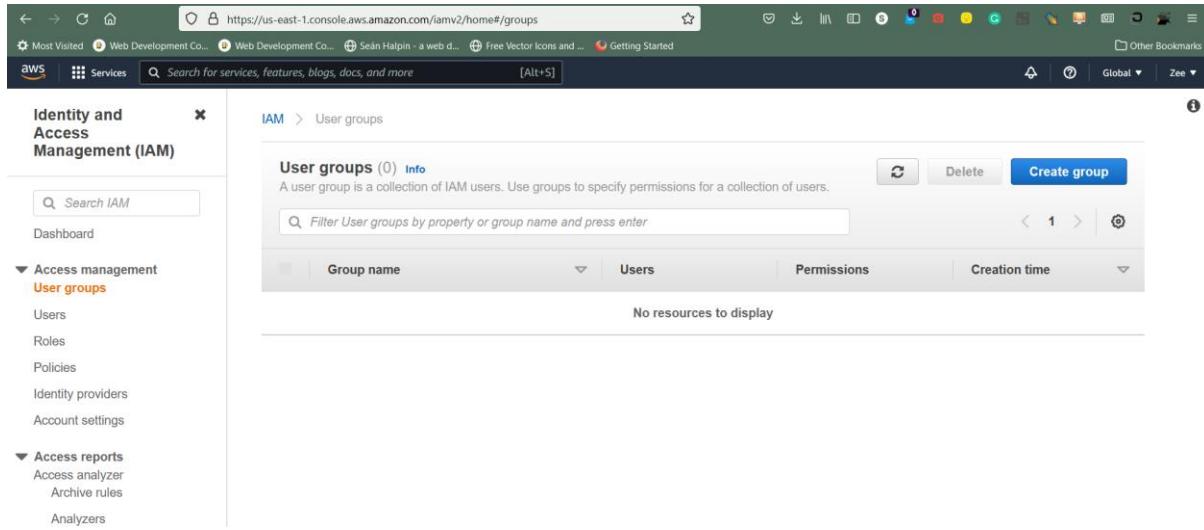
The screenshot shows the 'Add user' success step in the AWS IAM console. At the top, there are five numbered tabs: 1, 2, 3, 4, and 5 (highlighted in blue). The main content area features a green 'Success' box containing the message: 'You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.' Below this message is a link: 'Users with AWS Management Console access can sign-in at: <https://091350768449.sigin.aws.amazon.com/console>'.

Below the success message is a 'Download .csv' button. To the right is a table with two columns: 'User' and 'Email login instructions'. The table contains one row for 'Zeeshan-Ali'. On the far right of the table is a 'Send email' button.

C) User Groups

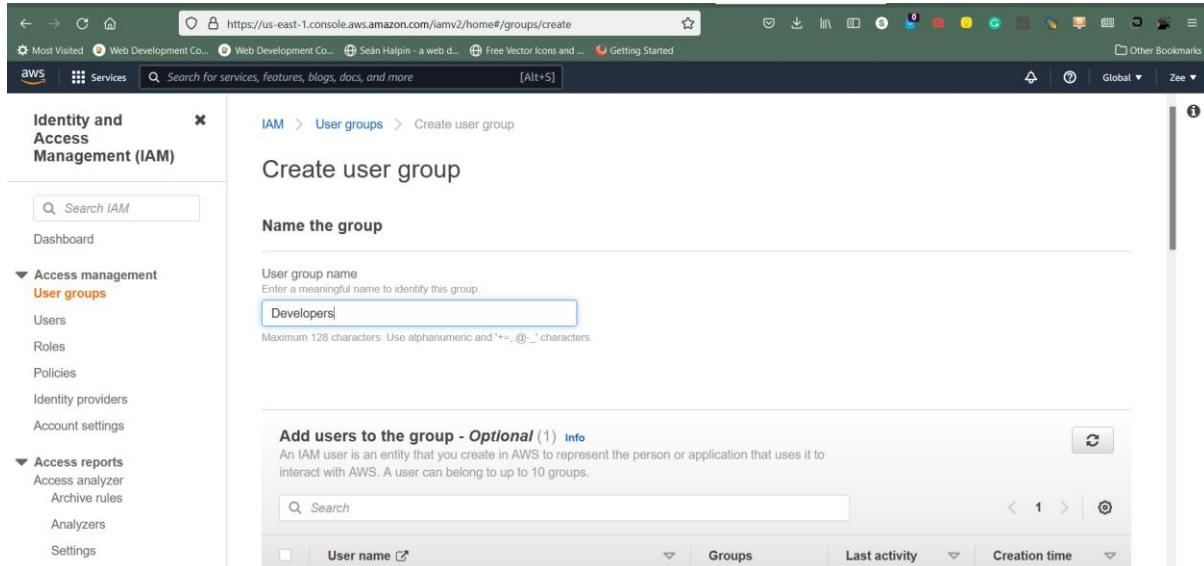
Steps:

1) Click-On Create Groups.



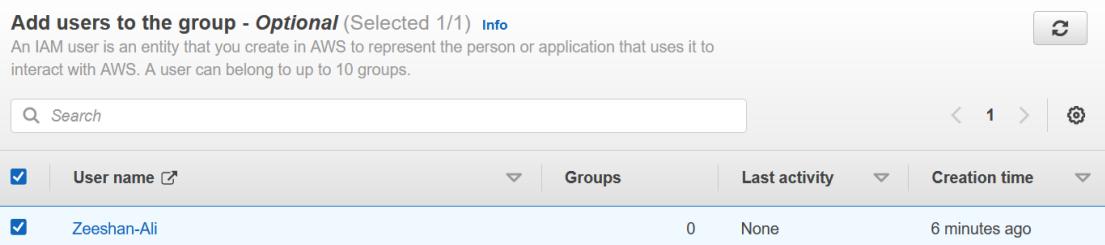
The screenshot shows the AWS IAM User groups page. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'User groups' is also selected. The main area shows a table with one row: 'No resources to display'. At the top right, there's a blue 'Create group' button.

2) Assign names to the group.



The screenshot shows the 'Create user group' page. The left sidebar is identical to the previous screenshot. In the main area, there's a 'Name the group' section with a text input field containing 'Developers'. Below it is an 'Add users to the group - Optional' section with a table showing one user: 'Zeeshan-Ali'.

3) Add Users to the group.



The screenshot shows the 'Add users to the group - Optional' page. It displays a table with one row: 'Zeeshan-Ali'. The 'User name' column has a checked checkbox next to it. The table includes columns for 'Groups', 'Last activity', and 'Creation time'.

4) Attach Permissions to the group.

5) Click-On create group.

Attach permissions policies - *Optional* (Selected 3/746)

Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Filter policies by property or policy name and press enter

Policy name	Type	Description
<input checked="" type="checkbox"/> AWSDirectConnectReadOnlyAccess	AWS managed	Provides read only access to AWS Direct Connect resources.
<input checked="" type="checkbox"/> AmazonGlacierReadOnlyAccess	AWS managed	Provides read only access to Amazon Glacier resources.
<input checked="" type="checkbox"/> AWSMarketplaceFullAccess	AWS managed	Provides the ability to manage AWS Marketplace products.
<input type="checkbox"/> AWSSSODirectoryAdministrator	AWS managed	Administrator access to AWS SSO Directory.
<input type="checkbox"/> AWSIoT1ClickReadOnlyAccess	AWS managed	Provides read only access to AWS IoT 1-Click resources.

6) User group generated.

https://us-east-1.console.aws.amazon.com/iamv2/home#/groups

Developer user group created.

IAM > User groups

User groups (1) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
Developers	1	Defined	Now

D) Create Policies

Steps:

- 1) Click-On user you want to update policies for

Add permissions to Zeeshan-Ali

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

Add user to group

Create group Refresh

Search Group Attached policies

No results

2) Create a Group

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name Admin

Create policy Refresh

Filter policies Search Showing 748 results

Policy name	Type	Used as	Description
<input type="checkbox"/> AdministratorAccess	Job function	None	Provides full access to AWS services and resources.
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct acce...
<input type="checkbox"/> AdministratorAccess-AWSElastic...	AWS managed	None	Grants account administrative permissions. Explicitly allows developers and a...

Cancel Create group

3) Add policies for the group.

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name Admin

Create policy Refresh

Filter policies Search Showing 748 results

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> AdministratorAccess	Job function	None	Provides full access to AWS services and resources.
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permissions while explicitly allowing direct acce...

Cancel Create group

4) Click On Generate Policy.

The screenshot shows the AWS IAM Permissions page for a user. At the top, there are tabs for 'Permissions', 'Groups (2)', 'Tags (1)', 'Security credentials', and 'Access Advisor'. The 'Permissions' tab is selected. Below the tabs, there's a section titled 'Permissions policies (5 policies applied)' with a 'Add permissions' button and a 'Add inline policy' link. A table lists attached policies: 'IAMUserChangePassword' (AWS managed policy). There are sections for 'Attached directly' and 'Attached from group', both currently empty. Below these, there's a 'Show 4 more' link. Further down, there's a 'Permissions boundary (not set)' section and a 'Generate policy based on CloudTrail events' section with a note about generating a new policy based on access activity. At the bottom, there's a 'Share your feedback' link and a prominent blue 'Generate policy' button.

Conclusion: IAM operation executed successfully.

Practical 8

Aim: Demonstrates how to make basic requests to Amazon DynamoDB including:

- how to add items to a table.
- how to get items from a table.
- how to query items on a table.
- how to scan items from a table using a filter expression.
- how to update items on a table.
- how to delete a table.

Steps:

1) Create table

The screenshot shows the AWS DynamoDB console interface. The top navigation bar includes links for Most Visited, Web Development Co..., Sean Hajipan - a web d..., Free Vector Icons and ..., Getting Started, and Other Bookmarks. The main navigation menu on the left has sections for Dashboard, Tables, Update settings, Explore items, PartQL editor (New), Backups, Exports to S3, Reserved capacity, DAX (Clusters, Subnet groups, Parameter groups, Events), Tell us what you think, and Return to the previous console. A feedback link is also present at the bottom of this menu. The central content area features a banner for the new DynamoDB console, followed by a main heading 'Amazon DynamoDB: A fast and flexible NoSQL database service for any scale'. Below this are sections for 'How it works' (with a screenshot of a browser showing the 'What is Amazon DynamoDB?' page), 'Get started' (with a 'Create table' button), and 'Pricing' (with information about charges). At the bottom of the main content area, there's a copyright notice: '© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences'. The bottom navigation bar shows the path 'DynamoDB > Tables > Create table' and a feedback link. The main form for creating a new table is displayed, with fields for 'Table name' (containing 'Zeeshan-Practical-8'), 'Partition key' (containing 'Roll-no' with a dropdown set to 'Number'), and 'Sort key - optional' (containing 'Enter the sort key name' with a dropdown set to 'String'). The bottom of the form also includes a 'Feedback' link and the same copyright notice.

Screenshot of the AWS DynamoDB console showing the creation of a new table. The table configuration includes:

- Read/write capacity: 5 units each with auto scaling enabled.
- Secondary indexes: None
- Key management for encryption at rest: AWS owned key
- Table class: Standard

The Tags section shows no tags are associated with the resource, with an option to "Add new tag".

Buttons at the bottom: Cancel and Create table (highlighted).

Screenshot of the AWS DynamoDB console showing the creation of a new table named "Zeeshan-practical-8". The status is "Creating".

Tables list:

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
Zeeshan-practical-8	Creating	Roll-no (S)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)

Screenshot of the AWS DynamoDB console showing the details of the newly created table "Zeeshan-Practical-8".

General information:

Partition key	Sort key	Capacity mode	Table status
Roll-no (Number)	-	Provisioned	Active No active alarms

Items summary:

Item count	Table size	Average item size
0	0 bytes	0 bytes

General information

Partition key	Sort key	Capacity mode
Roll-no (Number)	-	Provisioned

Items summary

Item count	Table size	Average item size
0	0 bytes	0 bytes

Scan/Query items

Table or index: Zeeshan-Practical-8

Filters:

Attribute name	Type	Condition	Value
Enter attribute	String	Equal to	Enter attribute

Run Reset

2) Create Item

Create item

Attributes

Attribute name	Value	Type
Roll-no - Partition key	99	Number
Name	suresh	String
Phone-No	6969696969	Number

Add new attribute ▾

Cancel Create item

3) Scan Items.

⌚ Completed Read capacity units consumed: 0.5

Items returned (3)

< 1 > | ⚙️ ✖️

	Roll-no	Name	Phone-No
<input type="checkbox"/>	99	Goliath	6969696969
<input type="checkbox"/>	1	John	
<input type="checkbox"/>	4	Zeeshan	

▼ Scan/Query items

Table or index

▼ Filters

Attribute name: Roll-no Type: Number Condition: Equal to Value: 4

⌚ Completed Read capacity units consumed: 0.5

Items returned (1)

< 1 > | ⚙️ ✖️

	Roll-no	Name
<input type="checkbox"/>	4	Zeeshan

4) Query Items.

The screenshot shows the AWS Lambda console's "Scan/Query items" interface. The "Query" tab is selected. In the "Table or index" dropdown, "Zeeshan-Practical-8" is chosen. Below it, the "Roll-no (Partition key)" field contains "99". Under the "Filters" section, there is a note to "Add a filter to get started" and a "Add filter" button. At the bottom are "Run" and "Reset" buttons. The results section shows a green "Completed" status with "Read capacity units consumed: 0.5". It displays one item in the "Items returned (1)" table:

	Roll-no	Name	Phone-No
<input type="checkbox"/>	99	Goliath	6969696969

5) Delete table

The screenshot shows a confirmation dialog titled "Delete table". It asks, "You are about to delete a table." and lists the table "Zeeshan-Practical-8". It includes two checkboxes: "Delete all CloudWatch alarms for this table." (checked) and "Create a backup of this table before deleting it." (unchecked). A note states, "If you do not select this check box, you will not be able to restore data being deleted." Below is a text input field with "delete" typed in, followed by "Cancel" and "Delete table" buttons.

The screenshot shows the AWS DynamoDB console interface. The left sidebar has a 'Tables' section selected. The main area displays a table titled 'Tables (1) Info'. A single row is present, representing a table named 'Zeehan-Practical-8'. The 'Status' column shows '(Deleting)', indicating the table is in the process of being deleted. The table has 0 indexes, Read capacity mode is Provisioned (1), Write capacity mode is Provisioned (1), Size is 0 bytes, and Table class is DynamoDB Standard. A success message at the top states: 'The request to delete the "Zeehan-Practical-8" table has been submitted successfully.'

Conclusion: Add, Update, Scan, Query and Delete table operation successfully implemented on DynamoDB.