Dharmsinh Desai University, Nadiad
Department of Information Technology
DC, IT717
B.Tech. IT, Sem: VII

Submitted By
Roll No.: IT032
Name: Jigar Patel

Experiment 1

Aim: a) Implement concurrent echo client-server application.
b) Implement concurrent day-time client-server application.

A) Implement concurrent echo client-server application.
Code:

1) **Server.c**

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int listensd,clientsd;

char buffer[MAXLINESIZE+1];

struct sockaddr_in servaddr;
struct sockaddr_in cliaddr;
int noBytesRead=0;
socklen_t len;
/*this function will server client that connects*/

void processClient(int);

int main(){
/*Create socket*/
  if((listensd=socket(AF_INET,SOCK_STREAM,0))<0)
  {
    fprintf(stderr,"Cannot create socket\n");
    exit(-1);
```

```
  }

  /*Initialize socket address structure*/
  bzero(&servaddr,sizeof(servaddr));
  servaddr.sin_family=AF_INET;
  servaddr.sin_port=htons(SERV_PORT);
  servaddr.sin_addr.s_addr=htonl(INADDR_ANY);

  /*bind socket address to the socket*/

if(bind(listensd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
{
  fprintf(stderr,"Error in bind\n");
  exit(-1);
}

/*Make the socket listening socket*/
if(listen(listensd,5)<0)
{
  fprintf(stderr,"Error in listen\n");
  exit(-1);
}
len = sizeof(cliaddr);
for(;;){
  /*wait for client connection*/

  clientsd=accept(listensd,(struct sockaddr*)NULL,NULL);

  if(fork()==0)
  {
    /*close listening socket in child. So that reference count remains one. The child serves
the   client. It does not need listening socket to do this. */
    close(listensd);
    /*server client*/
    processClient(clientsd);
    /*close connected socket*/
    close(clientsd);
    exit(0);
  }
  close(clientsd);
 }
  return 0;
}
void processClient(int clientsd)
```

```
{
  /*read message from client and send back*/
  while((noBytesRead=read(clientsd,buffer,sizeof(buffer)))>0)
 {
  write(clientsd,buffer,noBytesRead);
  fprintf(stdout,"%s\n",buffer);
 }
}
```

2) Client.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#include<stdlib.h>

#define MAXLINESIZE 100
#define SERV_PORT 5555

int main(int argc,char** argv){

  int connectsd;


char sendBuffer[MAXLINESIZE+1];
char recvBuffer[MAXLINESIZE+1];

struct sockaddr_in servaddr;
int noBytesRead=0;

if(argc!=2){
fprintf(stderr,"Usage: %s IP-Address\n",argv[0]);
exit(-1);
}

if((connectsd=socket(AF_INET,SOCK_STREAM,0))<0){
fprintf(stderr,"Cannot create socket\n");
exit(-1);
}

bzero(&servaddr,sizeof(servaddr));
```

```
servaddr.sin_family=AF_INET;
servaddr.sin_port=htons(SERV_PORT);
if(inet_pton(PF_INET,argv[1],&servaddr.sin_addr)<=0){
fprintf(stderr,"Error in inet_pton\n");
exit(-1);
}
if(connect(connectsd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0){
fprintf(stderr,"Error in connect\n");
exit(-1);
}

for(;gets(sendBuffer)!=NULL;){
write(connectsd,sendBuffer,strlen(sendBuffer)+1);
if(noBytesRead=read(connectsd,recvBuffer,sizeof(recvBuffer))<0)
exit(0);
fprintf(stdout,"%s\n",recvBuffer);
}
return 0;
}
```

Output:

1)

```
user@ubuntu:~/Desktop$ ./client.out 127.0.0.1
Hello World
Hello World
```

2)

```
user@ubuntu:~/Desktop$ ./server.out
Hello World
```

A) Implement concurrent day-time client-server application.

Code:

1) Server.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <time.h>
#include<string.h>
```

```
/* This for the buffer size */
#define MAXLINESIZE 100
/* This is for the Port */
#define SERV_PORT 8000

int sockfd;
struct sockaddr_in servaddr;
struct sockaddr_in cliaddr;
int n;
socklen_t len;
char msg[MAXLINESIZE];
char ipstr[INET_ADDRSTRLEN];

int main()
{
   sockfd=socket(AF_INET,SOCK_DGRAM,0);

  bzero(&servaddr,sizeof(servaddr));

  servaddr.sin_family = AF_INET;
  servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
  servaddr.sin_port = htons(SERV_PORT);

  bind(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));

  for(;;)
  {
    len=sizeof(cliaddr);


    n=recvfrom(sockfd,msg,MAXLINESIZE,0,(struct sockaddr*)&cliaddr,&len);


          time_t t = time(NULL);
          struct tm tm = *localtime(&t);

          char timeString[100];
          char dateString[100];

          sprintf(dateString,"%d-%02d-%02d",tm.tm_year + 1900, tm.tm_mon + 1,
tm.tm_mday );
          sprintf(timeString,"%02d:%02d:%02d",tm.tm_hour, tm.tm_min, tm.tm_sec );

          if(strcmp(msg,"1")==0)
```

```
        {
                printf("Request for date:\n");
                sendto(sockfd,timeString,strlen(timeString),0,(struct
sockaddr*)&cliaddr,len);
        }
        if(strcmp(msg,"2")==0){

                printf("Request for time:\n ");
                sendto(sockfd,dateString,strlen(dateString),0,(struct
sockaddr*)&cliaddr,len);
        }



 }

 return 0;
}
```

2) Client.c

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<string.h>
#include<stdlib.h>
/* This for the buffer size */
#define MAXLINESIZE 100
/* This is for the Port */
#define SERV_PORT 8000

int sockfd;
struct sockaddr_in servaddr;
int n;
char msg[MAXLINESIZE];
char recvbuffer[MAXLINESIZE];
int main(int argc, int** argv)
{
            if(argc!=2){
              fprintf(stderr,"Usage: %s IP-Address\n",argv[0]);
               exit(-1);
            }

  sockfd = socket(AF_INET,SOCK_DGRAM,0);
```

```
  bzero(&servaddr,sizeof(servaddr));
  servaddr.sin_family = AF_INET;
  if(inet_pton(AF_INET,argv[1], &servaddr.sin_addr)<0)
    fprintf(stderr,"Error in inet_pton");
  servaddr.sin_port = htons(SERV_PORT);
 for(;;){
          printf("1. Time : \n");
          printf("2. Date : \n");
          printf("Select any option : ");


          gets(msg);
          printf("\n\n");
          //if(gets(msg)==NULL) break;
  sendto(sockfd,msg,strlen(msg),0,(struct sockaddr*)&servaddr,sizeof(servaddr));
  n = recvfrom(sockfd,recvbuffer,MAXLINESIZE,0,NULL,NULL);
  recvbuffer[n]='\0';
  fprintf(stdout,"%s\n",recvbuffer);
          printf("\n\n\n");
 }
  return 0;
 }
```

Output:

1)

```
~$ gcc dateServer.c -o datServer
~$ ./datServer
Request for date:
Request for time:
 Request for date:
Request for date:
Request for time:
```

2)

```
~$ gcc dateClient.c -o dateClient
dateClient.c: In function ‘main’:
dateClient.c:20:43: warning: format ‘%s’ expects argument of type ‘char *’, but argument 3 has type ‘int *’ [-Wformat
=]
   20 |                  fprintf(stderr,"Usage: %s IP-Address\n",argv[0]);
      |                                         ~^                ~~~~~~~
      |                                          |                |
      |                                          char *           int *
      |                                         %ls
dateClient.c:28:6: warning: implicit declaration of function ‘inet_pton’ [-Wimplicit-function-declaration]
   28 |   if(inet_pton(AF_INET,argv[1], &servaddr.sin_addr)<0)
      |      ^~~~~~~~~
dateClient.c:40:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-decla
ration]
   40 |  if(gets(msg)==NULL) break;
      |     ^~~~
      |     fgets
dateClient.c:40:14: warning: comparison between pointer and integer
   40 |  if(gets(msg)==NULL) break;
      |              ^~
/usr/bin/ld: /tmp/ccdD7TGz.o: in function `main':
dateClient.c:(.text+0x119): warning: the `gets' function is dangerous and should not be used.
~$ ./dateClient 0.0.0.0
Hello Roll number 34.
 Enter the choice
1. Time :
2. Date :
Select any option : 1
07:13:06



1. Time :
2. Date :
Select any option : 2
2020-11-01
```

Experiment 2

Aim: Configure following options on server socket and tests them: SO_KEEPALIVE, SO_LINGER, SO_SNDBUF, SO_RCVBUF, TCP_NODELAY.

Code:

1) **KeepAlive.c**

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#include <stdlib.h>
#define MAXLINESIZE 100
#define SERV_PORT 5877
int main(int argc, char **argv)
{
   int fd;
   char sendBuffer[MAXLINESIZE + 1];
   char recvBuffer[MAXLINESIZE + 1];
   struct sockaddr_in servaddr;
   int noBytesRead = 0;
   if (argc != 2)
   {
      fprintf(stderr, "Usage : %s IP-Address\n", argv[0]);
      exit(-1);
   }
   if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
   {
      fprintf(stderr, "Cannot create socket\n");
      exit(-1);
   }

   int result = 0;
   int sendbuff, optlen;

   result = getsockopt(fd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff,
sizeof(sendbuff));
   printf("SO_KEEPALIVE is %s\n", (sendbuff ? "ON" : "OFF"));
   sendbuff = 1;
   printf("Setting SO_KEEPALIVE to %d\n", sendbuff);
   result = setsockopt(fd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff, sizeof(sendbuff));
   // Get flag
```

```
    optlen = sizeof(sendbuff);
    result = getsockopt(fd, SOL_SOCKET, SO_KEEPALIVE, &sendbuff, &optlen);
    printf("SO_KEEPALIVE is %s\n", (sendbuff ? "ON" : "OFF"));
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(SERV_PORT);
    if (inet_pton(PF_INET, argv[1], &servaddr.sin_addr) <= 0)
    {
       fprintf(stderr, "Error while setting inet_pton");
       exit(-1);
    }
    if (connect(fd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
       fprintf(stderr, "Error while connecting");
       exit(-1);
    }
    for (; gets(sendBuffer) != NULL;)
    {
       write(fd, sendBuffer, strlen(sendBuffer) + 1);
       if (noBytesRead = read(fd, recvBuffer, strlen(recvBuffer)) < 0)
          exit(0);
       fprintf(stdout, "%s\n", recvBuffer);
    }
    return 0;
}
```

Output:

```
user@ubuntu:~/Desktop$ ./client.out 127.0.0.1
SO_KEEPALIVE is OFF
set SO_KEEPALIVE to 1
SO_KEEPALIVE is ON
HELLO SERVER
```

2) **Linger.c**

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#include <stdlib.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555
int main(int argc, char **argv)
```

```
{
    int fd;
    char sendBuffer[MAXLINESIZE + 1];
    char recvBuffer[MAXLINESIZE + 1];
    struct sockaddr_in servaddr;
    int noBytesRead = 0;
    if (argc != 2)
    {
        fprintf(stderr, "Usage : %s IP-Address\n", argv[0]);
        exit(-1);
    }
    if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        fprintf(stderr, "Cannot create socket\n");
        exit(-1);
    }
    struct linger
    {
        int l_onoff;
        int l_linger;
    } l;
    socklen_t optlen;
    int res = 0;
    optlen = sizeof(l);
    res = getsockopt(fd, SOL_SOCKET, SO_LINGER, &l, &optlen);
    printf("SO_LINGER is %d for time %d\n", l.l_onoff, l.l_linger);
    l.l_onoff = 1;
    l.l_linger = 10;
    printf("SO_LINGER is set to %d for time %d\n", l.l_onoff, l.l_linger);
    res = setsockopt(fd, SOL_SOCKET, SO_LINGER, &l, sizeof(l));
    optlen = sizeof(l);
    res = getsockopt(fd, SOL_SOCKET, SO_LINGER, &l, &optlen);
    printf("SO_LINGER is %d for time %d\n", l.l_onoff, l.l_linger);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(SERV_PORT);
    if (inet_pton(PF_INET, argv[1], &servaddr.sin_addr) <= 0)
    {
        fprintf(stderr, "Error in inet_pton");
        exit(-1);
    }
    if (connect(fd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
        fprintf(stderr, "Error in connect");
```

```
        exit(-1);
    }
    for (; gets(sendBuffer) != NULL;)
    {
        write(fd, sendBuffer, strlen(sendBuffer) + 1);
        if (noBytesRead = read(fd, recvBuffer, strlen(recvBuffer)) < 0)
            exit(0);
        fprintf(stdout, "%s\n", recvBuffer);
    }
    return 0;
}
```

Output:

```
user@ubuntu:~/Desktop$ ./client.out 127.0.0.1
SO_LINGER is 0 for time 0
SO_LINGER is set to 1 for time 10
SO_LINGER is 1 for time 10
Hello world
```

3) **Sendbuf.c**

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#include <stdlib.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555
int main(int argc, char **argv)
{
    int fd;
    char sendBuffer[MAXLINESIZE + 1];
    char recvBuffer[MAXLINESIZE + 1];
    struct sockaddr_in servaddr;
    int noBytesRead = 0;
    if (argc != 2)
    {
        fprintf(stderr, "Usage : %s IP-Address\n", argv[0]);
        exit(-1);
    }
    if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        fprintf(stderr, "Cannot create socket\n");
        exit(-1);
```

```
    }

    int sendbuff;
    socklen_t optlen;
    int res = 0;
    // Get buffer size
    optlen = sizeof(sendbuff);
    res = getsockopt(fd, SOL_SOCKET, SO_SNDBUF, &sendbuff, &optlen);
    printf("send buffer size = %d\n", sendbuff);
    // Set buffer size
    sendbuff = 40480;
    printf("sets the send buffer to %d\n", sendbuff);
    res = setsockopt(fd, SOL_SOCKET, SO_SNDBUF, &sendbuff, sizeof(sendbuff));
    // Get buffer size
    optlen = sizeof(sendbuff);
    res = getsockopt(fd, SOL_SOCKET, SO_SNDBUF, &sendbuff, &optlen);
    printf("send buffer size = %d\n", sendbuff);
    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(SERV_PORT);
    if (inet_pton(PF_INET, argv[1], &servaddr.sin_addr) <= 0)
    {
       fprintf(stderr, "Error in inet_pton");
       exit(-1);
    }
    if (connect(fd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
       fprintf(stderr, "Error in connect");
       exit(-1);
    }
    for (; gets(sendBuffer) != NULL;)
    {
       write(fd, sendBuffer, strlen(sendBuffer) + 1);
       if (noBytesRead = read(fd, recvBuffer, strlen(recvBuffer)) < 0)
          exit(0);
       fprintf(stdout, "%s\n", recvBuffer);
    }
    return 0;
}
```

Output:

```
user@ubuntu:~/Desktop$ ./client.out 127.0.0.1
send buffer size = 16384
sets the send buffer to 40480
send buffer size = 80960
Hello World
```

4) **Recbuf.c**

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include<stdlib.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555
int listensd,clientsd;
char buffer[MAXLINESIZE+1];
struct sockaddr_in servaddr;
struct sockaddr_in peeraddr;
int noBytesRead=0;
//
void processClient(int);
int main()
{
if((listensd=socket(AF_INET,SOCK_STREAM,0))<0)
{
fprintf(stderr,"Cannont create socket\n");
exit(-1);
}
 int sockfd, recvbuff;
 socklen_t optlen;

 int res = 0;
 optlen = sizeof(recvbuff);
 res = getsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, &optlen);
 printf("receive buffer size = %d\n", recvbuff);
 recvbuff = 40480;
 printf("sets the recv buffer to %d\n", recvbuff);
 res = setsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, sizeof(recvbuff));
 optlen = sizeof(recvbuff);
 res = getsockopt(sockfd, SOL_SOCKET, SO_RCVBUF, &recvbuff, &optlen);
 printf("receive buffer size = %d\n", recvbuff);
```

```
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_port=htons(SERV_PORT);
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
if(bind(listensd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0) {
fprintf(stderr,"Error in bind\n");
exit(-1);
}
if(listen(listensd,5)<0) {
fprintf(stderr,"Error in listen\n");
exit(-1);
}
for(;;) {
clientsd=accept(listensd,(struct sockaddr*)NULL,NULL);
if(fork()==0) {
int len = sizeof(peeraddr);
int n=getpeername(clientsd,(struct sockaddr*)&peeraddr,&len);
char ip[MAXLINESIZE];
if(n==-1) {
fprintf(stderr,"Peer Call Error!");
exit(-1);
}
const char* res=inet_ntop(AF_INET,&peeraddr.sin_addr,ip,MAXLINESIZE);
fprintf(stdout,"IP:%s & Port: %d\n",ip,peeraddr.sin_port);
close(listensd);
processClient(clientsd);
close(clientsd);
exit(0);
}
close(clientsd);
}
return 0;
}
void processClient(int clientsd)
{
while((noBytesRead=read(clientsd,buffer,sizeof(buffer)))>0){
fprintf(stdout,"%s\n",buffer);
write(clientsd,buffer,noBytesRead);
}
}
```

Output:

```
user@ubuntu:~/Desktop$ ./server.out
receive buffer size = 1899102824
sets the recv buffer to 40480
receive buffer size = 40480
IP:127.0.0.1 & Port: 30935
Hello World
```

5) **Nodelay.C**

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <string.h>
#include <stdlib.h>
#include <netinet/tcp.h>
#define MAXLINESIZE 100
#define SERV_PORT 5555
int main(int argc, char **argv)
{
   int fd;
   char sendBuffer[MAXLINESIZE + 1];
   char recvBuffer[MAXLINESIZE + 1];
   struct sockaddr_in servaddr;
   int noBytesRead = 0;
   if (argc != 2)
   {
      fprintf(stderr, "Usage : %s IP-Address\n", argv[0]);
      exit(-1);
   }
   if ((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
   {
      fprintf(stderr, "Cannot create socket\n");
      exit(-1);
   }

   int flag;
   socklen_t optlen;
   int result = 0;
   optlen = sizeof(flag);
   result = getsockopt(fd, IPPROTO_TCP, TCP_NODELAY, &flag, &optlen);
   printf("Get TCP_NODELAY option %s\n", (flag ? "ON" : "OFF"));
```

```
    flag = 1;
    printf("sets the TCP_NODELAY\n");
    result = setsockopt(fd, IPPROTO_TCP, TCP_NODELAY, &flag, sizeof(int));
    flag = 0;
    optlen = sizeof(flag);
    result = getsockopt(fd, IPPROTO_TCP, TCP_NODELAY, &flag, &optlen);
    printf("Now TCP_NODELAY is: %s\n", (flag ? "ON" : "OFF"));
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(SERV_PORT);
    if (inet_pton(PF_INET, argv[1], &servaddr.sin_addr) <= 0)
    {
        fprintf(stderr, "Error in inet_pton");
        exit(-1);
    }
    if (connect(fd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
        fprintf(stderr, "Error in connect");
        exit(-1);
    }
    for (; gets(sendBuffer) != NULL;)
    {
        write(fd, sendBuffer, strlen(sendBuffer) + 1);
        if (noBytesRead = read(fd, recvBuffer, strlen(recvBuffer)) < 0)
            exit(0);
        fprintf(stdout, "%s\n", recvBuffer);
    }
    return 0;
}
```

Output:

```
user@ubuntu:~/Desktop$ ./client.out 127.0.0.1
Get TCP_NODELAY option OFF
sets the TCP_NODELAY
Now TCP_NODELAY is: ON
HELLO WORLD
```

Experiment 3

Aim: Data Representation and Data Validation: XML Schema and XML instance document, JSON.

Procedure:

1) Design a schema for the student list. A student has information such as name, semester, roll no, email-ids, phone-nos, etc.
2) Write an XML instance document for the designed schema and validate this instance document against the schema.

1) **Student.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema version="1.0"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">
 <xs:element name="Student">
    <xs:complexType>
    <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="age" type="xs:integer"/>
    <xs:element name="roll" type="xs:string"/>
    <xs:element name="email" type="xs:string"/>
    <xs:element name="phone" type="xs:integer"/>
    <xs:element name="branch" type="xs:string"/>
    </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

2) **Student.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<Student
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xsi:noNamespaceSchemaLocation='student.xsd'>
    <name>Jigar</name>
 <age>21</age>
 <roll>IT-032</roll>
 <email>jigar123@gmail.com</email>
 <phone>7985421252</phone>
 <branch>IT</branch>
</Student>
```

Ouput:

```
XML validation started.
Checking file:/C:/Users/A/Documents/NetBeansProjects/JavaApplication1/src/javaapplication1/student.xml...
Referenced entity at "file:/C:/Users/A/Documents/NetBeansProjects/JavaApplication1/src/javaapplication1/student.xsd".
XML validation finished.
```

Experiment 4

Aim: WSDL based webservice and its monitoring: Implement ArithmeticService that implements add and subtract operations / Java based: Implement TrigonometricService that implements sin, and cos operations. Monitor SOAP request and response packets. Analyze parts of it and compare them with the operations (java functions) headers.

Tools: Netbeans 6.0, GlassFish Server, Web Service.

Code:

1) **ArithmeticService.java**

```
package websvc;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
@WebService(serviceName = "ArithematicService")
public class ArithematicService {
   @WebMethod(operationName = "Addition")
   public String Addition(@WebParam(name = "operand1") final double operand1,
@WebParam(name = "operand2") final double operand2) {
      return (operand1+operand2)+"";
   }
   @WebMethod(operationName = "Subtraction")
   public String Subtraction(@WebParam(name = "operand1") final double operand1,
@WebParam(name = "operand2") final double operand2) {
      return (operand1-operand2)+"";
   }
}
```

2) **ArithmeticClient.java**

```
package websvc;
public class ArithmeticClient {

   public static void main(String[] args) {
      // TODO code application logic here
      System.out.println("add "+add(100,200));

      System.out.println("sub "+sub(500,200));


   }

   private static Double add(double input1, double input2) {
      src.ArithmeticService_Service service = new src.ArithmeticService_Service();
      src.ArithmeticService port = service.getArithmeticServicePort();
```

```
        return port.add(input1, input2);
    }

    private static Double sub(double input1, double input2) {
        src.ArithmeticService_Service service = new src.ArithmeticService_Service();
        src.ArithmeticService port = service.getArithmeticServicePort();
        return port.sub(input1, input2);
    }
}
```

3) TrigoService.java

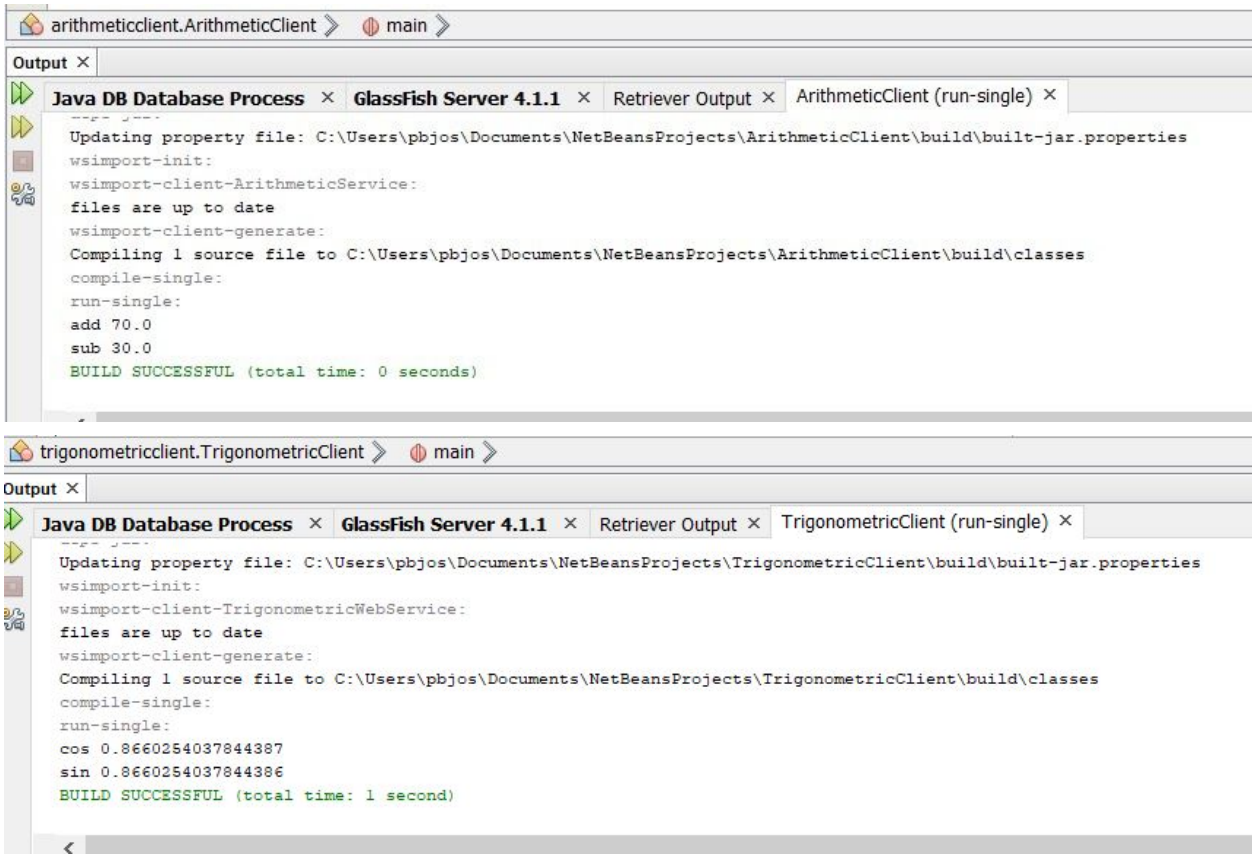
```
package src;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
@WebService(serviceName = "TrigonometricWebService")
public class TrigonometricWebService {
    @WebMethod(operationName = "Sin")
    public Double Sin(@WebParam(name = "input") double input) {
        return Math.sin( input*Math.PI/180);
    }
    @WebMethod(operationName = "Cos")
    public Double Cos(@WebParam(name = "input") double input) {
        return Math.cos(input*Math.PI/180);
    }
}
```

4) TrigoClient.java

```
Package src;
public class TrigonometricClient {
    public static void main(String[] args) {
        System.out.println("cos "+cos(45));
        System.out.println("sin "+sin(45));
    }

    private static Double cos(double input) {
        src.TrigonometricWebService_Service service = new
src.TrigonometricWebService_Service();
        src.TrigonometricWebService port = service.getTrigonometricWebServicePort();
        return port.cos(input);
    }

    private static Double sin(double input) {
        src.TrigonometricWebService_Service service = new
src.TrigonometricWebService_Service();
```

```
        src.TrigonometricWebService port = service.getTrigonometricWebServicePort();
        return port.sin(input);
    }

}
```

Output:



Experiment 5

Aim: Design and test BPEL module that composes ArithmeticService and Trigonometric Service.

Tools: GlassFish Server, NetBeans 6.0.

Code:

1) **ArithmeticService WSDL**

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at
http://jaxws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT.
--><!-- Generated by
JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI
2.1.3.1-hudson417SNAPSHOT. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis200401wss-wssecurity-utility-1
.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://src/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://src/"
name="ArithmeticServiceService">
<types>
<xsd:schema>
<xsd:import namespace="http://src/"
schemaLocation="ArithmeticServiceService_xsd_1.xsd"></xsd:import>
</xsd:schema>
</types>
<message name="addition">
<part name="parameters" element="tns:addition"></part>
</message>
<message name="additionResponse">
<part name="parameters" element="tns:additionResponse"></part>
</message>
<portType name="ArithmeticService">
<operation name="addition">
<input message="tns:addition"></input>
<output message="tns:additionResponse"></output>
</operation>
</portType>
<binding name="ArithmeticServicePortBinding" type="tns:ArithmeticService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"></soap:binding>
<operation name="addition">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
```

```
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
</binding>
<service name="ArithmeticServiceService">
<port name="ArithmeticServicePort" binding="tns:ArithmeticServicePortBinding">
<soap:address
location="http://localhost:8080/MyArithmeticService/ArithmeticServiceService"></soap:a
ddress>
</port>
</service>
</definitions>
```

2) **ArithmeticServiceWrapper WSDL**

```
<definitions
        xmlns="http://schemas.xmlsoap.org/wsdl/"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        name="ArithmeticServiceServiceWrapper"
        targetNamespace="http://enterprise.netbeans.org/bpel/ArithmeticServiceServiceW
rapper"
xmlns:tns="http://enterprise.netbeans.org/bpel/ArithmeticServiceServiceWrapper"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:ns="http://src/">
 <import location="ArithmeticServiceService.wsdl" namespace="http://src/"/>
 <plnk:partnerLinkType name="ArithmeticService4">
 <plnk:role name="ArithmeticServiceRole" portType="ns:ArithmeticService"/>
 </plnk:partnerLinkType>
 <plnk:partnerLinkType name="ArithmeticServiceLinkType">
 <plnk:role name="ArithmeticServiceRole" portType="ns:ArithmeticService"/>
</plnk:partnerLinkType>
</definitions>
```

3) **ArithmeticService.xsd**

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at
http://jaxws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --
><xs:schema xmlns:tns="http://src/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.0" targetNamespace="http://src/">
<xs:element name="addition" type="tns:addition"></xs:element>
<xs:element name="additionResponse" type="tns:additionResponse"></xs:element>
<xs:complexType name="addition">
<xs:sequence>
<xs:element name="input1" type="xs:double"></xs:element>
```

<xs:element name="input2" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="additionResponse">
<xs:sequence>
<xs:element name="return" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>

4) **ScientificServices WSDL**

<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jaxws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --><!-- Generated by
JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson417SNAPSHOT. --><definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis200401wss-wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://src/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://src/" name="DualScientifcServiceService">
<types>
<xsd:schema>
<xsd:import namespace="http://src/" schemaLocation="DualScientifcServiceService_xsd_1.xsd"></xsd:import>
</xsd:schema>
</types>
<message name="sin">
<part name="parameters" element="tns:sin"></part>
</message>
<message name="sinResponse">
<part name="parameters" element="tns:sinResponse"></part>
</message>
<message name="cos">
<part name="parameters" element="tns:cos"></part>
</message>
<message name="cosResponse">
<part name="parameters" element="tns:cosResponse"></part>
</message>
<portType name="DualScientifcService">
<operation name="sin">
<input message="tns:sin"></input>
<output message="tns:sinResponse"></output>
</operation>
<operation name="cos">

```
<input message="tns:cos"></input>
<output message="tns:cosResponse"></output>
</operation>
</portType>
<binding name="DualScientifcServicePortBinding" type="tns:DualScientifcService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"></soap:binding>
<operation name="sin">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="cos">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
</binding>
<service name="DualScientifcServiceService">
<port name="DualScientifcServicePort" binding="tns:DualScientifcServicePortBinding">
<soap:address
location="http://localhost:8080/MyDualScientificService/DualScientifcServiceService"></
soap:a
dd ress> </port>
</service>
</definitions>
```

5) **ScientificServiceWrapper WSDL**

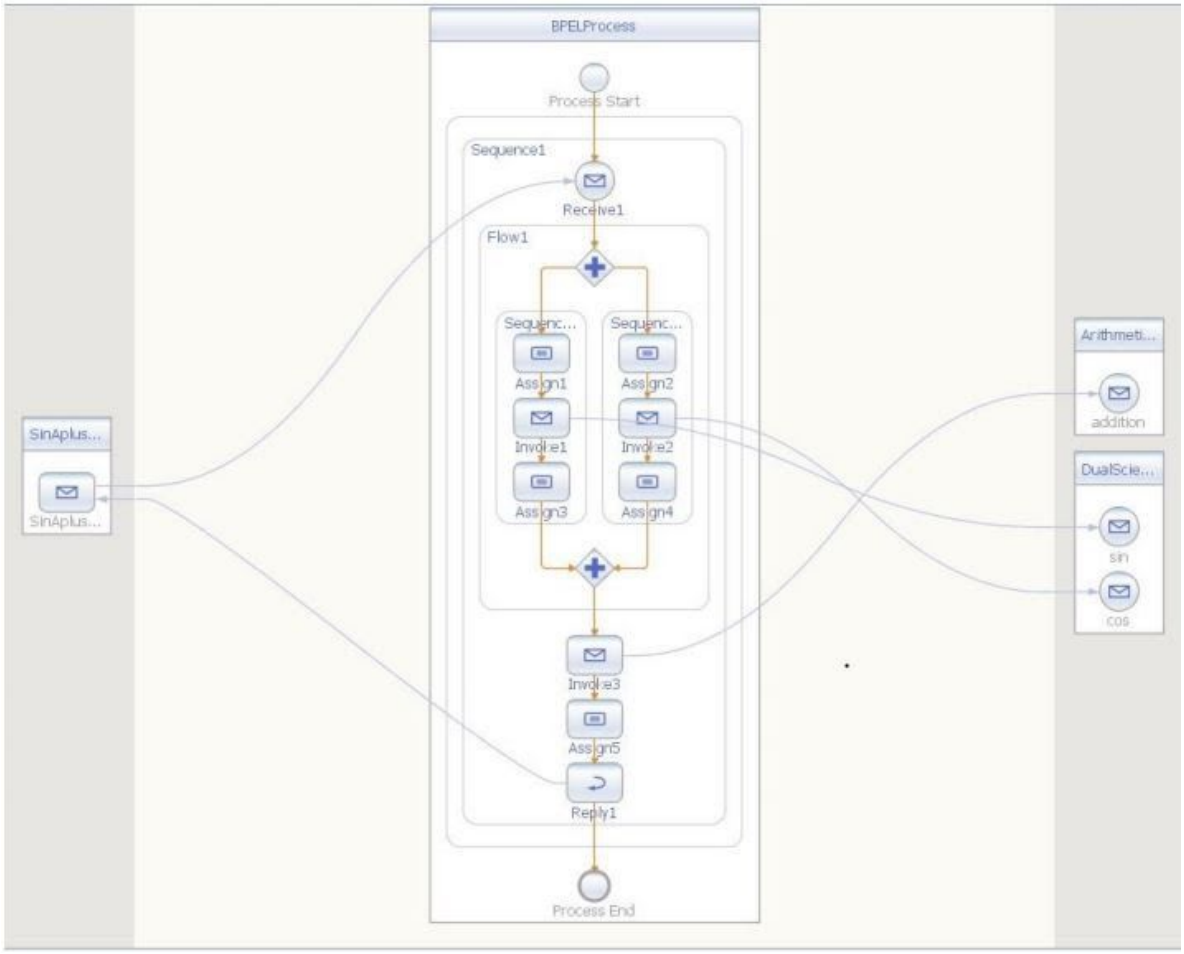
```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
 xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="DualScientifcServiceServiceWrapper"
targetNamespace="http://enterprise.netbeans.org/bpel/DualScientifcServiceServiceWrap
per"
xmlns:tns="http://enterprise.netbeans.org/bpel/DualScientifcServiceServiceWrapper"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype" xmlns:ns="http://src/">
 <import location="DualScientifcServiceService.wsdl" namespace="http://src/"/>
 <plnk:partnerLinkType name="DualScientifcService1">
<plnk:role name="DualScientifcServiceRole" portType="ns:DualScientifcService"/>
</plnk:partnerLinkType>
 <plnk:partnerLinkType name="DualScientifcServiceLinkType">
 <plnk:role name="DualScientifcServiceRole" portType="ns:DualScientifcService"/>
</plnk:partnerLinkType>
</definitions>
```

6) **ScientificServices.xsd**

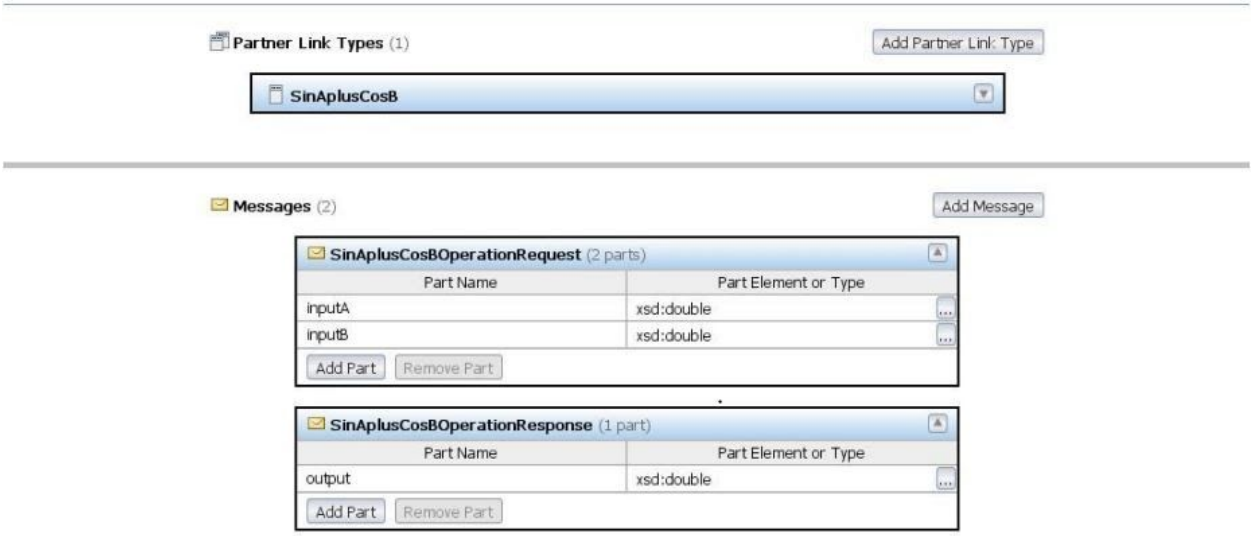
```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at
http://jaxws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-417-SNAPSHOT. --
><xs:schema xmlns:tns="http://src/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.0" targetNamespace="http://src/">
<xs:element name="cos" type="tns:cos"></xs:element>
<xs:element name="cosResponse" type="tns:cosResponse"></xs:element>
<xs:element name="sin" type="tns:sin"></xs:element>
<xs:element name="sinResponse" type="tns:sinResponse"></xs:element>
<xs:complexType name="sin">
<xs:sequence>
<xs:element name="inputSin" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="sinResponse">
<xs:sequence>
<xs:element name="return" type="xs:double" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="cos">
<xs:sequence>
<xs:element name="inputCos" type="xs:double"></xs:element>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="cosResponse">
<xs:sequence>
<xs:element name="return" type="xs:double" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Output:



sinA plus cosB



Experiment 6

Aim: Deployment of a HADOOP cluster and monitoring status of its components.

Tools: NetBeans 6.0, Hadoop Common, Hadoop Distributed File System, Hadoop YARN, Hadoop MapReduce.

Procedure:

- Install the appropriate version of java for your Hadoop.
- ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.
- Download the appropriate hadoop file system from the link given below. http://www.apache.org/dyn/closer.cgi/hadoop/common/
- Unpack the downloaded Hadoop distribution. In the distribution, edit the file etc/hadoop/hadoop-env.sh to define some parameters as follows: " #export JAVA_HOME=/usr/java/latest".
- The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

  $ mkdir input
  $ cp etc/hadoop/*.xml input
  $ bin/hadoop jar share/hadoop/mapreduce/hadoop-example.jar | grep input output 'dfs[a-z.]+'
  $ cat output/*
  $ ssh localhost

  To run MapReduce job locally.

  1) Format the filesystem:
     $ bin/hdfs namenode -format
  2) Start NameNode daemon and DataNode daemon:
     $ sbin/start-dfs.sh
     • The hadoop daemon log output is written to the $HADOOP_LOG_DIR directory (defaults to $HADOOP_HOME/logs).
     • Browse the web interface for the NameNode; by default it is available at:
     NameNode - http://localhost:50070/
     • Make the HDFS directories required to execute MapReduce jobs:
     $ bin/hdfs dfs -mkdir /user
     $ bin/hdfs dfs -mkdir /user/<username>
     • Copy the input files into the distributed filesystem:
     $ bin/hdfs dfs -put etc/hadoop input
     • Run some of the examples provided:
     $ bin/hadoop jar share/hadoop/mapreduce/hadoo-example.jar grep input output 'dfs[a-z.]+'

• Examine the output files: Copy the output files from the distributed file-system to the local filesystem and examine them:

```
$ bin/hdfs dfs -get output output
$ cat output/*
```

Experiment 7

Aim: Perform data intensive computing using map-reduce based programming on a HADOOP cluster.

Tools: NetBeans 6.0, Hadoop Common, Hadoop Distributed File System, Hadoop YARN, Hadoop MapReduce, Ubuntu.

Procedure:

1. Basic installation and configuration :

1.1 configure etc/hosts for master and slaves nodes

```
$ sudo gedit /etc/hosts
# Add following hostname and their ip in host table
192.168.2.14 HadoopMaster
192.168.2.15 HadoopSlave1
192.168.2.16 HadoopSlave2
```

1.2 Create hadoop as group and hduser as user in all Machines (if not created !!).

```
$ sudo addgroup hadoop
$ sudo adduser --ingroup hadoop hduser
$ sudo usermod -a -G sudo hduser
```

1.3 Install rsync for sharing hadoop source with rest all Machines, and reboot all the Machine.

```
$ sudo apt-get install rsync
```

1.4 To make the above changes reflected, we need to reboot all of the Machines.

```
$ sudo reboot
```

2. Applying Common Hadoop Configuration

2.1 Update core-site.xml

Update this file by changing hostname from localhost to HadoopMaster

To edit file, fire the below given command

```
$ sudo gedit core-site.xml
```

Paste these lines into <configuration> tag OR Just update it by replacing localhost with

```
master
<property>
<name>fs.default.name</name>
<value>hdfs://HadoopMaster:9000</value>
</property>
```

2.2 Update hdfs-site.xml

Update this file by updating repliction factor from 1 to 3.

To edit file, fire the below given command

```
$ sudo gedit hdfs-site.xml
```

Paste/Update these lines into <configuration> tag

```
<property>
<name>dfs.replication</name>
<value>3</value>
</property>
```

2.3 Update yarn-site.xml

Update this file by updating the following three properties by updating hostname from localhost to HadoopMaster

To edit file, fire the below given command

```
$ sudo gedit yarn-site.xml
## Paste/Update these lines into <configuration> tag
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>HadoopMaster:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>HadoopMaster:8035</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>HadoopMaster:8050</value>
</property>
```

2.4 Update Mapred-site.xml

Update this file by updating and adding following properties,

To edit file, fire the below given command

```
$ sudo gedit mapred-site.xml
```

Paste/Update these lines into <configuration> tag

```
<property>
<name>mapreduce.job.tracker</name>
<value>HadoopMaster:5431</value>
</property>
<property>
<name>mapred.framework.name</name>
<value>yarn</value>
</property>
```

2.5 Update masters

Update the directory of master nodes of Hadoop cluster

To edit file, fire the below given command

```
$ sudo gedit masters
```

Add name of master nodes HadoopMaster

2.6 Update slaves

Update the directory of slave nodes of Hadoop cluster

To edit file, fire the below given command

```
$ sudo gedit slaves
```

Add name of slave nodes
HadoopSlave1
HadoopSlave2

3. Copying/Sharing/Distributing Hadoop config files to rest all nodes – master/slaves

3.1 Use rsync for distributing configured Hadoop source among rest of nodes via network.

```
# In HadoopSlave1 machine
$ sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave1:/usr/local/hadoop/
# In HadoopSlave2 machine
$ sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave2:/usr/local/hadoop/
```

4 Applying Master node specific Hadoop configuration: (Only for master nodes)

4.1 Remove existing Hadoop_data folder (which was created while single node hadoop setup.)

```
$ sudo rm -rf /usr/local/hadoop_tmp/
```

4.2 : Make same (/usr/local/hadoop_tmp/hdfs) directory and create NameNode (/usr/local/hadoop_tmp/hdfs/namenode) directory

```
$ sudo mkdir -p /usr/local/hadoop_tmp/
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
```

4.3 : Make hduser as owner of that directory.

```
$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

5 Applying Slave node specific Hadoop configuration : (Only for slave nodes)

5.1 Remove existing Hadoop_data folder (which was created while single node hadoop setup)

```
$ sudo rm -rf /usr/local/hadoop_tmp/hdfs/
```

5.2 Creates same (/usr/local/hadoop_tmp/) directory/folder, an inside this folder again Create

DataNode (/usr/local/hadoop_tmp/hdfs/namenode) directory/folder

```
$ sudo mkdir -p /usr/local/hadoop_tmp/
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

5.3 Make hduser as owner of that directory

```
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

6 Copying ssh key for Setting up passwordless ssh access from Master to Slave node :

hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub
hduser@HadoopSlave1
hduser@HadoopMaster: ~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@HadoopSlave2

7. Format Namenonde (Run on MasterNode) :

Run this command from Masternode

hduser@HadoopMaster: usr/local/hadoop/$ hdfs namenode -format

8. Starting up Hadoop cluster daemons : (Run on MasterNode)
   Start HDFS daemons:
      hduser@HadoopMaster:/usr/local/hadoop$ start-dfs.sh
9. Start MapReduce daemons:
   hduser@HadoopMaster:/usr/local/hadoop$ start-yarn.sh

10. Instead both of these above command you can also use start-all.sh, but its now deprecated so its not recommended to be used for better Hadoop operations.

11. Track/Monitor/Verify Hadoop cluster : (Run on any Node)
   Verify Hadoop daemons on Master and slaves(All slave):
      hduser@HadoopMaster: jps

Experiment 8

Aim: Create a Restful Webservice and test it using Postman.

Tools: GlassFish Server, NetBeans 6.0, Postman.

Code:

1) **User.java**

```
package restfull;
import java.io.Serializable;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "user") public class User implements Serializable
{
private
   static final long uid = 1L;
private
   int id;
private
   String name;
private
   String profession;
public
   User(int id, String name, String profession)
   {
      this.id = id;
      this.name = name;
      this.profession = profession;
   }
public
   User()
   {
   }

public
   int getId()
   {
      return id;
   }
public
   void setId(int id)
   {
      this.id = id;
```

```
    }
public
    String getName()
    {
        return name;
    }
public
    void setName(String name)
    {
        this.name = name;
    }
public
    String getProfession()
    {
        return profession;
    }
public
    void setProfession(String profession)
    {
        this.profession = profession;
    }
}
```

2) UserService.java

```
import java.util.Iterator;
import java.util.List;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.*;
@Path("/UserService")
public class UserService {
 UserContext userDao = new UserContext();
 private static final String SUCCESS_RESULT = "<result>success</result>";
 private static final String FAILURE_RESULT = "<result>failure</result>";
 @GET
 @Path("/users")
 @Produces(MediaType.APPLICATION_JSON)
JSON format
 public List<User> getUsers() {
 return userDao.getAllUsers();
 }
 @GET
 @Path("/users/{userid}")
```

```
@Produces(MediaType.APPLICATION_JSON)
public User getUser(@PathParam("userid") int userid) {
System.out.println("The ID received in GET is" + userid);
return userDao.getUser(userid);
}
@POST
@Path("/insertuser")
@Produces(MediaType.TEXT_PLAIN)
public String InsertUsers() {
List<User> NewList = userDao.getAllUsers();
User newuser = new User(3, "ABC", "XYZ");
NewList.add(newuser);
```



```
userDao.saveUserList(NewList);
return "Inserted";
}
@POST
@Path("/adduser")
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
public User AddUser(User user) {
System.out.println("Inside Add User Method");
List<User> NewList = userDao.getAllUsers();
NewList.add(user);
userDao.saveUserList(NewList);
return user;
}
@PUT
@Path("/updateuser")
@Produces(MediaType.APPLICATION_JSON)
@Consumes(MediaType.APPLICATION_JSON)
public User UpdateUser(User user) {
System.out.println("The ID received in GET is" + user.getId());
int result = userDao.updateUser(user);
if (result == 1) {
System.out.println("Success in Update");
} else {
System.out.println("Failure in Update");
}
return user;
}
@DELETE
@Path("/deleteuser/{userid}")
```

```
@Produces(MediaType.TEXT_PLAIN)
@Consumes(MediaType.APPLICATION_JSON)
public String deleteUser(@PathParam("userid") int userid) {
System.out.println("The ID received in DELETE is" + userid);
int result = userDao.deleteUser(userid);
System.out.println("Value of Result is" + result);
if (result == 1) {
return "SUCCESS";
}
return "FAILURE";
}
}
```

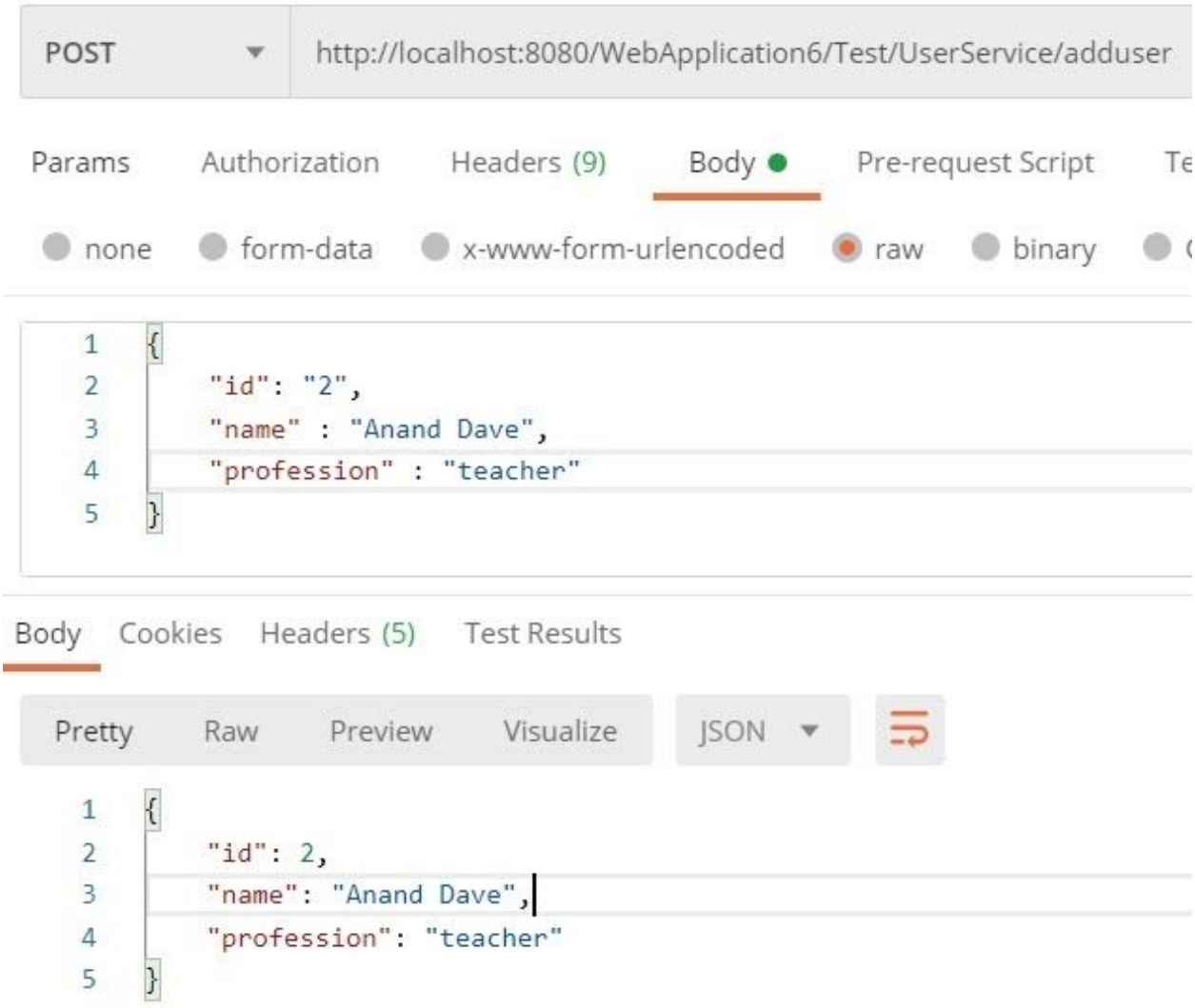
3) UserContext.java

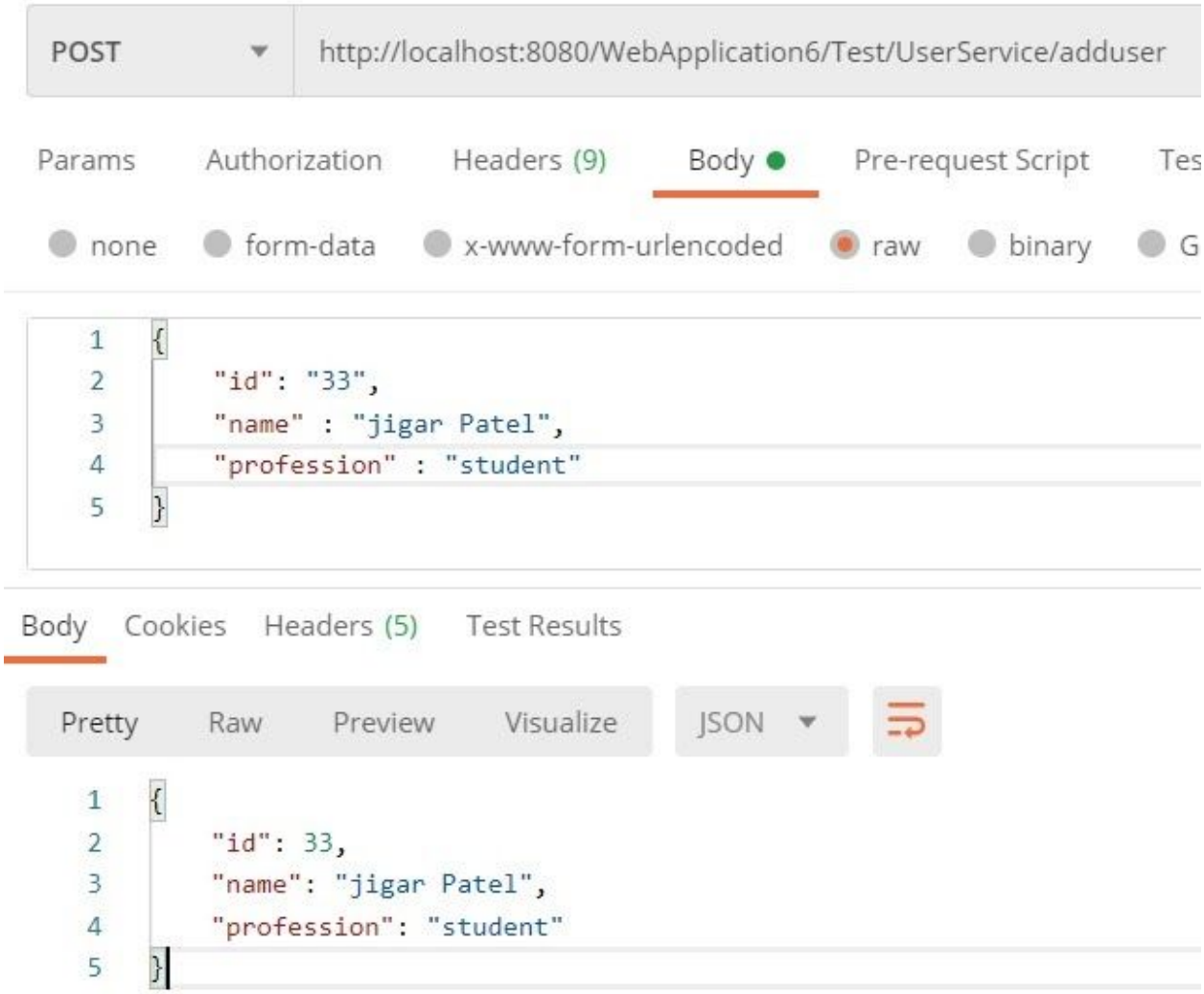
```
package restFull;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.List;
public class UserContext {
 File file = new File("Users.dat");
 public List<User> getAllUsers() {
 List<User> userList = null;
 try {
 if (!file.exists()) {
 User user = new User(1, "Mahesh", "Teacher");
 userList = new ArrayList<User>();
 userList.add(user);
 saveUserList(userList);
 } else {
 FileInputStream fis = new FileInputStream(file);
 ObjectInputStream ois = new ObjectInputStream(fis);
 userList = (List<User>) ois.readObject();
 ois.close();
 }
 } catch (IOException e) {
 e.printStackTrace();
 } catch (ClassNotFoundException e) {
 }
 return userList;
```

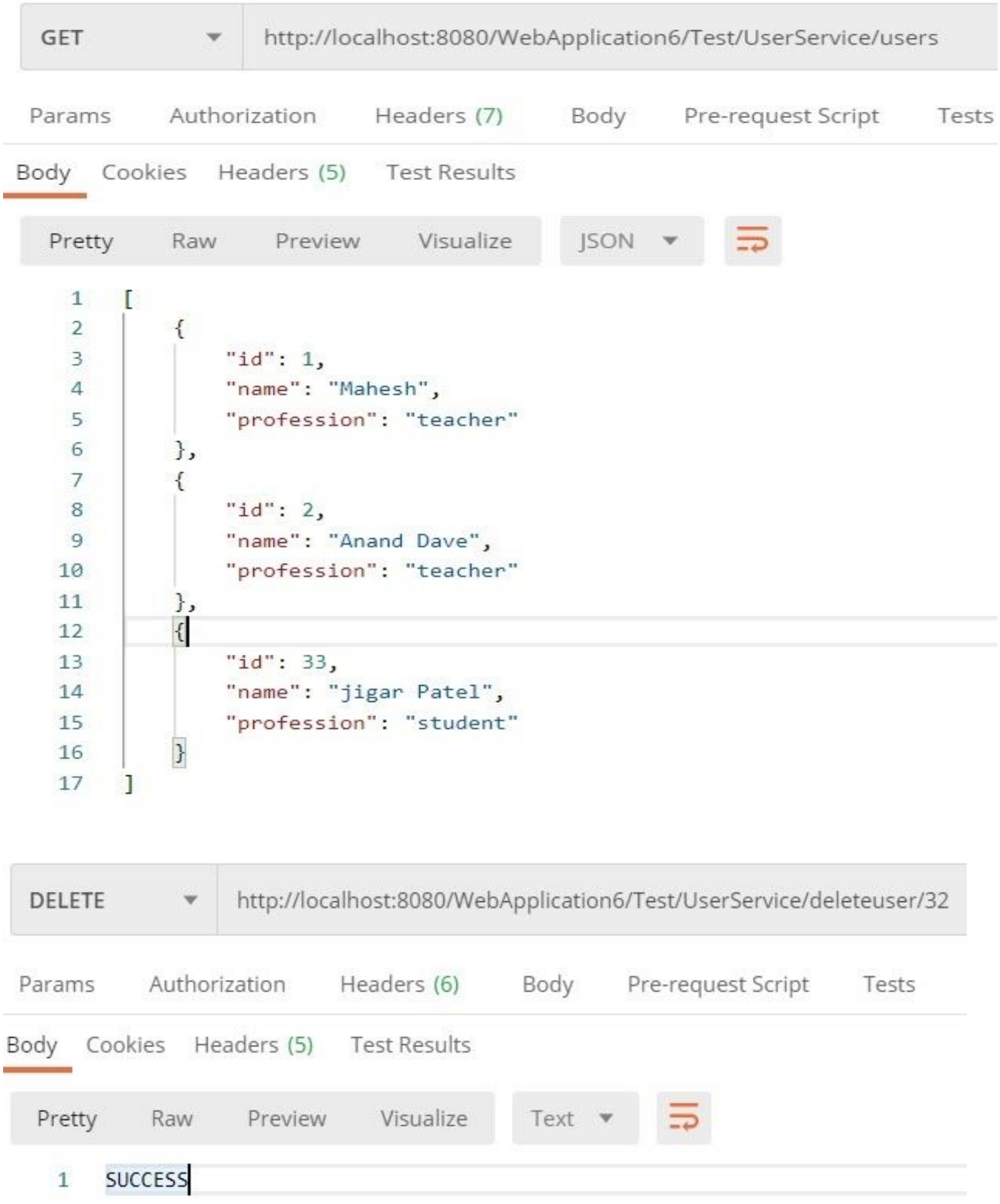
```
}
public void saveUserList(List<User> userList1) {
try {
FileOutputStream fos;
fos = new FileOutputStream(file);
ObjectOutputStream oos = new ObjectOutputStream(fos);
oos.writeObject(userList1);
Lab Work: Distributed Computing IT053
Department of Information Technology, Dharmsinh Desai University 42
oos.close();
} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
public User getUser(int id) {
List<User> users = getAllUsers();
for (User user : users) {
if (user.getId() == id) {
System.out.println("getID is" + user.getId() + "AND ID is" + id);
return user;
}
}
return null;
}
public int deleteUser(int id) {
System.out.println("Inside Delete User Method of User Context");
List<User> userList = getAllUsers();
for (User user : userList) {
if (user.getId() == id) {
System.out.println("Inside Delete User getID is" + user.getId() + "AND ID is" + id);
int index = userList.indexOf(user);
userList.remove(index);
saveUserList(userList);
return 1;
}
}
return 0;
}
public int updateUser(User pUser) {
List<User> userList = getAllUsers();
for (User user : userList) {
if (user.getId() == pUser.getId()) {
```

```
int index = userList.indexOf(user);
userList.set(index, pUser);
saveUserList(userList);
return 1;
}
}
return 0;
}
```

Output:




POST
http://localhost:8080/WebApplication6/Test/UserService/adduser
Params
Authorization
Headers (9)
Body
Pre-request Script
Tes
none
form-data
x-www-form-urlencoded
raw
binary
G
1 {
2 "id": "33",
3 "name" : "jigar Patel",
4 "profession" : "student"
5 }
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "id": 33,
3 "name": "jigar Patel",
4 "profession": "student"
5 }


GET
http://localhost:8080/WebApplication6/Test/UserService/users
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"name": "Mahesh",
"profession": "teacher"
},
{
"id": 2,
"name": "Anand Dave",
"profession": "teacher"
},
{
"id": 33,
"name": "jigar Patel",
"profession": "student"
}
]
DELETE
http://localhost:8080/WebApplication6/Test/UserService/deleteuser/32
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Body
Cookies
Headers (5)
Test Results
Pretty
Raw
Preview
Visualize
Text
SUCCESS

Experiment 9

Aim: Create Microservice based application using Spring Boot.

Tools: GlassFish Server, Netbean 6.0, Postman.

Code:

1) **Movie.java**

```
package MovieService.models;
public class Movie {
 private int movieId;
 private String name;
 public Movie() {
 }
 public Movie(int movieId, String name) {
 this.movieId = movieId;
 this.name = name;
 }
 public int getMovieId() {
 return movieId;
 }
 public void setMovieId(int movieId) {
 this.movieId = movieId;
 }
 public String getName() {
 return name;
 }
 public void setName(String name) {
 this.name = name;
 }
}
```

2) **Rating.java**

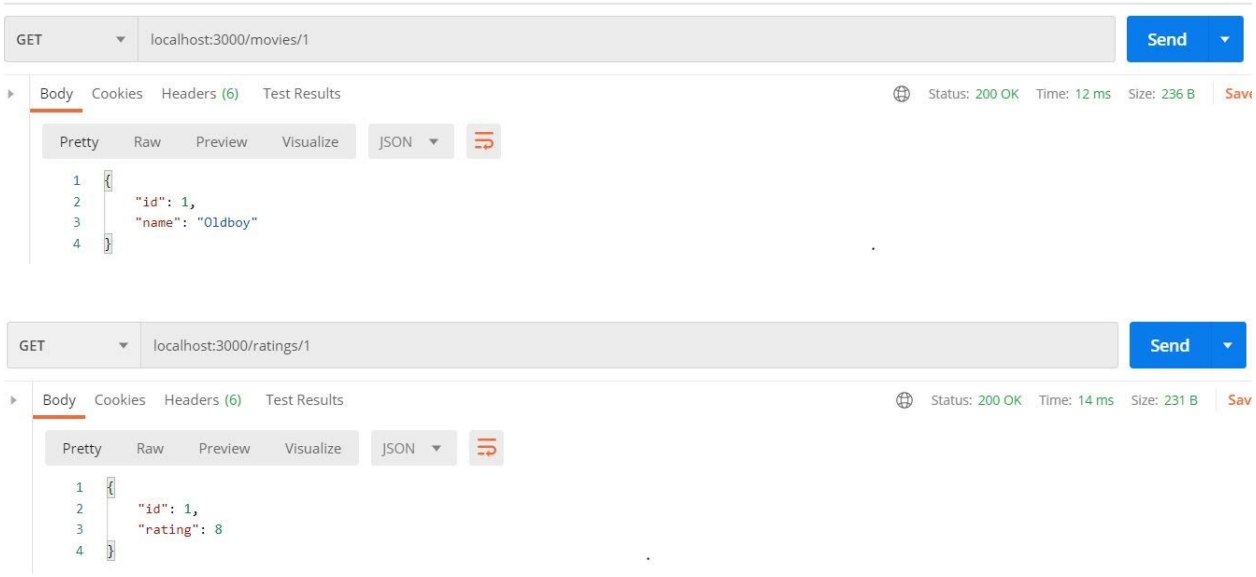
```
package MovieService.models;
public class Rating {
 private String movieId;
 private int ratings;
 public Rating(String movieId, int ratings) {
 this.movieId = movieId;
 this.ratings = ratings;
 }
 public String getMovieId() {
 return movieId;
 }
```

```
 public void setMovieId(String movieId) {
 this.movieId = movieId;
 }
 public int getRatings() {
 return ratings;
 }
 public void setRatings(int ratings) {
 this.ratings = ratings;
 }
}
```

3) **MovieService.java**

```
import MovieService.models.Movie;
import MovieService.models.Rating;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import javax.websocket.server.PathParam;

@RestController
@RequestMapping("/movies")
public class MovieResource {
 @RequestMapping("/{movieId}")
 public Movie getMovieInfo(@PathParam("movieId") String movieId) {
    if(movieID==1){
        return new Movie(1, "Oldboy");
    }else if(movieID==2){
        return new Movie(2, "Silenced");
    }
 }
}
@RestController
@RequestMapping("/ratings")
public class RatingResource {
 @RequestMapping("/{movieId}")
 public Rating getRating(@PathParam("movieId") String movieId) {
    if(movieID==1){
        return new Rating(1, 8);
    }else if(movieID==2){
        return new Rating(2, 9);
    }
 }
}
```

Output:


GET
localhost:3000/movies/1
Send
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 12 ms
Size: 236 B
Pretty
Raw
Preview
Visualize
JSON
{
"id": 1,
"name": "Oldboy"
}

GET
localhost:3000/ratings/1
Send
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 14 ms
Size: 231 B
Pretty
Raw
Preview
Visualize
JSON
{
"id": 1,
"rating": 8
}

Experiment 10

Aim: Implementation JMS based application using Publish-Subscribe paradigm.

Tools: GlassFish Server, NetBeans 6.0, JMS libraries.

Code:

1) **Sender.java**

```
package JMSDemo;
import java.io.IOException;
import javax.naming.*;
import javax.jms.*;

public class Sender {

public static void main(String[] args){
 try
 {
 InitialContext ctx=new InitialContext();
 QueueConnectionFactory
 f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");
 QueueConnection con=f.createQueueConnection();
 con.start();
 QueueSession ses=con.createQueueSession(false,
 Session.AUTO_ACKNOWLEDGE);
 Queue t=(Queue)ctx.lookup("myQueue");
 QueueSender sender=ses.createSender(t);
 TextMessage msg=ses.createTextMessage();
 msg.setText("Hello World");
 sender.send(msg);
 System.out.println("Message successfully sent.");
 con.close();
 }
 catch(Exception e)
 {System.out.println(e);}
 }
}
```

2) **Receiver.java**

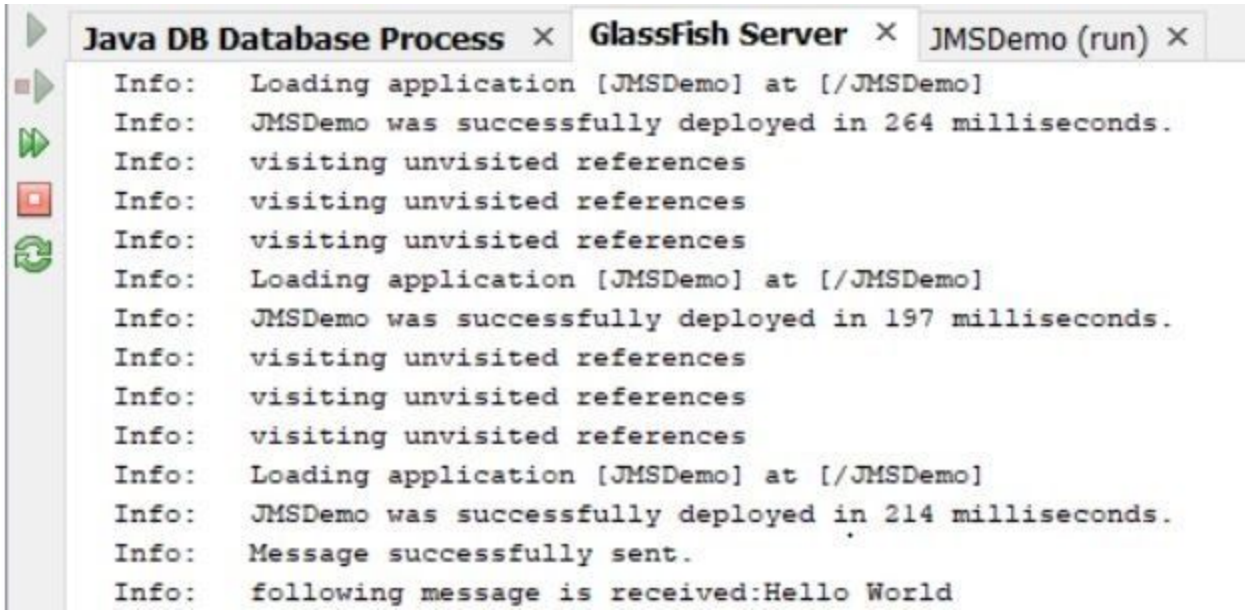
```
import javax.jms.*;
import javax.naming.InitialContext;
public class Receiver{

public static void main(String[] args){
 try{
 InitialContext ctx=new InitialContext();
 QueueConnectionFactory
 f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");
 QueueConnection con=f.createQueueConnection();
 con.start();
 QueueSession ses=con.createQueueSession(false,
 Session.AUTO_ACKNOWLEDGE);
 Queue t=(Queue)ctx.lookup("myQueue");
 QueueReceiver receiver=ses.createReceiver(t);
 MyListener listener=new MyListener();
 receiver.setMessageListener(listener);
 System.out.println("Receiver is waiting for messages...");
 }
 catch(Exception e){System.out.println(e);}

 }
 }
```

3) **MyReceiver.java**

```
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;
public class MyListener implements MessageListener {
 @Override
 public void onMessage(Message message) {
 try{
 TextMessage msg=(TextMessage)message;
 System.out.println("following message is received:"+msg.getText());
 }
 catch(JMSException e)
 {System.out.println(e);}
 }
}
```

Output:


Java DB Database Process
GlassFish Server
JMSDemo (run)
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 197 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 214 milliseconds.
Info: Message successfully sent.
Info: following message is received:Hello World
Info: Receiver1 is ready, waiting for messages...


Java DB Database Process
GlassFish Server
JMSDemo (run)
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 264 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 197 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Loading application [JMSDemo] at [/JMSDemo]
Info: JMSDemo was successfully deployed in 214 milliseconds.
Info: Message successfully sent.
Info: following message is received:Hello World