Bharati Vidyapeeth's

Institute of Management & Information Technology

C.B.D. Belapur, Navi Mumbai 400614

Vision:
Providing high quality, innovative and value-based education in information technology to build competent professionals.

Mission
M1. Technical Skills: To provide solid technical foundation theoretically as well as practically capable of providing quality services to industry.
M2. Development: Department caters to the needs of students through comprehensive educational programs and promotes lifelong learning in the field of computer Applications.
M3. Ethical leadership: Department develops ethical leadership insight in the students to succeed in industry, government and academia

CERTIFICATE

This is to certify that the journal is the work of **Mr. Sayas Sonawane** Roll No. **51** of MCA (Sem: - **III** Div: - **B**) For the Academic Year 2021-2023

Subject Code: - MCAL32

Subject Name: Distributed System and Cloud Computing Lab

_____ _____
Subject-in-charge Principal

Date: _____ _____
External Examiner

Bharati Vidyapeeth's Institute of Management & Information Technology

MCA Sem III AY– 2021-23

Subject and Code: MCAL32: Distributed System and Cloud Computing Lab

INDEX

Roll No: 51 **Name: Sayas Sonawane** **Division: B**

Sr. No.	Topic	Date	Sign
1	**Remote Process Communication/ Inter Process Communication:**		
	Develop a multi-client chat server application where multiple clients chat with each other concurrently. The messages sent by different clients are first communicated to the server and then the server, on behalf of the source client, communicates the messages to the appropriate destination client.	13/09/2022	
2	**Remote Procedure Call:**		
	RPC using Datagramsocket a.Implement a Date Time server containing date() and time()	29/09/2022	
	b. Implement a Age calculator server which displays age where the client provide his/her birth year.	06/10/2022	
	c. Implement server which greets clinet according to the current time of the server “good morning”, “good afternoon”, “Good evening” and “Good Night”	06/10/2022	
	d. Implement a Server calculator containing ADD(), MUL(), SUB(), etc using Datagramsocket	15/10/2022	
	e. RPC to implement Equation solver using Datagram. The client should provide an equation to the Server through an interface. The server will solve the expression given by the client. (a-b)2 = a2 –2ab + b2; If a = 5 and b = 2 then return value = 52 – 2.5.2 + 22 = 9.	22/10/2022	
	f. RPC to implement server to print the string is palindrome	22/10/2022	
	g. RPC to implement server to print the if it is palindrome number	29/10/2022	
3	**Remote Method Invocation**		
	The client should provide an equation to the server through an interface. The server will solve the expression given by the client.	17/11/2022	
4	**Remote Object Communication:**		
	a. To retrieve day, time and date function from server to client. This program should display server day, time and date. (Use Concept of JDBC and RMI for accessing multiple data access objects)	20/11/2022	
	b. Using MySQL create Student database (id, name, subject,marks) and retrieve the information from the database using Remote Object Communication concept.	24/11/2022	

	c. Using MySQL create Elecrtic_Bill database. Create table Bill. Create table Bill (consumer_name, bill_due_date, bill_amount) and retrieve the Bill information from the Elecrtic_Bill database using Remote Object Communication concept.	24/11/2022	
5	**Mutual Exclusion:**		
	Implementation of mutual exclusion using Token Ring technique concept -This technique solves the mutual exclusion existing in the process communication.	24/11/2022	
6	**Implementation of Cloud Computing Services:**		
	Implementation of Storage as a Service using Google Docs/AWS	28/11/2022	
7	**Implementation of Identity Management using Cloud Computing concept**		
	Implementation of Identity Management	28/11/2022	
8	**App Development using Cloud Computing**		
	To develop Application for windows Azure / Amazon AWS using Windows Azure Platform Training Kit and Visual Studio.	10/11/2022	
	To develop applications using Google App Engine by using Eclipse IDE	10/11/2022	

Practical No.: 01 Remote Process Communication/ Inter Process Communication

A) Develop a multi-client chat server application where multiple clients chat with each other concurrently. The messages sent by different clients are first communicated to the server and then the server, on behalf of the source client, communicates the messages to the appropriate destination client.

Program:

ServerMain.java

```
import java.io.DataInputStream;
import java.io.PrintStream;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;
public class ServerMain {
public static void main(String args[]) {
ServerSocket echoServer = null;
String line;
DataInputStream is;
PrintStream os;
Socket clientSocket = null;
try {
echoServer = new ServerSocket(2222);
} catch (IOException e) {
System.out.println(e);
}            try {
clientSocket = echoServer.accept();
is = new DataInputStream(clientSocket.getInputStream());
os = new PrintStream(clientSocket.getOutputStream());
/* As long as we receive data, echo that data back to the client.
*/
while (true) {
line = is.readLine();
os.println("From server: " + line);
}
} catch (IOException e) {
System.out.println(e);
}
}
}
```

ClientMain.java

```
import java.io.DataInputStream;
import java.io.PrintStream; import
java.io.BufferedInputStream; import
java.io.IOException; import
java.net.Socket;
import java.net.UnknownHostException;

public class ClientMain {
@SuppressWarnings("deprecation")
public static void main(String[] args) {

Socket clientSocket = null;
DataInputStream is = null;
PrintStream os = null;
DataInputStream inputLine = null;

/*
* Open a socket on port 2222. Open the input and the output streams.
*/
try {
clientSocket = new Socket("localhost", 2222);
os = new PrintStream(clientSocket.getOutputStream());
is = new DataInputStream(clientSocket.getInputStream());
inputLine = new DataInputStream(new BufferedInputStream(System.in));
} catch (UnknownHostException e) {
System.err.println("Don't know about host");
} catch (IOException e) {
System.err.println("Couldn't get I/O for the connection to host");
}
if (clientSocket != null &&os != null && is != null) {
try {
/*
* Keep on reading from/to the socket till we receive the "Ok" from the
* server, once we received that then we break.
*/
System.out.println("The client started. Type any text. To quit it type 'Ok'.");
String responseLine;   os.println(inputLine.readLine());
while ((responseLine = is.readLine()) != null) {
System.out.println(responseLine);
if (responseLine.indexOf("Ok") != -1) {
break;            }
os.println(inputLine.readLine());
}
```

```
os.close();
is.close();
clientSocket.close();
} catch (UnknownHostException e) {
System.err.println("Trying to connect to unknown host: " + e);
} catch (IOException e) {
System.err.println("IOException:  " + e);
}
}
}
}
```

Output

```
ClientMain [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe (09
The client started. Type any text. To quit it type 'Ok'.
hello shubham
From server: hello shubham
```

Practical No.: 02
Remote Procedure Call
RPC using Datagram socket

A) Implement a Date Time server containing date() and time()

Program:

RPCServerTime.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.StringTokenizer;
public class RPCServertime {
DatagramSocket ds;
DatagramPacket dp;
String str,methodName,result;
int val1,val2;
RPCServertime(){
try
{
ds=new DatagramSocket(2222);
byte b[]=new byte[4096];
while(true) {
dp=new DatagramPacket(b, b.length);
ds.receive(dp);
str=new String(dp.getData(),0,dp.getLength());
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
} else{
StringTokenizer st=new StringTokenizer(str," ");
int i=0;
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
}
}
Calendar c=Calendar.getInstance();
```

```
SimpleDateFormat dateFormat=new SimpleDateFormat("dd/MM/yyyy");
Date d = c.getTime();
InetAddress ia=InetAddress.getLocalHost();
if(methodName.equalsIgnoreCase("date"))
{
result=""+dateFormat.format(d);
}
else if(methodName.equalsIgnoreCase("time"))
{
result=""+c.get(Calendar.HOUR_OF_DAY)+" :
"+c.get(Calendar.MINUTE)+":"+c.get(Calendar.SECOND);
}
byte b1[]=result.getBytes();
DatagramSocket ds1=new DatagramSocket();
DatagramPacket dp1=new DatagramPacket(b1,b1.length,ia,1300);
System.out.println("result: "+result+"\n");
ds1.send(dp1);
} }
catch(IOException e)
{
} }
public static void main(String[] args)
{
new RPCServertime();
}
}
```

RPCClientTime.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class RPCclienttime {
RPCclienttime() {
try{
InetAddress ia=InetAddress.getLocalHost();
DatagramSocket ds=new DatagramSocket();
byte b1[]=new byte[50];
DatagramSocket ds1=new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("Enter date for getting current date and enter time for current time\n");
while(true) {
```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
String str =br.readLine();  byte
b[]=str.getBytes();
DatagramPacket dp=new DatagramPacket(b, b.length,ia,2222);
ds.send(dp);
dp=new DatagramPacket(b1, b1.length);
ds1.receive(dp);
String s=new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = "+s+"\n");
} }
catch(IOException e)
{
} }
public static void main(String[] args) {
new RPCclienttime();
}
}
```

Output:

```
RPCclienttime [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\j

RPC Client

Enter date for getting current date and enter time for current time

time

Result = 18 : 47:48

date

Result = 27/10/2021
```

```
RPCServertime [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre
result: 18 : 47:48

result: 27/10/2021
```

B) Implement an Age calculator server which displays age where the client provides his/her birth year.

RPCClientAge.java Program:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class RPCClientAge {

public RPCClientAge() {
try {
InetAddress ia = InetAddress.getLocalHost();
DatagramSocket ds = new DatagramSocket();
byte b1[] = new byte[50];
DatagramSocket ds1 = new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("Enter Your DOB(DD/MM/YYYY) to continue...\n");
while (true) {
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
String str = br.readLine();
byte b[] = str.getBytes();
DatagramPacket dp = new DatagramPacket(b, b.length, ia,
1200);
ds.send(dp);
dp = new DatagramPacket(b1, b1.length);
ds1.receive(dp);
String s = new String(dp.getData(), 0, dp.getLength());
System.out.println("\nResult (Age) = " + s + " \n");
}
} catch (IOException e) {
}
}
public static void main(String[] args) {
RPCClientAge rpcClientAge;
rpcClientAge = new RPCClientAge();
}
}
```

RPCServerAge.java Program:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.Month;
import java.time.Period;
import java.util.Calendar;
import java.util.Date;
import java.util.StringTokenizer;
import java.util.logging.Level;
import java.util.logging.Logger;

public class RPCServerAge {
DatagramSocket ds;
DatagramPacket dp;
String str, methodName, result;
public RPCServerAge() {
try
{
ds = new DatagramSocket(1200);
byte b[] = new byte[4096];
while (true) {
dp = new DatagramPacket(b, b.length);
ds.receive(dp);
str = new String(dp.getData(), 0, dp.getLength());
if (str.equalsIgnoreCase("q")) {
System.exit(1);
} else {
StringTokenizer st = new StringTokenizer(str, " ");
int i = 0;
while (st.hasMoreTokens()) {
String token = st.nextToken();
methodName = token;
}
}
InetAddress ia = InetAddress.getLocalHost();
SimpleDateFormat sdf = new
SimpleDateFormat("dd/MM/yyyy");
Date d = sdf.parse(methodName);
Calendar c = Calendar.getInstance();
```

```
c.setTime(d);
int year = c.get(Calendar.YEAR);
int month = c.get(Calendar.MONTH) + 1;
int date = c.get(Calendar.DATE);
LocalDate l1 = LocalDate.of(year, month, date);
LocalDate now1 = LocalDate.now();
Period diff1 = Period.between(l1, now1);
result = "age:" + String.valueOf(diff1.getYears()) + "years";
byte b1[] = result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new DatagramPacket(b1, b1.length, ia,1300);
System.out.println("result: " + result + "\n");
ds1.send(dp1);
}
} catch (IOException e) {
} catch (ParseException ex) {
Logger.getLogger(RPCServerAge.class.getName()).log(Level.SEVERE, null, ex);
}
}

public static void main(String[] args) {
RPCServerAge rpcServerAge;
rpcServerAge = new RPCServerAge();
}
}
```

Output:

```
RPCClientAge [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe (07-Dec-2021, 10:06:30 pm)

RPC Client

Enter Your DOB(DD/MM/YYYY) to continue...

23/08/1999

Result (Age) = age:22years
```

```
RPCServerAge [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe (07-Dec-2021, 10:06:18 pm)
result: age:22years
```

C) Implement server which greets clinet according to the current time of the server "good morning", "good afternoon", "Good evening" and "Good Night"

ServerGreet.java Program:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.text.SimpleDateFormat;
import java.util.*;

public class ServerGreet {
DatagramSocket ds;
DatagramPacket dp;
String str,methodName,result;
int hours;
int val1,val2;
public ServerGreet(){
try
{
ds=new DatagramSocket(1200);
byte b[]=new byte[4096];
while(true)
{
dp=new DatagramPacket(b, b.length);
ds.receive(dp);
str=new String(dp.getData(),0,dp.getLength());
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
}
else{
StringTokenizer st=new StringTokenizer(str," ");
int i=0;
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
}
}
Calendar c=Calendar.getInstance();
SimpleDateFormat dateFormat=new SimpleDateFormat("dd/MM/yyyy");
Date d = c.getTime();
InetAddress ia=InetAddress.getLocalHost();
hours=c.get(Calendar.HOUR_OF_DAY);
```

```
if(hours>=1 && hours<12)
{
result="Good Morning" +methodName;
}
else if(hours>=12 && hours<16)
{
result="Good Afternoon" +methodName;
}
else if(hours>=16 && hours<21)
{
result="Good Evening" +methodName;
}
else if(hours>=21 && hours<24)
{
result="Good Night" +methodName;
}
byte b1[]=result.getBytes();
DatagramSocket ds1=new DatagramSocket();
DatagramPacket dp1=new DatagramPacket(b1,b1.length,ia,1300);
System.out.println("result: "+result+"\n");
ds1.send(dp1);
}
}
catch(IOException e)
{
}
}
public static void main(String[] args)
{
ServerGreet rpcServerGreet;
rpcServerGreet = new ServerGreet();
}
}
```

ClientGreet.java Program:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class ClientGreet {
public ClientGreet() {
```

```
try{
InetAddress ia=InetAddress.getLocalHost();
DatagramSocket ds=new DatagramSocket();
byte b1[]=new byte[50];
DatagramSocket ds1=new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("\nWelcome\n");
System.out.println("\nEnter your Name\n");
while(true){
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
String str =br.readLine();
byte b[]=str.getBytes();
DatagramPacket dp=new DatagramPacket(b, b.length,ia,1200);
ds.send(dp);
dp=new DatagramPacket(b1, b1.length);
ds1.receive(dp);
String s=new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = "+s+"\n");
}
}
catch(IOException e){
}
} public static void main(String[] args)
{
{ClientGreet rpcClientGreet;
rpcClientGreet = new ClientGreet();
}
}
}
```

Output:

```
ClientGreet [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\j

RPC Client


Welcome


Enter your Name

Shubham

Result = Good EveningShubham
```

```
ServerGreet [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jr
result: Good EveningShubham
```

D) Implement a Server calculator containing ADD(), MUL(), SUB(), etc using Datagram socket

Program:

RPCClientCalc.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class RPCClientCalc
{
public RPCClientCalc()
{ try{
InetAddress ia=InetAddress.getLocalHost();
DatagramSocket ds=new DatagramSocket();
DatagramSocket ds1=new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("Enter the method name line E.g add 3 4\n");
System.out.println("Enter the method name line E.g sub 3 4\n");
System.out.println("Enter the method name line E.g mul 3 4\n");
System.out.println("Enter the method name line E.g div 10 2\n");
while(true) {
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
String str =br.readLine();
byte b[]=str.getBytes();
DatagramPacket dp=new DatagramPacket(b, b.length,ia,1200);
ds.send(dp); //receiving data
dp=new DatagramPacket(b, b.length);
ds1.receive(dp);
String s=new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = "+s+"\n");
} }
catch(IOException e)
{
}
}
public static void main(String[] args) {
RPCClientCalc rpcClientCalc;
rpcClientCalc = new RPCClientCalc();
}
}
```

RPCServerCalc.java

```
import java.io.IOException;
import java.net.*;
import java.util.*;
public class RPCServerCalc
{
DatagramSocket ds;
DatagramPacket dp;
String str,methodName,result;
int val1,val2;
public RPCServerCalc()
{
try
{
ds=new DatagramSocket(1200);
byte b[]=new byte[4096];
while(true) {
dp=new DatagramPacket(b, b.length);
ds.receive(dp);
str=new
String(dp.getData(),0,dp.getLength())
; if(str.equalsIgnoreCase("q"))
{
System.exit(1);
} else{
StringTokenizer st=new StringTokenizer(str," ");
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
val1=Integer.parseInt(st.nextToken());
val2=Integer.parseInt(st.nextToken());
}
}
System.out.println(str);
InetAddress ia=InetAddress.getLocalHost();
if(methodName.equalsIgnoreCase("add"))
{
result=""+add(val1,val2);
}
else if(methodName.equalsIgnoreCase("sub"))
{
result=""+sub(val1,val2);
}
```

```
else if(methodName.equalsIgnoreCase("mul"))
{
result=""+mul(val1,val2);
}
else if(methodName.equalsIgnoreCase("div"))
{
result=""+div(val1,val2);
}
byte b1[]=result.getBytes();
DatagramSocket ds1=new DatagramSocket();
DatagramPacket dp1=new DatagramPacket(b1,b1.length,ia,1300);
System.out.println("result: "+result+"\n"); ds1.send(dp1);
}
}
catch(IOException | NumberFormatException e)
{
} }
 private int mul(int val1, int val2) { return val1*val2;
 //To change body of generated methods, choose Tools | Templates.
}
private int div(int val1, int val2)
{
return val1/val2;
}
private int add(int val1, int val2)
{
return val1+val2;
}
private int sub(int val1, int val2)
{
return val1-val2;
}
public static void main(String[] args)
{
RPCServerCalc rpcServerCalc= new RPCServerCalc();
}
}
```

Output:

```
RPCClientCalc [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\j

RPC Client

Enter the method name line E.g add 3 4

Enter the method name line E.g sub 3 4

Enter the method name line E.g mul 3 4

Enter the method name line E.g div 10 2

add 3 4

Result = 7

sub 5 4

Result = 1

mul 3 8

Result = 24

div 40 8

Result = 5
```

```
RPCServerCalc [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jr
add 3 4
result: 7

sub 5 4
result: 1

mul 3 8
result: 24

div 40 8
result: 5
```

E) RPC to implement Equation solver using Datagram. The client should provide an equation to the Server through an interface. The server will solve the expression given by the client. (a-b)2 = a2 –2ab + b2; If a = 5 and b = 2 then return value = 52 – 2.5.2 + 22 = 9.

Program:

serverEqSolve.java

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.rmi.RemoteException;
import java.util.StringTokenizer;

public class serverEqSolve {
DatagramSocket ds;
DatagramPacket dp;
String str,result;
int v1,v2,power;
String operator, mName;

public serverEqSolve() {
try {
ds=new DatagramSocket(1200);
byte b[]=new byte[4096];
System.out.println("RPC Server...");
while(true) {
dp=new DatagramPacket(b,b.length);
ds.receive(dp);
str=new String(dp.getData(),0,dp.getLength());
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
}
else {
StringTokenizer st=new StringTokenizer(str," ");
while(st.hasMoreTokens()) {
v1=Integer.parseInt(st.nextToken());
operator=st.nextToken();
mName=operator;
v2=Integer.parseInt(st.nextToken());
power =Integer.parseInt(st.nextToken());
}
}
System.out.println(" Equation : " + str);
```

```
InetAddress ia=InetAddress.getLocalHost();
if(mName.equalsIgnoreCase("-")&& power == 2) {
result=" "+ solveEq1(v1,v2);
//operator = null;
}
if(mName.equalsIgnoreCase("+")&& power == 2) {
   result=" "+ solveEq2(v1,v2);
//operator = null;
}
if(mName.equalsIgnoreCase("-")&& power == 3) {
   result=" "+ solveEq3(v1,v2);
//operator = null;
}
if(mName.equalsIgnoreCase("+")&& power == 3) {
   result=" "+ solveEq4(v1,v2);
//operator = null;
}
byte b1[]=result.getBytes();
DatagramSocket ds1=new DatagramSocket();
DatagramPacket dp1=new DatagramPacket(b1,b1.length,ia,1300);
System.out.println("Result: "+result+"\n");
ds1.send(dp1);
}
}
catch(Exception e) {
System.out.println(e);
}
}
public int solveEq1(int v1, int v2) {
int ans = (v1*v1)-(2*v1*v2)+(v2*v2);
return ans;
}
public int solveEq2(int v1, int v2) {
int ans = (v1*v1)+(2*v1*v2)+(v2*v2);
return ans;
}
public int solveEq3(int a,int b){
int ans = (a*a*a)-(3*a*a*b)+(3*a*b*b)-(b*b*b);
return ans ;
}
public int solveEq4(int a,int b) throws RemoteException
{
int ans=(a*a*a)+(3*a*a*b)+(3*a*b*b)+(b*b*b);  return ans;
}
public static void main(String[] args) {
```

```
serverEqSolve s = new serverEqSolve();
}
}
```

clientEqSolve.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class clientEqSolve {
clientEqSolve(){
try {
int num1, num2, choice;
InetAddress i=InetAddress.getLocalHost();
DatagramSocket ds=new DatagramSocket();
DatagramSocket ds1=new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("Equations:-");

System.out.println("1. For(a-b)^2 enter : a - b 2");
System.out.println("2. For(a+b)^2 enter : a + b 2");
System.out.println("3. For(a-b)^3 enter : a - b 3");
System.out.println("4. For(a+b)^3 enter : a + b 3");
while(true) {
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
String st=br.readLine();
byte b[]=st.getBytes();
DatagramPacket dp=new DatagramPacket(b,b.length,i,1200);
ds.send(dp);
dp=new DatagramPacket(b,b.length);
ds1.receive(dp);
String res=new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult= "+res+"\n");
}
}
catch(Exception ex) {
System.out.println(ex);
}
}
public static void main(String[] args) {
```

```
clientEqSolve c = new clientEqSolve();
}
}
```

Output:

```
clientEqSolve [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe  (09-Dec-2021, 9:22:18 pm)

RPC Client

Equations:-
1. For(a-b)^2 enter : a - b 2
2. For(a+b)^2 enter : a + b 2
3. For(a-b)^3 enter : a - b 3
4. For(a+b)^3 enter : a + b 3
5 - 3 2

Result=  4

5 + 3 2

Result=  64

5 - 3 3

Result=  8

5 + 3 3

Result=  512
```

```
serverEqSolve [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe  (09-Dec-2021, 9:22:12 pm)
RPC Server...
 Equation : 5 - 3 2
Result:  4

 Equation : 5 + 3 2
Result:  64

 Equation : 5 - 3 3
Result:  8

 Equation : 5 + 3 3
Result:  512
```

F) RPC to implement server to print the string is palindrome

String Program:
ServerPali.java

```
import java.io.*;
import java.net.*;
public class ServerPali {
ServerSocket ss;
Socket socket;
BufferedReader sock_in,kdb_in;
PrintWriter sock_out;
String str;
public ServerPali()
{
try{
ss=new ServerSocket(8765);
socket=ss.accept();
kdb_in=new BufferedReader(new InputStreamReader(System.in));
sock_in=new BufferedReader(new InputStreamReader(socket.getInputStream()));
sock_out=new PrintWriter(socket.getOutputStream());
while(true)
{
str=sock_in.readLine();
int k=str.length();
System.out.println(str);
int left=0,right=k-1;int flag=1;
while(left<=right)
{
if(str.charAt(left)!=(str.charAt(right)))
{
flag=0;
break;
}
else
{
left++;right--;
}
}
if(flag==1)
str="Palindrome";
else
```

```
str="Not Palindrome”;
sock_out.println(str);
sock_out.flush();
if(str.equals("bye"))
break;
}
}catch (Exception e) { }
}
public static void main(String arg[])
{
new ServerPali();
}}
```

ClientPali.java

```
import java.io.*;
import java.net.*;
public class ClientPali{
Socket socket;
BufferedReader sock_in,kdb_in;
PrintWriter sock_out;
String str;
public ClientPali()
{
try{
Socket socket=new Socket("127.0.0.1",8765);
kdb_in=new BufferedReader(new InputStreamReader(System.in));
 sock_in=new BufferedReader(new InputStreamReader(socket.getInputStream()));
sock_out=new PrintWriter(socket.getOutputStream());
while(true)
{
System.out.println("Enter the msg");
str=kdb_in.readLine();
sock_out.println(str);
sock_out.flush();
str=sock_in.readLine();
System.out.println(str);
if(str.equals("q"))
break;
}
socket.close();
}catch (Exception e) { }
}
```

```
public static void main(String arg[])
{
new ClientPali();
}}
```

Output:

```
ClientPali [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jr
Enter the msg
nayan
Palindrome
Enter the msg
shubham
Not Palindrome
ServerPali [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\j
nayan
shubham
```

G) RPC to implement server to print the if it is palindrome number

Program:

RPCServerPalindromeNum.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.StringTokenizer;
public class RPCServerPallindromeNum {
String str,methodName;
RPCServerPallindromeNum(){
try
{
DatagramSocket ds = new DatagramSocket(1200);
byte b[]=new byte[4096];
while(true)
{
DatagramPacket dp = new DatagramPacket(b, b.length);
ds.receive(dp);
String str = new String(dp.getData(),0,dp.getLength());
//String methodName = null;
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
}
else{
StringTokenizer st=new StringTokenizer(str," ");
int i=0;
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
}
}
String str2 = "";
int len = methodName.length();
for(int i=len-1;i>=0;i--)
{
str2 = str2 + methodName.charAt(i);
}
String result;
if(methodName.equals(str2))
```

```
{
result = "Number is Palindrome";
}
else
{
result = "Number is Not Palindrome";
}
byte b1[]=result.getBytes();
DatagramSocket ds1=new DatagramSocket();
InetAddress ia = InetAddress.getLocalHost();
DatagramPacket dp1=new
DatagramPacket(b1,b1.length,ia,1300);
System.out.println("result: "+result+"\n");
ds1.send(dp1);
}
}
catch(IOException e)
{
}
}
public static void main(String[] args) {
RPCServerPallindromeNum s = new RPCServerPallindromeNum();
}
}
```

RPCClientPalindromeNum.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class RPCClientPallindromeNum {
RPCClientPallindromeNum(){
try {
InetAddress ia=InetAddress.getLocalHost();
DatagramSocket ds=new DatagramSocket();
byte b1[]=new byte[50];
DatagramSocket ds1=new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("Enter Number to continue...\n");
while(true)
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
String str =br.readLine();
byte b[]=str.getBytes();
DatagramPacket dp=new DatagramPacket(b, b.length,ia,1200);
ds.send(dp);
dp=new DatagramPacket(b1, b1.length);
ds1.receive(dp);
String s=new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = "+s+"\n");
}
}
catch (IOException e)
{
}
}
public static void main(String[] args) {
RPCClientPallindromeNum c = new RPCClientPallindromeNum();
}
}
```

Output:

```
RPCClientPallindromeNum [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jr

RPC Client

Enter Number to continue...

12321

Result = Number is Palindrome

12213

Result = Number is Not Palindrome
```

```
RPCServerPallindromeNum [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jr
result: Number is Palindrome

result: Number is Not Palindrome
```

Practical No.: 03 Remote Method Invocation

A) The client should provide an equation to the server through an interface. The server will solve the expression given by the client.

Program:

intfEqSolve.java

```
package rmi_eqsolve;
import java.rmi.*;
public interface intfEqSolve extends Remote
{
public int solveEq1(int a,int b)throws RemoteException;
public int solveEq2(int a,int b)throws RemoteException;
public int solveEq3(int a,int b)throws RemoteException;
public int solveEq4(int a,int b)throws RemoteException;
}
```

implEqSolve.java

```
package rmi_eqsolve;
import java.rmi.*;
import java.rmi.server.*;
public class implEqSolve extends UnicastRemoteObject implements intfEqSolve{
protected implEqSolve() throws RemoteException {
super();
}
public static void main(String[] args) {
}

@Override
public int solveEq1(int a, int b) throws RemoteException {
int ans = (a*a)-(2*a*b)+(b*b);
return ans;
}
@Override
public int solveEq2(int a, int b) throws RemoteException {
int ans = (a*a)+(2*a*b)+(b*b);
return ans;
}
@Override
public int solveEq3(int a, int b) throws RemoteException {
int ans = (a*a*a)-(3*a*a*b)+(3*a*b*b)-(b*b*b);
return ans ;
}
```

```
@Override
public int solveEq4(int a, int b) throws RemoteException {
int ans=(a*a*a)+(3*a*a*b)+(3*a*b*b)+(b*b*b);
return ans;
}
}
```

serverEqSolve.java

```
package rmi_eqsolve;

import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class serverEqSolve {
public static void main(String[] args) {
try{
implEqSolve obj =new implEqSolve();
Registry registry = LocateRegistry.createRegistry(1099);
Naming.rebind("hello",obj);
System.err.println("Server ready");
}
catch(Exception e)
{
System.out.println(e);
 }
}
}
```

clientEqSolve.java

```
package rmi_eqsolve;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.rmi.Naming;

public class clientEqSolve {

public static void main(String[] args) {
try {
int num1, num2, res=0, choice;

intfEqSolve object =(intfEqSolve)Naming.lookup("hello");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Equations:-");
```

```
System.out.println("1.(a-b)2");
System.out.println("2.(a+b)2");
System.out.println("3.(a-b)3");
System.out.println("4.(a+b)3");
System.out.println("Choose the equation:");
choice=Integer.parseInt(br.readLine());
System.out.println("Enter the value of a and b");
num1=Integer.parseInt(br.readLine());
num2=Integer.parseInt(br.readLine());
switch(choice)
{
case 1: res=object.solveEq1(num1,num2);
break;

case 2: res=object.solveEq2(num1,num2);
break;

case 3: res=object.solveEq3(num1,num2);
break;

case 4: res=object.solveEq4(num1,num2);
break;

default: System.out.println("Invalid option");
break;
}

System.out.println("The answer is : "+res);
}
catch(Exception ex) {
System.out.println(ex);
}
}
}
```

Output:

```
Console
serverEqSolve (2) [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe  (08-Dec-2021, 12:31:04 pm)
Server ready
```

```
Console
<terminated> clientEqSolve (2) [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe  (0
Equations:-
1.(a-b)2
2.(a+b)2
3.(a-b)3
4.(a+b)3
Choose the equation:
2
Enter the value of a and b
5
3
The answer is : 64
```

Practical No.: 04 Remote Object Communication

A) To retrieve day, time and date function from server to client. This program should display server day, time and date. (Use Concept of JDBC and RMI for accessing multiple data access objects)

Program:

TimeServerInf.java

```
package rmi_datetime;
import java.rmi.*;
public interface TimeServerInf extends Remote
{
public String getTime() throws RemoteException;
}
```

TimeServerImpl.java

```
package rmi_datetime;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
public class TimeServerImpl extends UnicastRemoteObject implements TimeServerInf{
protected TimeServerImpl() throws RemoteException {
super();
// TODO Auto-generated constructor stub
}
@Override
public String getTime() throws RemoteException {
return new java.util.Date().toString();
}
public static void main( String args[] ) throws Exception
{
System.err.println( "Initializing server: please wait.." );
Registry registry = LocateRegistry.createRegistry(1099);
Naming.rebind( "//localhost/Time", new TimeServerImpl() );
System.err.println("The Time Server is up and running." );
}
}
```

TimeClient.java package
rmi_datetime;

```
import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
public class TimeClient {
public static void main(String[] args) throws Exception{
TimeServerInf ts = (TimeServerInf) Naming.lookup( "//localhost/Time" );
System.out.println("Date and Time ==> " + ts.getTime());
}

}
```

Output :

```
Console
TimeServerImpl [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe  (08-Dec-2021, 12:48:45 pm)
Initializing server: please wait..
The Time Server is up and running.
```

```
Console
<terminated> TimeClient [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.ex
Date and Time ==> Wed Dec 08 12:49:26 IST 2021
```

B) Using MySQL create Student database(id, name, subject,marks) and retrieve the information from the database using Remote Object Communication concept.

Idb.java

```
package data;
import java.rmi.*;
public interface IDb extends Remote {
public String getData(String s, String db) throws RemoteException; }
```

DBImpl.java

```
package data;
import java.rmi.*;
import java.rmi.server.*;
import java.sql.*;
public class DBImpl extends UnicastRemoteObject implements IDb {
String str, str1;
public DBImpl() throws RemoteException {
}
public String getData(String sql, String dsn) {
String URL = "jdbc:odbc:" + dsn;
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection(URL);
Statement s = con.createStatement();
ResultSet rs = s.executeQuery(sql);
ResultSetMetaData rsmd = rs.getMetaData();
str = "";
str1 = "";
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
str1 = str1 + rsmd.getColumnName(i) + "\t";
}
System.out.println();
while (rs.next()) {
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
str = str + rs.getString(i) + "\t";
}
str = str + "\n";
}
} catch (Exception e) {
e.printStackTrace();
}
return (str1 + "\n" + str);
}
}
```

DBServer.java

```
package data;
import java.rmi.*;
public class DBServer {
public static void main(String[] args) {
try {
DBImpl di = new DBImpl();
Naming.rebind("rmi://127.0.0.1/DBServer", di);
System.out.println("Server Registered.");
} catch (Exception e1) {
e1.printStackTrace();
}
}
}
```

DBClient.java

```
package data;
import java.rmi.*;
import java.io.*;
public class DBClient {
public static void main(String[] args) {
String db = "", sql = "", ch = "", ch1 = "", res = "";
try {
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
while (true) {
System.out.println("Select an Option");
System.out.println("1:Retrieve Student Information");
System.out.println("2:Exit");
System.out.println("Enter your Choice");
ch = br.readLine();
if (ch.equals("1")) {
db = "clgdb";
System.out.println("Select an Option");
System.out.println("a:Retrieve Student Information");
System.out.println("b:Retrieve Books Informatin");
System.out.println("Enter your Choice:");
ch1 = br.readLine();
if (ch1.equals("a")) {
sql = "select * from Student";
}
else if (ch1.equals("b"))
{
sql = "select * from Book";
}
else {
```

```
System.out.println("Please select an option");
}
}
else if (ch.equals("2"))
{
System.exit(0);
} else {
System.out.println("Please select valid Option");
}
String url = "rmi://127.0.0.1/DBServer";
IDb id = (IDb) Naming.lookup(url);
res = id.getData(sql, db);
System.out.println(res);
}
} catch (Exception e) {
e.printStackTrace();
}
}
}
```

Output:

```
D:\rmidb>set path="C:\Program Files\Java\jdk1.6.0_20\bin"

D:\rmidb>start rmiregistry

D:\rmidb>javac DBImpl.java

D:\rmidb>javac DBServer.java

D:\rmidb>java DBServer
Server Registered.
```

```
D:\>cd rmidb

D:\rmidb>set path="C:\Program Files\Java\jdk1.6.0_20\bin"

D:\rmidb>javac DBClient.java

D:\rmidb>java DBClient
a
ID      fname   age
1       swapnali        21
2       snehal  21
3       tasneem 21
4       mahesh  22
```

C) Using MySQL create Elecrtic_Bill database. Create table Bill (consumer_name, bill_due_date, bill_amount) and retrieve the Bill information from the Elecrtic_Bill database using Remote Object Communication concept.

Idb.java

```
package data;
import java.rmi.*;
public interface IDb extends Remote {
public String getData(String s, String db) throws RemoteException; }
```

DBImpl.java

```
package data;
import java.rmi.*;
import java.rmi.server.*;
import java.sql.*;
public class DBImpl extends UnicastRemoteObject implements IDb {
String str, str1;
public DBImpl() throws RemoteException {
}
public String getData(String sql, String dsn) {
String URL = "jdbc:odbc:" + dsn;
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection(URL);
Statement s = con.createStatement();
ResultSet rs = s.executeQuery(sql);
ResultSetMetaData rsmd = rs.getMetaData();
str = "";
str1 = "";
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
str1 = str1 + rsmd.getColumnName(i) + "\t";
}
System.out.println();
while (rs.next()) {
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
str = str + rs.getString(i) + "\t";
}
str = str + "\n";
}
} catch (Exception e) {
e.printStackTrace();
}
return (str1 + "\n" + str);
}
}
```

DBServer.java

```
package data;
import java.rmi.*;
public class DBServer {
public static void main(String[] args) {
try {
DBImpl di = new DBImpl();
Naming.rebind("rmi://127.0.0.1/DBServer", di);
System.out.println("Server Registered.");
} catch (Exception e1) {
e1.printStackTrace();
}
}
}
```

DBClient.java

```
package data;
import java.rmi.*;
import java.io.*;
public class DBClient {
public static void main(String[] args) {
String db = "", sql = "", ch = "", ch1 = "", res = "";
try {
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
while (true) {
System.out.println("Select an Option");
System.out.println("1:Retrieve MTNL Billing Informatin");
System.out.println("2:Exit");
System.out.println("Enter your Choice");
ch = br.readLine();
if (ch.equals("1")) {
db = "mtnldb";
sql = "Select * from Bill";
} else if (ch.equals("2")) {
System.exit(0);
} else {
System.out.println("Please select valid Option");
}
String url = "rmi://127.0.0.1/DBServer";
IDb id = (IDb) Naming.lookup(url);
res = id.getData(sql, db);
System.out.println(res);
}
} catch (Exception e) {
e.printStackTrace(); }}}
```

Output:

```
D:\rmidb>set path="C:\Program Files\Java\jdk1.6.0_20\bin"

D:\rmidb>start rmiregistry

D:\rmidb>javac DBImpl.java

D:\rmidb>javac DBServer.java

D:\rmidb>java DBServer
Server Registered.
```

```
a
ID      fname   age
1       swapnali        21
2       snehal  21
3       tasneem 21
4       mahesh  22
```

```
ID      cname   phno    amount
1       mahesh  9965321412      500.0000
2       swapnali        7765893214      455.0000
```

Practical No.: 05 Mutual Exclusion

A) Implementation of mutual exclusion using Token Ring technique concept -This technique solves the mutual exclusion existing in the process communication.

Program:

TokenServer.java

```
package MutExclu;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class TokenServer {
public static DatagramSocket ds;
public static DatagramPacket dp;
public static void main(String[] args) throws Exception{
try
{
ds=new DatagramSocket(1000);
}
catch(Exception e)
{
e.printStackTrace();
}
while(true)
{
byte buff[]=new byte[1024]; System.out.println("Server Started...!");
ds.receive(dp=new DatagramPacket(buff, buff.length));
String str=new String(dp.getData(),0,dp.getLength());
System.out.println("Message from"+str);
}
}
}
```

TokenClient1.java

```
package MutExclu;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class TokenClient1
{
public static DatagramSocket ds;
public static DatagramPacket dp;
```

```
public static BufferedReader br;
public static void main(String[] args) throws Exception
{
boolean hasToken;
try
{
ds= new DatagramSocket(100);
}
catch(Exception e)
{
e.printStackTrace();
}
hasToken=true; while(true)
{
if(hasToken==true)
{
System.out.println("Do you want to enter data...(yes/no:)");
br = new BufferedReader(new InputStreamReader(System.in));
String ans = br.readLine();
if(ans.equalsIgnoreCase("Yes"))
{
br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Ready to send");
System.out.println("Sending");
System.out.println("Enter the data");
br = new BufferedReader(new
InputStreamReader(System.in));
String str="Client-1 ===>" + br.readLine();
byte buff[] = new byte[1024];
buff = str.getBytes();
ds.send(new DatagramPacket(buff,buff.length,InetAddress.getLocalHost(),1000));
System.out.println("Now Sending");
}
else if(ans.equalsIgnoreCase("No"))
{
System.out.println("I am busy state");
String msg="Token";
byte bf1[]= new byte[1024];
bf1=msg.getBytes();
ds.send(new DatagramPacket(bf1,bf1.length,InetAddress.getLocalHost(),200));
hasToken=false;
byte bf2[]=new byte[1024];
ds.receive(dp=new DatagramPacket(bf2,bf2.length));
String clientmsg=new String(dp.getData(),0,dp.getLength());
System.out.println("The data is"+clientmsg);
```

```
if(clientmsg.equals("Token"))
hasToken=true;
System.out.println("I am leavino busy state");
}
} else
{
hasToken=true;
System.out.println("I am leaving busy state");
System.out.println("Entering in receiving mode...");
byte bf[]=new byte[1024];
ds.receive(dp=new DatagramPacket(bf,bf.length));
String clientmsg1=new String(dp.getData(),0,dp.getLength());
System.out.println("The data is"+clientmsg1);
{
hasToken=true;
}
}
}
}
}
```

TokenClient2.java

```
package MutExclu;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class TokenClient2 {
static DatagramSocket ds;
static DatagramPacket dp;
static BufferedReader br;
public static void main(String[] args) throws Exception
{try {

ds = new DatagramSocket(200);
} catch (Exception e) {
e.printStackTrace();
}
boolean hasToken = true;
while (true) {
if (hasToken == true) {
System.out.println("Do you want to enter data...(yes/no:)");
```

```
br = new BufferedReader(new InputStreamReader(System.in));
String str = br.readLine();
if (str.equalsIgnoreCase("Yes")) {
System.out.println("Enter the data");
br = new BufferedReader(new
InputStreamReader(System.in));
String msg = "Client-2 ===>" + br.readLine();
byte bf1[] = new byte[1024];
bf1 = msg.getBytes();
ds.send(new DatagramPacket(bf1, bf1.length,InetAddress.getLocalHost(), 1000));
System.out.println("Data Sent");
} else {
String clientmsg = "Token";
byte bf1[] = new byte[1024];
bf1 = clientmsg.getBytes();
ds.send(new DatagramPacket(bf1, bf1.length,InetAddress.getLocalHost(), 100));
hasToken = false;
}
} else
{
try {
byte buff[] = new byte[1024];
System.out.println("Entering in receiving mode...");
ds.receive(dp = new DatagramPacket(buff, buff.length));
String clientmsg1 = new String(dp.getData(), 0, dp.getLength());
System.out.println("The data is" + clientmsg1);
if (clientmsg1.equals("Token"))
hasToken = true;
} catch (Exception e) {
e.printStackTrace();
}
}
}
}
}
```

Output:

```
TokenServer [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe  (09-Dec-2021, 5:52:05 pm)
Server Started...!
Message fromClient-1 ===>hello
Server Started...!
Message fromClient-2 ===>hii
Server Started...!
```

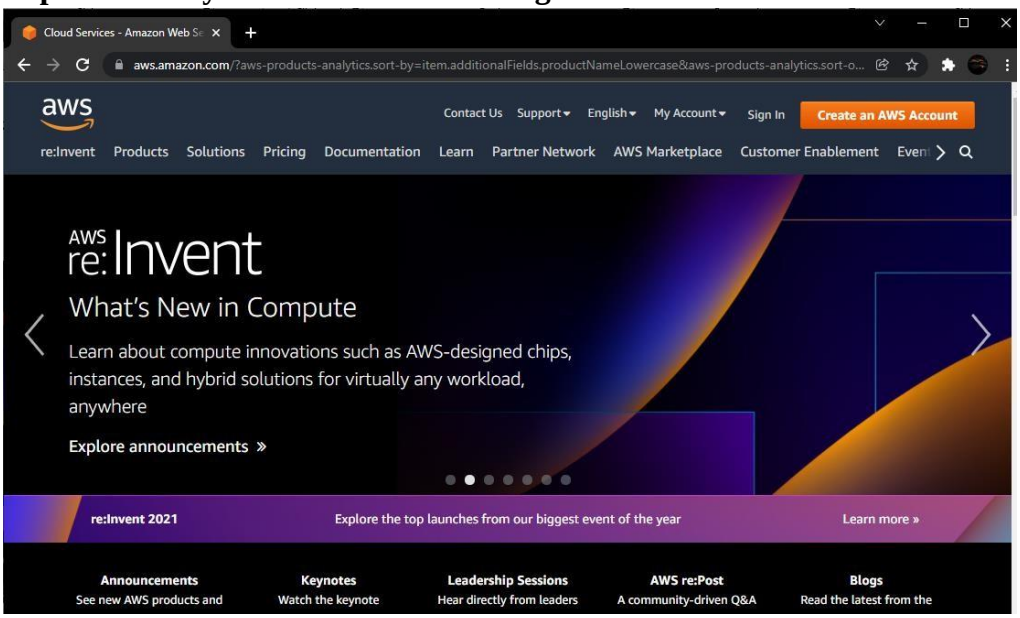
```
TokenClient1 [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe  (09-Dec-2021, 5:52:13 pm)
Do you want to enter data...(yes/no:)
yes
Ready to send
Sending
Enter the data
hello
Now Sending
Do you want to enter data...(yes/no:)
```

```
TokenClient2 [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe  (09-Dec-2021, 5:52:19 pm)
Do you want to enter data...(yes/no:)
yes
Enter the data
hii
Data Sent
Do you want to enter data...(yes/no:)
no
Entering in receiving mode...
```

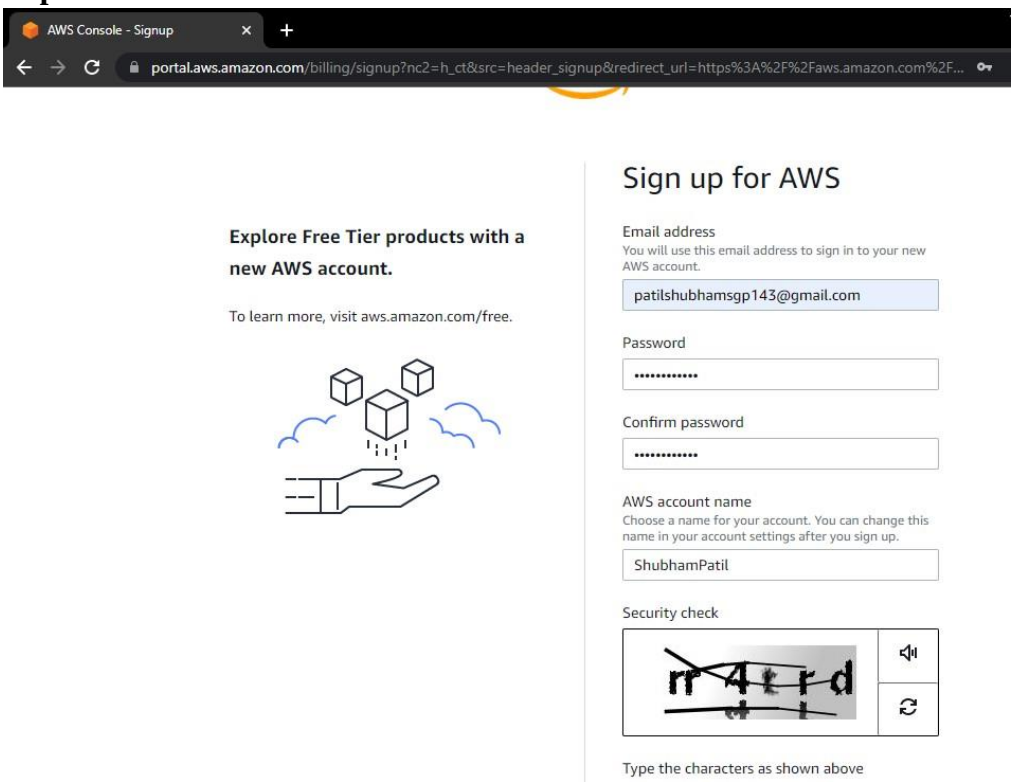
Practical No.: 07 Implementation of Identity Management using Cloud Computing concept

A) Study and implementation of Identity Management

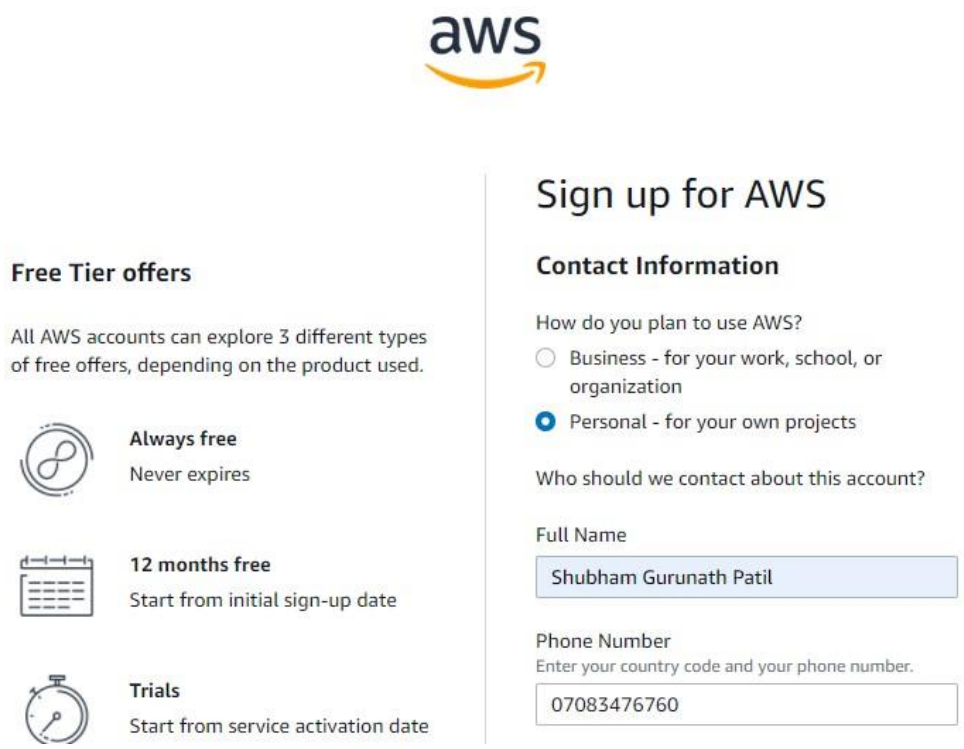
Step1: Open the following link https://aws.amazon.com/
Step2: Go to my Account-> AWS management console



Step3: click on Create new user AWS account
Step4: Fill all the details and click on Continue



Step5: Fill your contact number and Home address and click on create account and continue

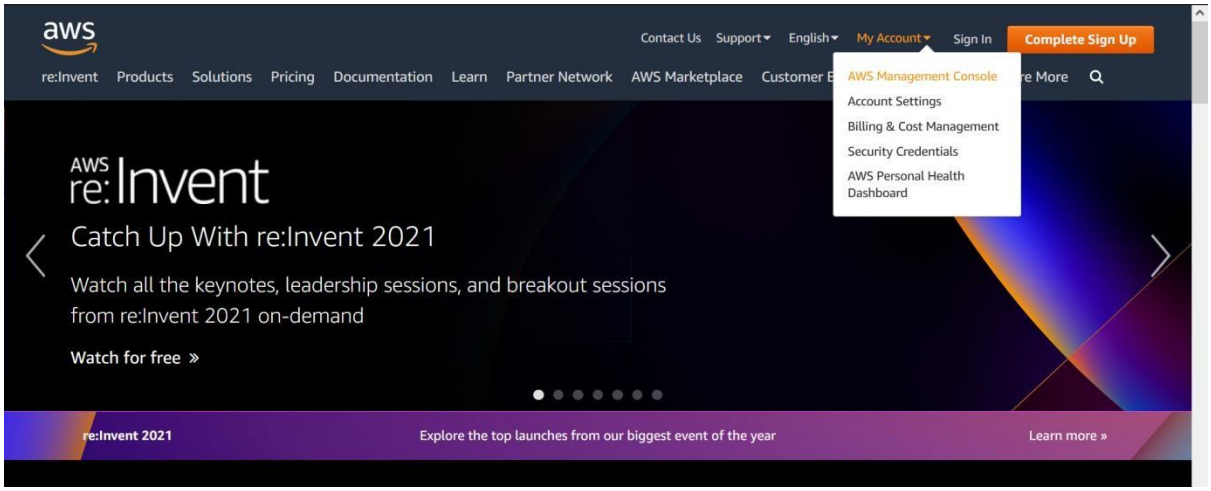


Step6: Now most curtail step AWS will ask for credit card and debit card details. You have to close the browser

Step7: now again open the link https://aws.amazon.com/

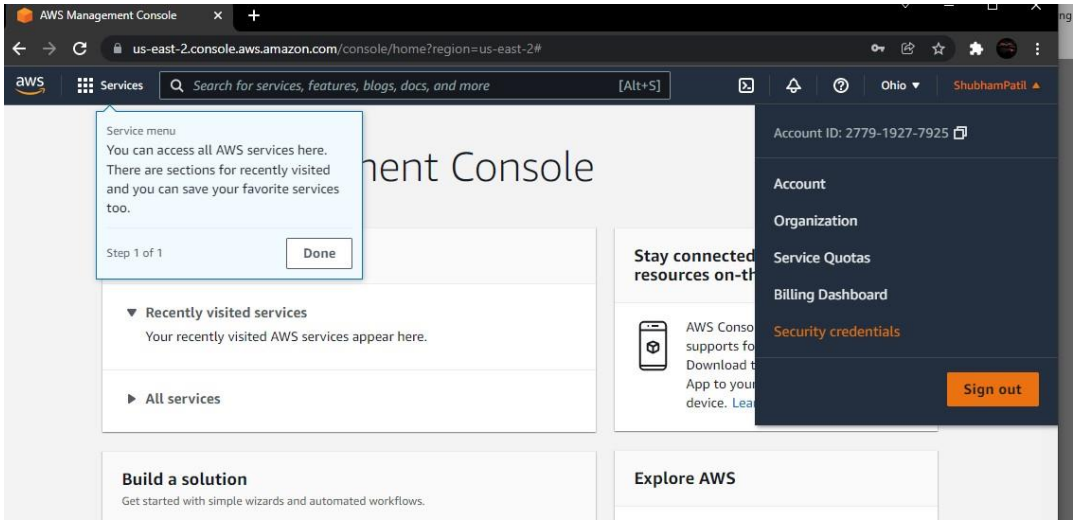
Step8: Go to my Account->AWS Management console

Enter your ID and click on next, After that enter password and click on sign in

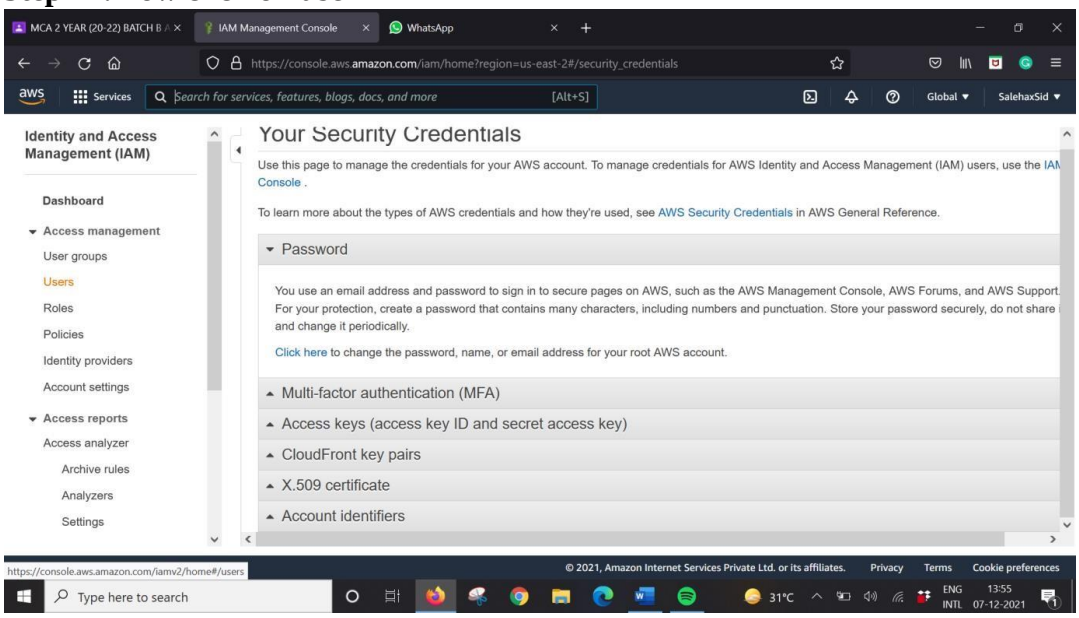


Step 9: you will get the following screen

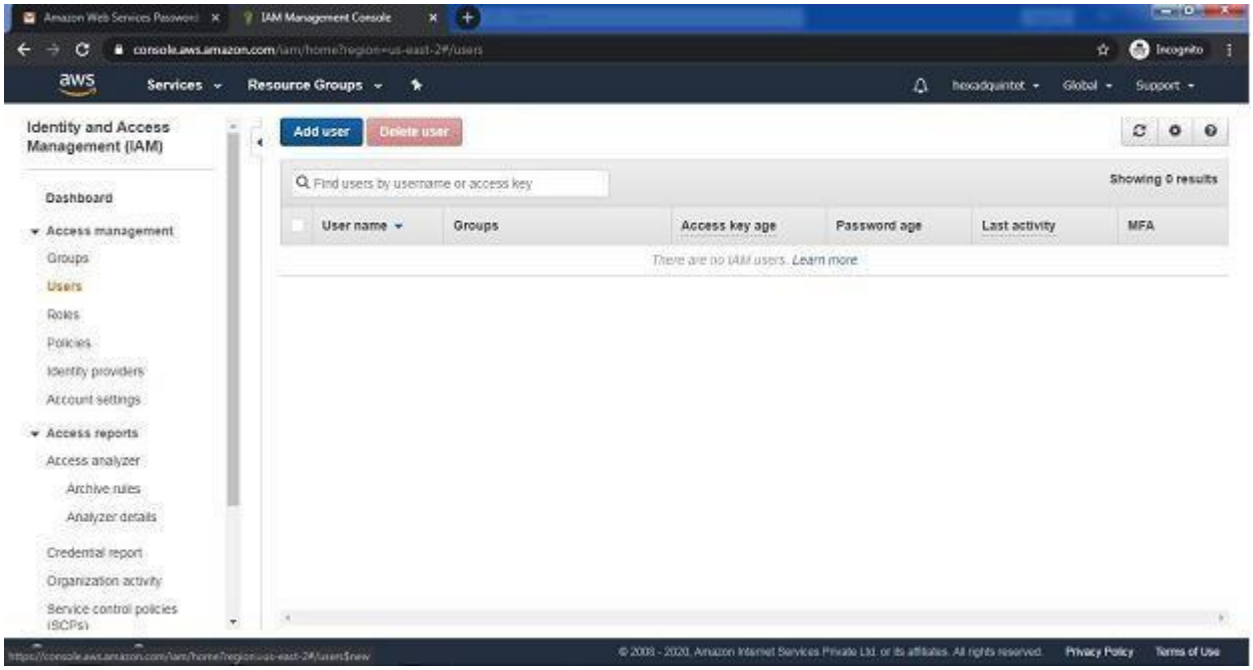
Step 10: Go to My Security credential



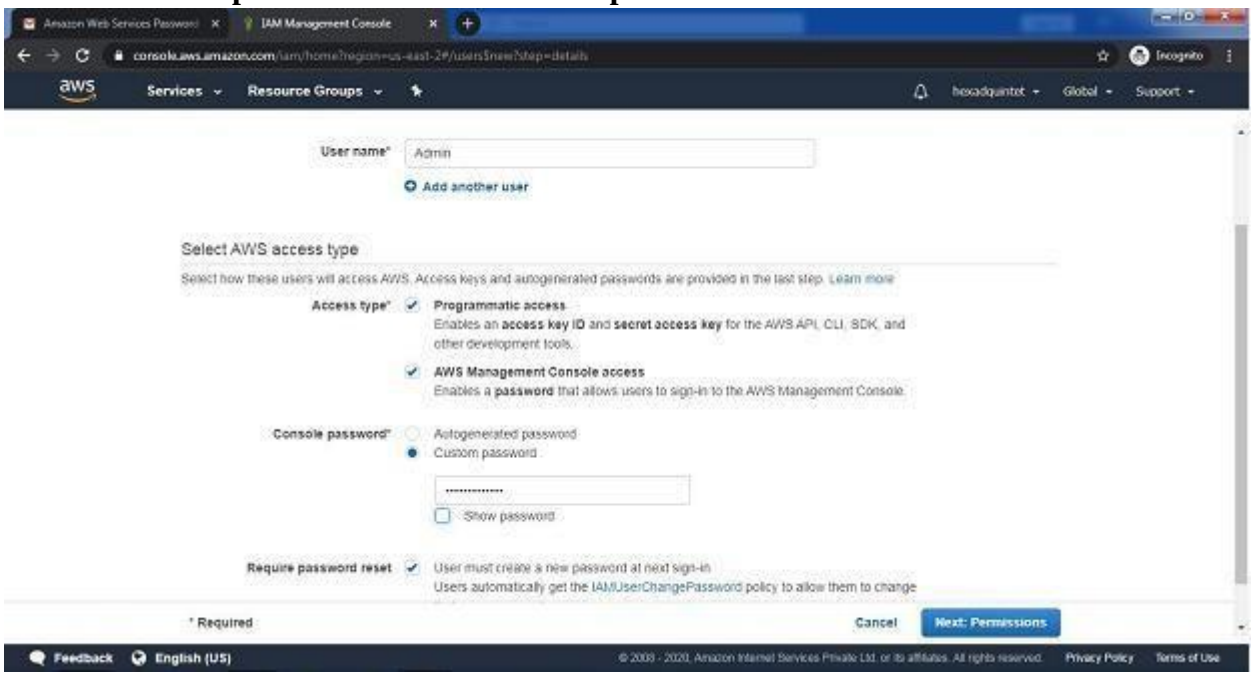
Step 11: now click on user



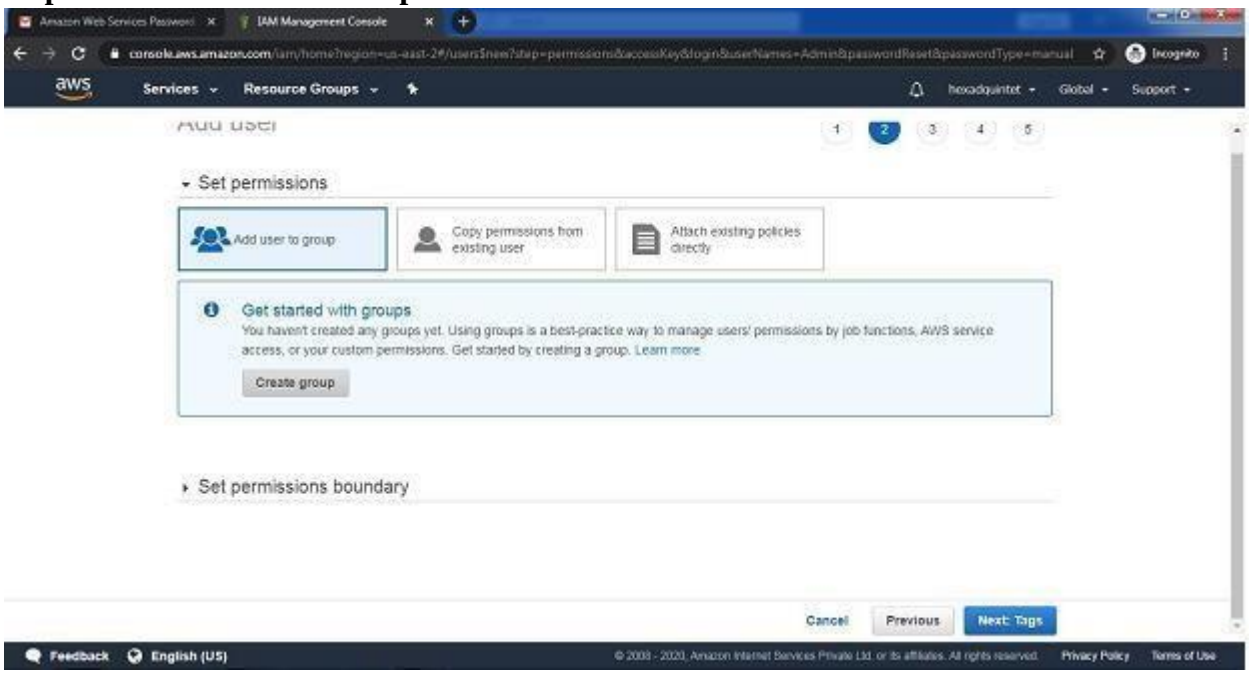
Step 12:Click on add user



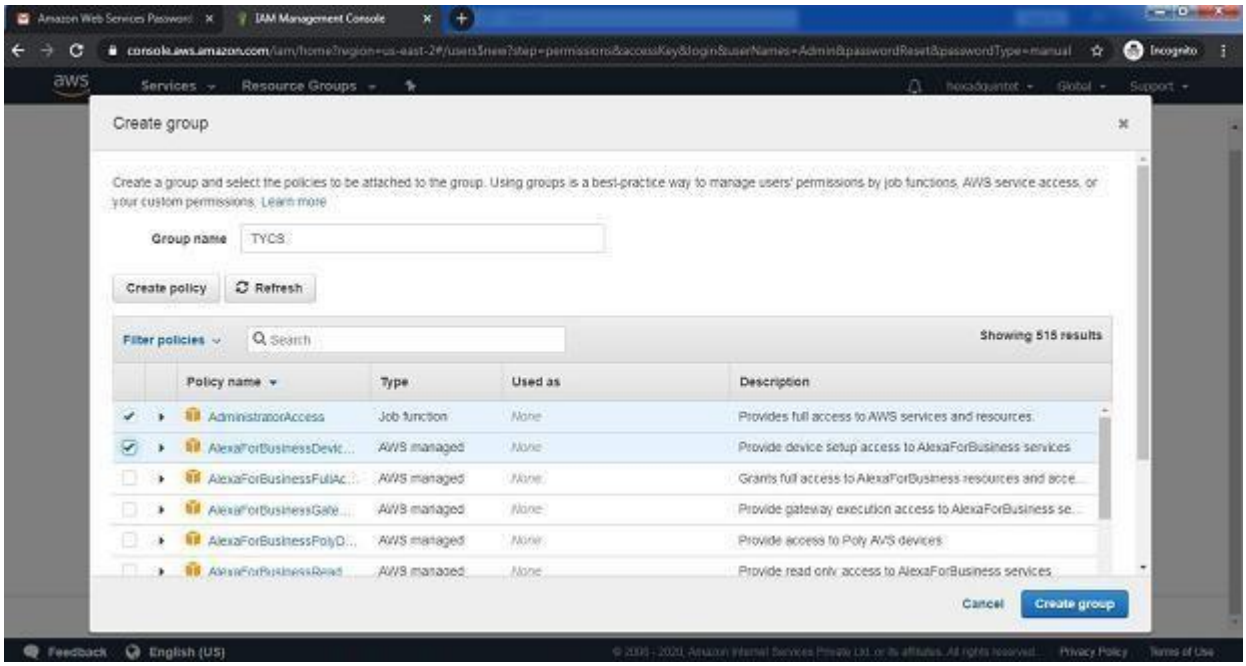
Step 13:Provide the user name and check the check box in front of programmatic access and AWS Management console Access and enter the password for new user Click on custom password and click on next permission



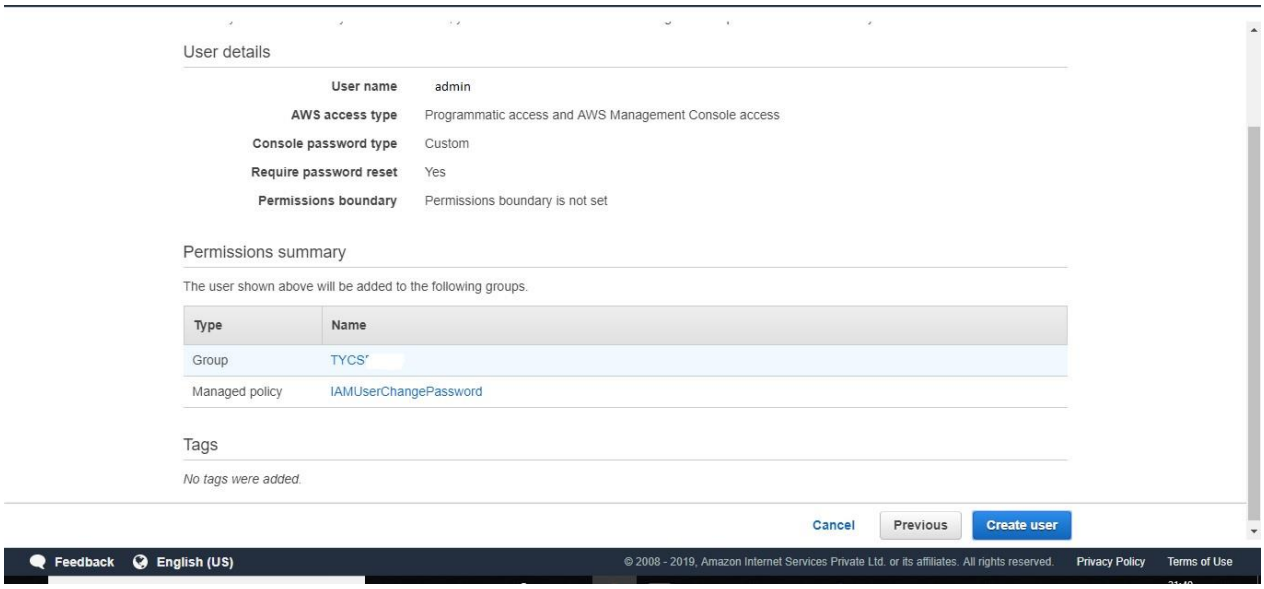
Step 14: click on create Group



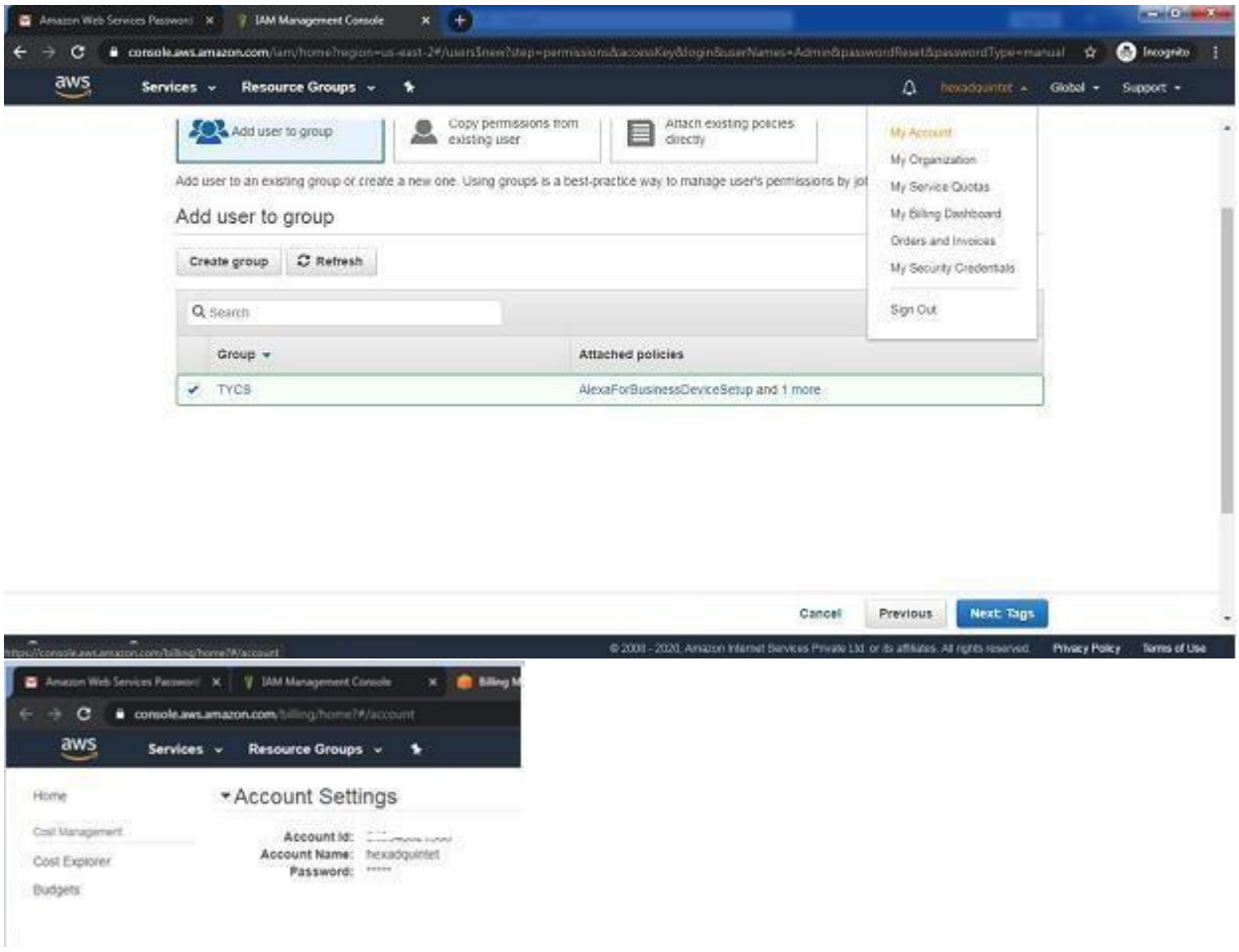
Step15:fill the information and click on Create Group



Step16:click on next tag leave blank , again click on next review leave as it is andclick on create user

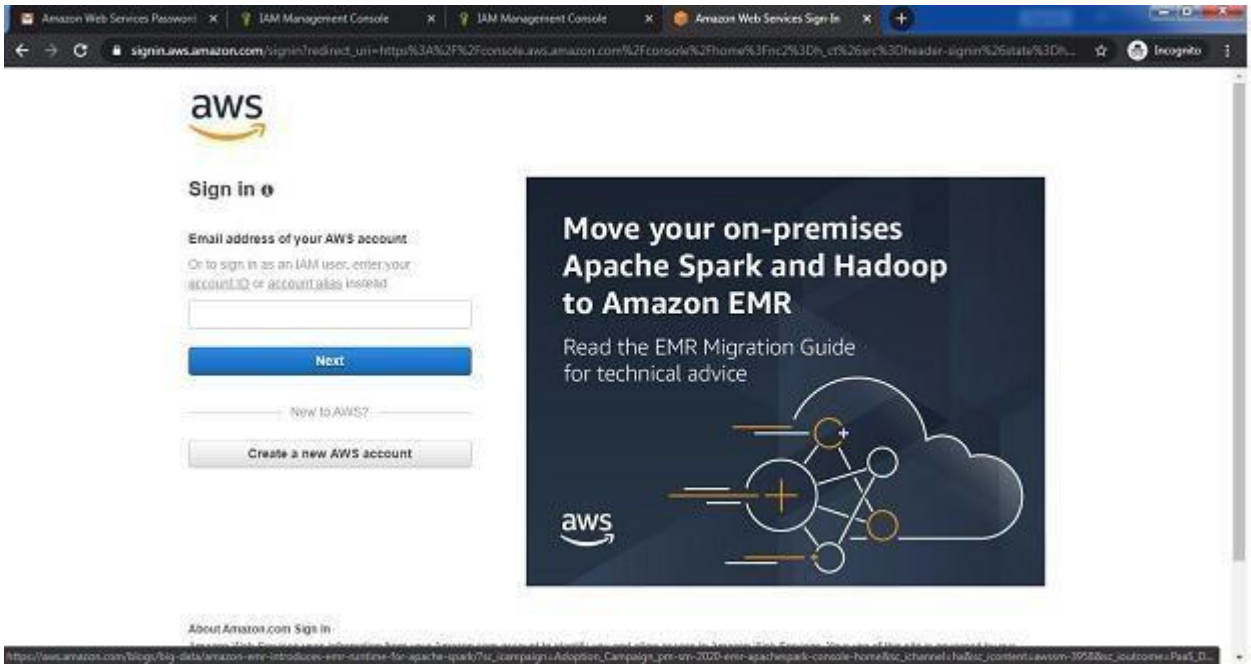


Step 17: click on close and COPY Account ID

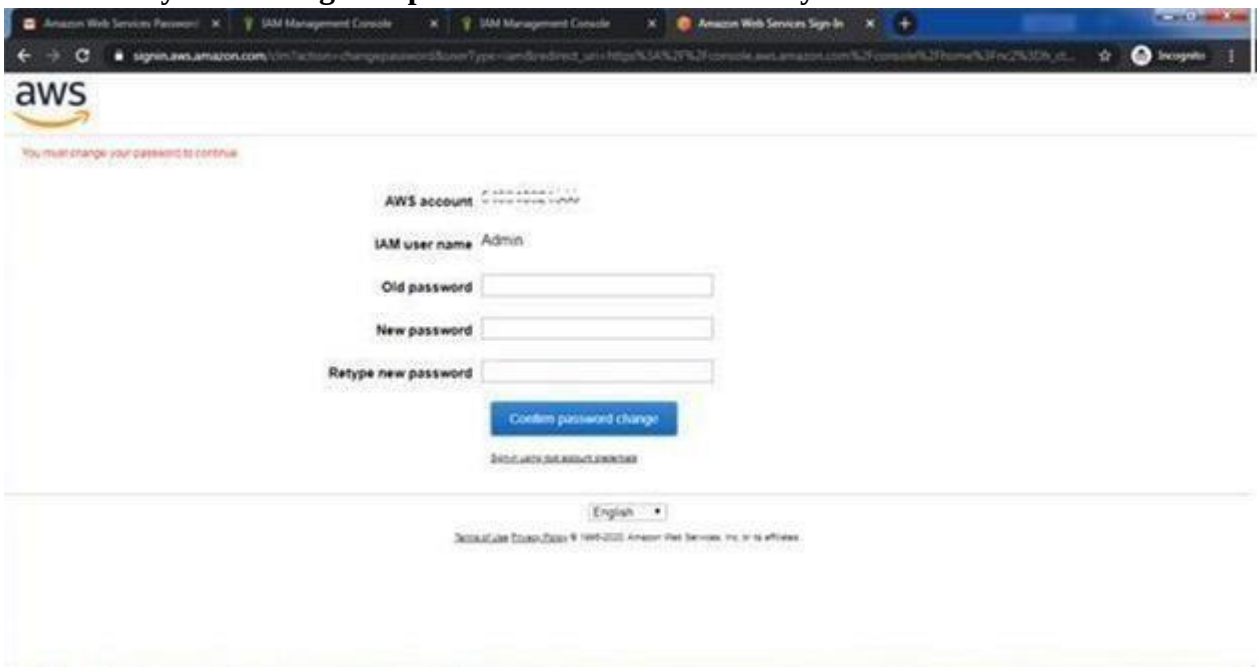


Now logout the admin account and try to login as user(newly created) .

Step 18: again Go to my Account->AWS Management console

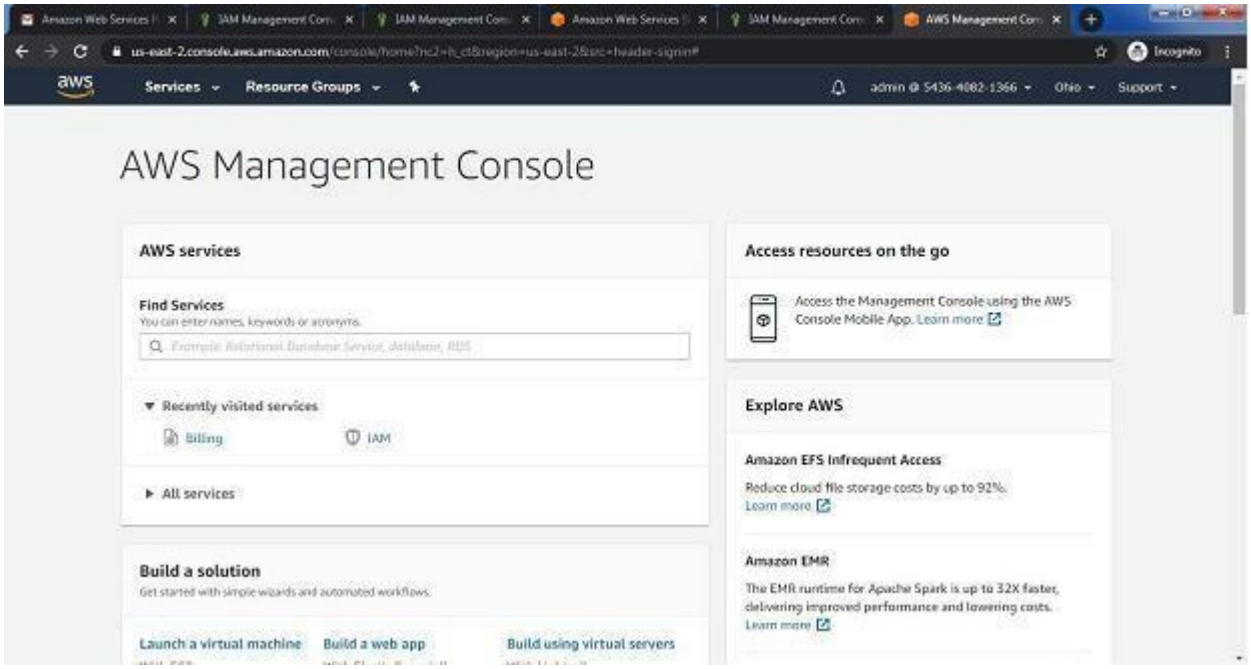


Click on next Provide the Account ID username and password and click on sign in
It will ask you to change the password which is been set by administrator



Name: Sayas Sonawane **Div: B** **Roll No. 51**

You will redirect to home screen

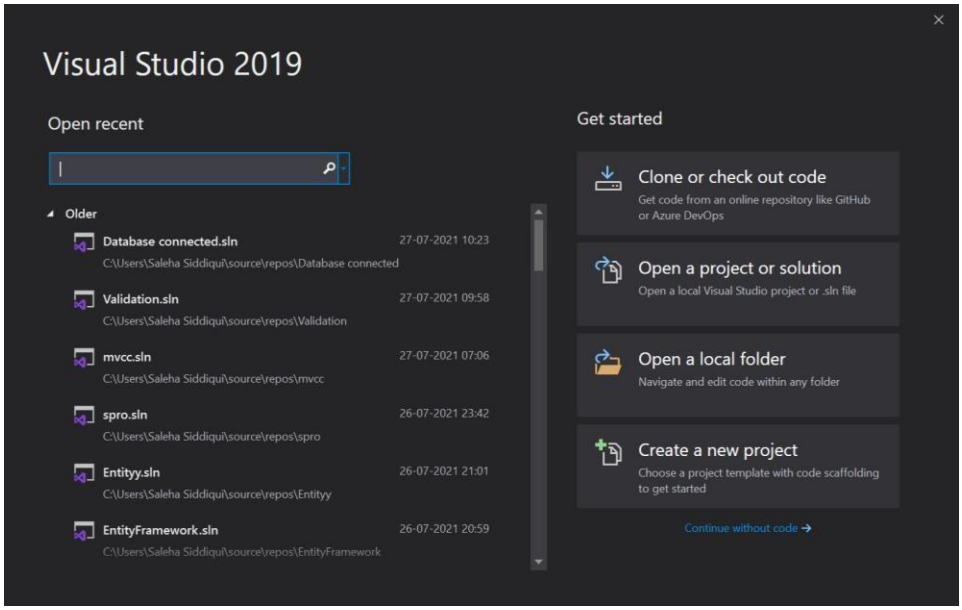


Conclusion: Hence we have studied the concept and implementation of identity management using amazon aws.

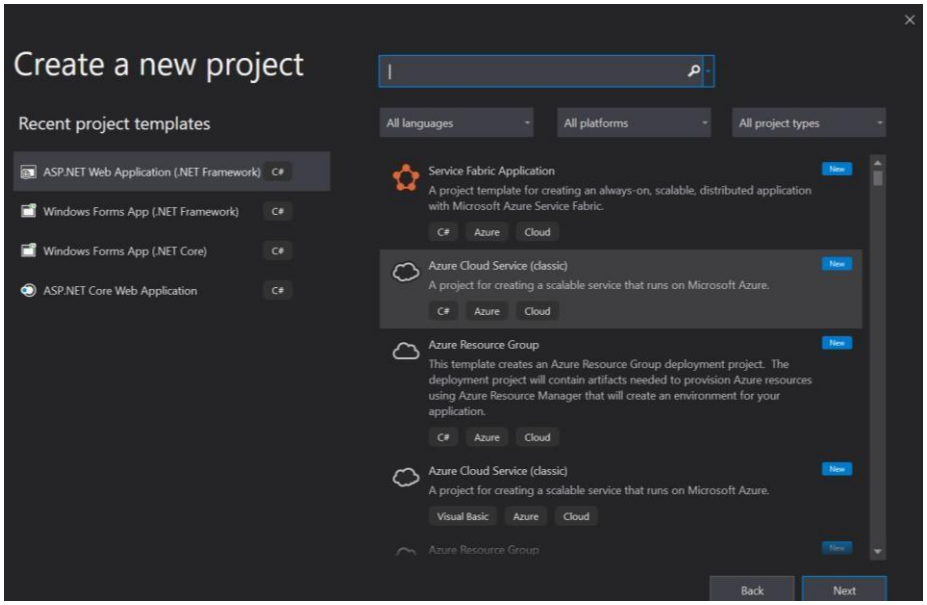
Practical No.: 08 App Development using Cloud Computing

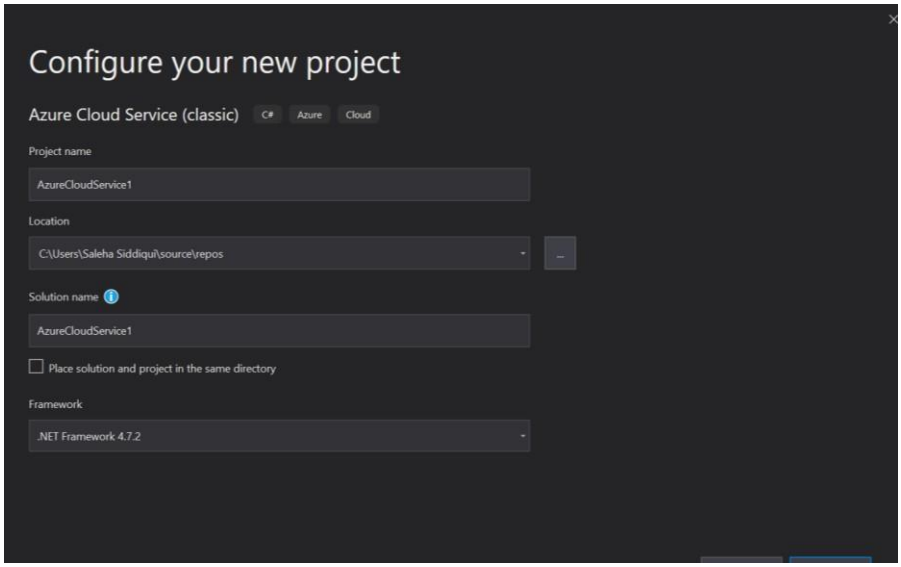
A) To develop Application for windows Azure / Amazon AWS using Windows Azure Platform Training Kit and Visual Studio.

Step 1: open VS2019

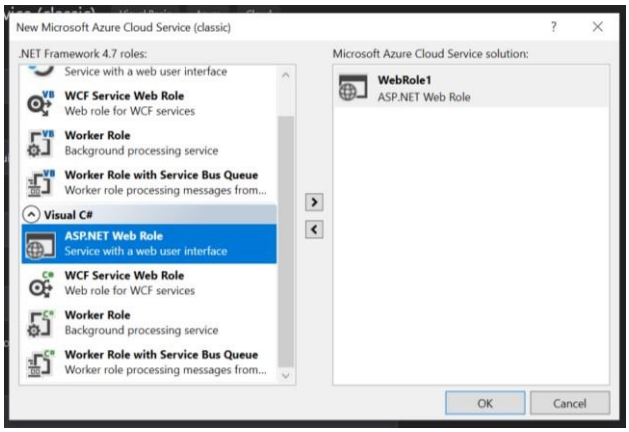


Step 2: Select Create new project > Azure Cloud Service Project

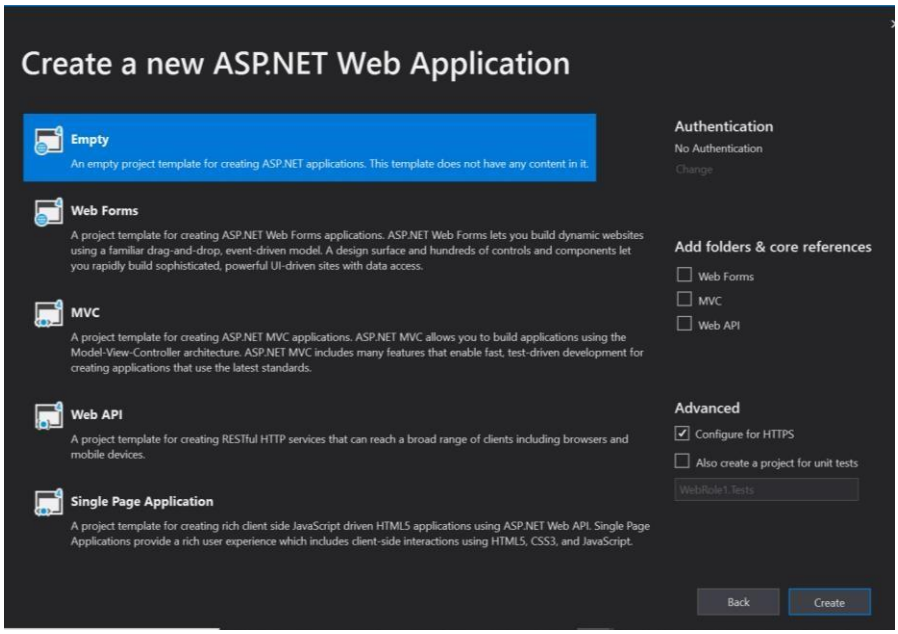




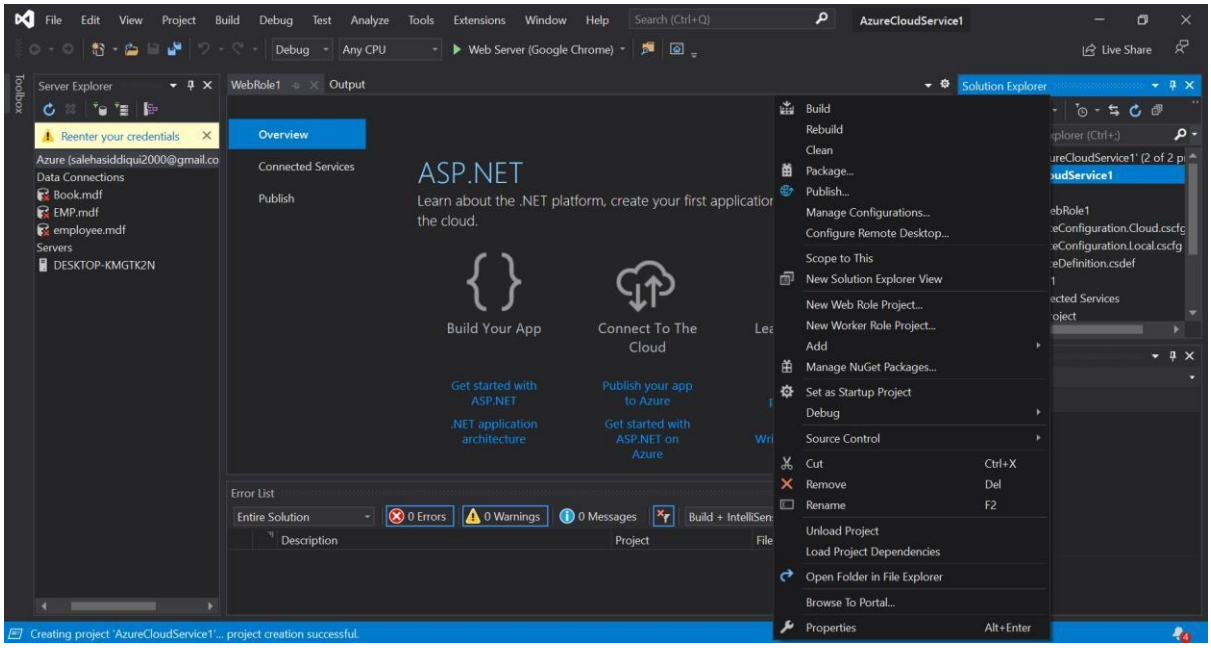
Step 03: Add Asp.net Visual c# WebRole 1

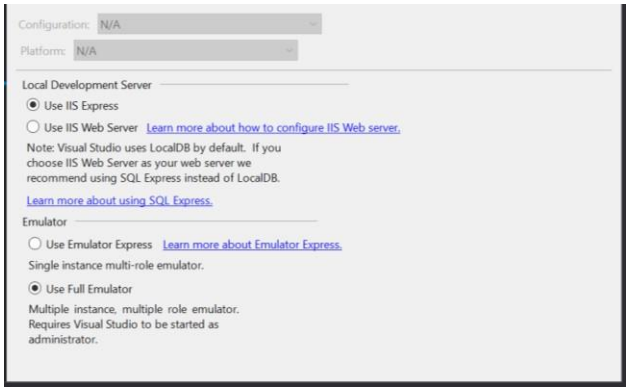


Step 04: Select Empty and then create

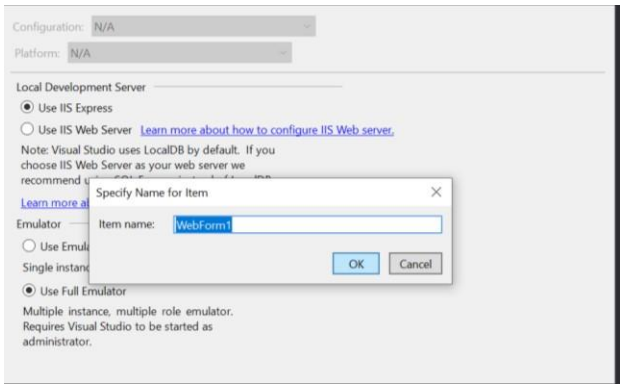


Step 05: Right Click on Application and go to properties then select web and select Full emulator and save it.

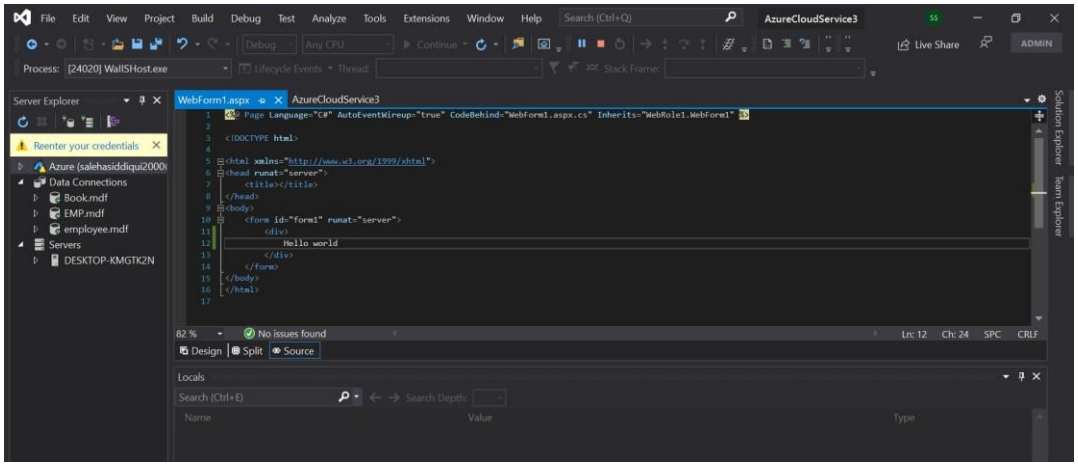




Step 06: Select WebRole1 > Select Add > Select New Item> Webform



Step 07: Add text hello world in div tag and Run the application.

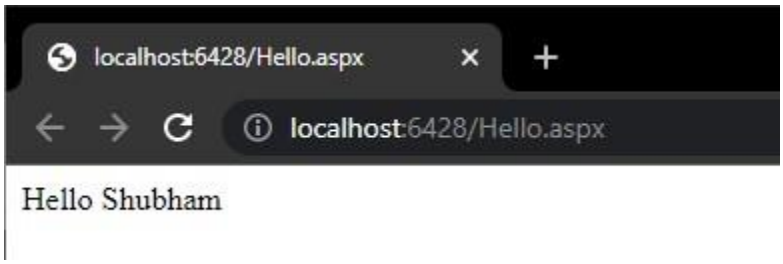


Program:

Hello.aspx

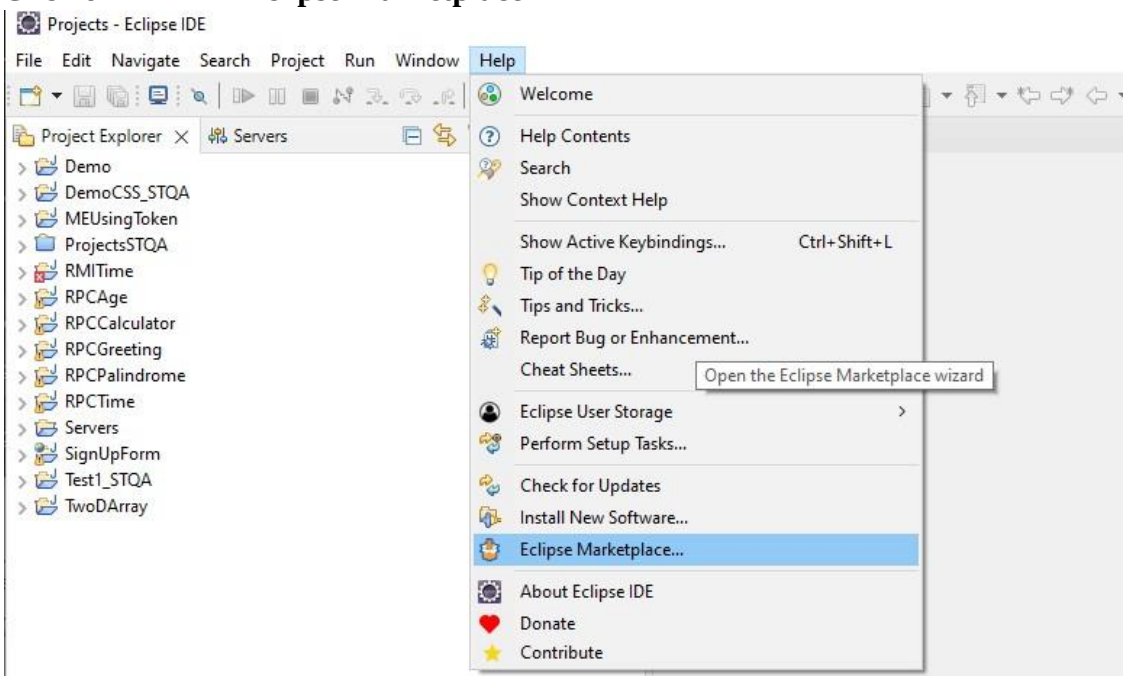
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Hello.aspx.cs"
Inherits="WebRole1.Hello" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
Hello Shubham
</div>     </form>
</body>
</html>
```

Output:

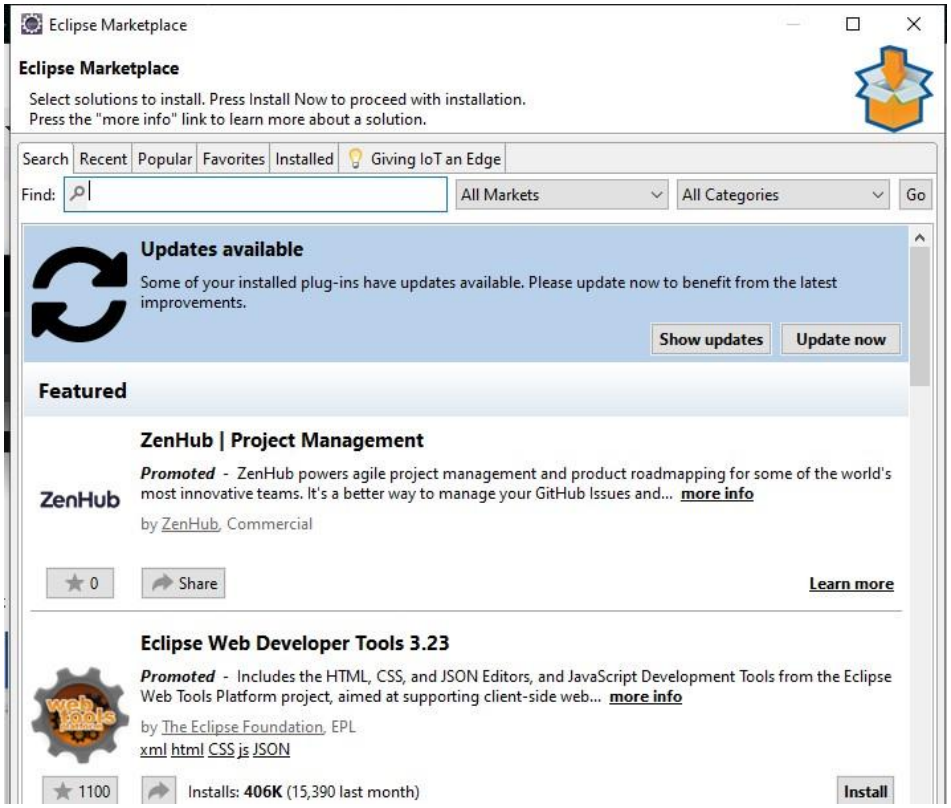


B) To develop applications using Google App Engine by using Eclipse IDE

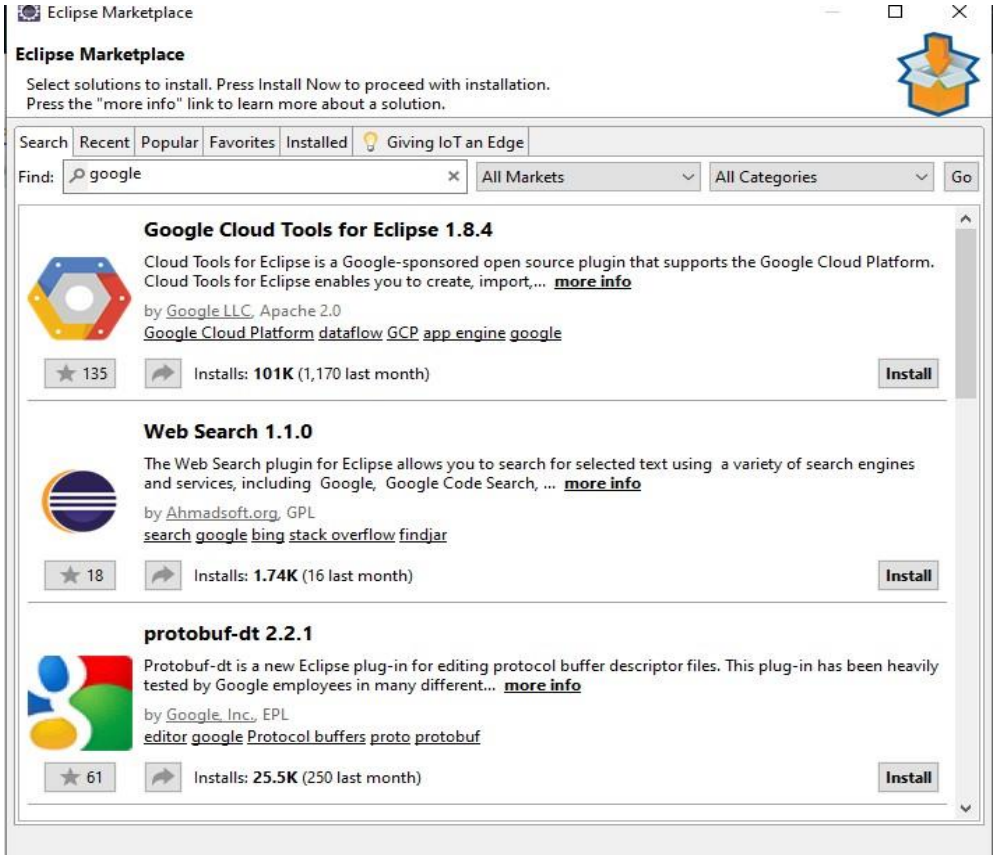
Open Workspace
Click on HELP→Eclipse Marketplace



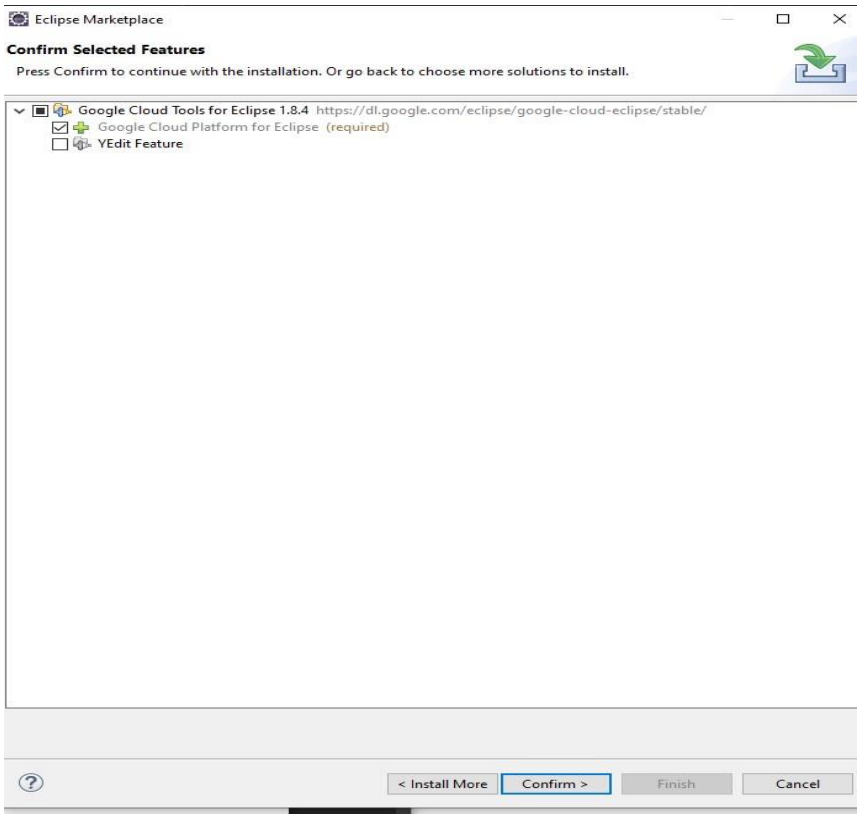
Once the Eclipse Marketplace window appear in search textbox write google click on enter.



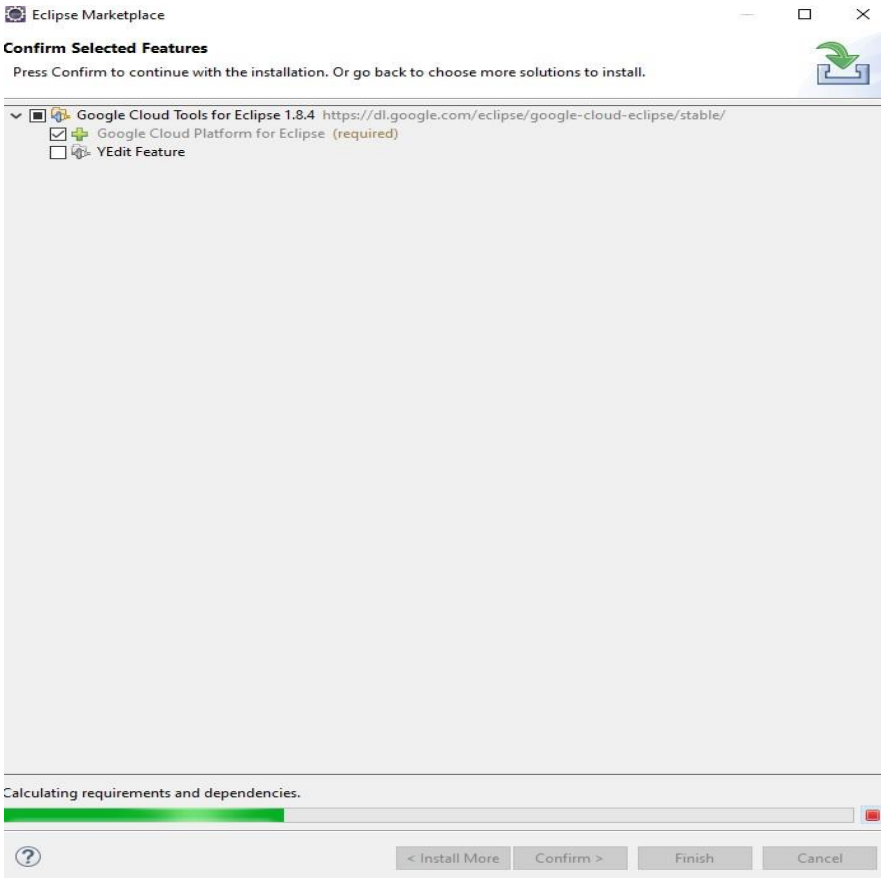
Now search Google Cloud



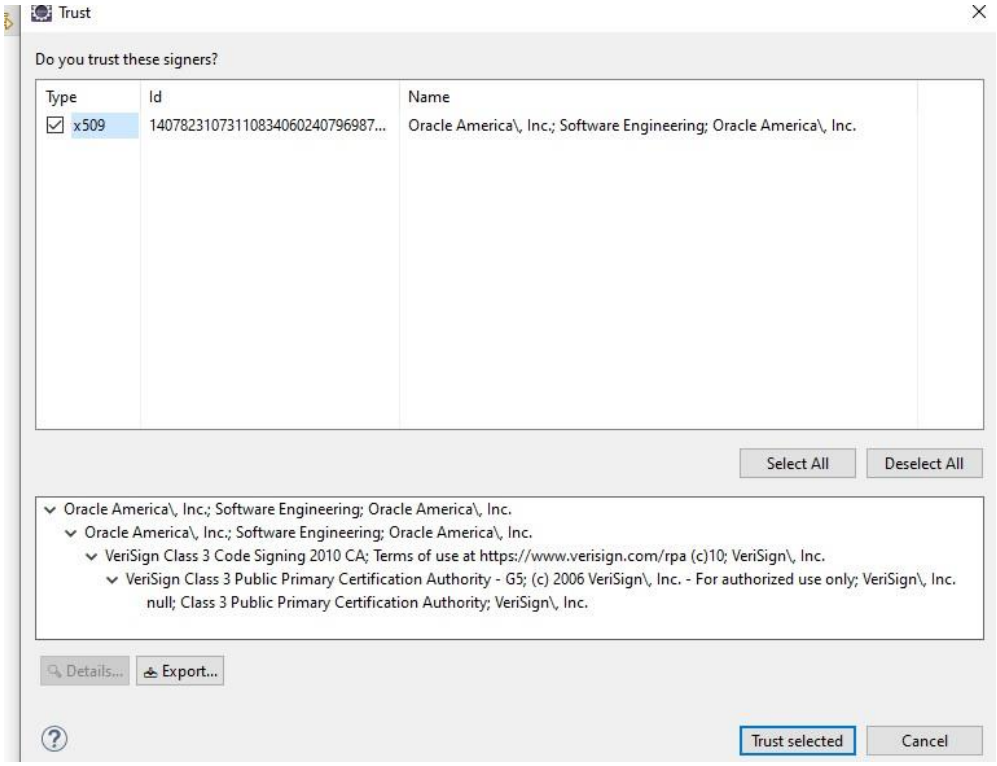
Click on Install



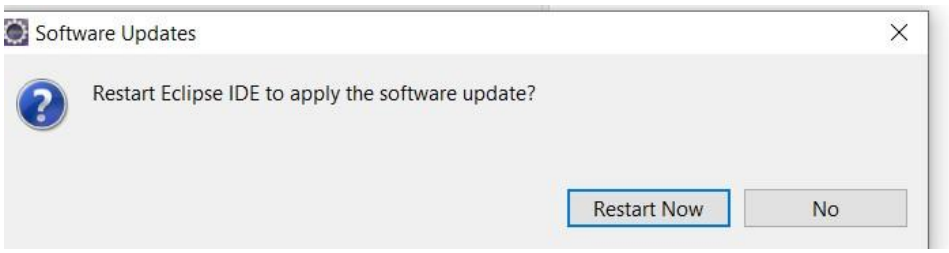
Click on Confirm



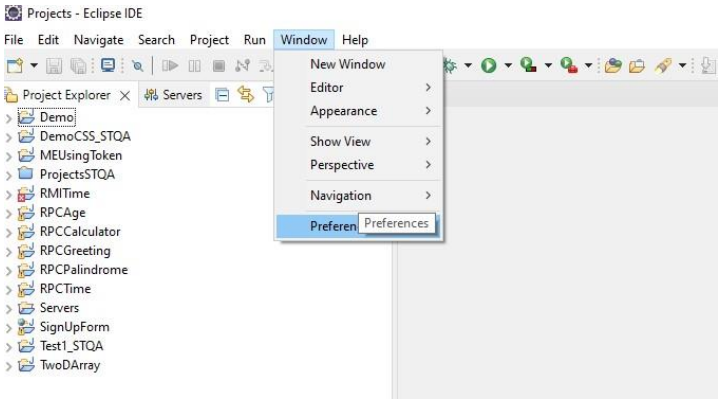
Now select x509 option and click on Trust selected.



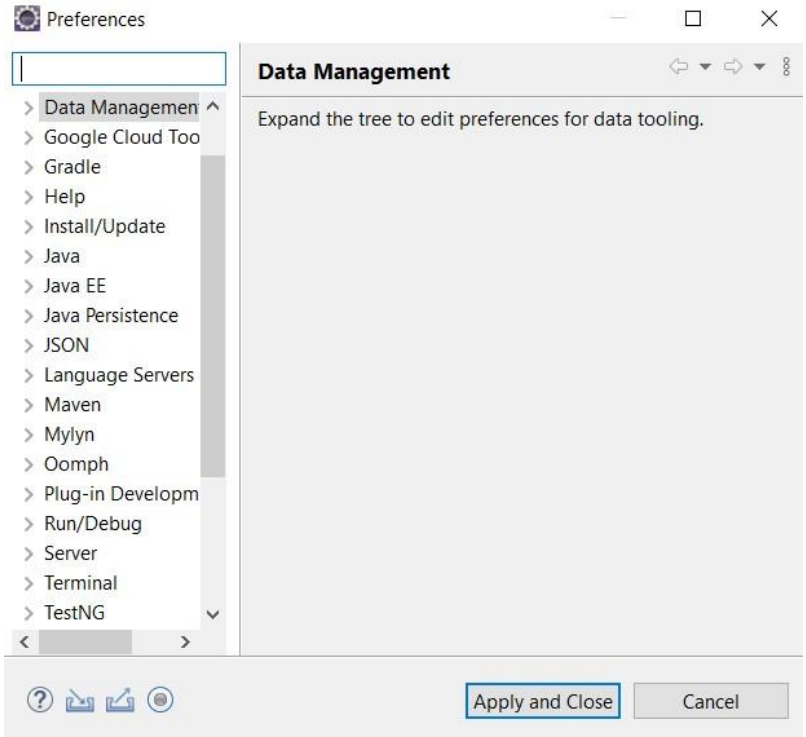
Restart Now



Now select workspace in windows tab select preferences

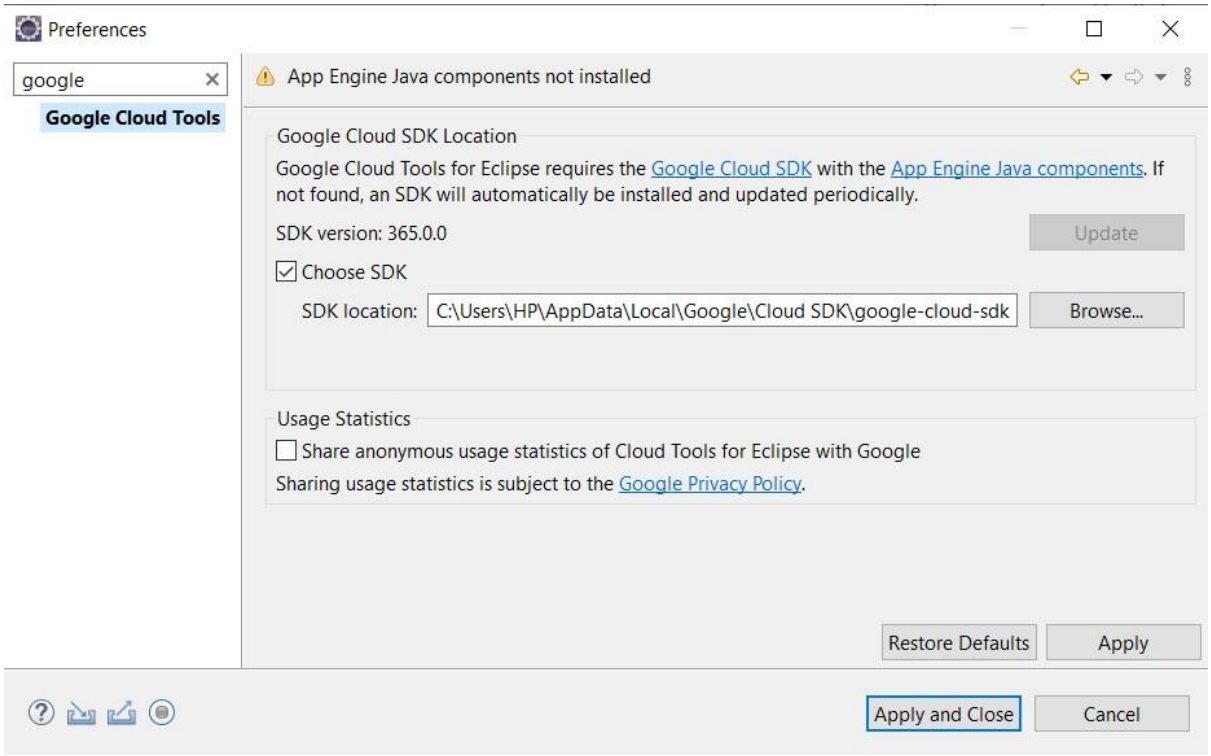


Below Preferences window will appear SEARCH FOR GOOGLE



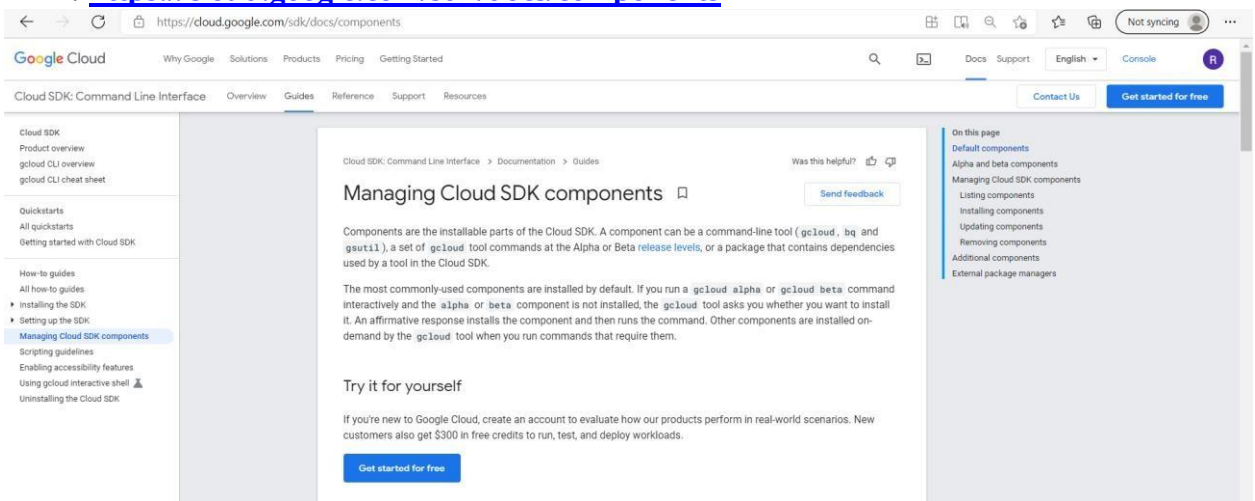
SEARCH FOR GOOGLE

Click for APP ENGINE JAVA COMPONENTS and click on APPLY AND CLOSE

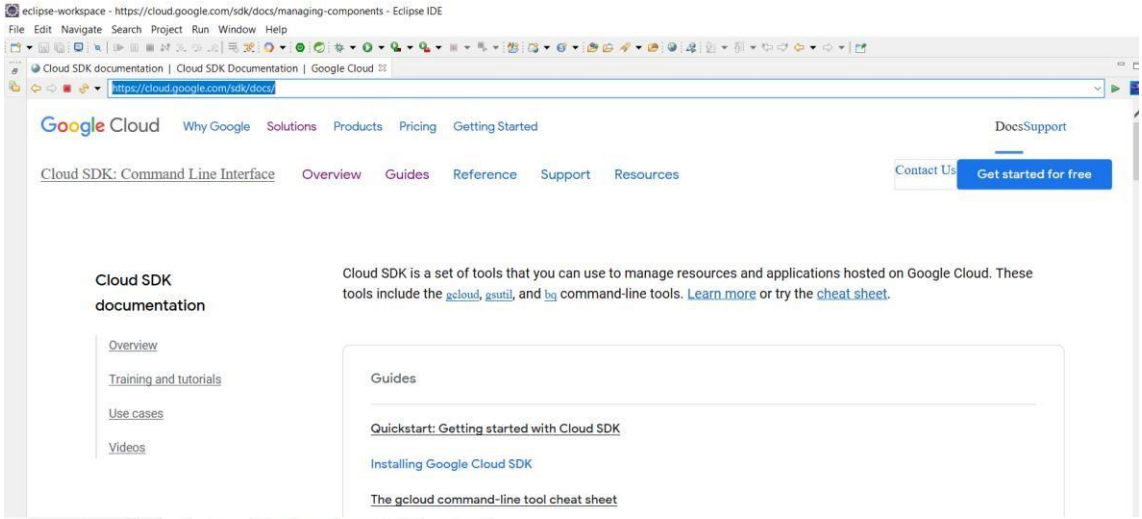


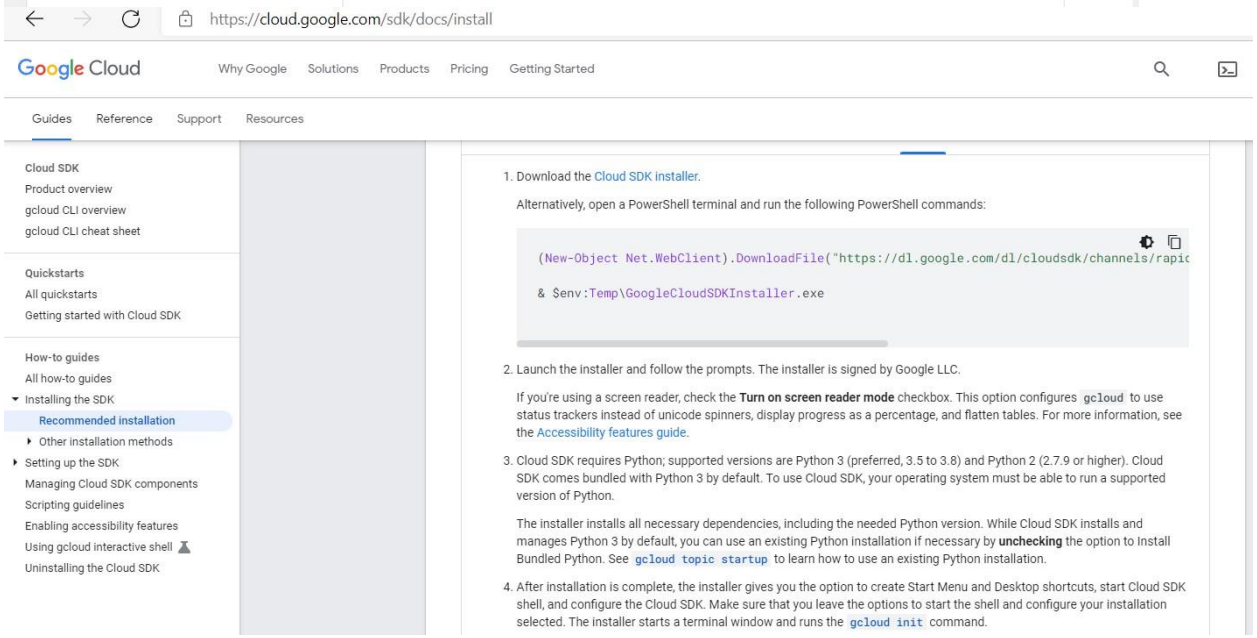
Click on below click

Link: https://cloud.google.com/sdk/docs/components



Click on installing Google Cloud SDK

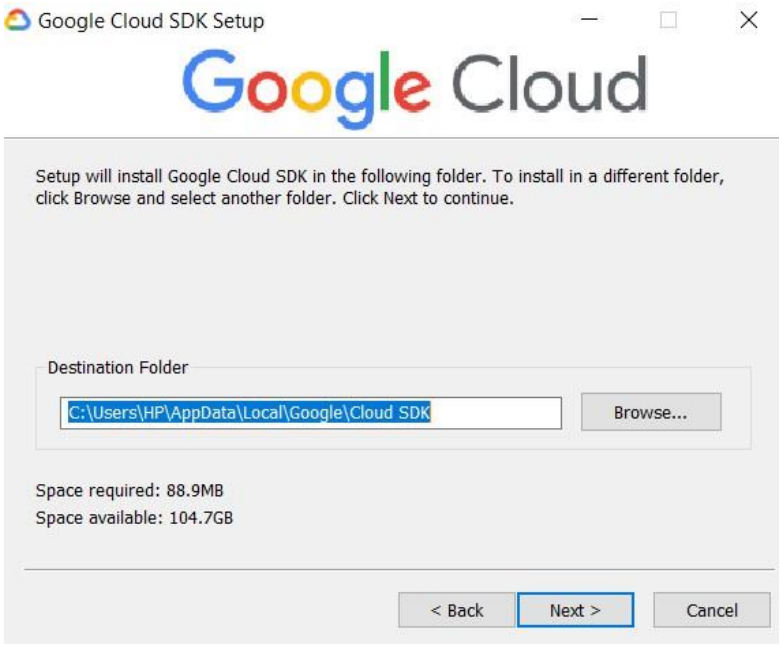


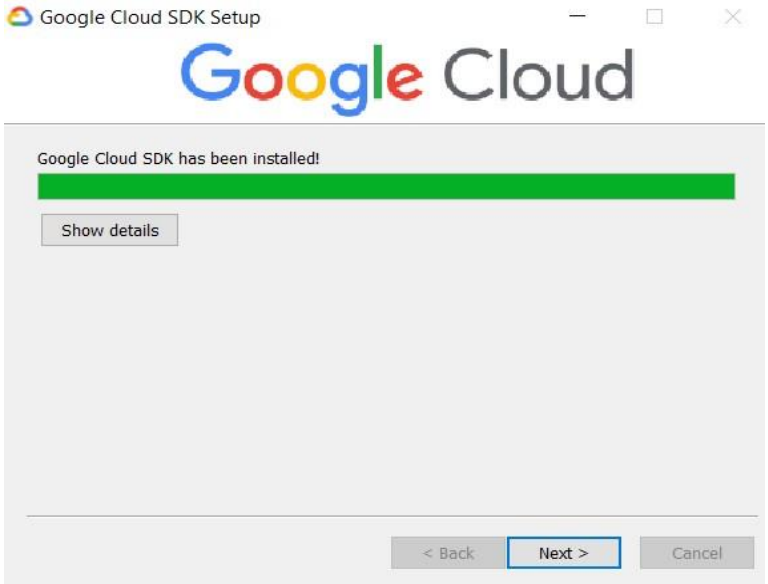


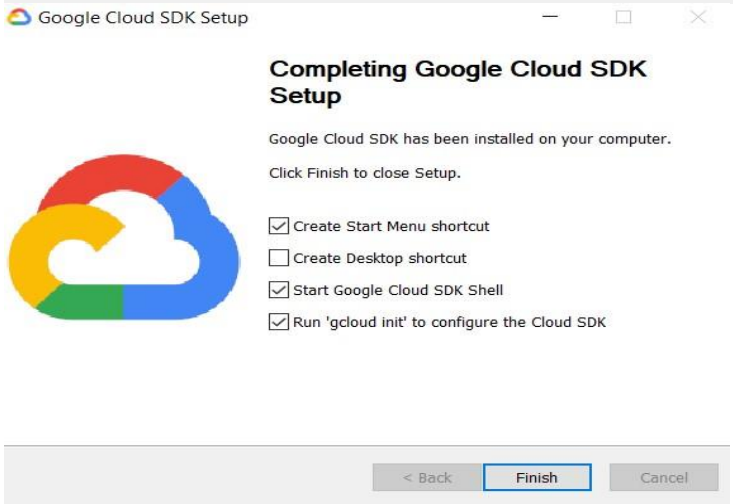
Below screen will appear


Google Cloud SDK Setup
Google Cloud
Press Page Down to see the rest of the agreement.
The Google Cloud SDK and its source code are licensed under Apache License v. 2.0 (the "License"), unless otherwise specified by an alternate license file.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Note that if you use the Cloud SDK with any Google Cloud Platform products, your use is additionally going to be governed by the license agreement or terms of service, as applicable, of the underlying Google Cloud Platform
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Google Cloud SDK.
< Back
I Agree
Cancel


Google Cloud SDK Setup
Google Cloud
Please select install type:
Single User (HP)
All users
< Back
Next >
Cancel


Google Cloud SDK Setup
Google Cloud
Setup will install Google Cloud SDK in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Users\HP\AppData\Local\Google\Cloud SDK
Browse...
Space required: 88.9MB
Space available: 104.7GB
< Back
Next >
Cancel


Google Cloud SDK Setup
Google Cloud
Google Cloud SDK has been installed!
Show details
< Back
Next >
Cancel


Google Cloud SDK Setup
Completing Google Cloud SDK Setup
Google Cloud SDK has been installed on your computer.
Click Finish to close Setup.
Create Start Menu shortcut
Create Desktop shortcut
Start Google Cloud SDK Shell
Run 'gcloud init' to configure the Cloud SDK
< Back
Finish
Cancel

Below gcloud init Command Prompt will appear

C:\Windows\SYSTEM32\cmd.exe - gcloud init

```
---
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)?  y

Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redire
ct_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fw
ww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=6LiYiQ7sM0z6vssnGpDQL
u7zDBU4IY&access_type=offline&code_challenge=U0KC2ewlUXo0BdzHbbdcv7VosOfZW_Vkl7MuXFFWC8Q&code_challenge_method=S256

You are logged in as: [ravitapatil919@gmail.com].

Pick cloud project to use:
 [1] languege-traslator-app
 [2] translator-app-e6568
 [3] Create a new project
Please enter numeric choice or text value (must exactly match list item):
```

Choose the option Log in with a new account
After login in
It will show list of project in your account if you receive the message this account has no projects.

C:\Windows\SYSTEM32\cmd.exe - gcloud init

```
---
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)?  y

Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redire
ct_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2
F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fw
ww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=6LiYiQ7sM0z6vssnGpDQL
u7zDBU4IY&access_type=offline&code_challenge=U0KC2ewlUXo0BdzHbbdcv7VosOfZW_Vkl7MuXFFWC8Q&code_challenge_method=S256

You are logged in as: [ravitapatil919@gmail.com].

Pick cloud project to use:
 [1] languege-traslator-app
 [2] translator-app-e6568
 [3] Create a new project
Please enter numeric choice or text value (must exactly match list item):
```

Then create new projects

C:\Windows\SYSTEM32\cmd.exe - gcloud init

```
You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
 [1] ravitapatil919@gmail.com
 [2] Log in with a new account
Please enter your numeric choice:  1

You are logged in as: [ravitapatil919@gmail.com].

Pick cloud project to use:
 [1] current-location-332813
 [2] languege-traslator-app
 [3] translator-app-e6568
 [4] Create a new project
Please enter numeric choice or text value (must exactly match list item):  4

Enter a Project ID. Note that a Project ID CANNOT be changed later.
Project IDs must be 6-30 characters (lowercase ASCII, digits, or
hyphens) in length and start with a lowercase letter.
```

Open google cloud SDK Shell

Google Cloud SDK Shell

```
C:\Users\HP\AppData\Local\Google\Cloud SDK>gcloud components update
Beginning update. This process may take several minutes.

All components are up to date.


To take a quick anonymous survey, run:
  $ gcloud survey


C:\Users\HP\AppData\Local\Google\Cloud SDK>
```

Google Cloud SDK Shell

```
C:\Users\HP\AppData\Local\Google\Cloud SDK>gcloud config set disable_usage_reporting false
Updated property [core/disable_usage_reporting].

C:\Users\HP\AppData\Local\Google\Cloud SDK>gcloud components install app-engine-go

Restarting command:
  $ gcloud components install app-engine-go


C:\Users\HP\AppData\Local\Google\Cloud SDK>
```

```
cmd.exe /c ""C:\Users\HP\AppData\Local\Temp\tmpl4ns6u5q\python\python.exe" "-S" "C:\Users\HP\AppData\Local\Google\Cloud ...

Your current Cloud SDK version is: 366.0.0
Installing components from version: 366.0.0

+-----------------------------------------------------+
|        These components will be installed.          |
+------------------------------+---------+-----------+
|             Name             | Version |   Size    |
+------------------------------+---------+-----------+
| App Engine Go Extensions     |  1.9.71 |  4.8 MiB  |
| Cloud Datastore Emulator     |   2.1.0 | 18.4 MiB  |
| gRPC Python library          |  1.20.0 |  1.5 MiB  |
| gcloud app Python Extensions |  1.9.98 |  7.8 MiB  |
+------------------------------+---------+-----------+

For the latest full release notes, please visit:
  https://cloud.google.com/sdk/release_notes

Do you want to continue (Y/n)?
```

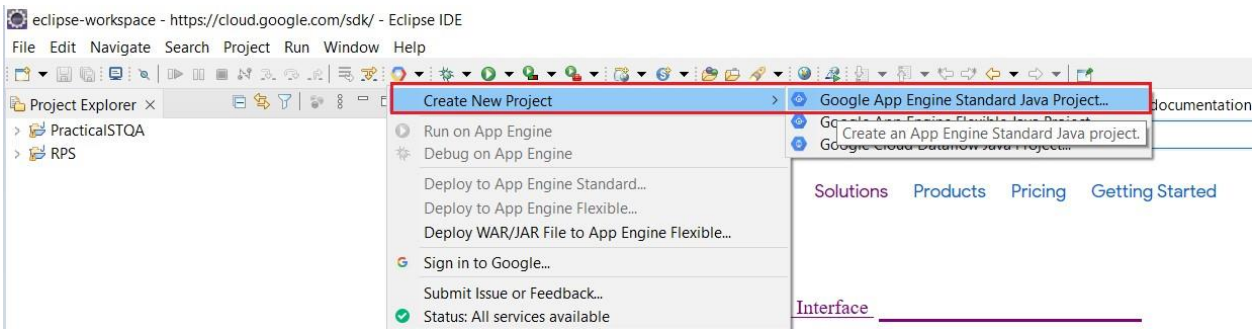
Do you want to continue? type y

```
cmd.exe /c ""C:\Users\HP\AppData\Local\Temp\tmpl4ns6u5q\python\python.exe" "-S" "C:\Users\HP\AppData\Local\Google\Cloud ...
| App Engine Go Extensions     |  1.9.71 |  4.8 MiB  |
| Cloud Datastore Emulator     |   2.1.0 | 18.4 MiB  |
| gRPC Python library          |  1.20.0 |  1.5 MiB  |
| gcloud app Python Extensions |  1.9.98 |  7.8 MiB  |
+------------------------------+---------+-----------+

For the latest full release notes, please visit:
  https://cloud.google.com/sdk/release_notes

Do you want to continue (Y/n)?  y

#=============================================================#
#= Creating update staging area                              =#
#=============================================================#
#= Installing: App Engine Go Extensions                      =#
#=============================================================#
#= Installing: App Engine Go Extensions                      =#
#=============================================================#
#= Installing: Cloud Datastore Emulator                      =#
#=============================================================#
#= Installing: gRPC Python library                           =#
#=============================================================#
#= Installing: gRPC Python library                           =#
#=============================================================#
#= Installing: gcloud app Python Extensions                  =#
#=============================================================#
#= Creating backup and activating new installation           =#
#=============================================================#

Performing post processing steps...-
```

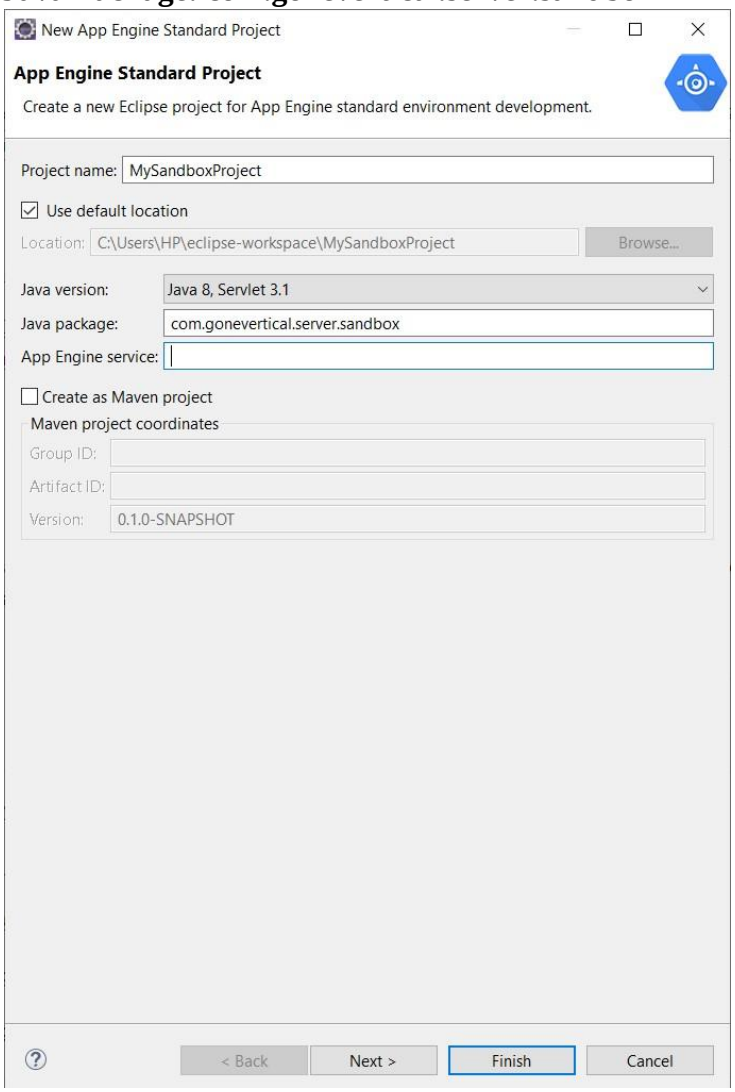
Close the command prompt

Name: Sayas Sonawane **Div: B** **Roll No. 51**

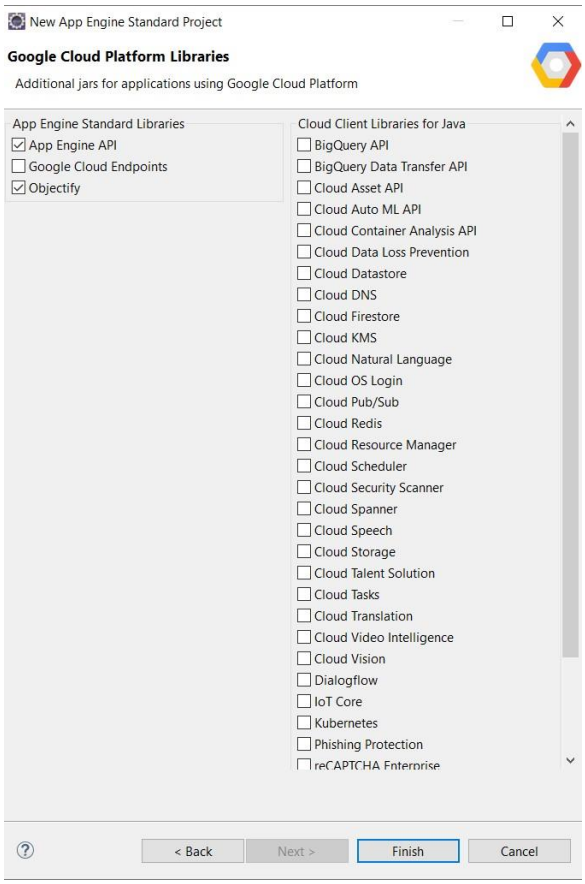
In Eclipse workspace click on dropdown →Click →Create New Project→Google App Engine Standard Java Project.

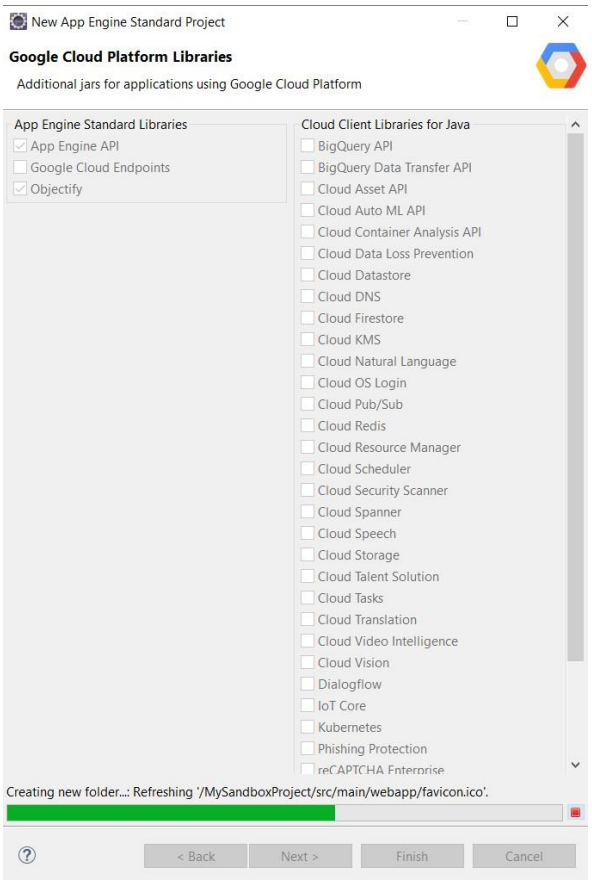


Below screen will appear New App Engine Standard Project
Provide the Project Name & Java Package as mentioned below
Project Name: MySandboxProject
Java Package: com.gonevertical.server.sandbox

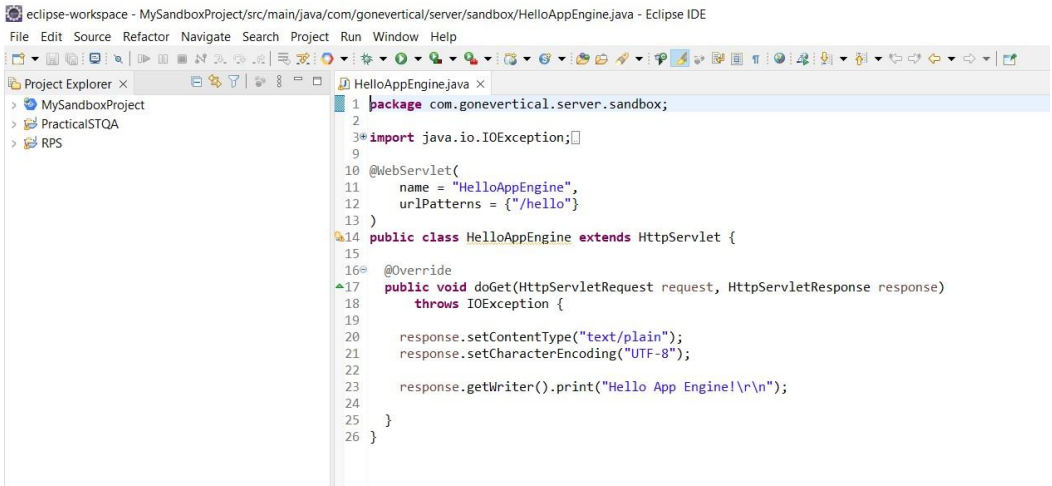


Below screen will appear
FROM App Engine Standard Libraries
Select 1.App Engine API
Select 2.Objectify
3.Click on Finish

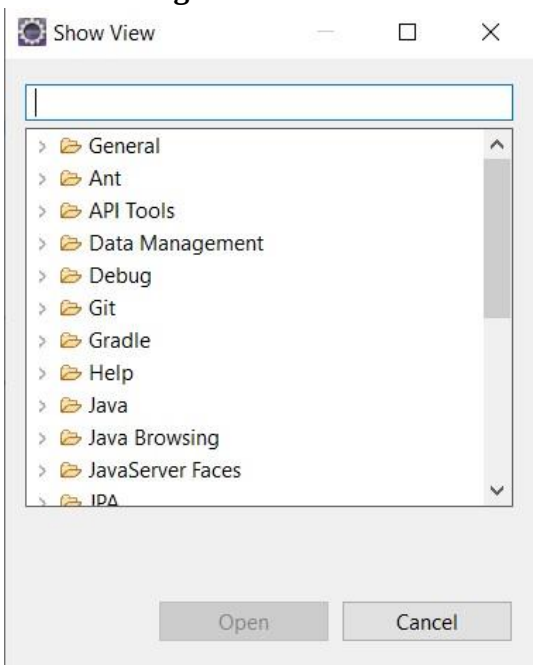




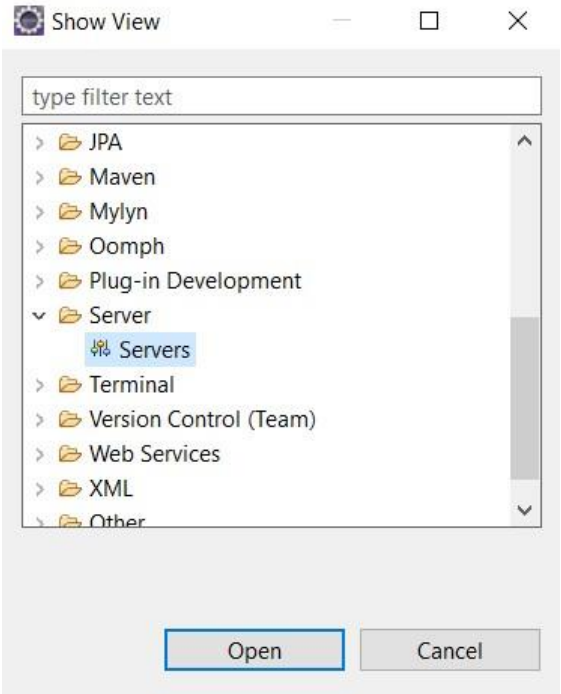
Below Screen will Appear HelloAppEngine.Java



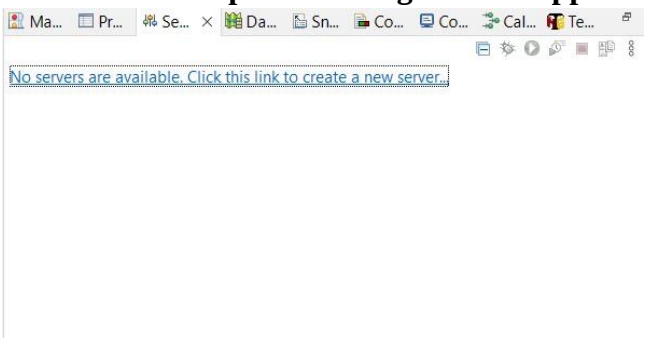
Go to Window→Show View→Other
After clicking on other below Screen of Show View will appear



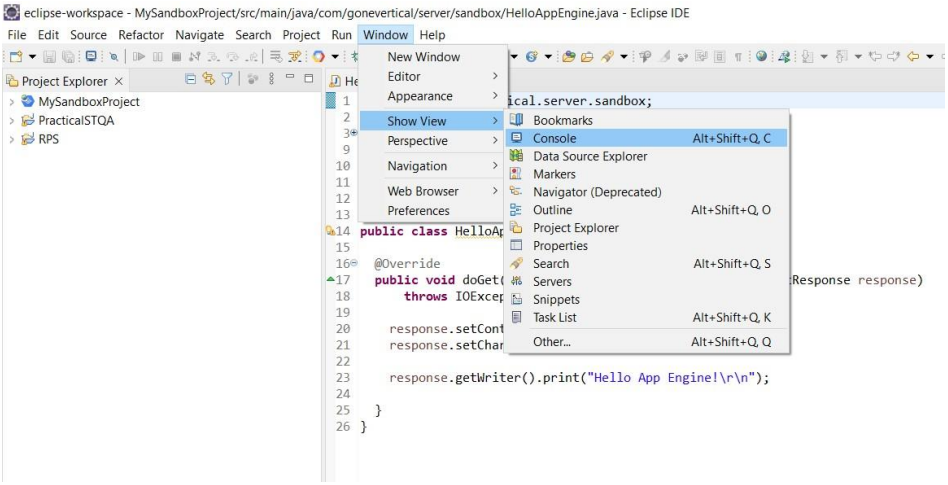
1.Click on Server→Servers
2. Click on Open



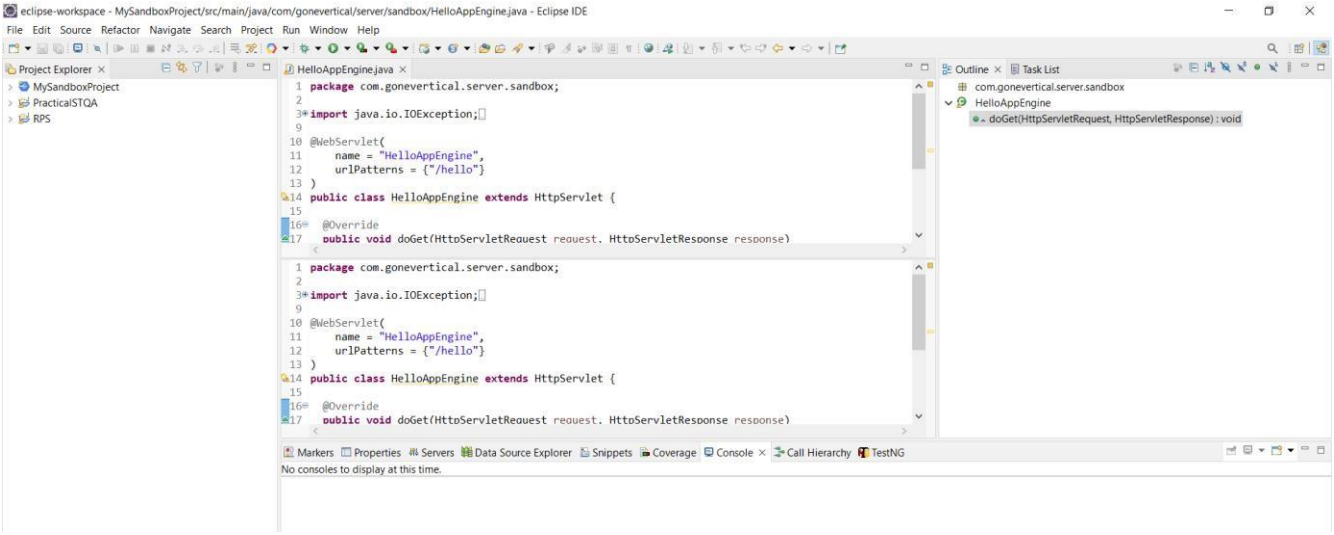
After Click on Open following window appeared.



Click on Window→Editor→Console

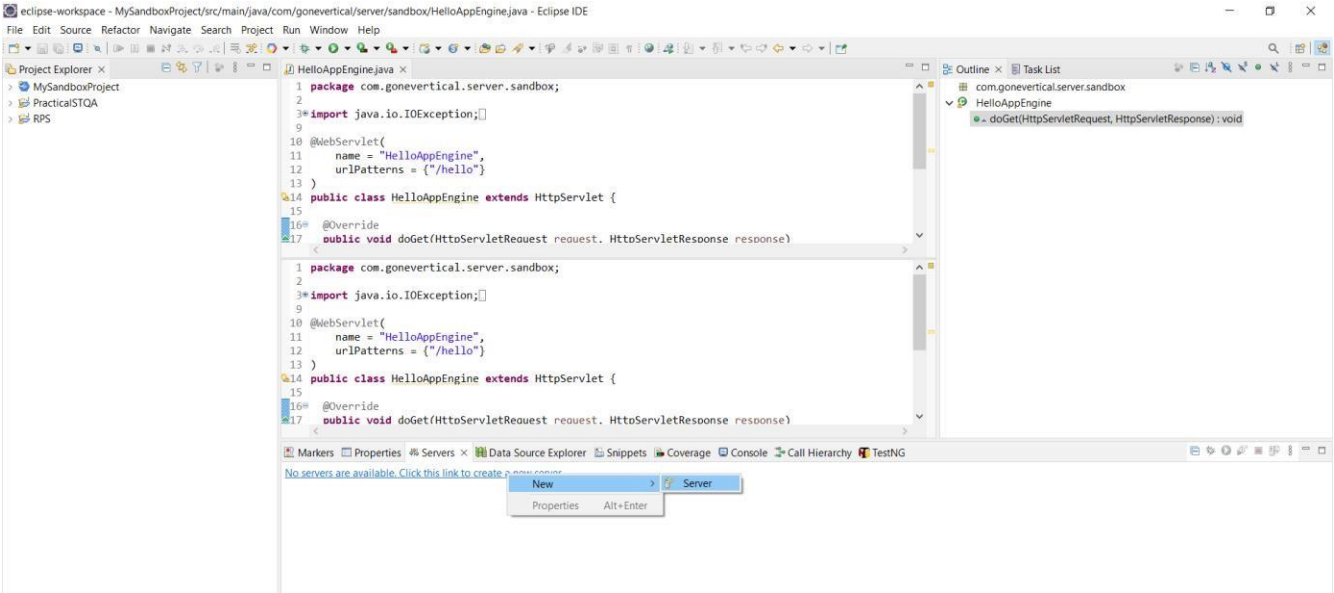


Once the Console screen appear →Click on Servers



Name: Sayas Sonawane **Div: B** **Roll No. 51**

Right click on server →New→Server

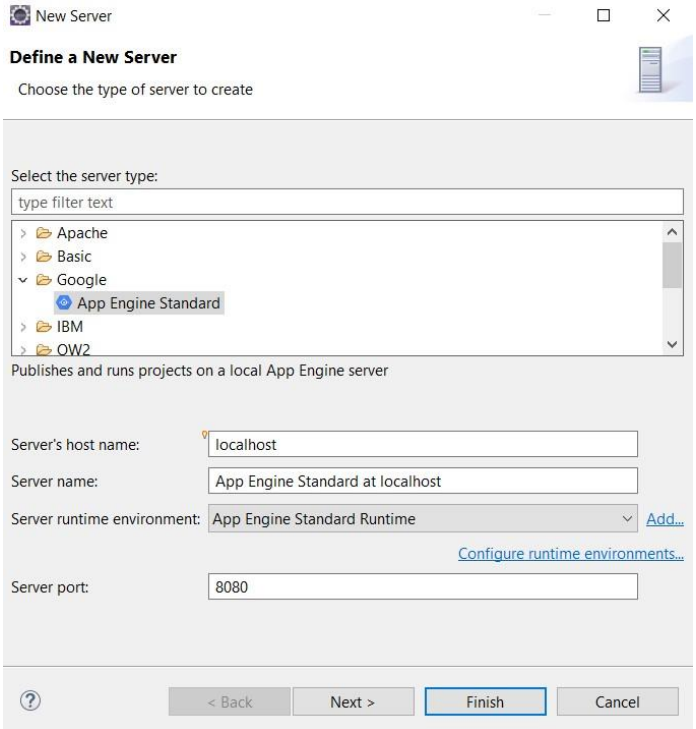


To define a new server will below screen appear

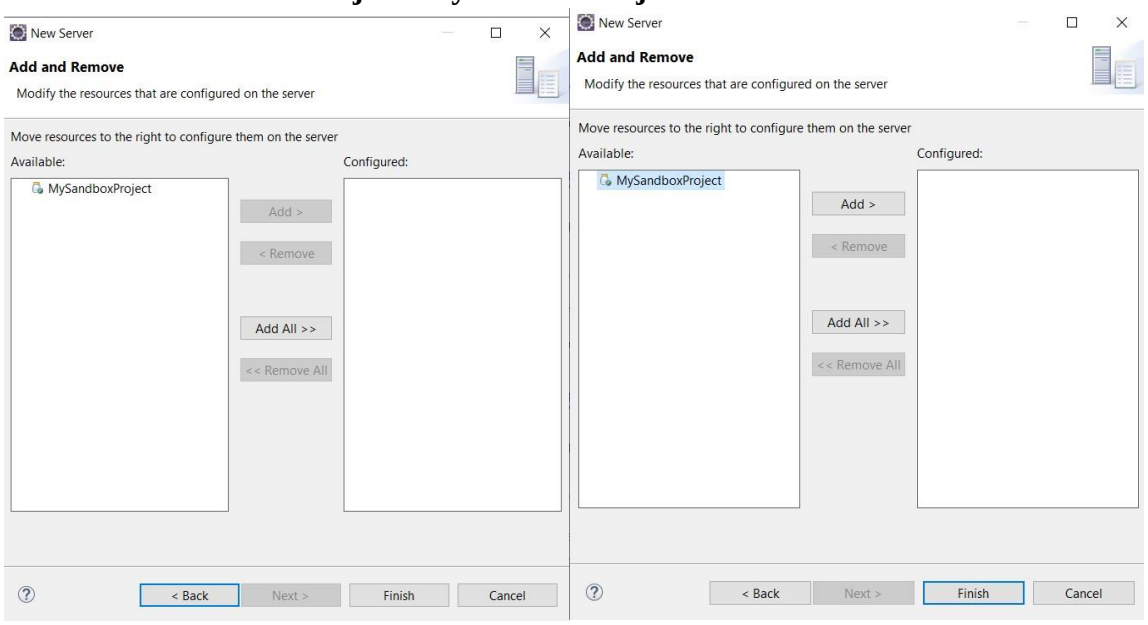
1.Select App Engine Standard

2.Click on Next

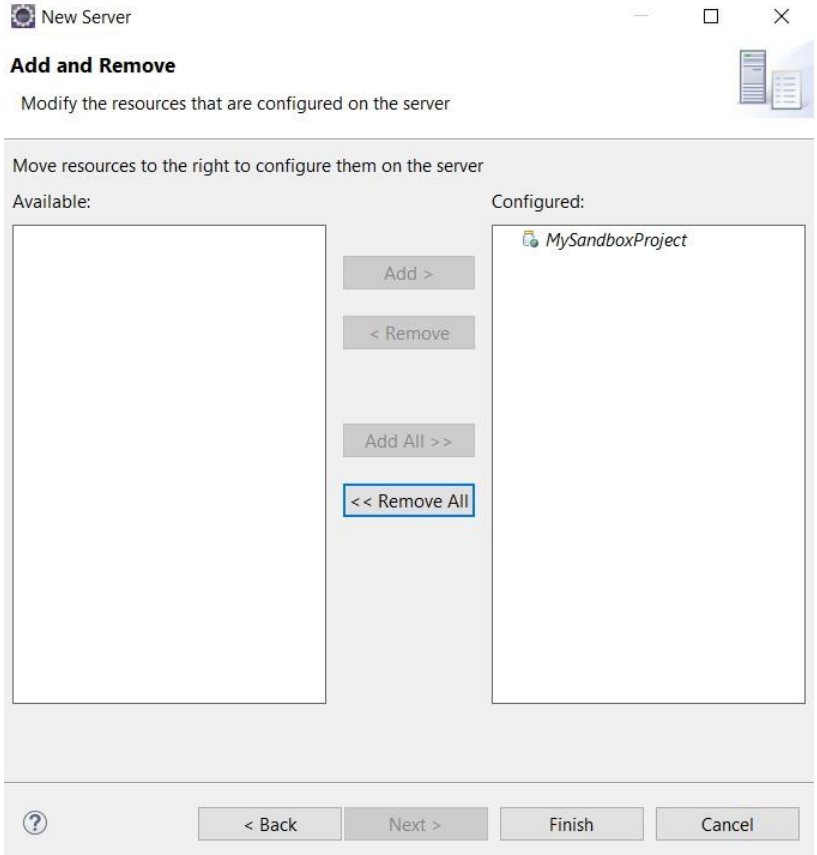
3.Click on Finish



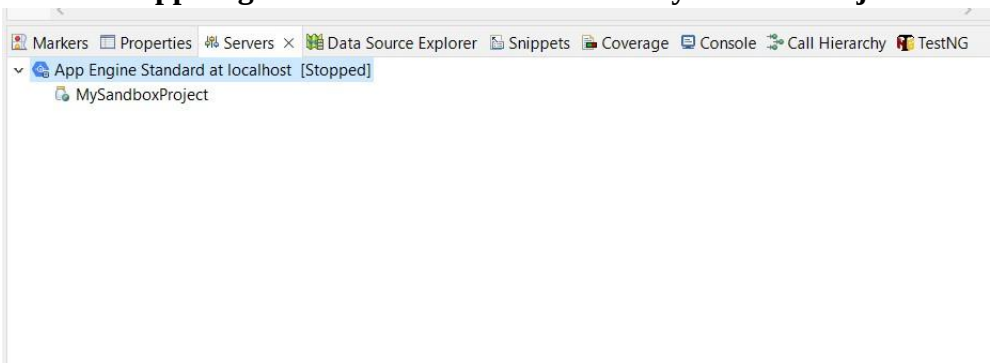
Select from Available Project: MySandboxProject and then click on Add Button



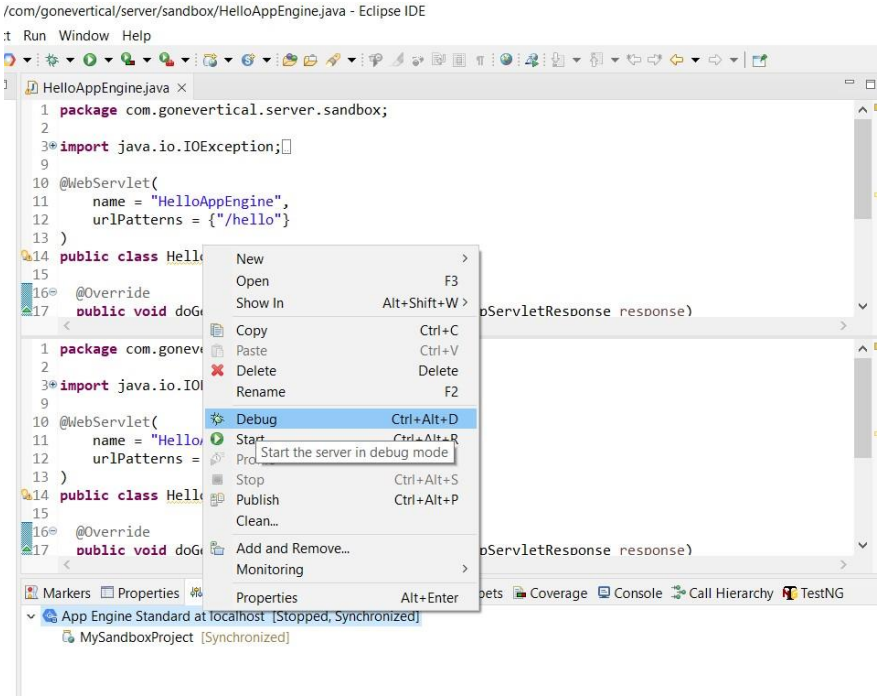
After adding MySandboxProject in Configured then Click on Finish Button



Click On App Engine Standard At localhost →MySandboxProject



Right click on App Engine Standard At localhost→Click on Debug



Final Output Screen will appear