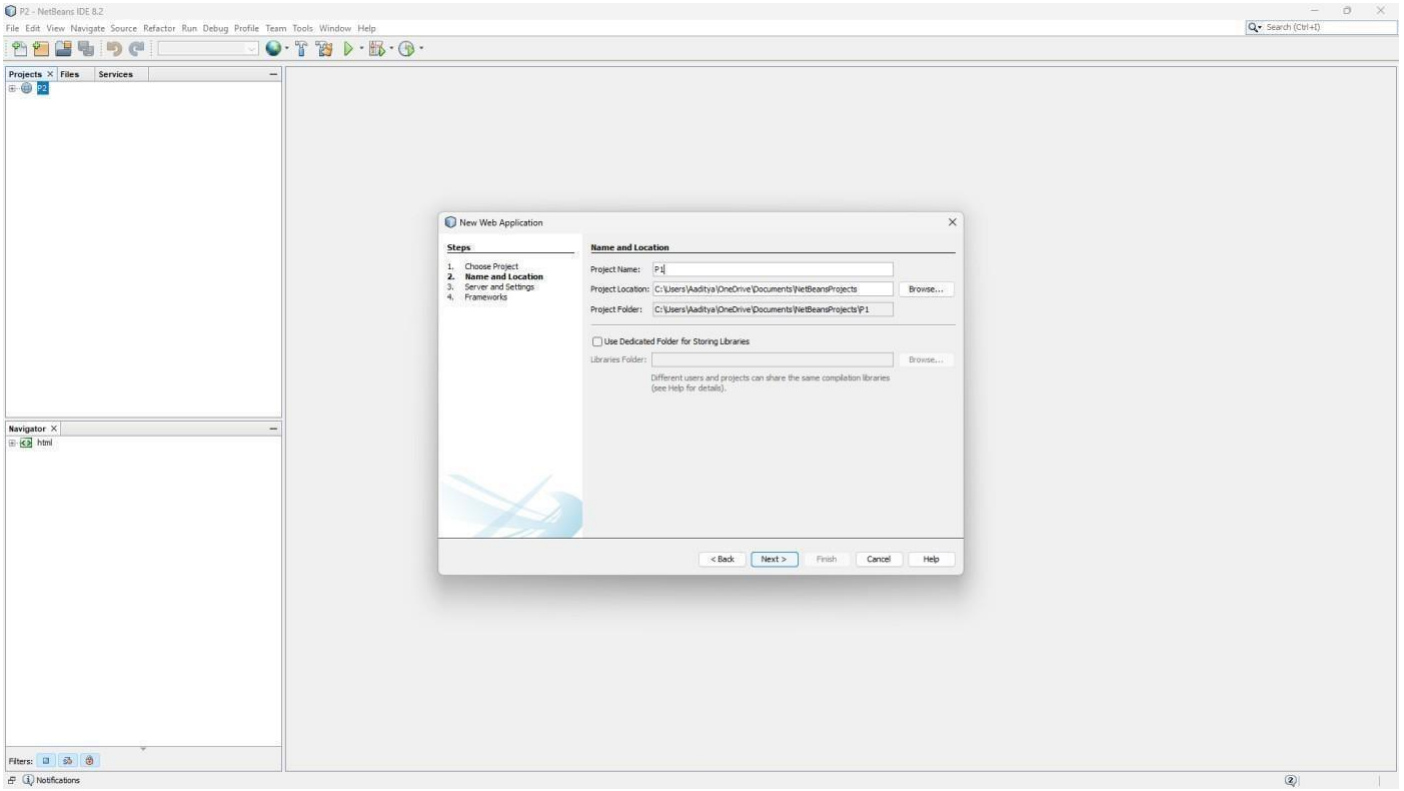
SR No.	INDEX	Date	Sign
1.	Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA and .NET		
2.	Create a Simple SOAP service.		
3.	Create a Simple REST Service.		
4.	Develop application to consume Google's search / Google's Map RESTful Web service.		
5.	Installation and Configuration of virtualization using KVM.		
6.	Develop application to download image/video from server or upload image/video to server using MTOM techniques		
7.	Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage		
8.	Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS),		
9.	Using AWS Flow Framework develop application that includes a simple workflow. Workflow calls an activity to print hello world to the console. It must define the basic usage of AWS Flow Framework, including defining contracts, implementation of activities and workflow coordination logic and worker programs to host them		
10.	Implementation of Openstack with user and private network creation.		

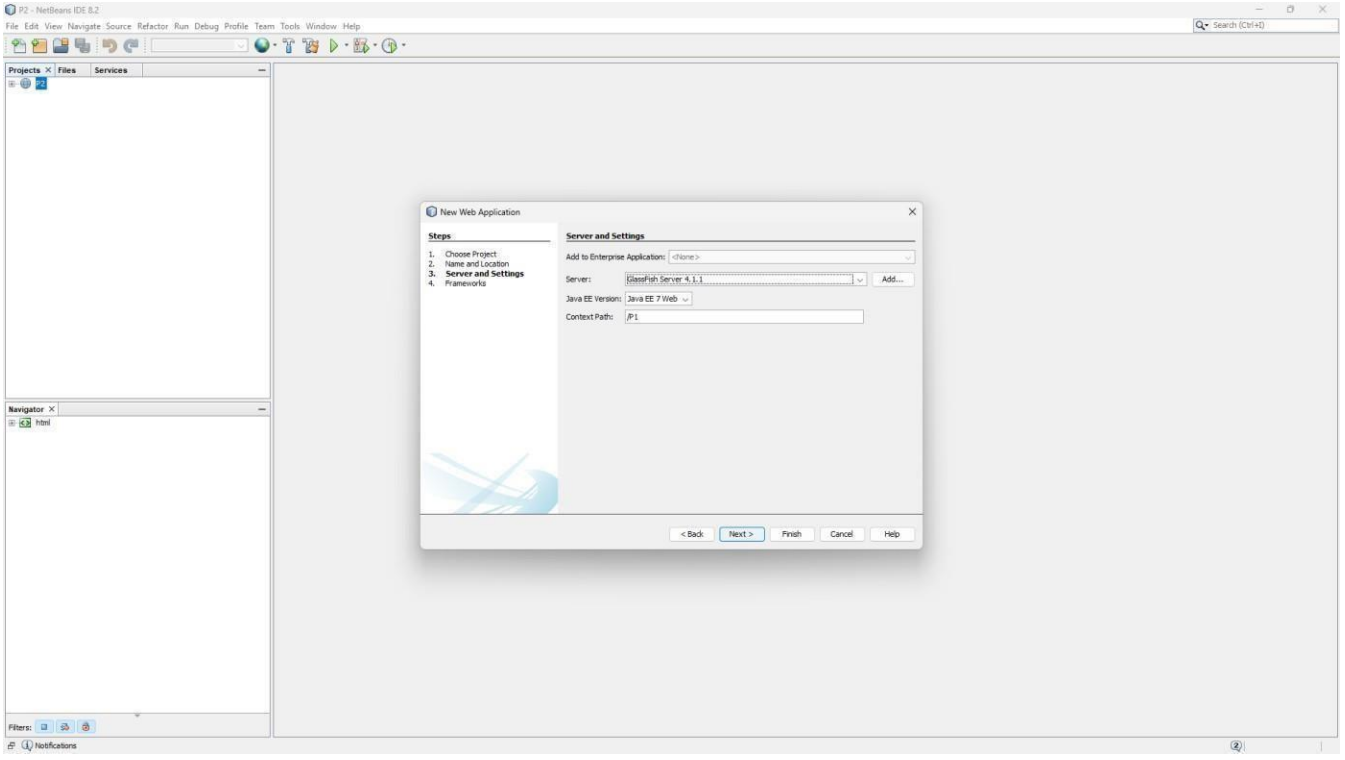
Practical :1

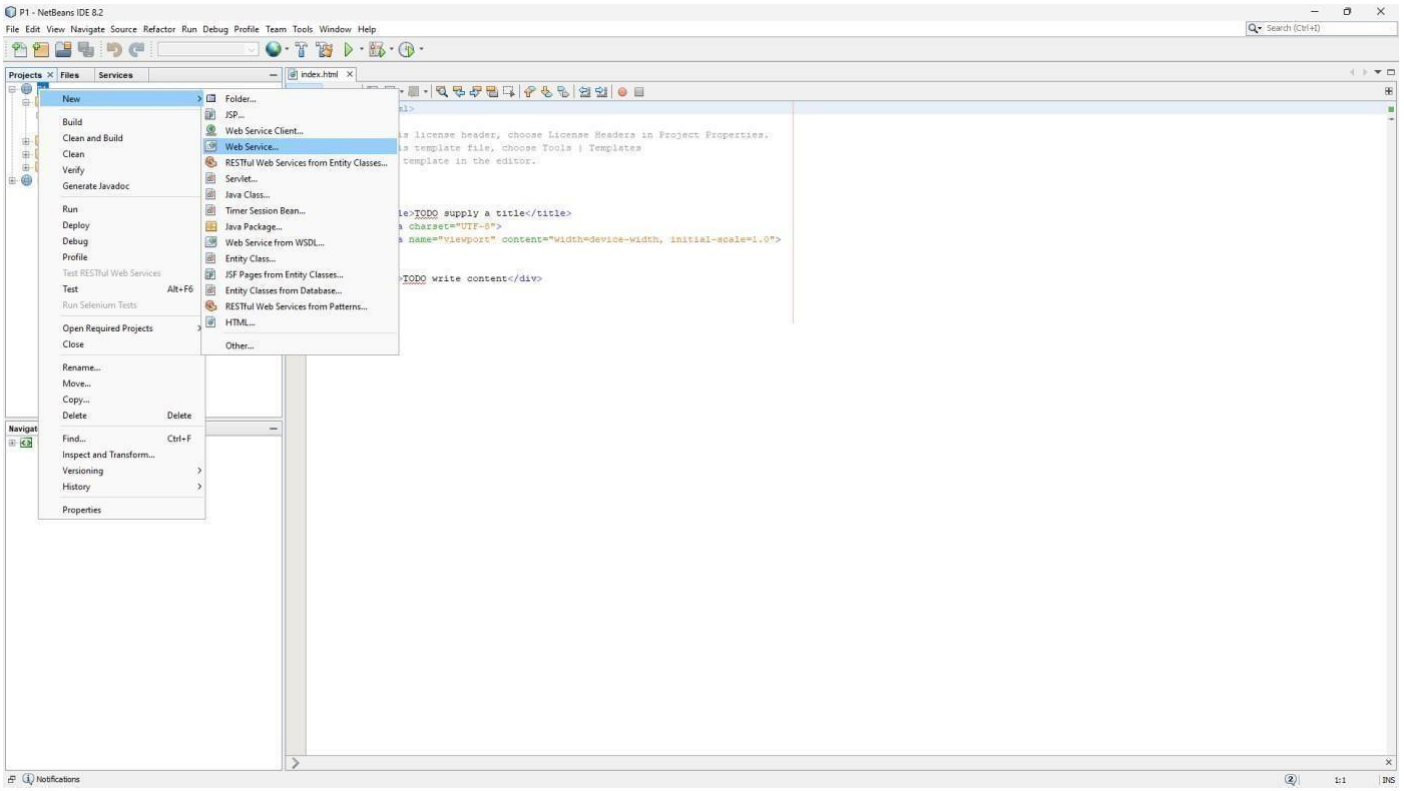
Aim: Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVAand .NET

Step-1

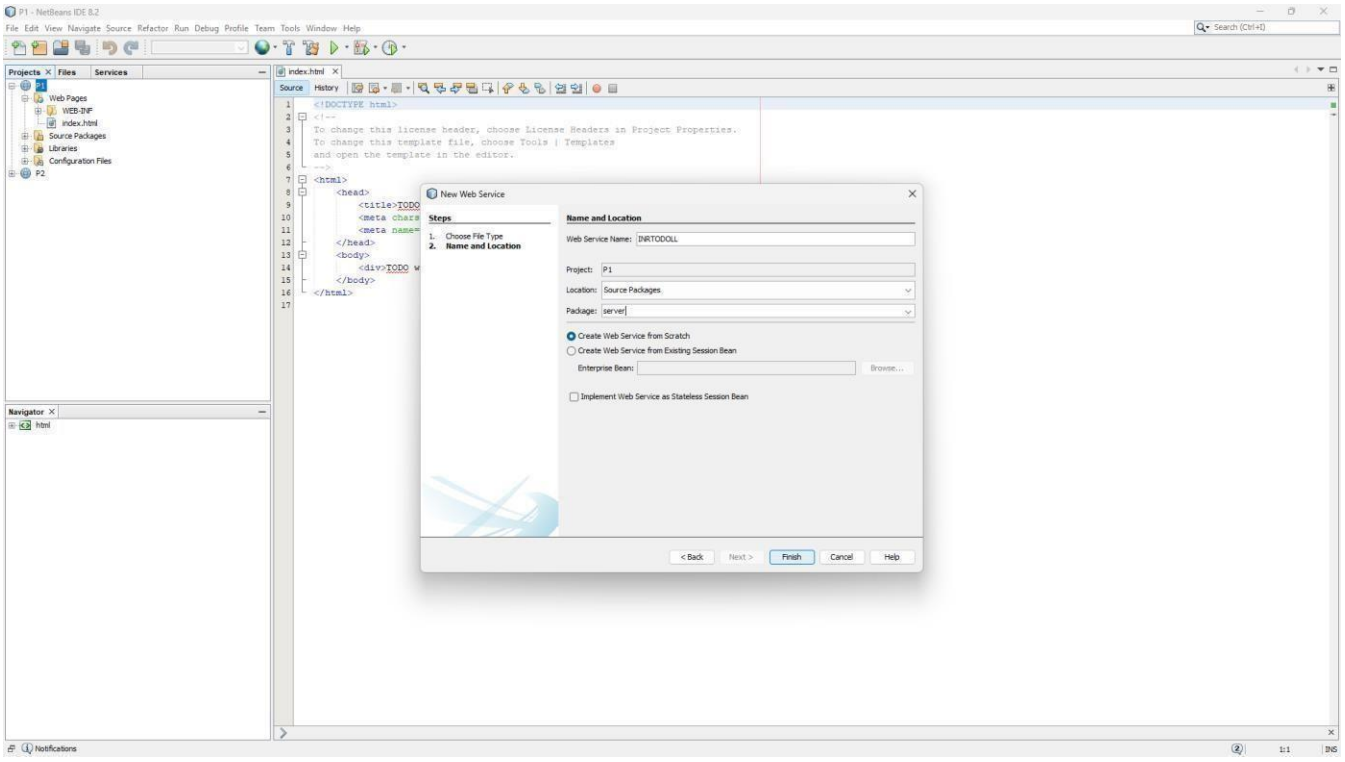


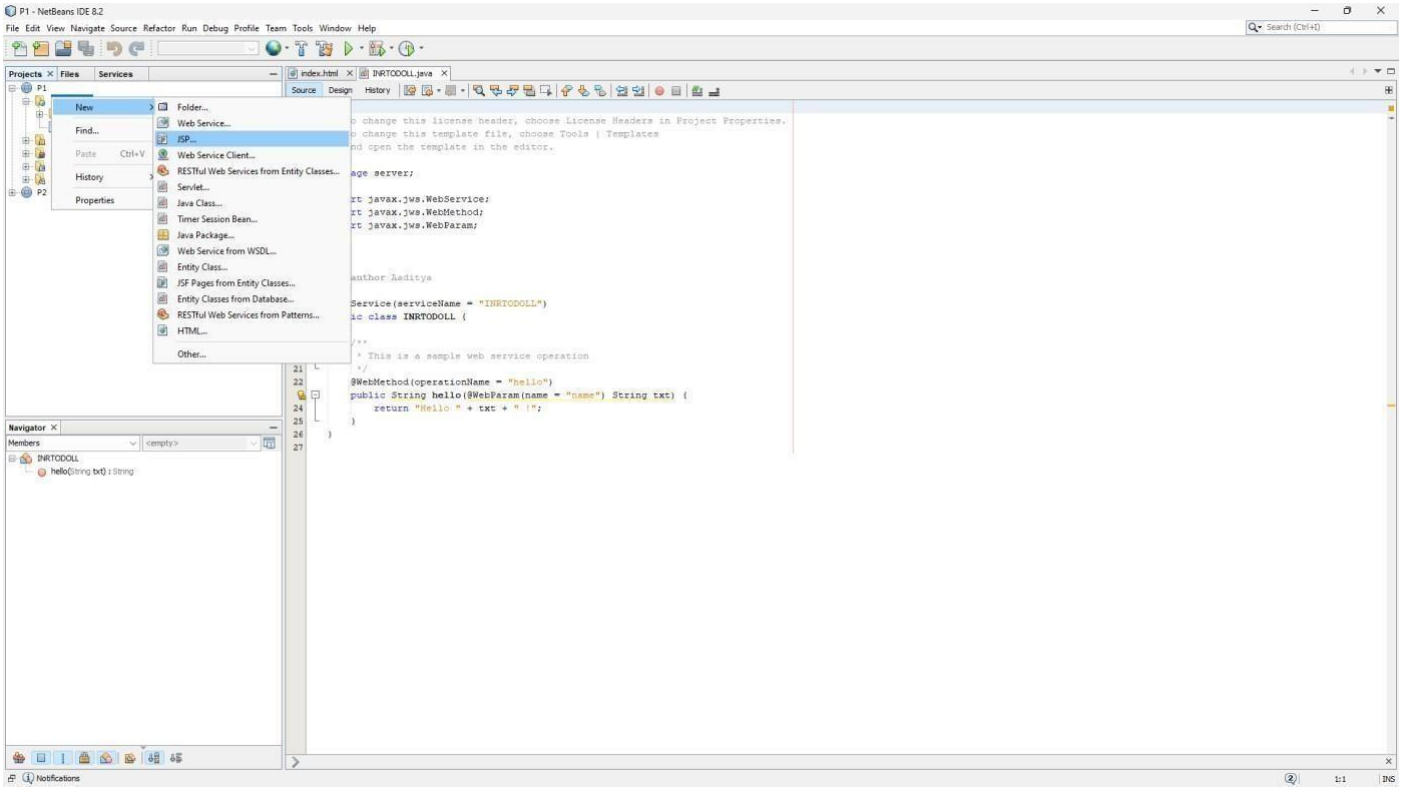
Step-2



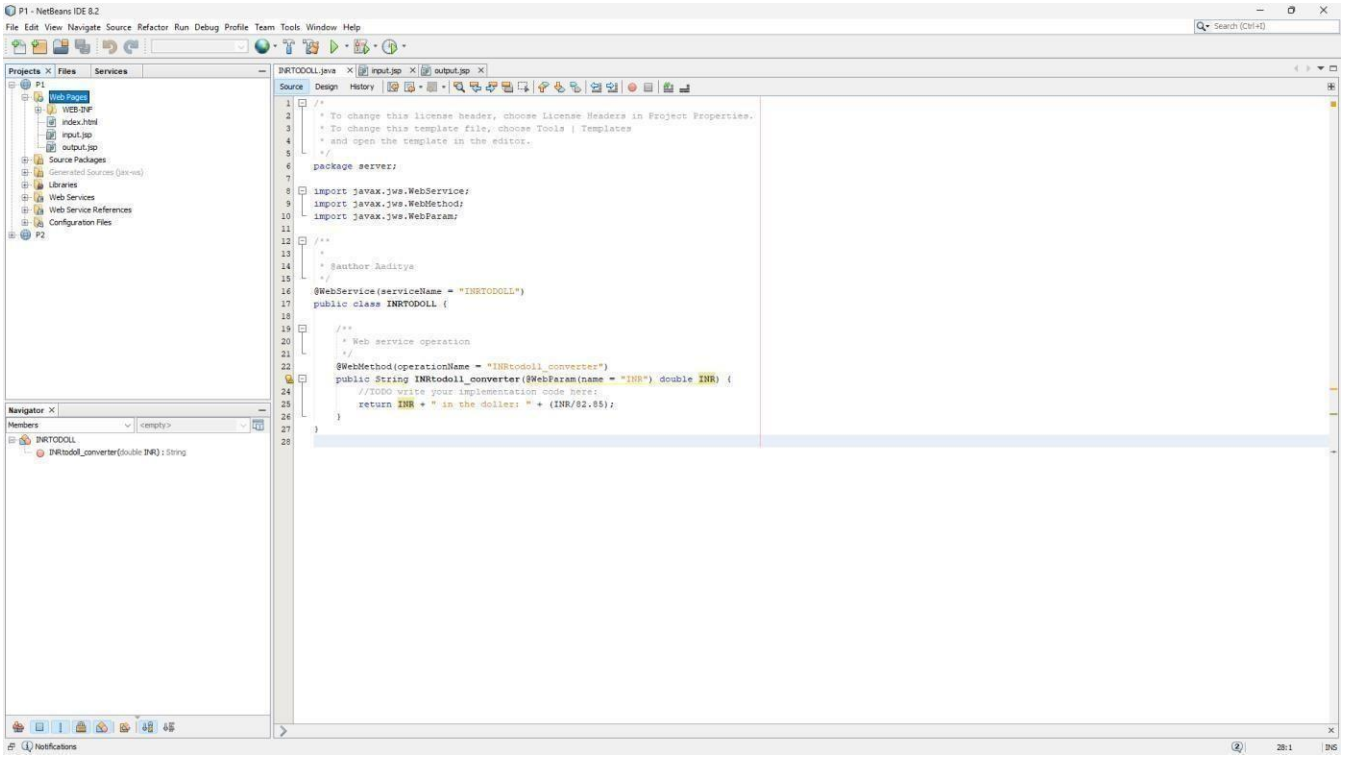

P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
index.html
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
JSP...
Web Service Client...
Web Service...
RESTful Web Services from Entity Classes...
Servlet...
Java Class...
Timer Session Bean...
Java Package...
Web Service from WSDL...
Entity Class...
JSF Pages from Entity Classes...
Entity Classes from Database...
RESTful Web Services from Patterns...
HTML...
Other...
Notifications
1:1
INS

Step-4

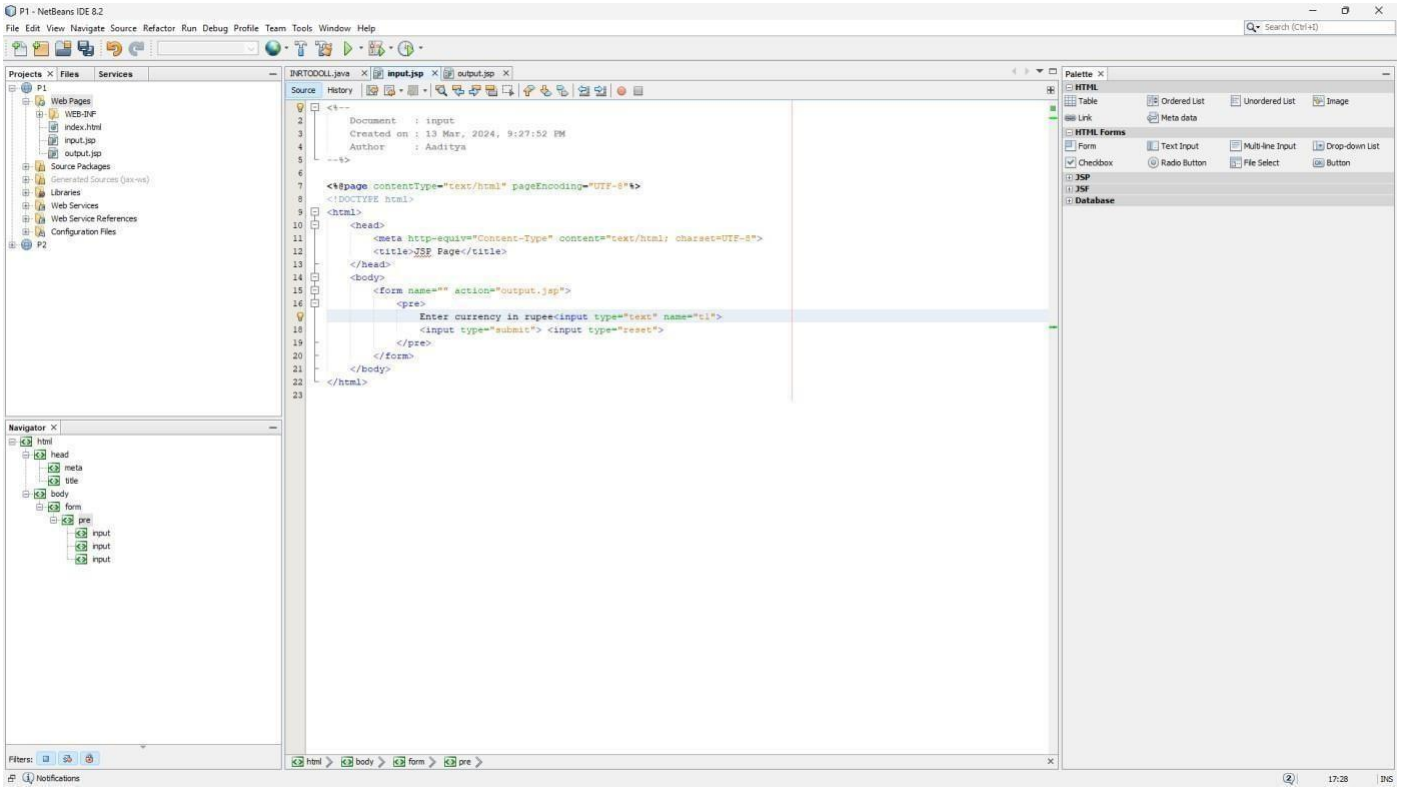

P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
Web Pages
WEB-INF
index.html
Source Packages
Libraries
Configuration Files
P2
Navigator
html
index.html
Source
History
New Web Service
Steps
1. Choose File Type
2. Name and Location
Name and Location
Web Service Name: INRTODOLL
Project: P1
Location: Source Packages
Package: server
Create Web Service from Scratch
Create Web Service from Existing Session Bean
Enterprise Bean:
Browse...
Implement Web Service as Stateless Session Bean
< Back
Next >
Finish
Cancel
Help
Notifications


P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
index.html
INRTODOLL.java
Source
Design
History
New
Find...
Paste Ctrl+V
History
Properties
Folder...
Web Service...
JSP...
Web Service Client...
RESTful Web Services from Entity Classes...
Servlet...
Java Class...
Timer Session Bean...
Java Package...
Web Service from WSDL...
Entity Class...
JSF Pages from Entity Classes...
Entity Classes from Database...
RESTful Web Services from Patterns...
HTML...
Other...
Navigator
Members
<empty>
INRTODOLL
hello(String txt) : String
Notifications

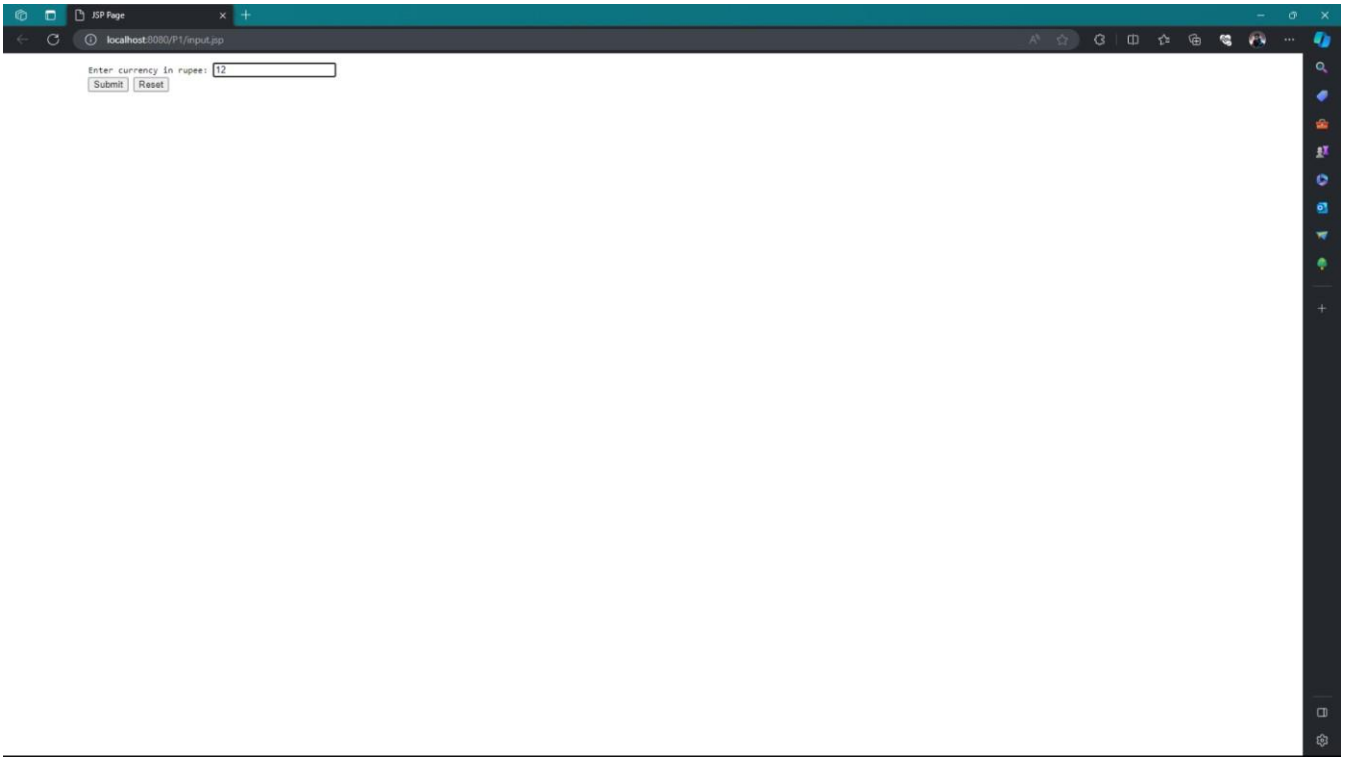
Step-6


P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
P1
Web Pages
WEB-INF
index.html
input.jsp
output.jsp
Source Packages
Libraries
Web Services
Web Service References
Configuration Files
P2
Navigator
Members
INRTODOLL
INRtodoll_converter(double INR) : String
INRTODOLL.java
input.jsp
output.jsp
Source Design History
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package server;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
/**
*
* @author Aaditya
*/
@WebService(serviceName = "INRTODOLL")
public class INRTODOLL {
/**
* Web service operation
*/
@WebMethod(operationName = "INRtodoll_converter")
public String INRtodoll_converter(@WebParam(name = "INR") double INR) {
//TODO write your implementation code here:
return INR + " in the doller: " + (INR/82.85);
}
}
Notifications
28:1
INS

Step -7

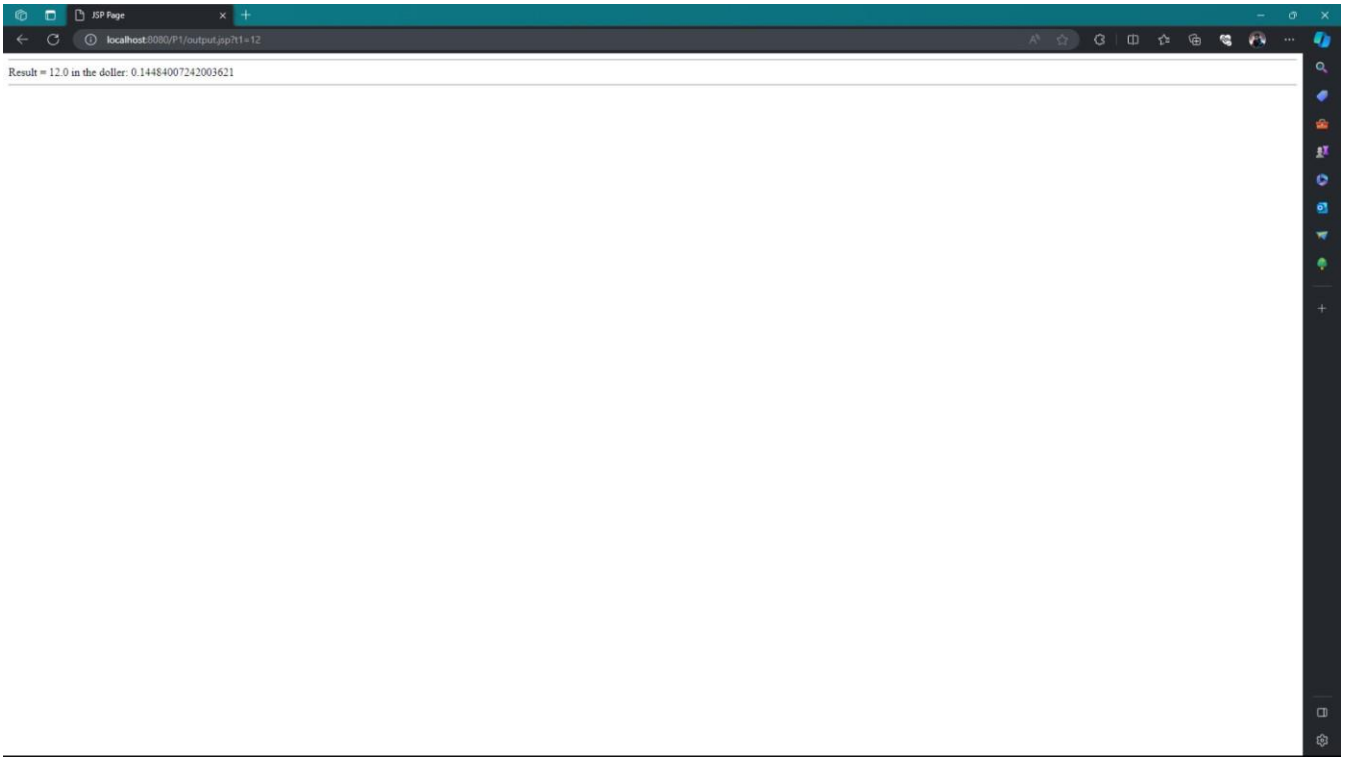

P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
P1
Web Pages
WEB-INF
index.html
input.jsp
output.jsp
Source Packages
Libraries
Web Services
Web Service References
Configuration Files
P2
INRTODOLL.java
input.jsp
output.jsp
Source History
<%--
Document : input
Created on : 13 Mar, 2024, 9:27:52 PM
Author : Aaditya
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form name="" action="output.jsp">
<pre>
Enter currency in rupee<input type="text" name="ti">
<input type="submit"> <input type="reset">
</pre>
</form>
</body>
</html>
Palette
HTML
Table
Ordered List
Unordered List
Image
Link
Meta data
HTML Forms
Form
Text Input
Multi-line Input
Drop-down List
Checkbox
Radio Button
File Select
Button
JSP
JSF
Database
Navigator
html
head
meta
title
body
form
pre
input
input
input
Filters:
html body form pre
Notifications
17:28
INS

Step-8


JSP Page
localhost:8080/P1/input.jsp
Enter currency in rupee: 12
Submit
Reset

Program: B.SC.CS (SEM VI)

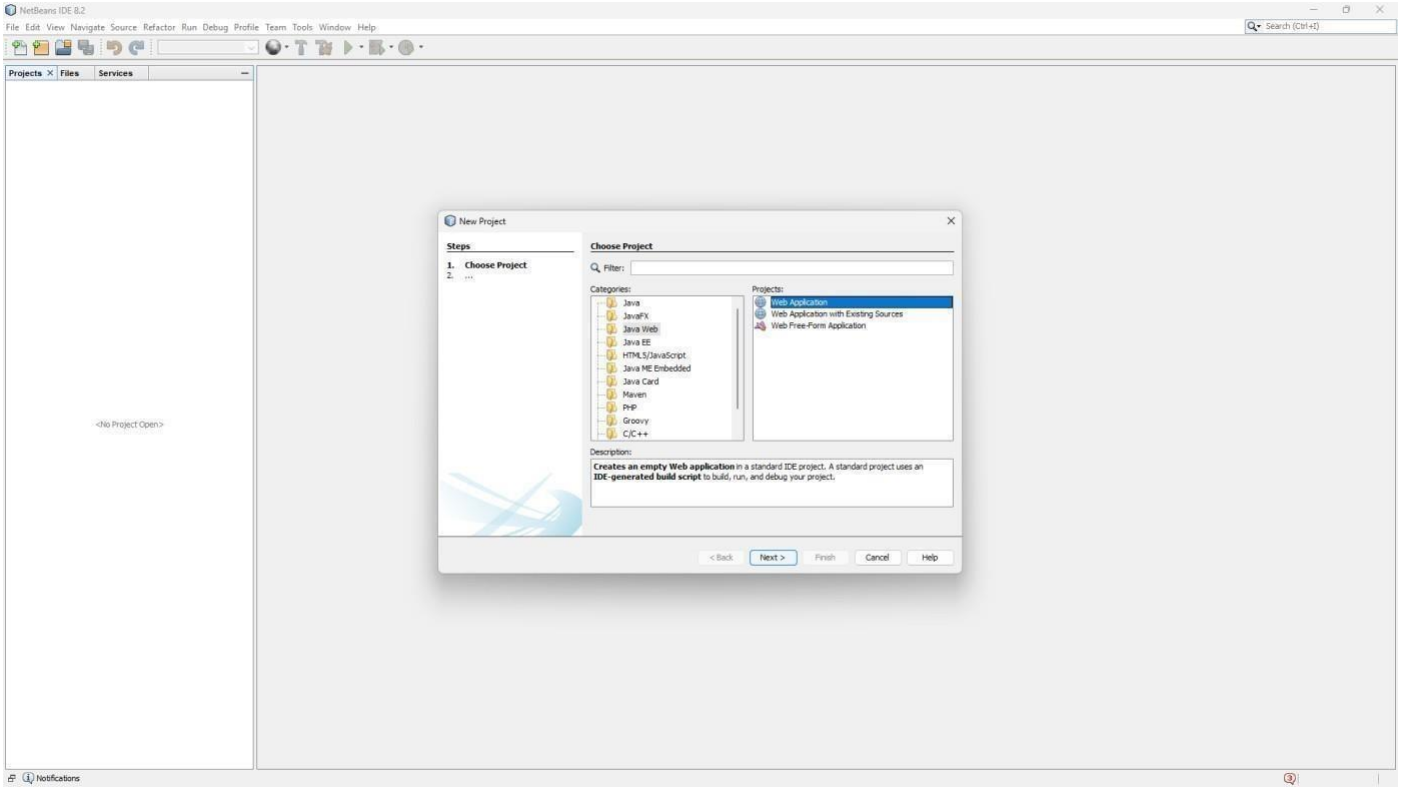
Step-9

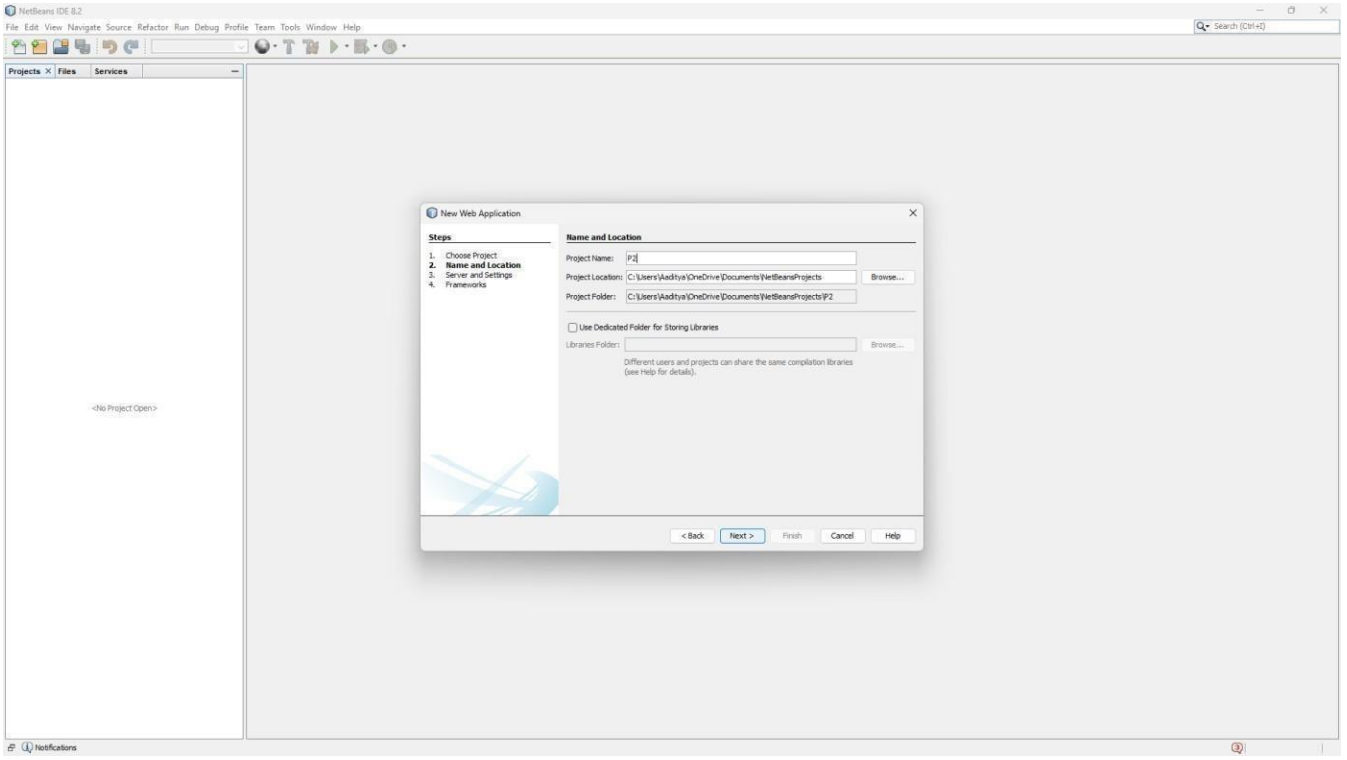


Practical: 2

Aim: Create a Simple SOAP service.

Step-1

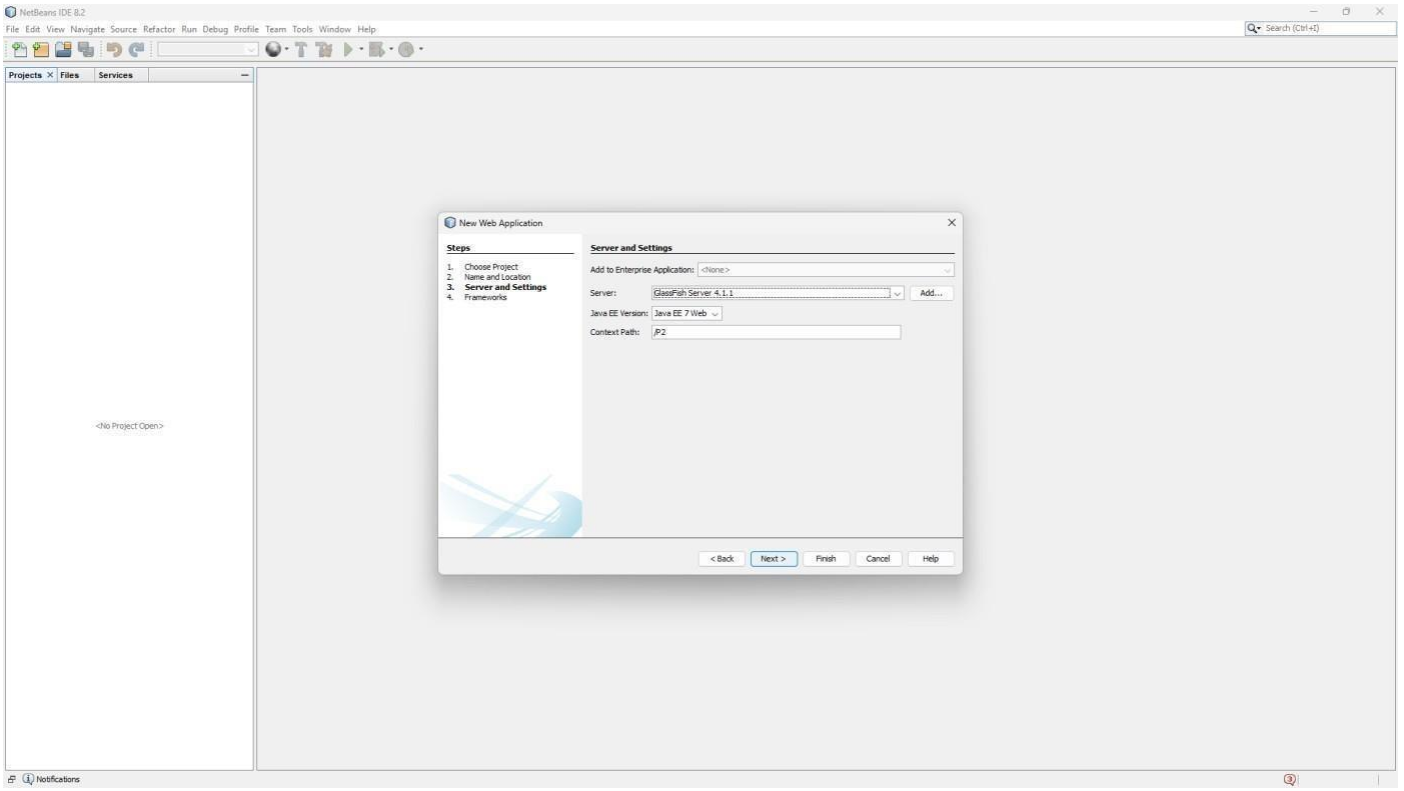




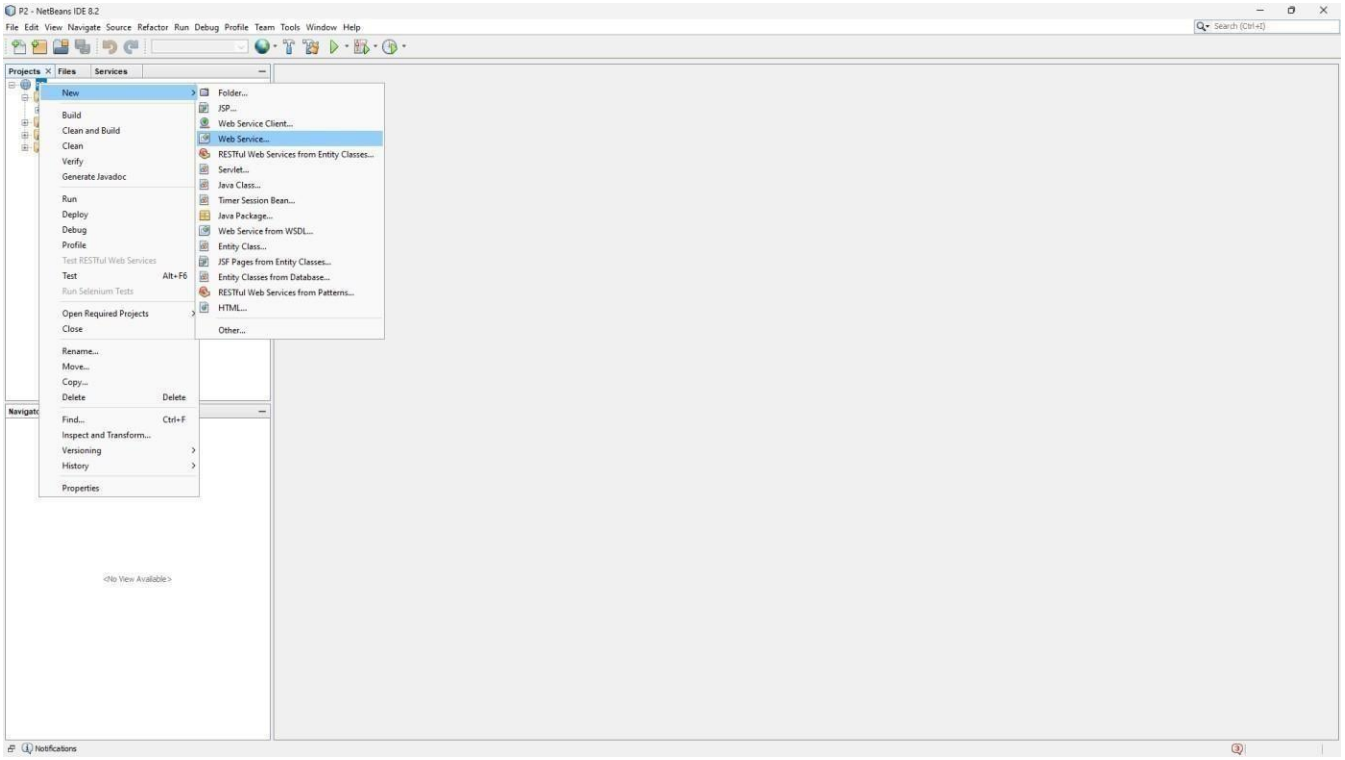
Step-2

Program: B.SC.CS (SEM VI)

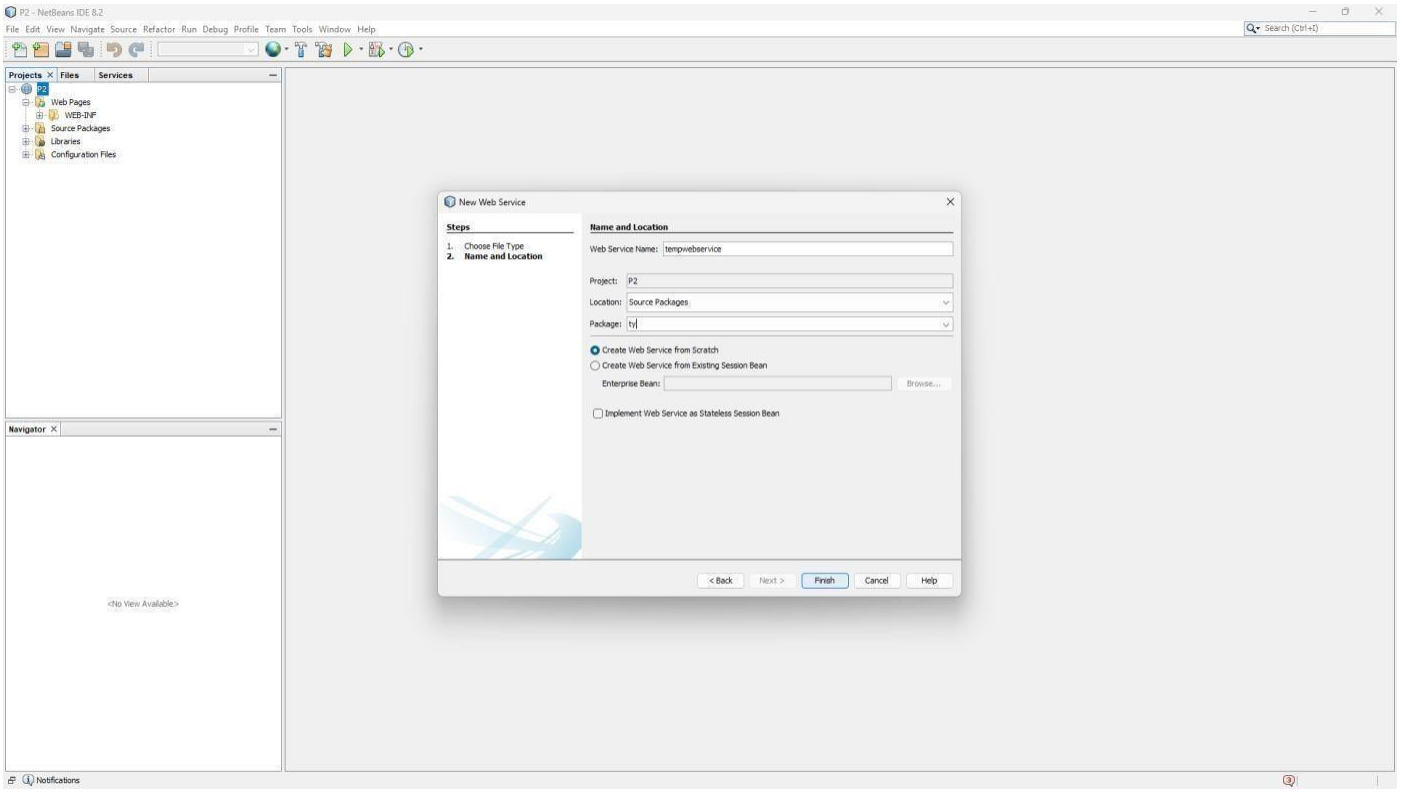
Step-3



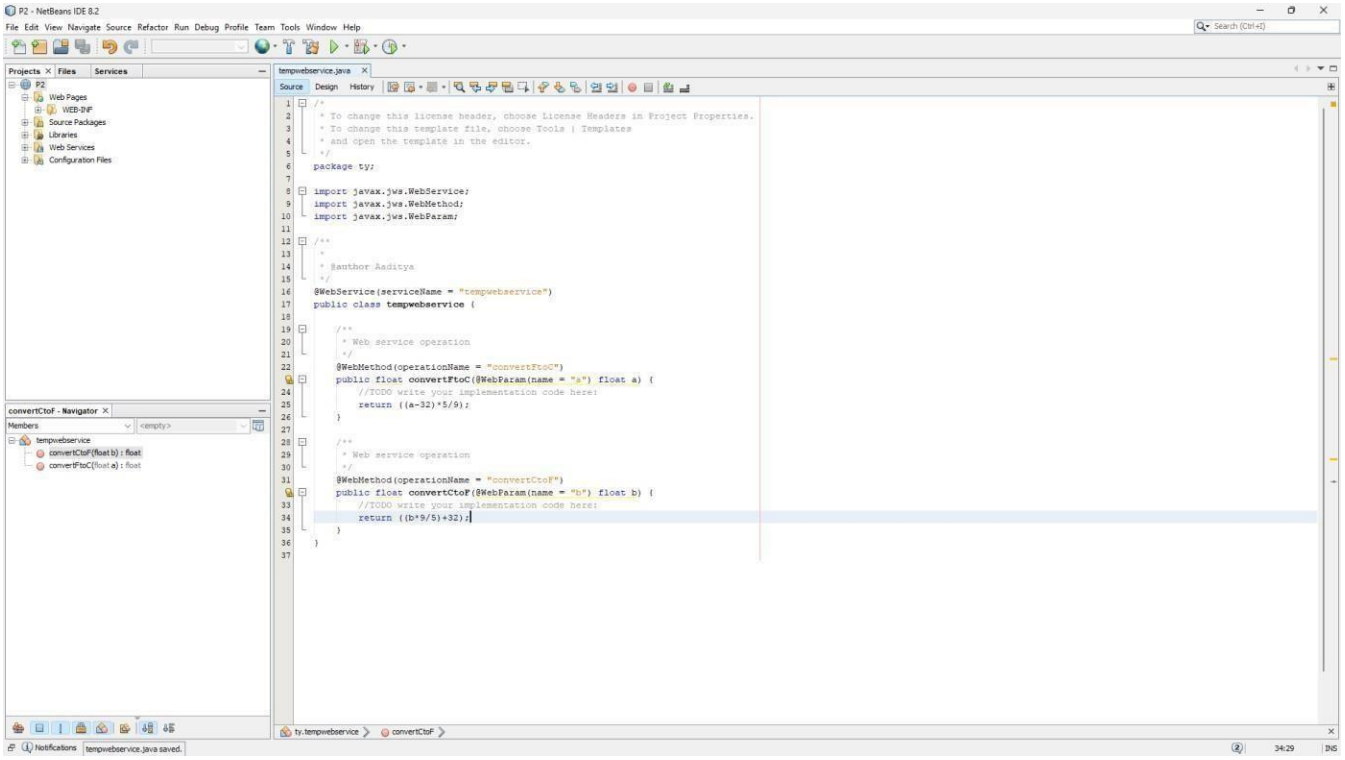
Step-4

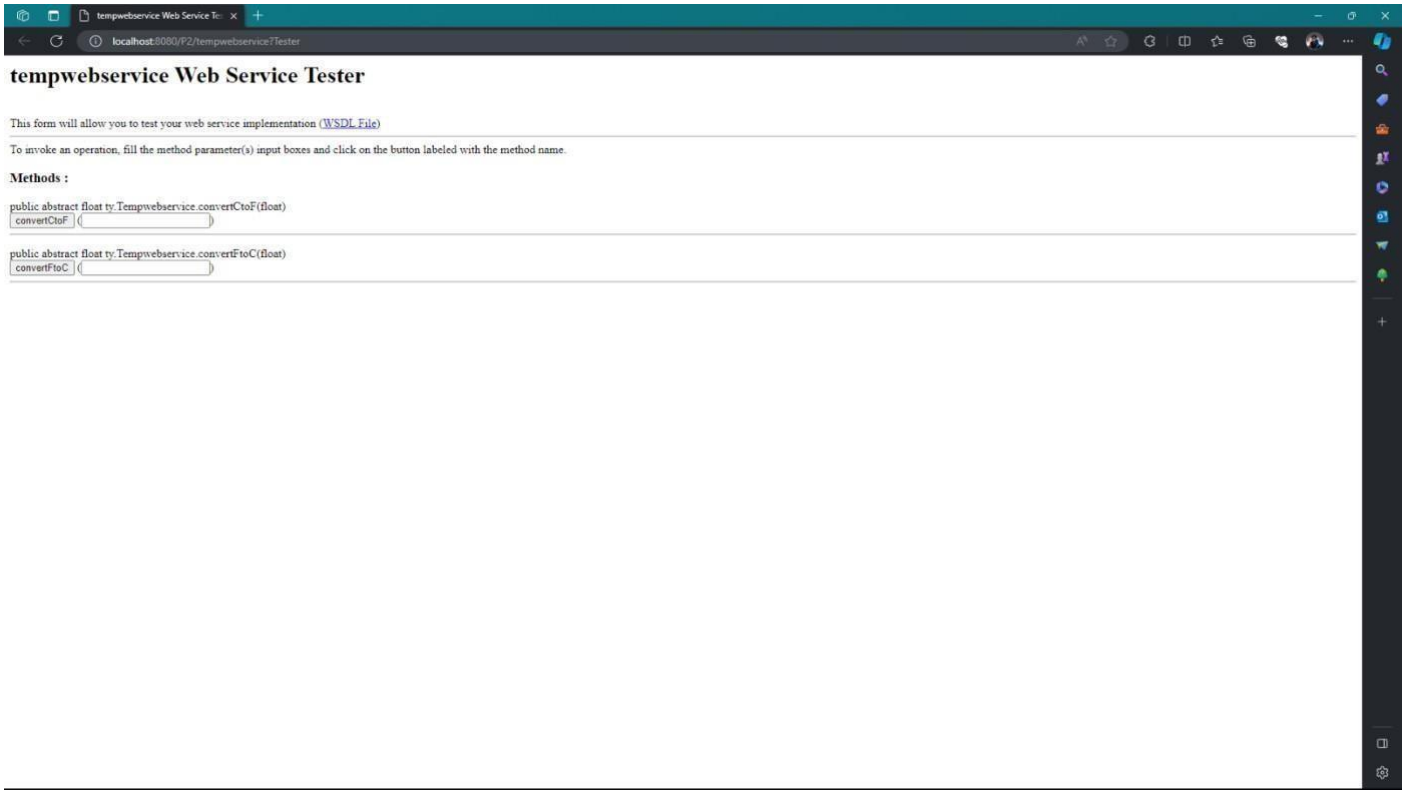

P2 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
JSP...
Web Service Client...
Web Service...
RESTful Web Services from Entity Classes...
Servlet...
Java Class...
Timer Session Bean...
Java Package...
Web Service from WSDL...
Entity Class...
JSF Pages from Entity Classes...
Entity Classes from Database...
RESTful Web Services from Patterns...
HTML...
Other...
<No View Available>
Notifications

Step-5

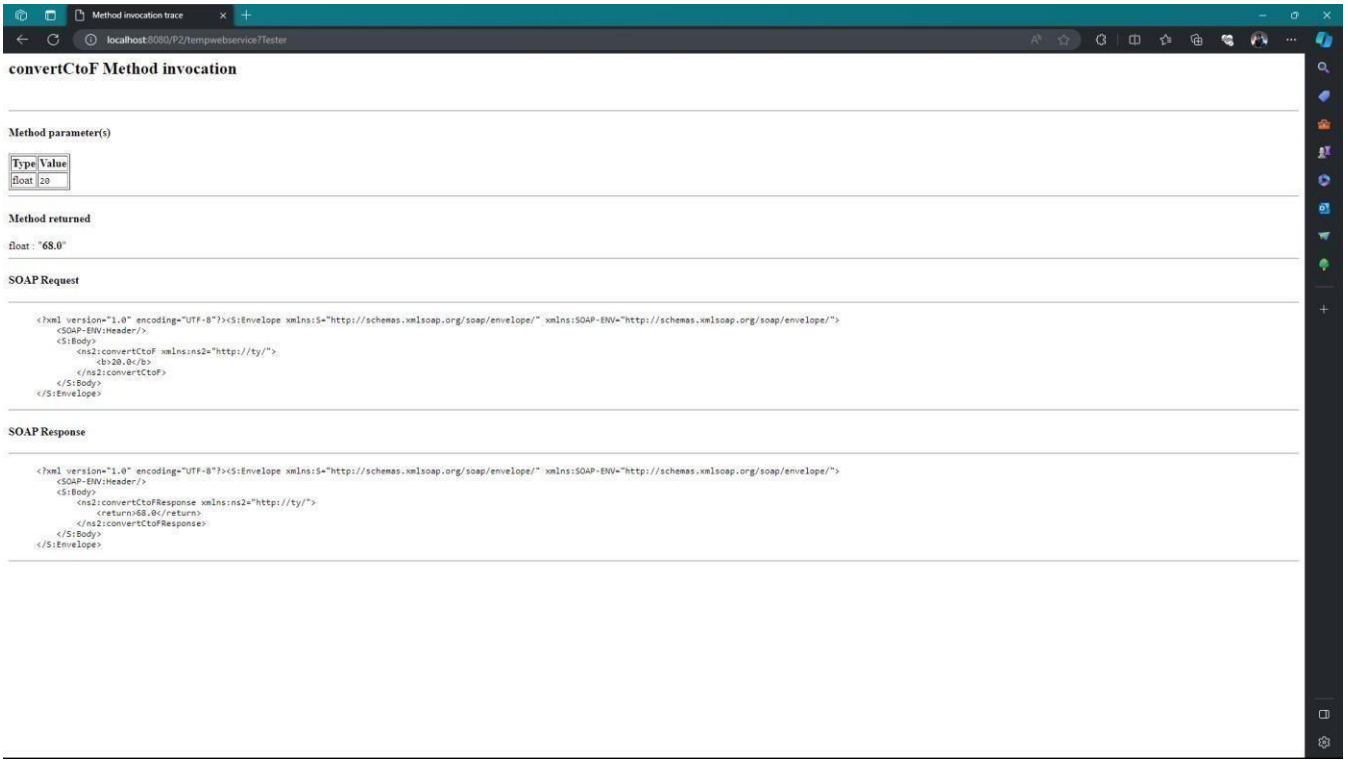

P2 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
P2
Web Pages
WEB-INF
Source Packages
Libraries
Configuration Files
Navigator
<No View Available>
Notifications
New Web Service
Steps
1. Choose File Type
2. Name and Location
Name and Location
Web Service Name: tempwebservice
Project: P2
Location: Source Packages
Package: ty
Create Web Service from Scratch
Create Web Service from Existing Session Bean
Enterprise Bean:
Browse...
Implement Web Service as Stateless Session Bean
< Back
Next >
Finish
Cancel
Help

Step-6

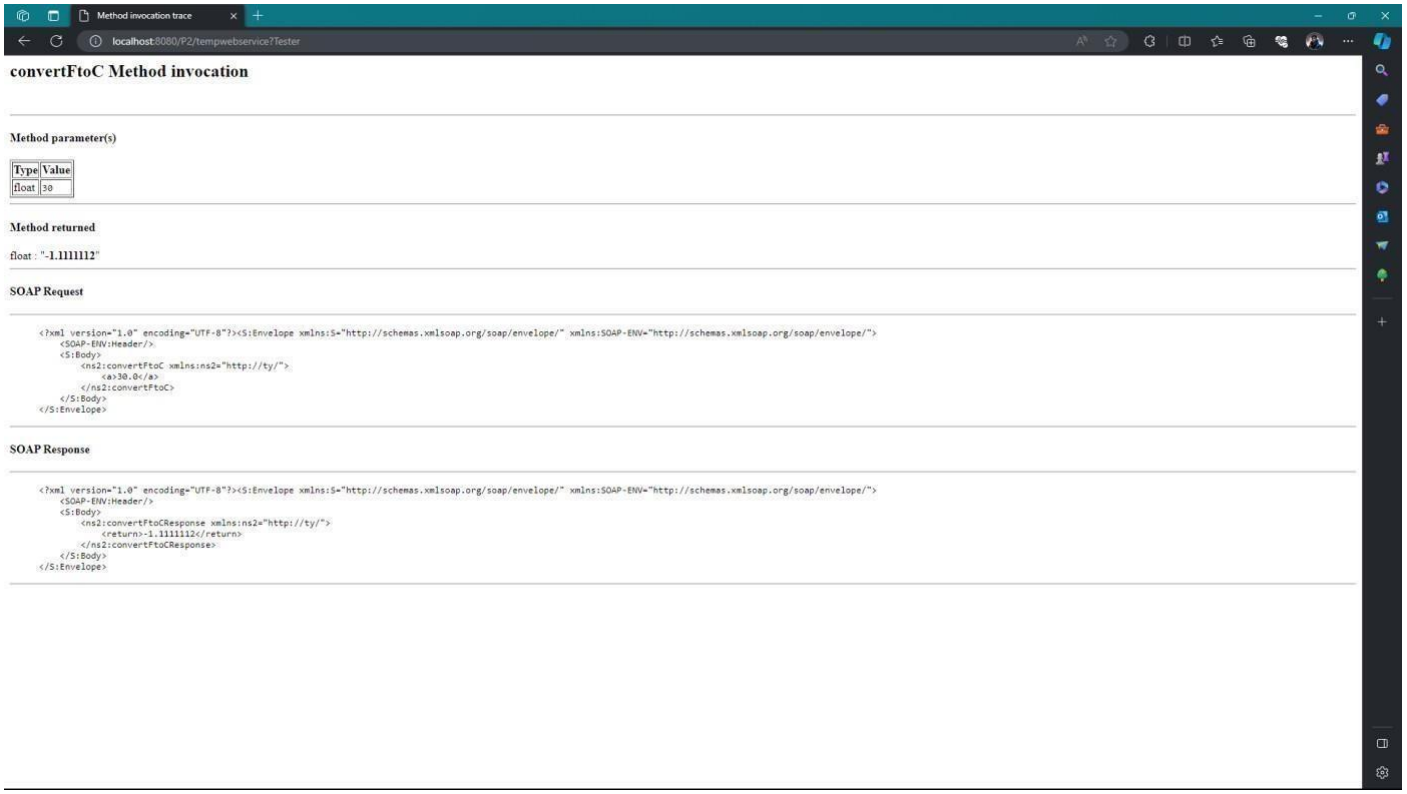

P2 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
P2
Web Pages
WEB-INF
Source Packages
Libraries
Web Services
Configuration Files
convertCtoF - Navigator
Members
<empty>
tempwebservice
convertCtoF(float b) : float
convertFtoC(float a) : float
tempwebservice.java
Source
Design
History
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package ty;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
/**
*
* @author Aaditya
*/
@WebService(serviceName = "tempwebservice")
public class tempwebservice {
/**
* Web service operation
*/
@WebMethod(operationName = "convertFtoC")
public float convertFtoC(@WebParam(name = "a") float a) {
//TODO write your implementation code here:
return ((a-32)*5/9);
}
/**
* Web service operation
*/
@WebMethod(operationName = "convertCtoF")
public float convertCtoF(@WebParam(name = "b") float b) {
//TODO write your implementation code here:
return ((b*9/5)+32);
}
}
ty.tempwebservice
convertCtoF
Notifications
tempwebservice.java saved.
34:29
INS

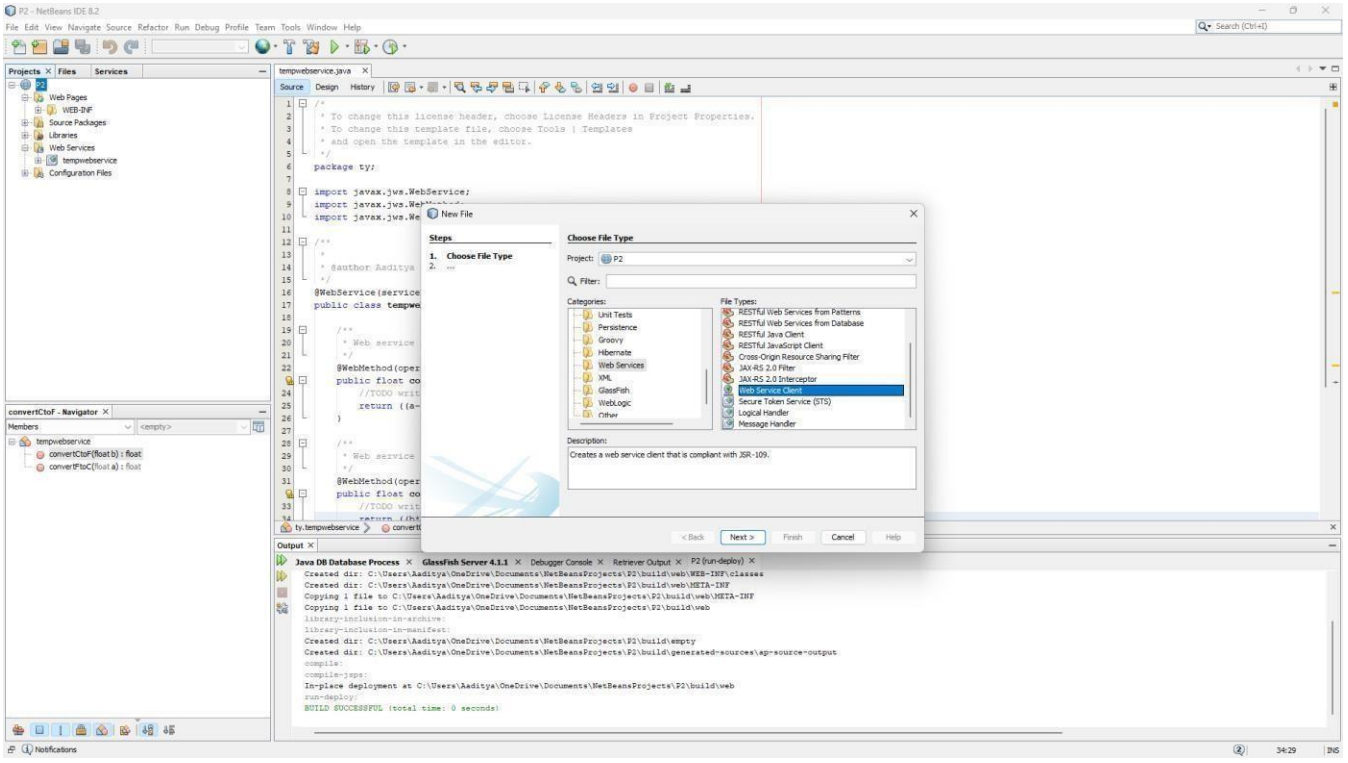

tempwebservice Web Service Te
localhost:8080/P2/tempwebservice?Tester
tempwebservice Web Service Tester
This form will allow you to test your web service implementation (WSDL File)
To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.
Methods :
public abstract float ty.Tempwebservice.convertCtoF(float)
convertCtoF
public abstract float ty.Tempwebservice.convertFtoC(float)
convertFtoC

Step-8


Method invocation trace
localhost:8080/P2/tempwebservice?Tester
convertCtoF Method invocation
Method parameter(s)
Type Value
float 20
Method returned
float : "68.0"
SOAP Request
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:convertCtoF xmlns:ns2="http://ty/">
20.0
</ns2:convertCtoF>
</S:Body>
</S:Envelope>
SOAP Response
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:convertCtoFResponse xmlns:ns2="http://ty/">
<return>68.0</return>
</ns2:convertCtoFResponse>
</S:Body>
</S:Envelope>

Step-9

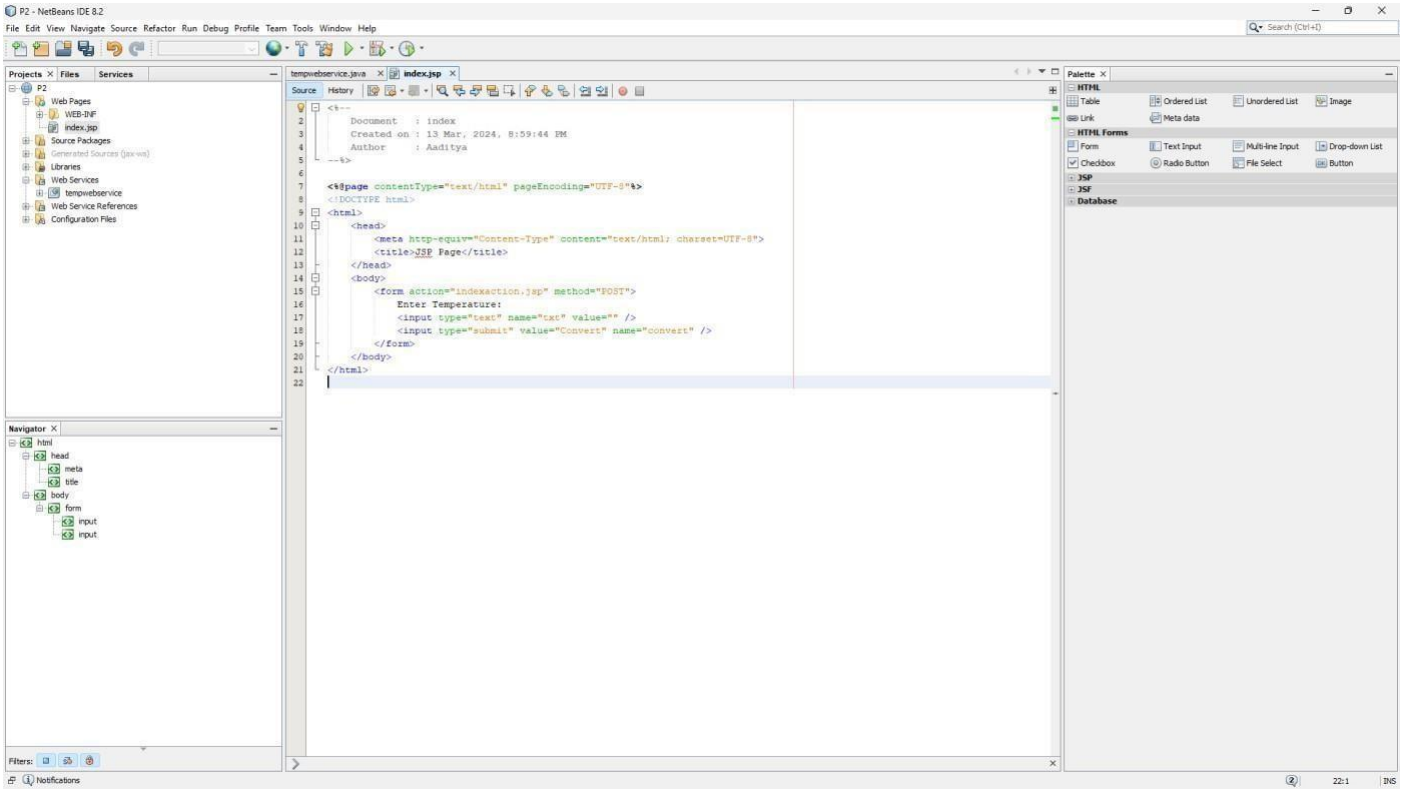

Method invocation trace
localhost:8080/P2/tempwebservice?Tester
convertFtoC Method invocation
Method parameter(s)
Type Value
float 30
Method returned
float : "-1.1111112"
SOAP Request
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:convertFtoC xmlns:ns2="http://ty/">
<a>30.0
</ns2:convertFtoC>
</S:Body>
</S:Envelope>
SOAP Response
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
<ns2:convertFtoCResponse xmlns:ns2="http://ty/">
<return>-1.1111112</return>
</ns2:convertFtoCResponse>
</S:Body>
</S:Envelope>


P2 - NetBeans IDE 8.2
New File
Steps
Choose File Type
Web Service Client
Creates a web service client that is compliant with JSR-109.
BUILD SUCCESSFUL (total time: 0 seconds)

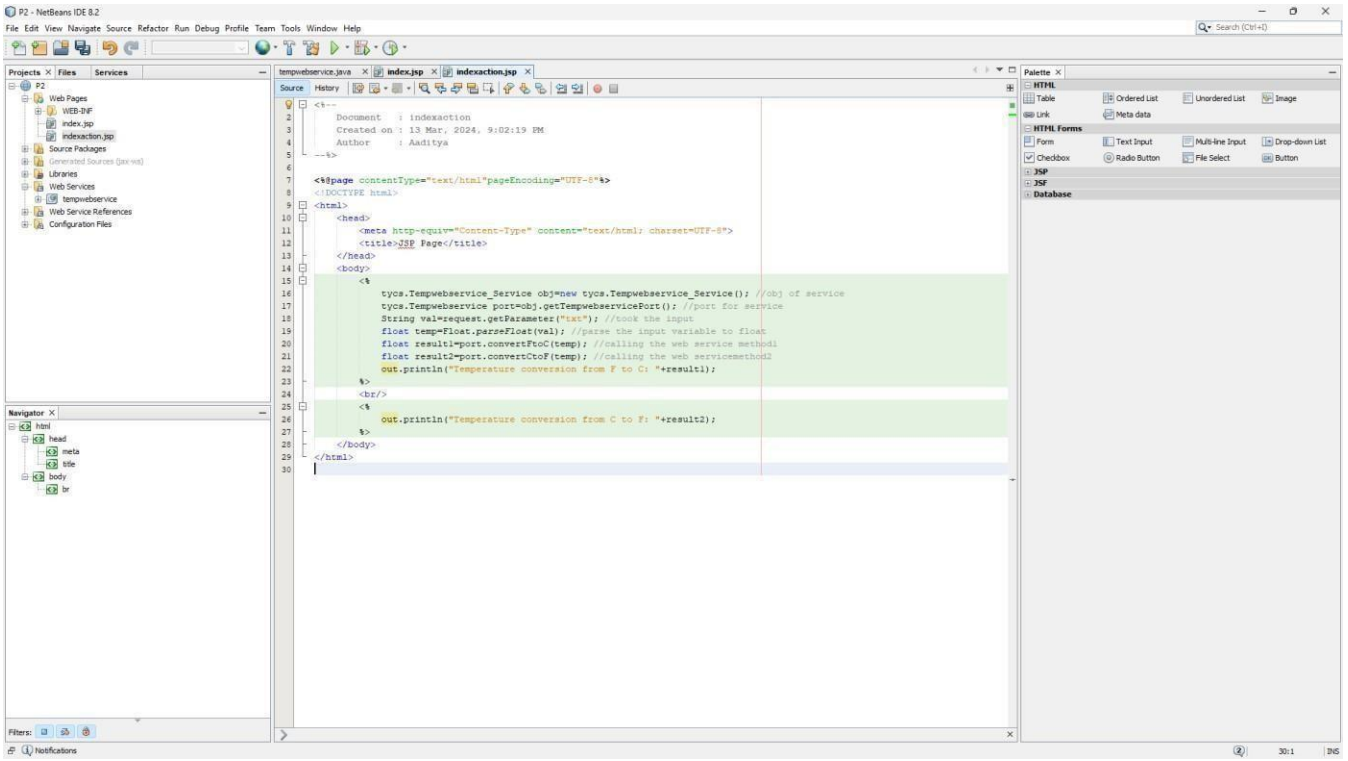
Step-10

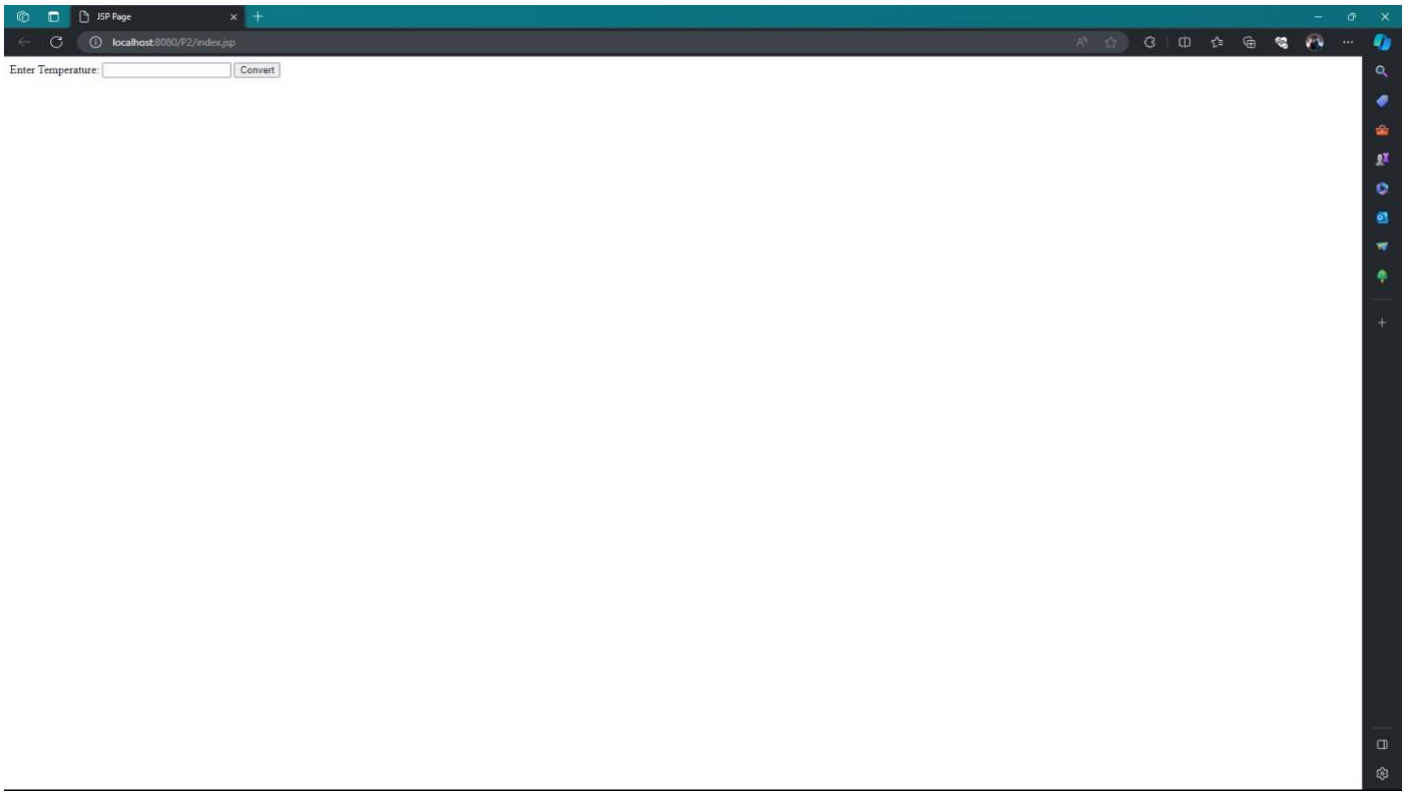
Program: B.SC.CS (SEM VI)

Step-11

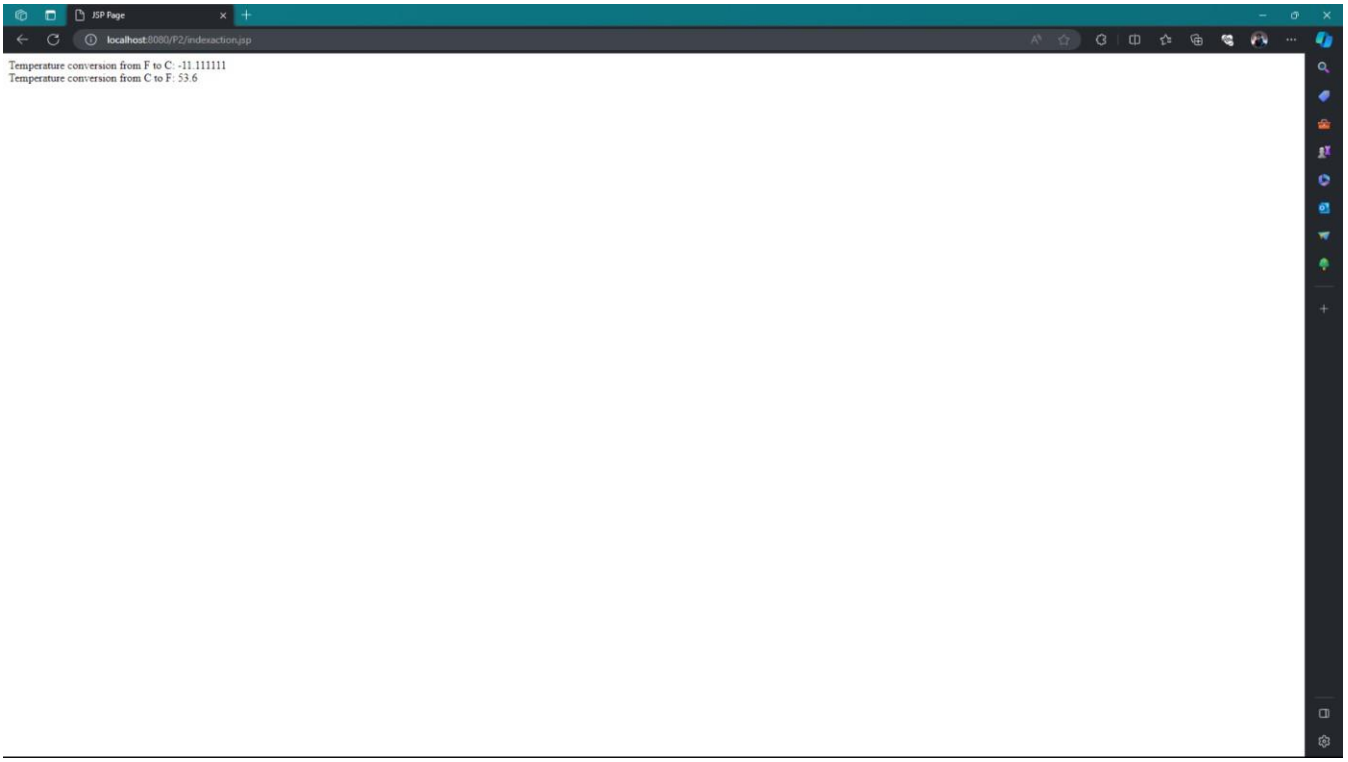


Step-12




JSP Page
localhost:8080/P2/index.jsp
Enter Temperature:
Convert

Step-14

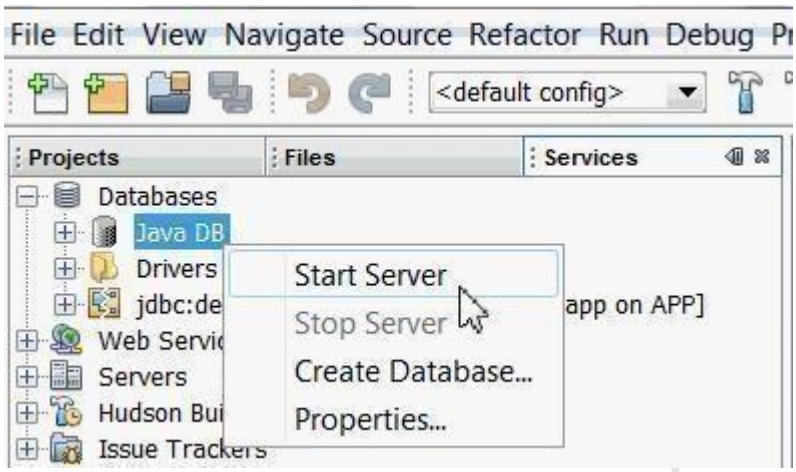

JSP Page
localhost:8080/P2/indexaction.jsp
Temperature conversion from F to C: -11.111111
Temperature conversion from C to F: 53.6

Practical: 3

Aim: Create a Simple REST Service.

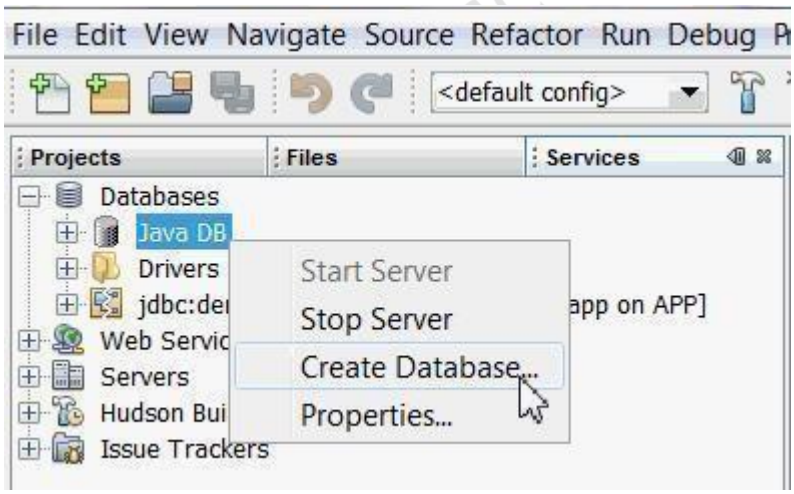
Demonstrate CRUD operations with suitable database using RESTful Web service.

1. To start the Java DB Database from NetBeans, perform the following steps. Click Services tab -> Expand Databases node. ->Right-click Java DB icon.
->Select Start Server



2. To create playerDB database, perform the following steps:

a. Right-click **Java DB** icon, select **Create Database**.



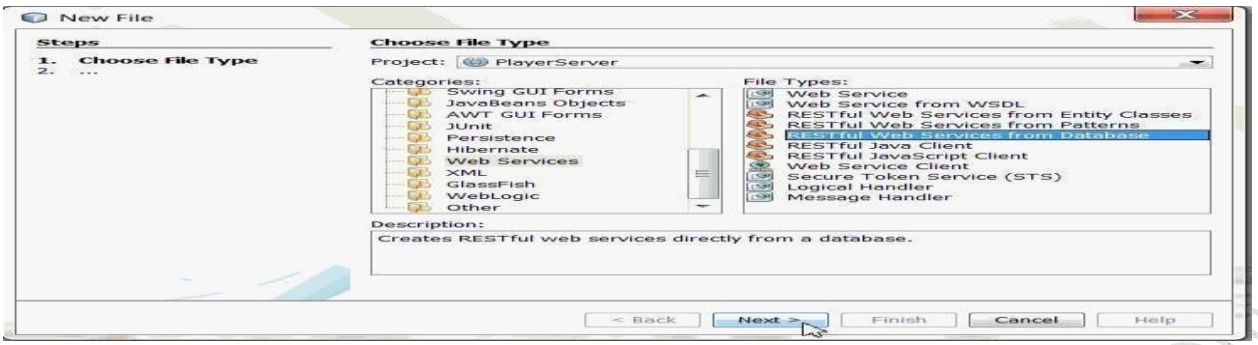
b) Enter the following information for the database:

Database Name: **playerDB**
User Name: **john** Password: **john** Confirm Password:**john**
Click OK.

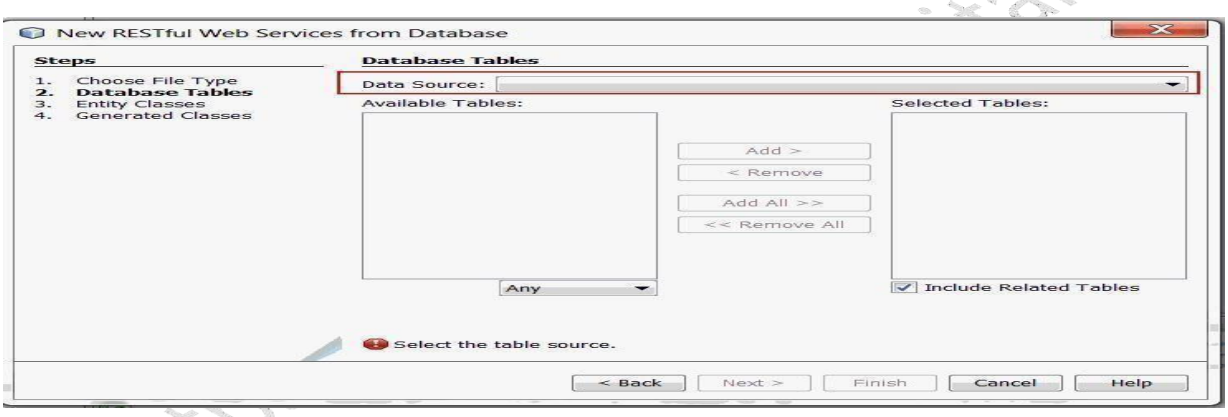
3. To connect to the newly created database playerDB, perform the following steps
   a. Right-click jdbc:derby://localhost:1527/playerDB connection.
   b. Select Connect.

4. Create tables and populate them with data in playerDB database.
   a. In NetBeans select File > Open File.
   b. In the file select playersDB.sql.
   c. Click Open. The script automatically opens in the SQL Editor.
5. Examine the contents of the database.
   a. In the Services window, expand the jdbc:derby://localhost:1527/playerDB connection under the Databases node. b. Right-click the connection and select Refresh.
   c. Expand the john schema. You see the nodes for Tables, Views, and Procedures. d. Expand the Tables node to see the PLAYER and TEAM tables.
   e. Right-click the PLAYER table node and select View Data.
   Player server
   To create RESTful Web Services, you need a Java Web application project. In the below section you will create a demo Java web project, PlayerServer.

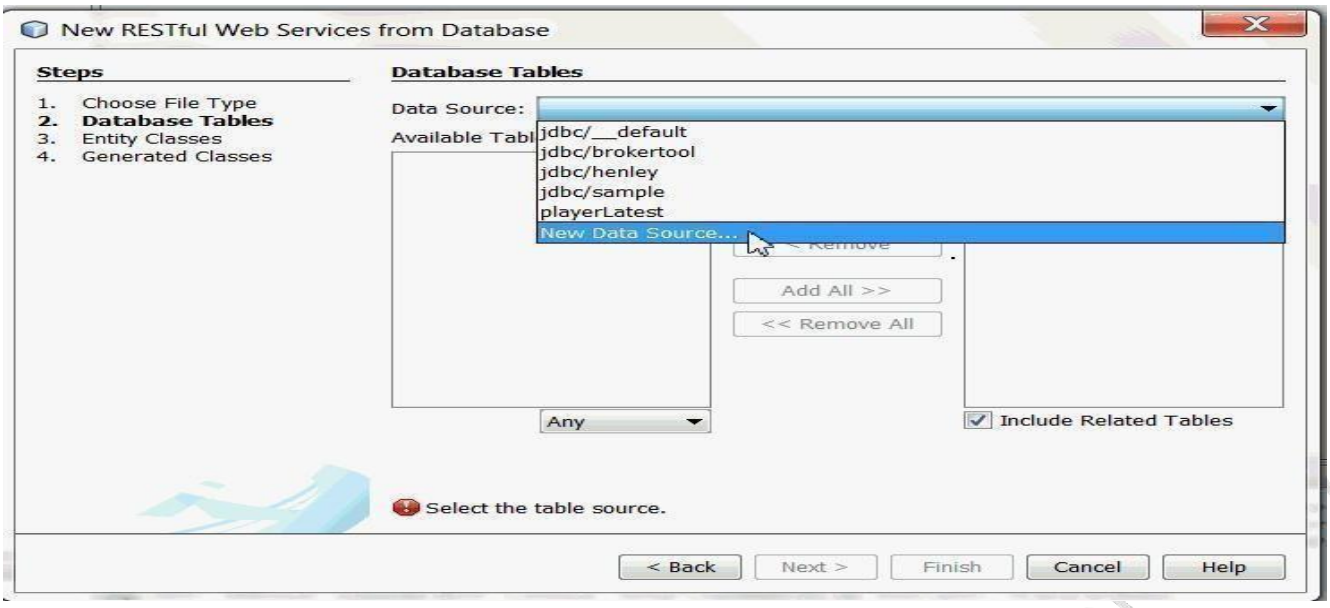
To create new Java Web Project, select File -> New Project -> Java web -> Web Application.
Generating web service
To generate RESTful Web Services do the following:
1. Right-click the PlayerServer and choose New > Other > Web Services > RESTful Web Services from Database. The New RESTful Web Service wizard opens on the Database Tables panel.



In the Database Tables window, select Data Source.



Next select "New Data Source" from the Drop-down list.



In the Create Data Source Window, enter the following information:

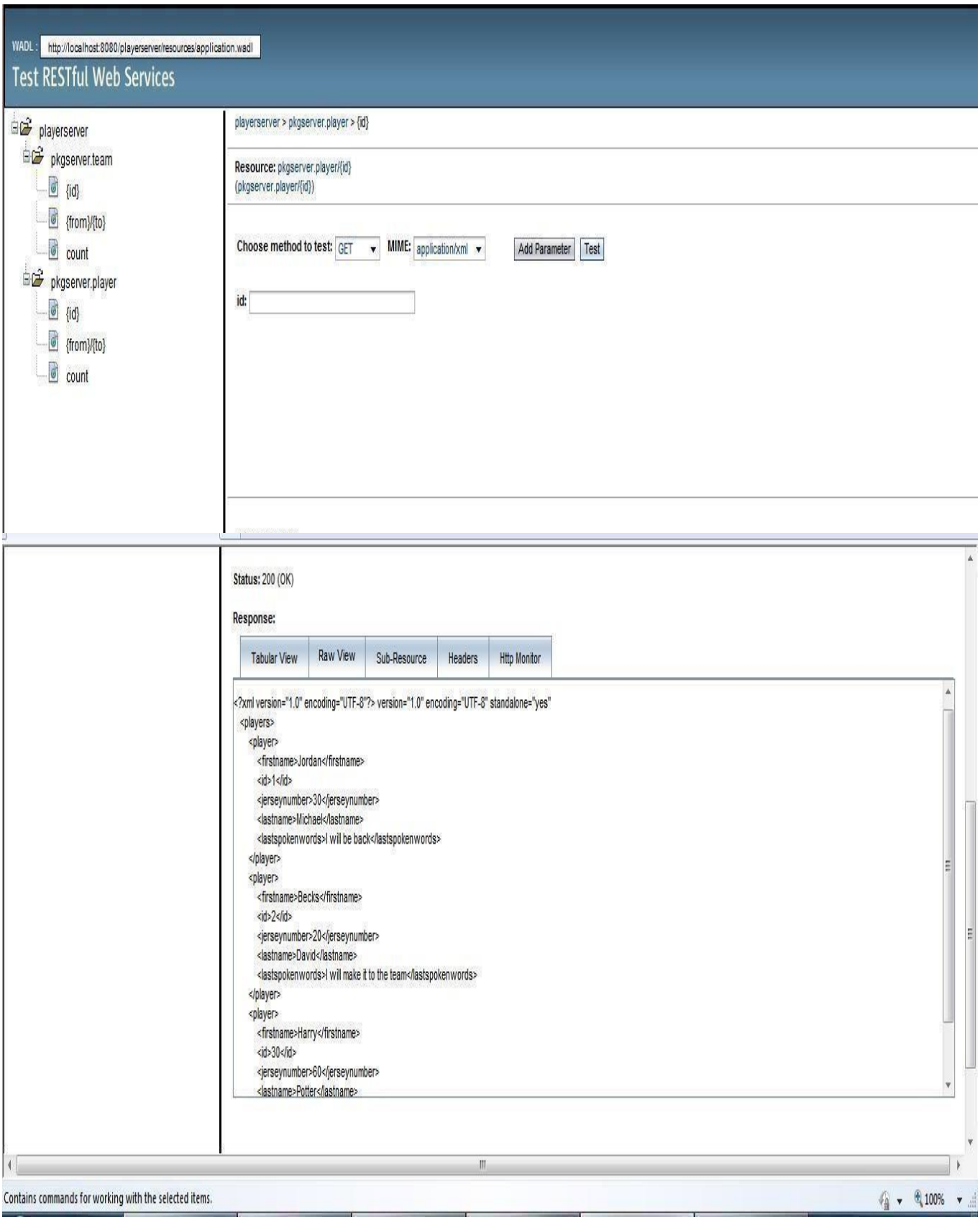
- **JNDI name**: jdbc/playerDB
- **Database connection :** select jdbc:derby://localhost:1527/playerDB[john on JOHN]

b. Click OK.

The PLAYER and TEAM tables are displayed under the Available Tables column.

b)

Click **Add All**. The PLAYER and TEAM tables are displayed under the Selected Tables column.


WADL : http://localhost:8080/playerserver/resources/application.wadl
Test RESTful Web Services
playerserver
pkgserver.team
{id}
{from}/{to}
count
pkgserver.player
{id}
{from}/{to}
count
playerserver > pkgserver.player > {id}
Resource: pkgserver.player/{id}
(pkgserver.player/{id})
Choose method to test: GET
MIME: application/xml
Add Parameter
Test
id:
Status: 200 (OK)
Response:
Tabular View
Raw View
Sub-Resource
Headers
Http Monitor
<?xml version="1.0" encoding="UTF-8"?> version="1.0" encoding="UTF-8" standalone="yes"
<players>
<player>
<firstname>Jordan</firstname>
<id>1</id>
<jerseynumber>30</jerseynumber>
<lastname>Michael</lastname>
<lastspokenwords>I will be back</lastspokenwords>
</player>
<player>
<firstname>Becks</firstname>
<id>2</id>
<jerseynumber>20</jerseynumber>
<lastname>David</lastname>
<lastspokenwords>I will make it to the team</lastspokenwords>
</player>
<player>
<firstname>Harry</firstname>
<id>30</id>
<jerseynumber>60</jerseynumber>
<lastname>Potter</lastname>
Contains commands for working with the selected items.
100%

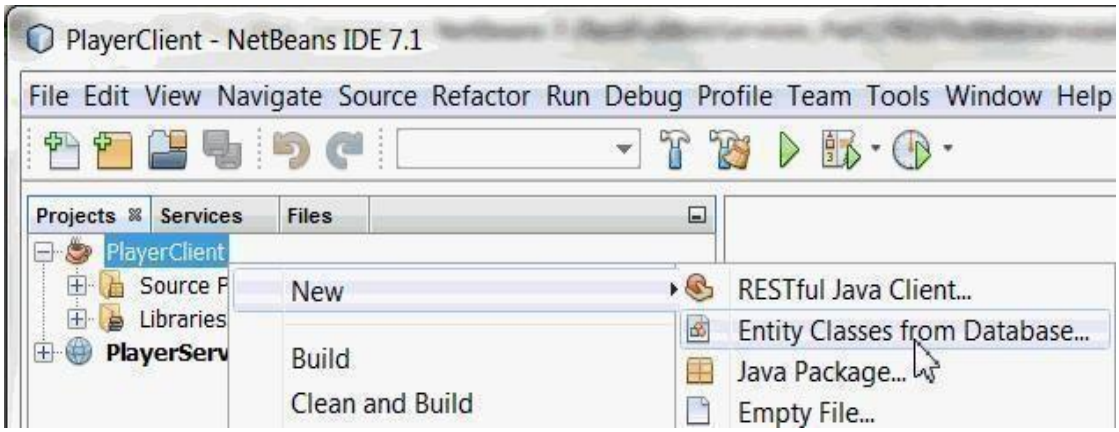
Player Client

To create new Java Project, select File > New Project >Playerclient

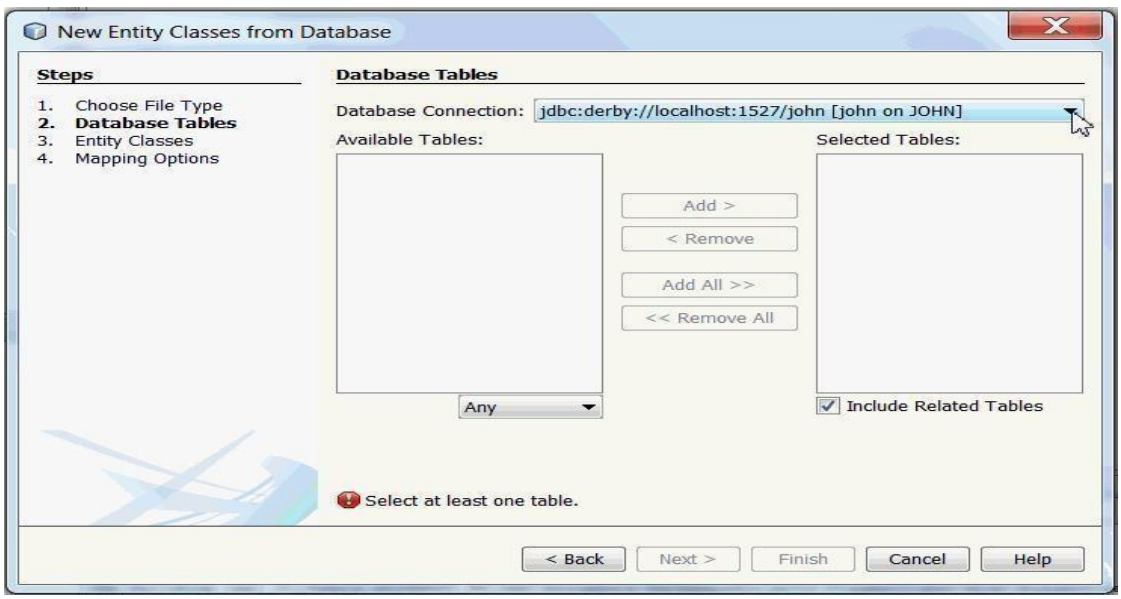
Generating Entity Classes from Database

Implementing CRUD operations on the database, requires creation of Entity Classes to communicate with the database. The following section demonstrates how to create Entity Classes from database.

1. Right-click PlayerClient Project and select New > Entity Classes From Database.

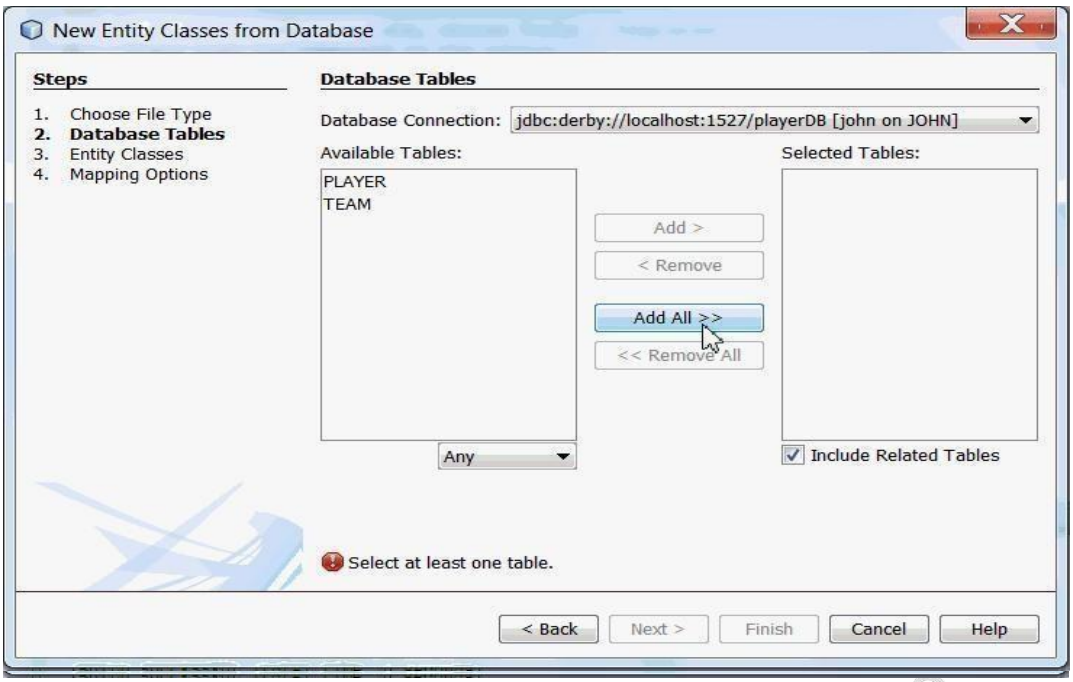


In the Database Tables window, **Database Connection** field select jdbc:derby://localhost:1527/playerDB[john on JOHN] from the drop-down list.



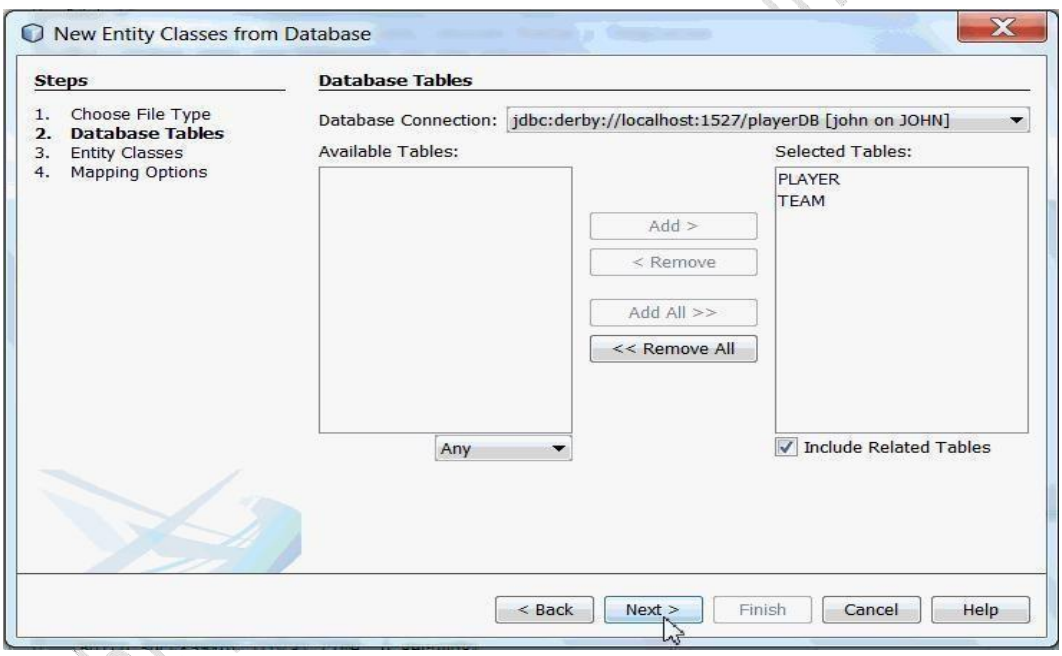
You see PLAYER and TEAM tables in Available Tables category

c. Click Add All

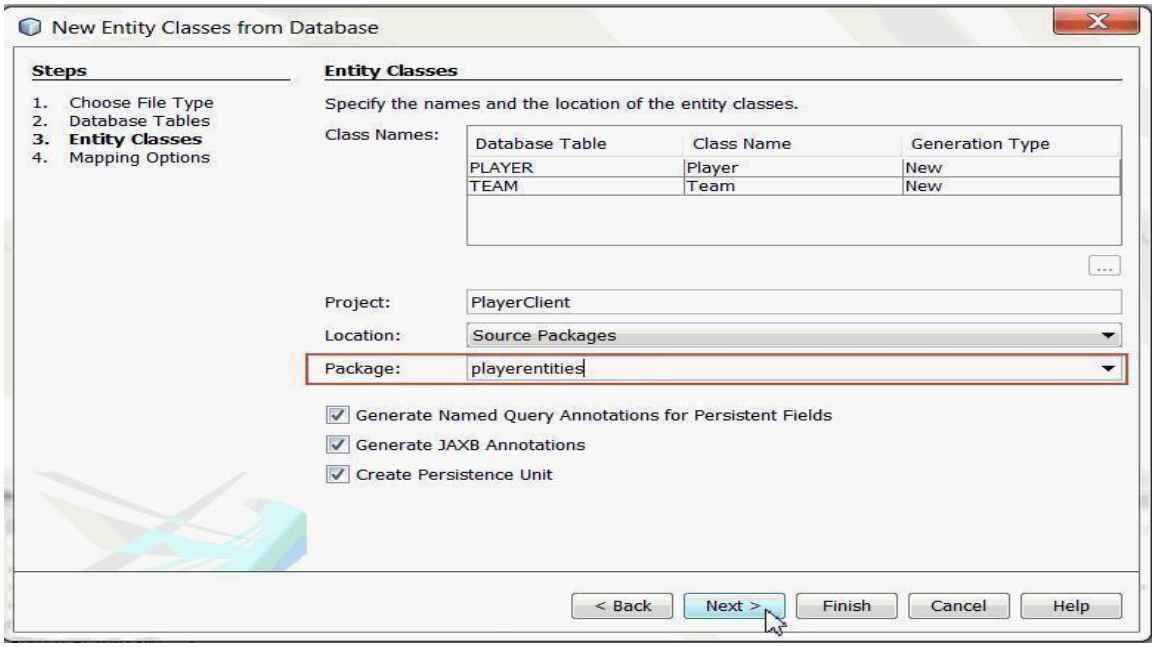


You see both the tables PLAYER and TEAM in Selected Tables Category.

c. Click Next



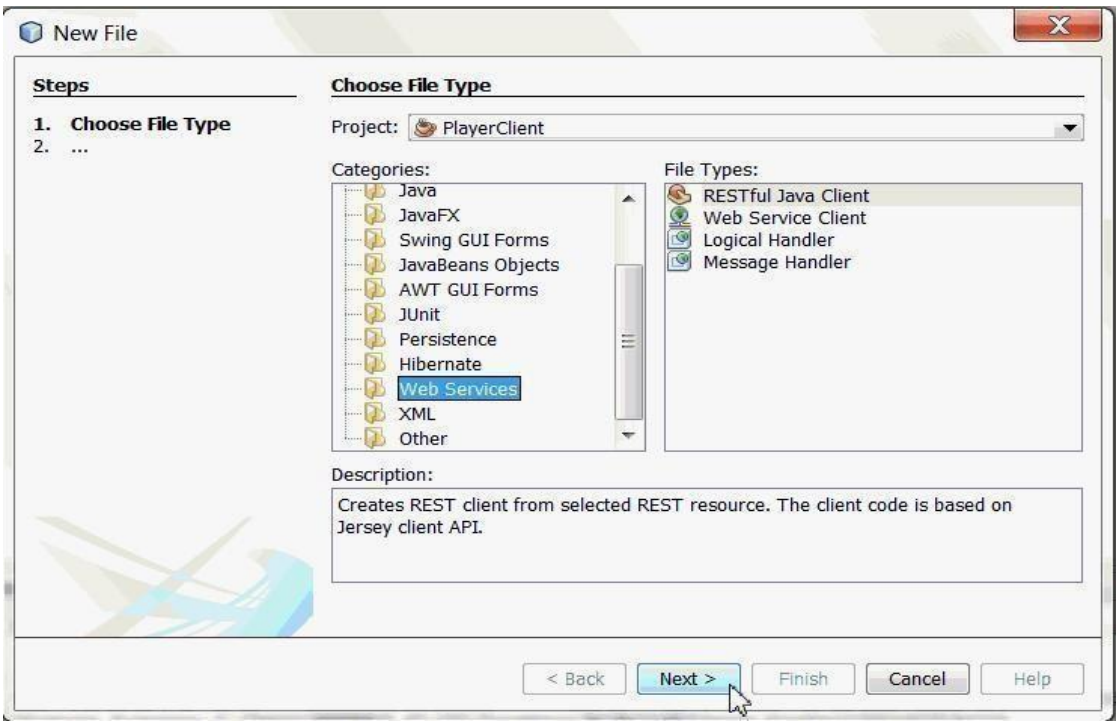
In the Entity classes Window, enter the Package Name as **playerentities** and click Next.



Create Clint:

Complete the following steps to create RESTful client in PlayerClient project.

a. Right-click the PlayerClient and choose New > Other > Web Services > RESTful Java Client. Click Next

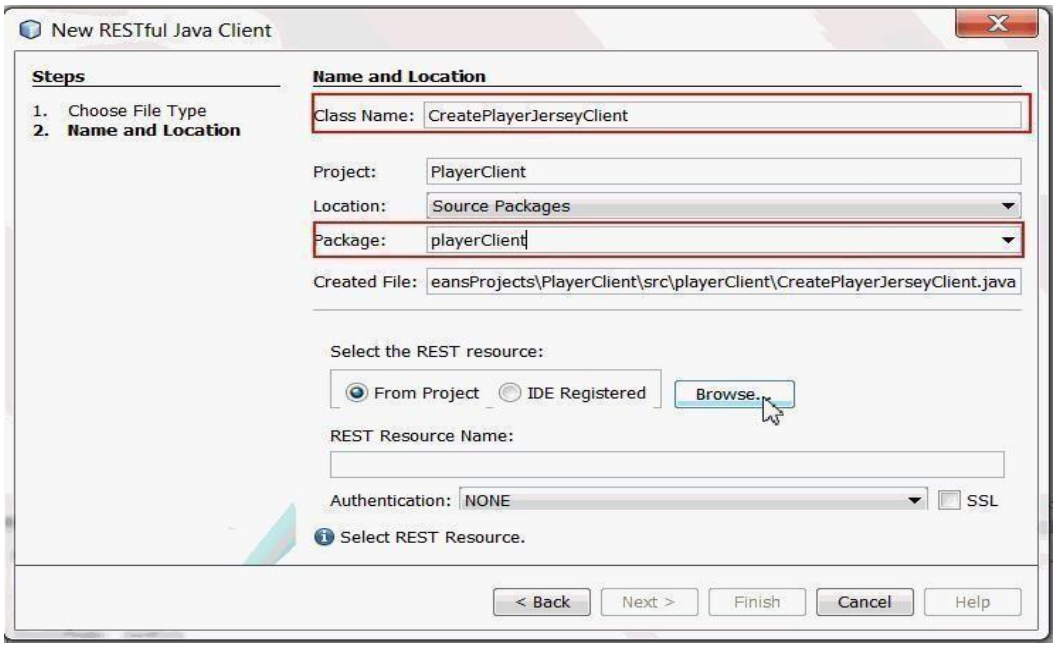


1) a.Enter the following details for the fields in the New Restful Java

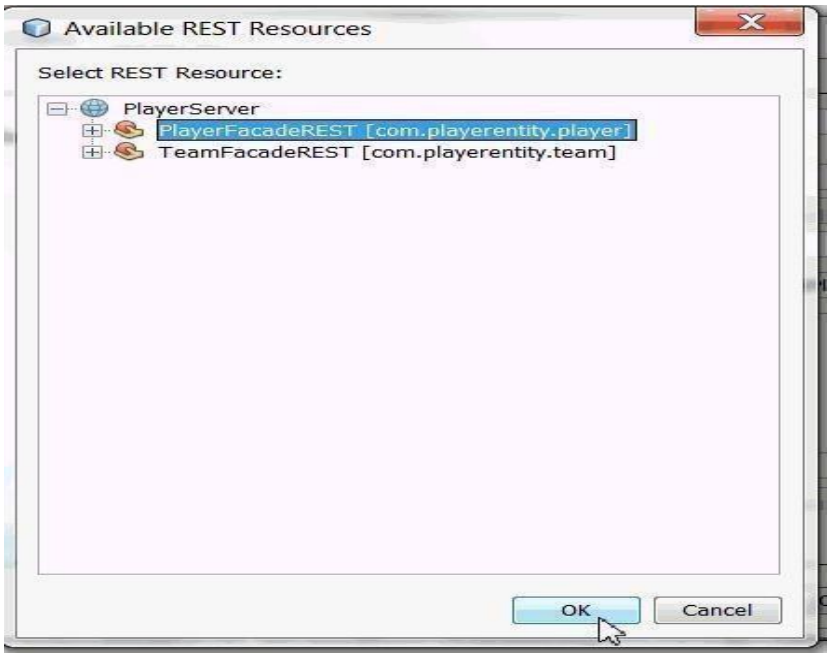
2) Client window :

- Class Name: CreatePlayerJerseyClient
- Package: playerClient

Select the REST resource: From Project and Click Browse.



b. Select PlayerServer.
c. Expand the PlayerServer application.
d. Select PlayerFacadeREST[com.playerentity.player].



Add the main method.

DatabaseConnectionCheck.java the source code
: import java.sql.Connection; import
java.sql.DriverManager; import
java.sql.ResultSet; import
java.sql.SQLException; import
java.sql.Statement; public class
DatabaseConnectionCheck {

```
    public static void main(String[] args) {
        // Define database connection parameters for Derby
        String url = "jdbc:derby://localhost:1527/playerdb";
        String username = "root"; // Adjust if your Derby setup requires a username
        String password = "root"; // Adjust if your Derby setup requires a password

        // Try connecting to the database and reading data
        try {
            // Load the Derby JDBC driver
            Class.forName("org.apache.derby.jdbc.ClientDriver");

            // Attempt to establish the connection
            Connection connection = DriverManager.getConnection(url, username, password);

            // If connection is successful, print a success message
            System.out.println("Connected to the database!");

            // Create a Statement object for executing SQL queries
            Statement statement = connection.createStatement();

            // Execute a query to select data from a table (assuming table name is "players")
            ResultSet resultSet = statement.executeQuery("SELECT * FROM player");

            // Print the retrieved data
            System.out.println("Retrieving and displaying player details:");
            while (resultSet.next()) {
                // Assuming the table has columns named "firstname", "id", and "jerseynumber"
                System.out.println("PlayerID: " + resultSet.getInt("id"));
                System.out.println("FirstName: " + resultSet.getString("playername"));

            }

            // Close the ResultSet, Statement, and
            Connection resultSet.close(); statement.close();
            connection.close();
        } catch (ClassNotFoundException e) {
            // Handle the case where the JDBC driver is not found
            System.err.println("Derby JDBC driver not found!");
            e.printStackTrace();
```

```
        } catch (SQLException e) {
            // Handle any SQL errors that might occur during the connection attempt or query execution


            System.err.println("Failed to connect to the database or execute the query!");
            e.printStackTrace();
        }
    }
}
```

Insert.java source code is:

```
package playerpkg;

import      java.sql.Connection;
import
java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
public class Inseart {

    public static void main(String[] args) {
        // Define database connection parameters for Derby
        String url = "jdbc:derby://localhost:1527/playerdb";
        String username = "root"; // Adjust if your Derby setup requires a username
        String password = "root"; // Adjust if your Derby setup requires a password

        // Try connecting to the database and performing the insert operation
        try {
            // Load the Derby JDBC driver
            Class.forName("org.apache.derby.jdbc.ClientDriver");

            // Attempt to establish the connection
            Connection connection = DriverManager.getConnection(url, username, password);

            // If connection is successful, print a success message
            System.out.println("Connected to the database!");

            // Create a Statement object for executing SQL queries
            Statement statement = connection.createStatement();

            // Execute an insert statement to add a new record to the table
            int rowsAffected = statement.executeUpdate("INSERT INTO player (id, playername)
    VALUES (3, 'Aman')");

            // Check if the insert was successful
            if (rowsAffected > 0) {
```

```
                System.out.println("Insert successful!");
            } else {
                System.out.println("Insert failed!");
            }


            // Close the Statement and Connection
            statement.close();
            connection.close();

        } catch (ClassNotFoundException e) {

            // Handle the case where the JDBC driver is not found
            System.err.println("Derby JDBC driver not found!");
            e.printStackTrace();
        } catch (SQLException e) {
            // Handle any SQL errors that might occur during the connection attempt or insert operation
            System.err.println("Failed to connect to the database or execute the insert operation!");
            e.printStackTrace();
        }
    }
}
```

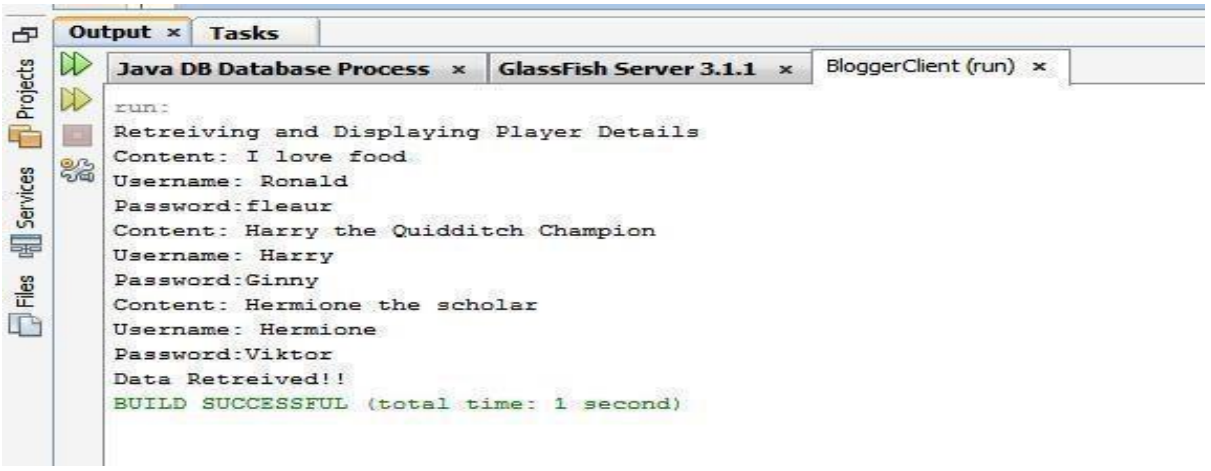
Update.java source code:

```
package playerpkg;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
public class Update {

    public static void main(String[] args) {
        // Define database connection parameters for Derby
        String url = "jdbc:derby://localhost:1527/playerdb";
        String username = "root"; // Adjust if your Derby setup requires a username
        String password = "root"; // Adjust if your Derby setup requires a password

        // Try connecting to the database and performing the update operation
        try {
            // Load the Derby JDBC driver
            Class.forName("org.apache.derby.jdbc.ClientDriver");

            // Attempt to establish the connection
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
// If connection is successful, print a success message
System.out.println("Connected to the database!");

// Create a Statement object for executing SQL queries
Statement statement = connection.createStatement();
```

```
            // Execute an update statement to modify an existing record in the table
            int rowsAffected = statement.executeUpdate("UPDATE player SET playername =
'Rohan'WHERE id = 2");

            // Check if the update was successfulif
            (rowsAffected > 0) {
                System.out.println("Update successful!");
            } else {
                System.out.println("Update failed!");
            }


            // Close the Statement and
            Connectionstatement.close();
            connection.close();

        } catch (ClassNotFoundException e) {
            // Handle the case where the JDBC driver is not foundSystem.err.println("Derby
            JDBC driver not
            found!"); e.printStackTrace();
        } catch (SQLException e) {
            // Handle any SQL errors that might occur during the connection attempt or update
        operationSystem.err.println("Failed to connect to the database or execute the update
        operation!"); e.printStackTrace(); }
    }
}
```

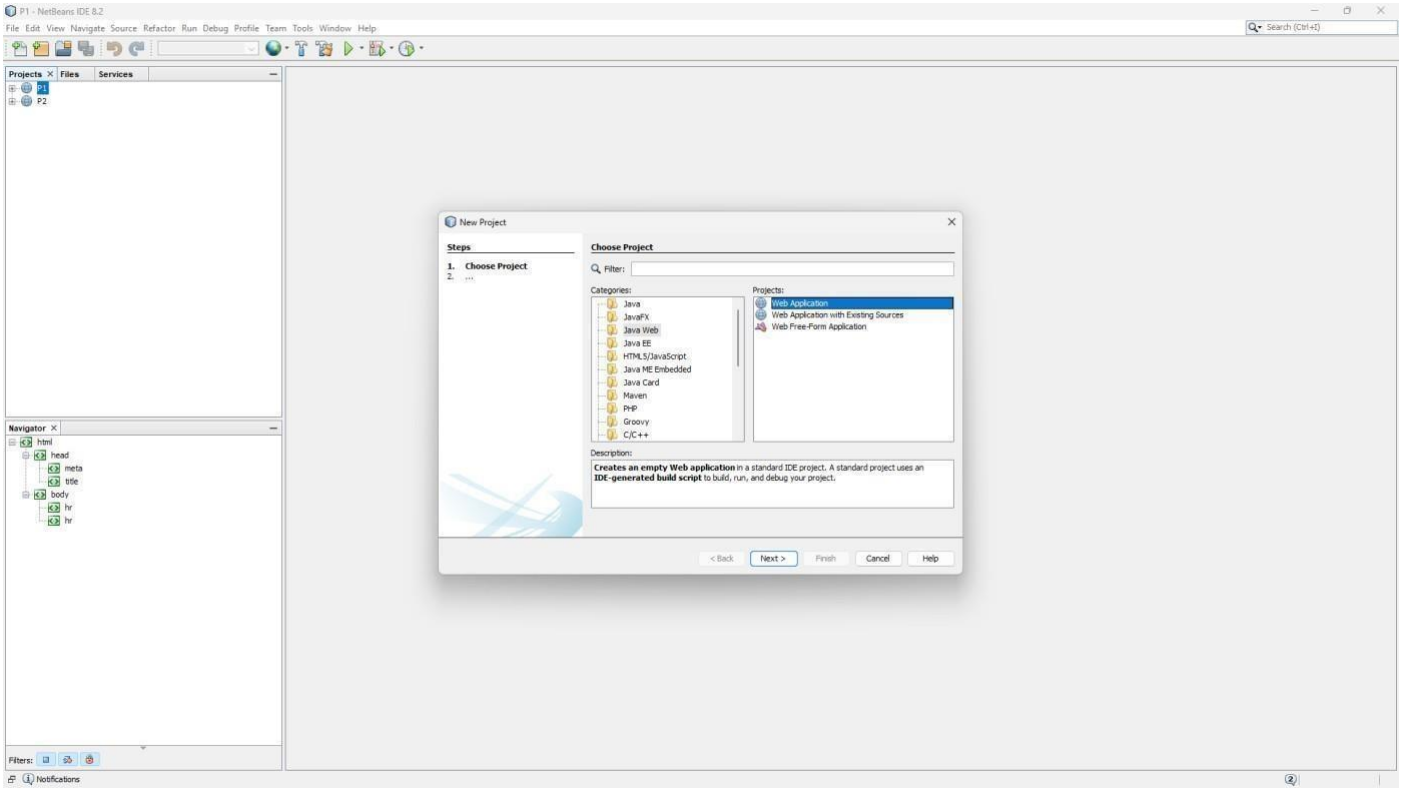
Output:



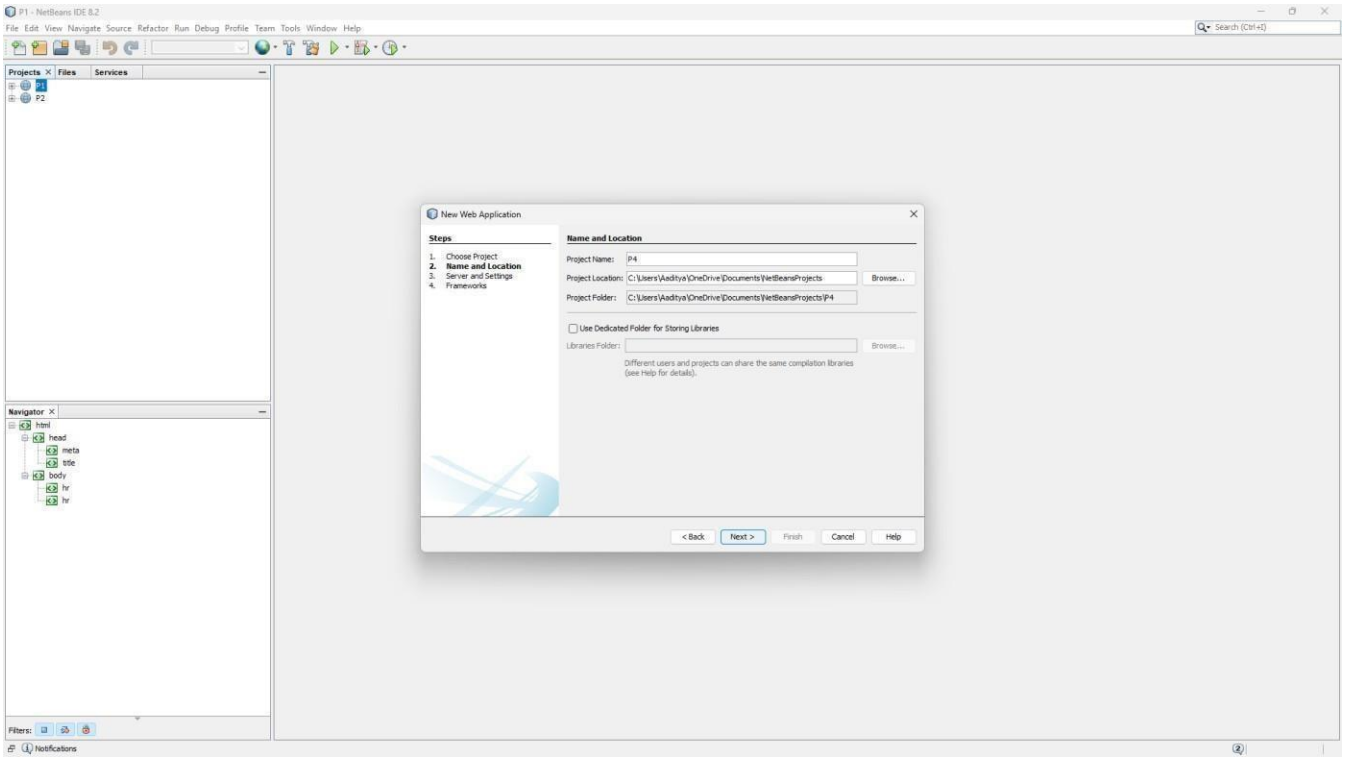
Practical: 4

Aim: Develop application to consume Google's search / Google's Map RESTful Web service.

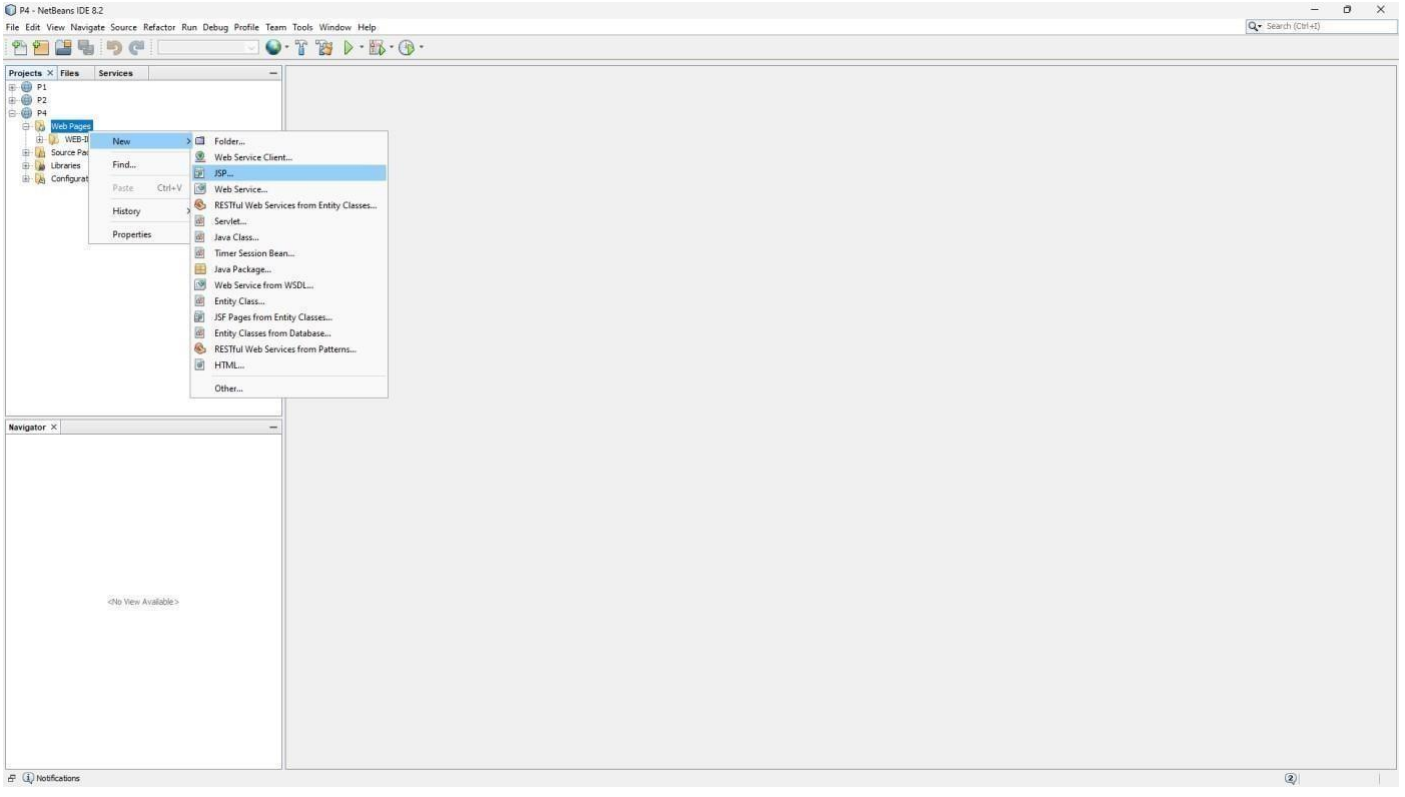
Step-1



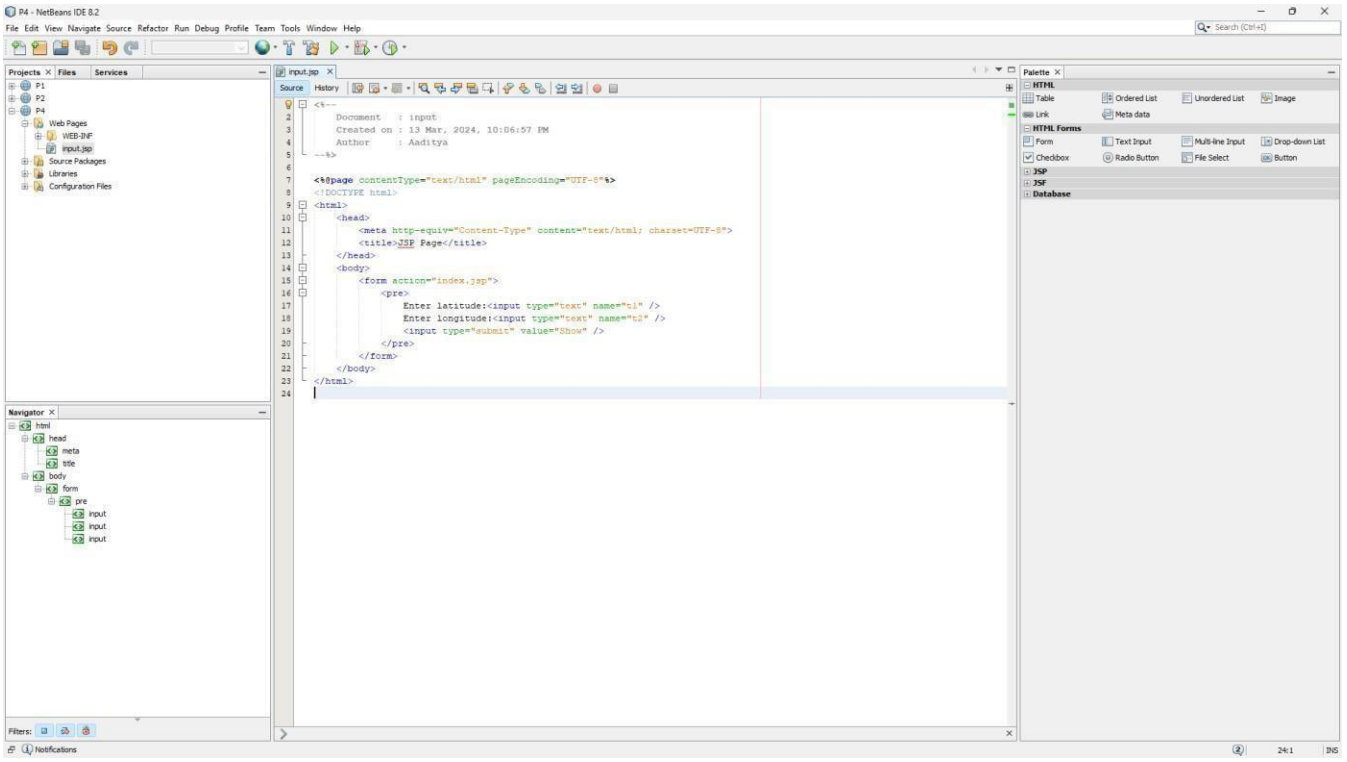
Step-2


P1 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
P1
P2
Navigator
html
head
meta
title
body
hr
hr
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: P4
Project Location: C:\Users\Aaditya\OneDrive\Documents\NetBeansProjects
Browse...
Project Folder: C:\Users\Aaditya\OneDrive\Documents\NetBeansProjects\P4
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries
(see Help for details).
< Back
Next >
Finish
Cancel
Help
Notifications

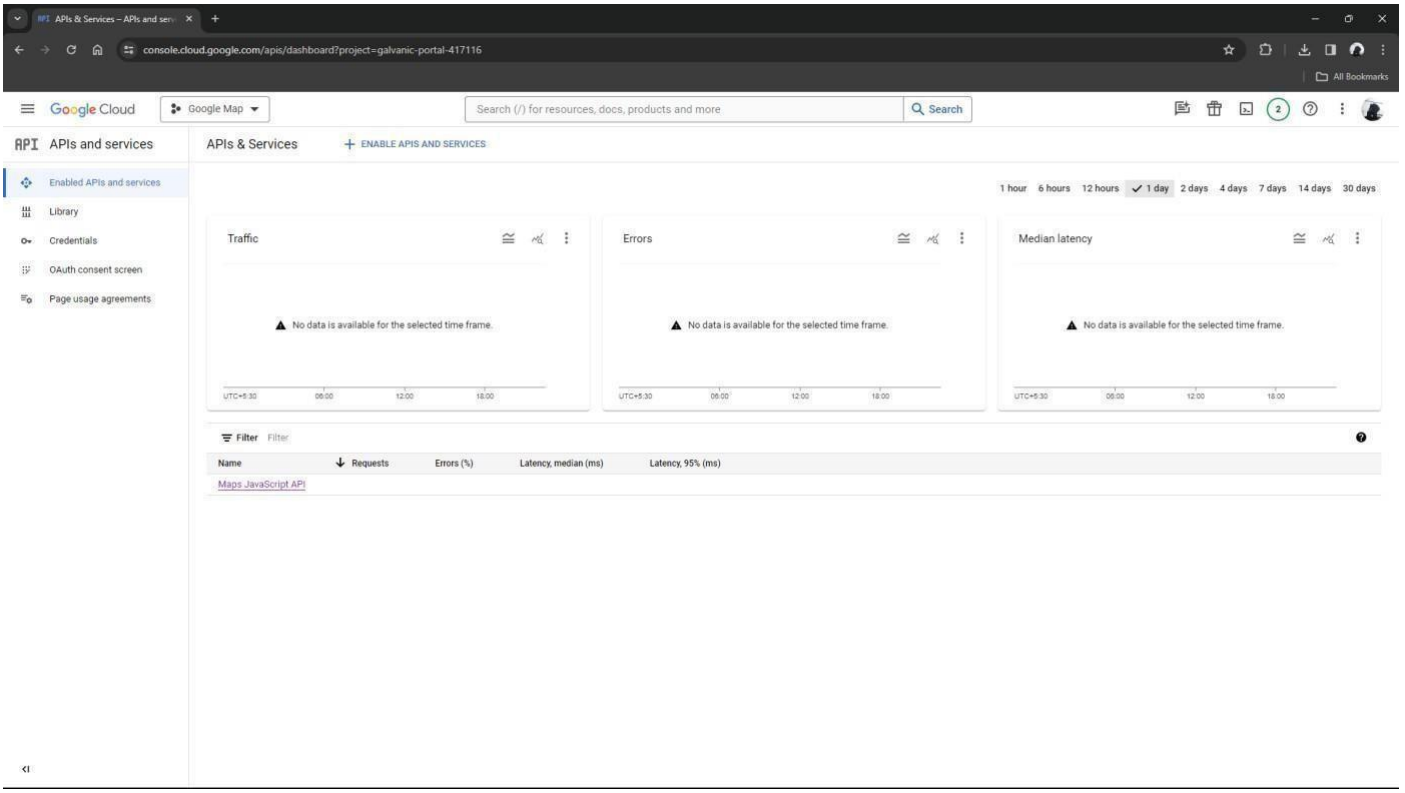
Step-3


P4 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
P1
P2
P4
Web Pages
Libraries
New
Find...
Paste Ctrl+V
History
Properties
Folder...
Web Service Client...
JSP...
Web Service...
RESTful Web Services from Entity Classes...
Servlet...
Java Class...
Timer Session Bean...
Java Package...
Web Service from WSDL...
Entity Class...
JSF Pages from Entity Classes...
Entity Classes from Database...
RESTful Web Services from Patterns...
HTML...
Other...
Navigator
<No View Available>
Notifications

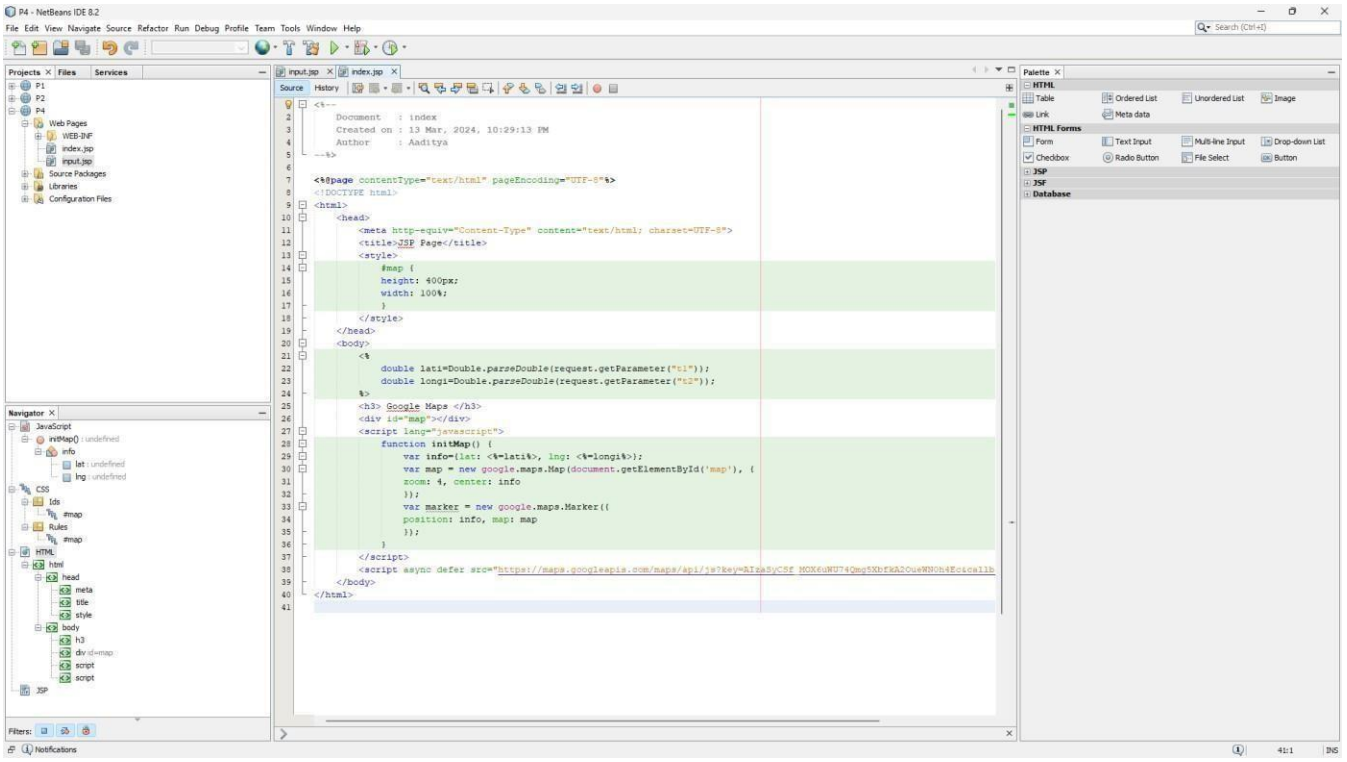
Step-4


P4 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
P1
P2
P4
Web Pages
WEB-INF
input.jsp
Source Packages
Libraries
Configuration Files
input.jsp
Source History
<%--
Document : input
Created on : 13 Mar, 2024, 10:06:57 PM
Author : Aaditya
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<form action="index.jsp">
<pre>
Enter latitude:<input type="text" name="t1" />
Enter longitude:<input type="text" name="t2" />
<input type="submit" value="Show" />
</pre>
</form>
</body>
</html>
Palette
HTML
Table
Ordered List
Unordered List
Image
Link
Meta data
HTML Forms
Form
Text Input
Multi-line Input
Drop-down List
Checkbox
Radio Button
File Select
Button
JSP
JSF
Database
Navigator
html
head
meta
title
body
form
pre
input
input
input
Filters:
Notifications
24:1
INS

Step-5

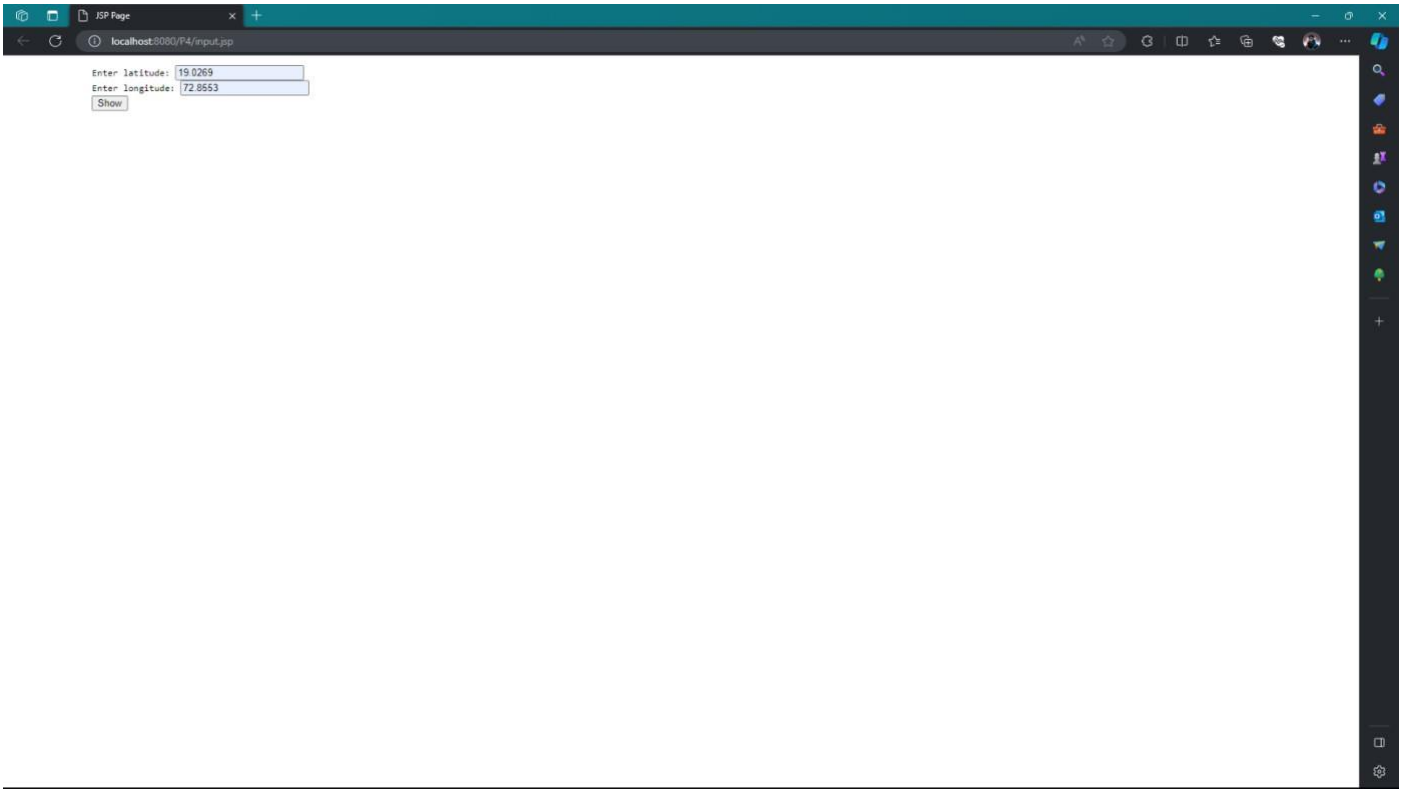

APIs & Services – APIs and ser
console.cloud.google.com/apis/dashboard?project=galvanic-portal-417116
All Bookmarks
Google Cloud
Google Map
Search (/) for resources, docs, products and more
Search
APIs and services
APIs & Services
ENABLE APIS AND SERVICES
Enabled APIs and services
Library
Credentials
OAuth consent screen
Page usage agreements
1 hour 6 hours 12 hours 1 day 2 days 4 days 7 days 14 days 30 days
Traffic
Errors
Median latency
No data is available for the selected time frame.
No data is available for the selected time frame.
No data is available for the selected time frame.
UTC+5:30 06:00 12:00 18:00
Filter Filter
Name Requests Errors (%) Latency, median (ms) Latency, 95% (ms)
Maps JavaScript API

Step-6

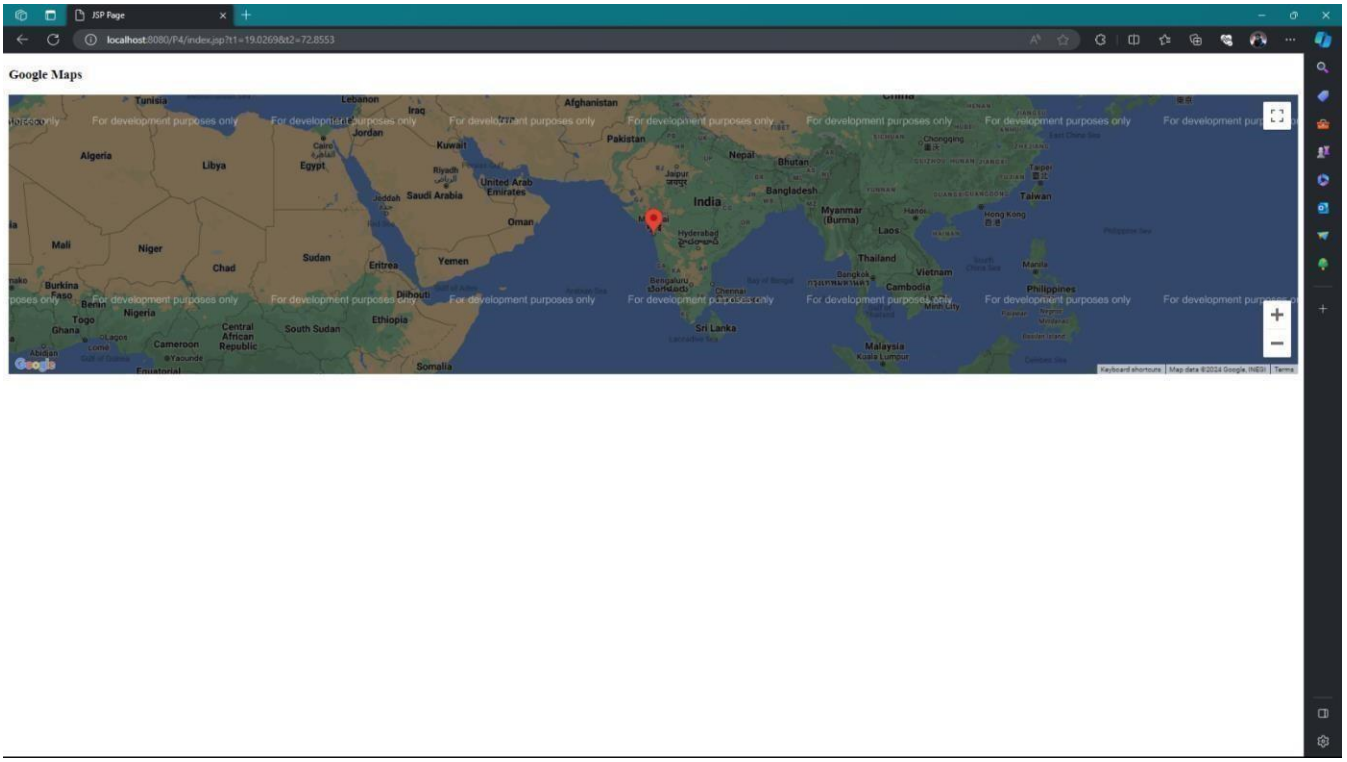

P4 - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
P1
P2
P4
Web Pages
WEB-INF
index.jsp
input.jsp
Source Packages
Libraries
Configuration Files
input.jsp
index.jsp
Source History
<%--
Document : index
Created on : 13 Mar, 2024, 10:29:13 PM
Author : Aaditya
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
<style>
#map {
height: 400px;
width: 100%;
}
</style>
</head>
<body>
<%
double lati=Double.parseDouble(request.getParameter("t1"));
double longi=Double.parseDouble(request.getParameter("t2"));
%>
<h3> Google Maps </h3>

<script lang="javascript">
function initMap() {
var info={lat: <%=lati%>, lng: <%=longi%>};
var map = new google.maps.Map(document.getElementById('map'), {
zoom: 4, center: info
});
var marker = new google.maps.Marker({
position: info, map: map
});
}
</script>
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC5f MOX6uNU74Qmg5XbfkA2OueWNOh4Eci&callb
</body>
</html>
Palette
HTML
Table
Ordered List
Unordered List
Image
Link
Meta data
HTML Forms
Form
Text Input
Multi-line Input
Drop-down List
Checkbox
Radio Button
File Select
Button
JSP
JSF
Database
Navigator
JavaScript
initMap() : undefined
info
lat : undefined
lng : undefined
CSS
Ids
#map
Rules
#map
HTML
html
head
meta
title
style
body
h3
div id=map
script
script
JSP
Filters:
Notifications
41:1
INS

Step-7


JSP Page
localhost:8080/P4/input.jsp
Enter latitude: 19.0269
Enter longitude: 72.8553
Show

Step-8

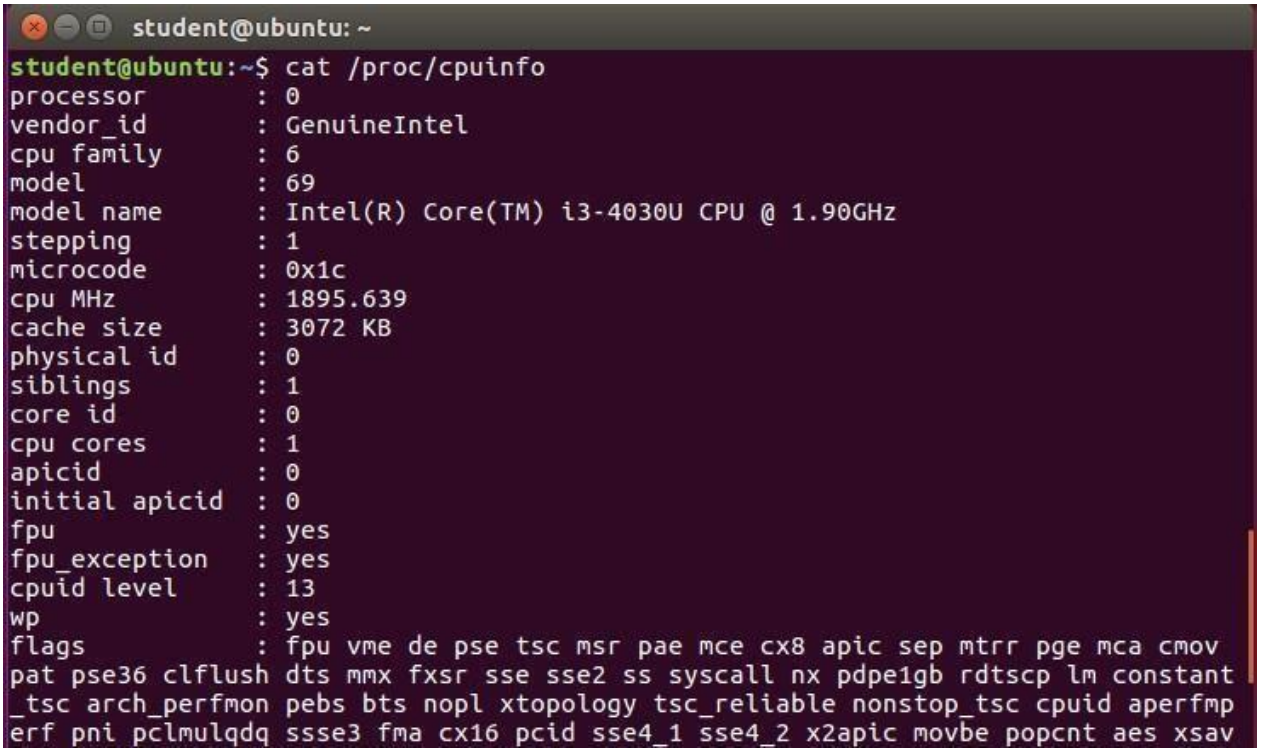


‘

Practical: 5

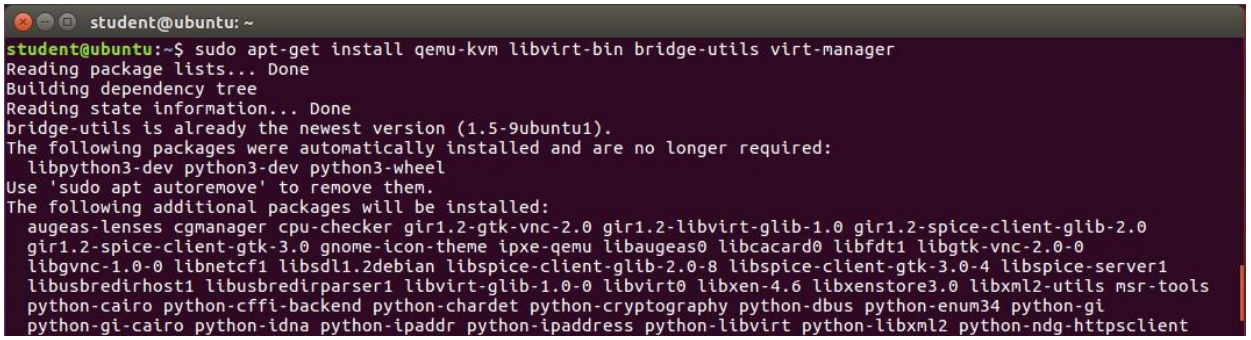
Aim: Installation and Configuration of Virtualization using KVM.

Step-1 : $cat/proc/cpuinfo



Step-2 : $sudo apt-get update

Step 3 : $sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager



Step 4 : $sudo adduser tycs

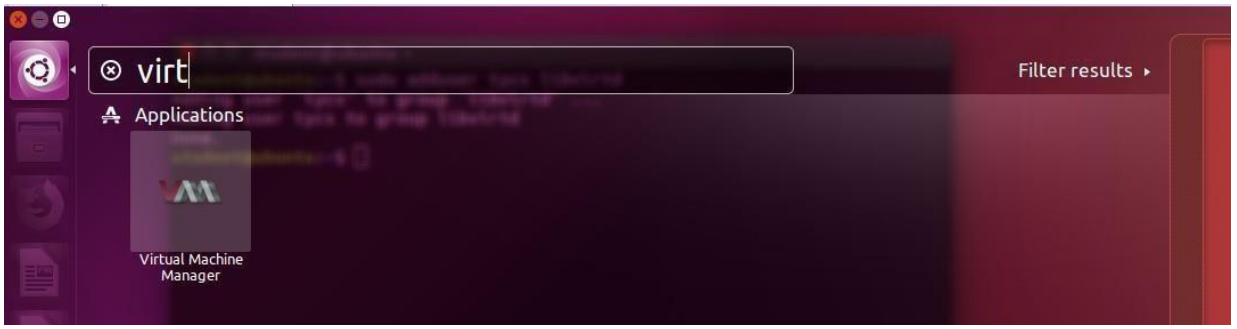
```
student@ubuntu: ~
student@ubuntu:~$ sudo adduser tycs
Adding user `tycs' ...
Adding new group `tycs' (1002) ...
Adding new user `tycs' (1002) with group `tycs' ...
Creating home directory `/home/tycs' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for tycs
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
student@ubuntu:~$
```

Step 5 : $sudo adduser tycs libvirtd

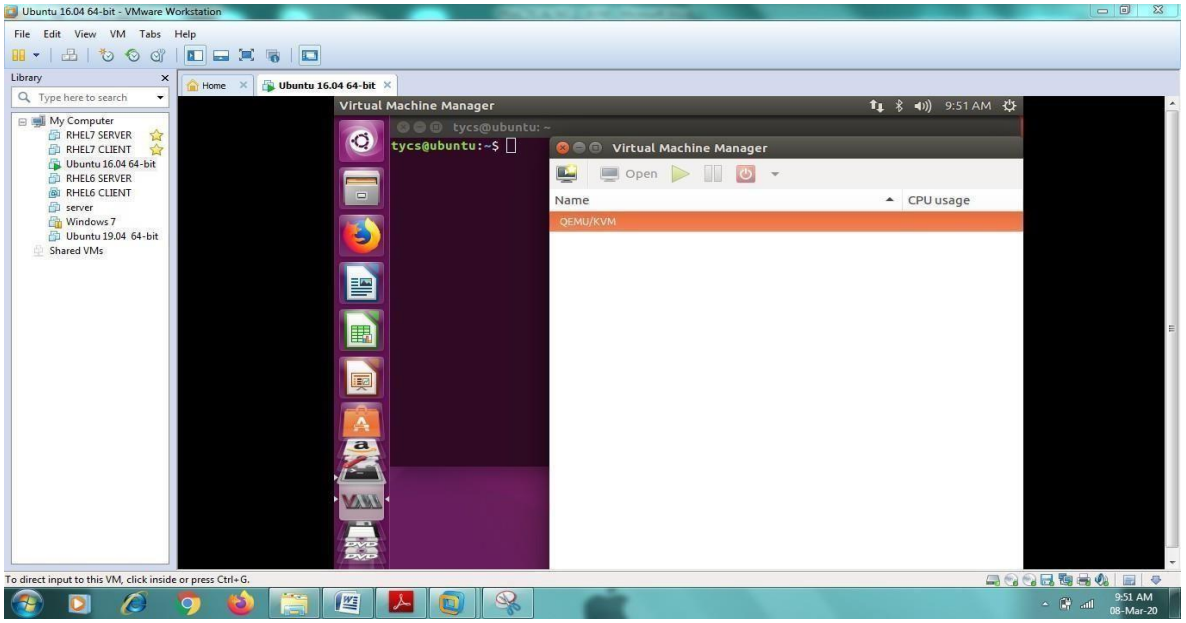
```
student@ubuntu: ~
student@ubuntu:~$ sudo adduser tycs libvirtd
Adding user `tycs' to group `libvirtd' ...
Adding user tycs to group libvirtd
Done.
student@ubuntu:~$
```

Step 6 : Go to search and Type virtual. It will show “Virtual

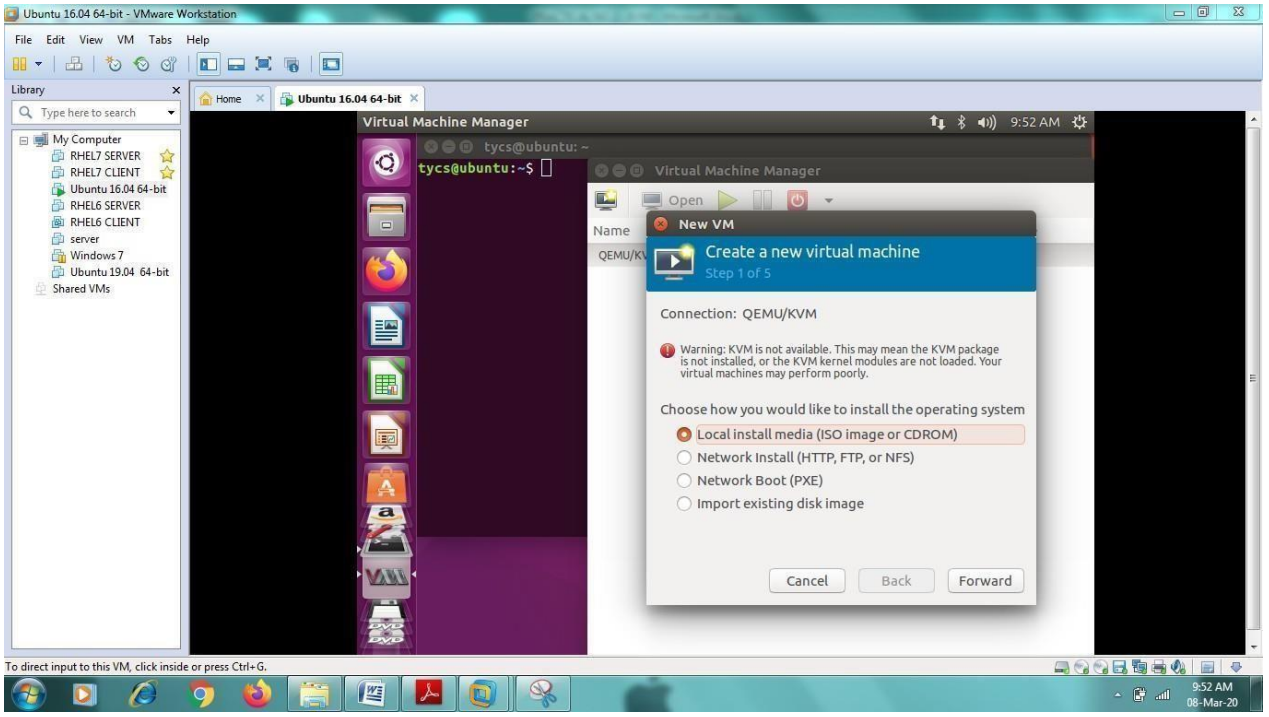
Machine Manager”



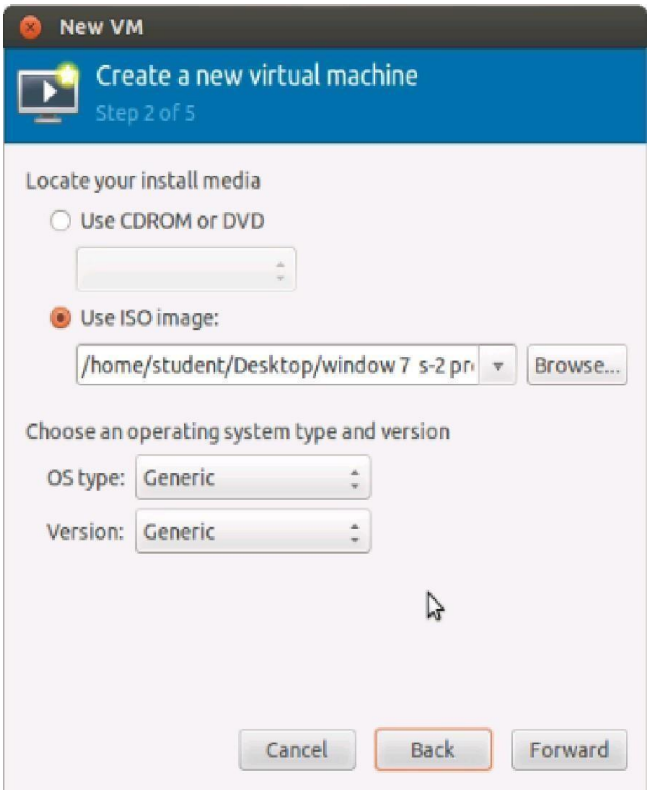
Or type commnd as $ virt-manager

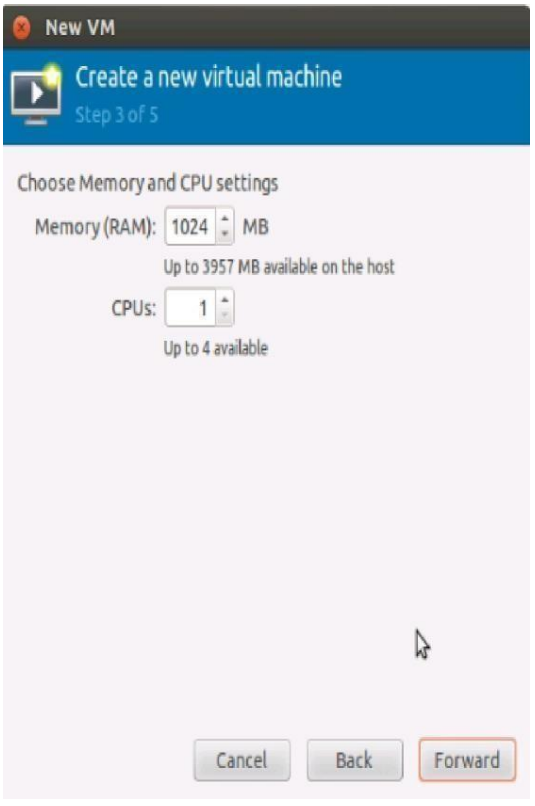


Step 7 : Click on Create Button



Step 8 : Create a new virtual machine and select iso file and click on forward.



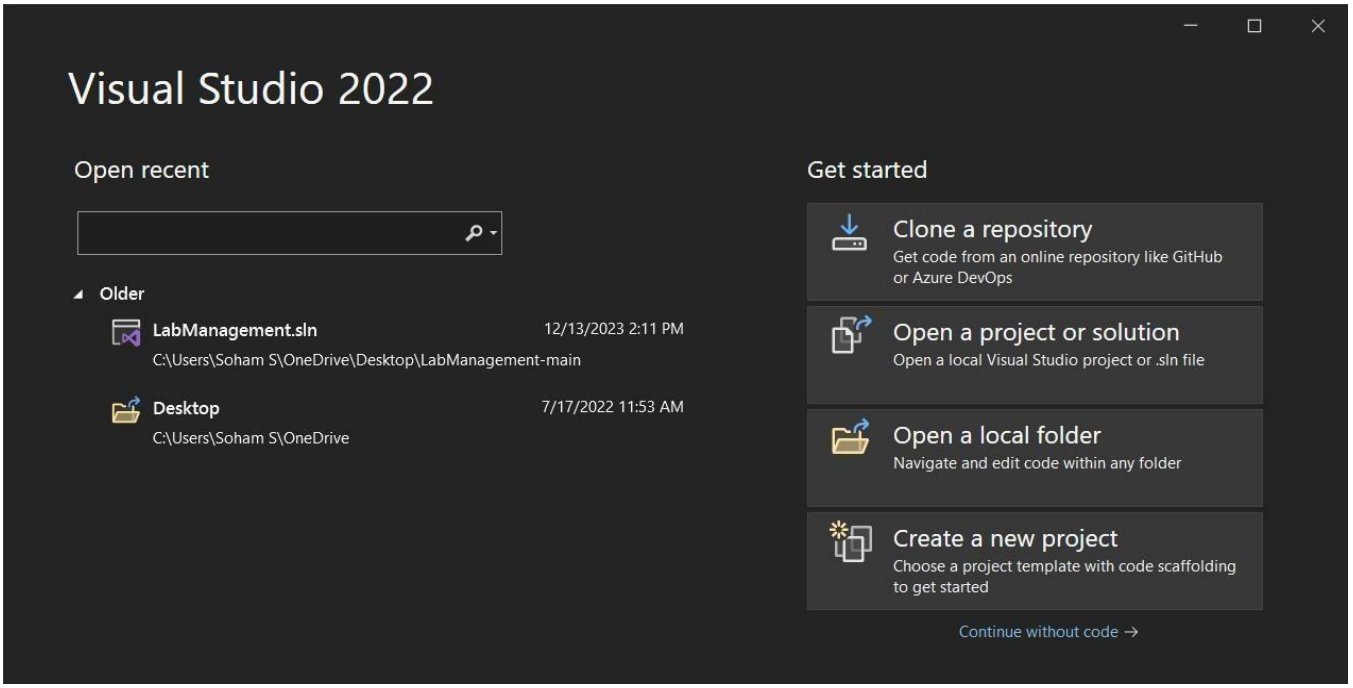


Provide RAM, disk storage and then select network and click on finish

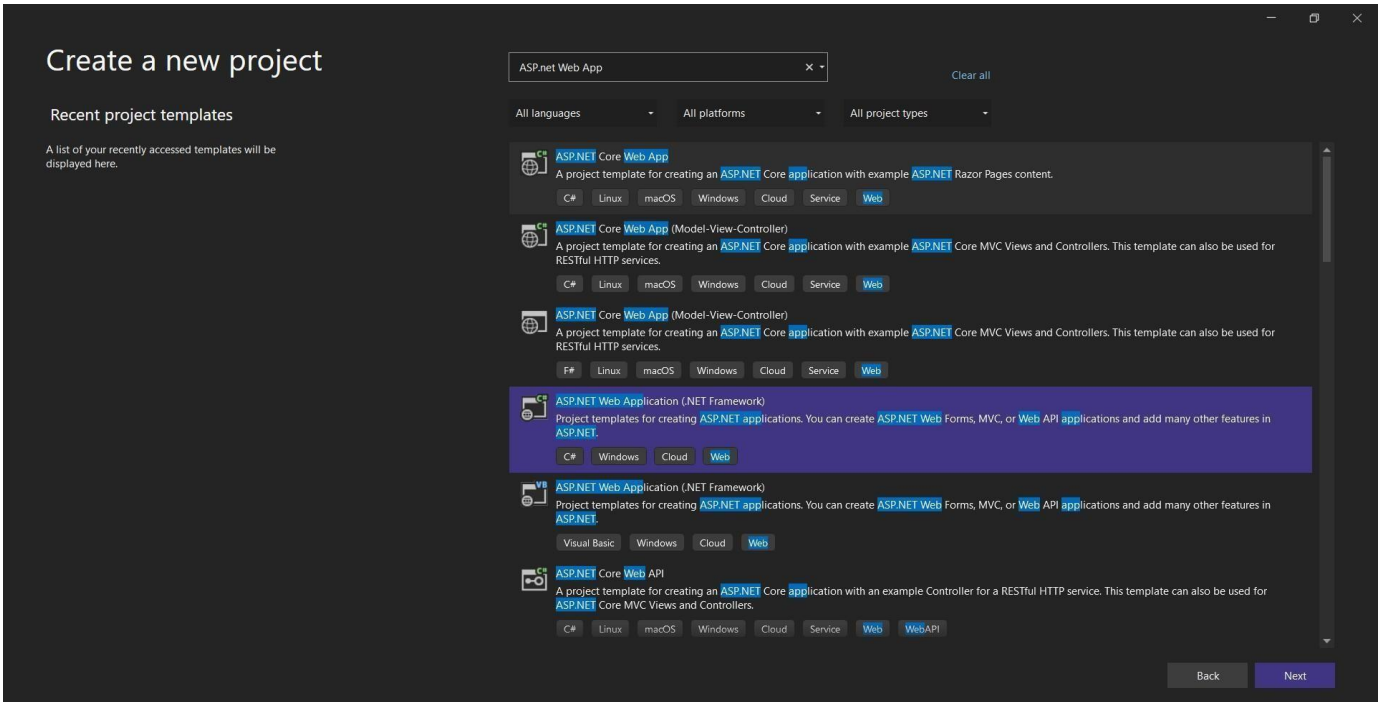
Practical 6

Aim: Develop application to download image/video from server or upload image/video to server using MTOM techniques

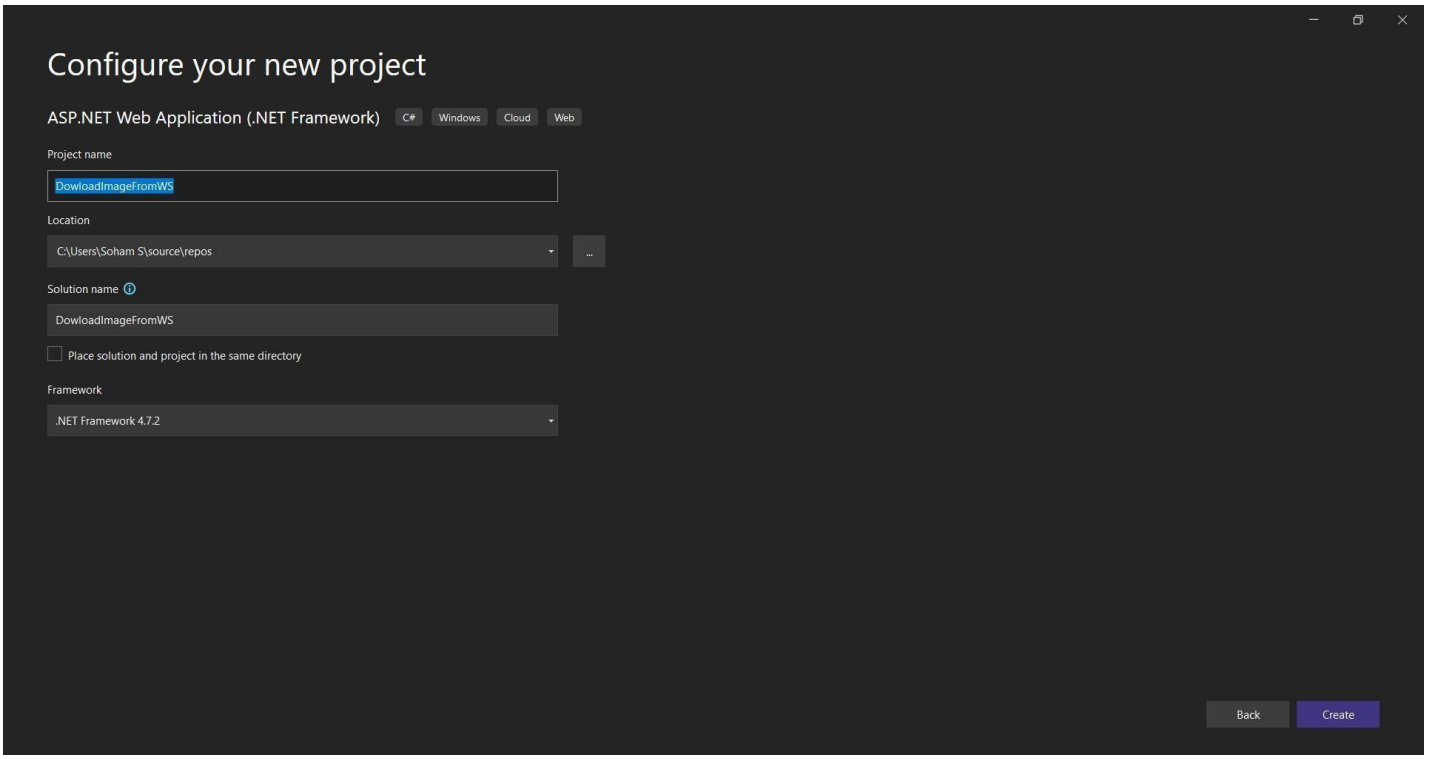
Step 1:- Open Visual Studio 2022 and click on Create a new project..



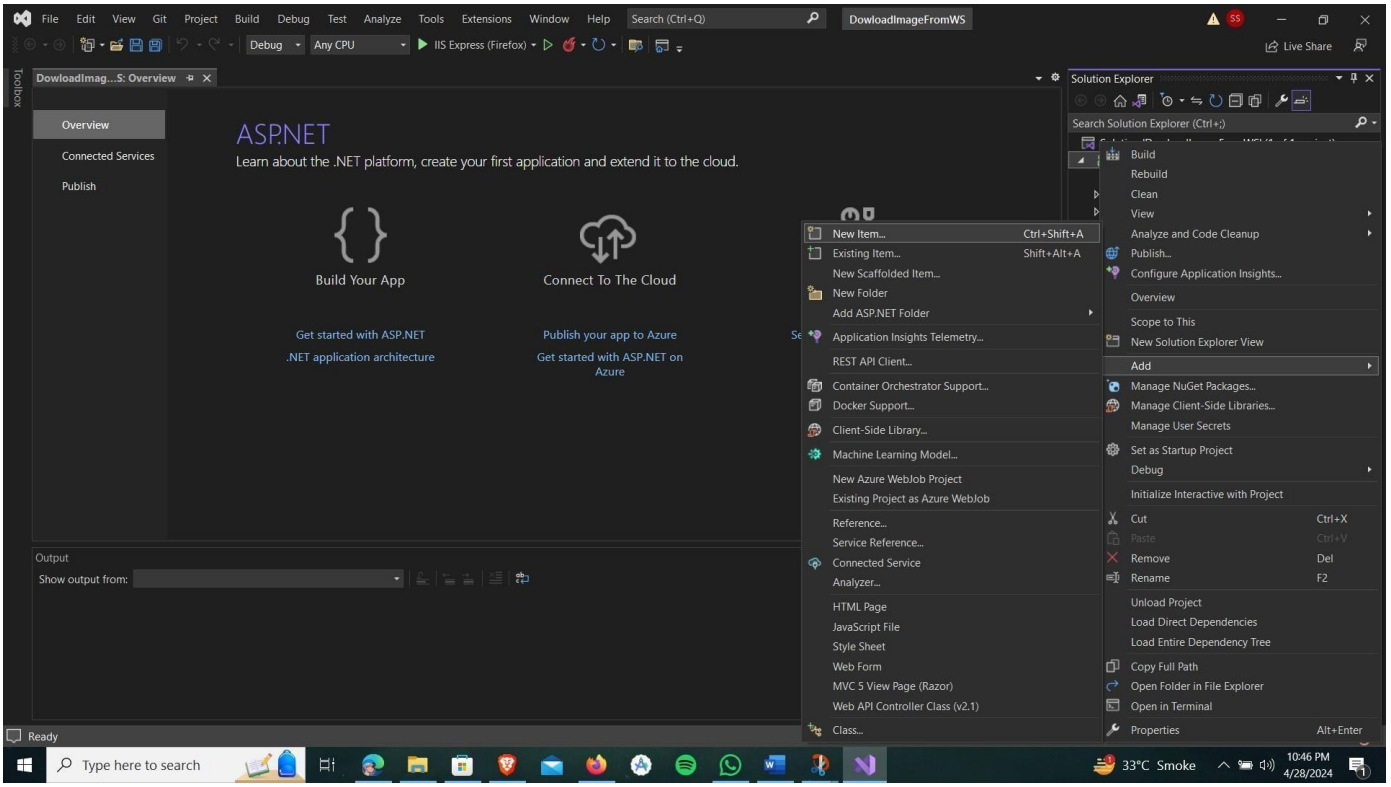
Step 2:- Choose ASP.net C++ framework option


Create a new project
Recent project templates
A list of your recently accessed templates will be displayed here.
ASP.net Web App
Clear all
All languages
All platforms
All project types
ASP.NET Core Web App
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
C# Linux macOS Windows Cloud Service Web
ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web
ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
F# Linux macOS Windows Cloud Service Web
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Cloud Web
ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web WebAPI
Back
Next

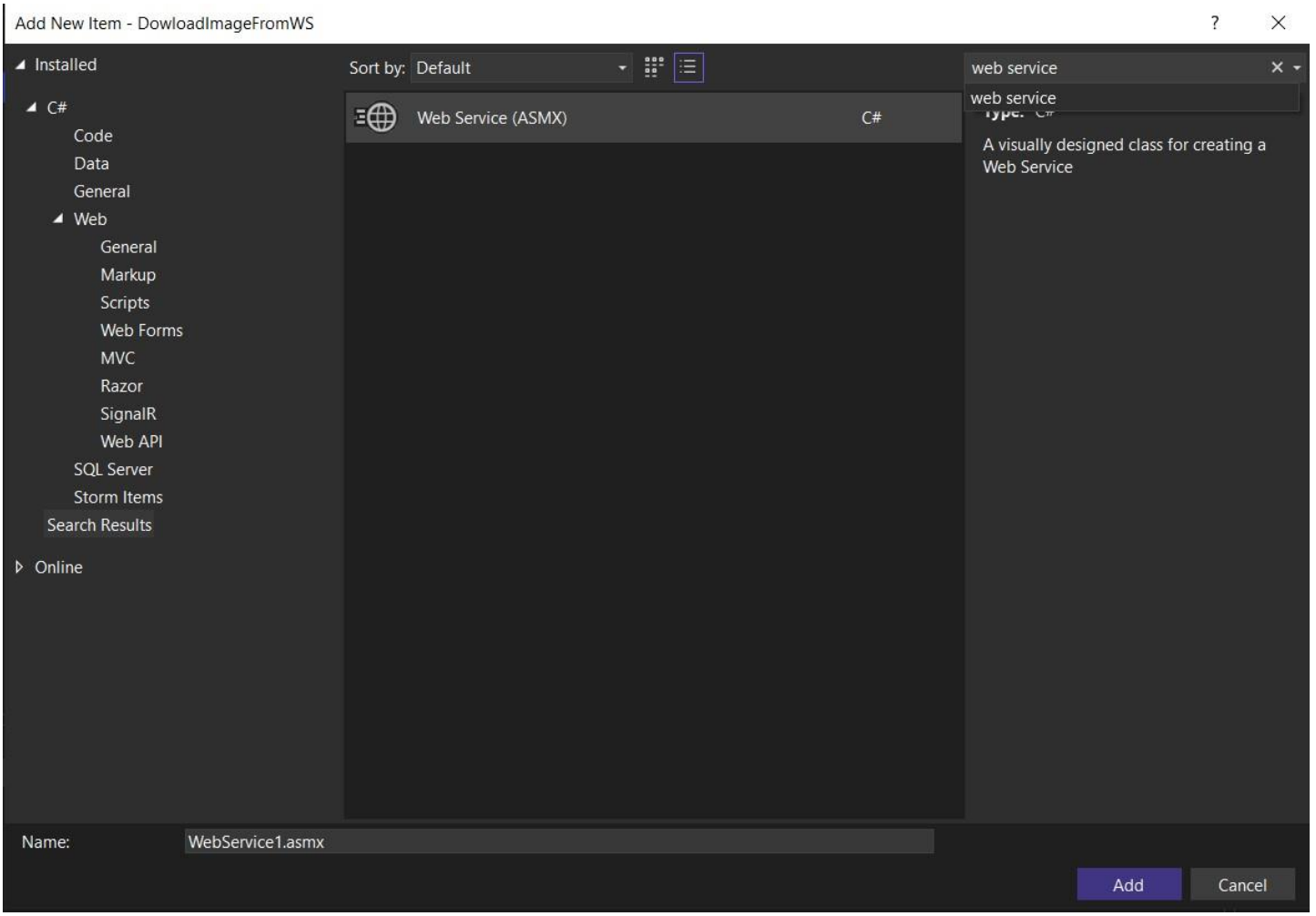
Step 3 :- Give name to your project and click create



Step 4 :- In next , in the right corner project resource file was there ,click right mouse button to get an menu to add web services from there



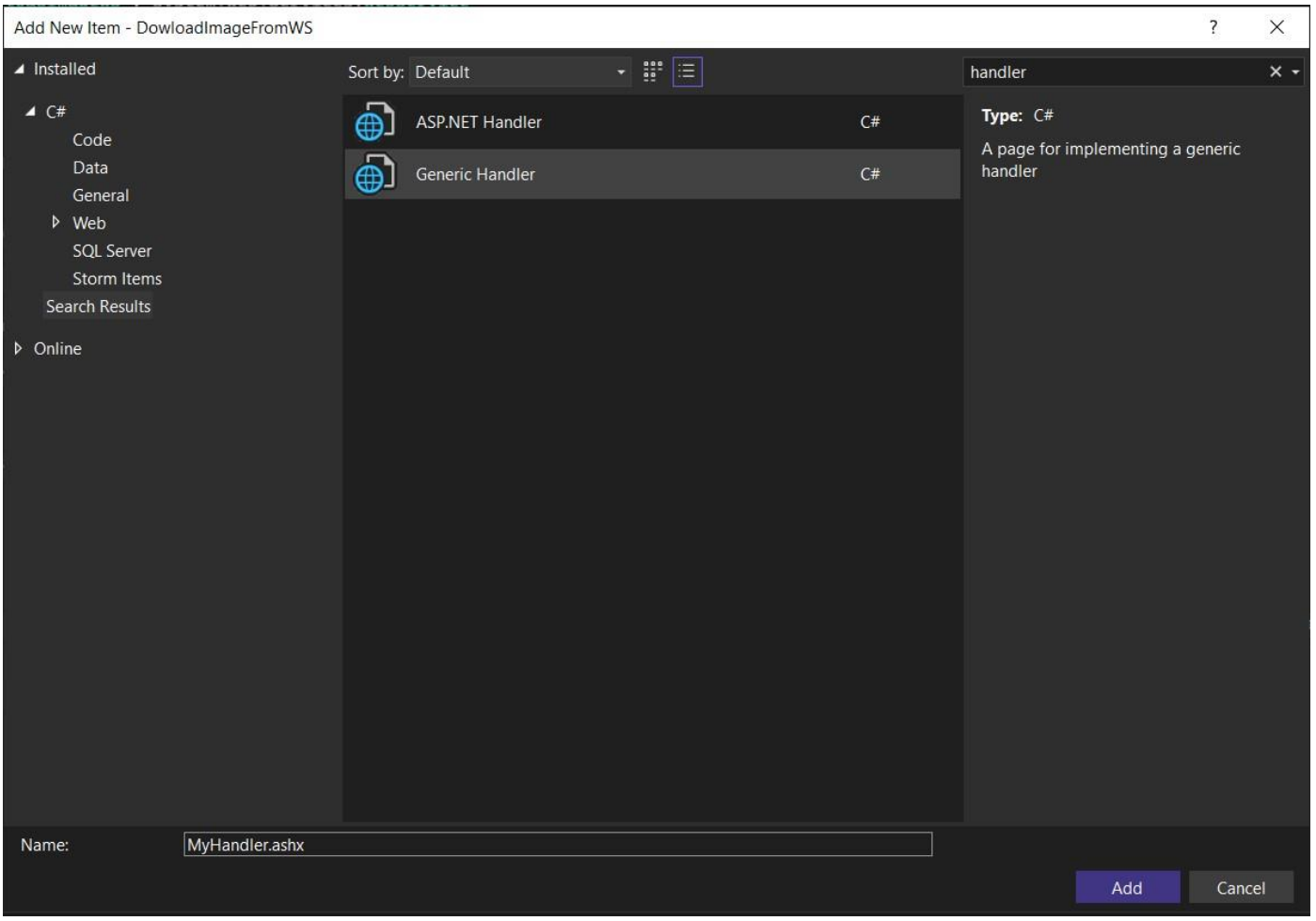
Step 5 : Search for Web Service (ASMX) and Add them..



Step 6 : Write a code in the web service window..

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Linq; using
System.Web;
using System.Web.Services;

namespace DowloadImageFromWS
{
    /// <summary>
    /// Summary description for DownloadImageWS
    /// </summary>
    [WebService( amespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = Wsiflrofiles.Basicflrofile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASfl. ET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class DownloadImageWS : System.Web.Services.WebService
    {

        [WebMethod]
        public string elloWorld()
        {
            return " Hello World";
        }


        [WebMethod, Description("Get Image Content")] public byte[]
        GetImageFile(String file ame)
        { if(System.IO.File.Exists(Server.Mapflath("~/Images/") + file ame))
            {
                return System.IO.File.ReadAllBytes(Server.Mapflath("~/Images/") + file ame);
            }

            else {
                return new byte[] { 0 };
            }
        }
    } }
```

Step 7 : Follow the Step 4 to add a handler class into the program and give them a name as “MyHandler”

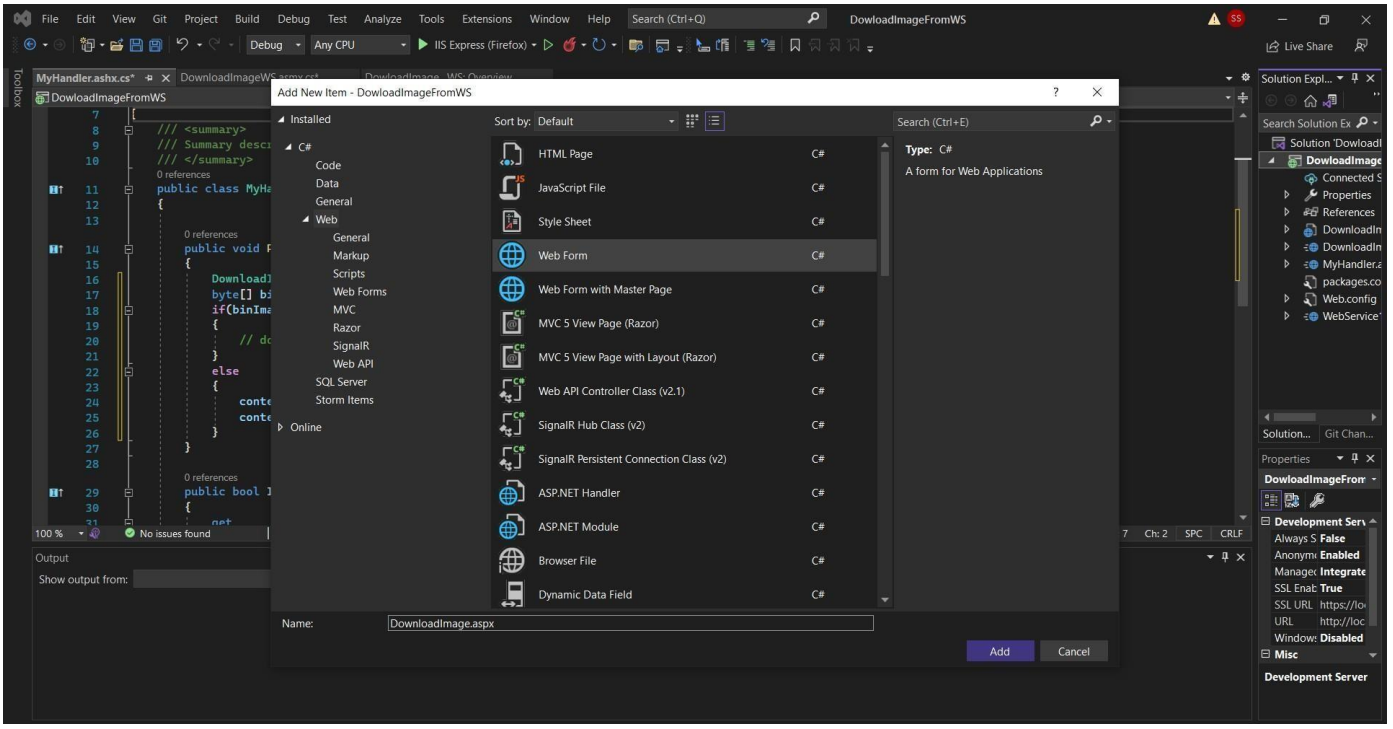


Step 8: Write the code into the Handler class

```
using System;
using System.Collections.Generic; using
System.Linq;
using System.Web;

namespace DowloadImageFromWS
{
    /// <summary>
    /// Summary description for My andler
    /// </summary>
    public class My andler : I ttp andler
    {

        public void flrocessRequest( ttpContext context)
```

```
        {
            DownloadImageWS ws = new DownloadImageWS();
            byte[] binImage = ws.GetImageFile(context.Request["file ame"]); if(binImage.Length == 1)
            {
                // do nothing
            }
            else
            {          context.Response.ContentType = "image/jpeg";
                context.Response.BinaryWrite(binImage);
            }
        }


        public bool IsReusable
        { get
            {
                return false;
            }
        }
    }
}
```

Step 9 : Follow the same step 4 to add a web form into the program



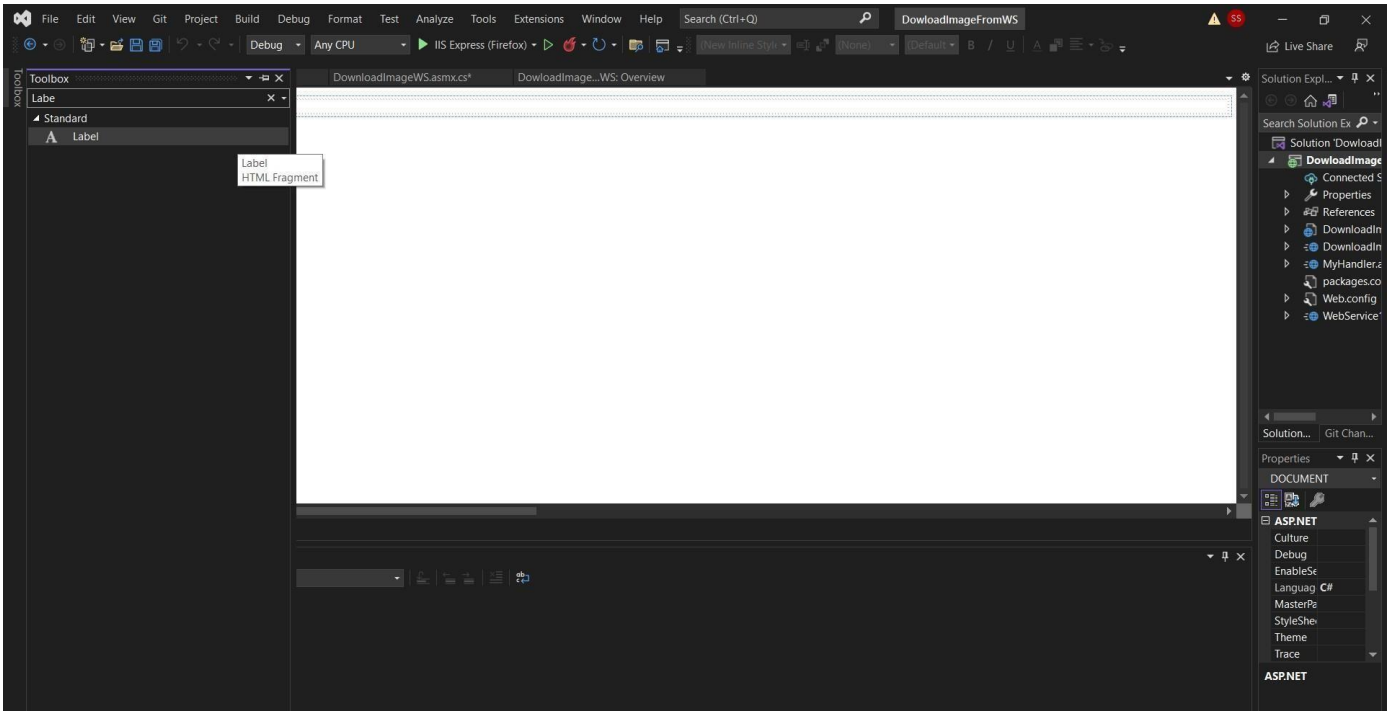
Step 10 :- Write the below code into the Web form

```
<%@ flage Language="C " AutoEventWireup="true" CodeBehind="DownloadImage.aspx.cs"
Inherits="DowloadImageFromWS.DownloadImage1" %>

<!DOCTYflE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
```

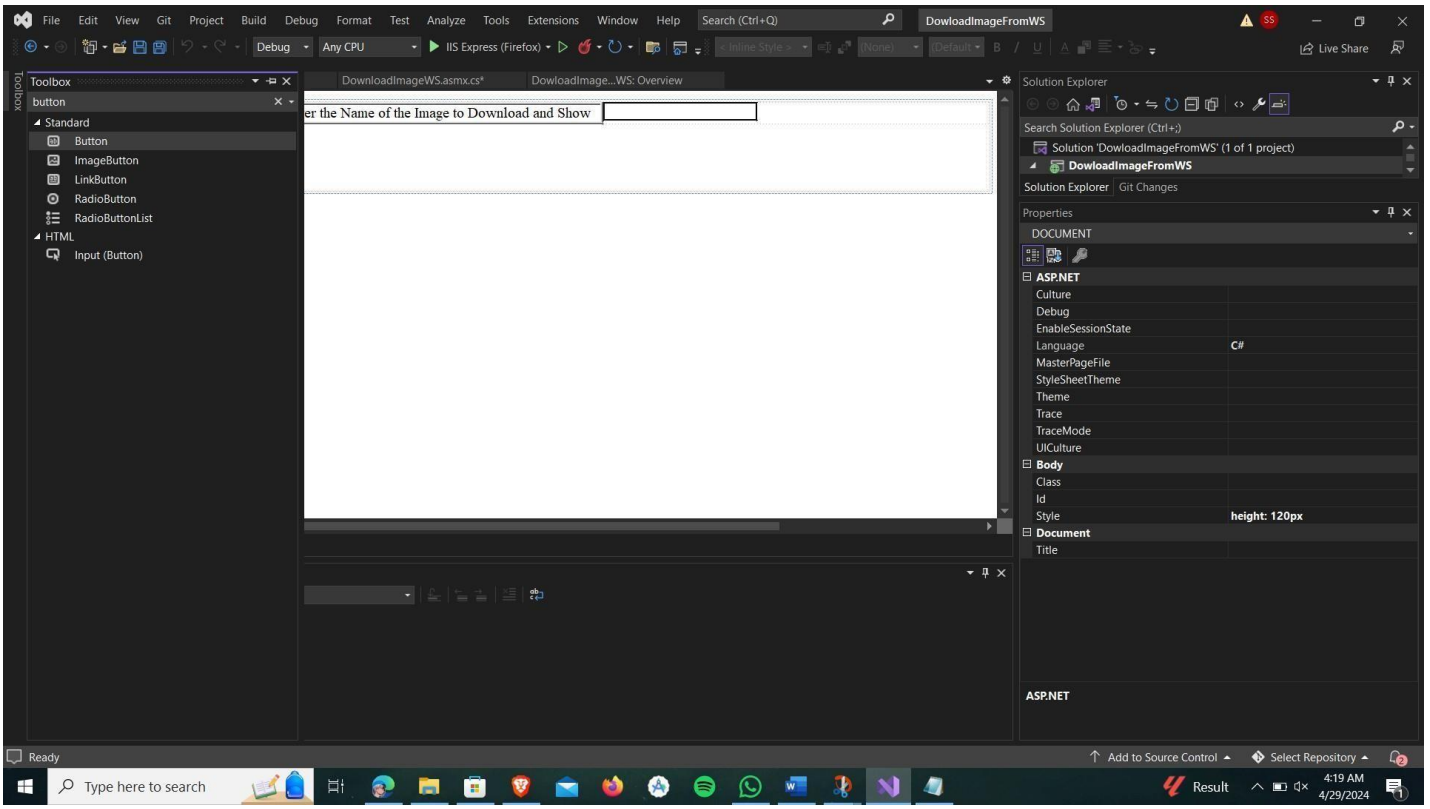
```
    <style    type="text/css">    #
        form1 { height: 186px;
        }
    </style>
</head>
<body id="btn1" style="height: 120px">
    <form id="form1" runat="server"> <center>
                                    <asp:Label ID="lbl1" runat="server" BorderStyle="Ridge" Text="Enter the ame of the
Image to Download and Show" Width="439px"></asp:Label>
            <asp:TextBox ID="txt1" runat="server" BorderStyle="Groove" Width="200px"></asp:TextBox>
        </center>
        <p> finbsp;</p>
        <asp:Button ID="btn1" runat="server" BorderStyle="Ridge" eight="38px"      H
OnClick="btn1_Click" style="margin-left: 494px" Text="Download and Show" Width="412px" />
        <p>
            <asp:Image ID="Image1" runat="server" eight="30px" style="margin-left: 671px" Width="27px" />
        </p>
    </form>
</body>
</html>
```

Step 11 : Click on design button to get an design interface and draw the interface as show below

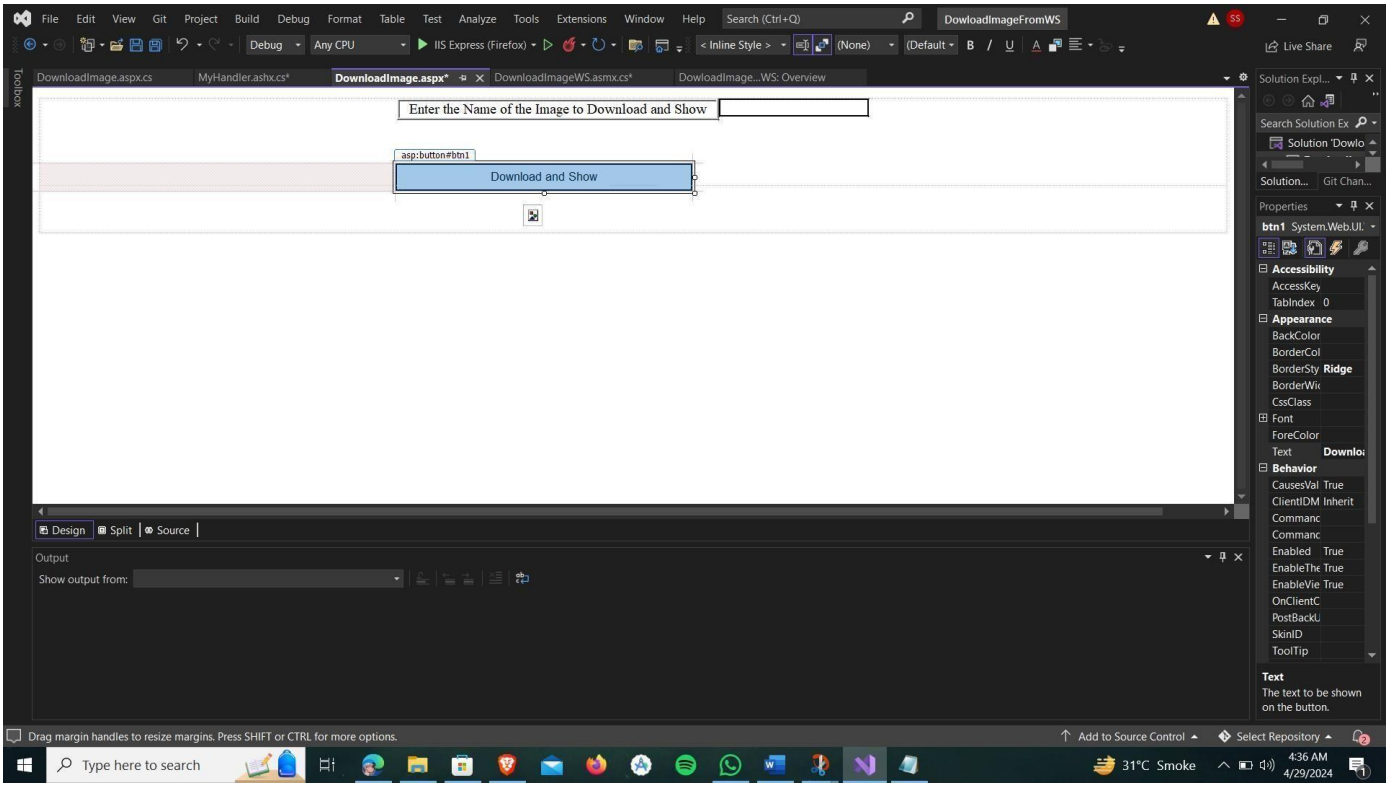
Click on toolbox at left corner and search for label



Search for textbox and button and drag them in to design area



Serach for Image and final interface will look like this…

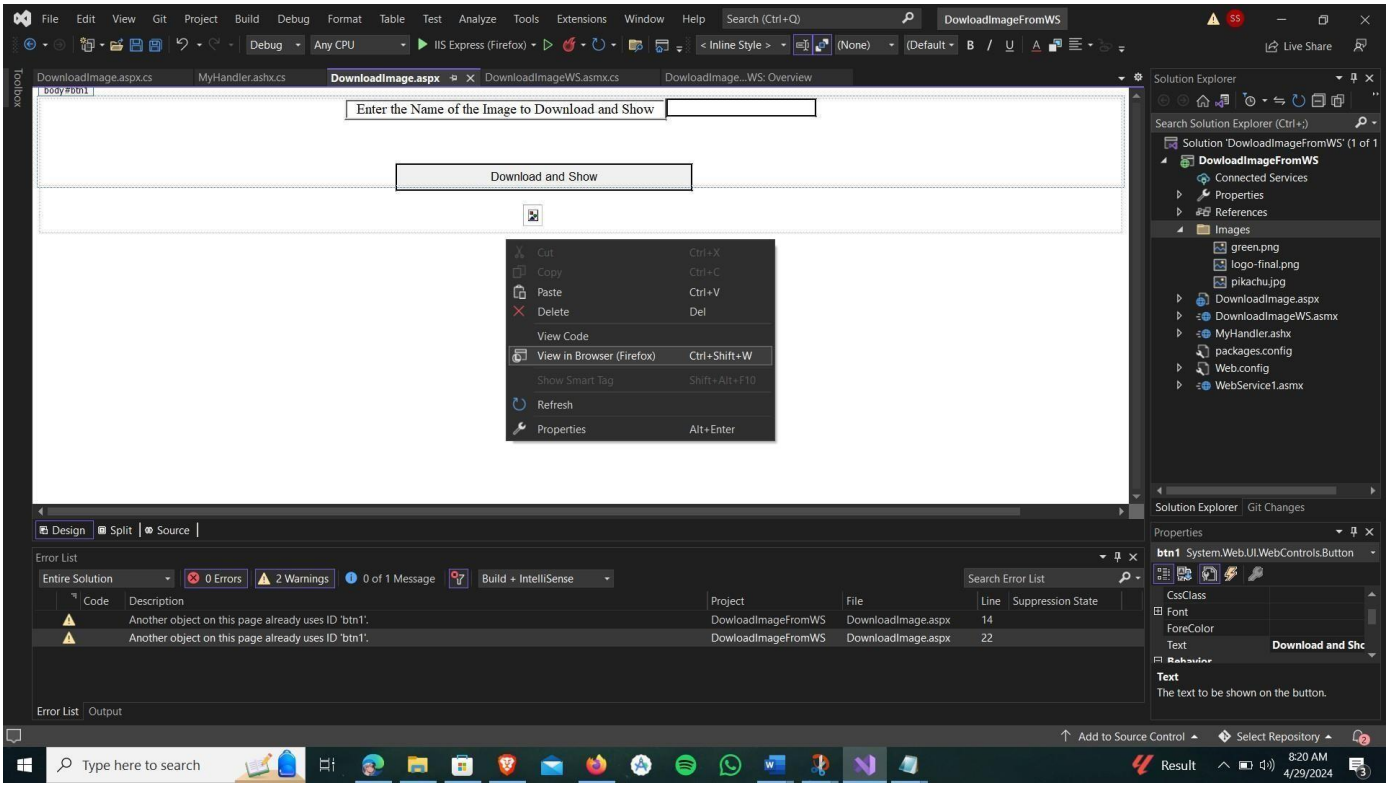


Step 12 : double click on button to get access to its source code and enter the below code there

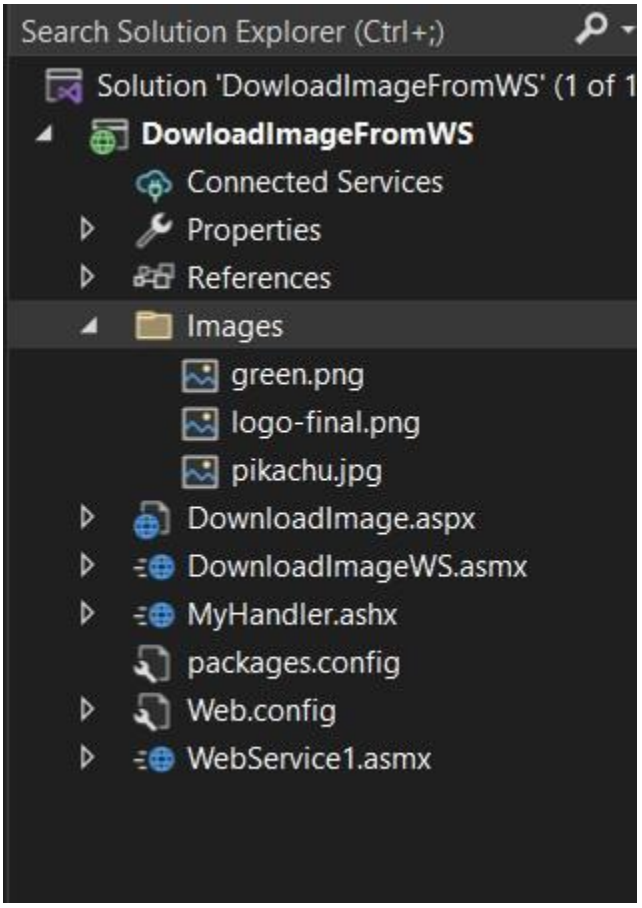
```
using System;
```

```
using System.Collections.Generic; using
System.Linq; using System.Web; using
System.Web.UI;
using System.Web.UI.WebControls;

namespace DowloadImageFromWS
{
    public partial class DownloadImage1 : System.Web.UI.flage
    {
        protected void flage_Load(object sender, EventArgs e)
        {

        }

        protected void btn1_Click(object sender, EventArgs e)
        {
            Image1.ImageUrl = "~/My andlerHashxofile ame=" + txt1.Text; Response.Write("Downloading of Image is
            done");


        }
    }

}
```

Step 13 : right click on design interface and test the interface in the web browser



Step 14 : Before going to browser create a file folder into the source file. As the name is “Images” and paste local images

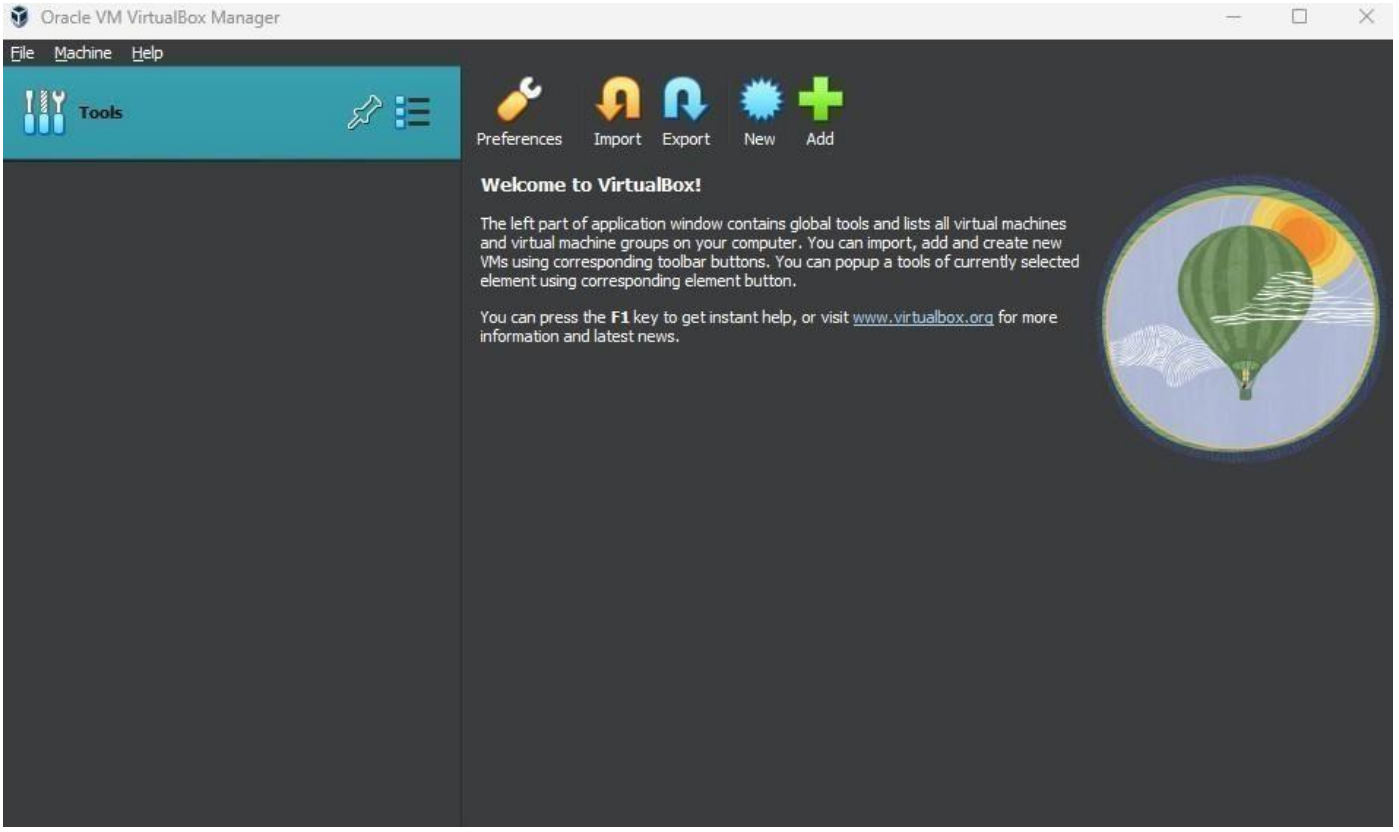


Step 15 : Access the image by giving the name of the image name

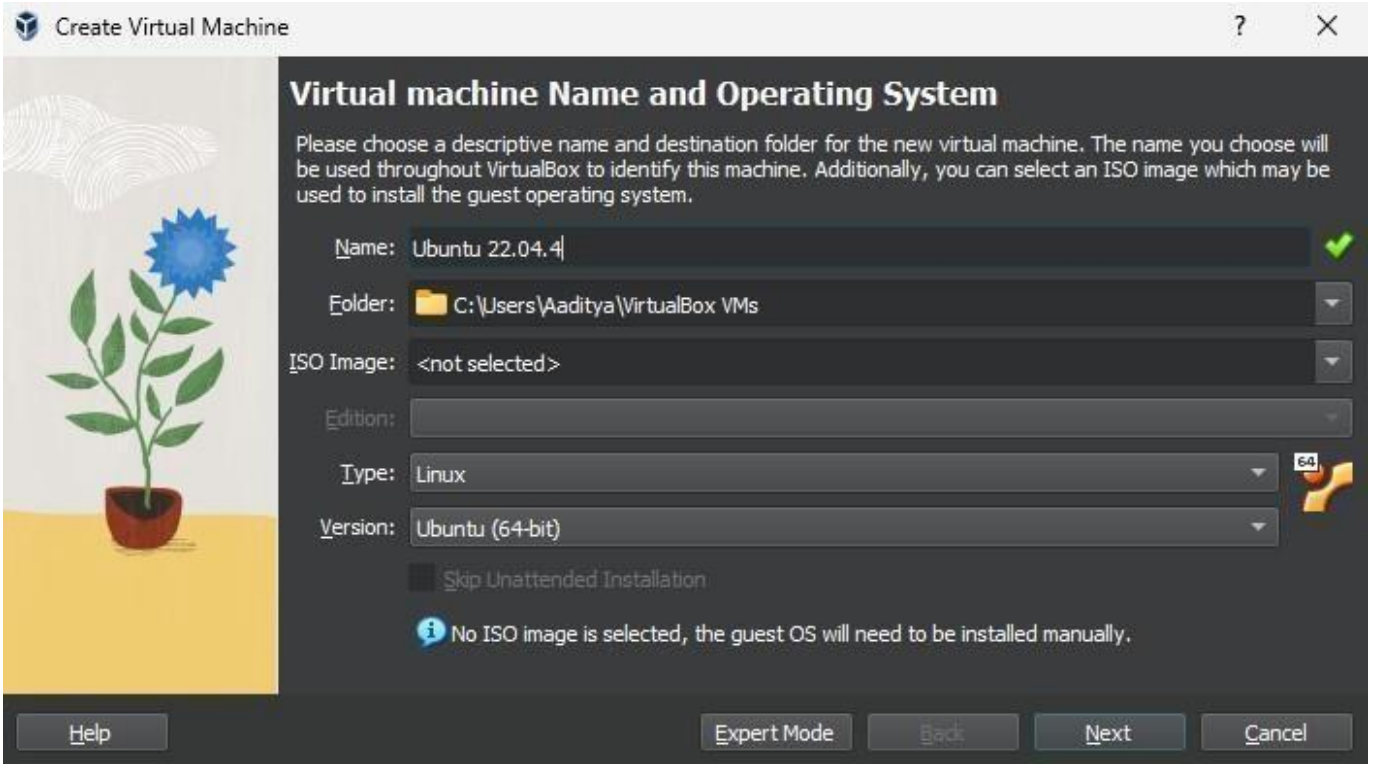
Practical 7

Aim: Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service(IaaS), Storage

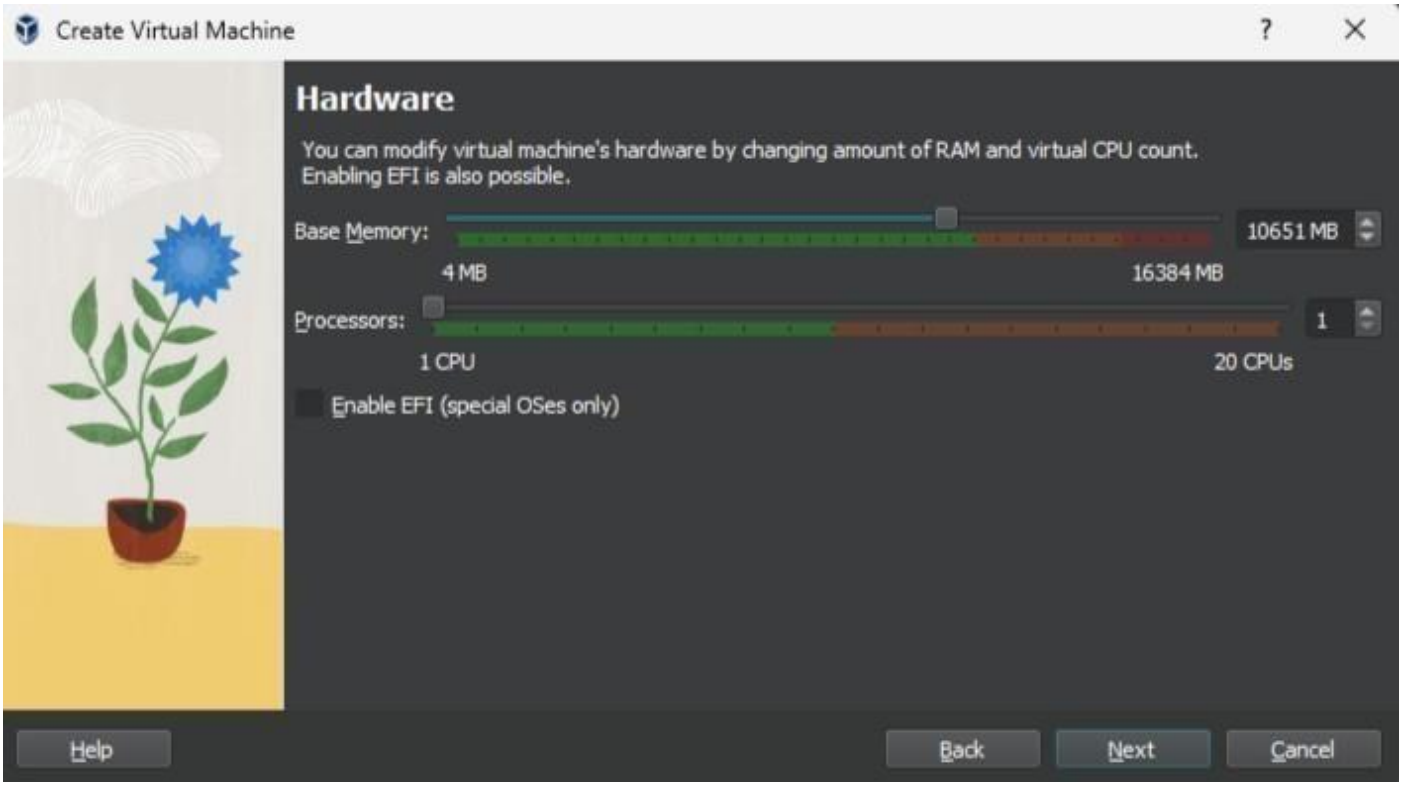
Step-1



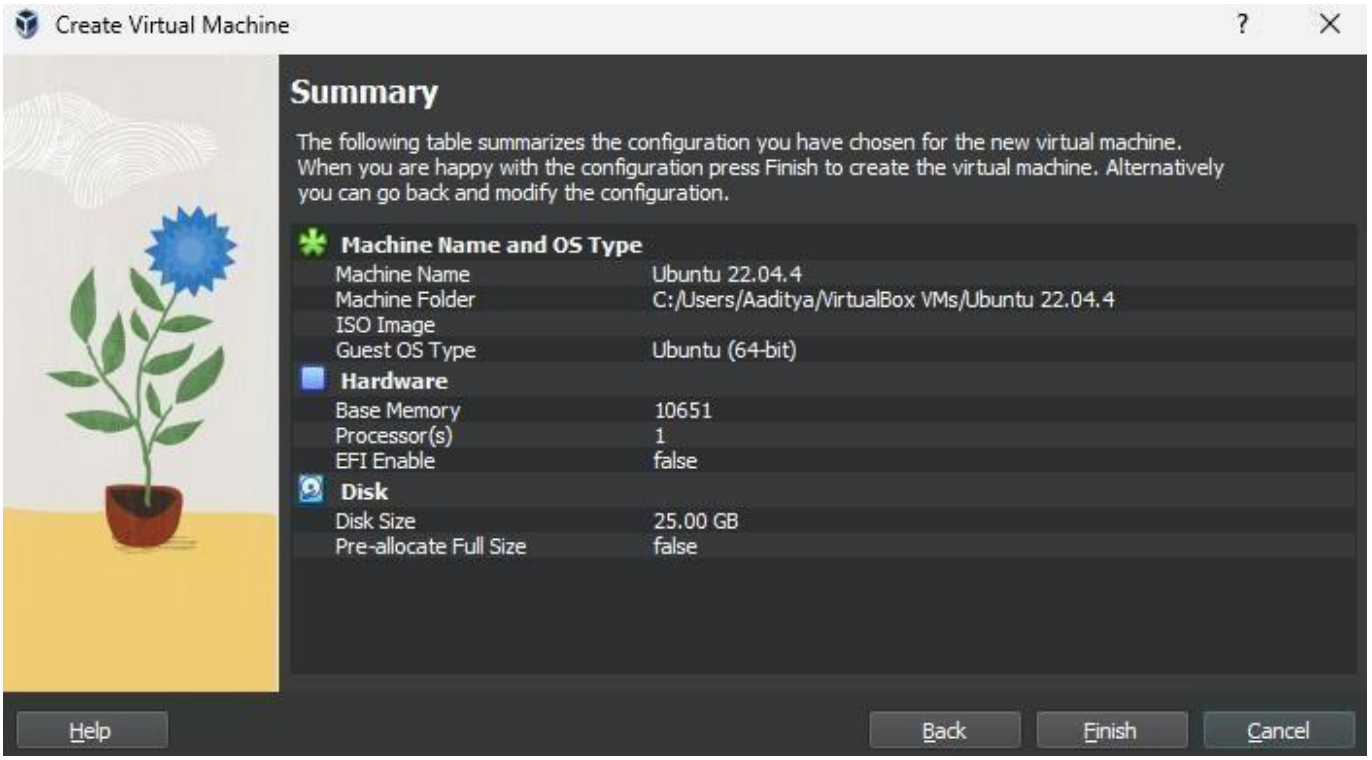
Step-2


Create Virtual Machine
Virtual machine Name and Operating System
Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.
Name: Ubuntu 22.04.4
Folder: C:\Users\Aaditya\VirtualBox VMs
ISO Image: <not selected>
Edition:
Type: Linux
Version: Ubuntu (64-bit)
Skip Unattended Installation
No ISO image is selected, the guest OS will need to be installed manually.
Help
Expert Mode
Back
Next
Cancel

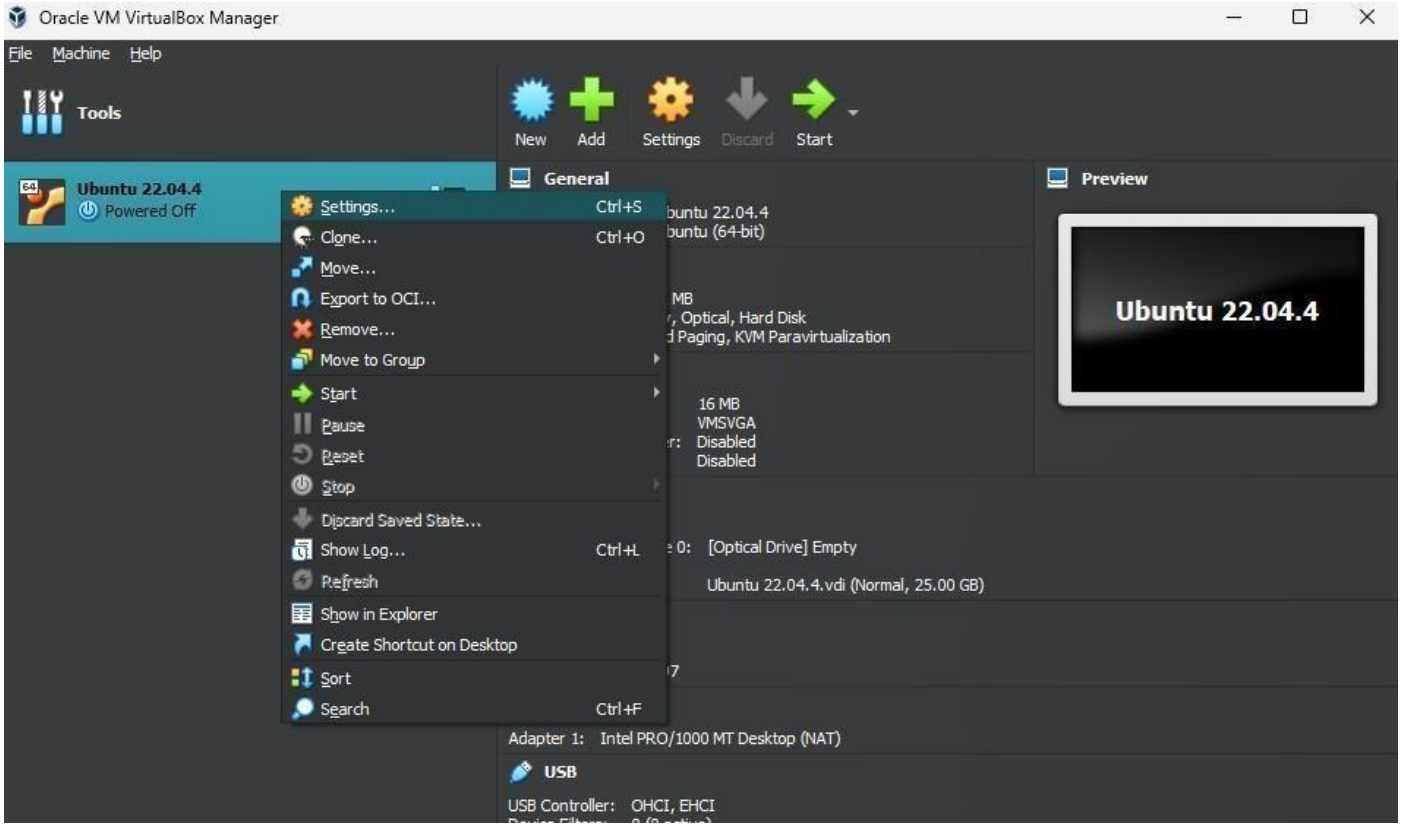
Step-3


Create Virtual Machine
Hardware
You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count.
Enabling EFI is also possible.
Base Memory:
10651 MB
4 MB
16384 MB
Processors:
1
1 CPU
20 CPUs
Enable EFI (special OSes only)
Help
Back
Next
Cancel

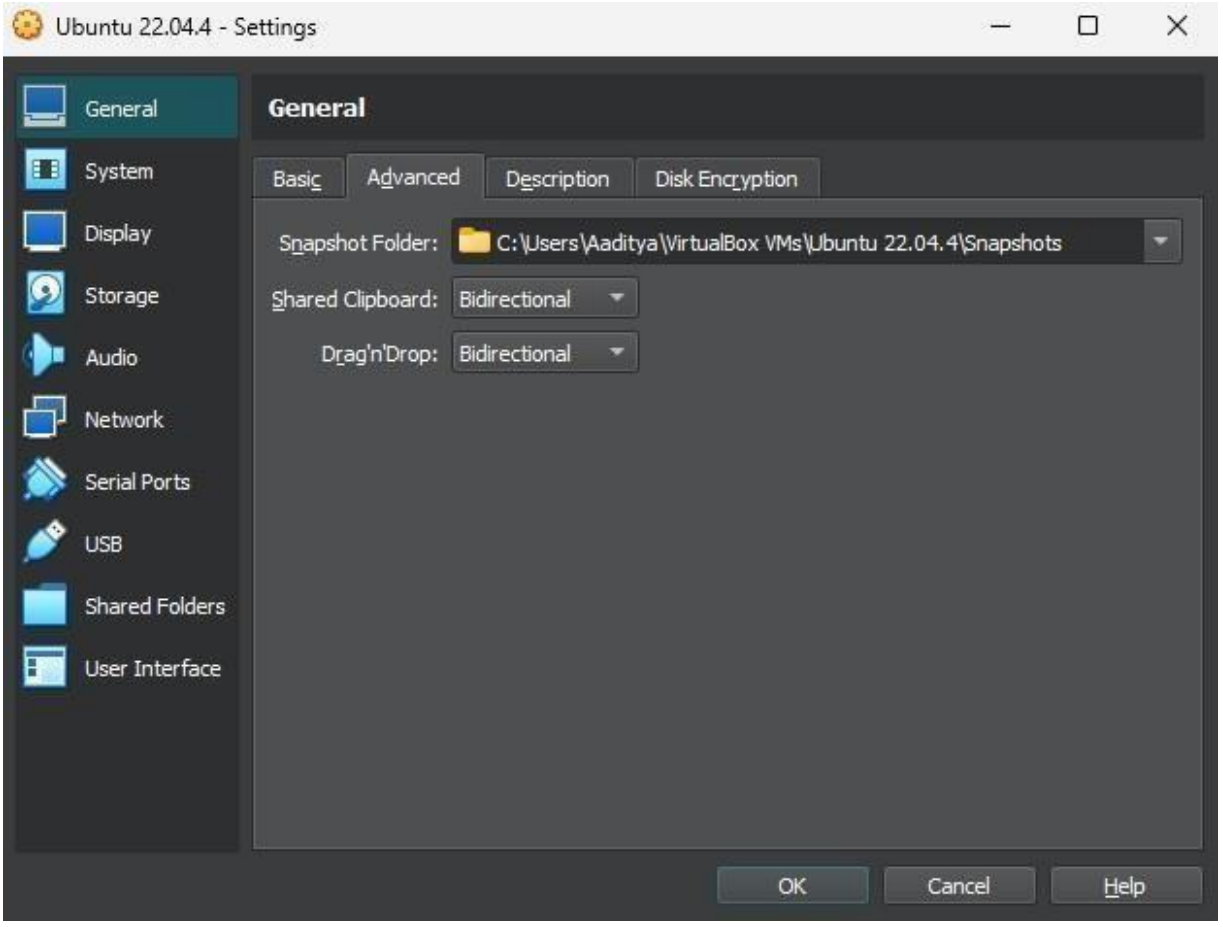
Step-4


Create Virtual Machine
Summary
The following table summarizes the configuration you have chosen for the new virtual machine.
When you are happy with the configuration press Finish to create the virtual machine. Alternatively
you can go back and modify the configuration.
Machine Name and OS Type
Machine Name
Ubuntu 22.04.4
Machine Folder
C:/Users/Aaditya/VirtualBox VMs/Ubuntu 22.04.4
ISO Image
Guest OS Type
Ubuntu (64-bit)
Hardware
Base Memory
10651
Processor(s)
1
EFI Enable
false
Disk
Disk Size
25.00 GB
Pre-allocate Full Size
false
Help
Back
Finish
Cancel

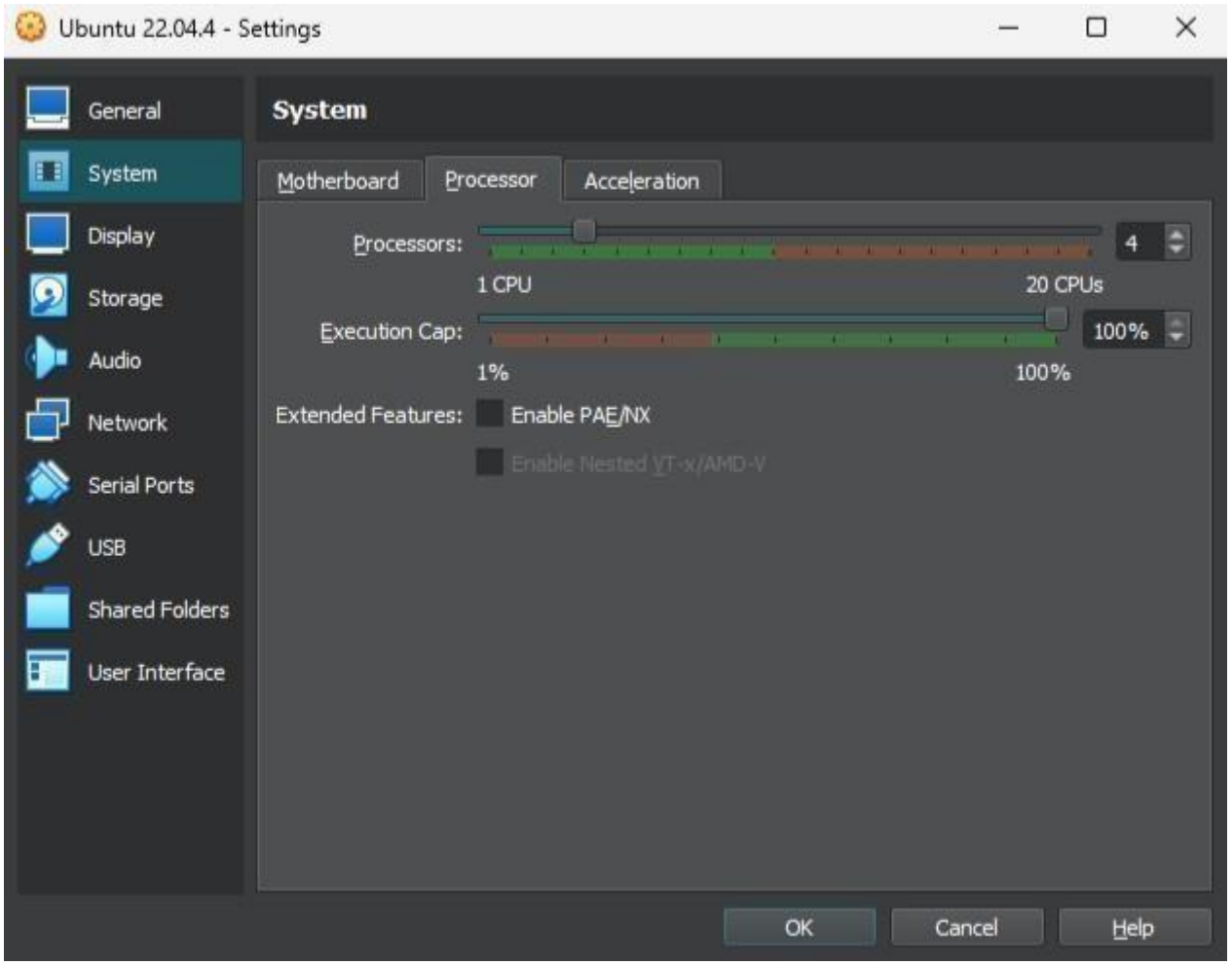
Step-5


Oracle VM VirtualBox Manager
File
Machine
Help
Tools
New
Add
Settings
Discard
Start
Ubuntu 22.04.4
Powered Off
General
Preview
Ubuntu 22.04.4
Settings... Ctrl+S
Clone... Ctrl+O
Move...
Export to OCI...
Remove...
Move to Group
Start
Pause
Reset
Stop
Discard Saved State...
Show Log... Ctrl+L
Refresh
Show in Explorer
Create Shortcut on Desktop
Sort
Search Ctrl+F
16 MB
VMSVGA
Disabled
Disabled
[Optical Drive] Empty
Ubuntu 22.04.4.vdi (Normal, 25.00 GB)
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI, EHCI

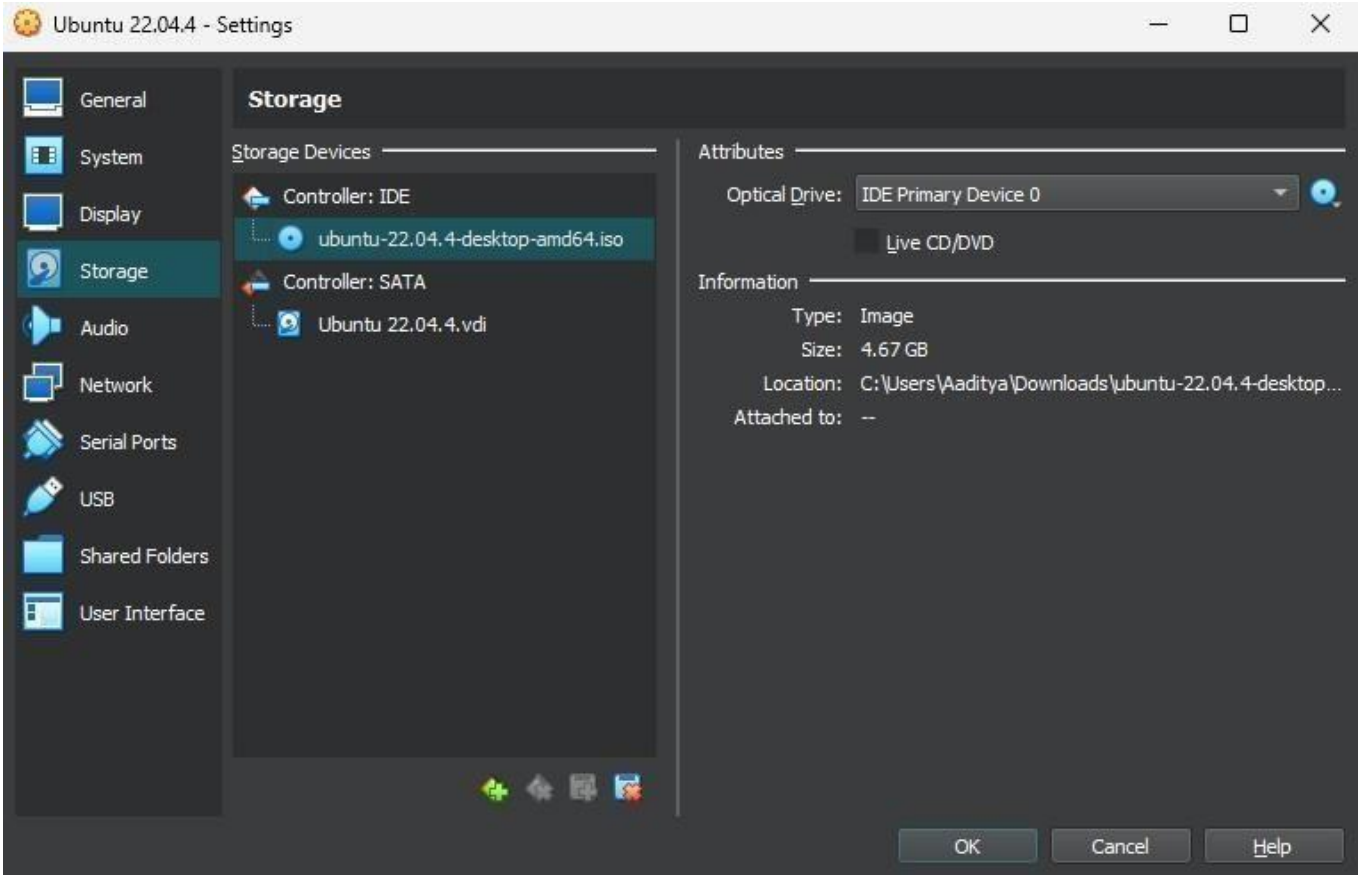
Step-6


Ubuntu 22.04.4 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
General
Basic
Advanced
Description
Disk Encryption
Snapshot Folder:
C:\Users\Aaditya\VirtualBox VMs\Ubuntu 22.04.4\Snapshots
Shared Clipboard:
Bidirectional
Drag'n'Drop:
Bidirectional
OK
Cancel
Help

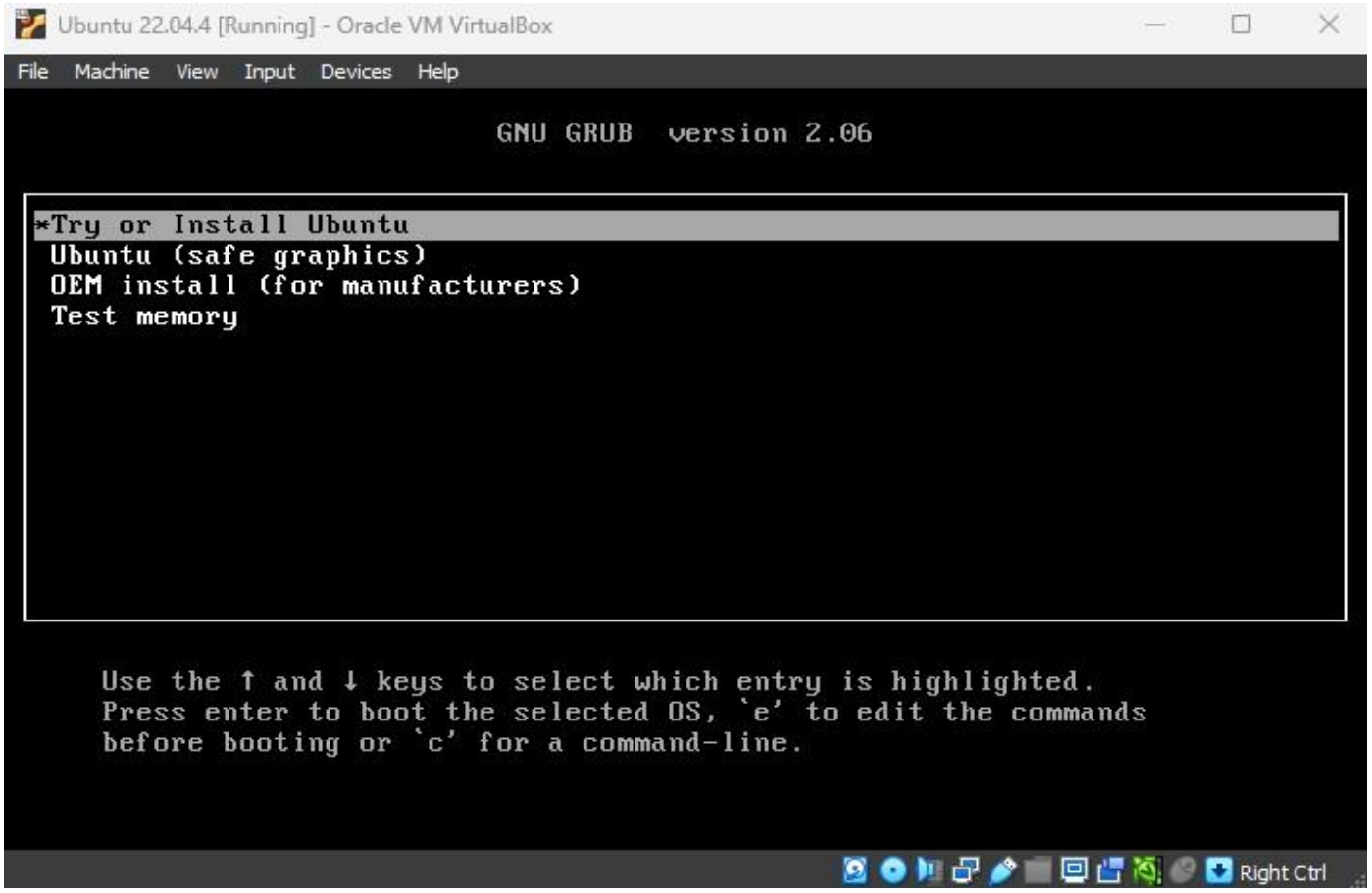
Step-7


Ubuntu 22.04.4 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
System
Motherboard
Processor
Acceleration
Processors:
4
1 CPU
20 CPUs
Execution Cap:
100%
1%
100%
Extended Features:
Enable PAE/NX
Enable Nested VT-x/AMD-V
OK
Cancel
Help

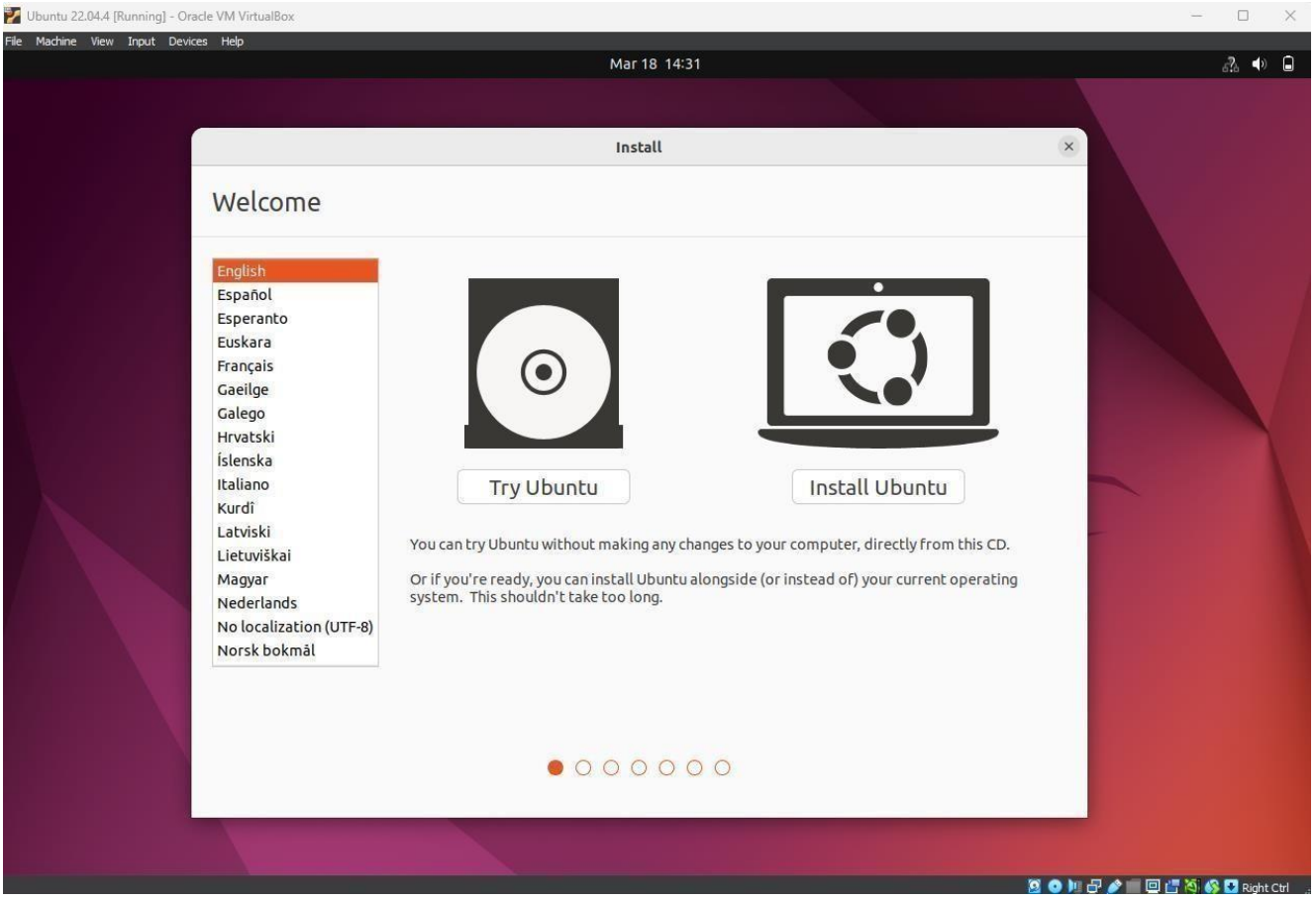
Step-8


Ubuntu 22.04.4 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Storage
Storage Devices
Controller: IDE
ubuntu-22.04.4-desktop-amd64.iso
Controller: SATA
Ubuntu 22.04.4.vdi
Attributes
Optical Drive:
IDE Primary Device 0
Live CD/DVD
Information
Type: Image
Size: 4.67 GB
Location: C:\Users\Aaditya\Downloads\ubuntu-22.04.4-desktop...
Attached to: --
OK
Cancel
Help

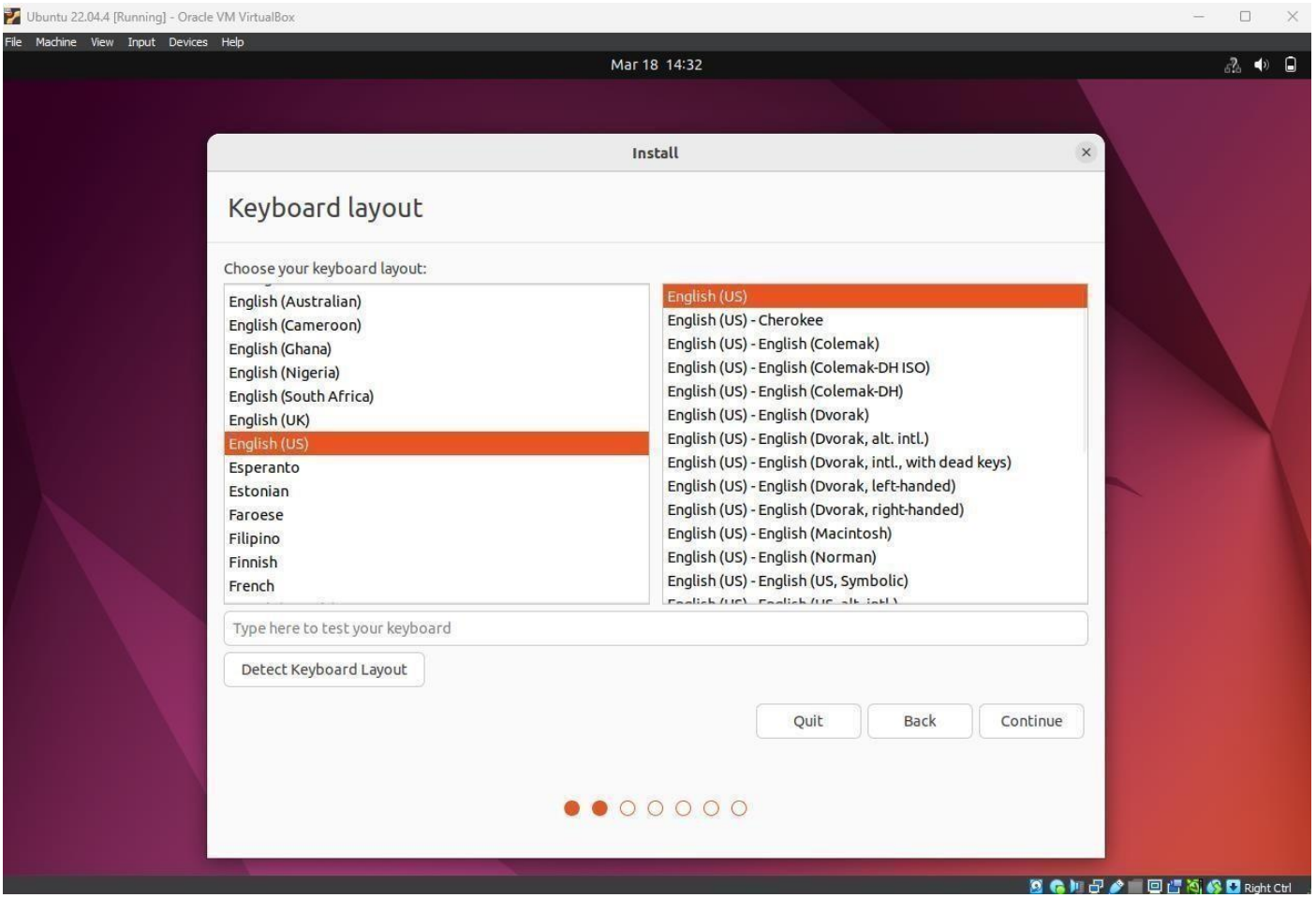
Step-9



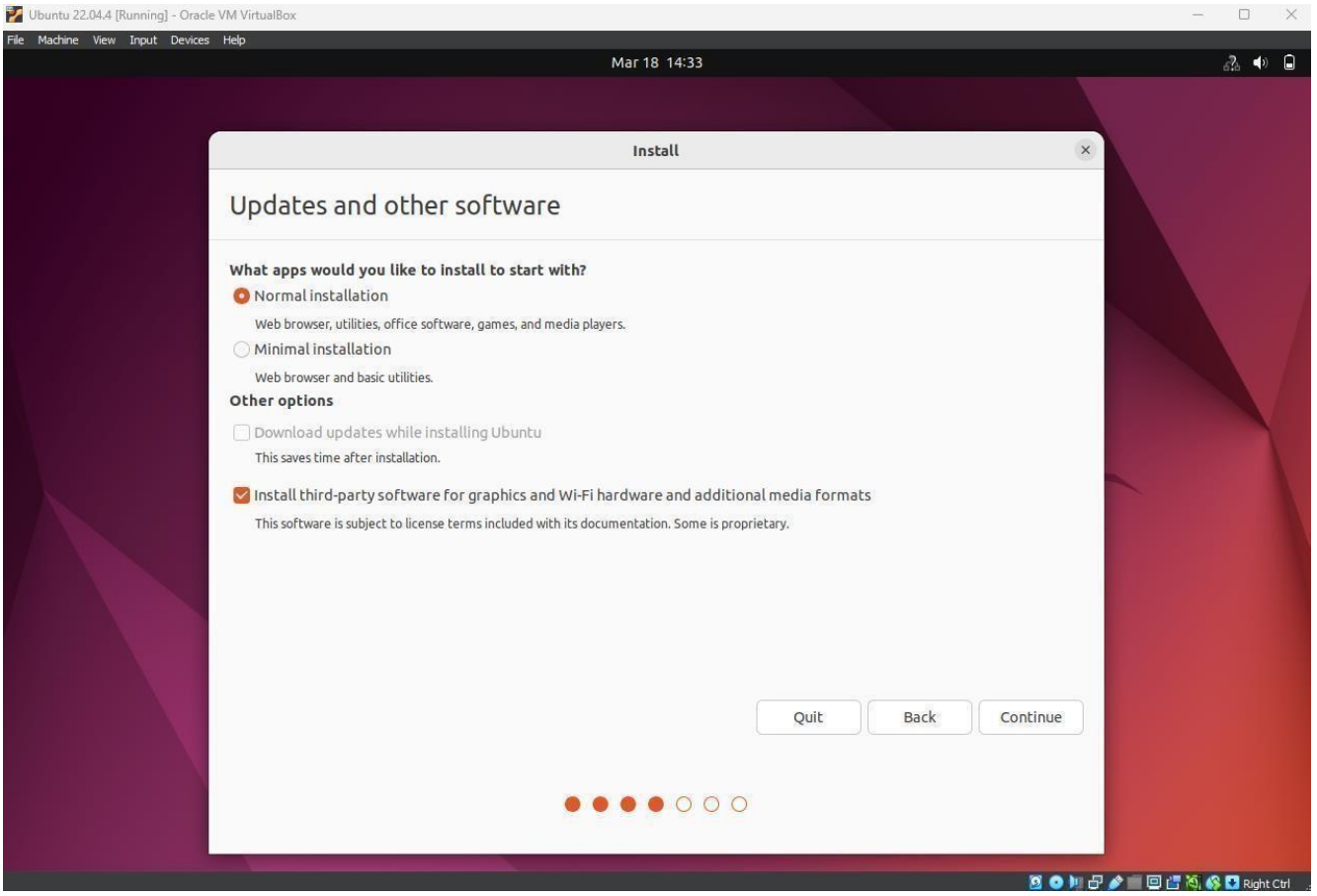
Step-10



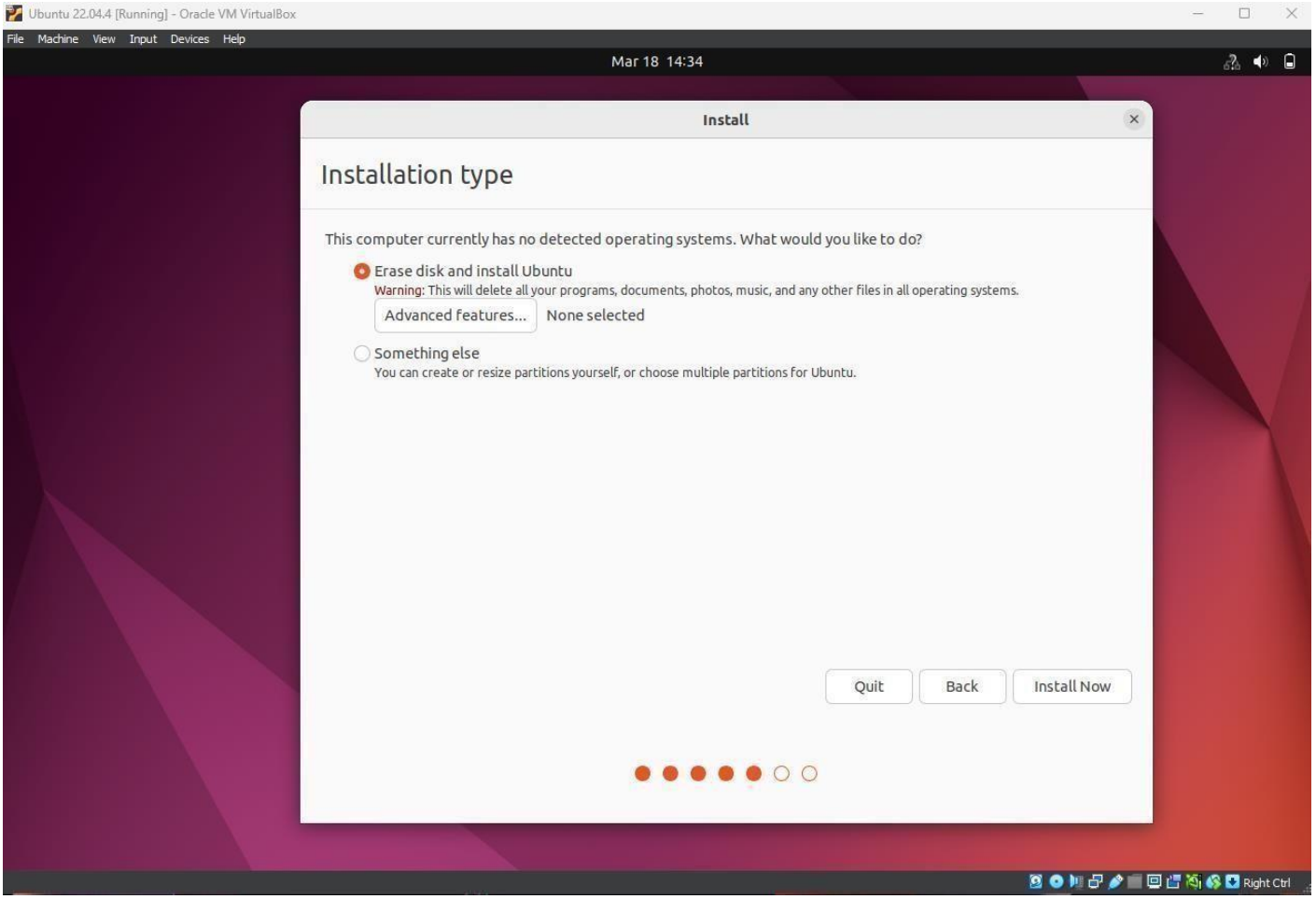
Step-11



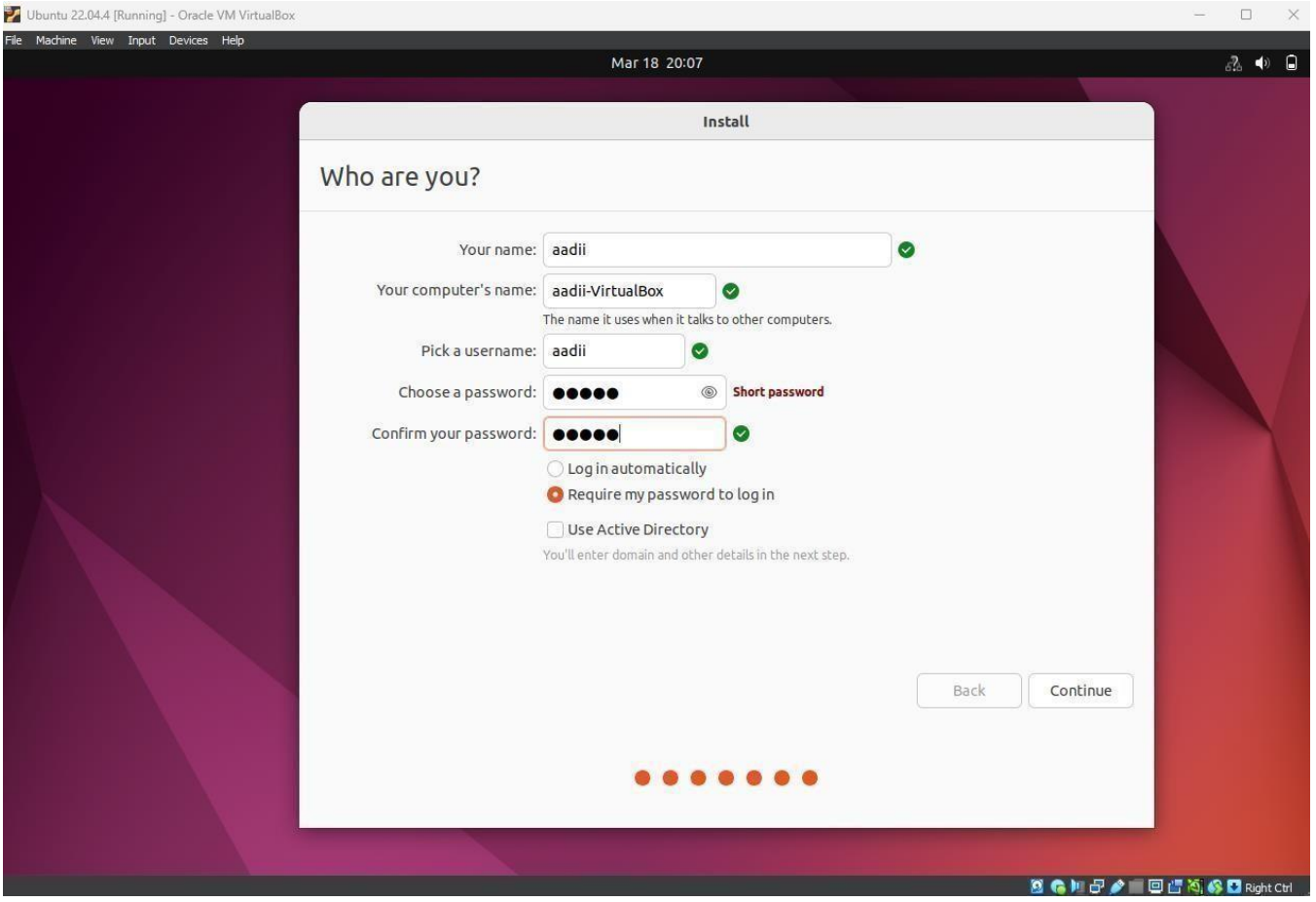
Step-12


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mar 18 14:33
Install
Updates and other software
What apps would you like to install to start with?
Normal installation
Web browser, utilities, office software, games, and media players.
Minimal installation
Web browser and basic utilities.
Other options
Download updates while installing Ubuntu
This saves time after installation.
Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.
Quit
Back
Continue
Right Ctrl

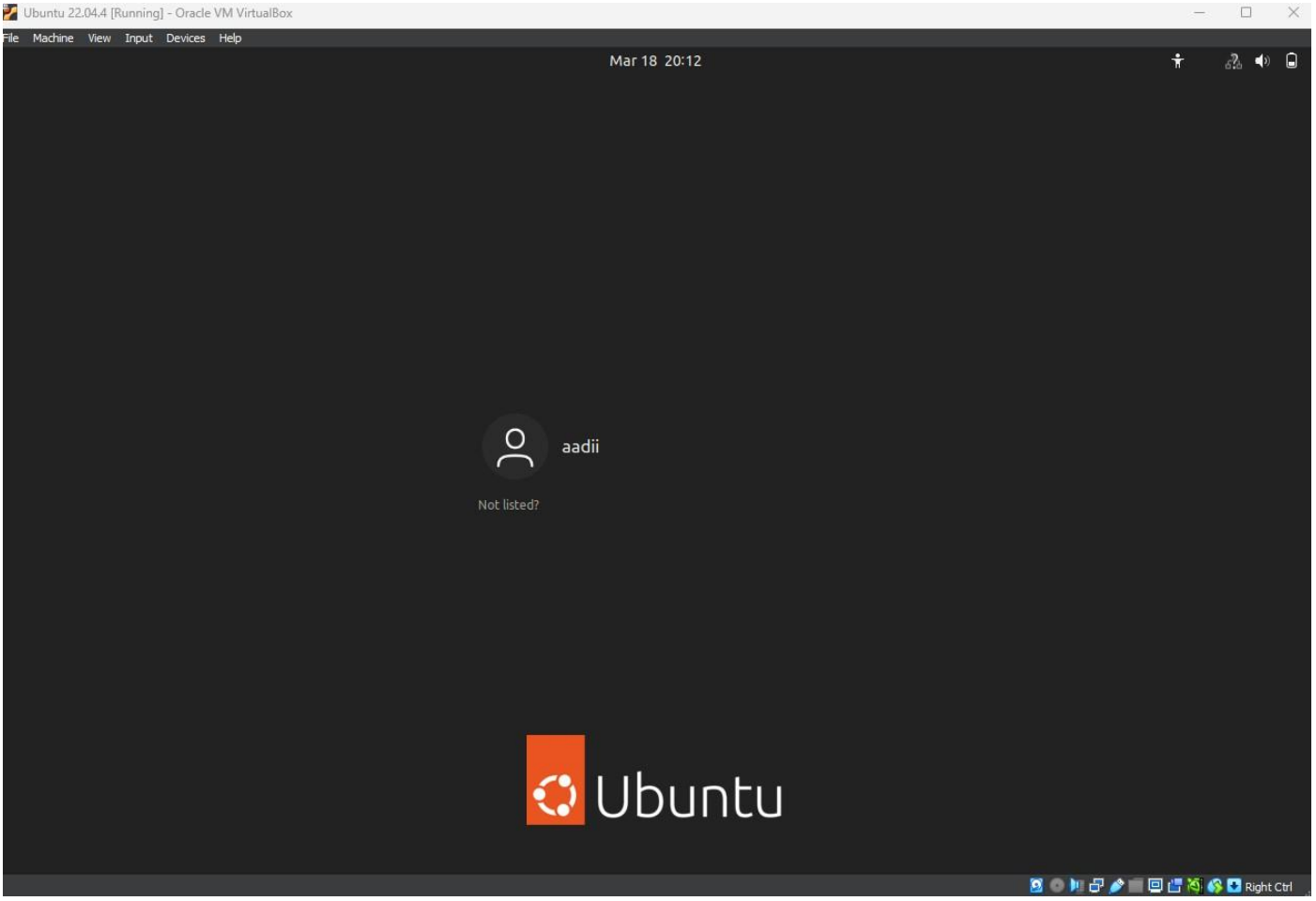
Step-13


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mar 18 14:34
Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features...
None selected
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.
Quit
Back
Install Now
Right Ctrl

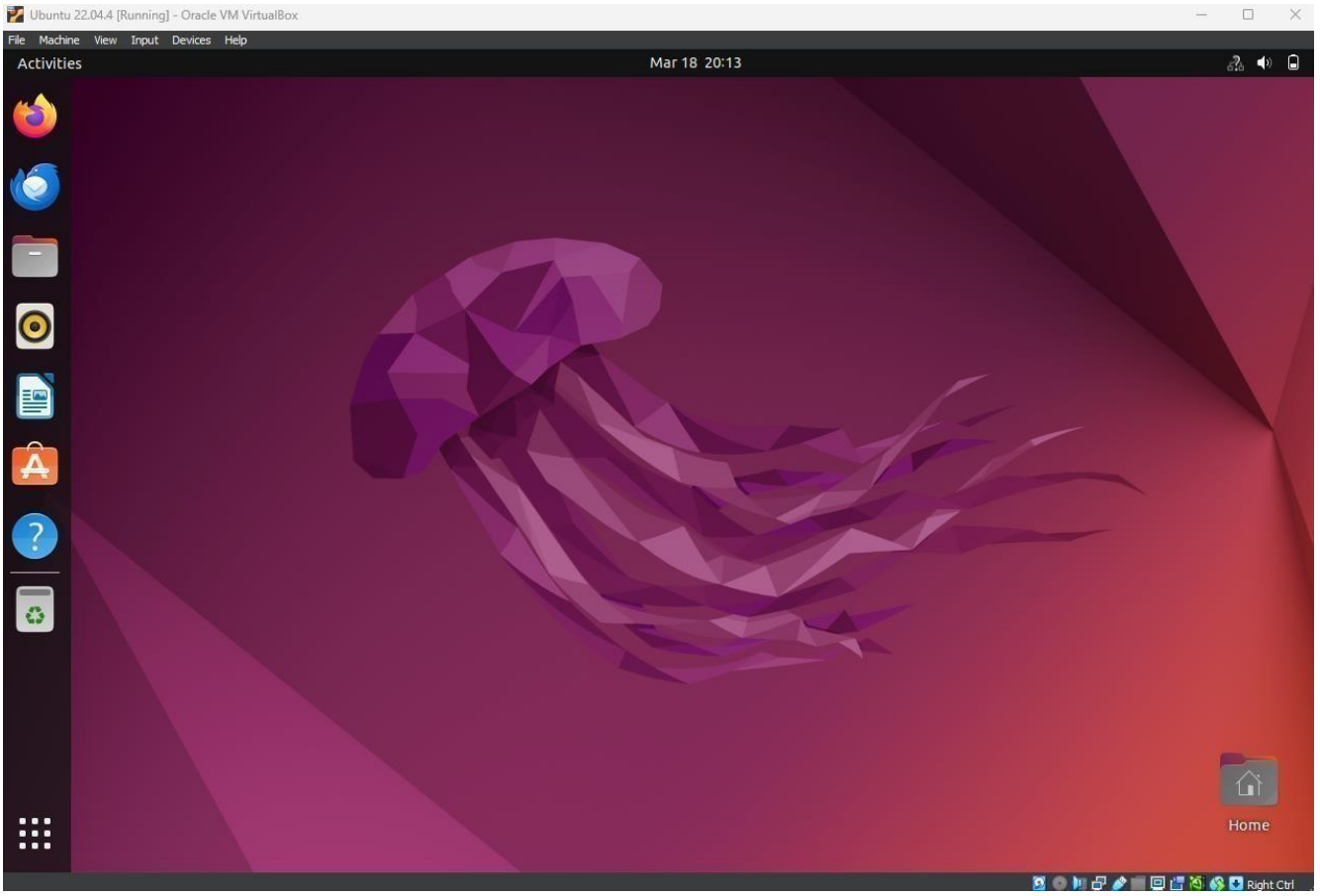
Step-14


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mar 18 20:07
Install
Who are you?
Your name: aadii
Your computer's name: aadii-VirtualBox
The name it uses when it talks to other computers.
Pick a username: aadii
Choose a password:
Short password
Confirm your password:
Log in automatically
Require my password to log in
Use Active Directory
You'll enter domain and other details in the next step.
Back
Continue
Right Ctrl

Step-15



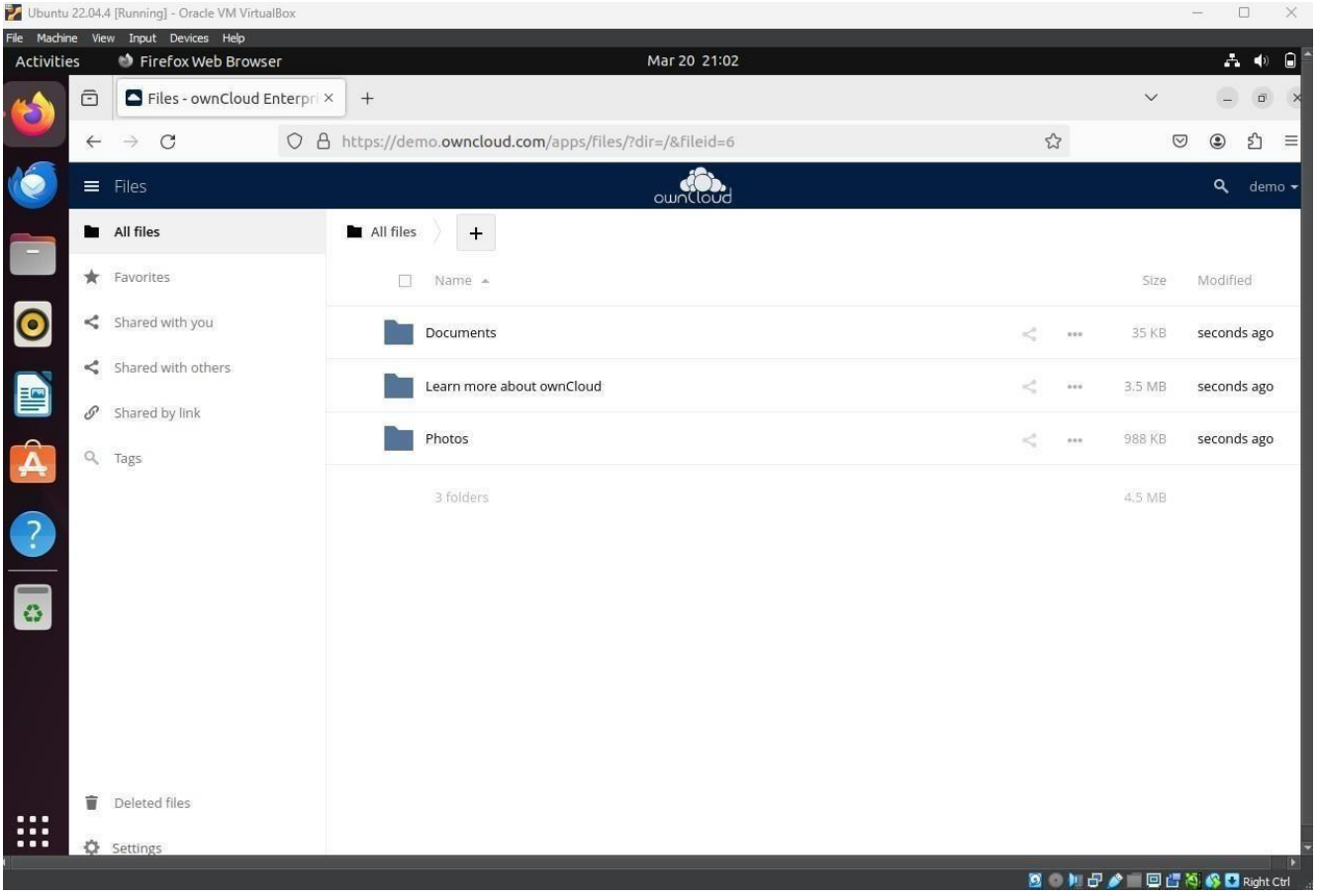
Step-16

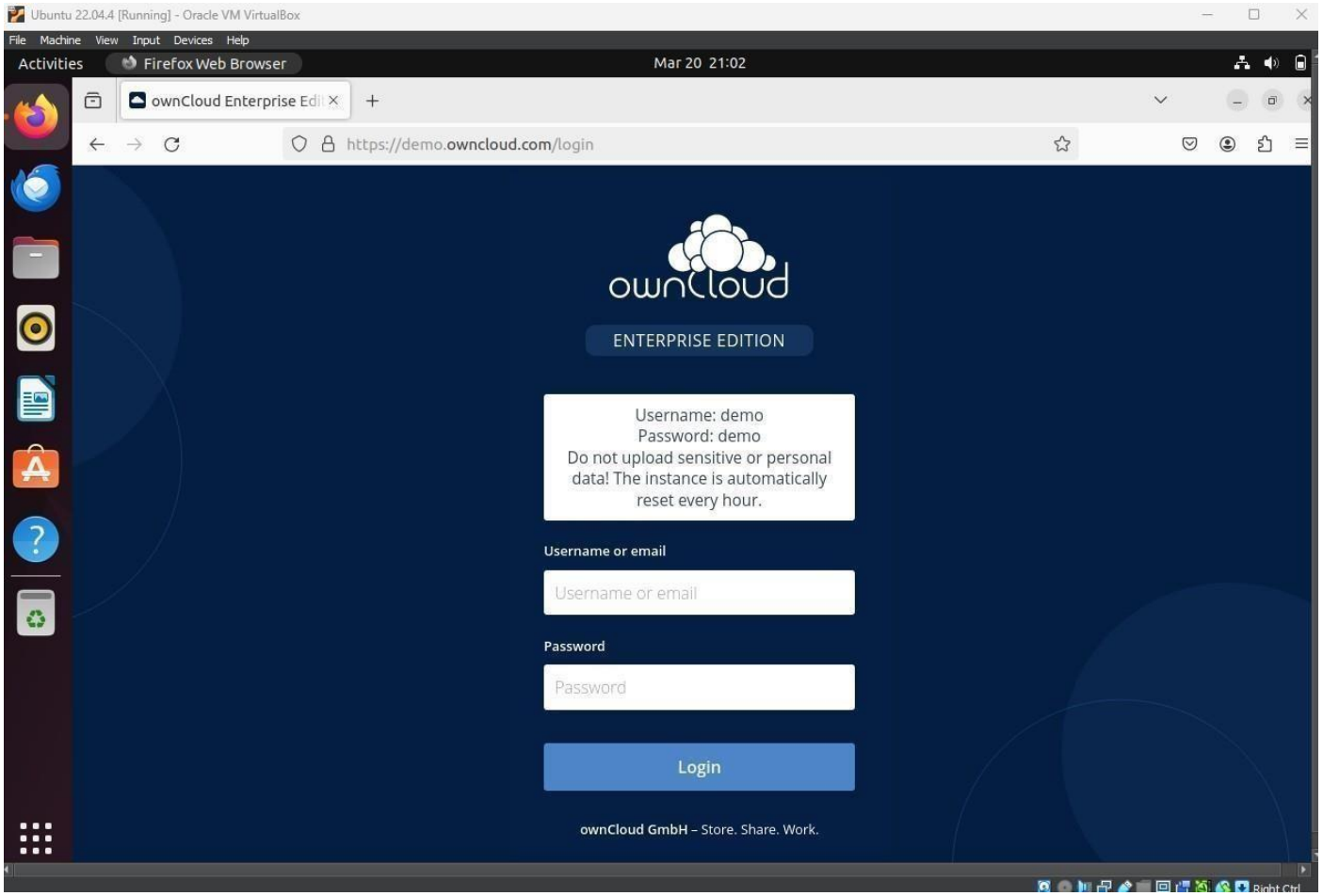

Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Mar 18 20:13
Home
Right Ctrl

Roll No:01

Services

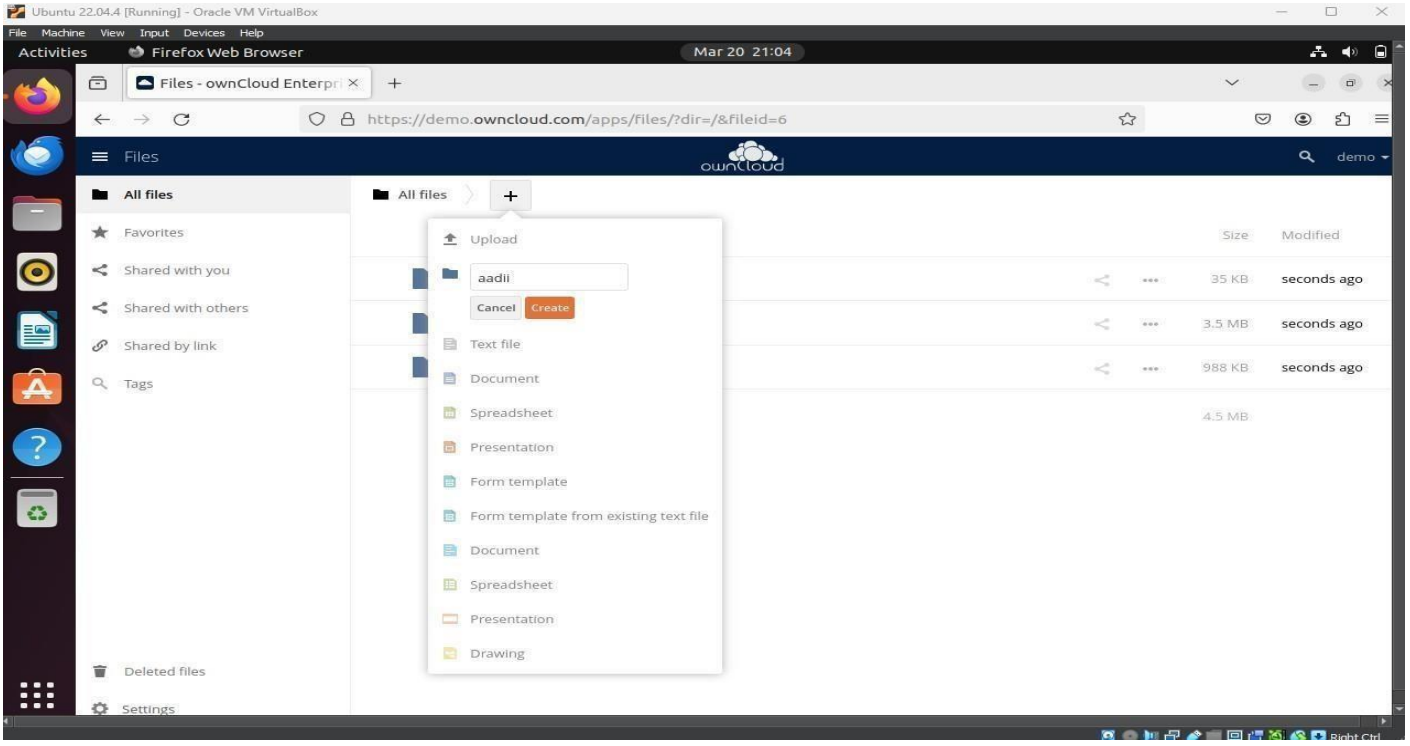
Step-17



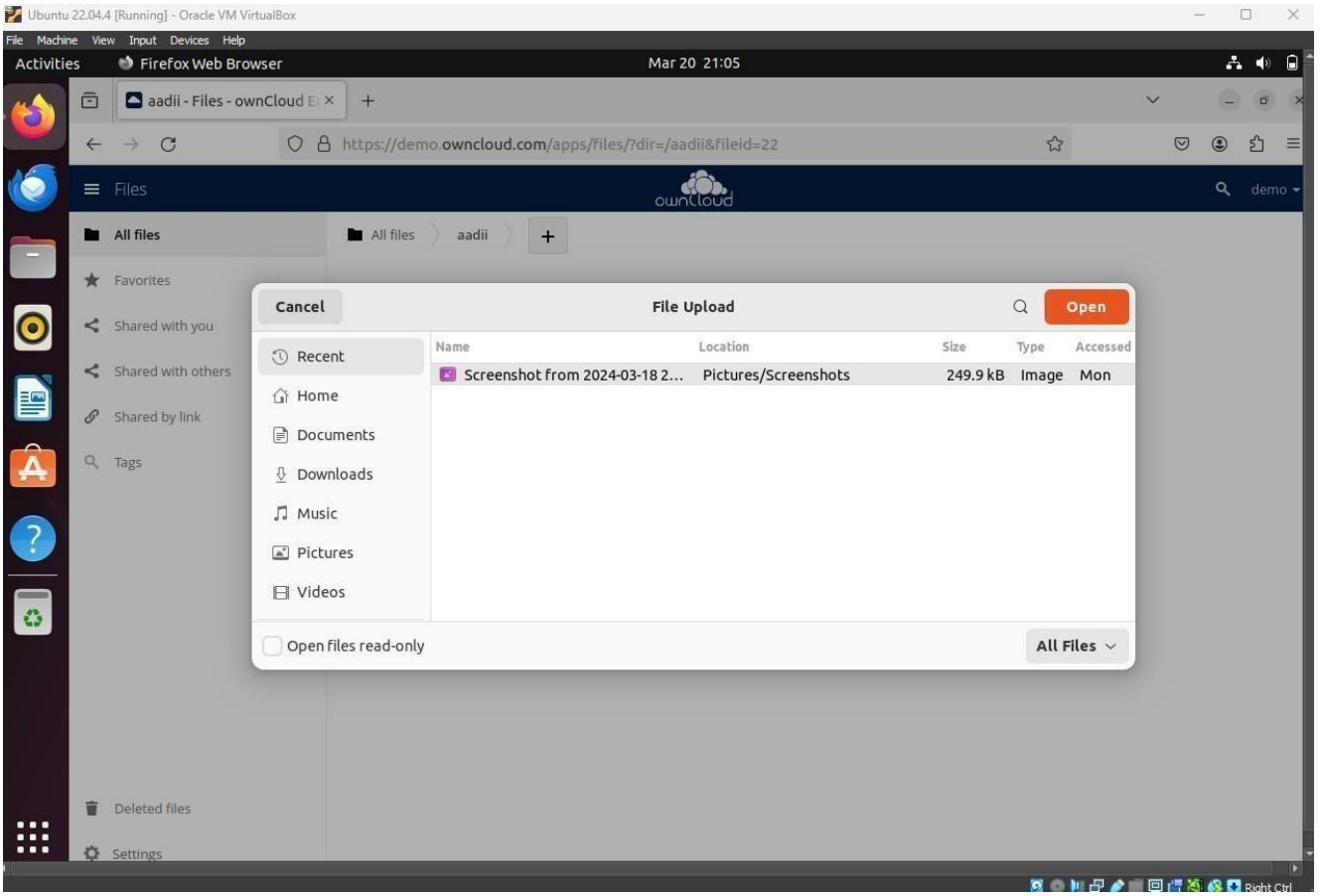


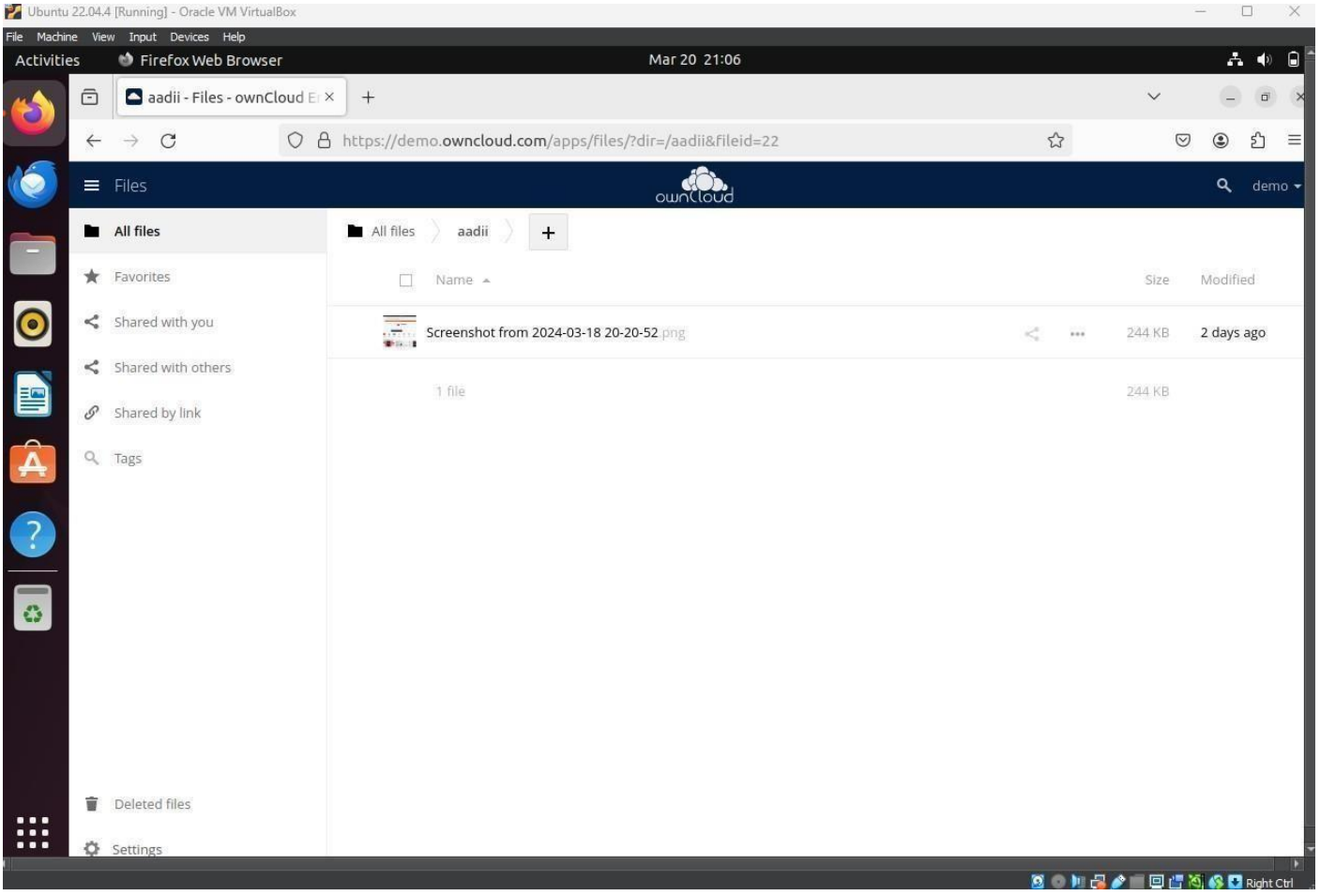
Step-18

Step-19

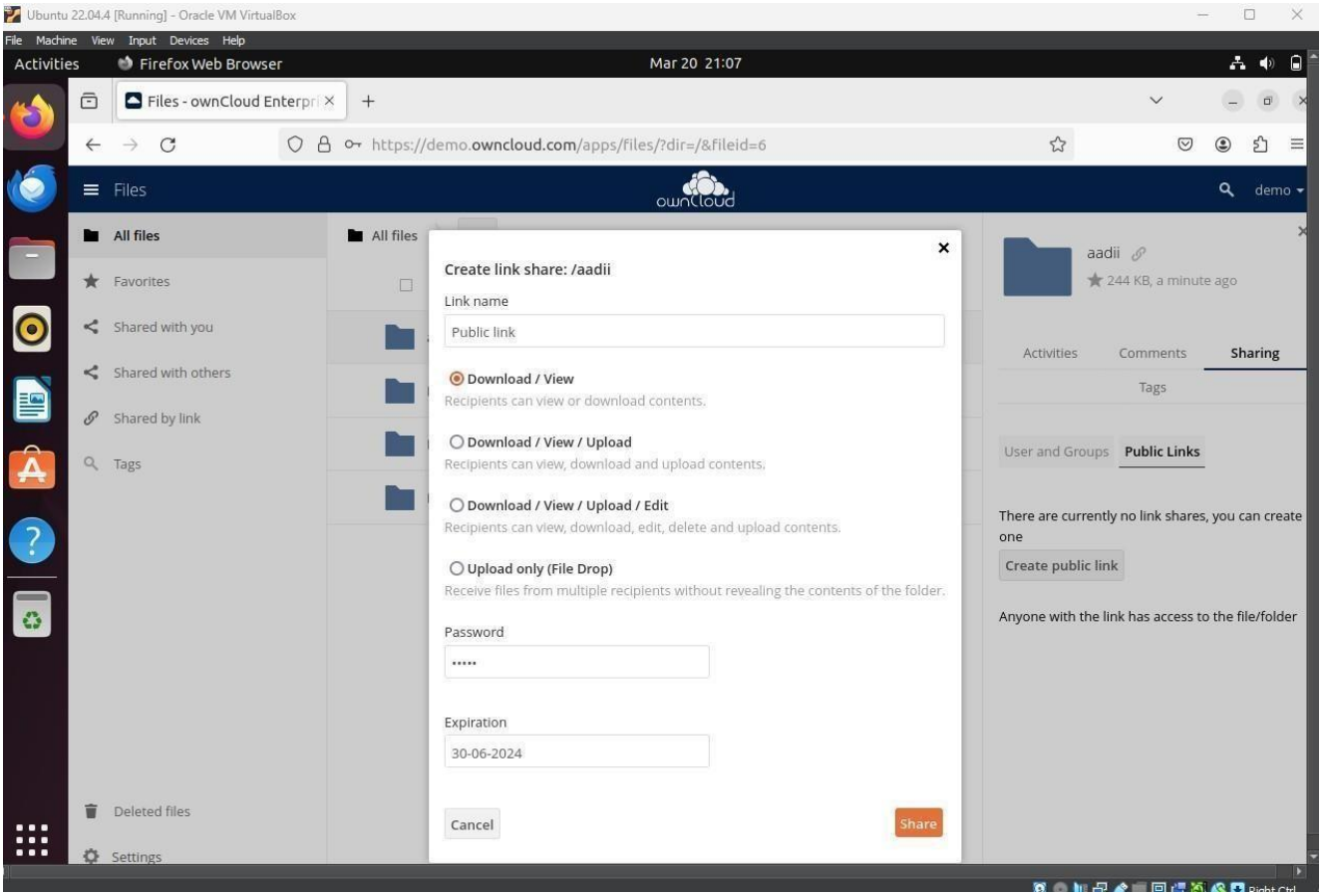


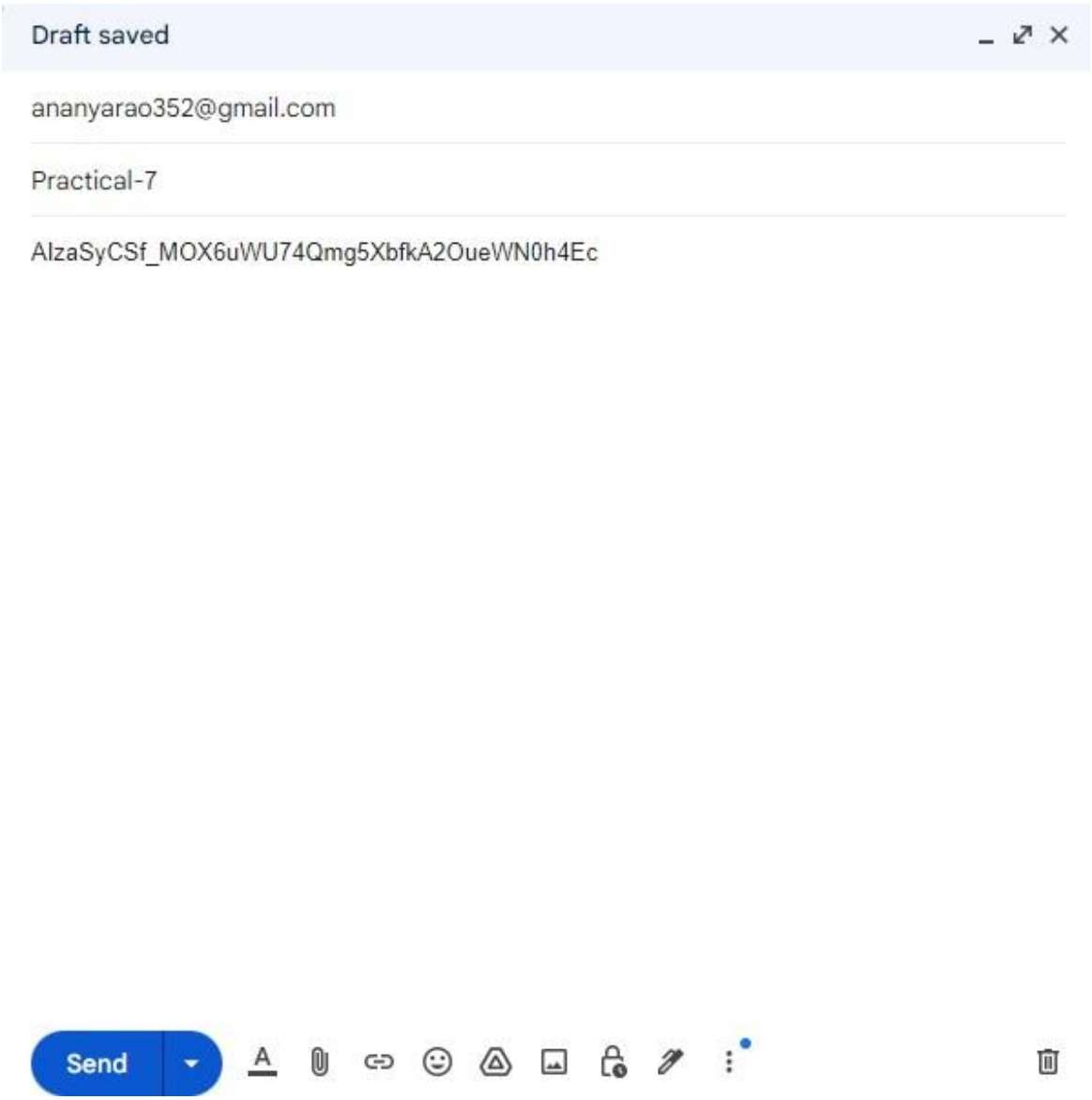
Step-20


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Mar 20 21:05
aadii - Files - ownCloud
https://demo.owncloud.com/apps/files/?dir=/aadii&fileid=22
Files
owncloud
demo
All files
Favorites
Shared with you
Shared with others
Shared by link
Tags
Deleted files
Settings
All files
aadii
Cancel
File Upload
Open
Recent
Home
Documents
Downloads
Music
Pictures
Videos
Name
Location
Size
Type
Accessed
Screenshot from 2024-03-18 2...
Pictures/Screenshots
249.9 kB
Image
Mon
Open files read-only
All Files
Right Ctrl


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Mar 20 21:06
aadii - Files - ownCloud
https://demo.owncloud.com/apps/files/?dir=/aadii&fileid=22
Files
owncloud
demo
All files
Favorites
Shared with you
Shared with others
Shared by link
Tags
Deleted files
Settings
All files
aadii
Name
Size
Modified
Screenshot from 2024-03-18 20-20-52 png
244 KB
2 days ago
1 file
244 KB
Right Ctrl

Step-22


Ubuntu 22.04.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Mar 20 21:07
Files - ownCloud Enterpri
https://demo.owncloud.com/apps/files/?dir=/&fileid=6
Files
ownCloud
demo
All files
Favorites
Shared with you
Shared with others
Shared by link
Tags
Deleted files
Settings
All files
Create link share: /aadii
Link name
Public link
Download / View
Recipients can view or download contents.
Download / View / Upload
Recipients can view, download and upload contents.
Download / View / Upload / Edit
Recipients can view, download, edit, delete and upload contents.
Upload only (File Drop)
Receive files from multiple recipients without revealing the contents of the folder.
Password
Expiration
30-06-2024
Cancel
Share
aadii
244 KB, a minute ago
Activities
Comments
Sharing
Tags
User and Groups
Public Links
There are currently no link shares, you can create one
Create public link
Anyone with the link has access to the file/folder

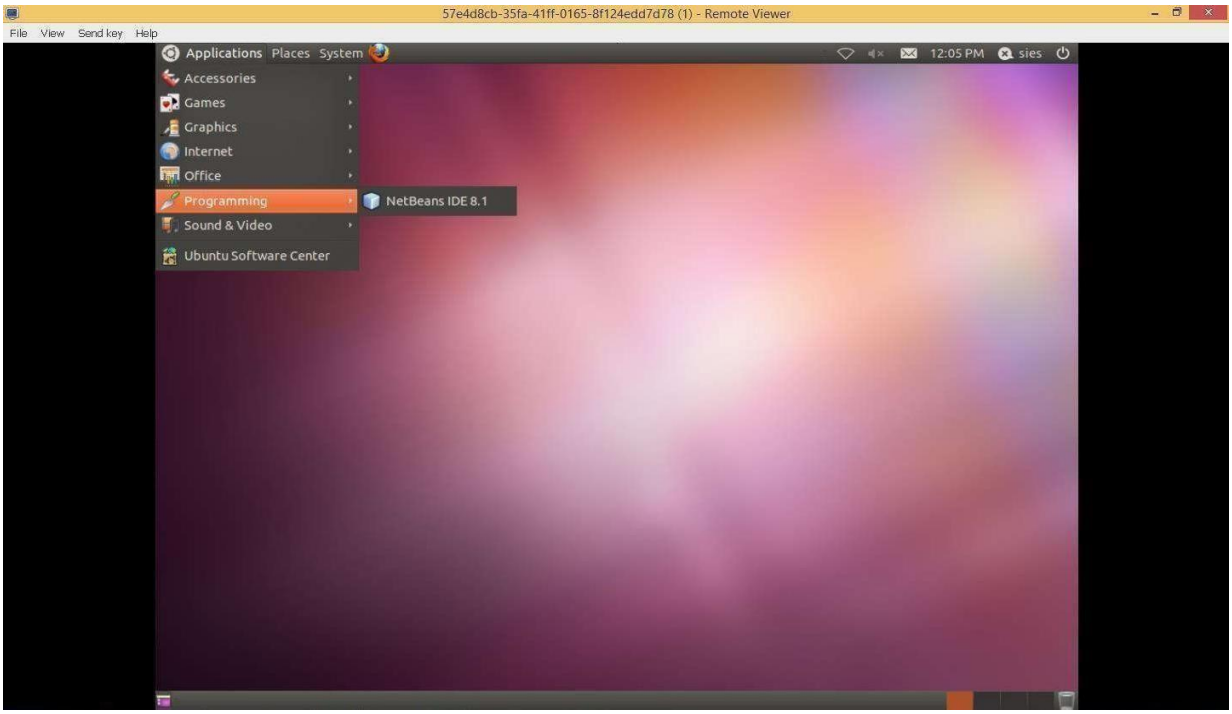


PRACTICAL 8

Aim: Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS),

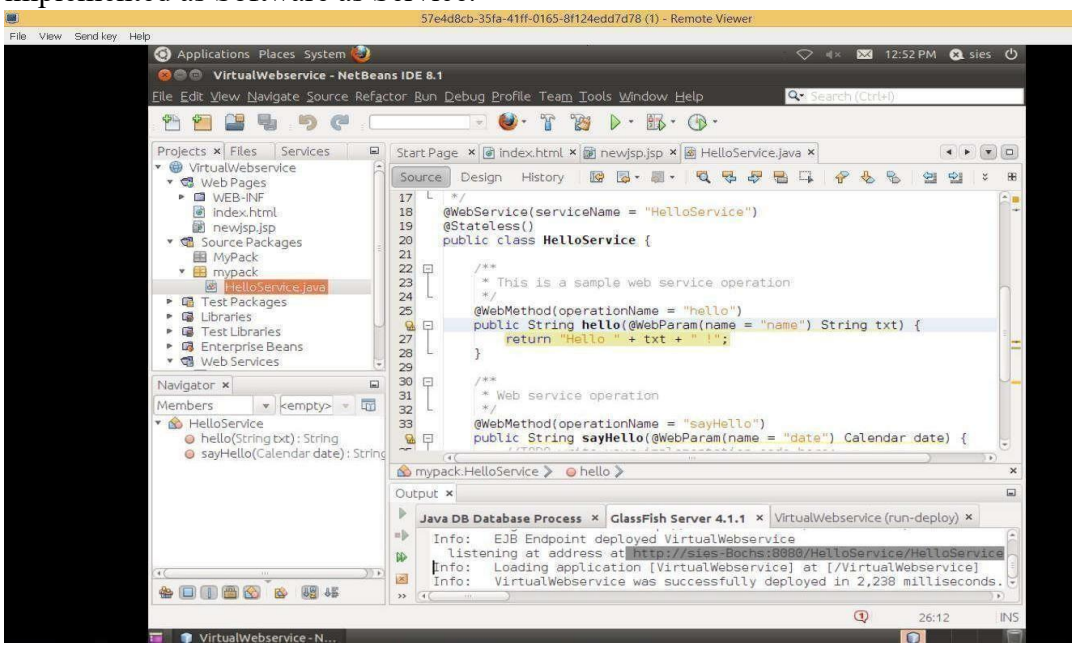
Software development Kits can be made available in the Virtual Machines that can be implemented as Platform as a Service.

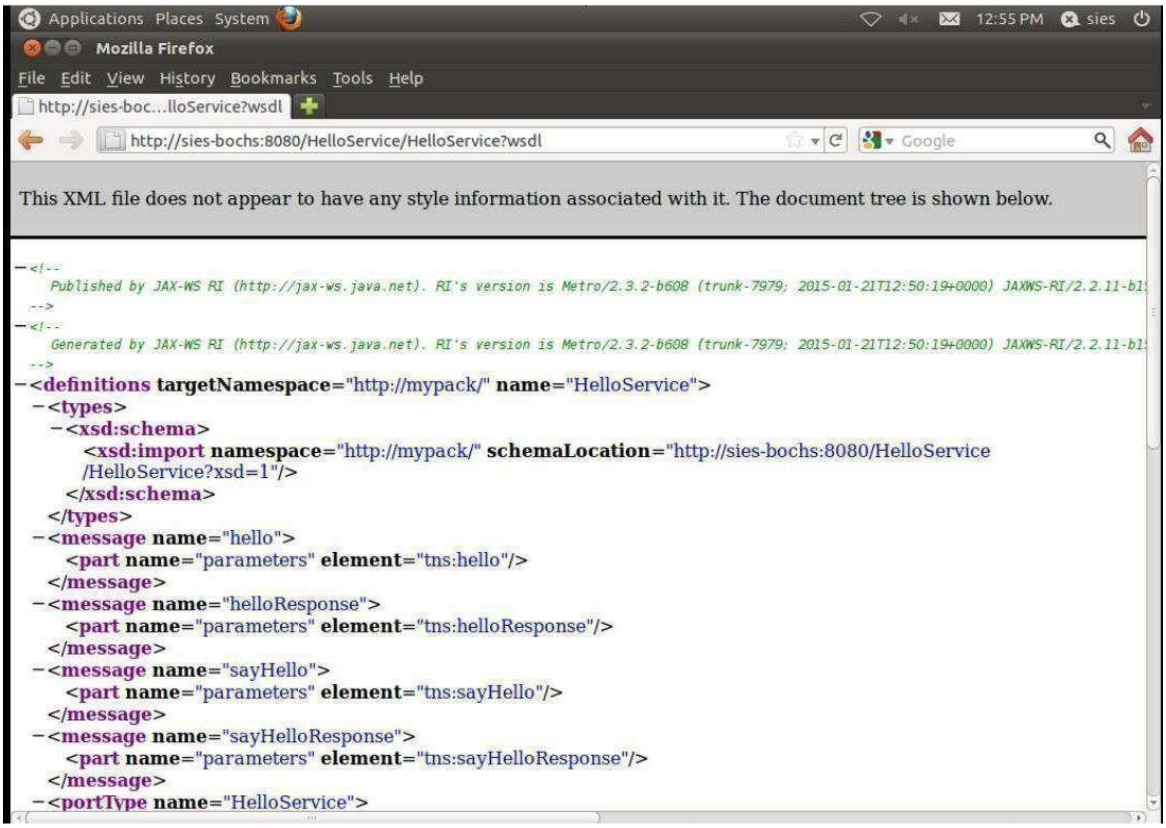
Installationof Netbeans, Eclipse, Visual Studio and DBMS can be done in the appropriate Virtual Machines.



Aim: Implement FOSS-Cloud Functionality VSI Software as a Service (SaaS)

Applications created and deployed in the virtual machines can be accessed by outside world. This can be implemented as Software as Service.



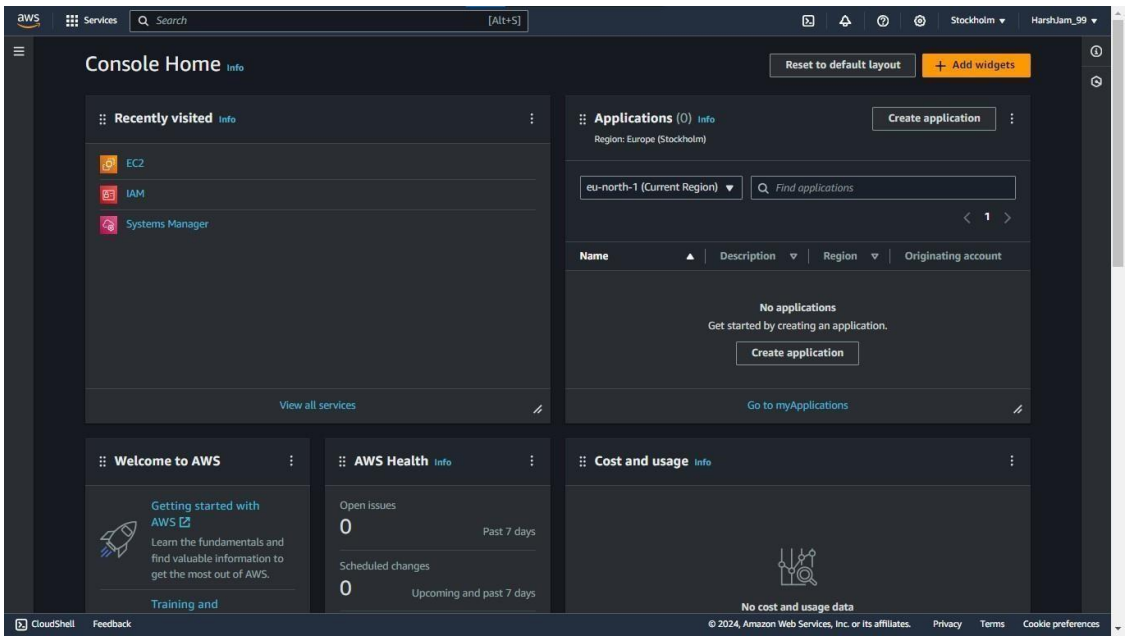

Applications Places System
12:55 PM sies
Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://sies-boc...lloService?wsdl
http://sies-bochs:8080/HelloService/HelloService?wsdl
Google
This XML file does not appear to have any style information associated with it. The document tree is shown below.
-<!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
-->
-<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
-->
-<definitions targetNamespace="http://mypack/" name="HelloService">
-<types>
-<xsd:schema>
<xsd:import namespace="http://mypack/" schemaLocation="http://sies-bochs:8080/HelloService /HelloService?xsd=1"/>
</xsd:schema>
</types>
-<message name="hello">
<part name="parameters" element="tns:hello"/>
</message>
-<message name="helloResponse">
<part name="parameters" element="tns:helloResponse"/>
</message>
-<message name="sayHello">
<part name="parameters" element="tns:sayHello"/>
</message>
-<message name="sayHelloResponse">
<part name="parameters" element="tns:sayHelloResponse"/>
</message>
-<portType name="HelloService">

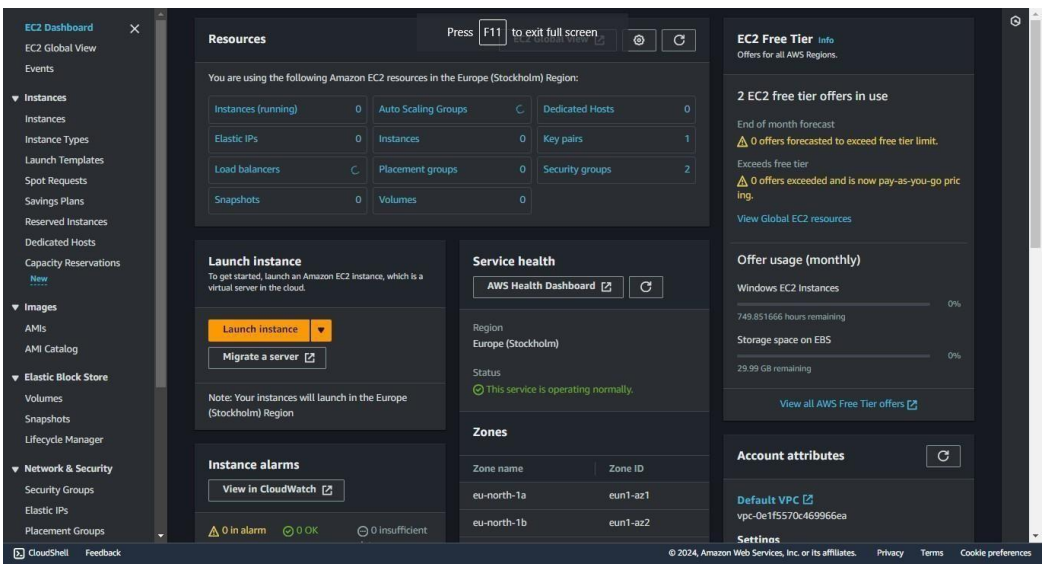
PRACTICAL 9

Aim: Using AWS Flow Framework develop application that includes a simple workflow. Workflow calls an activity to print hello world to the console. It mustdefine the basic usage of AWS Flow Framework, including defining contracts, implementation of activities and workflow coordination logic and worker programs to host them

Step 1: Opening the Amazon EC2 console

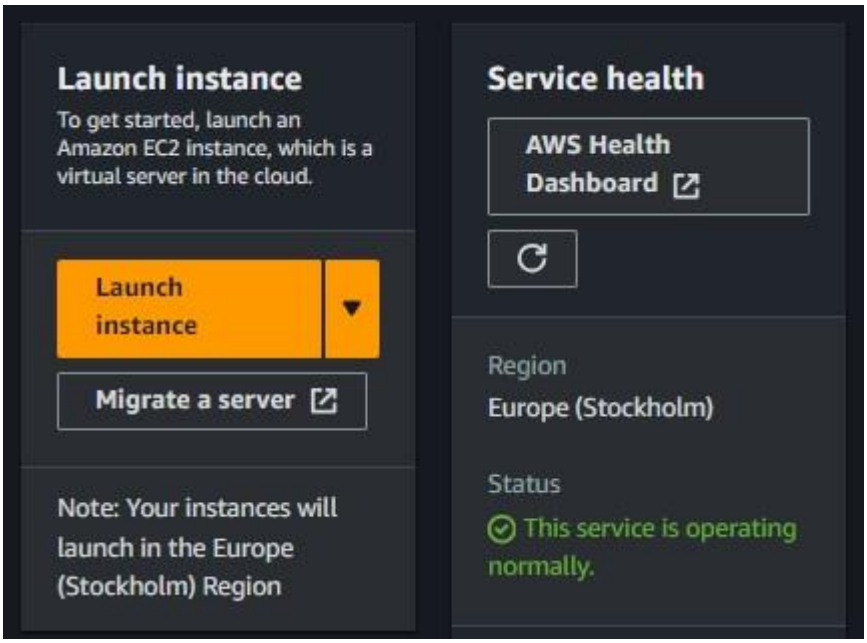
1.1 In the console navigation pane, choose **EC2 Dashboard**



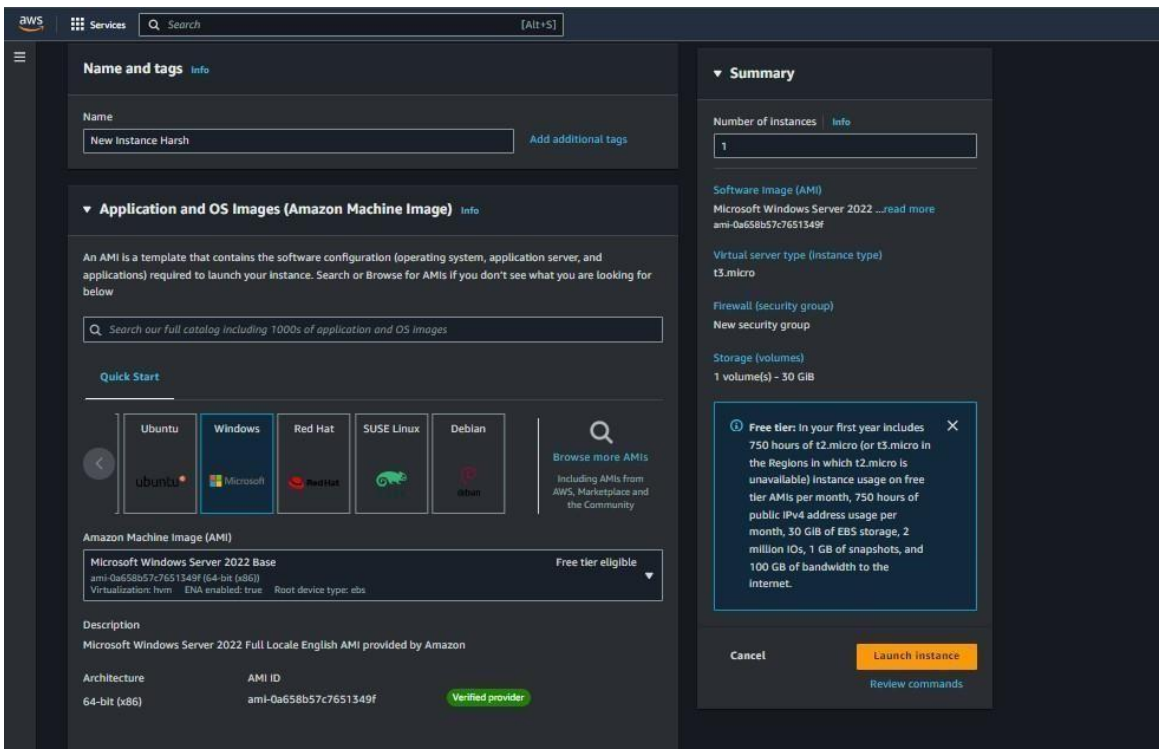


Step 2: Creating an EC2 instance and launching the instance

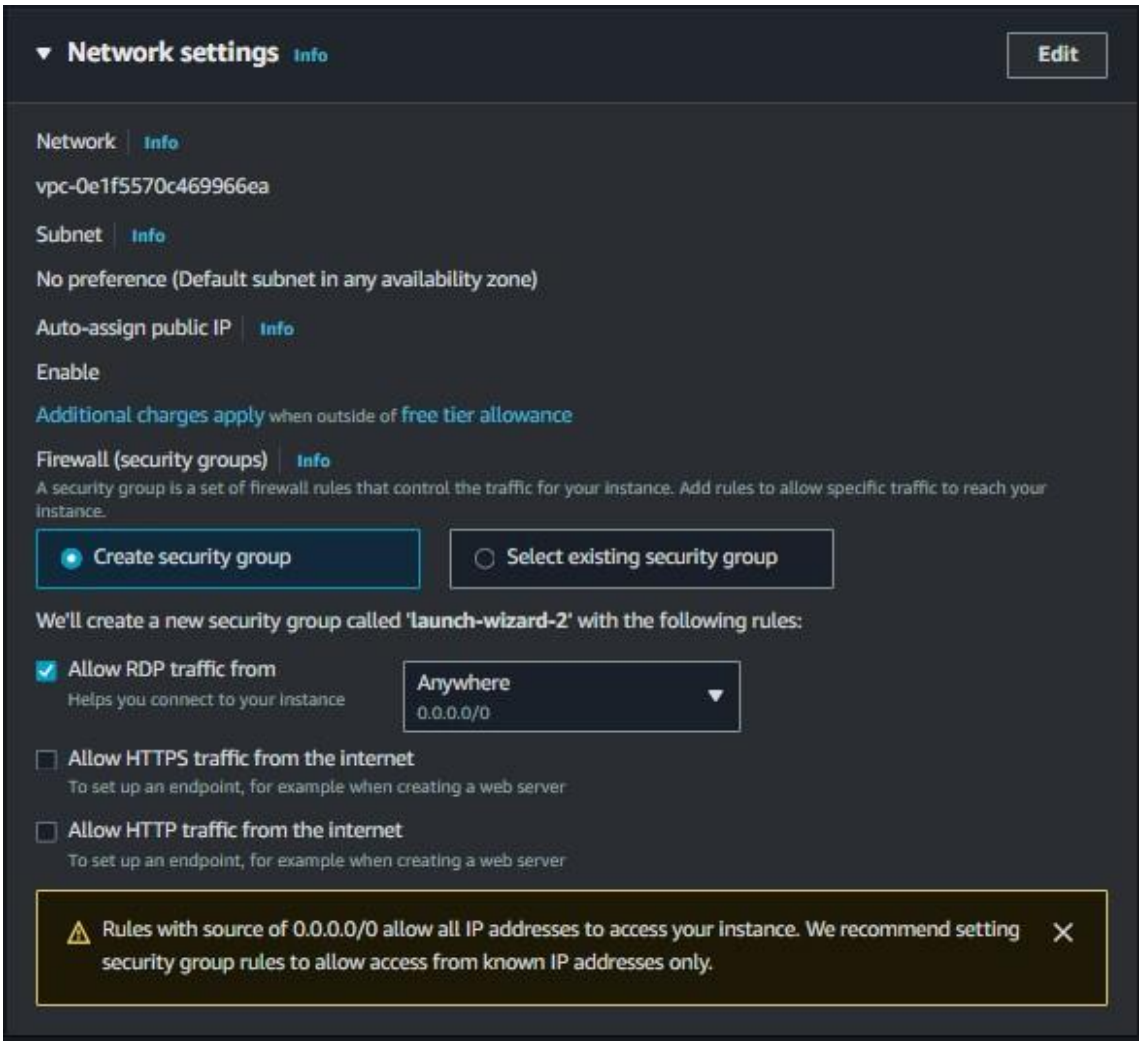
2.1 In the **Launch instance section** of the console, choose **Launch instance**



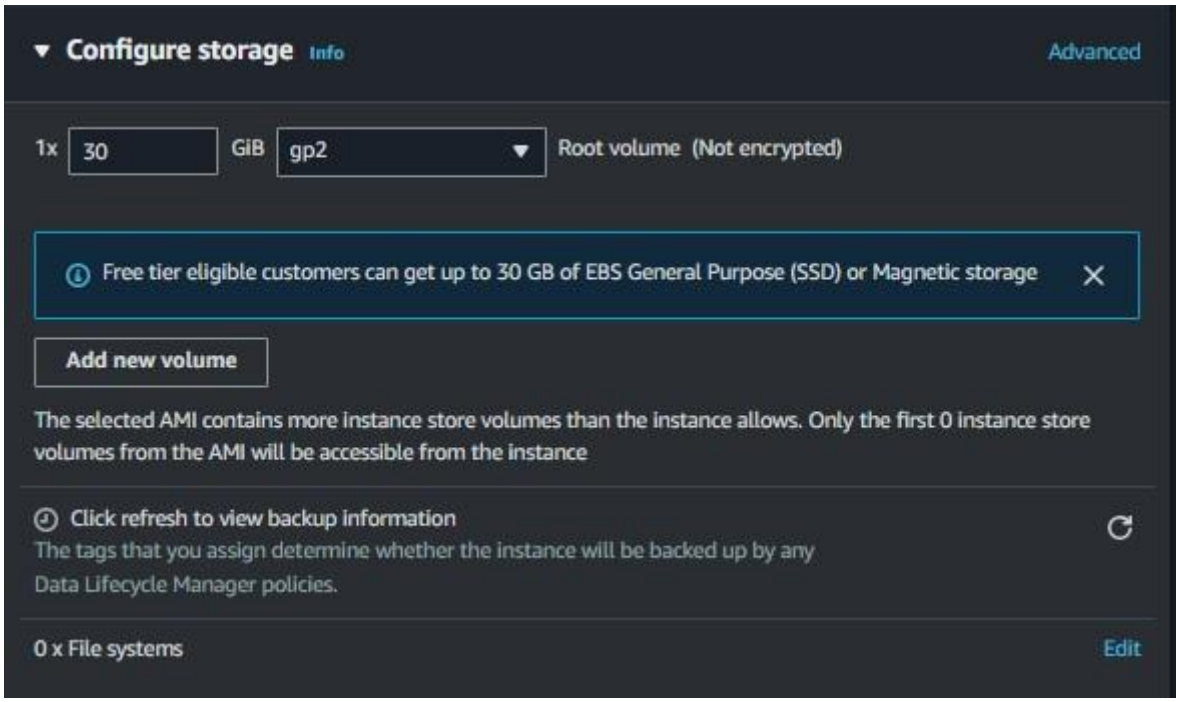
2.1 On the **Step 1: Choose an Amazon Machine Image (AMI)** page, find the **Windows AMI Free tier eligible AMI**, and then click on **Select**

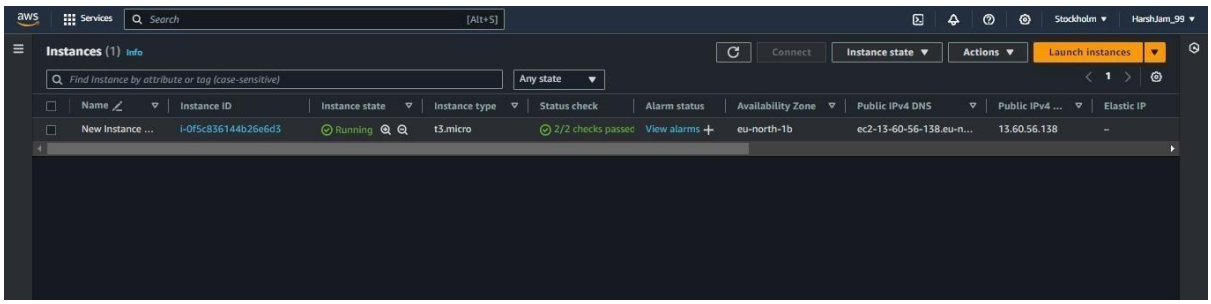


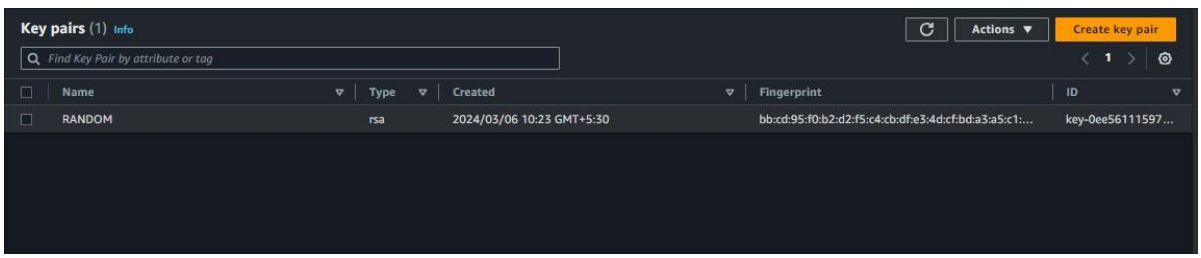

Instance type Info | Get advice
Instance type
t3.micro
Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0708 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software

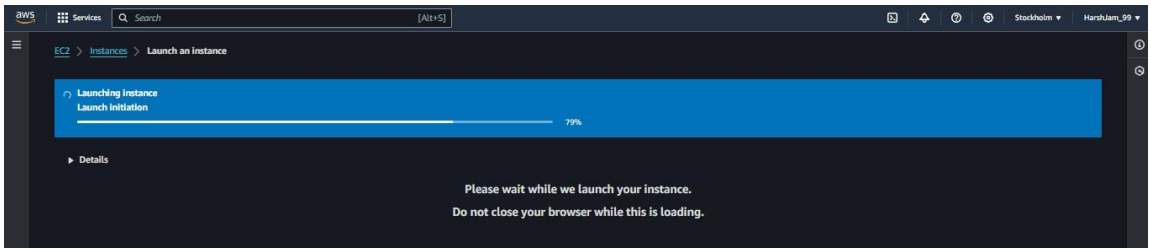

Network settings Info
Edit
Network Info
vpc-0e1f5570c469966ea
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-2' with the following rules:
Allow RDP traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

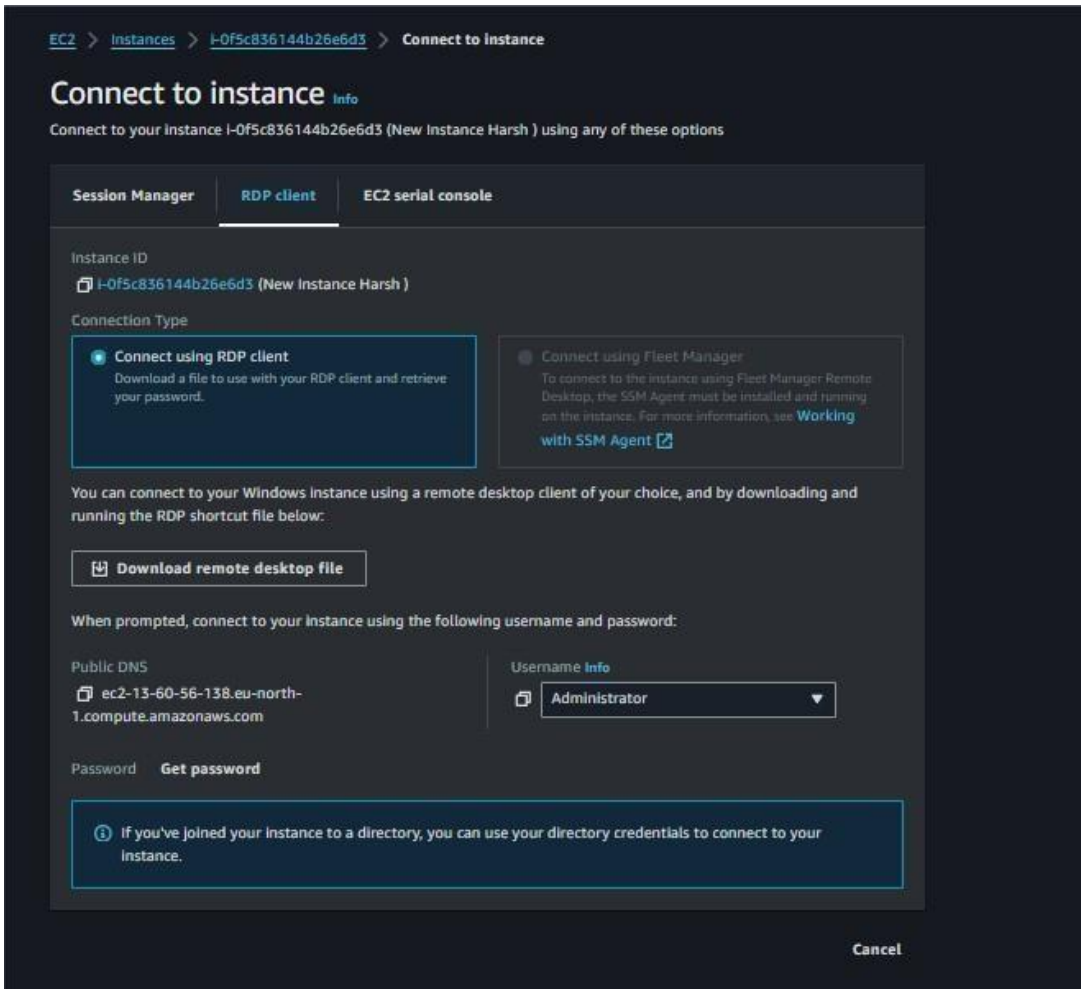
Subject: Cloud Computing and Web Services

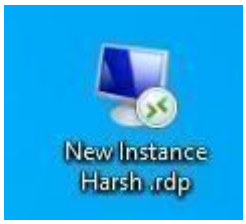


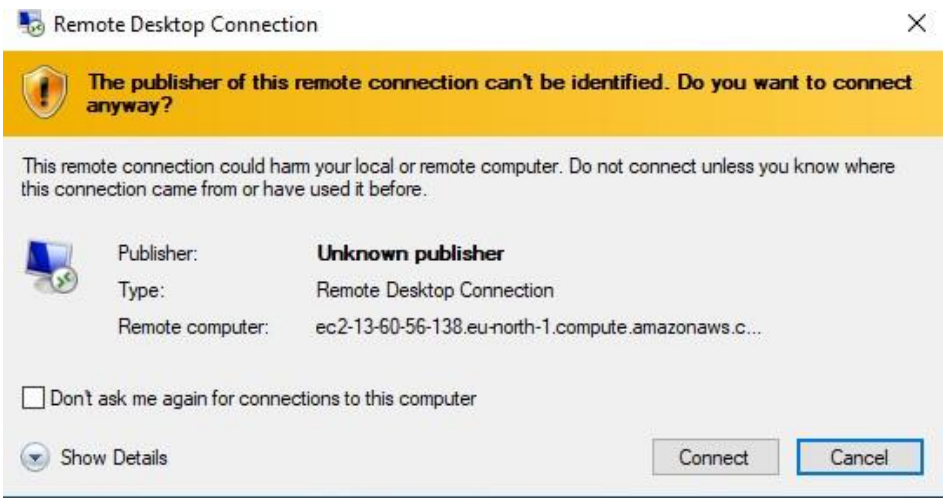




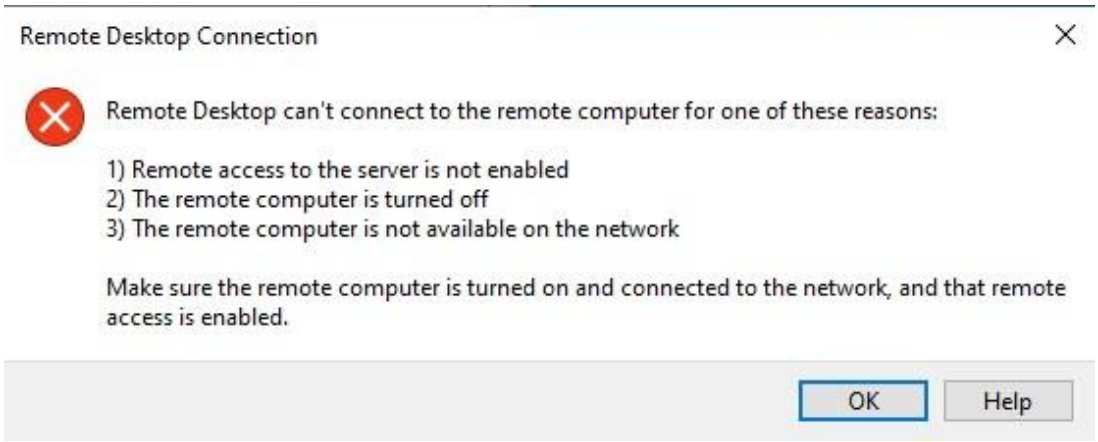



EC2 > Instances > i-0f5c836144b26e6d3 > Connect to instance
Connect to instance Info
Connect to your instance i-0f5c836144b26e6d3 (New Instance Harsh) using any of these options
Session Manager
RDP client
EC2 serial console
Instance ID
i-0f5c836144b26e6d3 (New Instance Harsh)
Connection Type
Connect using RDP client
Download a file to use with your RDP client and retrieve your password.
Connect using Fleet Manager
To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see Working with SSM Agent
You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:
Download remote desktop file
When prompted, connect to your instance using the following username and password:
Public DNS
ec2-13-60-56-138.eu-north-1.compute.amazonaws.com
Username Info
Administrator
Password
Get password
If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.
Cancel


New Instance
Harsh .rdp


Remote Desktop Connection
The publisher of this remote connection can't be identified. Do you want to connect anyway?
This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.
Publisher:
Unknown publisher
Type:
Remote Desktop Connection
Remote computer:
ec2-13-60-56-138.eu-north-1.compute.amazonaws.c...
Don't ask me again for connections to this computer
Show Details
Connect
Cancel

Subject: Cloud Computing and Web Services



PRACTICAL 10

Aim: Implementation of Openstack with user and private network creation.

There are a few questions that will help you narrow down the path you might want to take. First though, itsimportant to clarify that there are many parts of OpenStack that handle different situations. These part of OpenStack are called Projects. **You do not need to learn about all Projects at this time to get started creating a cloud**.

The number of Projects can be daunting, but in a typical starting setup to learn how to implement OpenStackyou will only use the following plus maybe one or two more based on your situation:

- OpenStack Nova – Virtual Machine based Compute
- OpenStack Neutron – private networking
- OpenStack Cinder – Block Storage
- OpenStack Horizon – dashboard for users and some administration
- OpenStack Keystone – user and API access permission management
- OpenStack Glance – VM images are stored here
- OpenStack Placement – tracks and controls resources and usage of them

With that out of the way, the next question. Are you learning how to build a private cloud with OpenStack foreducational purposes or because your company will need a cloud in the near future?

Steps to Install OpenStack

The steps required to install OpenStack are as follows:

Step 1: Install Virtual Box or Create Virtual Machine.

Download the Oracle virtual box and create the VM machine with a specific configuration of 64 bit OS with8GB RAM and 300 GB of memory. After creating your VM for a specific OS that you required, open the terminal and disable the firewall.

Step 2: Download the OpenStack version.

Use the below command to download the OpenStack version through the terminal. The command is ‘yuminstall –y centos-release-OpenStack-newton.’

Step 3: Update the packages.

Use the below command to update the package. The command is ‘yum update –y’.

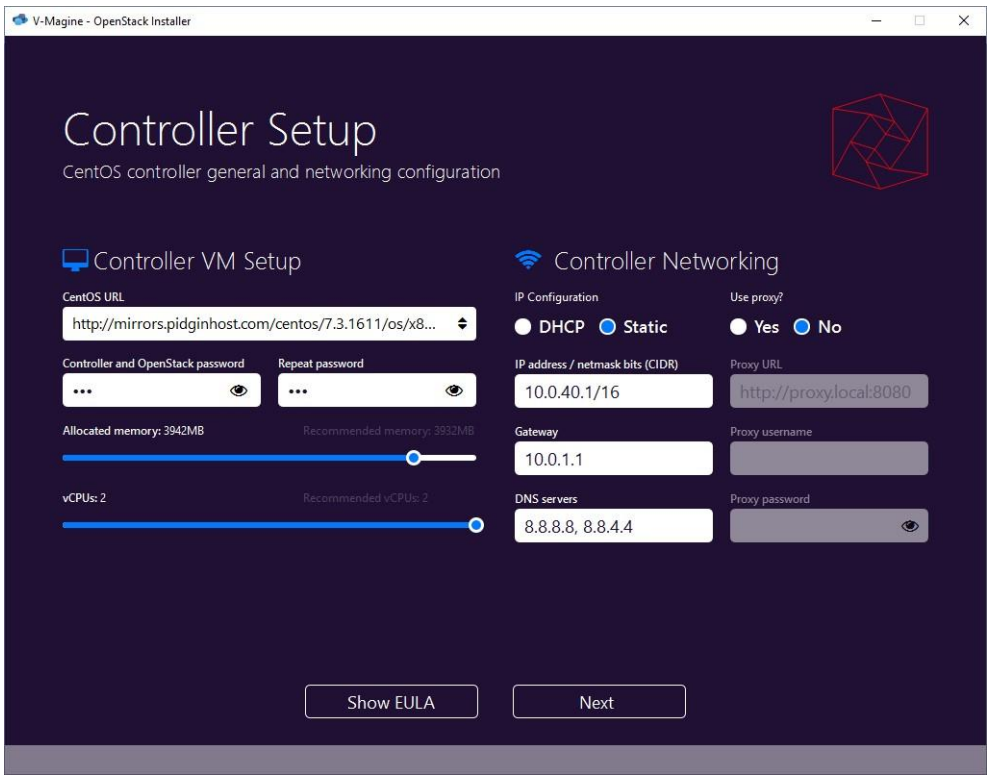
Step 4: Use the tool to install OpenStack.

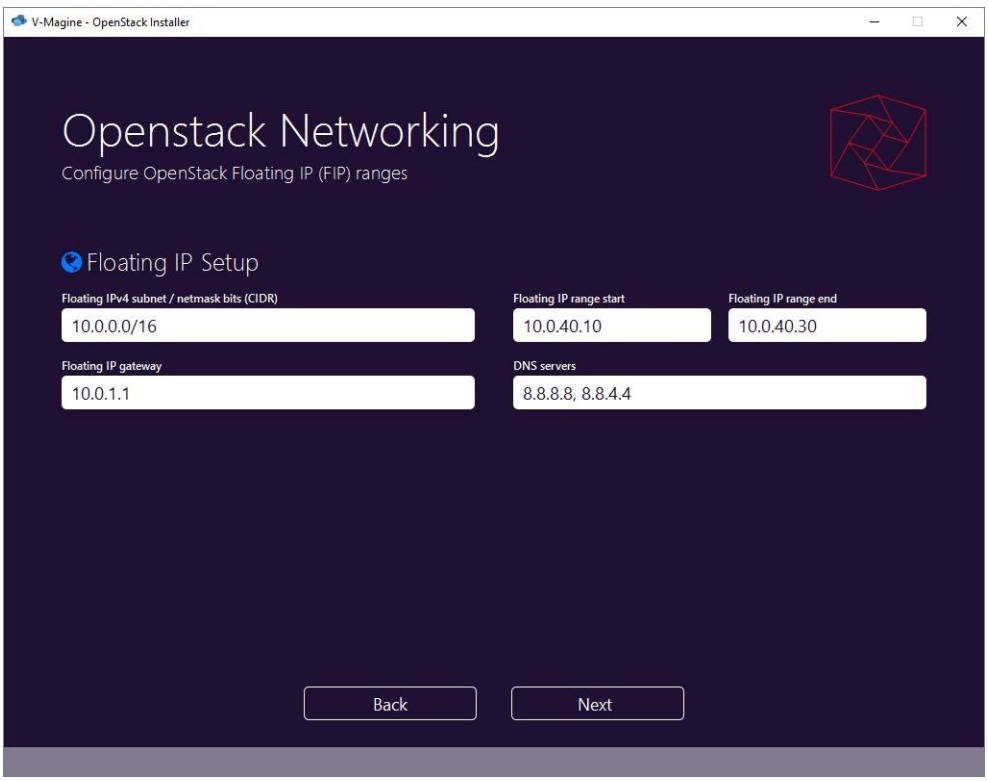
With the help of packstack, all the OpenStack services can be installed in one go, and it will take care of all theinstall part for OpenStack. It can be installed without the tool’s help as well, but it

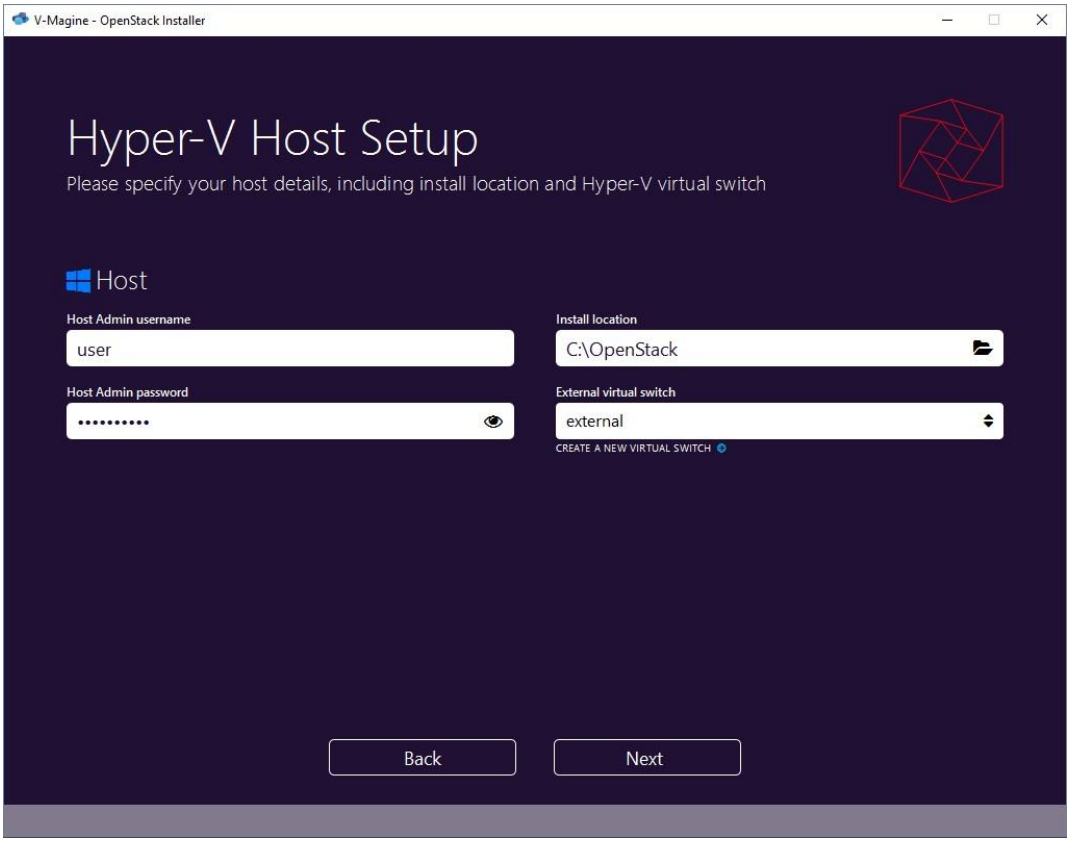
requires a lot of time to install the services one by one. The command is 'yum install –y OpenStack-packstack'.

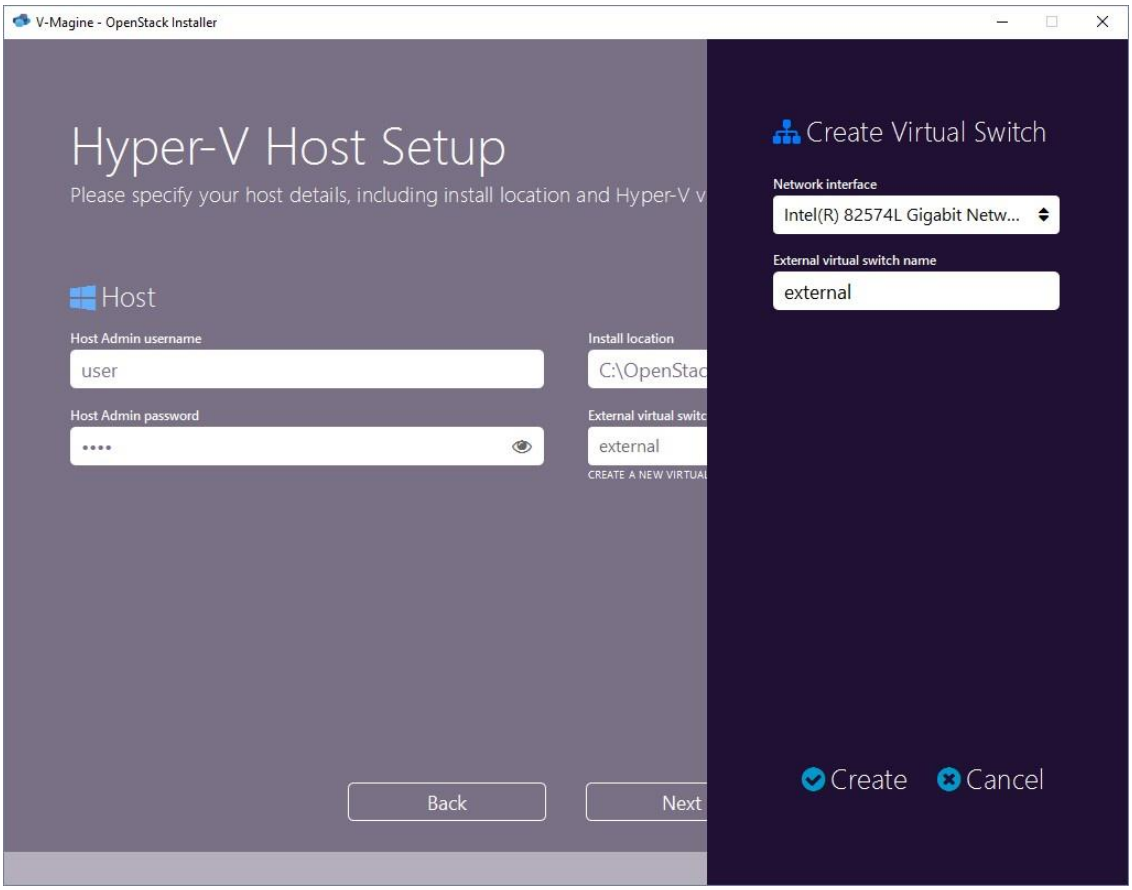
Step 5: Installing services.

The below command is used to install all the services or components for OpenStack. The command is'packstack –allinone'. It will install everything and installation complete for OpenStack.

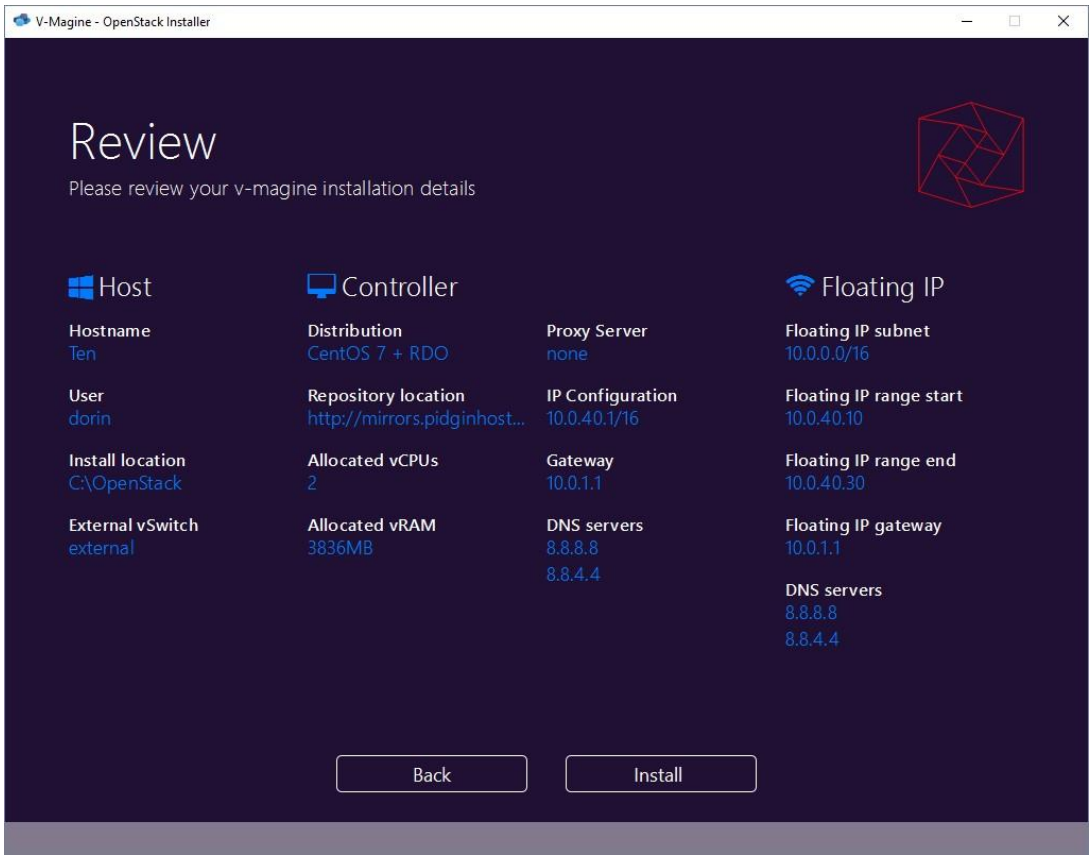


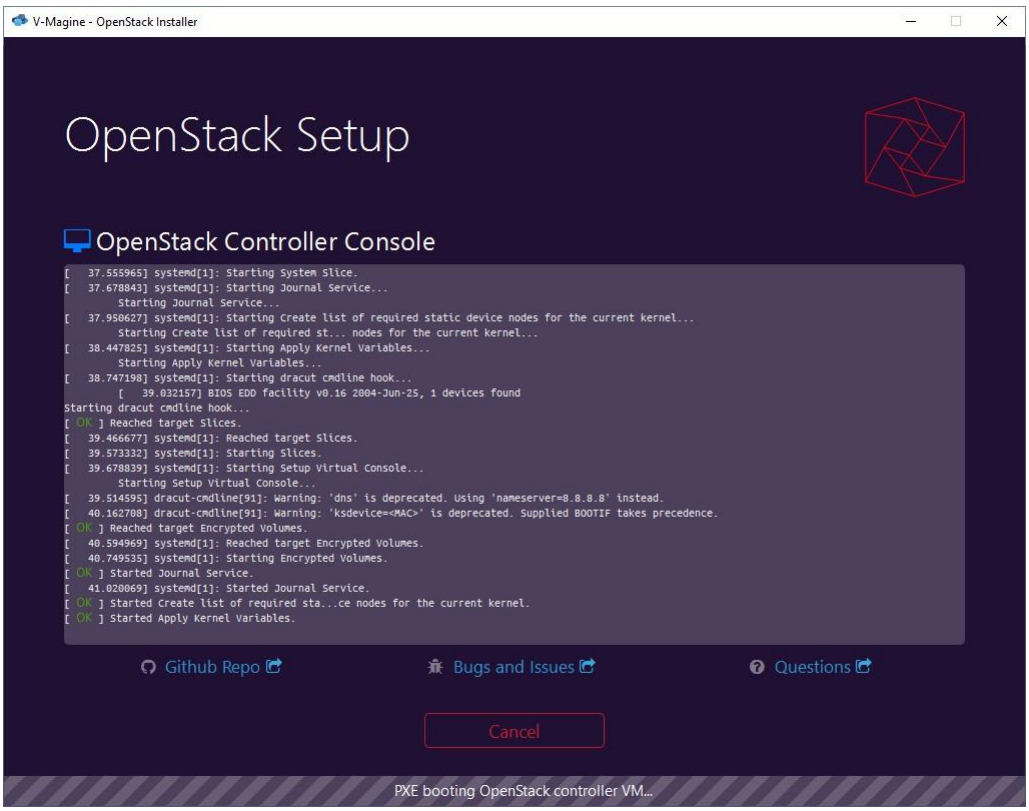


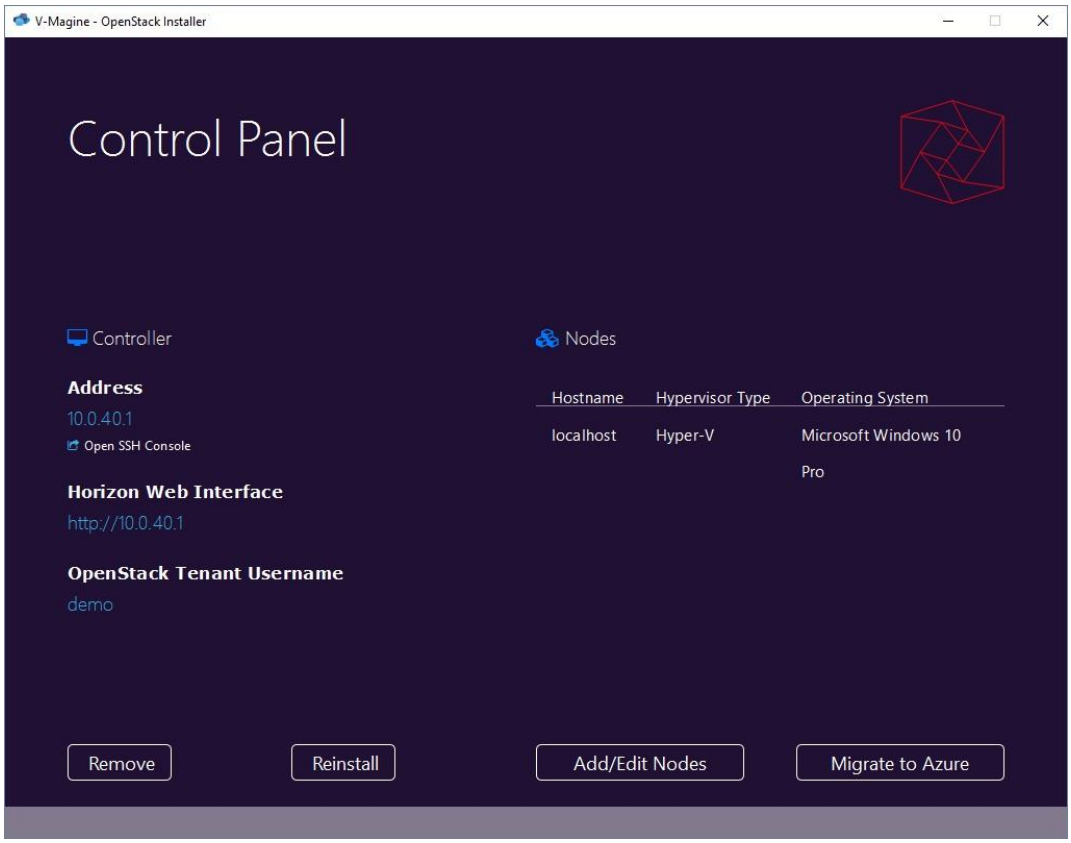

V-Magine - OpenStack Installer
Hyper-V Host Setup
Please specify your host details, including install location and Hyper-V virtual switch
Host
Host Admin username
user
Host Admin password
Install location
C:\OpenStack
External virtual switch
external
CREATE A NEW VIRTUAL SWITCH
Back
Next


V-Magine - OpenStack Installer
Hyper-V Host Setup
Please specify your host details, including install location and Hyper-V v
Host
Host Admin username
user
Host Admin password
Install location
C:\OpenStac
external
Back
Next
Create Virtual Switch
Network interface
Intel(R) 82574L Gigabit Netw...
External virtual switch name
external
Create
Cancel

Subject: Cloud Computing and Web Services







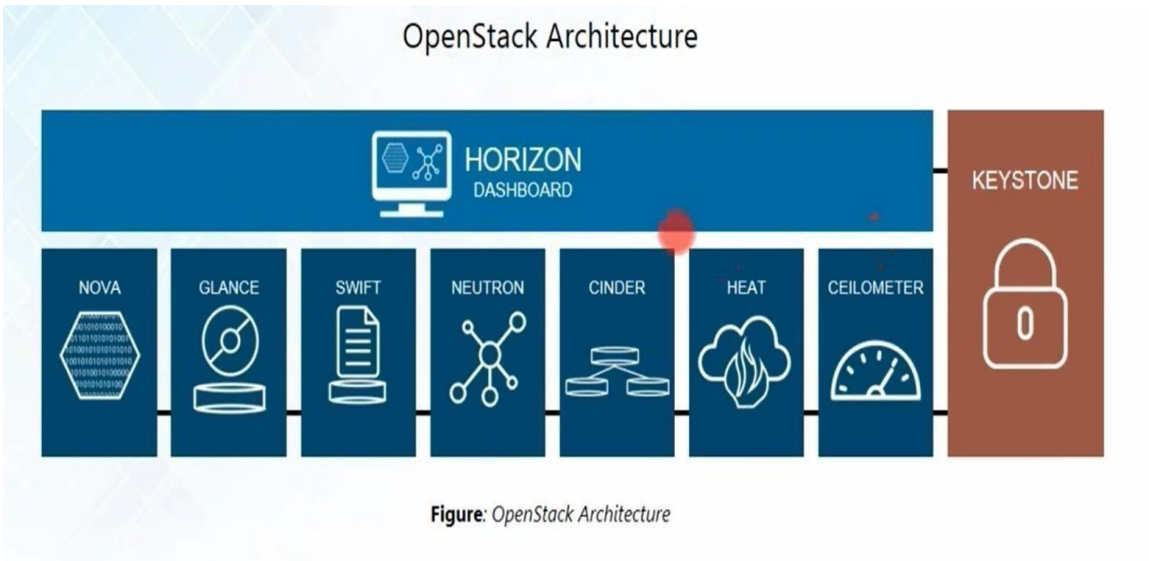
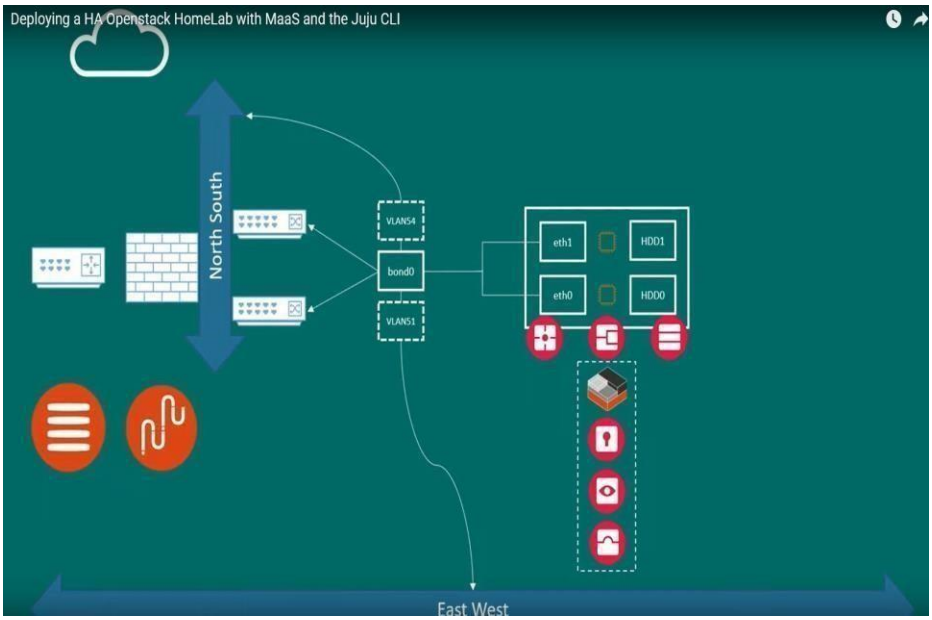
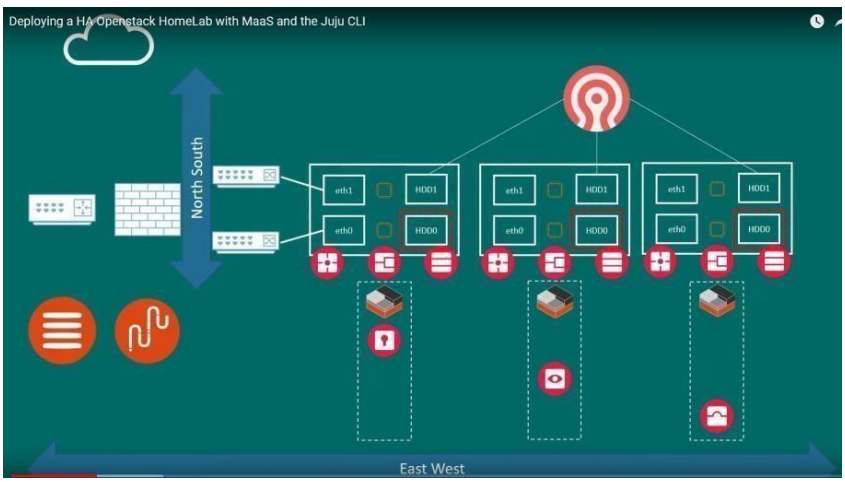


Figure: *OpenStack Architecture*

Subject: Cloud Computing and Web Services





FABRIC	VLAN	DHCP	SUBNET	AVAILABLE IPS	SPACE
homelab-fabric	untagged	MAAS-provided	192.168.50.0/24	49%	pxe-network
	49	MAAS-provided	192.168.49.0/24	76%	ipmi-space
	51	MAAS-provided	192.168.51.0/24	53%	admin-network
	52	MAAS-provided	192.168.52.0/24	53%	internal-network
	53	MAAS-provided	192.168.53.0/24	53%	public-network

(This script we're using is part of DevStack itself)

The script will install the listed features for your OpenStack environment –

- Horizon – OpenStack Dashboard
- Keystone – Identity Service
- Nova – Compute Service
- Glance – Image Service
- Neutron – Network Service
- Placement – Placement API
- Cinder – Block Storage Service