|  |  |
| --- | --- |
| **Roll No.:** A059 | **Name:** Chinmay Parikh |
| **Prg/Yr/Sem:** B.Tech(I.T.)/4th /7 | **Batch:** A3 |
| **Date of Experiment:** 6/9/2014 | **Date of Submission: 29**/10/2014 |

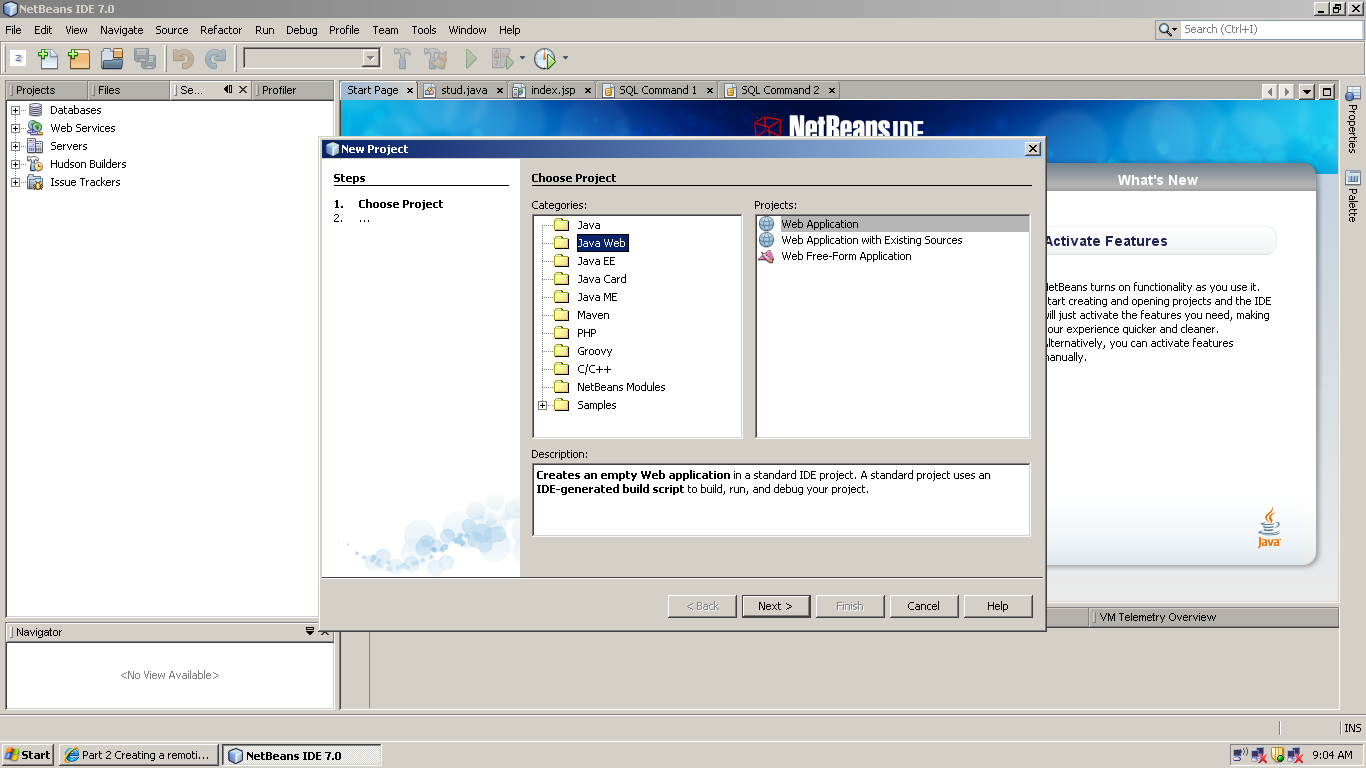
**Aim:** Creating a Java Web Service Application which uses web services provided by third parties (Google Maps).

**Scenario:** Creating a Java Web Service Application which uses web services provided by third parties (GoogleMaps).

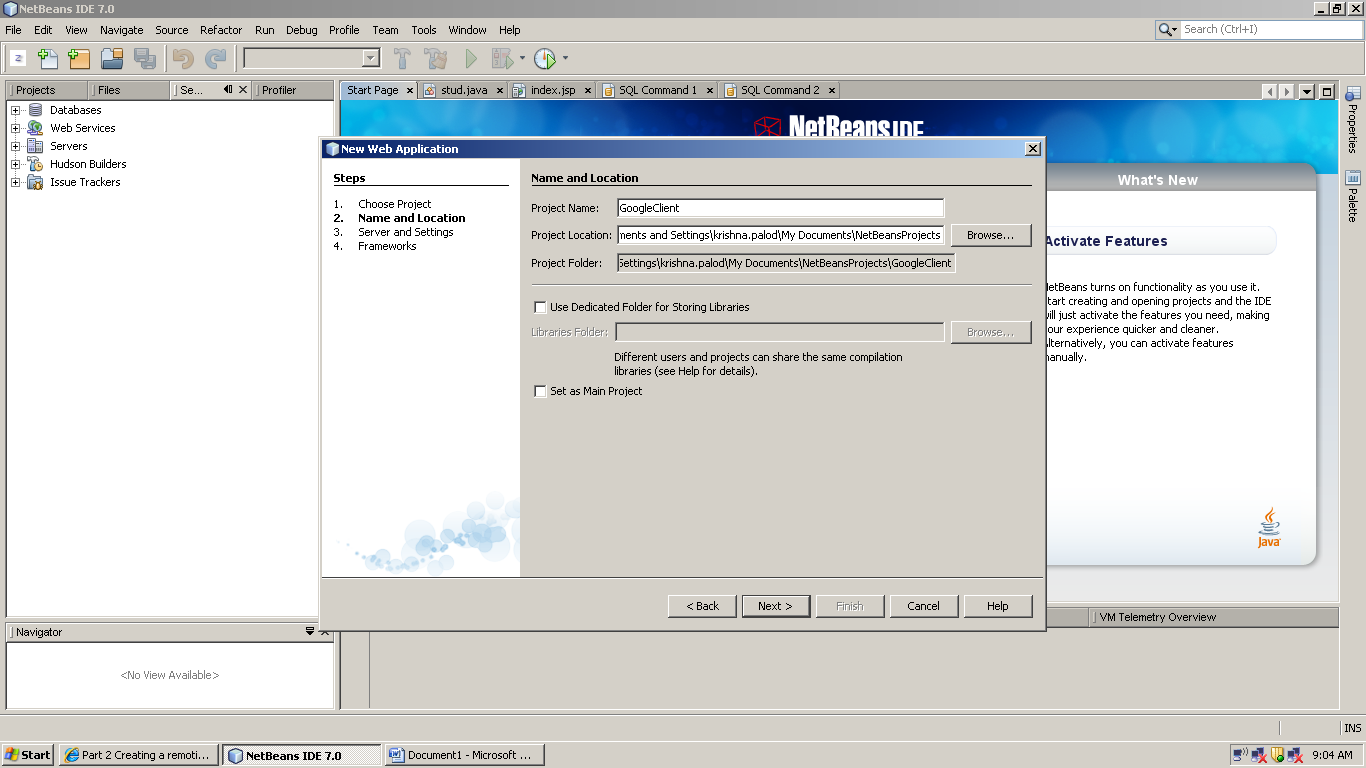
**Detailed Steps for creating a web service & web service client in Net Beans 7.0.**

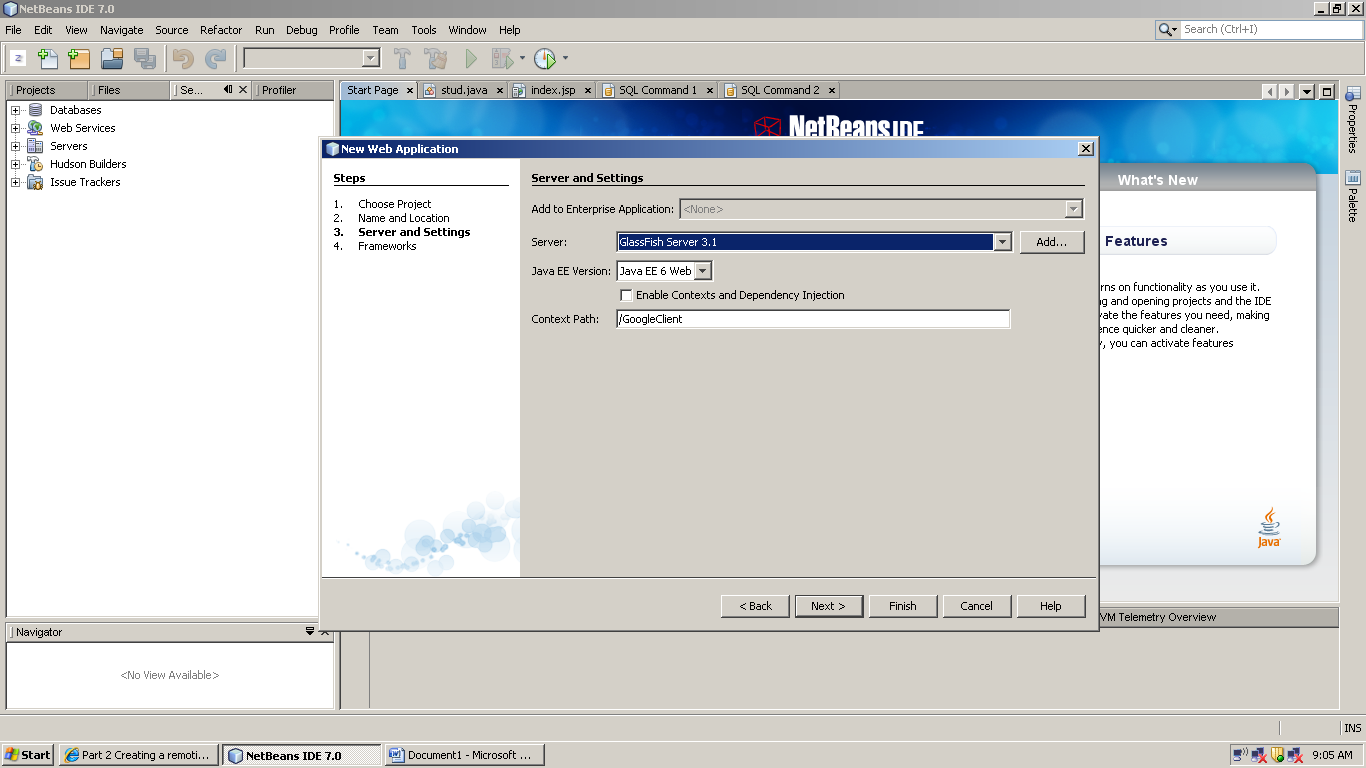
**Creating the Java Web Project :**

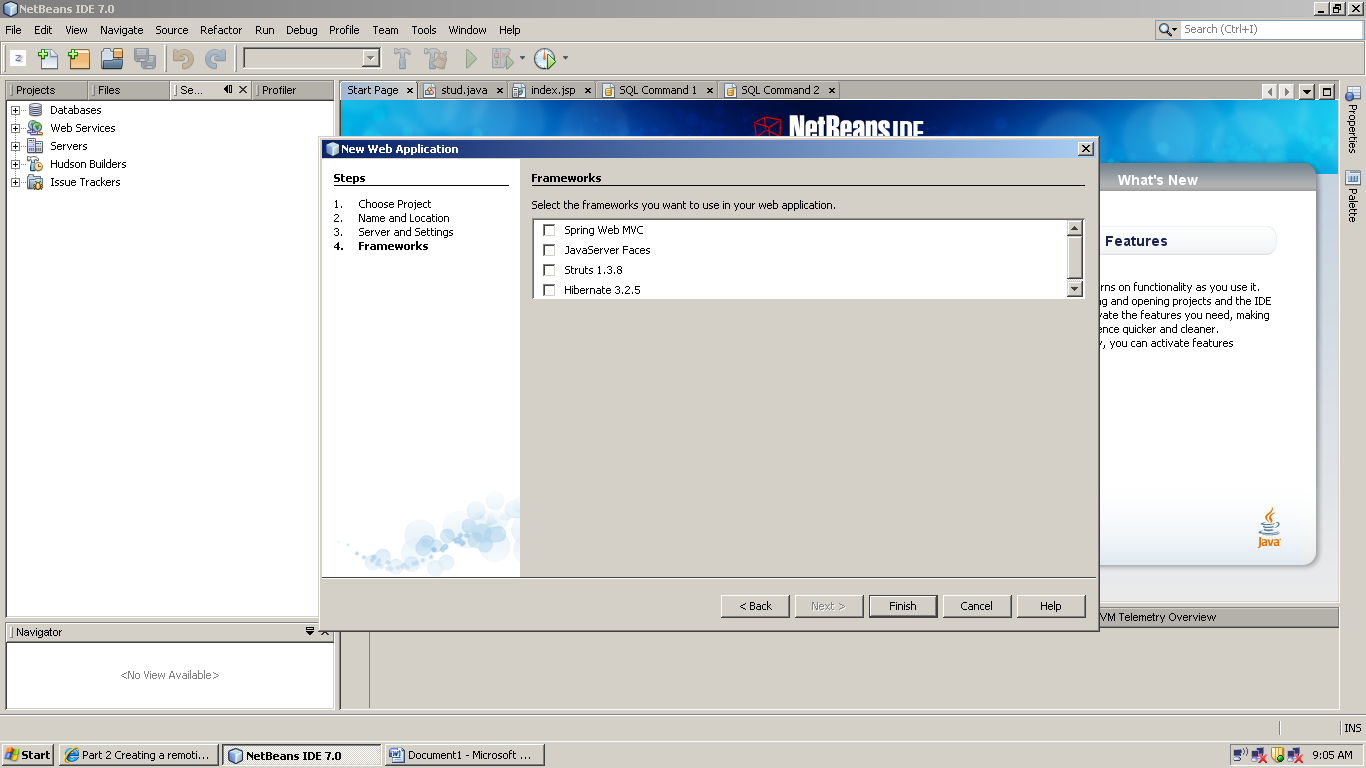
1. Click **File** and then **New Project**.
2. For **New Project**: On the **Categories** side, choose **Java Web** and on the **Projects side**, select **WebApplication**.
3. Click **Next**.
4. For **Name and Location**, under Project Name, enter **WebServiceClient**.
5. Tick the box on **Use Dedicated Folder for Storing Libraries**.
6. Now, either type the folder path or select one by clicking on **browse**.
7. After choosing the folder, we can proceed by clicking **Next**.
8. For **Server and Settings**: Under **Server**, choose **GlassFish Server 3.1**.
9. Leave the other options with their default values and click **Finish**.

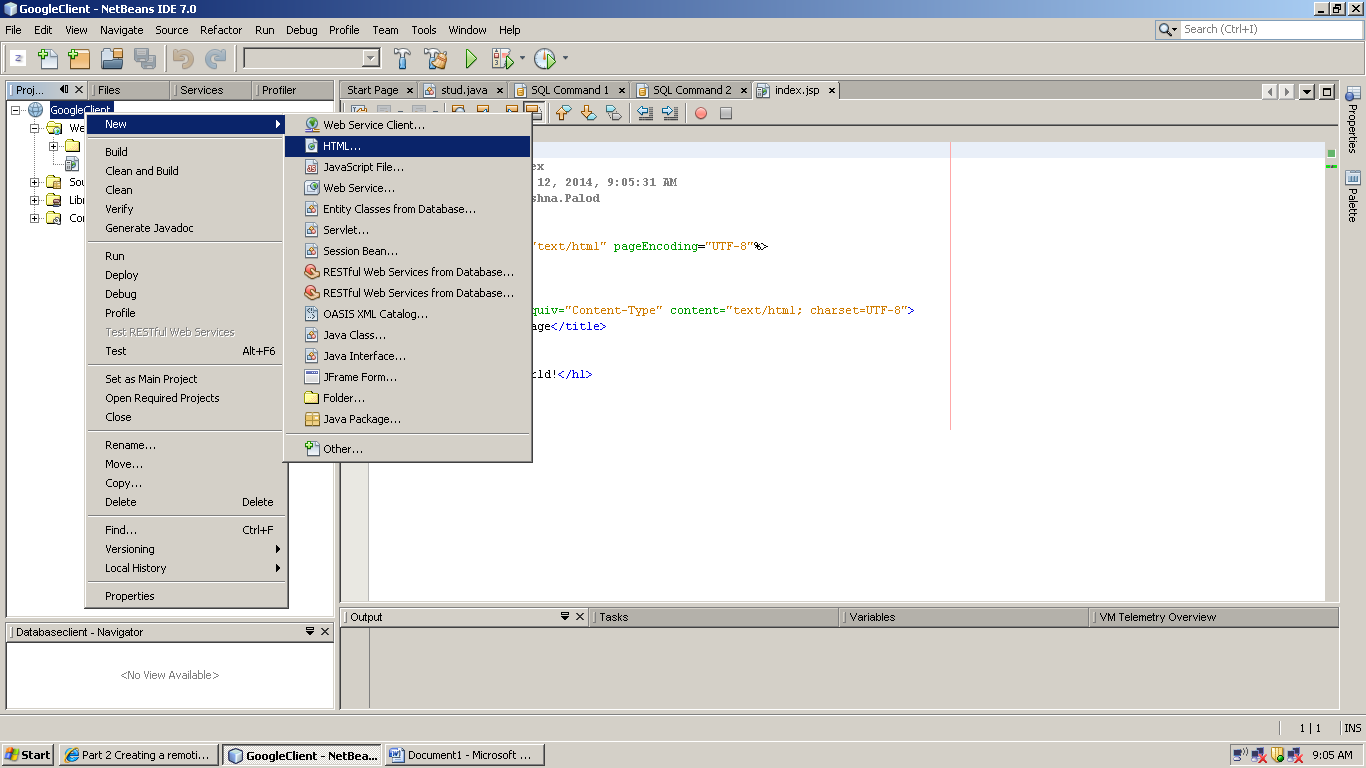


Create a Web Application called GoogleClient.

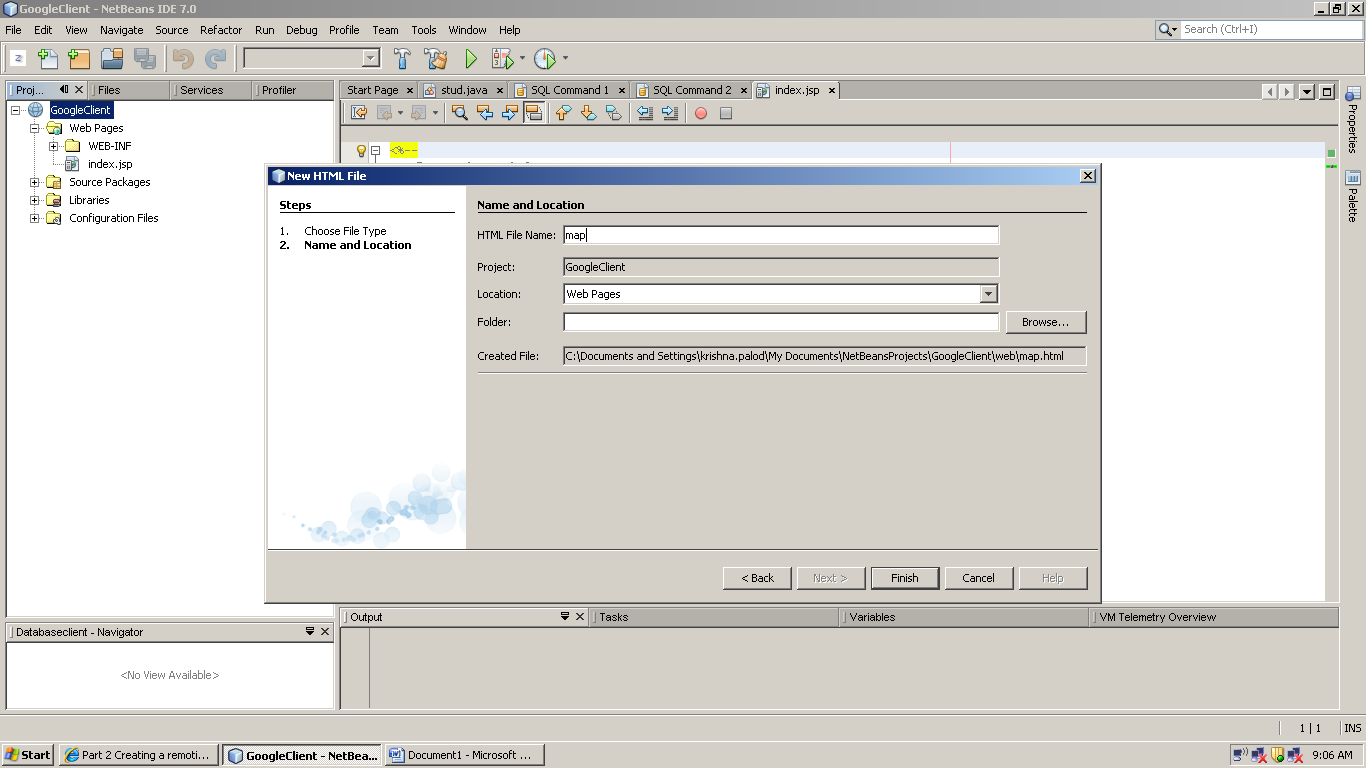


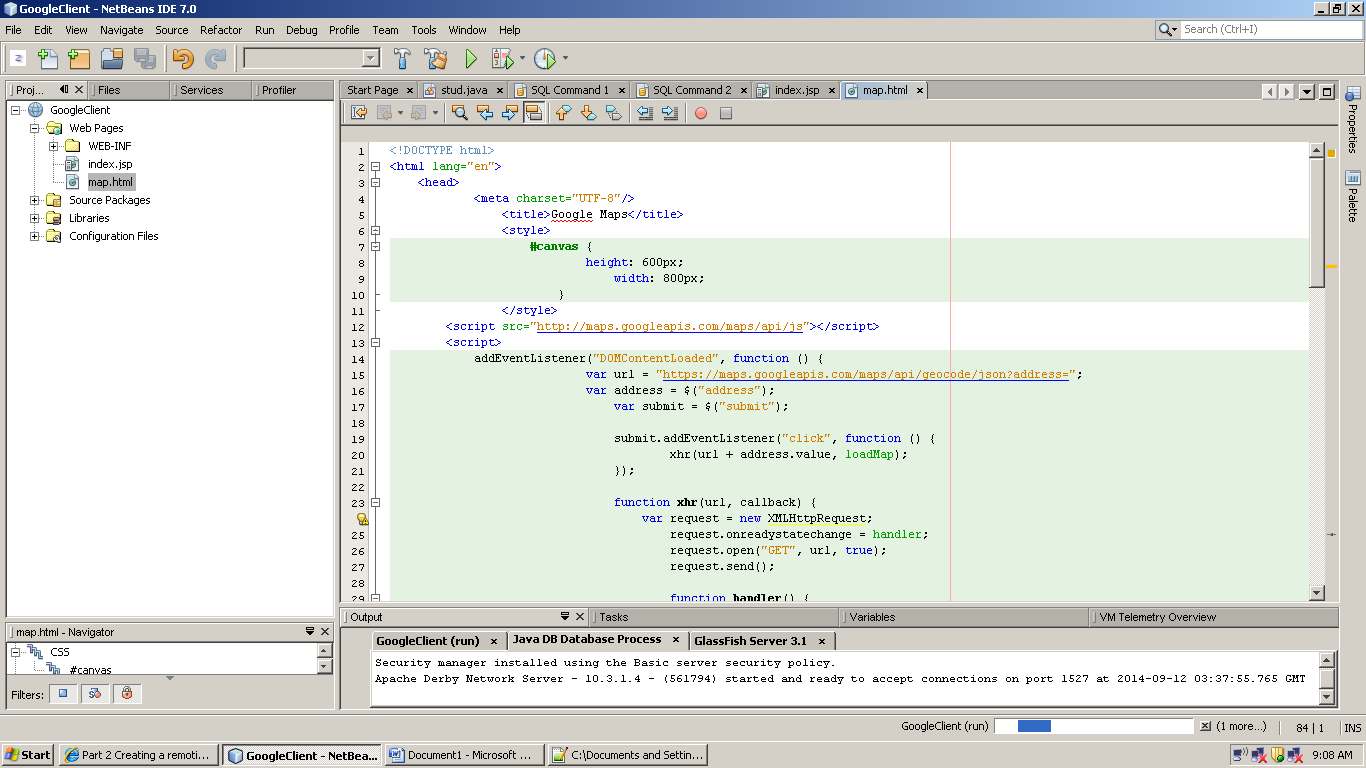


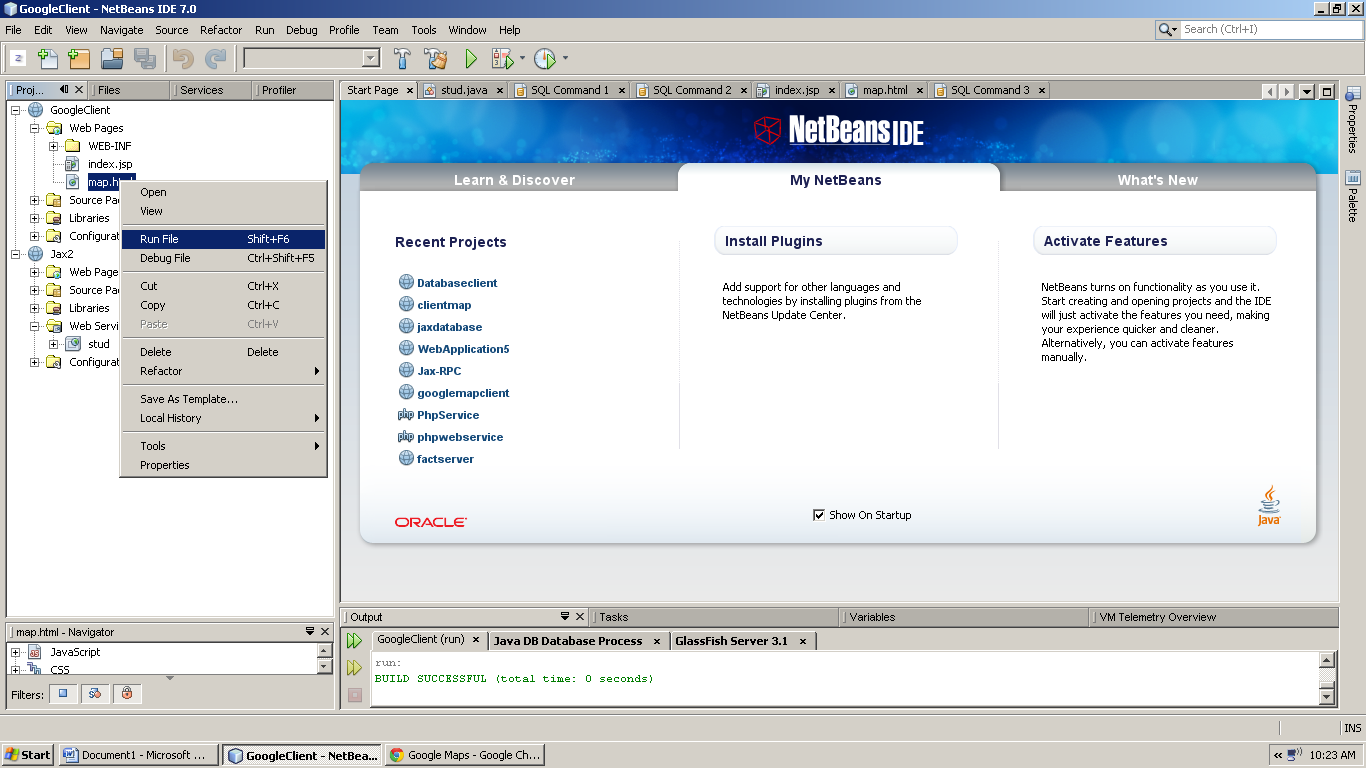




Create a HTML web page called map.html.







The default browser should be Google Chrome or higher version of Internet Explorer i.e. IE8.0.

**Code:**

**map.html :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8"/>

<title>Google Maps</title>

<style>

#canvas {

height: 600px;

width: 800px;

}

</style>

<script src="http://maps.googleapis.com/maps/api/js"></script>

<script>

addEventListener("DOMContentLoaded", function () {

var url = "https://maps.googleapis.com/maps/api/geocode/json?address=";

var address = $("address");

var submit = $("submit");

submit.addEventListener("click", function () {

xhr(url + address.value, loadMap);

});

function xhr(url, callback) {

var request = new XMLHttpRequest;

request.onreadystatechange = handler;

request.open("GET", url, true);

request.send();

function handler() {

if (request.readyState === 4 && request.status === 200)

callback(request.responseText);

}

}

function loadMap(result) {

var json = JSON.parse(result);

var location = json.results[0].geometry.location;

var lat = location.lat;

var lng = location.lng;

canvas.innerHTML = "";

var position = new maps.LatLng(lat, lng);

var map = new maps.Map(canvas, {

mapTypeId: maps.MapTypeId.HYBRID,

center: position,

zoom: 18

});

var marker = new maps.Marker({

title: json.results[0].formatted\_address,

position: position,

map: map

});

}

var maps = google.maps;

var canvas = $("canvas");

var position = new maps.LatLng(18.9750, 72.8258);

var map = new maps.Map(canvas, {

mapTypeId: maps.MapTypeId.HYBRID,

center: position,

zoom: 18

});

var marker = new maps.Marker({

title: "",

position: position,

map: map

});

function $(id) {

return document.getElementById(id);

}

});

</script>

</head>

<body>

<label for="address">Address:</label> <input type="text" id="address"/> <button id="submit">Submit</button>

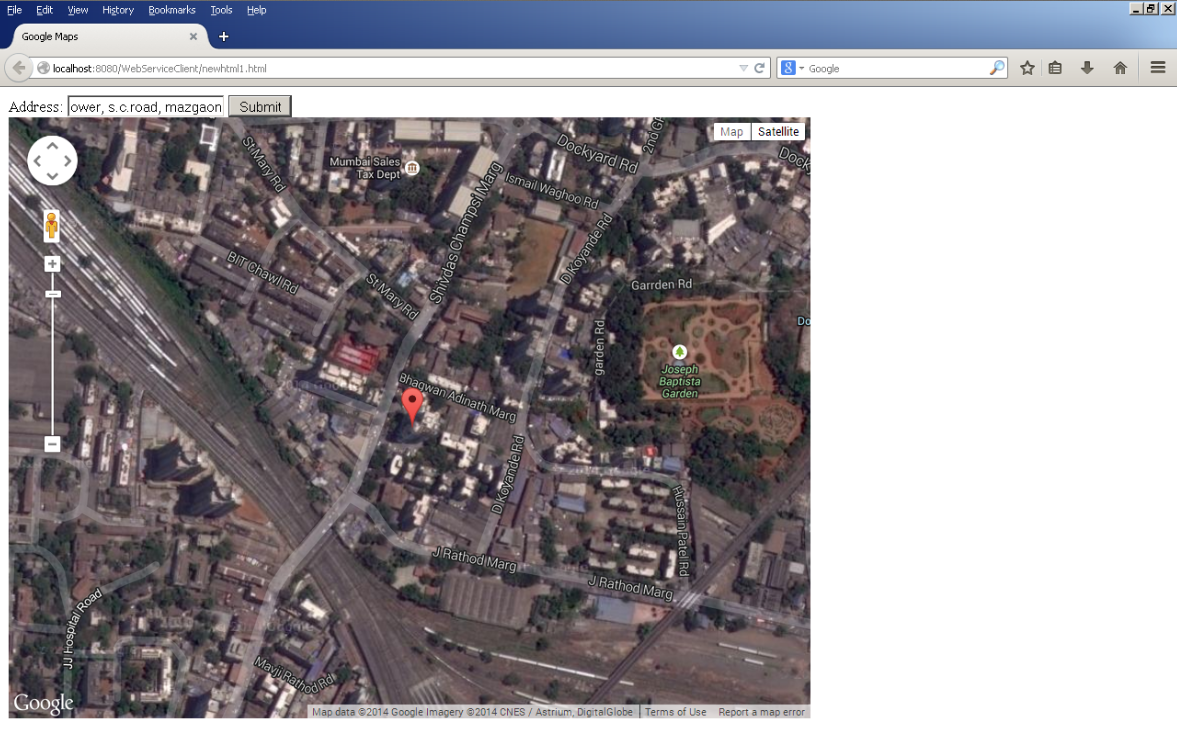
<div id="canvas"></div>

</body>

</html>

**Printout:**

* **Snapshot of google maps output.**



**Conclusion:** Thus we have identified the need for accessing a web service within an application and also created a client web service to access the online web service.