Practical 4

Develop application to consume Google’s search / Google’s Map RESTful Web service.

To create a simple application that searches for a place using the Google Places API and retrieves the details from the Google Maps Geocoding API. Steps are as follows:

**Software Tools Required**

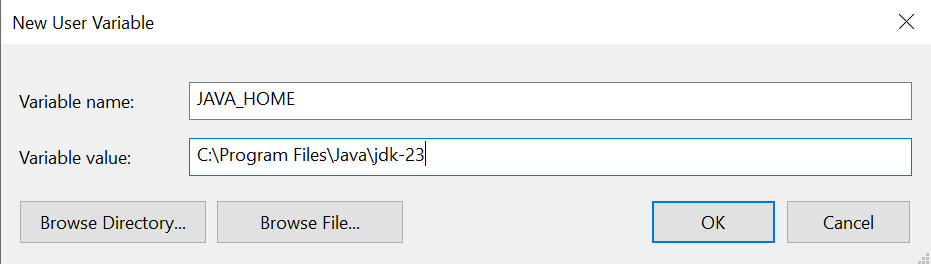
* **Code Editor:** VS Code
* **Gmail:** Before Starting this Practical, your personal Gmail account must be logged in the web browser.
* **Google Cloud SDK**: For managing Google APIs. [Free Trial and Free Tier Services and Products | Google Cloud](https://cloud.google.com/free/?utm_source=google&utm_medium=cpc&utm_campaign=japac-IN-all-en-dr-BKWS-all-core-athena-EXA-dr-1605216&utm_content=text-ad-none-none-DEV_c-CRE_644159077391-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+-GCP-General-core+brand-main-KWID_43700081107836716-kwd-26415313501&userloc_9062244-network_g&utm_term=KW_google+cloud+platform&gad_source=1&gclid=Cj0KCQiA9667BhDoARIsANnamQZDVr01vSn4N_YPQk9tzr7YlxN7h8y52xilRPwHTHDw2cj8DD1LpFYaAicfEALw_wcB&gclsrc=aw.ds&hl=en)

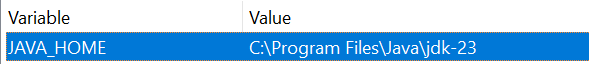
**Downloads Required:**

* JDK: [Java Downloads | Oracle India](https://www.oracle.com/in/java/technologies/downloads/#jdk23-windows) (x64 MSI Installer)
* Apache Tomcat: [Apache Tomcat Downloads](https://tomcat.apache.org/download-90.cgi). (x64 bit Windows zip)

After Downloading JDK, Set the Path in Environment Variables in User Variables > Path > New and Provide the Variable name and values to set the Path for JDK as shown below:

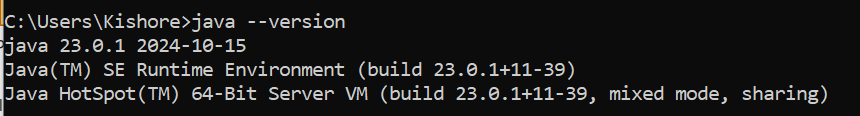
**Demonstration:**

****

****

Click OK to remaining opened Edit environment variable, Environment Variables and System Properties windows.

Now check the versions that shows path is set for Java JDK

****

**Step 1:** Download and Extract apache-tomcat-9.0.98 zip file into folder, and open apache-tomcat-9.0.98 folder till bin and select the directory path, cut (press backspace) and type **cmd.**

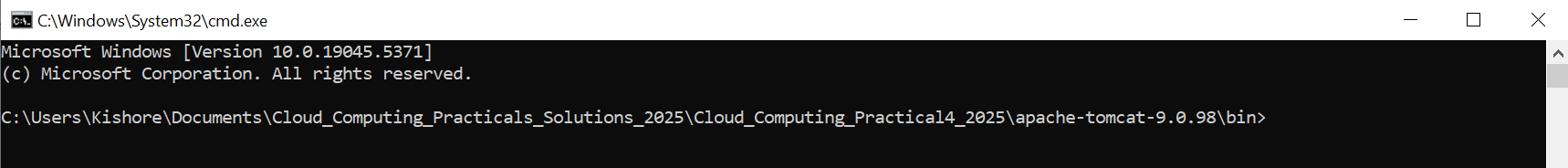
**Demonstration:**

**Directory Path (Your directory path depends where you place your apache-tomcat folder)**

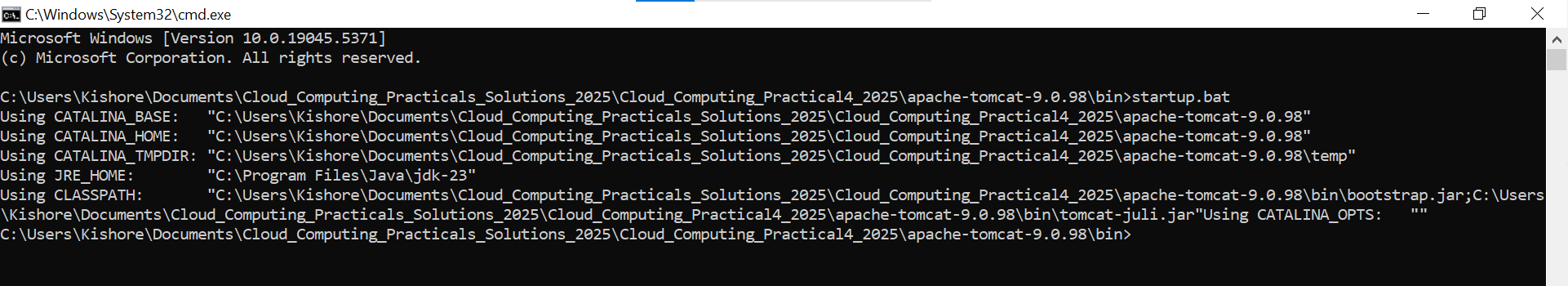
**Select directory path**



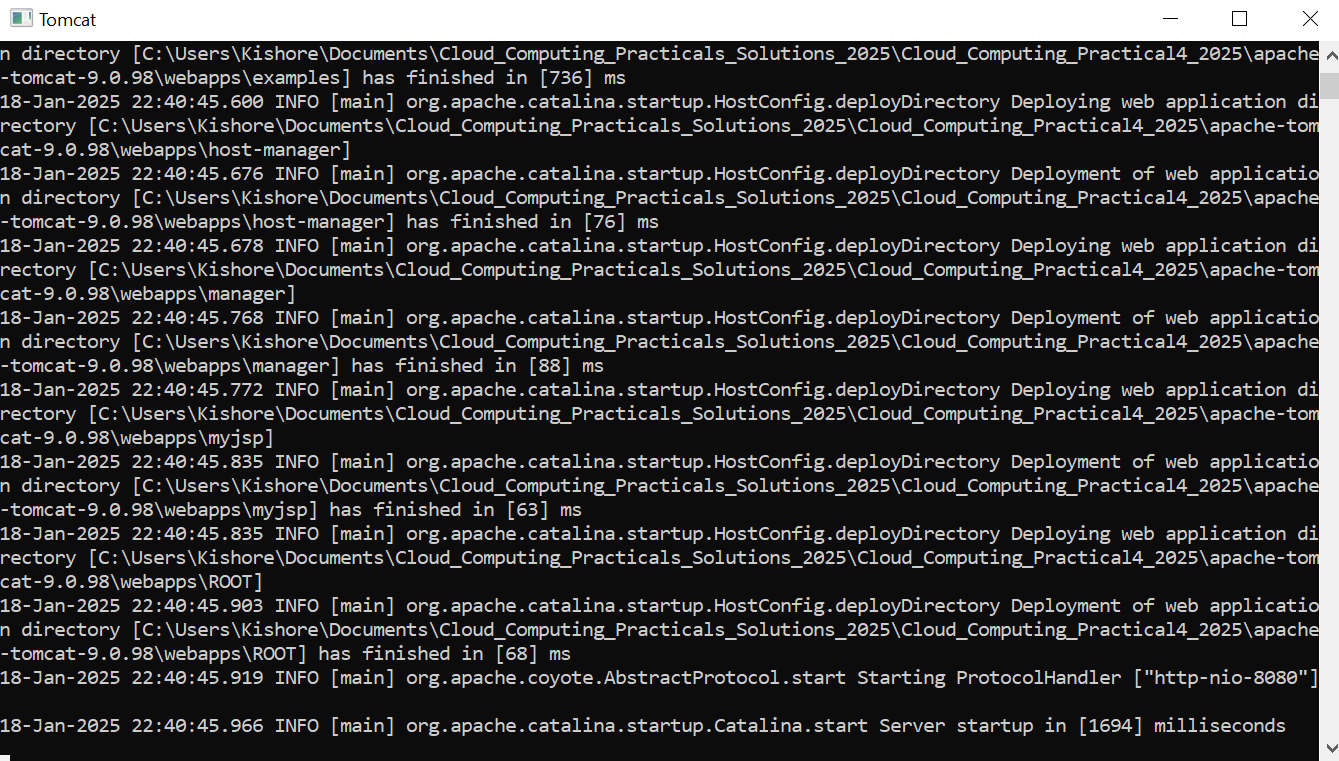
Type cmd and command prompt will open to the same directory path



Now type: **startup.bat**

****

We will get this tomcat loaded

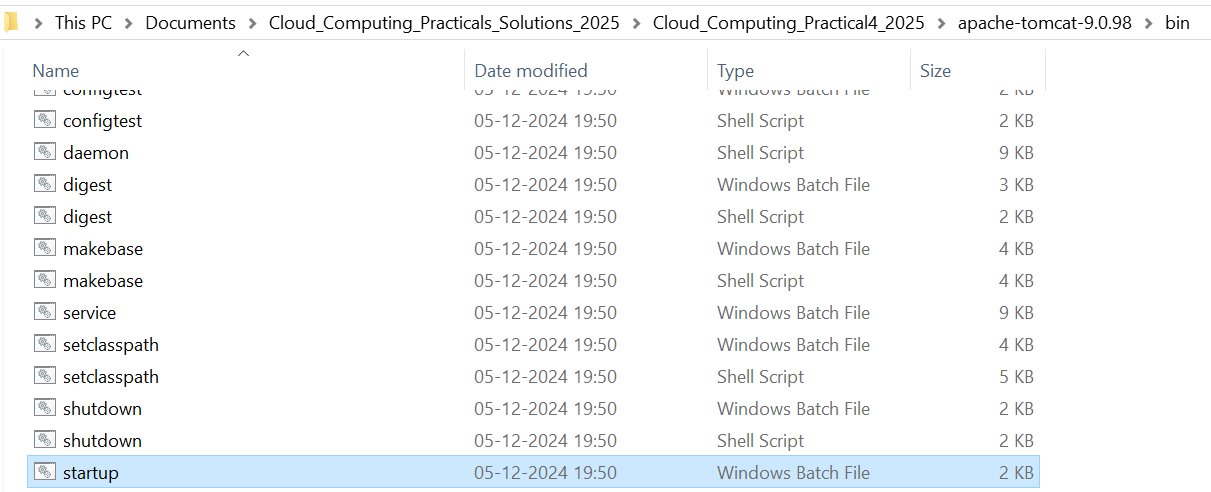
****

**Note:**

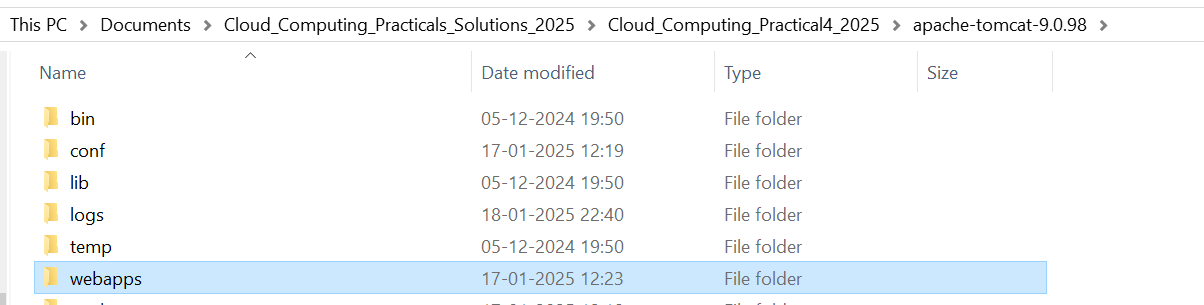
**This Tomcat Server must be kept running in the background by minimizing this window. Otherwise, the we won’t get the output.**

**Another way to load Tomcat**

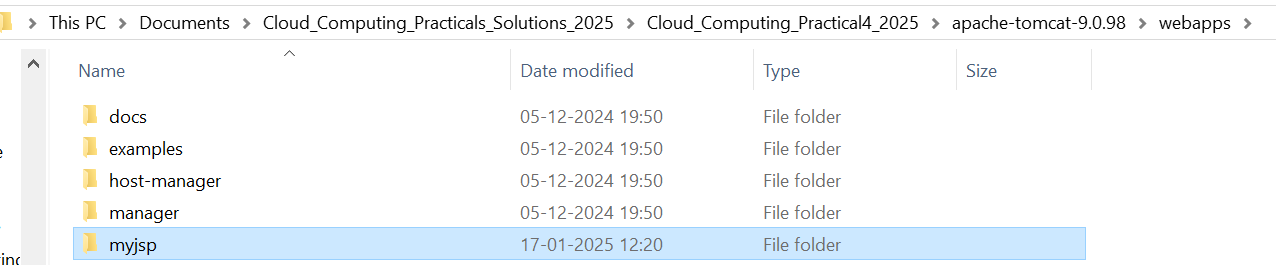
Also, in bin folder we will have startup as Windows Batch File, double click Run and it will load Tomcat.

****

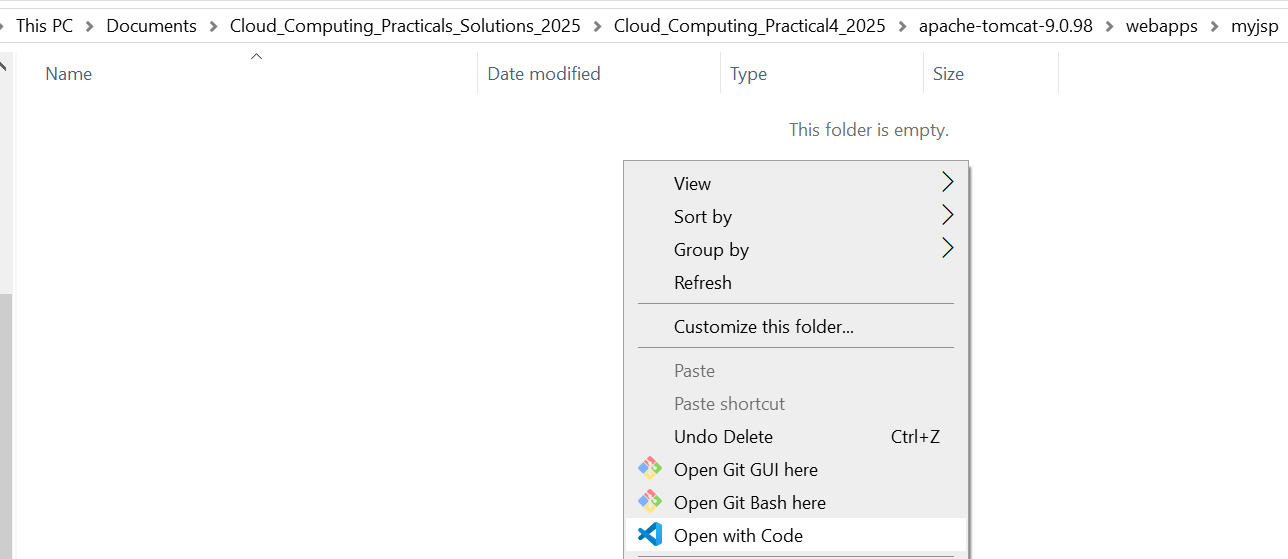
**Step 2:** After completing Step 1, Now open apache-tomcat-9.0.98 folder, open a folder name **webapps** which is already created in apache-tomcat-9.0.98 folder

****

Open **webapps** folder, create a new folder and name it as: **myjsp**

****

**Step 3:** Open **myjsp** folder and open it with VS Code



Create 2 jsp files with name **myindex.jsp** and **myinput.jsp**

* 1. **Filename:** myindex.jsp : **Display Map with Google Maps API**

**Code:**

<%@ page contentType="text/html" pageEncoding="UTF-8" %>

<!DOCTYPE html>

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Google Maps Location</title>

    <!-- Google Maps JavaScript API -->

    <script src="https://maps.googleapis.com/maps/api/js?key=**YOUR\_API\_KEY** &callback=initMap" async defer></script>

    <style>

        /\* Set the size of the map container \*/

        #map {

            height: 400px;

            width: 100%;

        }

    </style>

</head>

<body>

<%

    // Fetch latitude and longitude from form inputs

    String latParam = request.getParameter("t1");

    String longParam = request.getParameter("t2");

    // Default values in case parameters are not provided

    double lati = 40.7128;  // Default latitude (New York)

    double longi = -74.0060; // Default longitude (New York)

    // If parameters exist, parse them to doubles

    if (latParam != null && !latParam.trim().isEmpty()) {

        lati = Double.parseDouble(latParam);

    }

    if (longParam != null && !longParam.trim().isEmpty()) {

        longi = Double.parseDouble(longParam);

    }

%>

<h3>Google Maps Location</h3>

<div id="map"></div>

<script>

    // Initialize the map

    function initMap() {

        var location = { lat: <%= lati %>, lng: <%= longi %> };  // Use JSP values for lat and lng

        // Create a new map centered on the location

        var map = new google.maps.Map(document.getElementById('map'), {

            zoom: 10,  // Zoom level

            center: location  // Center the map on the provided coordinates

        });

        // Add a marker at the location

        var marker = new google.maps.Marker({

            position: location,

            map: map

        });

    }

</script>

</body>

</html>

* 1. **Filename**: myinput.jsp : **Form to Get Latitude and Longitude**

**Code:**

<%@ page contentType="text/html" pageEncoding="UTF-8" %>

<!DOCTYPE html>

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Enter Latitude and Longitude</title>

</head>

<body>

    <!-- Form to accept input values for latitude and longitude -->

    <form action="myindex.jsp" method="get">

        <pre>

            Enter latitude: <input type="text" name="t1" />

            Enter longitude: <input type="text" name="t2" />

            <input type="submit" value="Show" />

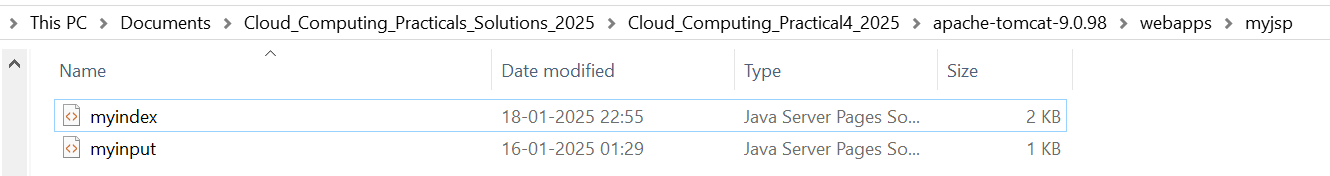
        </pre>

    </form>

</body>

</html>

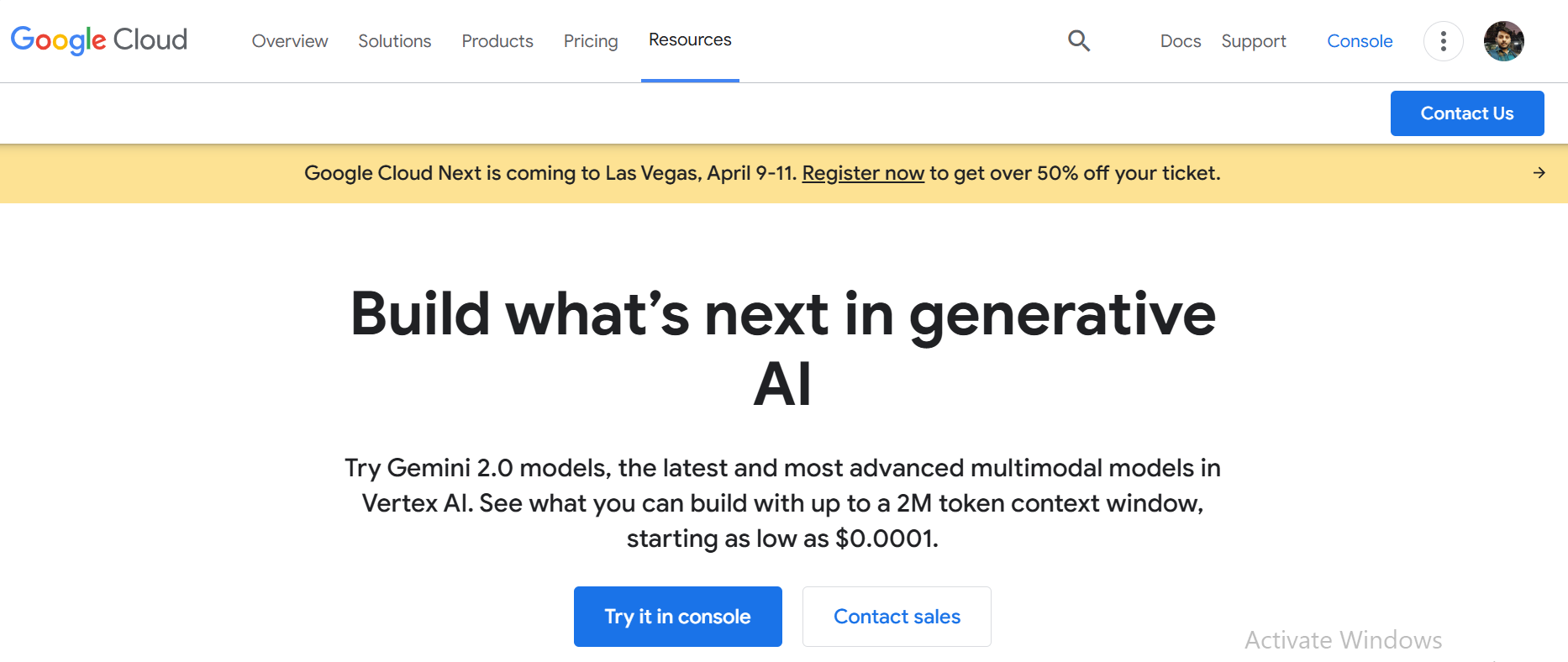
Your **myjsp** folder will look like this



**Step 4:** Open Google Cloud Platform Console,

[Free Trial and Free Tier Services and Products | Google Cloud](https://cloud.google.com/free/?utm_source=google&utm_medium=cpc&utm_campaign=japac-IN-all-en-dr-BKWS-all-core-athena-EXA-dr-1605216&utm_content=text-ad-none-none-DEV_c-CRE_644159077391-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+-GCP-General-core+brand-main-KWID_43700081107836716-kwd-26415313501&userloc_9062244-network_g&utm_term=KW_google+cloud+platform&gad_source=1&gclid=Cj0KCQiA9667BhDoARIsANnamQZDVr01vSn4N_YPQk9tzr7YlxN7h8y52xilRPwHTHDw2cj8DD1LpFYaAicfEALw_wcB&gclsrc=aw.ds&hl=en)

Click on Console

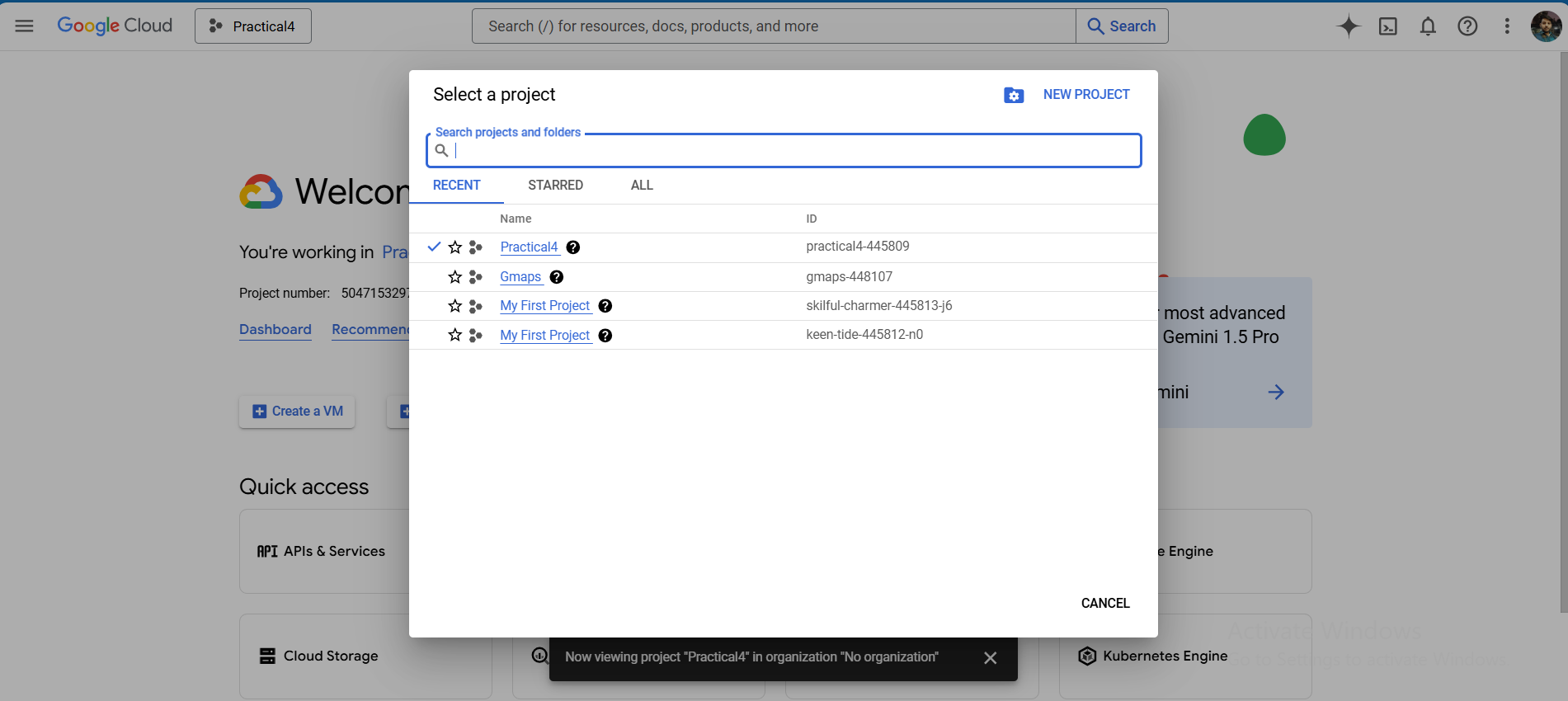


**Step-by-Step Guide to Implement Google Maps API with JSP**

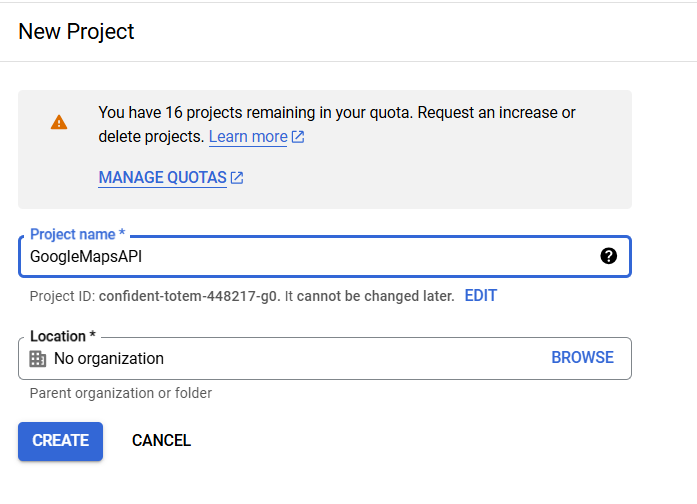
**1. Get Your Google Maps API Key**

* First, you need to obtain an API key from **Google Cloud Console** to use the **Google Maps JavaScript API**. Follow these steps:
  + Go to [Google Cloud Console](https://console.cloud.google.com/).
  + Create a new project (or select an existing one).
  + Enable the **Maps JavaScript API**. (Instead of Places API and Geocoding API)
  + Under **Credentials**, create an **API Key** and copy it.

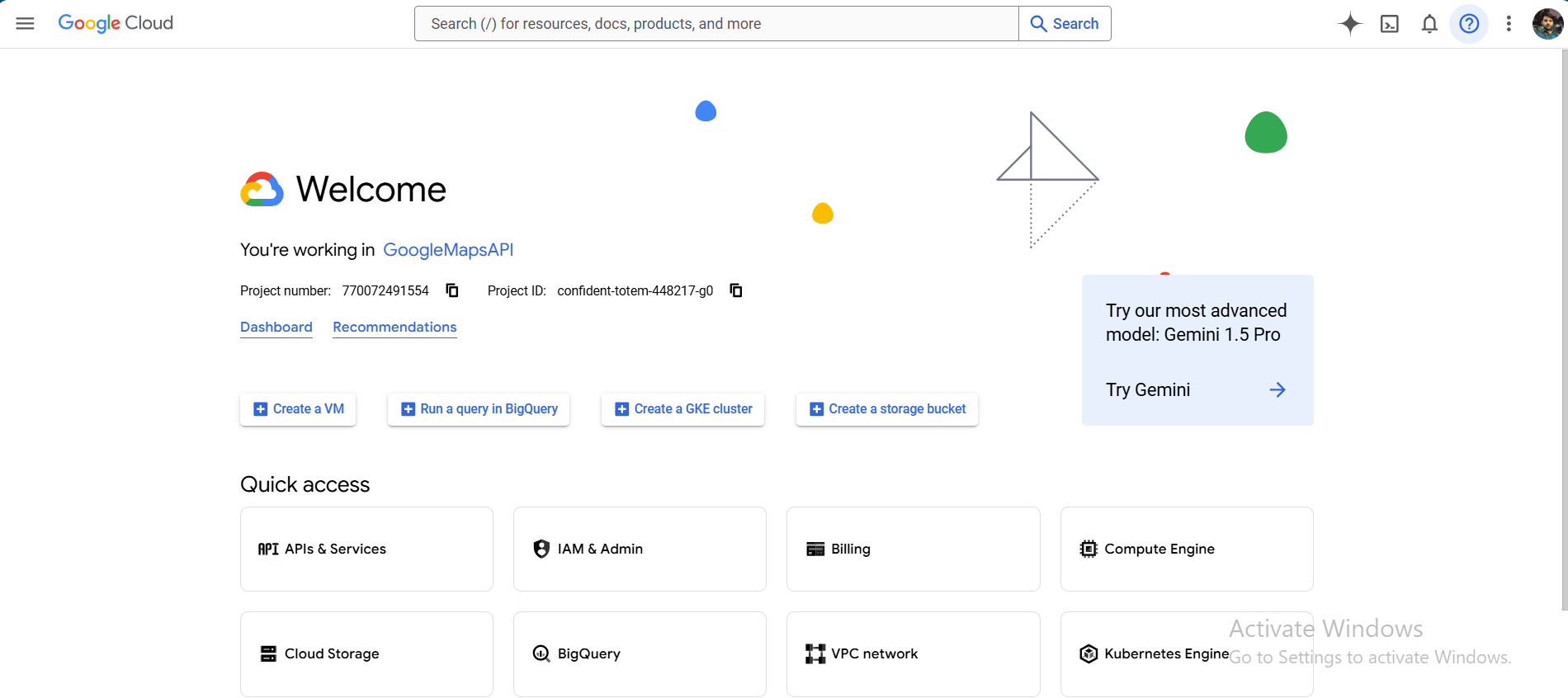
**Demonstration:** Here we have existing projects, now click on **NEW PROJECT**



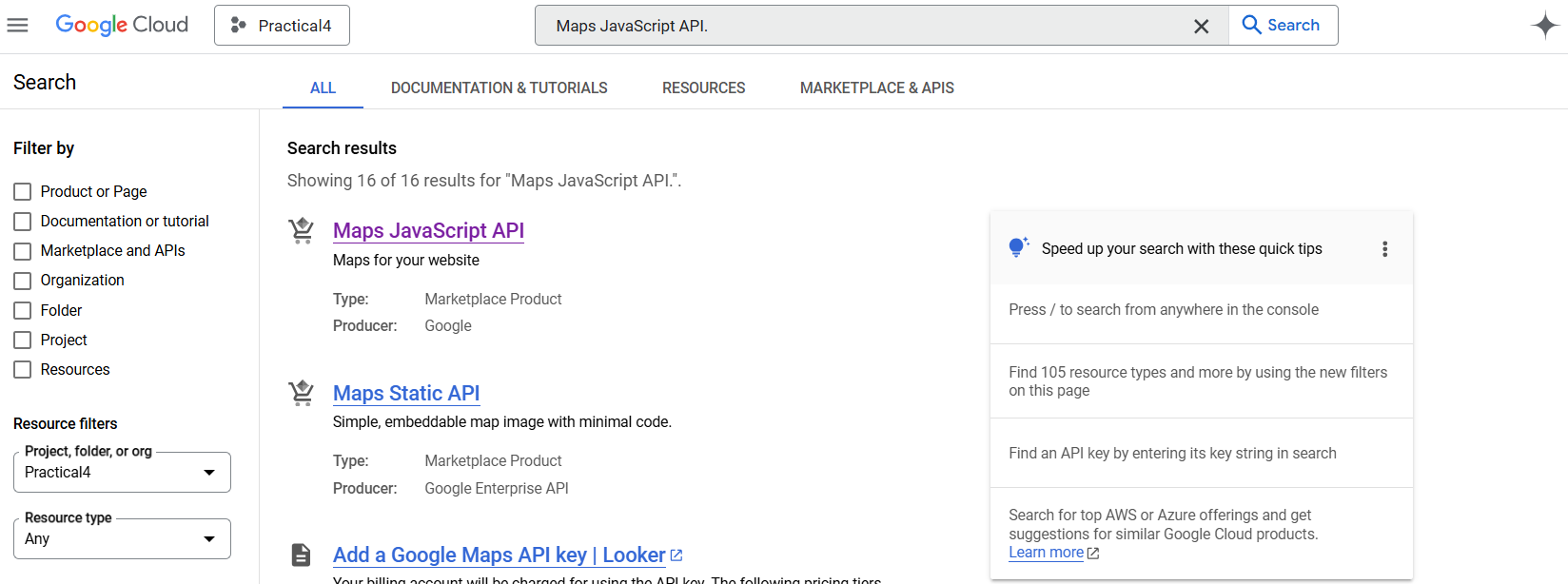
**Project Name:** GoogleMapsAPI and click CREATE

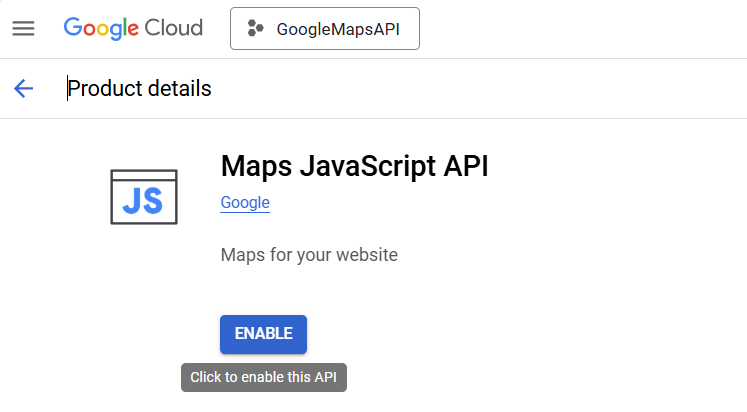


Now we can see our new project GoogleMapsAPI and Notifications of new project

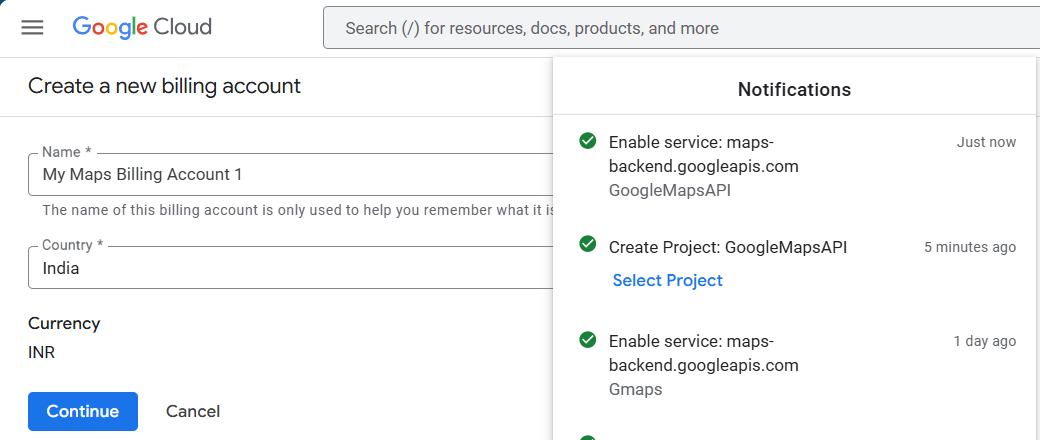


Search, Click and Enable the **Maps JavaScript API**.

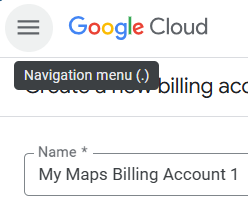




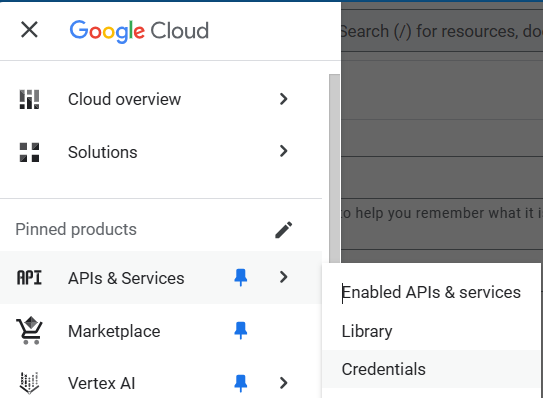
After Enabling Maps JavaScript API, it will show notifications and also redirect to billing method (Ignore billing method)



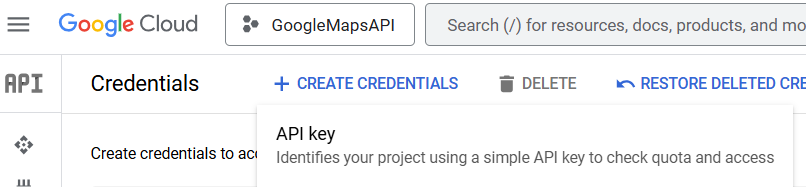
Now click on Navigation Menu

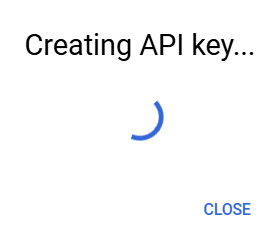


**APIs & Services > Credentials**

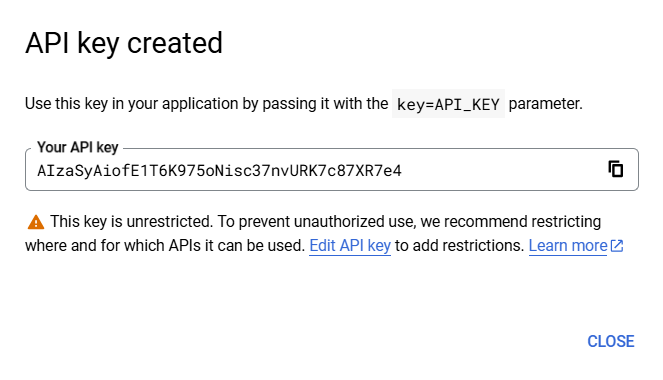


**Click + CREATE CREDENTIALS > API Key**





We got the API Key Created, Copy the API Key and paste it in **myindex.jsp** file.



Now, open **myjsp** folder and open with VS Code and paste the API\_KEY in **myindex.jsp** script tag:

**Shown below:**

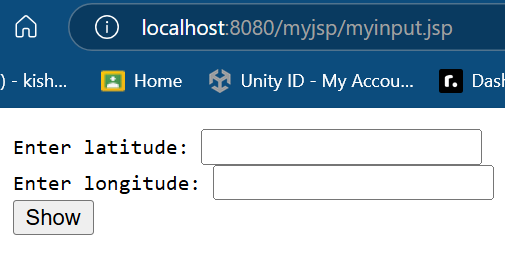
**Before:**<!-- Google Maps JavaScript API -->

    <script src="https://maps.googleapis.com/maps/api/js?key=**YOUR\_API\_KEY** &callback=initMap" async defer></script>

**After: :**<!-- Google Maps JavaScript API --><script src="https://maps.googleapis.com/maps/api/js?key=**AIzaSyAiofE1T6K975oNisc37nvURK7c87XR7e4** &callback=initMap" async defer></script>

**Step 5:** Now to check the output, open a web browser Google Chrome or Microsoft Edge Ctrl + Click to follow the Link, URL: <http://localhost:8080/myjsp/myinput.jsp>

**Demonstration:**



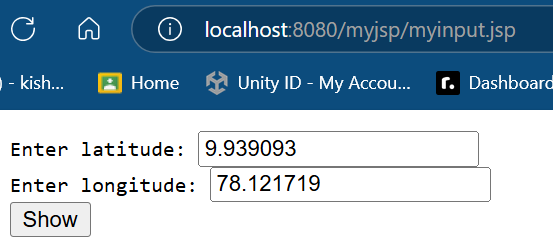
Enter Example latitude and longitude of any place

Here, latitude and longitude coordinates of city: Madurai

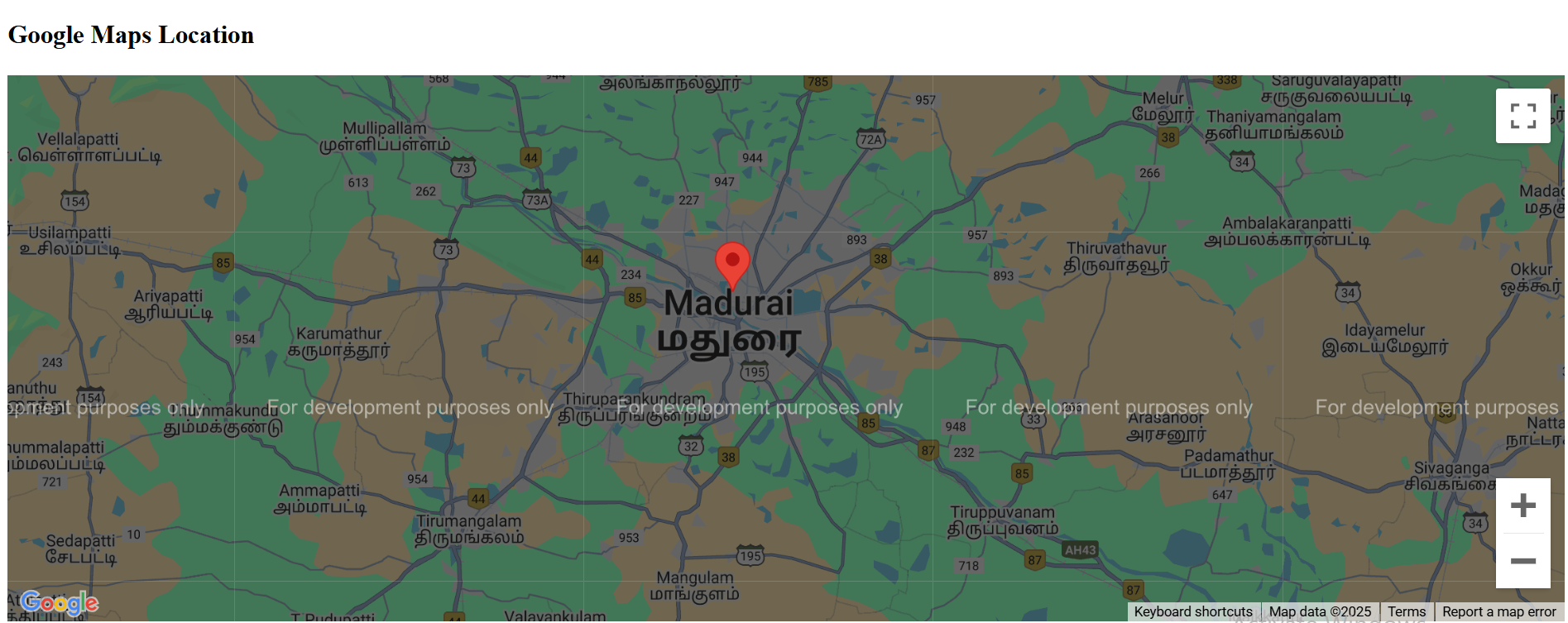
Latitude: 9.939093

Longitude:  78.121719

**Output: Enter co-ordinates and click on Show**



**It will redirect to the Google Maps: Exact Coordinates Location**



-----------------