FINAL PROJECT REPORT

Developing a Web Application named “Intensity Map Prediction” for generating the intensity map of registered Earthquake

Prepared for:

Mr. Ravikant Singh (Scientist E)

National Center of Seismology (NCS)

India Meteorological Department

Prepared by:

Saurav Prateek (University Roll No. 1549210096)

Ashish Saxena (University Roll No. 1549210031)

June 2018, Bachelor of Technology Computer Science (6th Semester) KCC ITM

DECLARATION

We, Mr. **Saurav Prateek** and Mr. **Ashish Saxena** hereby declare that this project report is the record of authentic work carried out by us during the period from **28th May 2018** to **09th July 2018** and has not been submitted to any other Company for the award of any certificate etc.

Signature:

Name of the student

Date: 9th July 2018

CERTIFICATE

I , **Saurav Prateek (1549210096)**, student of **B.Tech (CSE), 3rd** year of **KCC Institute of Technology and Management, Greater Noida** completed my summer training project from **INDIA METEOROLOGICAL DEPARTMENT, New Delhi** from **May 28, 2018 to July 09, 2018.**

During the mentioned period I developed a **Web Application** and completed my summer training project entitled "**Intensity Map Prediction**" under the guidance of **Mr. Ravikant Singh (Scientist E).**

Date: **Name:** Saurav Prateek

CERTIFICATE

I , **Ashish Saxena (1549210031)**, student of **B.Tech (CSE), 3rd** year of **KCC Institute of Technology and Management, Greater Noida** completed my summer training project from **INDIA METEOROLOGICAL DEPARTMENT, New Delhi** from **May 28, 2018 to July 09, 2018.**

During the mentioned period I developed a **Web Application** and completed my summer training project entitled "**Intensity Map Prediction**" under the guidance of **Mr. Ravikant Singh (Scientist E).**

Date: **Name:** Ashish Saxena

ACKNOWLEDGEMENT

There is always a sense of gratitude which one express towards others for their help and supervision in achieving the goals. This formal piece of acknowledgement is an attempt to express the feeling of gratitude towards people who help me in successfully completing of my training.

I would like to express my deep gratitude to **Mr. Ravikant Singh**, my training coordinator for their constant co-operation. He was always there with his competent guidance and valuable suggestion throughout the pursuance of this project.

I would also like to place of appreciation to all the respondents and group members whose responses and coordination were of utmost importance for the project.

Above all no words can express my feelings to my parents, friends all those persons who supported me during my project. I am also thankful to all the respondents whose cooperation and support has helped me a lot in collecting necessary information.

(SAURAV PRATEEK)

(ASHISH SAXENA)

Executive Summary

The report represents the task completed during summer internship at India Meteorological Department (IMD) which is given below:

Intensity Map Prediction (Web Application):

The web application was named as “Intensity Map Prediction” as the name suggests, it generates an Intensity Map of Earthquake registered at a specific location in India.

The Web Application implemented the following technologies:

- HTML
- CSS
- Java
- JSP (Java Server Pages)
- Google India Maps API
- JavaScript (JS)
- Servlets (A Java Class File)

The application also provided with the information about top cities affected by the earthquake along with the intensity experienced by the affected city and its epicenter distance.

Table of Contents

1.1) Introduction:

The Web Application takes data as an input in two formats:

- Read from User Input
- Read from text File

It takes the following data as an input:

- Latitude
- Longitude
- Magnitude
- Depth (km)
- Date
- Time (UTC)

The following input data is further processed and these outputs are generated:

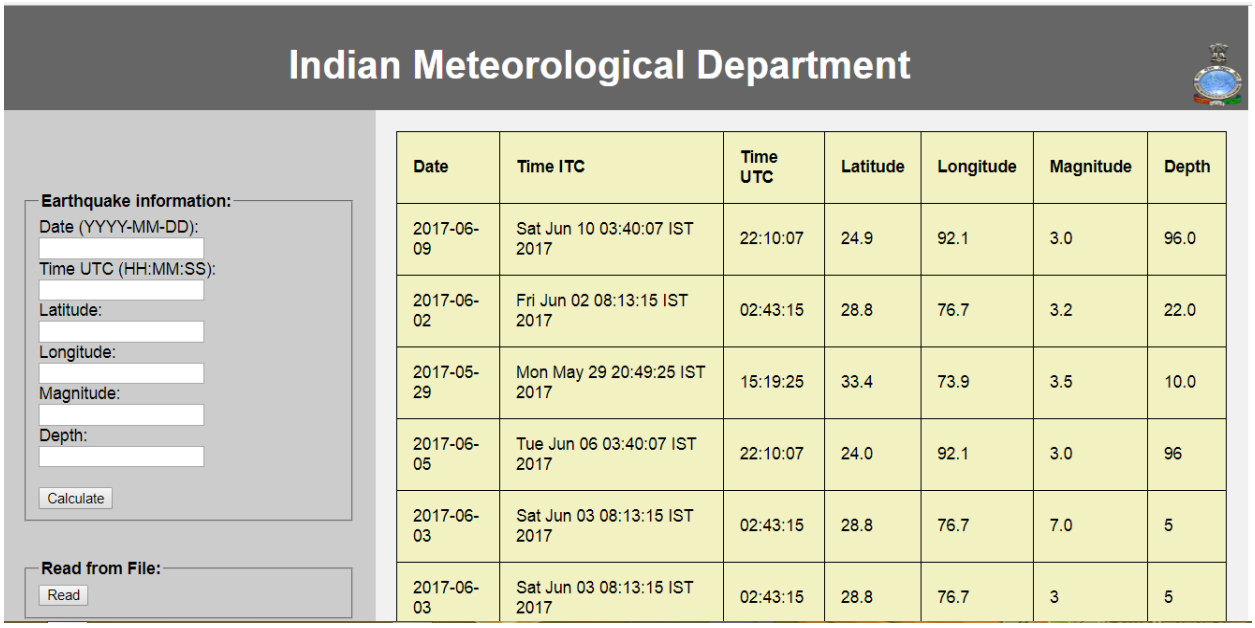
- An Intensity Map which shows the intensity of the earthquake over map of India implemented through “Google India Maps API”.
- A list of top cities with the range of Intensity and epicenter distance which were affected by the registered earthquake.

The Web Application can predict the intensity of the earthquake at various locations and provide user with a graphical or pictorial view of the severity of the Earthquake.

Moreover it can also list the cities which were most severely affected due to the registered earthquake along with the Intensity Range at that location and their epicenter distance.

1.2) Index Page: Intensity Predict

The name of the Index Page/ Welcome Page of Web Application is “Intensity Predict”. The front end design of the page looks as given below.



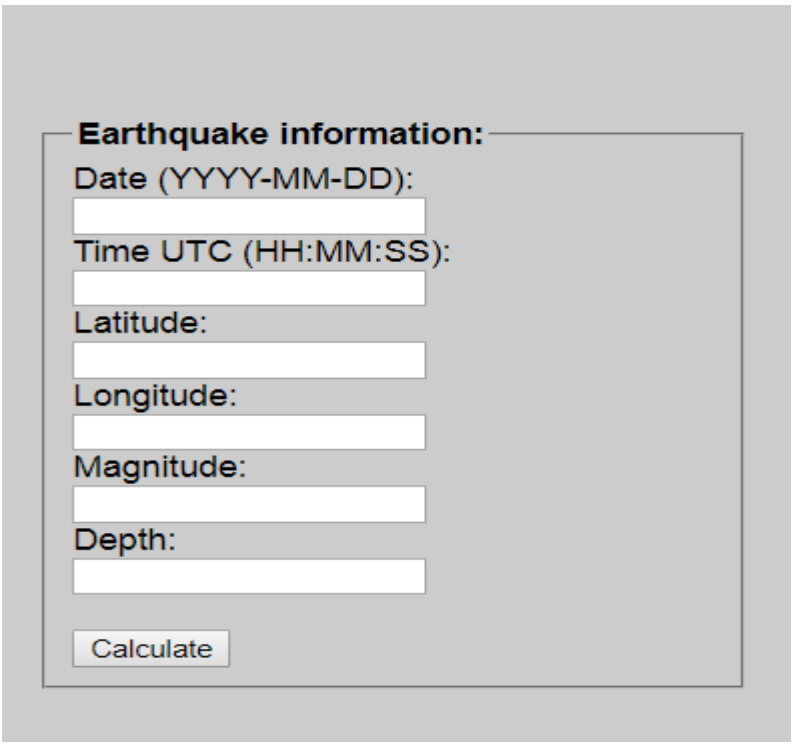
Date	Time ITC	Time UTC	Latitude	Longitude	Magnitude	Depth
2017-06-09	Sat Jun 10 03:40:07 IST 2017	22:10:07	24.9	92.1	3.0	96.0
2017-06-02	Fri Jun 02 08:13:15 IST 2017	02:43:15	28.8	76.7	3.2	22.0
2017-05-29	Mon May 29 20:49:25 IST 2017	15:19:25	33.4	73.9	3.5	10.0
2017-06-05	Tue Jun 06 03:40:07 IST 2017	22:10:07	24.0	92.1	3.0	96
2017-06-03	Sat Jun 03 08:13:15 IST 2017	02:43:15	28.8	76.7	7.0	5
2017-06-03	Sat Jun 03 08:13:15 IST 2017	02:43:15	28.8	76.7	3	5

The “Intensity Predict” is a JSP Page which was developed so that user can input their data or read from “SMS_L2.txt” file.

It supports input in two formats:

1.2.1) Read from User Input:

In this part the Earthquake data is recorded by the user manually. The manual input Form looks like:



The code for the above form looks like:

```
 <form action="Calculate">
<fieldset>
  <legend><b>Earthquake information:</b></legend>
  Date (YYYY-MM-DD):<br>
  <input type="text" name="date" ><br>
  Time UTC (HH:MM:SS):<br>
  <input type="text" name="utc" ><br>
  Latitude:<br>
  <input type="text" name="lat" ><br>
  Longitude:<br>
  <input type="text" name="long" ><br>
  Magnitude:<br>
  <input type="text" name="mag" ><br>
  Depth:<br>
  <input type="text" name="heig" ><br><br>
  <input type="submit" value="Calculate">
</fieldset>
<br>
<p>
</form>
```

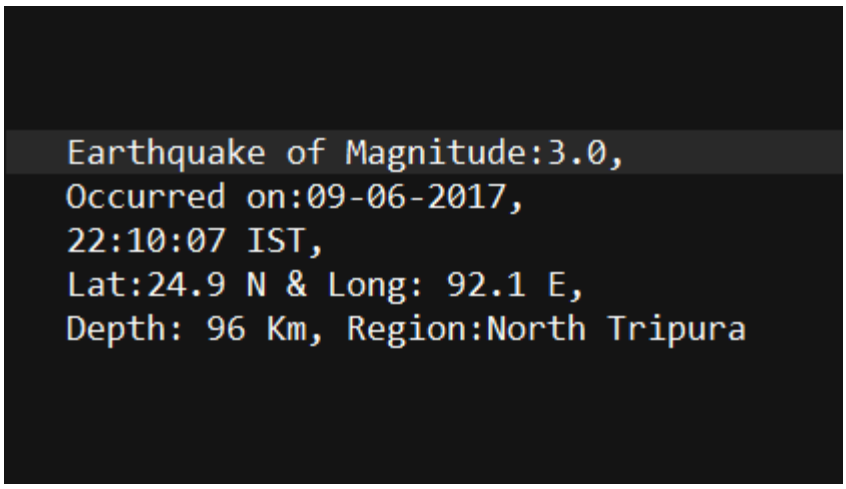
When the “Calculate” button is clicked then the data filled in the fields are directed to a servlet named: “Calculate”.

Note: The functionality of “Calculate” servlet will be discussed further.

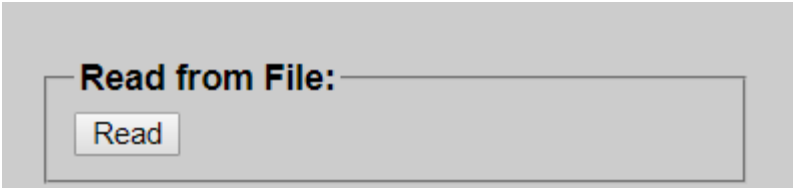
1.2.2) Read from “SMS_L2.txt” File:

In this part the values required as input is read from “SMS_L2.txt” file. It is a text file which stores the record of an earthquake data.

The structure of “SMS_L2.txt” file is:



The Read from File form looks like:



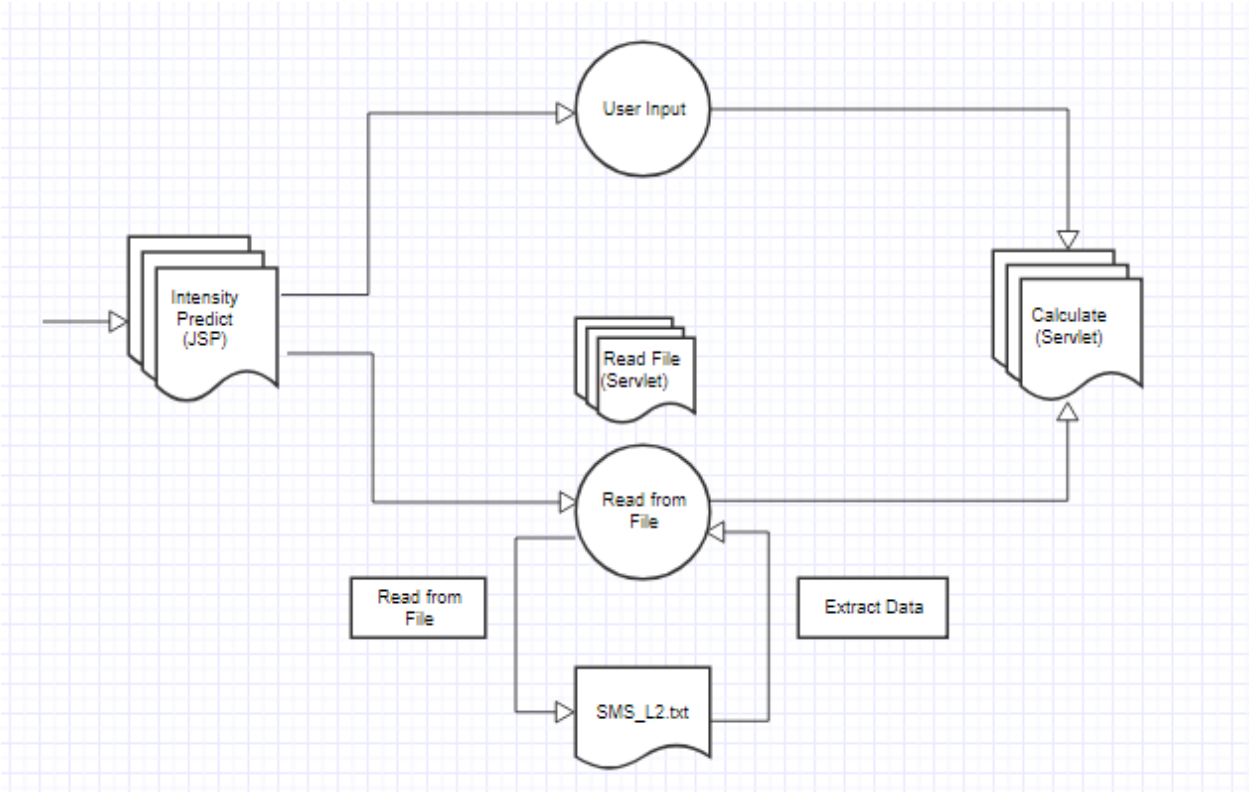
When ‘Read’ button is clicked then the control flow is directed to a Servlet named ‘ReadFile’.

The function of ‘ReadFile’ Servlet is to Read Data from ‘SMS_L2.txt’ File, extracts the required data and directs them to the ‘Calculate’ servlet.

Code for reading file “SMS_L2.txt” file:

```
FileRead obj = new FileRead();
String res = obj.getData(); //fetching data from SMS_L2.txt file
String[] results = res.split("#");

double longi = Double.parseDouble(results[2]);
double lati = Double.parseDouble(results[1]);
double magni = Double.parseDouble(results[0]);
double heigh = Double.parseDouble(results[3]);

String date = results[4];
String[] dates = date.split("-");
date = dates[2]+"-"+dates[1]+"-"+dates[0];
String timeutc = results[5];
String timeitc = "invalid UTC format";


UTCtoIST istobj = new UTCtoIST();
timeitc = istobj.convertUTC(date+" "+timeutc);

String data = date+"#"+timeitc+"#"+timeutc+"#"+lati+"#"+longi+"#"+magni+"#"+heigh;
SaveData sd1 = new SaveData();
sd1.saveToFile(data);

TopCities toc1 = new TopCities(lati,longi,magni,heigh);
String[] det = toc1.getDetails();

request.setAttribute("Longitude",longi);
request.setAttribute("Latitude",lati);
request.setAttribute("Magnitude",magni);
request.setAttribute("Height",heigh);
```

The control flow of the Input process can be shown as:



1.3) Storing Input Data:

The recorded earthquake data is submitted to a text file named "EQData.txt".

The format of "EQData.txt" file is:

Date#ITCTime#UTCTime#Latitude#Longitude#Magnitude#Depth

When the extracted or recorded input data is sent to "Calculate" servlet, it directs the input data to a servlet named "SaveData".

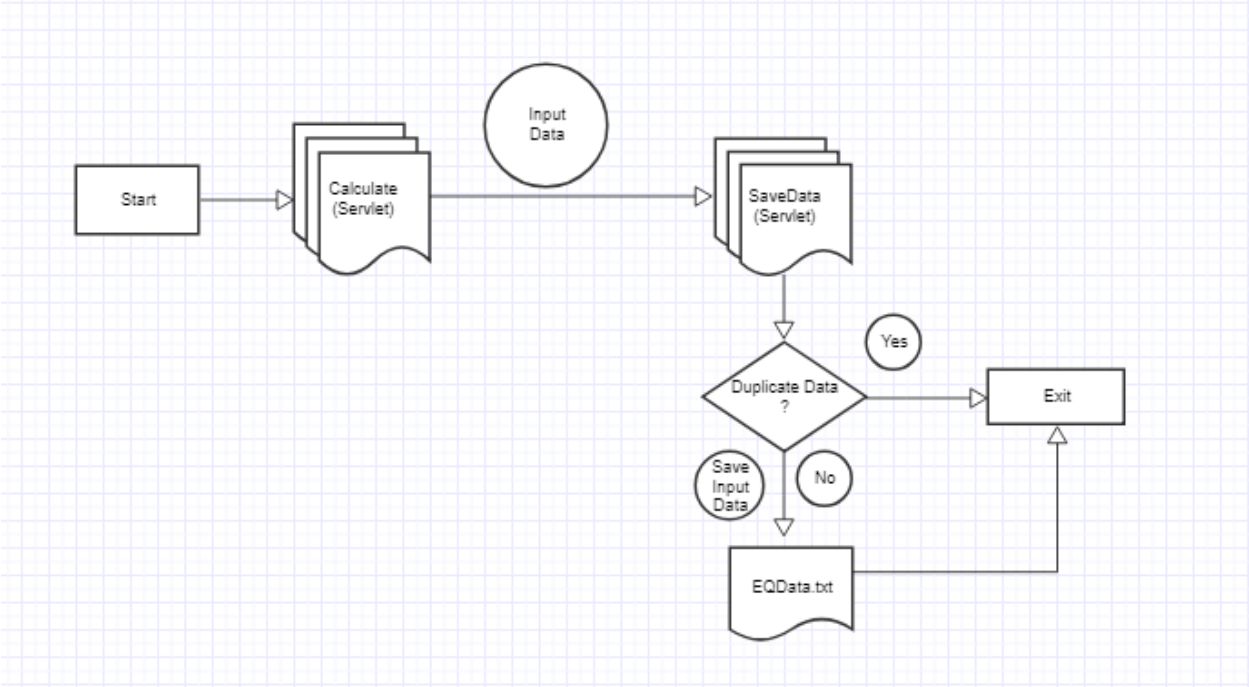
The “SaveData” servlet converts the received data in the format supported by “EQData.txt” file and then stores it into that text file.

It also checks for the presence of duplicate entry and in that case the saving of input data is restricted.

Code for Saving Data is:

```
import java.io.*;
public class SaveData {

    public void saveToFile(String data){
        try{
        File file = new File("C:\\\\Users\\\\hp\\\\Documents\\\\Web Applications\\\\IntensityCalculate\\\\src\\\\java\\\\EQData.txt");
        BufferedReader br = new BufferedReader(new FileReader(file));
        String st;
        int found=0;
        while((st = br.readLine()) != null){
            if(st.equals(data)){
                found=1;
                break;
            }
        }
        if(found==0){

          BufferedWriter out = new BufferedWriter(
                  new FileWriter(file, true));
          out.write(data+"\n");
          out.close();
        }
      }
      catch(Exception e){
          System.out.println(e);
      }
    }
}
```

The control flow for Saving Data can be shown as:



The data stored in the "EQData.txt" file is represented at the Index File i.e. "Intensity Predict" in a tabular format shown as:

Date	Time ITC	Time UTC	Latitude	Longitude	Magnitude	Depth
2017-06-09	Sat Jun 10 03:40:07 IST 2017	22:10:07	24.9	92.1	3.0	96.0
2017-06-02	Fri Jun 02 08:13:15 IST 2017	02:43:15	28.8	76.7	3.2	22.0
2017-05-29	Mon May 29 20:49:25 IST 2017	15:19:25	33.4	73.9	3.5	10.0
2017-06-05	Tue Jun 06 03:40:07 IST 2017	22:10:07	24.0	92.1	3.0	96
2017-06-03	Sat Jun 03 08:13:15 IST 2017	02:43:15	28.8	76.7	7.0	5

The table shows the top 9 data entries, if more than 9 entries are present in the “EQData.txt” file.

The backend code for reading the top 9 entries from “EQData.txt” file is given as:

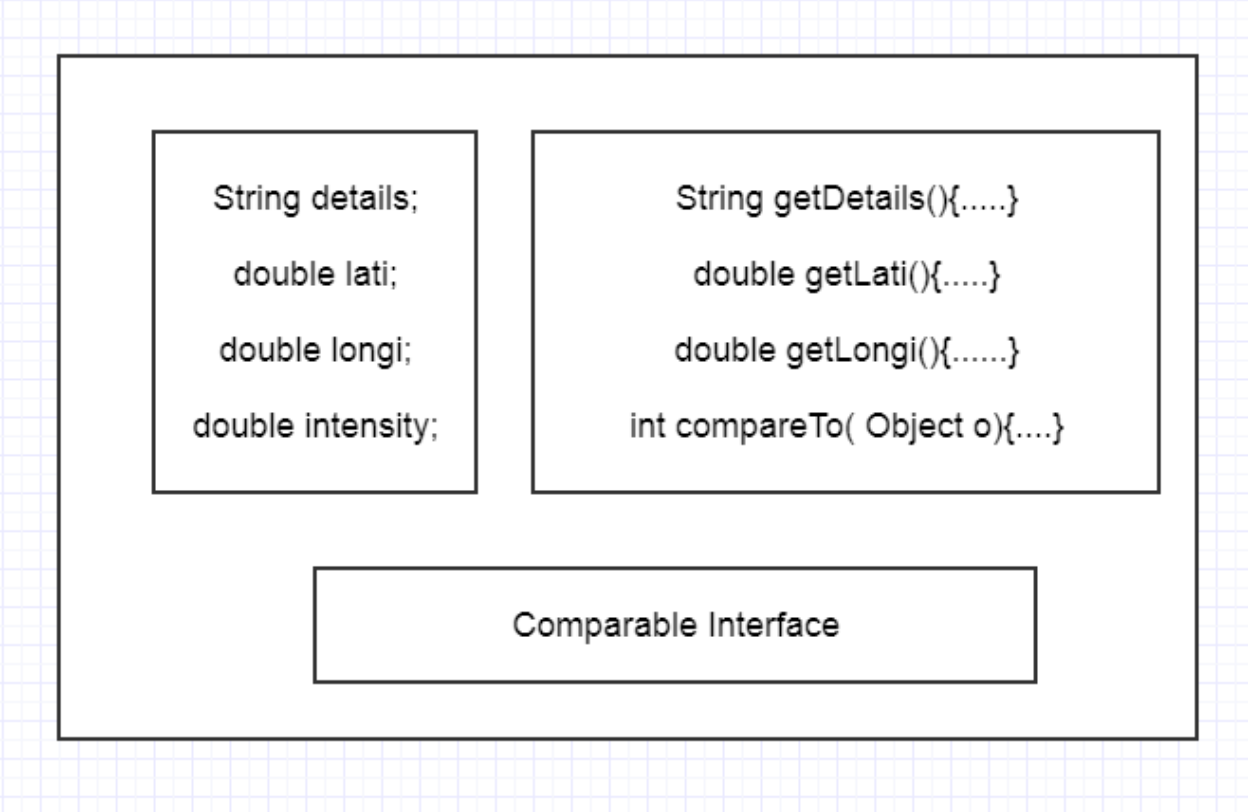
```
File file = new File("C:\\\\Users\\\\hp\\\\Documents\\\\Web Applications\\\\IntensityCalculate\\\\src\\\\java\\\\EQData.txt");
BufferedReader br = new BufferedReader(new FileReader(file));
String st;
String[][] str = new String[10][7];
int ch=0;
while((st = br.readLine()) != null){
    String[] lin = st.split("#");
    str[ch][0] = lin[0];
    str[ch][1] = lin[1];
    str[ch][2] = lin[2];
    str[ch][3] = lin[3];
    str[ch][4] = lin[4];
    str[ch][5] = lin[5];
    str[ch][6] = lin[6];
    ch++;
    if(ch==10){
      break;
    }
}
```

The top 9 entries from “EQData.txt” File is stored in a two-dimensional Array of Strings named as “str[][]”.

1.4) Backend Structure:

1.4.1) Generating List of Top Cities:

When the Input data is directed to “Calculate” servlet then the data is sent to a Java Class File named “TopCities.java”.
The “TopCities” class file takes the input data and returns the set of top cities affected by the Earthquake.
The list of various cities along with its location was stored in a “CityModified.txt” text file. The details were taken from this file and an object of City was created. The structure of the object is:



The list of top cities is created through a Greedy Algorithm in which the distance of all the cities from the epicenter is calculated and then the algorithm greedily selects the top cities having the

minimum calculated distance. This list is further returned. The code for selecting the top cities is given as:

```
ArrayList<City> arr = new ArrayList<>();
    String line = null;

try{
    FileReader fileReader = new FileReader("C:\\\\Users\\\\hp\\\\Documents\\\\Web Applications\\\\IntensityCalculate\\\\src\\\\java\\\\CityModified.txt");
       BufferedReader bufferedReader = new BufferedReader(fileReader);
           while((line = bufferedReader.readLine()) != null) {
              String[] str = line.split("#");
              String loc = str[0];
              double lati = Double.parseDouble(str[1].substring(0,str[1].length()-2));
              double longi = Double.parseDouble(str[2].substring(0,str[2].length()-2));
              double intensity = calci(lati,longi);
              double distance = getDistanceFromLatLonInKm(plati,plongi,lati,longi);
              City obj = new City(loc,lati,longi,intensity,distance);
              arr.add(obj);
              //System.out.println(loc+"#"+lati+"#"+longi);
           }
}
catch(Exception e){
       System.out.println(e);
}
    Collections.sort(arr);
    String[] result = new String[5];

    for(int i=0;i<5;i++){
       City c1 = arr.get(i);
           String det = c1.getDetails()+"   "+(getRoman(c1.getIntensity()))+" || ("+Double.toString(c1.getDistance()).substring(0,7)+" km)";
           result[i]=det;
    }
    return result;
```

The intensity at a particular point (Latitude, Longitude) is calculated through a function “calci” which is given as:

```
public double calci(double lati,double longi){
        double r = 6371;
        double x = plongi;
        double y = plati;
        double a = Math.sin(toRadians(90-y))*Math.sin(toRadians(90-lati));
        double p = Math.cos(toRadians(x-longi));
        double xi = Math.cos(toRadians(90-y))*Math.cos(toRadians(90-lati));
        double di = Math.acos(xi+(a*p))*r;
        double d = Math.sqrt(di*di + hei*hei);
        double intensity = 6.15+1.13*magni-0.0006*d-3.12*Math.log10(d);

        return intensity;
  }
```

The “TopCities” class file returns an array of string name “result” which holds the name of top cities affected by the Earthquake along with their intensity range.

1.4.2) Generating Intensity Map:

The Input data along with the list of top cities are directed to a JSP File named as “map1.jsp”.

The Input data is used to calculate the Intensity of Earthquake at different latitudes and longitudes over the surface. These value ranges from:

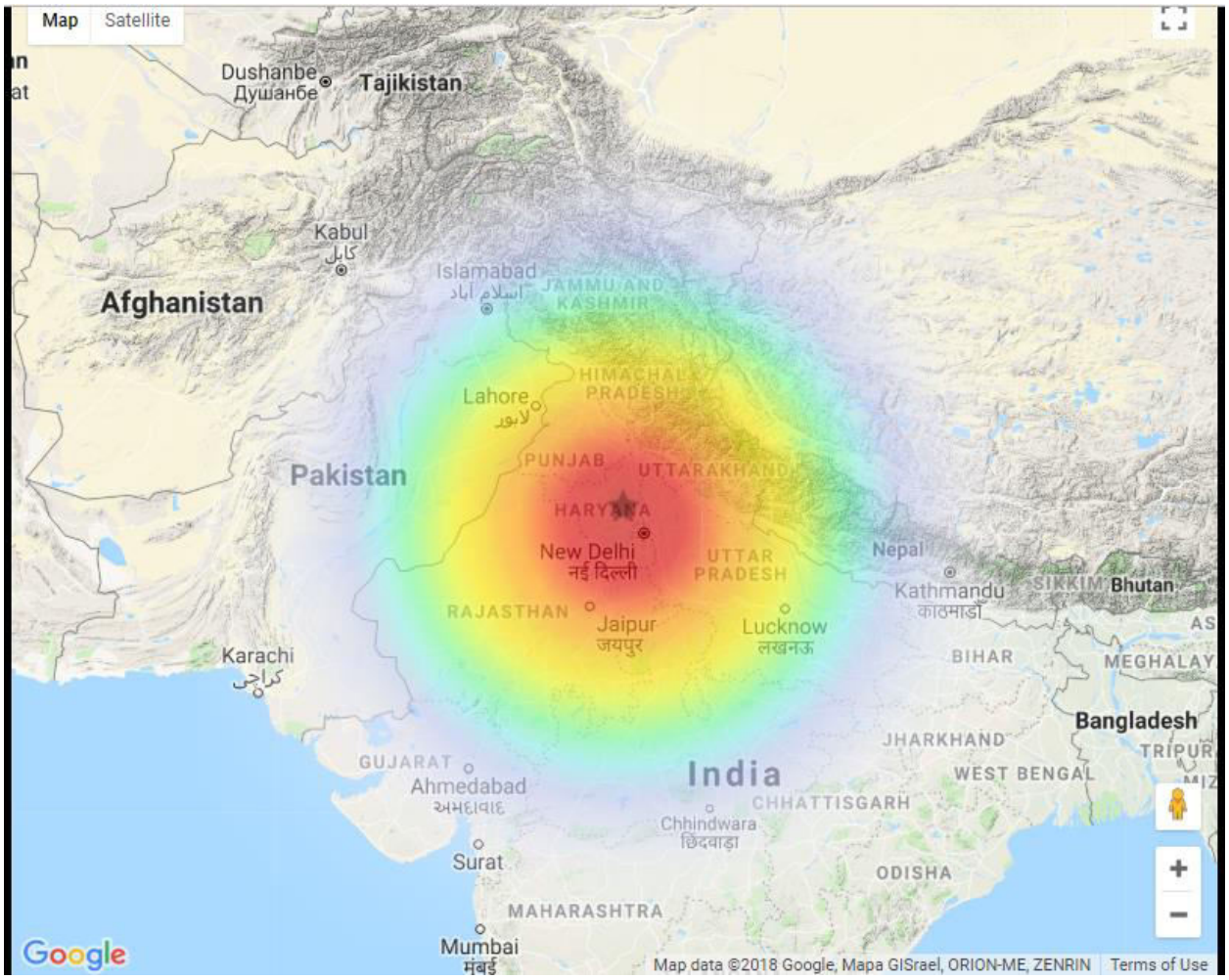
- Latitude (0 to 40)
- Longitude (60 to 100)

These list of points along with intensity is stored in a “heatMapData” array and passed to the “initMap()” function which generates a Heat Map of the input data over Google India Maps API.

The code for initializing the map and generating a Heat Map is given as:

```
function initMap() {

    map = new google.maps.Map(document.getElementById('map'), {
    zoom: 5,
    center: {lat:${Latitude}, lng:${Longitude}},
    mapTypeId: 'terrain'
    });
    var eqdata = CalculateIntensity(${Longitude},${Latitude},${Magnitude},${Height});
    var heatMapData = [{location: new google.maps.LatLng(${Latitude},${Longitude}), weight:eqdata}];
    var heatmap = new google.maps.visualization.HeatmapLayer({
     data: heatMapData,
     dissipating : false,
     gradient : [
       'rgba(255,255,255,0)',
       'rgb(226,230,249)',
       'rgb(193,207,253)',
       'rgb(143,242,249)',
       'rgb(118,253,170)',
       'rgb(172,254,95)',
       'rgb(231,252,40)',
       'rgb(253,232,5)',
       'rgb(248,213,0)',
       'rgb(255,172,9)',
       'rgb(255,150,4)',
       'rgb(252,78,9)',
       'rgb(229,2,4)',
       'rgb(212,0,0)',
       'rgb(201,1,8)',],
        radius : 300,
        maxIntensity : 11.5
    });
    heatmap.setMap(map);
```

The map of predicted intensity is given as:



Moreover the list of Top Cities received by the JSP file is also represented on the page in the format given as:

Predicted Intensity Map:

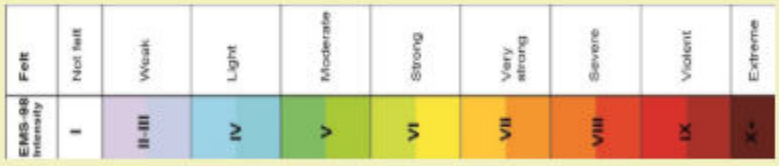
Longitude : **76.7 N**
Latitude : **28.9 E**
Magnitude : **6.0**
Depth (km) : **5.0**

Intensities at Nearby Cities:
1) Rohtak (Haryana) IX || (11.6816 km)
2) Gohana (Haryana) VIII || (25.5748 km)
3) Bahadurgarh (Haryana) VIII || (31.6994 km)
4) Sonipat (Haryana) VIII || (31.7812 km)
5) Sultanpur Majra (Delhi) VIII || (38.3685 km)

A legend is also placed in order to determine the strength of the Earthquake registered and plotted over the Intensity Map.

The Legend is shown as:

Legend:



1.5) Output:

The output generated by the Web Application after processing the input of Earthquake data is shown as: