

CS6308 JAVA PROGRAMMING

PROJECT DOCUMENTATION

Title: Smart Weather GUI Application with Travel Guidance.

By,

G VIGNESH

2018103078

3rd Year DCSE

Anna University.

**Abstract:**

Weather is the state of the atmosphere at a given place and time in regards to heat, cloudiness, dryness, sunshine, wind, and rain. Of all the geophysical phenomena weather is the most significant one that influences us. Weather can vary greatly and largely depends on climate, seasons and various other factors. The chief goal of this work is to get the weather forecast of any city throughout the world through an application and provide Global warming data such as ultra violet radiation index, visibility, humidity etc. Alongside providing customized travel suggestions to tourists and travellers based on the temperature,in that location for the upcoming week and representing in a line chart for easy visualization of the users. If the customer is a registered userthen a report is generated in file and sent the registered e-mail address using smtp protocol if the customer wishes.

This project aims at creating a GUI Based webapplication using Java which asks the user to register and login to their account renders weather reports from internet using OPENWEATHERMAP API at run time by placing a HTTP request and response.The customer gets the report in a visualized chart for better understanding and a report is sent to their registered mail for further reference.

**Introduction:**

In Day-to-Day life we use may have used weather forecast applications which just shows us temperature maximum, minimum, chances of rain etc. which helps us updating ourself for our daily activities and planning our work according to weather. But your Smart weather GUI asks user to login using the credentials if not registered allows us to register. Then when user searches for a location shows the data such as dew-point, cloud density, percentage of chances of rain, sunrise, sunset time, humidity, wind speed, pressure, dew point etc. Our application provides global warming data such as UV radiation index, visibility etc. It also provides Travel Guidance by collecting the temperature details of day and night for 1 week and represent them in Line chart for visually understanding the weather. If the registered user wants the weather report is copied into a file and sent to the registered mail address for him to download and use it where ever he goes.

Problem Statement:

Weather applications present today provides only a raw unprocessed data such as maximum, minimum temperature

Which does not help us to plan our travel activities. Alsoweather report Data is available where internet facility is available and when the user is at out of reach of signal during his travelling such as hill stations or forests he/she does not get access to the weather report and he cannot schedule his/her activities accordingly it.

Objective:

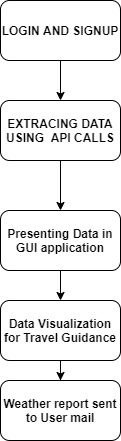
But if the raw data is processed by the application and provided in a visualized graph/ chart manner it helps users to easily understand the weather situationin a particular location and make his/her travel planning accordingly.

If the weather report is sent to our mail address and if we have downloaded once we can access it anywhere without the help of the internet makes it easy for the users to get in touch with the weather report in no network areas such as hill stations etc.

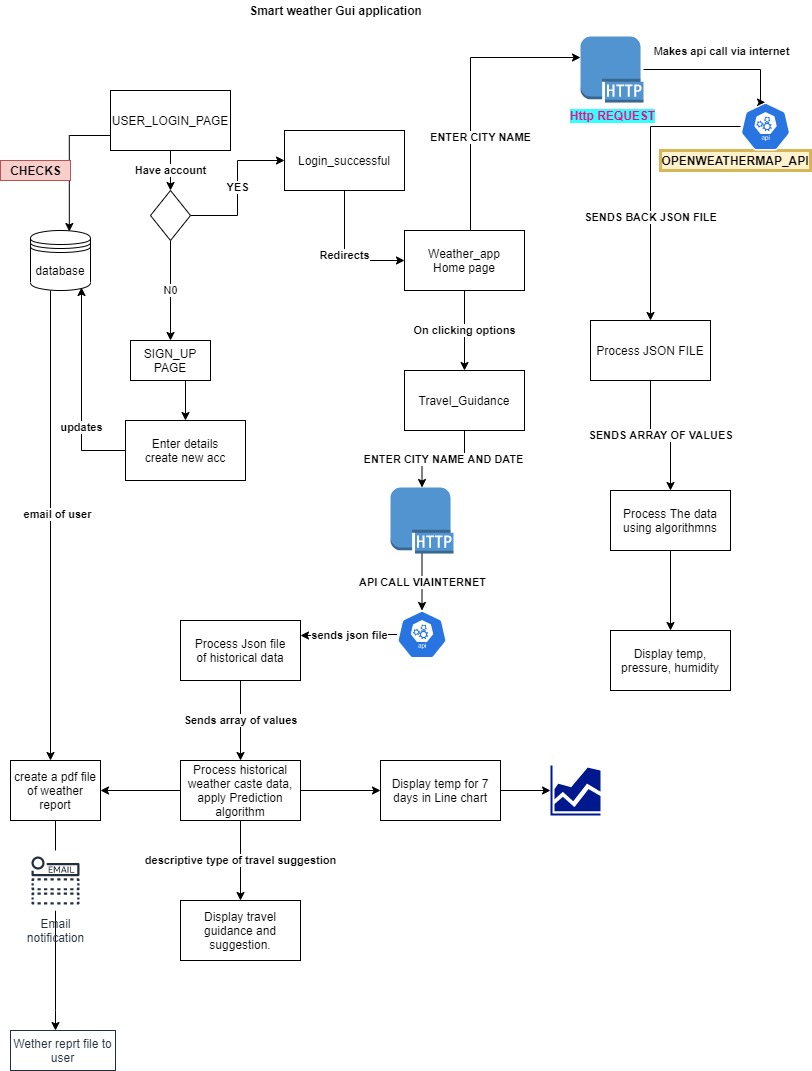
**List of Modules:**

* **Login and signup**
* **Extracting data using API calls**
* **Presenting data in GUI application.**
* **Data visualization for Travel Guidance**
* **Weather report sent to User mail.**

**Block Diagram:**

****

**Flow chart:**

****

**Module-1: Login and Signup**

For keep track of customer details, Users are requested to signup in our application and then login to the system suing their credentials. For this MYSQL database is used and a table is created and connected to our project via JDBC. During Registration user details are inserted into the table Using SQL commands via JDBC drivers. The details included in registration page are first-name, last-name, username, password, email address and phone number. When the user enters his login credentials his/her login in successful.

Pseudo-code:

Registration page:

Connectionconnection = DriverManager.getConnection("jdbc:mysql://localhost:3307/project","root","root");

Stringquery = "INSERT INTO account values('" + firstName + "','" + lastName + "','" + userName + "','" + password + "','" + emailId + "','" + mobileNumber + "')";

Statementsta = connection.createStatement();

Login page:

Connectionconnection = (Connection) DriverManager

      .getConnection("jdbc:mysql://localhost:3307/project","root","root");

PreparedStatementst = (PreparedStatement) connection.prepareStatement(

       "Select user\_name, password from account where user\_name=? and password=?");

 st.setString(1,user\_name);

 st.setString(2,password);

 ResultSetrs = st.executeQuery();

Module-2:Extracting data using API calls

Once the user is successfully logged in. He/She search for a location name say Coimbatore. Then an API call is made to OPENWEATHERMAP API. Which gives us raw data output in the form of JSON file containing weather report details needed for our project for that particular location. This API call is made using HTTP request and JSON file is returned as HTTP response. OPENWEATHERMAP API provides data to users for educational purpose for limited locations and not for very rural locations. First we are requested to register and unique key is provided which is used every time by us when making api request.

**PSEUDO CODE:**

 Stringkey = "dcec2f2ed1135ea7fb32573bc472c23e";

StringcreateUrl = "https://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid=" + key+ "&units=metric";

try {

            xurl = newURL(createUrl);

} catch (MalformedURLExceptione) {

            Stringerror = "Invalid URL";

            returnerror;

}

InputStreamis = (InputStream) xurl.getContent();

BufferedReaderbr = newBufferedReader(newInputStreamReader(is));

Stringline = null;

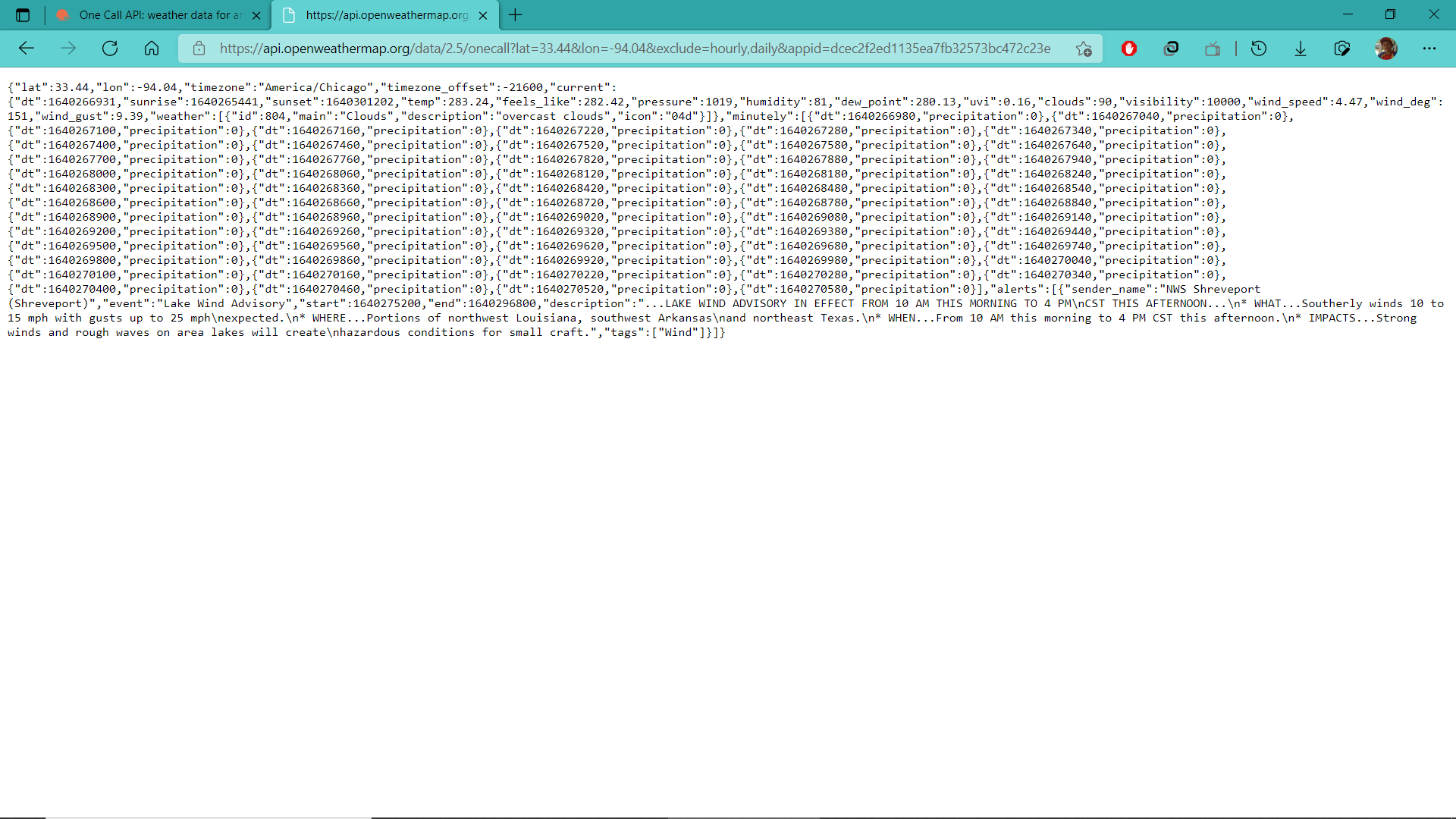
StringBuffersb = newStringBuffer();

while ((line = br.readLine()) != null) {

            sb.append(line);

}

**Output JSON file:**

****

**Module -3 Presenting data in GUI application.**

Output from module-2 is a JSON file has to processed and stored into appropriate variables to make it able to display in our GUI based application. Some data are provided in other Standard format. For example sunrise and sunset time is provided in Unix UTC time (i.e in terms of secs from from start of JAN 1,1970).

**JSON file processing:**

Raw json file containing of JSON arrays and JSON objects is processed and stored in temporary variables.

**Pseudo code:**

 Name[6] = Integer.toString(vi.getJSONObject("main").getInt("pressure"));

            Name[7] = Integer.toString(vi.getJSONObject("main").getInt("humidity"));

            longsunr\_unix = vi.getJSONObject("sys").getLong("sunrise");

            Datesunr = newDate(sunr\_unix \* 1000L);

            SimpleDateFormatsdf = newSimpleDateFormat("HH:mm:ss");

            Name[8] = sdf.format(sunr);

The values obtained are stored in variables are passed as an argument to event driven programming, where Button press event on triggering updates data in the text-labels of our GUI application.

Pseudo Code:

 button.addActionListener(newActionListener() {

  publicvoidactionPerformed(ActionEvente) {

                cityn = t1.getText();

                try {

                    extrae1 = newextra();

                    Strings1[] = e1.exec(cityn);

                    ta1.setText("Maximum temp" + s1[4] + "C");

                    ta2.setText("Minimum temp" + s1[3] + "C");

                    ta3.setText("Feels-Like" + s1[2] + "C");

}

Module-4: Data visualization for travel Guidance

To provide users travel guidance we are providing the temperature data during day and night time for the next 7 days. our program produces a line chart to visually represent the data to make it easy for the user to make their travel plans accordingly. To create Line chart JFREE chart jar is used which provides necessary functions for us to make chart and convert it into jpeg making it easy display in our GUI application.

Pseudo Code:

line\_chart\_dataset.addValue(t[12],"Night","nextday");

line\_chart\_dataset.addValue(t[13],"Night","nextday");

JFreeChartlineChartObject = ChartFactory.createLineChart(

            "Temp\*c Vs days","week-days",

            "Temperature \*c",

            line\_chart\_dataset,PlotOrientation.VERTICAL,

            true,true,false);

intwidth = 640;/\* Width of the image \*/

intheight = 400;/\* Height of the image \*/

Strings = c + ".jpeg";

FilelineChart = newFile(s);

ChartUtilities.saveChartAsJPEG(lineChart,lineChartObject,width,height);

Module-5: Weather report sent to user mail

When the user wishes to receive a copy of the weather report to his email address. Our application makes a connection to fetch the registered email address of the user and creates a text file and copies the weather report into it. Using email jar files an smtp connection is made and the created text report is sent to the user using our email address. This is helpful for the user to keep a downloaded copy of the weather report and use it where every he/she goes especially in no network areas.

Pseudo Code:

PrintWriterout = newPrintWriter("C:/Users/vigne/Desktop/WeatherGui/mailfile/Weatherreport.txt");

out.println("Weather Report at " + cityn);

 out.println("Temperature is:=" + s1[5] + "\*c");

 out.println("Feels\_like:=" + s1[6] + "\*c");

 out.println("Humidity:=" + s1[7] + "\*c");

 out.println("Dew-point:=" + s1[2]);

 Propertiesproperties = newProperties();

 properties.put("mail.smtp.auth","true");

properties.put("mail.smtp.starttls.enable","true");

 properties.put("mail.smtp.host","smtp.gmail.com");

 properties.put("mail.smtp.port","587");

textBodyPart.setText("Weather report at " + t1.getText());

MimeBodyPartpdfAttachment = newMimeBodyPart();

pdfAttachment.attachFile("C:/Users/vigne/Desktop/WeatherGui/mailfile/Weatherreport.txt");

                            // Attach body parts

 emailContent.addBodyPart(textBodyPart);

emailContent.addBodyPart(pdfAttachment);

                            // Attach multipart to message

 msg.setContent(emailContent);

 Transport.send(msg);

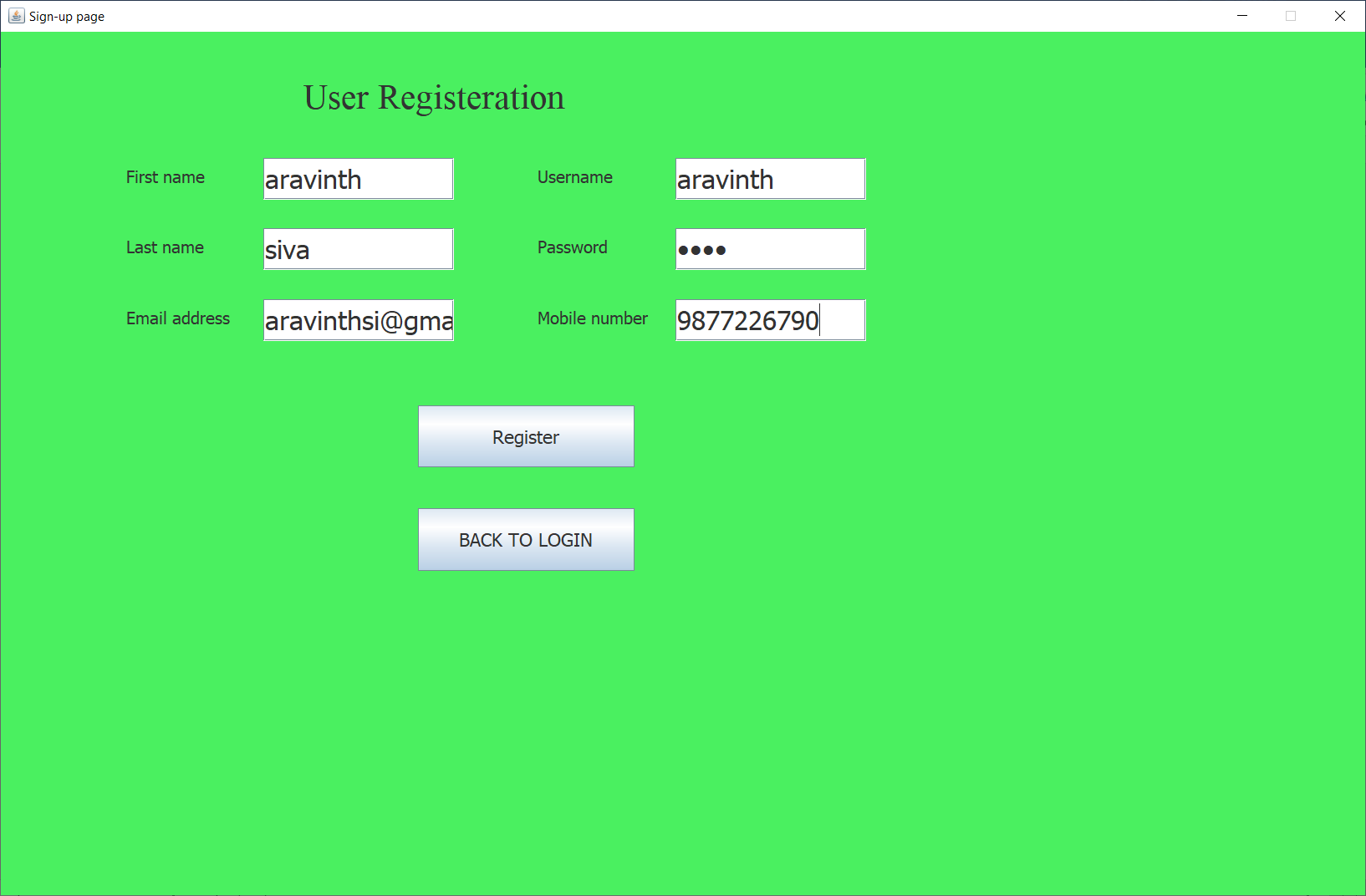
 System.out.println("Sent message");

A weather report is generated during runtime and sent to the user registered email address using smtp protocol from our application.

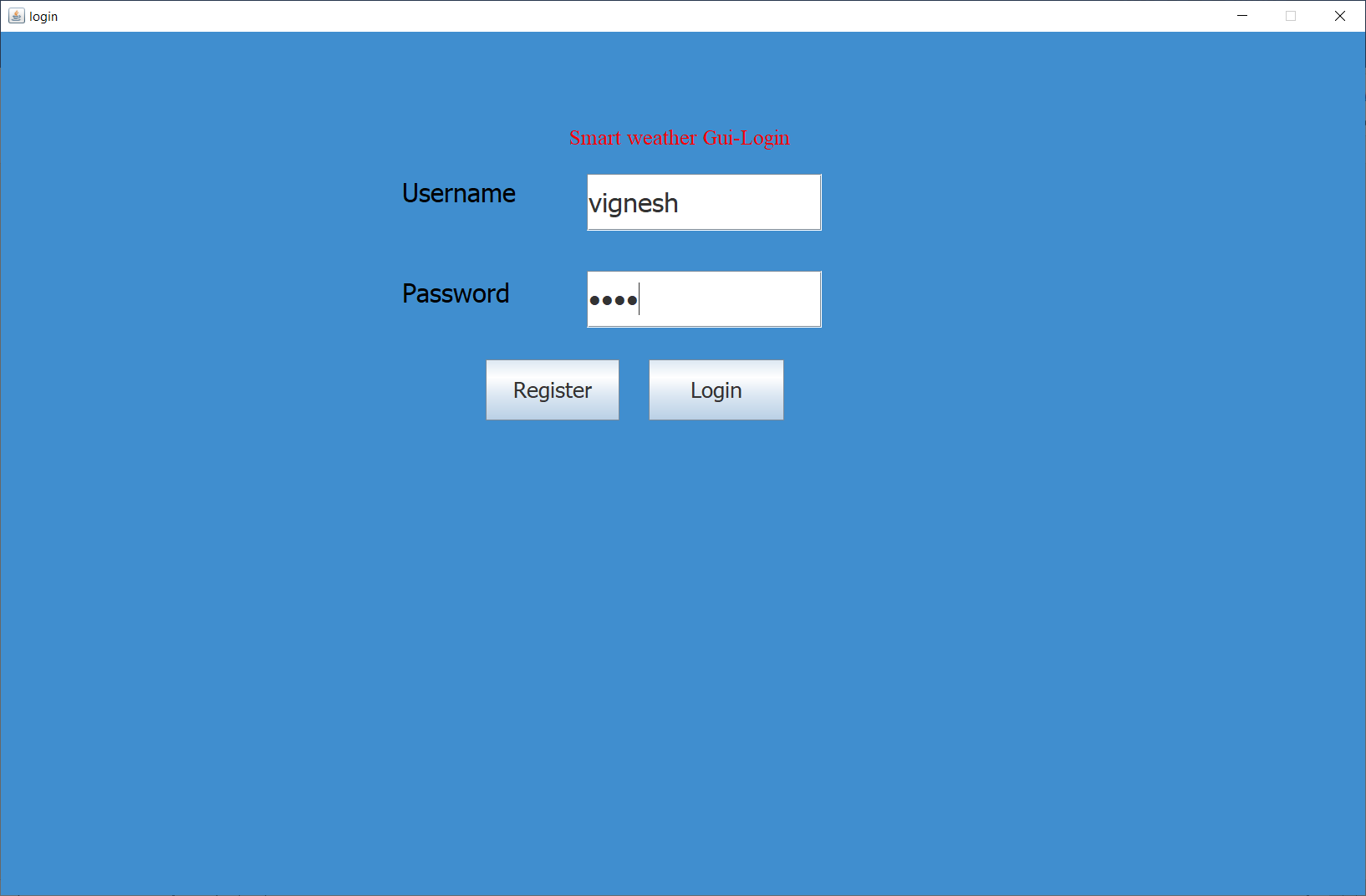
**Implementation:**

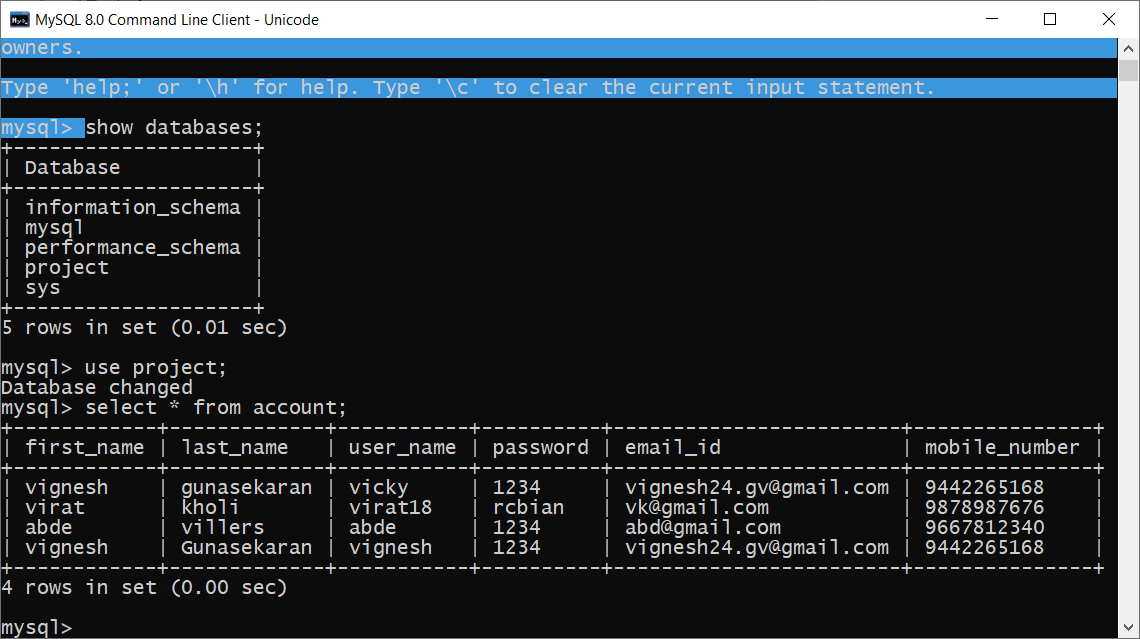
**Module-1: Login and signup page:**

**Registration page:**

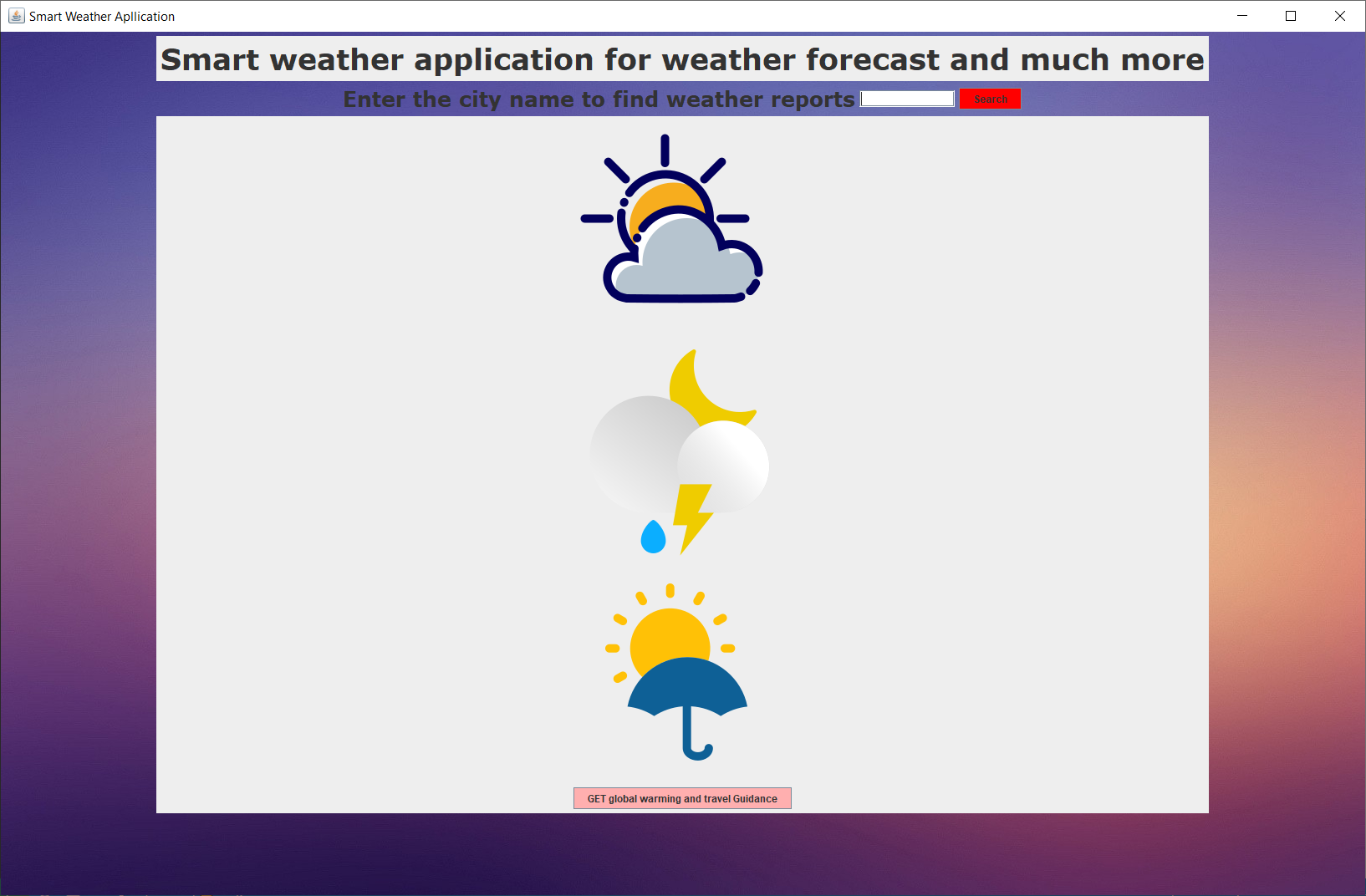


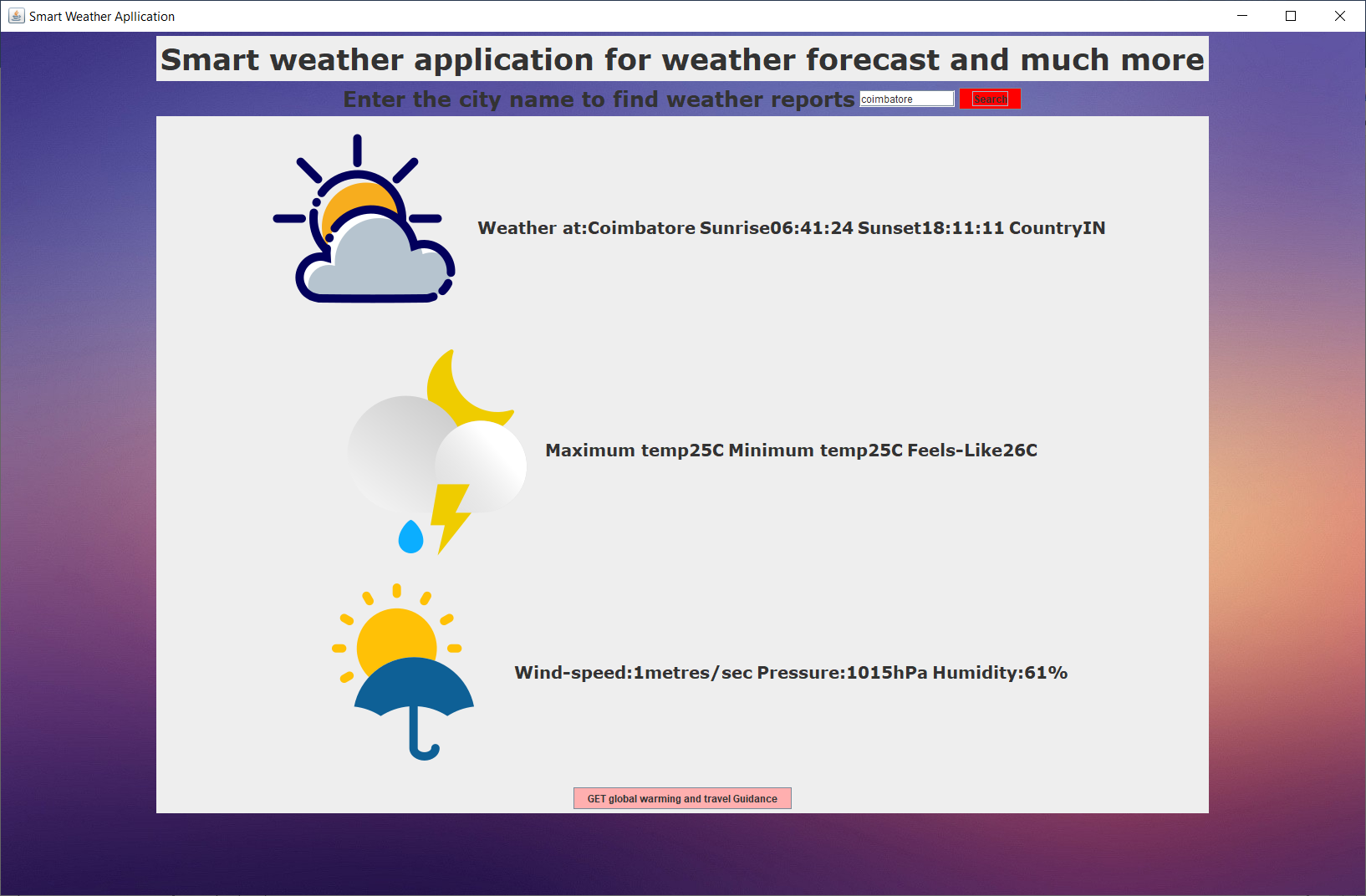
**Login Page:**

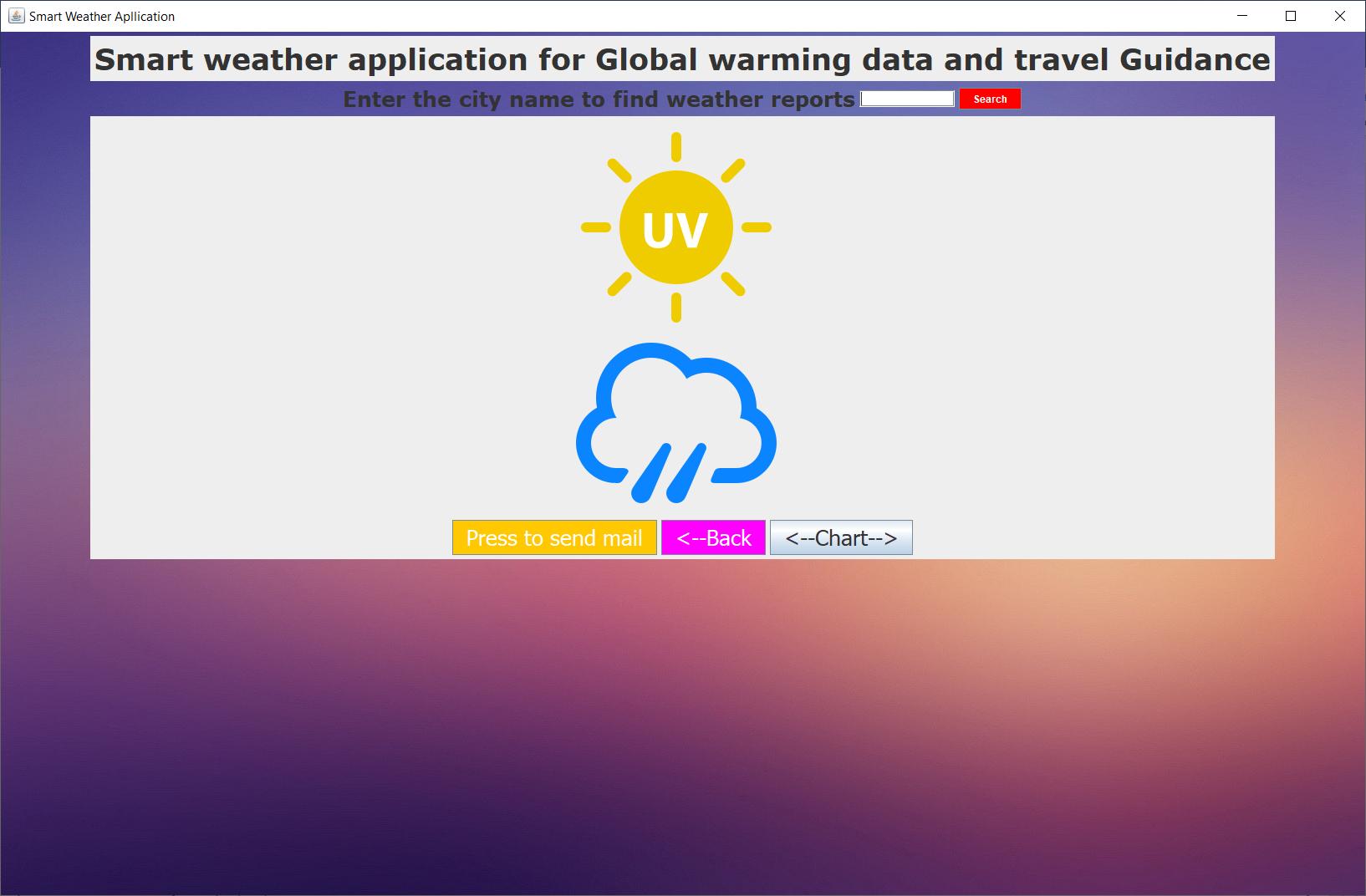


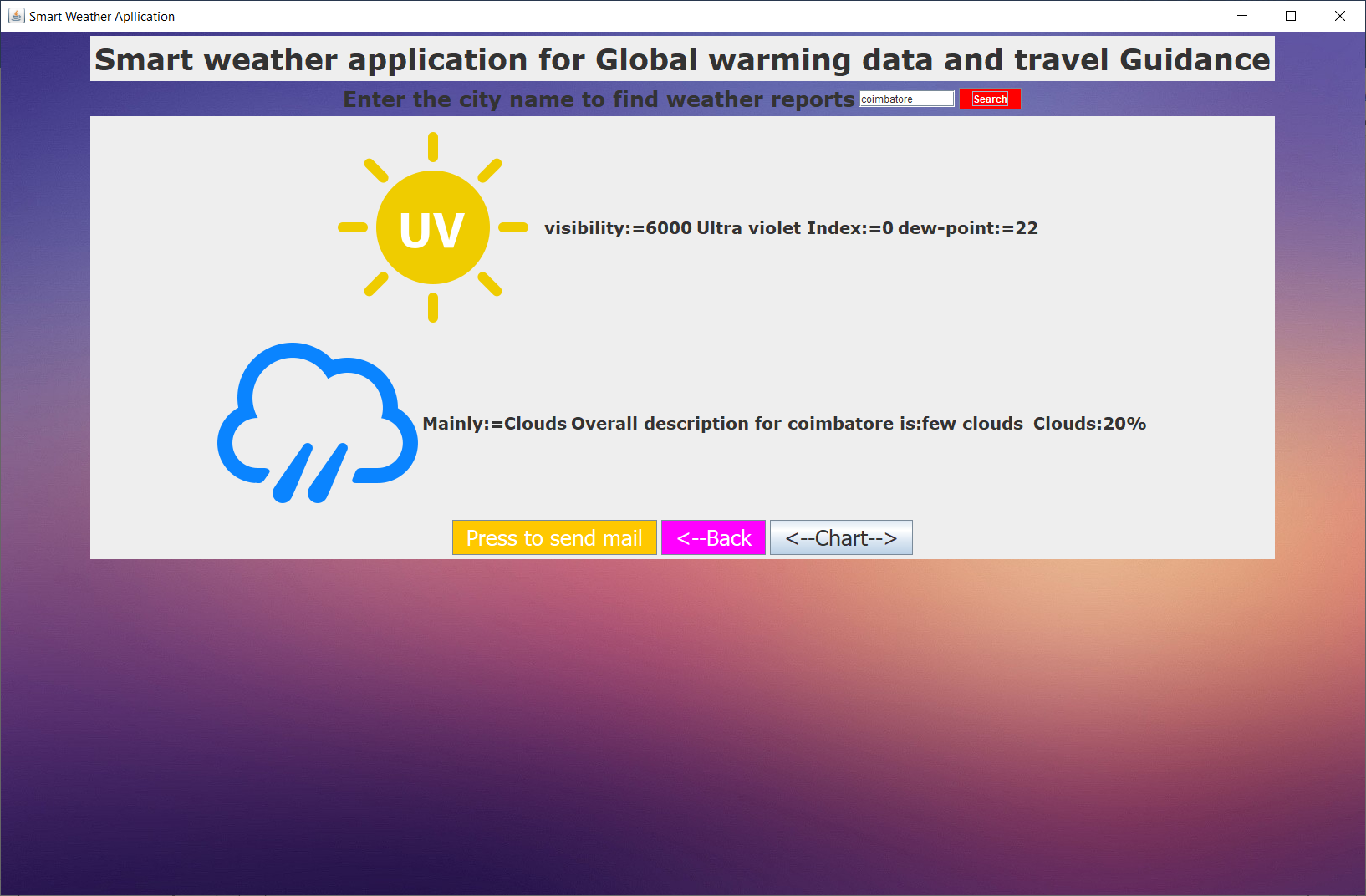


**Module-2&3: Extracting data and displaying in GUI application.**

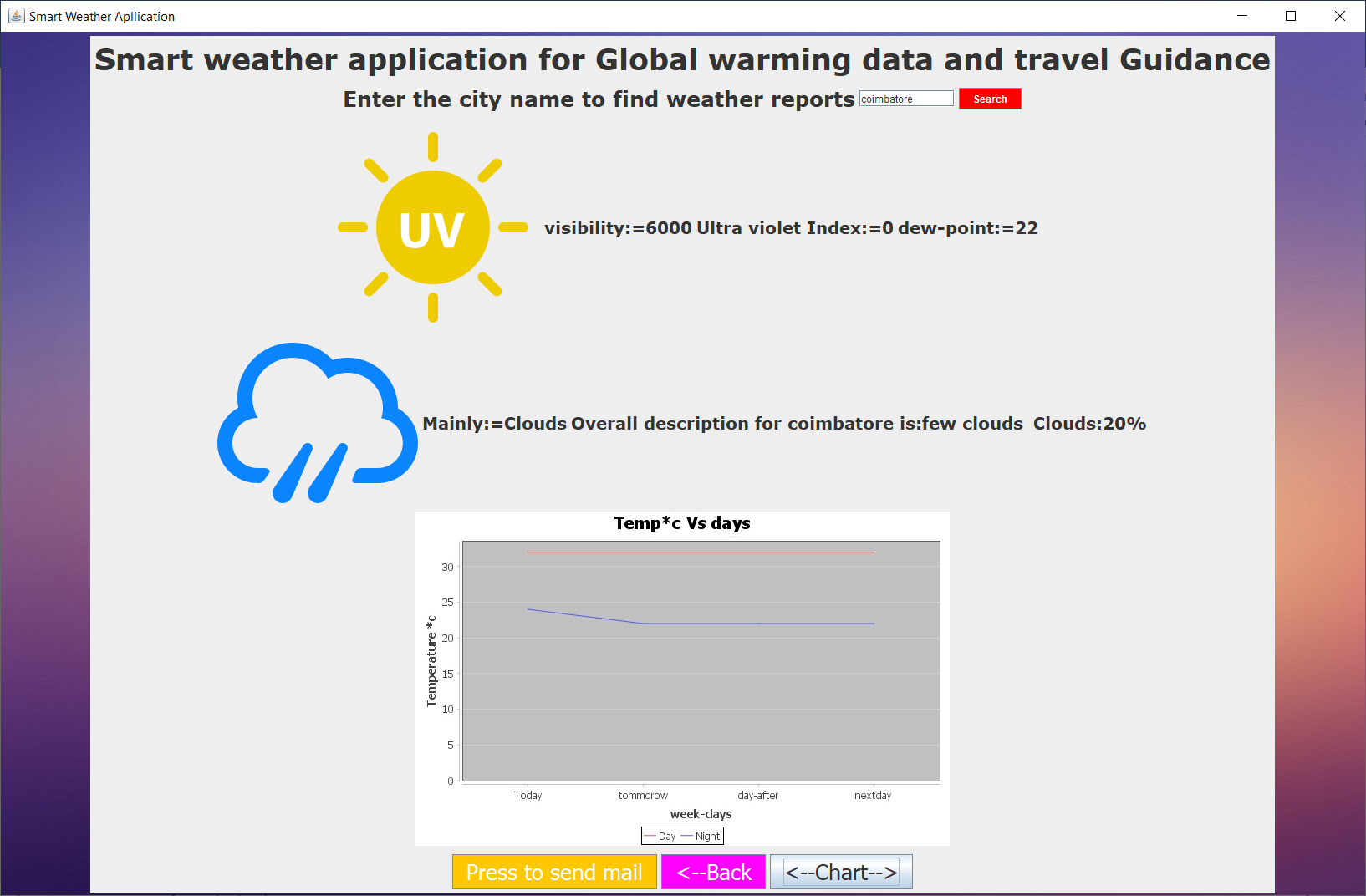


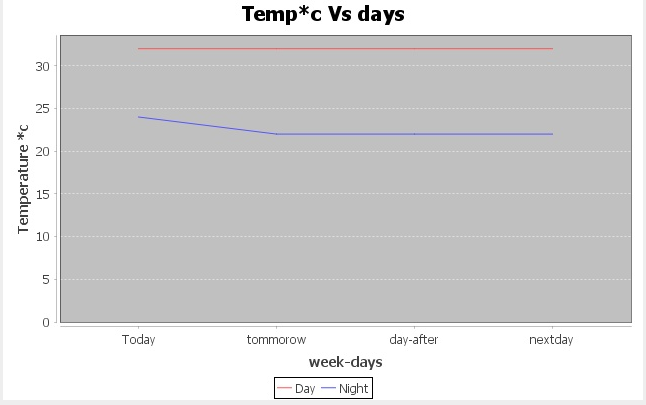


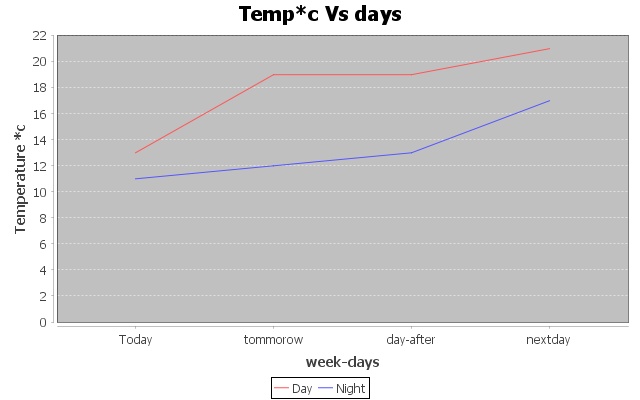




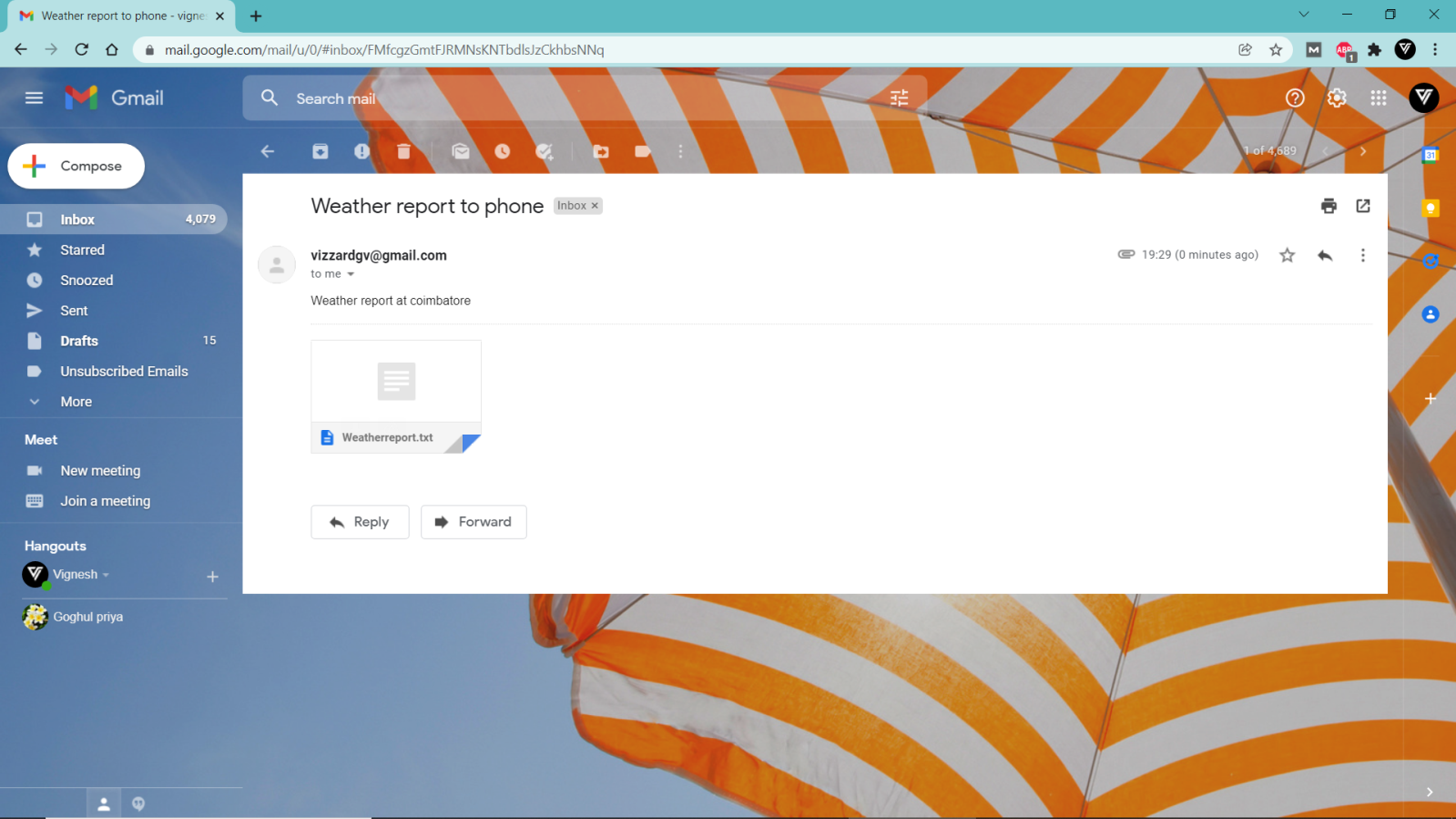
**Module-4: Data visualization using graph for travel Guidance.**



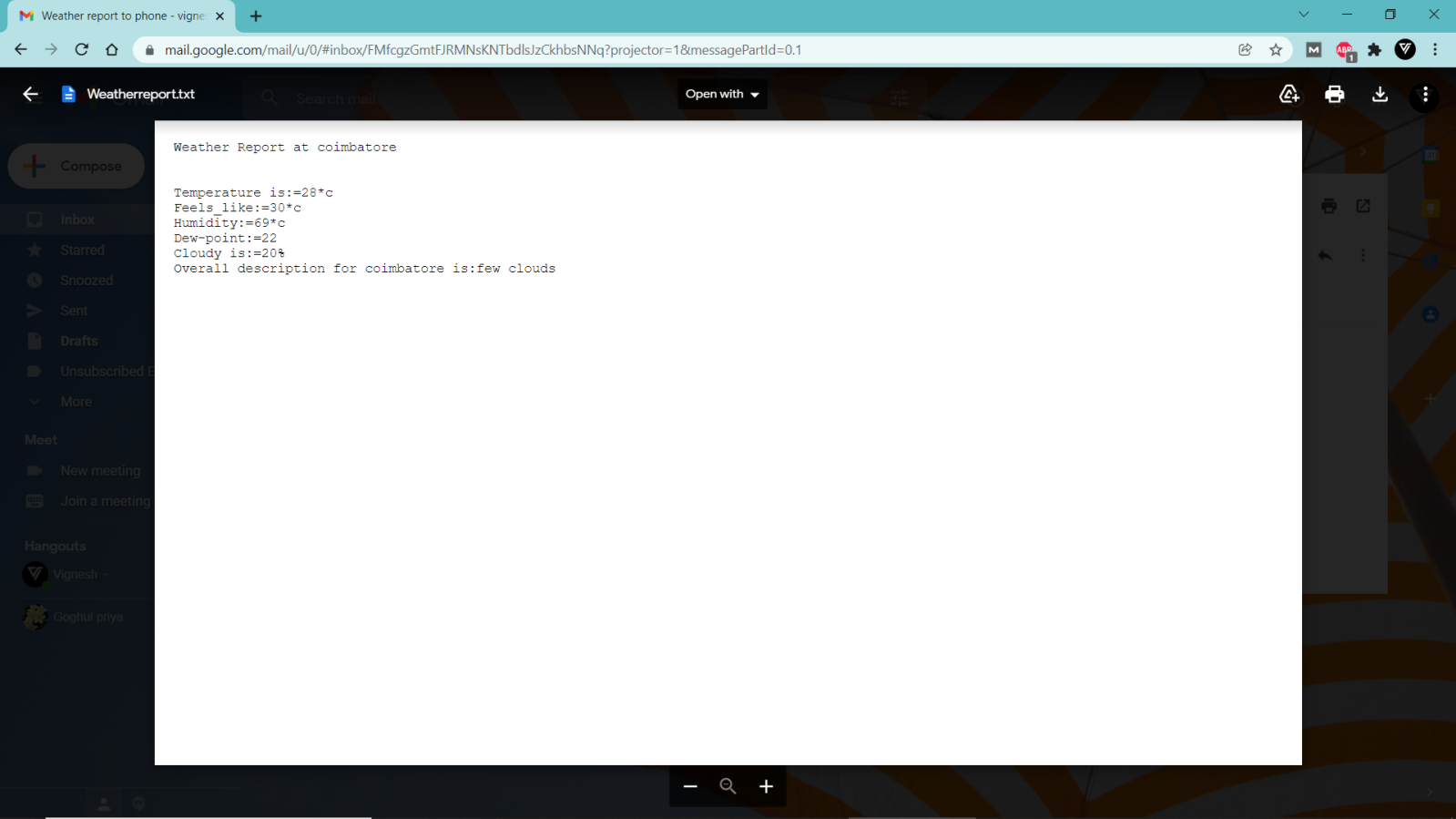


****

**Module-5: Weather report send to User mail.**



**Text weather report file:**



**Conclusion:**

Smart weather GUI application helps us to keep updated about the weather in our locality and make our planning for our daily activities accordingly whether to make any precautionary measures are like carrying an umbrella etc. Weather reports sent to our email helps us to keep track of weather in no network areas. Travel suggestions can be made by visualized graph comparing the temperature for the upcoming week making it easy to understand take travel plans.