**PRACTICAL 01**

**Aim**: Design and develop location-based messaging app.

**Activity\_Main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:background="@color/blue"  
android:gravity="center"  
android:orientation="vertical">  
  
<TextView  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:gravity="center"  
android:text="Get Location Latitude and Longitude"  
android:textColor="@color/white"  
android:textSize="25dp" />  
  
<ImageView  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:layout\_gravity="center"  
android:layout\_marginTop="30dp"  
android:src="@drawable/location" />  
  
<TextView  
android:id="@+id/textview\_latitude"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:layout\_gravity="center"  
android:layout\_marginTop="30dp"  
android:textColor="@color/white"  
android:textSize="22dp" />  
  
<TextView  
android:id="@+id/textview\_longitude"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:layout\_gravity="center"  
android:layout\_marginTop="30dp"  
android:textColor="@color/white"  
android:textSize="22dp" />  
  
</LinearLayout>

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package=""> // here it will be your own package name

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<activity

android:name=".AndroidLocationActivity"

android:label="@string/app\_name"

android:screenOrientation="portrait"

android:theme="@android:style/Theme.NoTitleBar.Fullscreen">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**Strings.xml ( in res-values folder)**

<resources>

<string name="app\_name">AndroidLocationAPIExample</string>

<string name="latitude\_string">Latitude now:</string>

<string name="longitude\_string">Longitude now:</string>

</resources>

**In gradle file add :-**

applyplugin: 'com.android.application'

android {

compileSdkVersion23

buildToolsVersion'23.0.0'

defaultConfig {

applicationId"" // it will be your own package name

minSdkVersion14

targetSdkVersion19

versionCode1

versionName"1.0"

}

buildTypes {

release {

minifyEnabledfalse

proguardFilesgetDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

compile'com.android.support:appcompat-v7:23.1.1'

compile"com.google.android.gms:play-services:8.3.0" // we added this dependency to include the google play service

}

**MainActivity.java**

package // here add your own package name

importandroid.app.Activity;

importandroid.location.Location;

importandroid.os.Bundle;

importandroid.os.Handler;

importandroid.widget.TextView;

importandroid.widget.Toast;

importcom.google.android.gms.common.ConnectionResult;

importcom.google.android.gms.common.GooglePlayServicesUtil;

importcom.google.android.gms.common.api.GoogleApiClient;

importcom.google.android.gms.location.LocationListener;

importcom.google.android.gms.location.LocationRequest;

importcom.google.android.gms.location.LocationServices;

public class AndroidLocationActivity extends Activity implements LocationListener {

// Google client to interact with Google API

privateGoogleApiClientmGoogleApiClient;

privateLocationRequestmLocationRequest;

privateTextView latitude, longitude;

private double fusedLatitude = 0.0;

private doublefusedLongitude = 0.0;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

initializeViews();

if(checkPlayServices()) {

startFusedLocation();

registerRequestUpdate(this);

}

}

private void initializeViews() {

latitude= (TextView) findViewById(R.id.*textview\_latitude*);

longitude= (TextView) findViewById(R.id.*textview\_longitude*);

}

@Override

protected void onStop() {

stopFusedLocation();

super.onStop();

}

*// check if google play services is installed on the device*

private booleancheckPlayServices() {

intresultCode = GooglePlayServicesUtil

.*isGooglePlayServicesAvailable*(this);

if(resultCode != ConnectionResult.*SUCCESS*) {

if(GooglePlayServicesUtil.*isUserRecoverableError*(resultCode)) {

Toast.*makeText*(getApplicationContext(),

"This device is supported. Please download google play services", Toast.*LENGTH\_LONG*)

.show();

} else {

Toast.*makeText*(getApplicationContext(),

"This device is not supported.", Toast.*LENGTH\_LONG*)

.show();

finish();

}

return false;

}

return true;

}

public void startFusedLocation() {

if(mGoogleApiClient == null) {

mGoogleApiClient= new GoogleApiClient.Builder(this).addApi(LocationServices.*API*)

.addConnectionCallbacks(new GoogleApiClient.ConnectionCallbacks() {

@Override

public void onConnectionSuspended(int cause) {

}

@Override

public void onConnected(Bundle connectionHint) {

}

}).addOnConnectionFailedListener(new GoogleApiClient.OnConnectionFailedListener() {

@Override

public void onConnectionFailed(ConnectionResult result) {

}

}).build();

mGoogleApiClient.connect();

} else {

mGoogleApiClient.connect();

}

}

public void stopFusedLocation() {

if(mGoogleApiClient != null) {

mGoogleApiClient.disconnect();

}

}

public void registerRequestUpdate(final LocationListener listener) {

mLocationRequest= LocationRequest.*create*();

mLocationRequest.setPriority(LocationRequest.*PRIORITY\_HIGH\_ACCURACY*);

mLocationRequest.setInterval(1000); *// every second*

newHandler().postDelayed(new Runnable() {

@Override

public void run() {

*// TODO Auto-generated method stub*

try{

LocationServices.*FusedLocationApi*.requestLocationUpdates(mGoogleApiClient, mLocationRequest, listener);

} catch (SecurityException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

if(!isGoogleApiClientConnected()) {

mGoogleApiClient.connect();

}

registerRequestUpdate(listener);

}

}

}, 1000);

}

public booleanisGoogleApiClientConnected() {

returnmGoogleApiClient != null &&mGoogleApiClient.isConnected();

}

@Override

public void onLocationChanged(Location location) {

setFusedLatitude(location.getLatitude());

setFusedLongitude(location.getLongitude());

Toast.*makeText*(getApplicationContext(), "NEW LOCATION RECEIVED", Toast.*LENGTH\_LONG*).show();

latitude.setText(getString(R.string.*latitude\_string*) +" "+ getFusedLatitude());

longitude.setText(getString(R.string.*longitude\_string*) +" "+ getFusedLongitude());

}

public void setFusedLatitude(double lat) {

fusedLatitude= lat;

}

public void setFusedLongitude(double lon) {

fusedLongitude= lon;

}

public double getFusedLatitude() {

returnfusedLatitude;

}

public double getFusedLongitude() {

returnfusedLongitude;

}

}

**PRACTICAL 02**

**Aim:** Install a Cloud Analyst and integrate with IDE Eclipse/Netbeans. Monitor the performance of an Existing Algorithm.

**Main Layout Xml File.**

This layout file contains the RecyclerView and the input text and send button.

**activity\_chat\_app.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ChatAppActivity">

<android.support.v7.widget.RecyclerView

android:id="@+id/chat\_recycler\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<LinearLayout

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<EditText

android:id="@+id/chat\_input\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:hint="Input message."

android:layout\_weight="1"

android:maxLines="2"/>

<Button

android:id="@+id/chat\_send\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Send"

android:textAllCaps="false"/>

</LinearLayout>

</LinearLayout>

**RecyclerView Item Layout Xml File.**

This layout xml file define the RecyclerView’s item view, it contains a left LinearLayout and a right LinearLayout object which contains received and sent message textview object.

**activity\_chat\_app\_item\_view.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="1dp">

<LinearLayout

android:id="@+id/chat\_left\_msg\_layout"

android:layout\_width="99dp"

android:layout\_height="94dp"

android:layout\_gravity="left"

android:background="@drawable/chat\_bubble"

android:orientation="vertical">

<TextView

android:id="@+id/chat\_left\_msg\_text\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

android:layout\_margin="25dp" />

</LinearLayout>

<LinearLayout

android:id="@+id/chat\_right\_msg\_layout"

android:layout\_width="97dp"

android:layout\_height="91dp"

android:layout\_gravity="right"

android:background="@drawable/chat\_bubble"

android:orientation="vertical">

<TextView

android:id="@+id/chat\_right\_msg\_text\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

android:layout\_margin="25dp" />

</LinearLayout>

</LinearLayout>

**ChatAppActivity.java**

package com.example.sahil.chatapp;

import android.app.Activity;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

//import // your own package name.R;

import java.util.ArrayList;

import java.util.List;

public class ChatAppActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_chat\_app);

setTitle("dev2qa.com - Android Chat App Example");

// Get RecyclerView object.

final RecyclerViewmsgRecyclerView = (RecyclerView)findViewById(R.id.chat\_recycler\_view);

// Set RecyclerView layout manager.

LinearLayoutManagerlinearLayoutManager = new LinearLayoutManager(this);

msgRecyclerView.setLayoutManager(linearLayoutManager);

// Create the initial data list.

final List<ChatAppMsgDTO>msgDtoList = new ArrayList<ChatAppMsgDTO>();

ChatAppMsgDTOmsgDto = new ChatAppMsgDTO(ChatAppMsgDTO.MSG\_TYPE\_RECEIVED, "hello");

msgDtoList.add(msgDto);

// Create the data adapter with above data list.

final ChatAppMsgAdapterchatAppMsgAdapter = new ChatAppMsgAdapter(msgDtoList);

// Set data adapter to RecyclerView.

msgRecyclerView.setAdapter(chatAppMsgAdapter);

final EditTextmsgInputText = (EditText)findViewById(R.id.chat\_input\_msg);

Button msgSendButton = (Button)findViewById(R.id.chat\_send\_msg);

msgSendButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String msgContent = msgInputText.getText().toString();

if(!TextUtils.isEmpty(msgContent))

{

// Add a new sent message to the list.

ChatAppMsgDTOmsgDto = new ChatAppMsgDTO(ChatAppMsgDTO.MSG\_TYPE\_SENT, msgContent);

msgDtoList.add(msgDto);

intnewMsgPosition = msgDtoList.size() - 1;

// Notify recycler view insert one new data.

chatAppMsgAdapter.notifyItemInserted(newMsgPosition);

// Scroll RecyclerView to the last message.

msgRecyclerView.scrollToPosition(newMsgPosition);

// Empty the input edit text box.

msgInputText.setText("");

}

}

});

}

}

**ChatAppMsgAdapter.java**

package com.example.sahil.chatapp;

import android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import java.util.ArrayList;

import java.util.List;

public class ChatAppMsgAdapter extends RecyclerView.Adapter<ChatAppMsgViewHolder> {

private List<ChatAppMsgDTO>msgDtoList = null;

public ChatAppMsgAdapter(List<ChatAppMsgDTO>msgDtoList) {

this.msgDtoList = msgDtoList;

}

@Override

public void onBindViewHolder(ChatAppMsgViewHolder holder, int position) {

ChatAppMsgDTOmsgDto = this.msgDtoList.get(position);

// If the message is a received message.

if(msgDto.MSG\_TYPE\_RECEIVED.equals(msgDto.getMsgType()))

{

// Show received message in left linearlayout.

holder.leftMsgLayout.setVisibility(LinearLayout.VISIBLE);

holder.leftMsgTextView.setText(msgDto.getMsgContent());

// Remove left linearlayout.The value should be GONE, can not be INVISIBLE

// Otherwise each iteview's distance is too big.

holder.rightMsgLayout.setVisibility(LinearLayout.GONE);

}

// If the message is a sent message.

else if(msgDto.MSG\_TYPE\_SENT.equals(msgDto.getMsgType()))

{

// Show sent message in right linearlayout.

holder.rightMsgLayout.setVisibility(LinearLayout.VISIBLE);

holder.rightMsgTextView.setText(msgDto.getMsgContent());

// Remove left linearlayout.The value should be GONE, can not be INVISIBLE

// Otherwise each iteview's distance is too big.

holder.leftMsgLayout.setVisibility(LinearLayout.GONE);

}

}

@Override

public ChatAppMsgViewHolderonCreateViewHolder(ViewGroup parent, intviewType) {

LayoutInflaterlayoutInflater = LayoutInflater.from(parent.getContext());

View view = layoutInflater.inflate(R.layout.activity\_chat\_app\_item\_view, parent, false);

return new ChatAppMsgViewHolder(view);

}

@Override

public intgetItemCount() {

if(msgDtoList==null)

{

msgDtoList = new ArrayList<ChatAppMsgDTO>();

}

return msgDtoList.size();

}

}

**ChatAppMsgViewHolder.java**

package com.example.sahil.chatapp;

import android.support.v7.widget.RecyclerView;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.TextView;

//import com.dev2qa.example.R;

public class ChatAppMsgViewHolder extends RecyclerView.ViewHolder {

LinearLayoutleftMsgLayout;

LinearLayoutrightMsgLayout;

TextViewleftMsgTextView;

TextViewrightMsgTextView;

public ChatAppMsgViewHolder(View itemView) {

super(itemView);

if(itemView!=null) {

leftMsgLayout = (LinearLayout) itemView.findViewById(R.id.chat\_left\_msg\_layout);

rightMsgLayout = (LinearLayout) itemView.findViewById(R.id.chat\_right\_msg\_layout);

leftMsgTextView = (TextView) itemView.findViewById(R.id.chat\_left\_msg\_text\_view);

rightMsgTextView = (TextView) itemView.findViewById(R.id.chat\_right\_msg\_text\_view);

}

}

}

**ChatAppMsgDTO.java**

package com.example.sahil.chatapp;

public class ChatAppMsgDTO {

public final static String MSG\_TYPE\_SENT = "MSG\_TYPE\_SENT";

public final static String MSG\_TYPE\_RECEIVED = "MSG\_TYPE\_RECEIVED";

// Message content.

private String msgContent;

// Message type.

private String msgType;

public ChatAppMsgDTO(String msgType, String msgContent) {

this.msgType = msgType;

this.msgContent = msgContent;

}

public String getMsgContent() {

return msgContent;

}

public void setMsgContent(String msgContent) {

this.msgContent = msgContent;

}

public String getMsgType() {

return msgType;

}

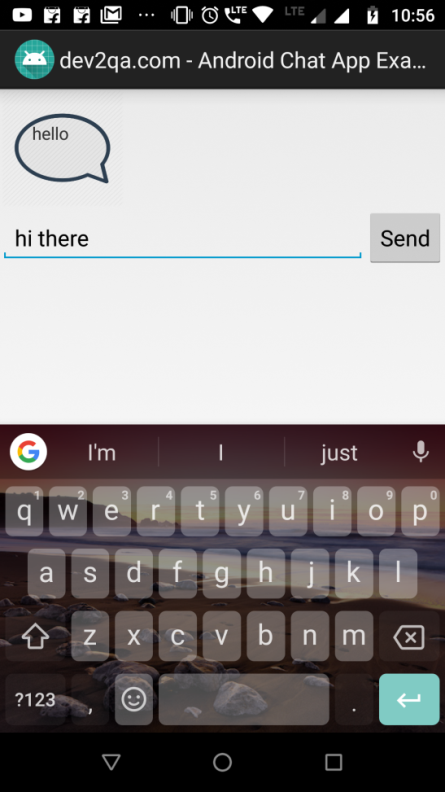
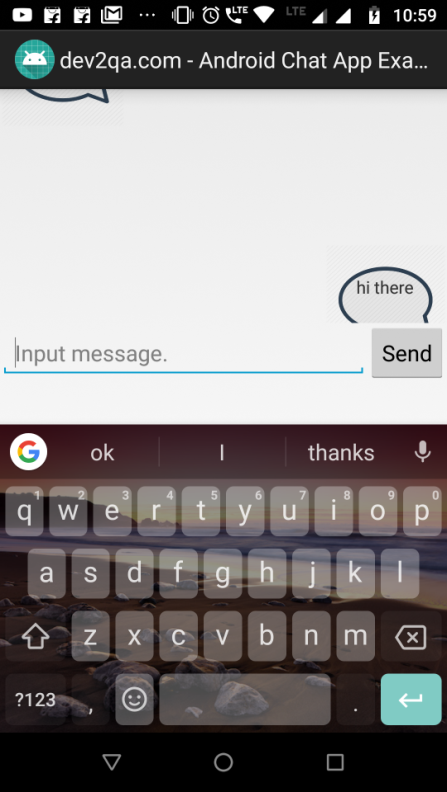
public void setMsgType(String msgType) {

this.msgType = msgType;

}

}

**Output :-**

**PRACTICAL 03**

**Aim:** Build an Application on Private Cloud.

**Step 1** Create new app in android studio

**Step 2** Go to Tools - Firebase - Connect to firebase - ( Make sure you have signed in to google account )

**Step 3** Create a new java class , add the following code

packagecom.example.abc.fcmexample;

importandroid.util.Log;

importcom.google.firebase.iid.FirebaseInstanceId;

importcom.google.firebase.iid.FirebaseInstanceIdService;

import static android.content.ContentValues.*TAG*;

public class MyFirebaseInstanceIdServiceextendsFirebaseInstanceIdService {

@Override

public void onTokenRefresh() {

*// Get updated InstanceID token.*

String refreshedToken = FirebaseInstanceId.*getInstance*().getToken();

Log.*d*(*TAG*, "Refreshed token: " + refreshedToken);

*// If you want to send messages to this application instance or*

*// manage this apps subscriptions on the server side, send the*

*// Instance ID token to your app server.*

*// sendRegistrationToServer(refreshedToken);*

}

}

**Step 4** Create a new Service , add the following code

packagecom.example.abc.fcmexample;

importandroid.app.NotificationManager;

importandroid.app.PendingIntent;

importandroid.app.Service;

importandroid.content.Context;

importandroid.content.Intent;

importandroid.media.Ringtone;

importandroid.media.RingtoneManager;

importandroid.net.Uri;

importandroid.os.IBinder;

importandroid.support.v4.app.NotificationCompat;

importandroid.util.Log;

importcom.google.firebase.messaging.FirebaseMessagingService;

importcom.google.firebase.messaging.RemoteMessage;

import static android.content.ContentValues.*TAG*;

public class MyFirebaseMessagingService extends FirebaseMessagingService {

publicMyFirebaseMessagingService() {

}

@Override

public void onMessageReceived(RemoteMessageremoteMessage) {

super.onMessageReceived(remoteMessage);

sendNotification(remoteMessage.getNotification().getBody());

}

private void sendNotification(String messageBody)

{

Intent intent = new Intent(this,MainActivity.class);

intent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);

PendingIntentpendingIntent = PendingIntent.*getActivity*(this,0,intent,PendingIntent.*FLAG\_ONE\_SHOT*);

Uri defaultSoundUri = RingtoneManager.*getDefaultUri*(RingtoneManager.*TYPE\_NOTIFICATION*);

NotificationCompat.BuildernotificationBuilder = new NotificationCompat.Builder(this);

notificationBuilder.setSmallIcon(R.drawable.*ic\_stat\_name*);

notificationBuilder.setContentTitle("FCM Example by Dipti");

notificationBuilder.setContentText(messageBody);

notificationBuilder.setAutoCancel(true);

notificationBuilder.setSound(defaultSoundUri);

notificationBuilder.setContentIntent(pendingIntent);

NotificationManagernotificationManager = (NotificationManager)getSystemService(Context.*NOTIFICATION\_SERVICE*);

notificationManager.notify(0,notificationBuilder.build());

}

}

**Step 5** Modify the manifest file

*<?*xml version="1.0" encoding="utf-8"*?>*

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.abc.fcmexample">

<uses-permissionandroid:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<service android:name=".MyFirebaseInstanceIdService">

<intent-filter>

<action android:name="com.google.firebase.INSTANCE\_ID\_EVENT" />

</intent-filter>

</service>

<service

android:name=".MyFirebaseMessagingService"

android:enabled="true"

android:exported="true">

<intent-filter>

<action android:name="com.google.firebase.INSTANCE\_ID\_EVENT" />

</intent-filter>

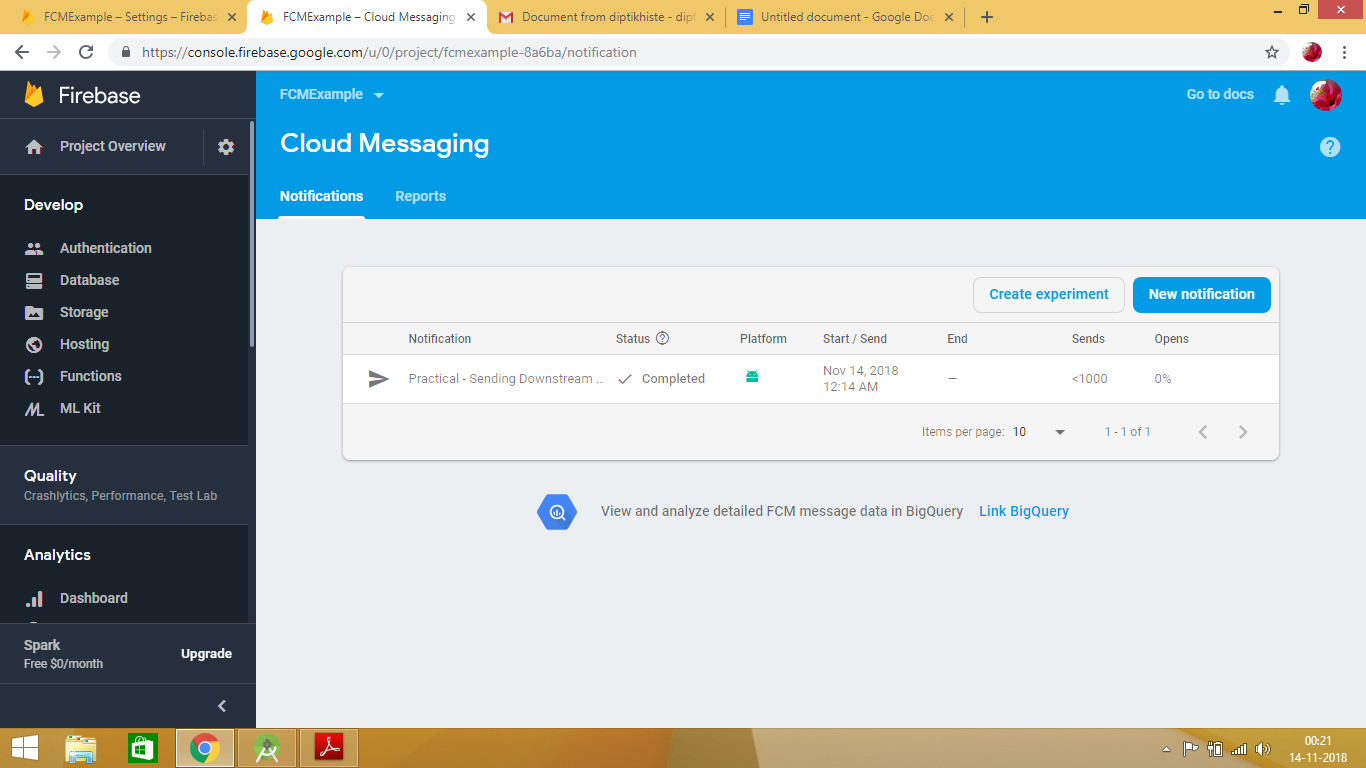
</service>

</application>

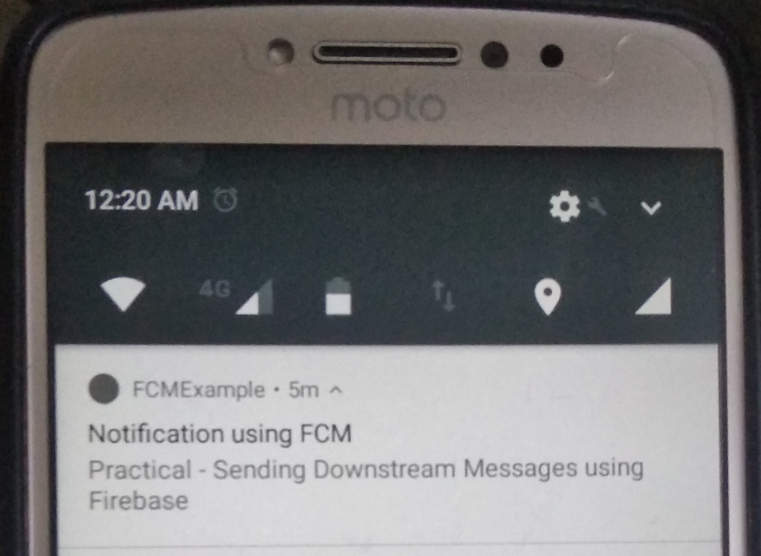
</manifest>

**Step 6** go to Notification composer in firebase console

Add the notification , click on publish



**Output:**

****

**PRACTICAL 04**

**Aim:** Demonstrate any Cloud Monitoring Tool.

**Required Software:**

JDK 1.6(Any Version)

XAMPP Server

**Mysql-java connector(copy .jar file and paste it into)**

----- C:\Program Files\Java\jdk1.8.0\_111\jre\lib\ext

----- C:\Program Files\Java\jre1.8.0\_111\lib\ext

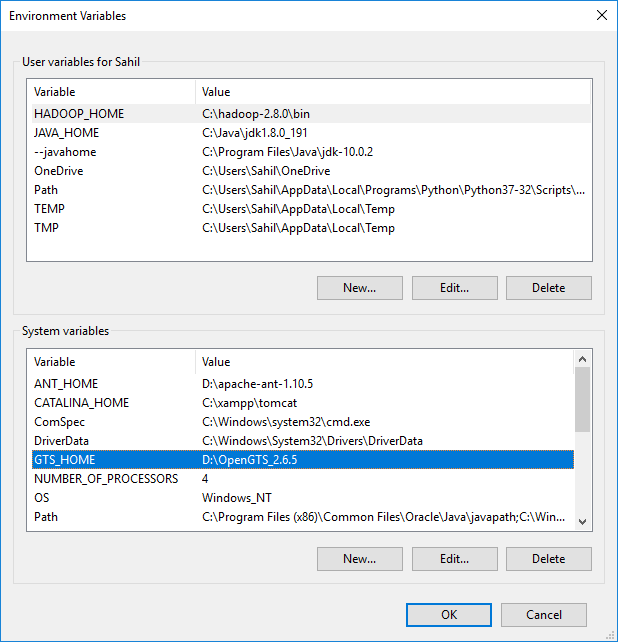
**OpenGTS application :-** [**http://www.opengts.org/**](http://www.opengts.org/)

**Extract File in the Drive :-** OpenGTS\_2.6.5.zip

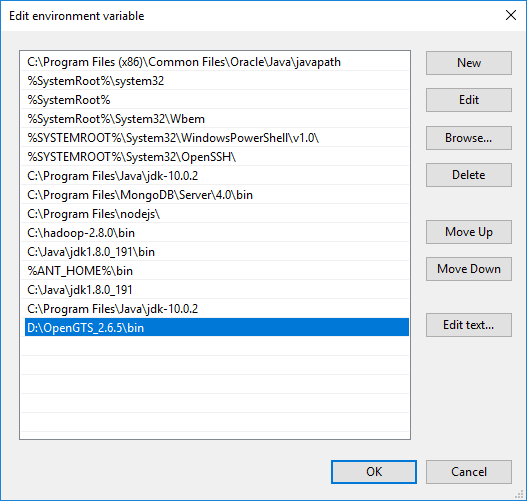
**Apache Ant :-** Extract File in the Drive :- apache-ant-1.10.5-bin.zip

**Settings :-**

Add paths in Environment Variables [Apache Ant, OpenGTS, Java JDK and Tomcat]



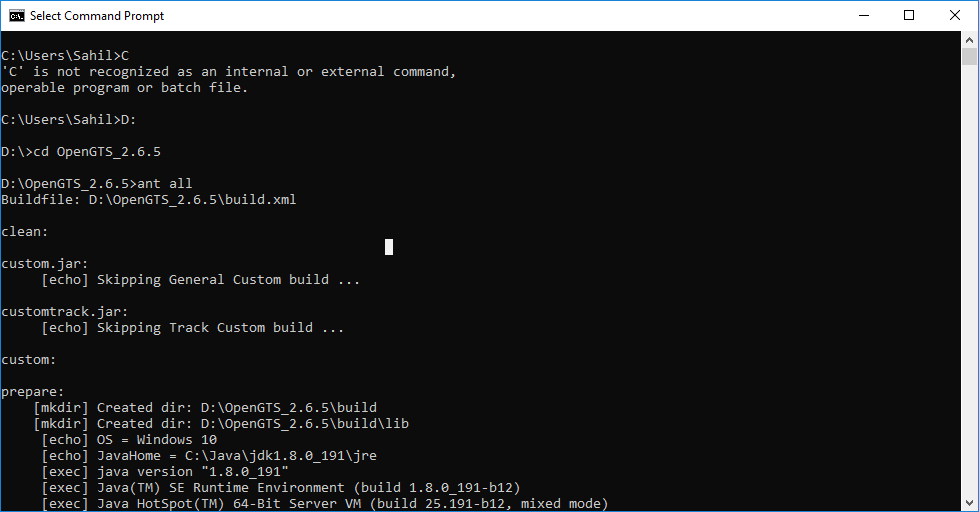
**Path Variable:**



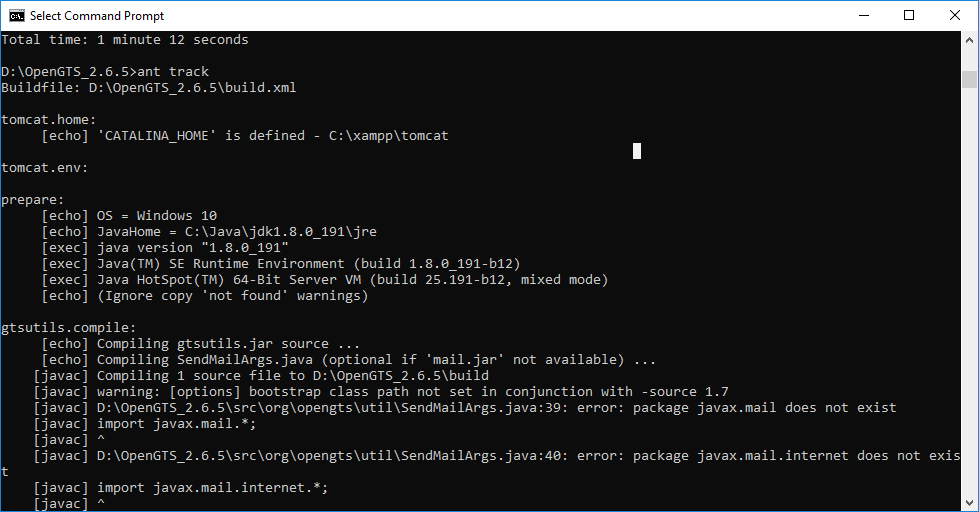
**ADMIN CMD :-**

Open command Prompt and go to D:\OpenGTS\_2.6.2

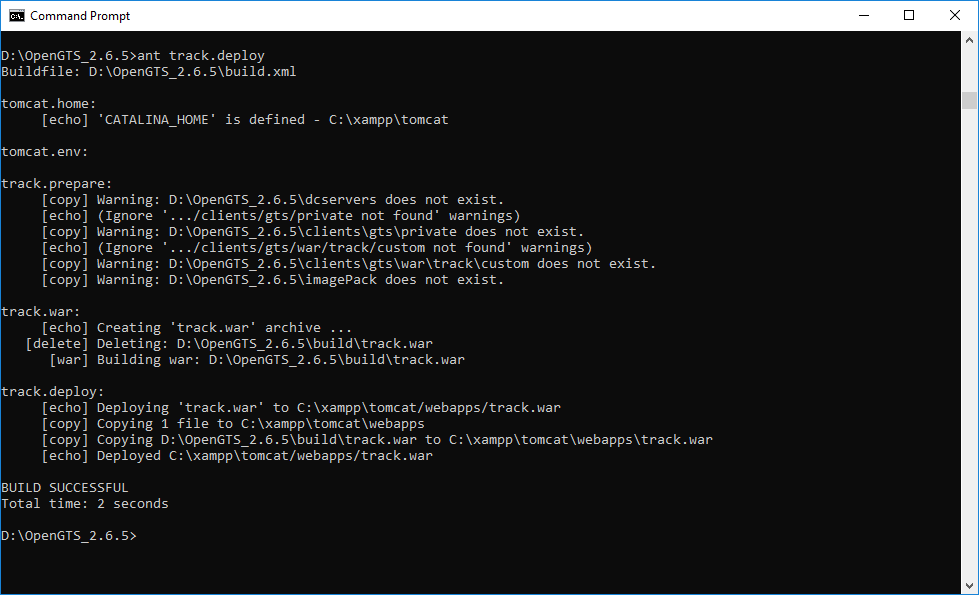
**Type command :-** ant all



**Type command :-**ant track



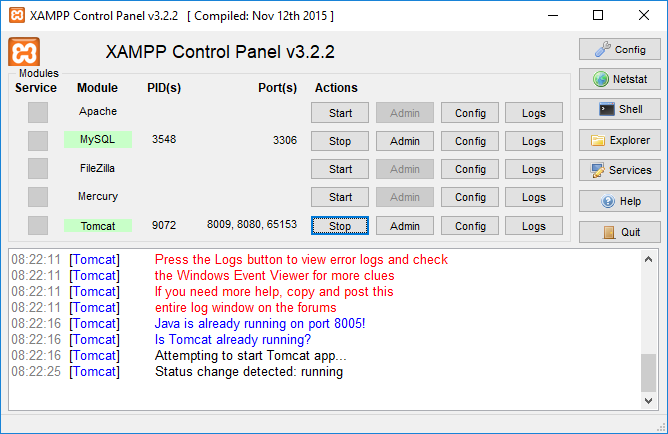
**Type command :-** ant track.deploy

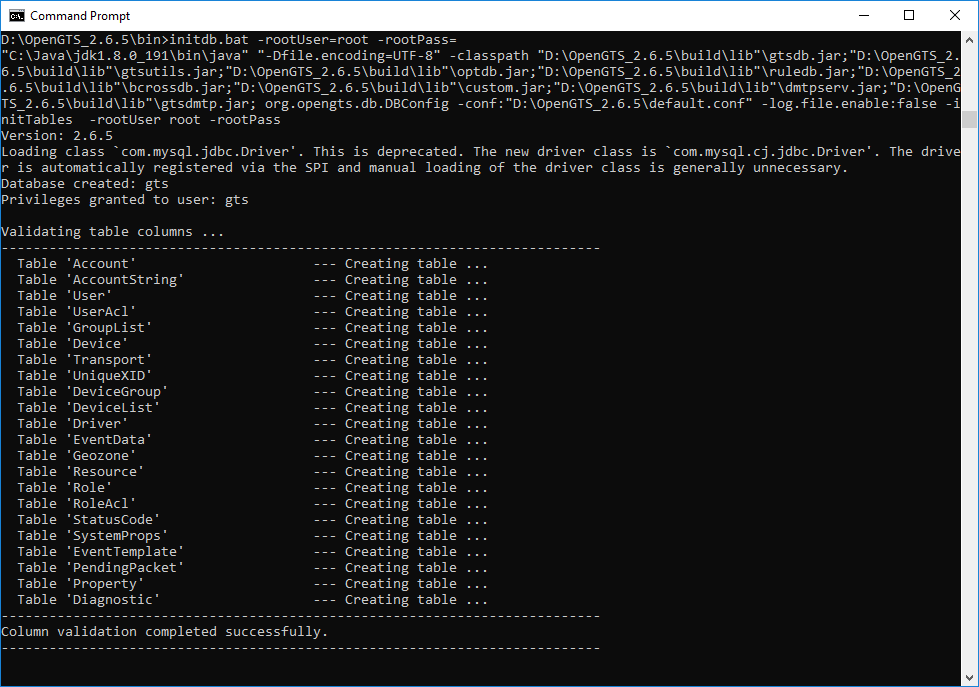


**Type :-** cd bin

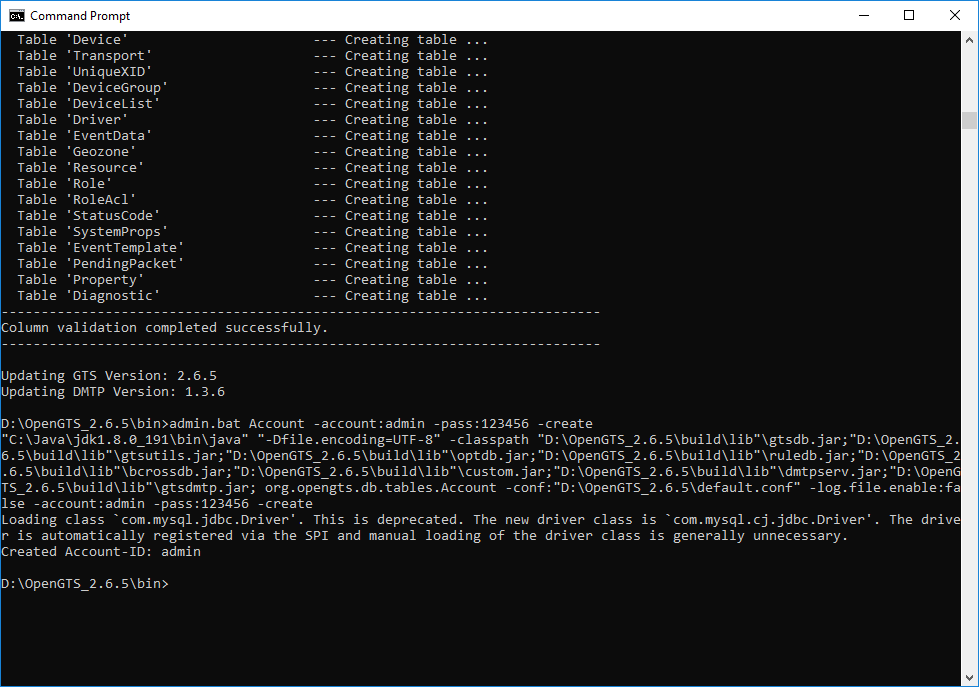
Then start Tomcat and MySQL in XAMMP Server.

**Type command :-** initdb.bat –rootUser=root –rootPass=

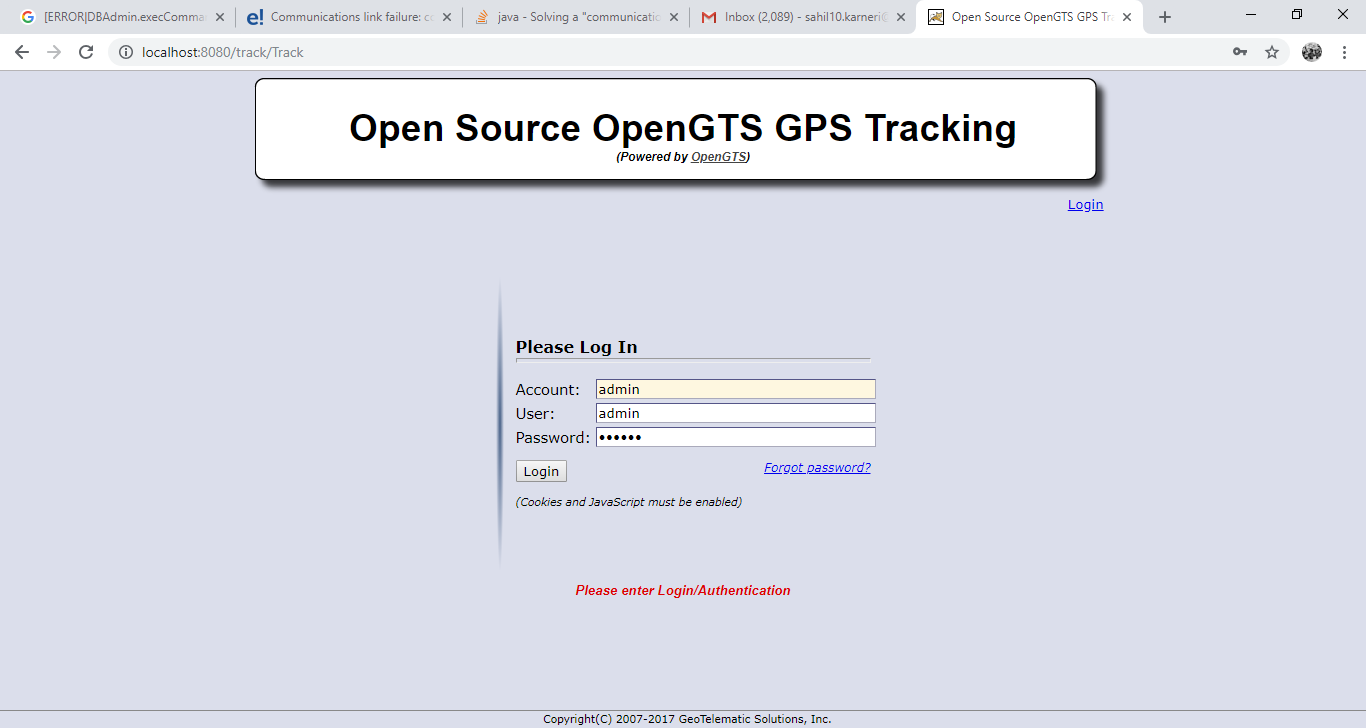


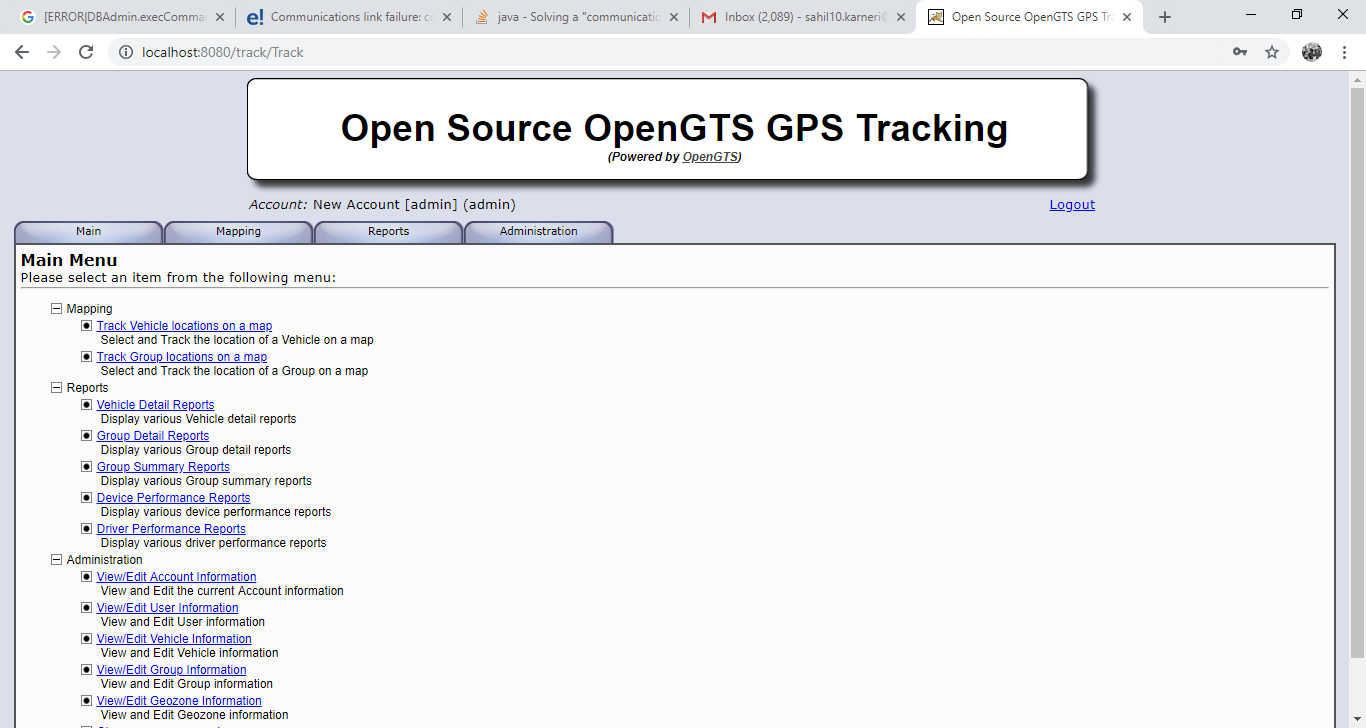


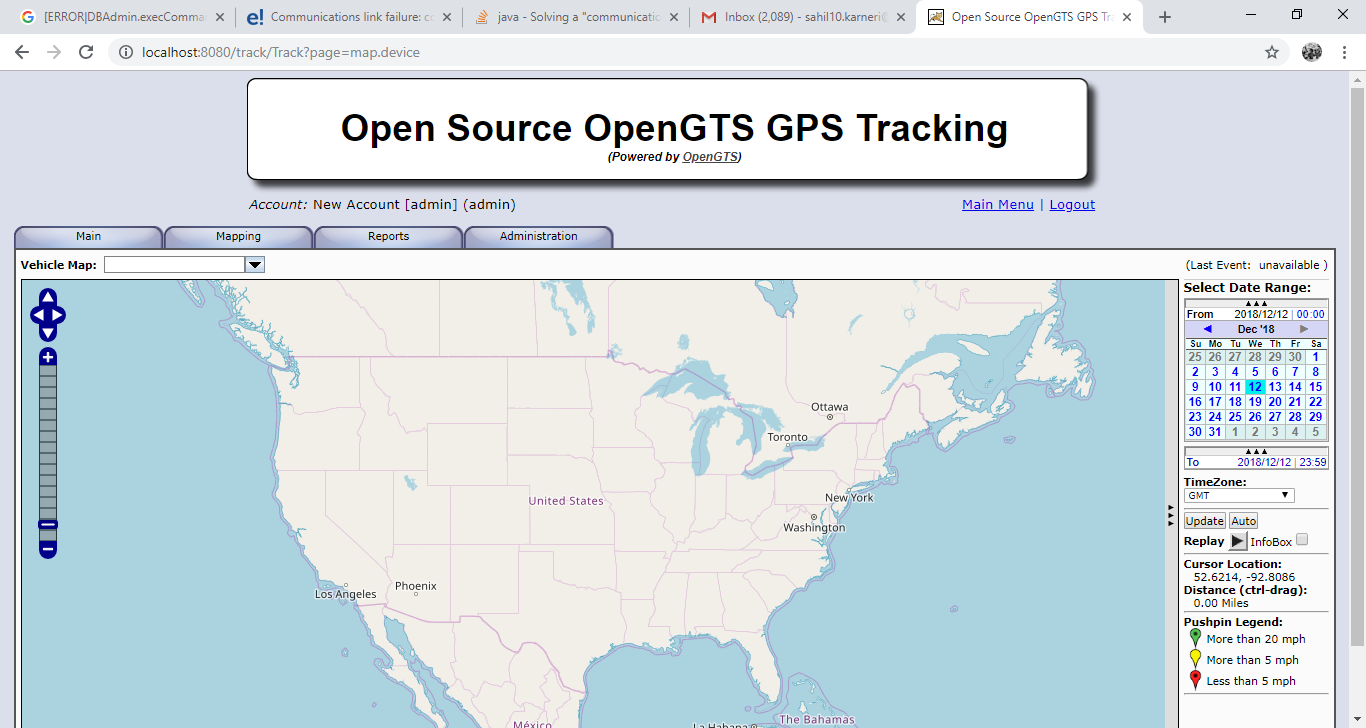
**Type Command :-** admin.bat Account –account:admin –pass:123456 –create



Type url 127.0.0.1:8080/track/Track and login with admin and 123456







**PRACTICAL 05**

**Aim:** Implement FOSS-Cloud Functionality – Virtual Desktop Infrastructure (VDI).

**Program :-**

package ppp;

import java.awt.Color;

import java.awt.Frame;

import java.awt.event.WindowEvent;

import java.awt.event.WindowListener;

public class CloseableSimpleWarning extends Frame implements WindowListener{

public CloseableSimpleWarning() {

setBackground(Color.CYAN);

setTitle("Warning");

setSize(250,250);

addWindowListener(this);

}

@Override

public void windowOpened(WindowEvent e) {

System.out.println("windowOpened");

}

@Override

public void windowClosing(WindowEvent e) {

//System.out.println("windowClosing");

System.exit(0);

}

@Override

public void windowClosed(WindowEvent e) {

System.out.println("windowClosed");

}

@Override

public void windowIconified(WindowEvent e) {

System.out.println("windowIconified");

}

@Override

public void windowDeiconified(WindowEvent e) {

System.out.println("windowDeiconified");

}

@Override

public void windowActivated(WindowEvent e) {

System.out.println("windowActivated");

}

@Override

public void windowDeactivated(WindowEvent e) {

System.out.println("windowDeactivated");

}

public static void main(String[] args){

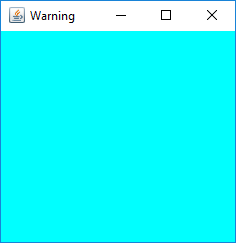
CloseableSimpleWarning frame = new CloseableSimpleWarning();

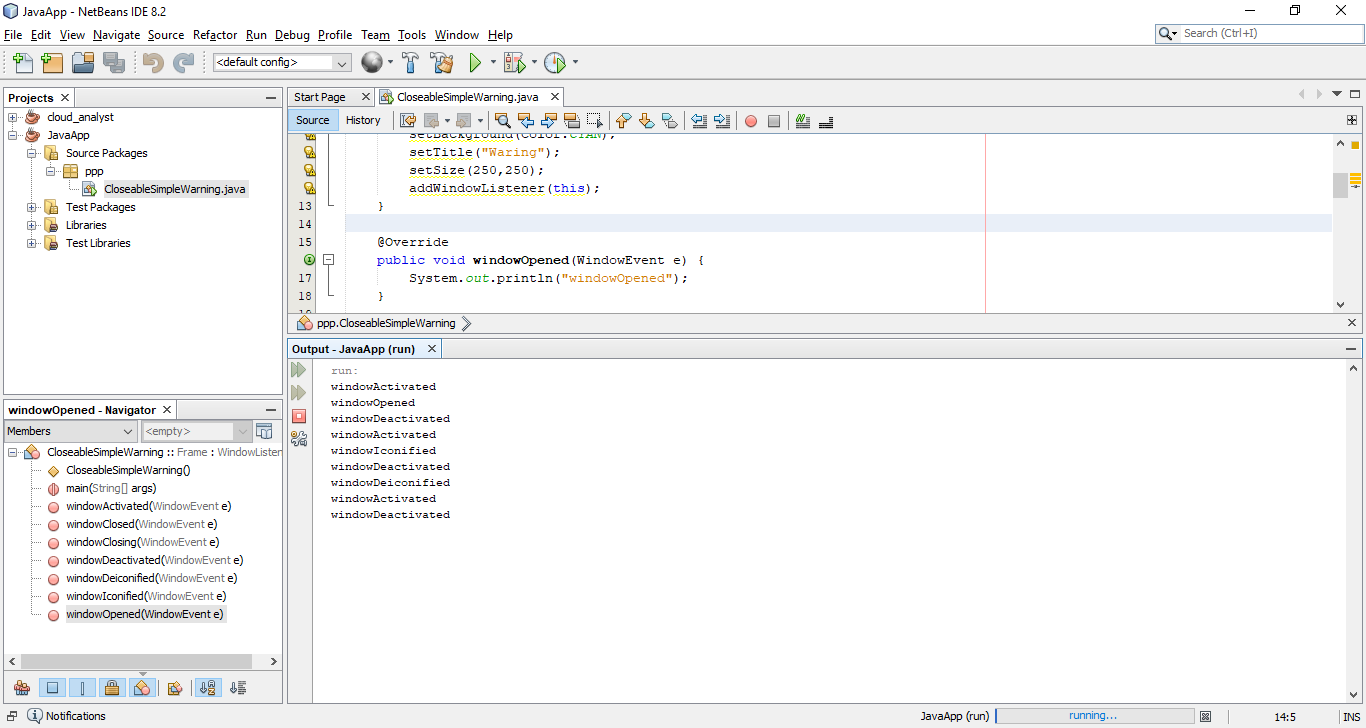
frame.setVisible(true);

}

}

**Output :-**





**PRACTICAL 06**

**Aim:** Implement FOSS-Cloud Functionality – Virtual Server Infrastructure (VSI) – Infrastructure as a Service (IaaS).

**Download Java Card Sdk from**

http://www.oracle.com/technetwork/java/embedded/javacard/downloads/javacard-sdk-2043229.

Html

**Install in Netbeans as plugin :-**

Tools -> Plugin

Create an application

**Code :-**

package classicapplet2;

importjavacard.framework.\*;

public class ClassicApplet2 extends Applet {

/\*\*

\* Installs this applet.

\* @parambArray

\* the array containing installation parameters

\* @parambOffset

\* the starting offset in bArray

\* @parambLength

\* the length in bytes of the parameter data in bArray

\*/

private byte[] received;

private static final short MAX\_LENGTH = 256;

private static final byte[] helloFidesmo = {(byte)'H',(byte)'e',(byte)'l',(byte)'l',(byte)'o',(byte)'

',(byte)'F',(byte)'i',(byte)'d',(byte)'e',(byte)'s',(byte)'m',(byte)'o',(byte)'!'};

public static void install(byte[] bArray, short bOffset, byte bLength) {

new ClassicApplet2();

}

/\*\*

\* Only this class's install method should create the applet object.

\*/

protected ClassicApplet2() {

received = new byte[MAX\_LENGTH];

register();

}

/\*\*

\* Processes an incoming APDU.

\*

\* @see APDU

\* @paramapdu

\* the incoming APDU

\*/

public void process(APDU apdu) {

//Insert your code here

byte buffer[] = apdu.getBuffer();

short length = (short) helloFidesmo.length;

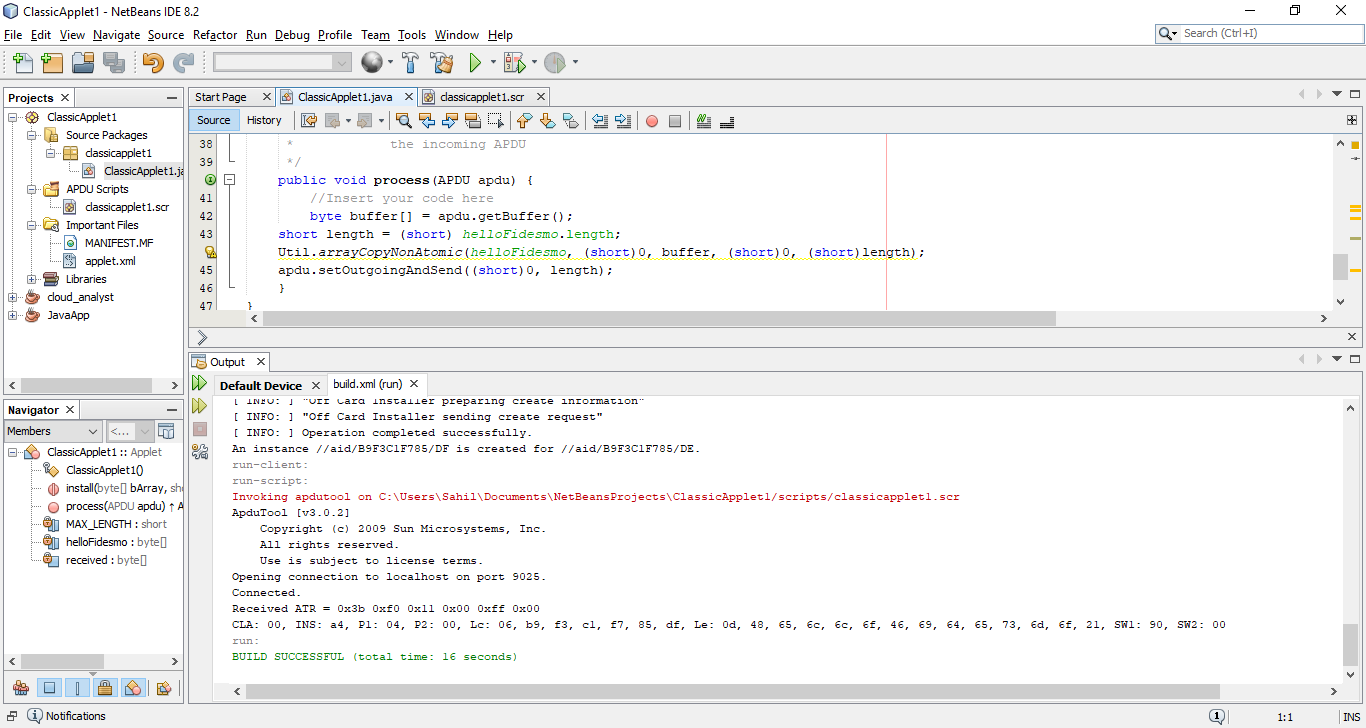
Util.arrayCopyNonAtomic(helloFidesmo, (short)0, buffer, (short)0, (short)length);

apdu.setOutgoingAndSend((short)0, length);

}

}

**Output :-**

****