**package** xyz.lifeissimple.hibuddy;  
  
**import** android.app.NotificationChannel;  
**import** android.app.NotificationManager;  
**import** android.app.PendingIntent;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.graphics.drawable.Drawable;  
**import** android.net.ConnectivityManager;  
**import** android.net.NetworkInfo;  
**import** android.net.Uri;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.Environment;  
**import** android.os.Handler;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.annotation.Nullable;  
  
**import** com.google.android.exoplayer2.DefaultLoadControl;  
**import** com.google.android.exoplayer2.ExoPlaybackException;  
**import** com.google.android.exoplayer2.ExoPlayer;  
**import** com.google.android.exoplayer2.ExoPlayerFactory;  
**import** com.google.android.exoplayer2.LoadControl;  
**import** com.google.android.exoplayer2.Player;  
**import** com.google.android.exoplayer2.SimpleExoPlayer;  
**import** com.google.android.exoplayer2.source.MediaSource;  
**import** com.google.android.exoplayer2.source.ProgressiveMediaSource;  
**import** com.google.android.exoplayer2.trackselection.AdaptiveTrackSelection;  
**import** com.google.android.exoplayer2.trackselection.DefaultTrackSelector;  
**import** com.google.android.exoplayer2.trackselection.TrackSelection;  
**import** com.google.android.exoplayer2.trackselection.TrackSelector;  
**import** com.google.android.exoplayer2.ui.PlayerView;  
**import** com.google.android.exoplayer2.upstream.BandwidthMeter;  
**import** com.google.android.exoplayer2.upstream.DefaultBandwidthMeter;  
**import** com.google.android.exoplayer2.upstream.DefaultDataSourceFactory;  
**import** com.google.android.material.bottomsheet.BottomSheetDialog;  
  
**import** androidx.core.widget.NestedScrollView;  
**import** androidx.fragment.app.Fragment;  
**import** androidx.core.app.NotificationCompat;  
**import** androidx.core.content.FileProvider;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.cardview.widget.CardView;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
**import** android.text.Spannable;  
**import** android.text.SpannableString;  
**import** android.text.Spanned;  
**import** android.text.TextPaint;  
**import** android.text.TextUtils;  
**import** android.text.method.LinkMovementMethod;  
**import** android.text.style.ClickableSpan;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.FrameLayout;  
**import** android.widget.ImageButton;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.ProgressBar;  
**import** android.widget.RelativeLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
  
**import** com.bumptech.glide.Glide;  
**import** com.bumptech.glide.load.DataSource;  
**import** com.bumptech.glide.load.engine.DiskCacheStrategy;  
**import** com.bumptech.glide.load.engine.GlideException;  
**import** com.bumptech.glide.request.RequestListener;  
**import** com.bumptech.glide.request.RequestOptions;  
**import** com.bumptech.glide.request.target.Target;  
**import** com.firebase.ui.firestore.FirestoreRecyclerAdapter;  
**import** com.firebase.ui.firestore.FirestoreRecyclerOptions;  
**import** com.google.android.gms.tasks.OnCompleteListener;  
**import** com.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.common.collect.ArrayListMultimap;  
**import** com.google.common.collect.ListMultimap;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
*//import com.google.firebase.database.Query;***import** com.google.firebase.database.ValueEventListener;  
**import** com.google.firebase.firestore.DocumentReference;  
**import** com.google.firebase.firestore.DocumentSnapshot;  
**import** com.google.firebase.firestore.FieldValue;  
**import** com.google.firebase.firestore.FirebaseFirestore;  
**import** com.google.firebase.firestore.Query;  
**import** com.google.firebase.firestore.QueryDocumentSnapshot;  
**import** com.google.firebase.firestore.QuerySnapshot;  
**import** com.google.firebase.firestore.SetOptions;  
**import** com.google.firebase.firestore.WriteBatch;  
**import** com.google.firebase.storage.FileDownloadTask;  
**import** com.google.firebase.storage.FirebaseStorage;  
**import** com.google.firebase.storage.OnProgressListener;  
**import** com.google.firebase.storage.StorageMetadata;  
**import** com.google.firebase.storage.StorageReference;  
  
**import** java.io.File;  
**import** java.util.Calendar;  
**import** java.util.Date;  
**import** java.util.HashMap;  
**import** java.util.Locale;  
**import** java.util.Map;  
**import** java.util.Random;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**import** uk.co.senab.photoview.PhotoView;  
**import** uk.co.senab.photoview.PhotoViewAttacher;  
  
  
**import static** xyz.lifeissimple.hibuddy.Tab\_xpressions\_main.*imageOn*;  
**import static** xyz.lifeissimple.hibuddy.Tab\_xpressions\_main.*videoOn*;  
  
  
  
**public class** frag\_tab\_xpression\_main\_popular **extends** Fragment **implements** View.OnClickListener{  
  
 **public** RecyclerView **recyclerView**;  
 **private** LinearLayoutManager **layoutManager**;  
 LinearLayout **layouttohide**;  
 **private static** DatabaseReference *mdatabase*;  
 **private static** FirebaseAuth *mauth*;  
 *//FirebaseAuth.AuthStateListener authStateListener;* **private** String **uid**;  
 *//private ImageButton btn\_trend;* **private** SharedPreferences **sharedPreferences**;  
 **private static** String *lang*=**"hi"**;  
 **private** RelativeLayout **relativeLayout**;  
 **private** ProgressBar **pbar**,**vbar**;*//to show before data is loaded* **private** ImageView **img\_tn**;  
 **private** String **nxtThumb**,**nxtVideo**,**nxtCat**,**nxtThumb1**,**nxtVideo1**,**selectedThumb**,**nxtpostkey**;*//to be used when running next video  
 //final HashMap<String,String> nxtthumbnail = new HashMap<String, String>();* ListMultimap<String, String> **nxtthumbnail** = ArrayListMultimap.*create*();  
  
 *//TextView nxttxt;* **public** PhotoView **photoView**;*//external lib for image zoom* **private** TextView **txttrend1**,**txttrend2**,**txttrend3**;  
 **private** Random **ran** = **new** Random();  
 **private int r**=**ran**.nextInt(50000000)+1000000; *// will generatea random int from 1000000 to 50000000.* **private** FirebaseFirestore **firestore**;  
 **private** NestedScrollView **nestedScrollView**;  
 **private** FirestoreRecyclerAdapter<xpression\_main\_java, xpress\_holder> **firestore\_adapter**;  
 **private** Long **timetostart** = Long.*valueOf*(18);  
 **private** PlayerView **playerView**;*//view for external lib exo player for video play* **private** SimpleExoPlayer **simpleExoPlayer**;  
 **private** Long **timenow** = Calendar.*getInstance*().getTimeInMillis() - **r**;  
 FrameLayout **layout\_tohide\_tab**;*//to hide buttons etc when videos plays* LinearLayout **layout\_tohide\_tab1**;  
  
  
 *//below is trial  
 //String a[] = new String[]{"-LWb79ZMizE2seg0rkr1","-Lb8-1d8VA9WqxA-qS2W","-Lb3cMZW0P4iMrjyHRzE"};  
 // List list= Arrays.asList(a);* @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View rootview = inflater.inflate(R.layout.***xml\_frag\_tab\_xpressions\_popular***, container, **false**);  
 setHasOptionsMenu(**true**);*//so that the fragment will call the method.  
 //Log.i("etfy", String.valueOf(r));* **recyclerView** = (RecyclerView) rootview.findViewById(R.id.***recycle\_xpression***);  
 **recyclerView**.hasFixedSize();  
 **layoutManager** = **new** LinearLayoutManager(**this**.getActivity());  
 **layoutManager**.setOrientation(RecyclerView.***VERTICAL***);  
 **recyclerView**.setLayoutManager(**layoutManager**);  
 **recyclerView**.setNestedScrollingEnabled(**false**);*//for smooth srcoll with nested scrolling view. After adding trends* **nestedScrollView**=rootview.findViewById(R.id.***nestedscroll***);  
 **layouttohide**=rootview.findViewById(R.id.***layout1***);  
 *//playerView= rootview.findViewById(R.id.videoview);* **playerView**= getActivity().findViewById(R.id.***videoview***);  
 **relativeLayout**=getActivity().findViewById(R.id.***rel\_lay***);*//used when video plays* **img\_tn**=getActivity().findViewById(R.id.***img\_tn***);**img\_tn**.setVisibility(View.***INVISIBLE***);  
 *//img\_next1=rootview.findViewById(R.id.img\_next1);img\_next1.setVisibility(View.INVISIBLE);img\_next1.setOnClickListener(this);  
 //img\_next2=rootview.findViewById(R.id.img\_next2);img\_next2.setVisibility(View.INVISIBLE);img\_next2.setOnClickListener(this);  
 // nxttxt=rootview.findViewById(R.id.txt\_nxt);nxttxt.setVisibility(View.INVISIBLE);* **txttrend1**=rootview.findViewById(R.id.***txt\_trend1***);**txttrend1**.setOnClickListener(**this**);  
 **txttrend2**=rootview.findViewById(R.id.***txt\_trend2***);**txttrend2**.setOnClickListener(**this**);  
 **txttrend3**=rootview.findViewById(R.id.***txt\_trend3***);**txttrend3**.setOnClickListener(**this**);  
  
 **sharedPreferences** = getActivity().getSharedPreferences(**"basicinfo"**, Context.***MODE\_PRIVATE***);  
 *lang*=**sharedPreferences**.getString(**"language"**,**"hi"**);  
  
 **pbar**=rootview.findViewById(R.id.***pbar***);  
 **pbar**.setVisibility(View.***VISIBLE***);  
 **vbar**=getActivity().findViewById(R.id.***vbar***);**vbar**.setVisibility(View.***INVISIBLE***);  
 **photoView**=rootview.findViewById(R.id.***photo\_zoom***);  
 *mauth* = FirebaseAuth.*getInstance*();  
  
  
 *mdatabase* = FirebaseDatabase.*getInstance*().getReference();  
 *//bandhoo\_users=FirebaseDatabase.getInstance("https://bandhoo-users.firebaseio.com/").getReference().child("users");* **firestore**=FirebaseFirestore.*getInstance*();  
 *//Log.i("tuy", String.valueOf(timenow));* Query query=**firestore**.collection(**"xpression"**).whereEqualTo(**"langg"**,*lang*).whereEqualTo(**"visibleto"**,**"all"**).orderBy(**"time"**, Query.Direction.***DESCENDING***).startAt(**timenow**).limit(20);  
 *// Log.i("langg",lang);  
  
 // Query query=firestore.collection("xpression").whereIn("postkey",list).orderBy("time", Query.Direction.DESCENDING);* loadData(query);  
 **layout\_tohide\_tab**= getActivity().findViewById(R.id.***tablay***);  
 **layout\_tohide\_tab1**=getActivity().findViewById(R.id.***lays***);  
 **return** rootview;  
  
 }  
  
 **private void** loadData(Query query) {  
 FirestoreRecyclerOptions<xpression\_main\_java> options= **new** FirestoreRecyclerOptions.Builder<xpression\_main\_java>().setQuery(query,xpression\_main\_java.**class**).build();  
 **firestore\_adapter**= **new** FirestoreRecyclerAdapter<xpression\_main\_java, xpress\_holder>(options) {  
 @Override  
 **protected void** onBindViewHolder(@NonNull **final** xpress\_holder viewHolder, **int** position, @NonNull **final** xpression\_main\_java model) {  
 ImageButton btn\_play = (ImageButton) viewHolder.**mview**.findViewById(R.id.***btn\_play***);  
 **final** Button btn\_like = viewHolder.**mview**.findViewById(R.id.***btn\_xpress\_like***);  
  
 **if** (model.getThumbnail() !=**null**) {  
 **if** (model.getThumbnail().equals(**""**)) {  
 btn\_play.setVisibility(View.***INVISIBLE***);  
 } **else** {  
 btn\_play.setVisibility(View.***VISIBLE***);  
 }  
 }**else** { btn\_play.setVisibility(View.***INVISIBLE***);}  
  
 **if** (model.getPostkey() !=**null**) {  
 *mdatabase*.child(**"xpress\_likes"**).child(model.getPostkey()).child(**uid**).addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **if** (dataSnapshot.exists()) {  
 Drawable top = viewHolder.**mview**.getResources().getDrawable(R.drawable.***liked***);  
 btn\_like.setCompoundDrawablesWithIntrinsicBounds(**null**, top , **null**, **null**);  
 *// btn\_like.setImageResource(R.drawable.liked);* btn\_like.setEnabled(**false**);  
 } **else** {  
 Drawable top = viewHolder.**mview**.getResources().getDrawable(R.drawable.***like***);  
 btn\_like.setCompoundDrawablesWithIntrinsicBounds(**null**, top , **null**, **null**);  
 *// btn\_like.setImageResource(R.drawable.like);* btn\_like.setEnabled(**true**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }  
 **else** {}  
  
  
 viewHolder.setPostkey(model.getPostkey());  
 viewHolder.setCat(model.getCat());  
 viewHolder.setMsg(model.getMsg());  
 viewHolder.setTime(getActivity(), model.getTime());  
 viewHolder.setUid(model.getUid());  
 viewHolder.setBuddyid(model.getBuddyid());  
 viewHolder.setImage(getActivity(), model.getImage());  
 viewHolder.setThumbnail(getActivity(), model.getThumbnail());  
 viewHolder.setProf\_pic(getActivity(), model.getProf\_pic());  
 viewHolder.setLikes(model.getLikes());  
 viewHolder.setWacount(model.getWacount());  
 viewHolder.setVid\_size(model.getVid\_size());  
 viewHolder.setComments(model.getComments());  
 viewHolder.setViews(model.getViews());  
 viewHolder.setLangg(model.getLangg());  
 viewHolder.setVideo(model.getVideo());  
 viewHolder.setRepostno(model.getRepostno());  
  
 *//to zoom image* **final** ImageView imageView=viewHolder.**mview**.findViewById(R.id.***img\_xpress\_image***);  
 imageView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (!TextUtils.*isEmpty*(model.getImage())) {  
 *//zoomImage(model.getImage());  
 // introduced PhotoView (External lib) for zooming of image* **if** (getActivity() != **null**){  
 *imageOn*=**true**;  
 Log.*i*(**"ertfw"**, String.*valueOf*(**photoView**));  
 **photoView**.setVisibility(View.***VISIBLE***);  
 **layouttohide**.setVisibility(View.***GONE***);  
 **layout\_tohide\_tab**.setVisibility(View.***GONE***);  
 **layout\_tohide\_tab1**.setVisibility(View.***GONE***);  
  
 *//recyclerView.setVisibility(View.INVISIBLE);* Glide.*with*(getActivity()).load(model.getImage()).into(**photoView**);}  
  
 }  
  
 }  
 });  
  
 btn\_play.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **firestore**.collection(**"xpression"**).document(model.getPostkey()).update(**"views"**, FieldValue.*increment*(1));  
 initializeExoPlayer();  
 *videoOn*=**true**; *//trial for back pressed* **vbar**.setVisibility(View.***VISIBLE***);  
 *//layouttohide.setVisibility(View.GONE);* **layout\_tohide\_tab**.setVisibility(View.***GONE***);  
 **layout\_tohide\_tab1**.setVisibility(View.***GONE***);  
  
  
  
  
 **final** String vidfilename= **new** java.text.SimpleDateFormat(**"yyyyMMddHHmmss"**, Locale.***US***).format(**new** java.util.Date(model.getTime()));  
  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/videos/"**);  
 File localFile = **new** File(rootpath,**"vid\_"**+model.getBuddyid()+**"\_"**+vidfilename+**".mp4"**);  
  
 **if** (localFile.exists() && localFile.length()>0)  
 {  
 playSavedVideo(Uri.*fromFile*(localFile),model.getCat(),model.getThumbnail(),model.getPostkey());  
 **vbar**.setVisibility(View.***INVISIBLE***);  
  
 }  
 **else** {  
 **if** (model.getVideo() != **null** && !model.getVideo().equals(**""**) && model.getThumbnail() != **null** && !model.getThumbnail().equals(**""**))  
 {  
 *//videoView.setVideoURI(Uri.parse(model.getVideo()));* playVideo(model.getVideo(),model.getThumbnail(),model.getCat(),model.getPostkey());  
  
  
 }  
 }  
  
 }  
 });  
 *//to hide photoView* **photoView**.setOnPhotoTapListener(**new** PhotoViewAttacher.OnPhotoTapListener() {  
 @Override  
 **public void** onPhotoTap(View view, **float** x, **float** y) {  
 **photoView**.setVisibility(View.***GONE***);  
 **layouttohide**.setVisibility(View.***VISIBLE***);  
 **layout\_tohide\_tab**.setVisibility(View.***VISIBLE***);  
 **layout\_tohide\_tab1**.setVisibility(View.***VISIBLE***);  
 *imageOn*=**false**;  
  
  
 }  
 });  
  
 }  
  
 @NonNull  
 @Override  
 **public** xpress\_holder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***xpression\_main\_layout***, parent, **false**);  
 **return new** xpress\_holder(view);  
 }  
 };  
  
 **firestore\_adapter**.registerAdapterDataObserver(**new** RecyclerView.AdapterDataObserver() {  
 @Override  
 **public void** onItemRangeInserted(**int** positionStart, **int** itemCount) {  
 **super**.onItemRangeInserted(positionStart, itemCount);  
 **int** count = **firestore\_adapter**.getItemCount();  
 **int** last\_msg = **layoutManager**.findLastCompletelyVisibleItemPosition();  
 **if** (last\_msg == -1 ||  
 (positionStart >= (count - 1)) && (last\_msg == (positionStart - 1))) {  
 **pbar**.setVisibility(View.***GONE***);  
  
 }  
 **if** (**firestore\_adapter**.getItemCount() > 18){  
 **timetostart** = **firestore\_adapter**.getItem(18).getTime();}  
 }  
  
 }  
  
 );  
 **recyclerView**.setAdapter(**firestore\_adapter**);  
 **firestore\_adapter**.notifyDataSetChanged();  
  
  
  
 **nestedScrollView**.setOnScrollChangeListener(**new** NestedScrollView.OnScrollChangeListener(){  
  
 @Override  
 **public void** onScrollChange(NestedScrollView v, **int** scrollX, **int** scrollY, **int** oldScrollX, **int** oldScrollY) {  
 **if**(v.getChildAt(v.getChildCount() - 1) != **null**) {  
 **if** ((scrollY >= (v.getChildAt(v.getChildCount() - 1).getMeasuredHeight() - v.getMeasuredHeight())) &&  
 scrollY > oldScrollY) {  
  
 *//code to fetch more data for endless scrolling* Query nextquery=**firestore**.collection(**"xpression"**).whereEqualTo(**"langg"**,*lang*).orderBy(**"time"**, Query.Direction.***DESCENDING***).startAt(**timetostart**).limit(20);  
 loadData(nextquery);  
 **firestore\_adapter**.startListening();  
 }  
 }  
 }  
 });  
  
 }  
  
  
  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
 **firestore\_adapter**.startListening();  
  
 *//setKhaas();* **if** (*mauth*.getCurrentUser() != **null**) {  
 **uid** = *mauth*.getCurrentUser().getUid();  
 *//firestore\_adapter.startListening();* setKhaas();  
 }  
  
 *//initializeExoPlayer();* }  
  
  
 **private void** initializeExoPlayer() {  
 **if** (**simpleExoPlayer**==**null**)  
 {  
 **simpleExoPlayer**= **new** SimpleExoPlayer.Builder(**this**.getActivity()).build();  
 **playerView**.setPlayer(**simpleExoPlayer**);  
 *//playerView.setVisibility(View.INVISIBLE);  
 //Log.i("uytr","startpopular");* }  
 }  
  
 **private void** setKhaas() {  
 **txttrend1**.setVisibility(View.***VISIBLE***);  
 Calendar sCalendar = Calendar.*getInstance*();  
 String trend1 = sCalendar.getDisplayName(Calendar.***DAY\_OF\_WEEK***, Calendar.***LONG***, Locale.*getDefault*());  
 **int** dayOfWeek = sCalendar.get(Calendar.***DAY\_OF\_WEEK***);  
 **if** ( **"en"**.equals(*lang*)){ **txttrend1**.setText(**"#Happy "**+trend1);}  
 **else if** (**"hi"**.equals(*lang*)){  
  
 **if** (dayOfWeek==2){**txttrend1**.setText(**"#शुभ सोमवार"**);}  
 **if** (dayOfWeek==3){**txttrend1**.setText(**"#शुभ मंगलवार"**);}  
 **if** (dayOfWeek==4){**txttrend1**.setText(**"#शुभ बुधवार"**);}  
 **if** (dayOfWeek==5){**txttrend1**.setText(**"#शुभ गुरुवार"**);}  
 **if** (dayOfWeek==6){**txttrend1**.setText(**"#शुभ शुक्रवार"**);}  
 **if** (dayOfWeek==7){**txttrend1**.setText(**"#शुभ शनिवार"**);}  
 **if** (dayOfWeek==1){**txttrend1**.setText(**"#शुभ रविवार"**);}  
 }  
 **else** {**txttrend1**.setVisibility(View.***GONE***);}  
 *//Log.i("ertyy",lang+trend1+dayOfWeek);  
  
  
 mdatabase*.child(**"khaas"**).addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if**(dataSnapshot.child(**"trend2"**).child(*lang*).exists())  
 {  
 **if** (!TextUtils.*isEmpty*(dataSnapshot.child(**"trend2"**).child(*lang*).getValue().toString())) {  
 String trend2 = dataSnapshot.child(**"trend2"**).child(*lang*).getValue().toString();  
 **txttrend2**.setVisibility(View.***VISIBLE***);  
 *//txttrend.setVisibility(View.VISIBLE);* **txttrend2**.setText(**"#"**+trend2);  
 }  
 }  
 **if**(dataSnapshot.child(**"trend3"**).child(*lang*).exists())  
 {  
 **if** (!TextUtils.*isEmpty*(dataSnapshot.child(**"trend3"**).child(*lang*).getValue().toString())) {  
 String trend3 = dataSnapshot.child(**"trend3"**).child(*lang*).getValue().toString();  
 **txttrend3**.setVisibility(View.***VISIBLE***);  
 *//txttrend.setVisibility(View.VISIBLE);* **txttrend3**.setText(**"#"**+trend3);  
 }  
 }  
  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
  
 **private void** playVideo(String video, **final** String thumbnail, **final** String cat, String postkey) {  
 loadVideosAdvance(cat,thumbnail);  
 **img\_tn**.setVisibility(View.***VISIBLE***);  
 **vbar**.setVisibility(View.***VISIBLE***);  
 Glide.*with*(getContext()).load(thumbnail).into(**img\_tn**);  
 *//playerView.setVisibility(View.VISIBLE);  
  
 //for exoplayer* **final** DefaultDataSourceFactory dataSourceFactory = **new** DefaultDataSourceFactory(**this**.getContext(), **"bandhoo"**);  
 MediaSource videoSource = **new** ProgressiveMediaSource.Factory(dataSourceFactory).createMediaSource(Uri.*parse*(video));  
  
 **simpleExoPlayer**.prepare(videoSource);  
 **simpleExoPlayer**.addListener(**new** Player.EventListener() {  
  
 @Override  
 **public void** onPlayerStateChanged(**boolean** playWhenReady, **int** playbackState) {  
 **switch** (playbackState) {  
 **case** ExoPlayer.***STATE\_BUFFERING***:  
 **vbar**.setVisibility(View.***VISIBLE***);  
 **break**;  
 **case** ExoPlayer.***STATE\_ENDED***:  
 **if**(TextUtils.*isEmpty*(**nxtVideo**))  
 {  
 stopVideo();}  
 **else** {  
 MediaSource videoSource = **new** ProgressiveMediaSource.Factory(dataSourceFactory).createMediaSource(Uri.*parse*(**nxtVideo**));  
 **simpleExoPlayer**.prepare(videoSource);  
 **simpleExoPlayer**.setPlayWhenReady(**true**);  
 }  
 **break**;  
 **case** ExoPlayer.***STATE\_IDLE***:  
 *//img\_tn.setVisibility(View.VISIBLE);* **vbar**.setVisibility(View.***VISIBLE***);  
  
 **break**;  
 **case** ExoPlayer.***STATE\_READY***:  
 **relativeLayout**.setVisibility(View.***VISIBLE***);  
 *//playerView.setVisibility(View.VISIBLE);* **img\_tn**.setVisibility(View.***GONE***);  
 **vbar**.setVisibility(View.***GONE***);  
  
  
  
 **playerView**.setOnTouchListener(**new** OnSwipeTouchListener(getActivity()){  
 @Override  
 **public void** onSwipeUp() {  
 **super**.onSwipeUp();  
  
 **if**(TextUtils.*isEmpty*(**nxtVideo**))  
 {  
 stopVideo();}  
 **else** {  
 playVideo(**nxtVideo**,**nxtThumb**,**nxtCat**, **nxtpostkey**);  
 readyNextVideo(**nxtCat**);  
 }  
  
 }  
 @Override  
 **public void** onClick() {  
 **super**.onClick();  
 **playerView**.showController();  
 }  
 });  
  
 *//prepare next video after delay of 2 sec* **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 readyNextVideo(**nxtCat**);  
 }  
 },2000);  
 **break**;  
 **default**:  
 **break**;  
 }  
  
 }  
  
 @Override  
 **public void** onPlayerError(ExoPlaybackException error) {  
  
 }  
 @Override  
 **public void** onIsPlayingChanged(**boolean** isPlaying) {  
 **if** (isPlaying) {  
 **img\_tn**.setVisibility(View.***GONE***);  
 }  
 }  
  
 });  
  
 **vbar**.setVisibility(View.***GONE***);  
 **simpleExoPlayer**.setPlayWhenReady(**true**);  
  
 }  
  
 **private void** readyNextVideo(String cat) {  
 **if** (**nxtthumbnail**.size() > 0) {  
 **nxtthumbnail**.removeAll(**selectedThumb**);  
 **if** (**nxtthumbnail**.size() > 0) {  
 **nxtCat** = cat;  
 String firtvalue = **nxtthumbnail**.keySet().toArray()[**nxtthumbnail**.keySet().size() - 1].toString();  
 **nxtThumb** = firtvalue;  
 **selectedThumb** = firtvalue;*//will be deleted from hashmap next time* **nxtVideo** = **nxtthumbnail**.get(firtvalue).get(0);  
 **nxtpostkey**=**nxtthumbnail**.get(firtvalue).get(1);  
  
 }  
 }  
 }  
  
 **private void** loadVideosAdvance(**final** String cat, **final** String thumbnail) {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
  
 **if**(**nxtthumbnail**.size() < 3) {  
 **firestore**.collection(**"xpression"**).whereArrayContains(**"htags"**, cat.toLowerCase()).orderBy(**"video"**, Query.Direction.***DESCENDING***).limit(30).get().addOnCompleteListener(**new** OnCompleteListener<QuerySnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<QuerySnapshot> task) {  
 **if** (task.isSuccessful()) {  
 **for** (QueryDocumentSnapshot dataSnapshot1 : task.getResult()) {  
 **if** (dataSnapshot1.get(**"thumbnail"**) != **null**) {  
 **nxtthumbnail**.put(dataSnapshot1.get(**"thumbnail"**).toString(), dataSnapshot1.get(**"video"**).toString());  
 **nxtthumbnail**.put(dataSnapshot1.get(**"thumbnail"**).toString(), dataSnapshot1.get(**"postkey"**).toString());  
 *//Log.i("sdfr",dataSnapshot1.get("thumbnail").toString());* }  
 }  
 **if** (**nxtthumbnail**.size() > 0) {  
 **nxtthumbnail**.removeAll(**selectedThumb**);  
 **if** (**nxtthumbnail**.size() > 0) {  
 **nxtCat** = cat;  
 String firtvalue = **nxtthumbnail**.keySet().toArray()[**nxtthumbnail**.keySet().size() - 1].toString();  
 **nxtThumb** = firtvalue;  
 **selectedThumb** = firtvalue;*//will be deleted from hashmap next time* **nxtVideo** = **nxtthumbnail**.get(firtvalue).get(0);  
 **nxtpostkey**=**nxtthumbnail**.get(firtvalue).get(1);  
  
 }  
 }  
  
 }  
 }  
 });  
 }  
 }  
 }).start();  
 };  
  
  
  
 **private void** playSavedVideo(Uri video, **final** String cat, **final** String thumbnail, String postkey) {  
 loadVideosAdvance(cat,thumbnail);  
 **playerView**.setVisibility(View.***VISIBLE***);  
 **final** DefaultDataSourceFactory dataSourceFactory = **new** DefaultDataSourceFactory(**this**.getContext(), **"bandhoo"**);  
 MediaSource videoSource = **new** ProgressiveMediaSource.Factory(dataSourceFactory).createMediaSource(video);  
  
 **simpleExoPlayer**.prepare(videoSource);  
  
 **simpleExoPlayer**.addListener(**new** Player.EventListener() {  
 @Override  
 **public void** onPlayerStateChanged(**boolean** playWhenReady, **int** playbackState) {  
 **switch** (playbackState) {  
  
 **case** ExoPlayer.***STATE\_ENDED***:  
 **if**(TextUtils.*isEmpty*(**nxtVideo**))  
 {  
 stopVideo();}  
 **else** {  
 MediaSource videoSource = **new** ProgressiveMediaSource.Factory(dataSourceFactory).createMediaSource(Uri.*parse*(**nxtVideo**));  
 **simpleExoPlayer**.prepare(videoSource);  
 **simpleExoPlayer**.setPlayWhenReady(**true**);}  
 **break**;  
 **case** ExoPlayer.***STATE\_IDLE***:  
 **vbar**.setVisibility(View.***VISIBLE***);  
 **break**;  
 **case** ExoPlayer.***STATE\_READY***:  
 **vbar**.setVisibility(View.***GONE***);  
 **playerView**.setOnTouchListener(**new** OnSwipeTouchListener(getActivity()){  
 @Override  
 **public void** onSwipeUp() {  
 **super**.onSwipeUp();  
  
 **if**(TextUtils.*isEmpty*(**nxtVideo**))  
 {  
 stopVideo();}  
 **else** {  
 playVideo(**nxtVideo**,**nxtThumb**,**nxtCat**, **nxtpostkey**);  
 readyNextVideo(**nxtCat**);  
 }  
  
 }  
 @Override  
 **public void** onClick() {  
 **super**.onClick();  
 **playerView**.showController();  
 }  
 });  
  
 *//prepare next video after delay of 2 sec* **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 readyNextVideo(**nxtCat**);  
 }  
 },2000);  
 **break**;  
 **default**:  
 **break**;  
 }  
  
 }  
 });  
 **simpleExoPlayer**.setPlayWhenReady(**true**);  
  
 }  
  
  
  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 *//initializeExoPlayer();* }  
  
 @Override  
 **public void** onPause()  
 {  
 **super**.onPause();  
 **if** (**pbar** != **null**)  
 {**pbar**.setVisibility(View.***GONE***);}  
 releaseExoPlayer();  
  
 }  
 @Override  
 **public void** onStop() {  
 **super**.onStop();  
  
 **if** (**firestore\_adapter** != **null**) {  
 **firestore\_adapter**.stopListening();  
 }  
 releaseExoPlayer();  
 }  
  
 **private void** releaseExoPlayer() {  
 **if** (**simpleExoPlayer**!= **null**)  
 {  
 **simpleExoPlayer**.release();  
 **simpleExoPlayer**=**null**;  
 *//playerView.setVisibility(View.INVISIBLE);* }  
 }  
  
  
 **public void** stopVideo()  
 {  
 releaseExoPlayer();  
  
 **layout\_tohide\_tab**.setVisibility(View.***VISIBLE***);  
 **layout\_tohide\_tab1**.setVisibility(View.***VISIBLE***);  
 **layouttohide**.setVisibility(View.***VISIBLE***);  
 **vbar**.setVisibility(View.***INVISIBLE***);  
 **relativeLayout**.setVisibility(View.***INVISIBLE***);  
 **img\_tn**.setVisibility(View.***INVISIBLE***);  
  
 **nxtthumbnail**.clear();  
 **nxtThumb**=**null**;  
 **nxtVideo**=**null**;  
 **nxtVideo1**=**null**;  
 **nxtThumb1**=**null**;  
 **selectedThumb**=**null**;  
  
  
 }  
  
 *//this is to stop play of video when scrolling horizontally* **public void** setUserVisibleHint(**boolean** isVisibleToUser)  
 {  
 **super**.setUserVisibleHint(isVisibleToUser);  
 **if** (**this**.isVisible())  
 {  
 **if** (!isVisibleToUser) *// If we are becoming invisible, then...* {  
 stopVideo();  
 **if** (**pbar** != **null**)  
 {**pbar**.setVisibility(View.***GONE***);}  
 }  
  
 **if** (isVisibleToUser) *// If we are becoming visible, then...* {  
 *//play your video* }  
 }  
 }  
  
  
  
 @Override  
 **public void** onClick(View view) {  
  
 **if** (view==**txttrend1**)  
 { **if** (!TextUtils.*isEmpty*(**txttrend1**.getText()))  
 {  
 Intent intent= **new** Intent(getActivity(),Xpress\_category.**class**);  
 intent.putExtra(**"cat"**,**txttrend1**.getText().toString().replaceAll(**" "**,**""**).substring(1));  
 startActivity(intent);  
 }  
 }  
 **if** (view==**txttrend2**)  
 { **if** (!TextUtils.*isEmpty*(**txttrend2**.getText()))  
 {  
 Intent intent= **new** Intent(getActivity(),Xpress\_category.**class**);  
 intent.putExtra(**"cat"**,**txttrend2**.getText().toString().replaceAll(**" "**,**""**).substring(1));  
 startActivity(intent);  
 }  
 }  
 **if** (view==**txttrend3**)  
 { **if** (!TextUtils.*isEmpty*(**txttrend3**.getText()))  
 {  
 Intent intent= **new** Intent(getActivity(),Xpress\_category.**class**);  
 intent.putExtra(**"cat"**,**txttrend3**.getText().toString().replaceAll(**" "**,**""**).substring(1));  
 startActivity(intent);  
 }  
 }  
  
 }  
  
  
  
  
 public static class xpress\_holder extends RecyclerView.ViewHolder implements View.OnClickListener  
 {  
 View **mview**;  
 Button **btn\_like**,**btn\_comment**,**btn\_save**,**btn\_wa**, **btn\_repost**;  
 ImageButton **btn\_play**,**btn\_delete**;  
 TextView **txt\_msg**,**txt\_buddyid**,**txt\_time**,**txt\_like**,**txt\_comments**,**txt\_wa\_perc**,**txt\_views**,**txt\_cat**,**txt\_wa\_count**,**txt\_repost\_count**;  
 ImageView **imageView**,**img\_profile\_pic**,**img\_sent**;  
 *// VideoView videoView;* String **post\_key**,**imagefilename**,**vidfilename**;  
 CardView **card**;  
 String **uidd**,**buddyidd**,**profile\_pic**,**timed**,**buddy\_uid**,**category**=**""**;  
 **private** String **fwd\_post\_msg**,**fwd\_post\_image**,**fwd\_thumbnail**,**fwd\_video\_file**;*//to forword post to chat screen* **private** Long **vidoe\_size**,**timed1**;  
 **private** ProgressBar **vbar**;  
 **private** String **lang1**;  
 Button **btnfollow**;  
 DatabaseReference **newdatabase**;  
 FirebaseFirestore **firestore**= FirebaseFirestore.*getInstance*();  
  
  
 **public** xpress\_holder(View itemView) {  
 **super**(itemView);  
 **mview**=itemView;  
 **btn\_like**=**mview**.findViewById(R.id.***btn\_xpress\_like***);  
 *// btn\_chat=(ImageButton)mview.findViewById(R.id.btn\_xpress\_chat);* **btn\_play**=(ImageButton)**mview**.findViewById(R.id.***btn\_play***);  
 *//btn\_share=(ImageButton)mview.findViewById(R.id.btn\_share);* **btn\_comment**=**mview**.findViewById(R.id.***btn\_xpress\_re***);  
 **btn\_delete**=(ImageButton)**mview**.findViewById(R.id.***btn\_delete***);  
 **imageView**=(ImageView)**mview**.findViewById(R.id.***img\_xpress\_image***);  
 **btn\_save**=**mview**.findViewById(R.id.***btn\_xpress\_save***);  
 **btn\_wa**=**mview**.findViewById(R.id.***btn\_xpress\_wa***);  
 **txt\_wa\_perc**=**mview**.findViewById(R.id.***txt\_wa\_perc***);  
 **txt\_wa\_count**=**mview**.findViewById(R.id.***txt\_wa\_no***);  
 **txt\_views**=**mview**.findViewById(R.id.***views***);  
 **btnfollow**=**mview**.findViewById(R.id.***btnfollow***);**btnfollow**.setOnClickListener(**this**);  
 **newdatabase**=FirebaseDatabase.*getInstance*(**"https://in-hibuddy-a846d-dc806.firebaseio.com"**).getReference();  
 *//img\_sent=(ImageView)mview.findViewById(R.id.img\_xpress\_image);  
 //videoView=(VideoView)mview.findViewById(R.id.vid\_xpress\_video);* **txt\_msg**=(TextView)**mview**.findViewById(R.id.***txt\_xpress\_text***);  
 **txt\_repost\_count**=**mview**.findViewById(R.id.***txt\_repost\_no***);  
 **txt\_buddyid**=(TextView)**mview**.findViewById(R.id.***txt\_buddyid***);**txt\_buddyid**.setOnClickListener(**this**);  
 **txt\_cat**=**mview**.findViewById(R.id.***txt\_cat***);**txt\_cat**.setOnClickListener(**this**);  
 **txt\_time**=(TextView)**mview**.findViewById(R.id.***txt\_time***);  
 **txt\_like**=(TextView)**mview**.findViewById(R.id.***txt\_xpress\_like***);  
 **txt\_comments**=(TextView)**mview**.findViewById(R.id.***txt\_xpress\_comments***);  
 **img\_profile\_pic**=(ImageView)**mview**.findViewById(R.id.***img\_prof\_pic***);  
 **card** = (CardView)**mview**.findViewById(R.id.***card\_xpression\_main***);  
 **vbar**=**mview**.findViewById(R.id.***vbar***);**vbar**.setVisibility(View.***VISIBLE***);  
 **btn\_repost**=**mview**.findViewById(R.id.***btn\_xpress\_repost***); **btn\_repost**.setOnClickListener(**this**);  
 **btn\_like**.setOnClickListener(**this**);  
 *//btn\_chat.setOnClickListener(this);* **btn\_comment**.setOnClickListener(**this**);  
 **btn\_delete**.setOnClickListener(**this**);  
 *//btn\_share.setOnClickListener(this);  
 //img\_sent.setOnClickListener(this);* **imageView**.setOnClickListener(**this**);  
 **btn\_play**.setOnClickListener(**this**);  
 **btn\_save**.setOnClickListener(**this**);  
 **img\_profile\_pic**.setOnClickListener(**this**);  
 **btn\_wa**.setOnClickListener(**this**);  
 **uidd** = FirebaseAuth.*getInstance*().getCurrentUser().getUid();  
  
 }  
 **public void** setRepostno(Long repostno){  
 **try**{  
 **if**(repostno ==**null**){**txt\_repost\_count**.setText(String.*valueOf*(0));}  
 **else** {**txt\_repost\_count**.setText(String.*valueOf*(repostno));}  
 }**catch** (Exception ex){}  
 }  
 **public void** setUid(String uid)  
 {  
 **buddy\_uid**=uid;  
 *// Log.i("tyyuu",buddy\_uid);* **firestore**.collection(**"followers"**).document(**buddy\_uid**).collection(**"user\_followers"**).document(**uidd**).get().addOnCompleteListener(**new** OnCompleteListener<DocumentSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<DocumentSnapshot> task) {  
 **if** (task.isSuccessful())  
 {  
 **if** (task.getResult().exists()){**btnfollow**.setText(**"✔"**);**btnfollow**.setEnabled(**false**);}  
 **else** {**btnfollow**.setText(**mview**.getResources().getString(R.string.***follow***));**btnfollow**.setEnabled(**true**);}  
 }  
 }  
 });  
  
 }  
 **public void** setBuddyid(String buddyid)  
 {  
 **buddyidd**=buddyid;  
 **txt\_buddyid**.setText(**"@"**+buddyid);  
  
  
  
  
  
 }  
 **public void** setLangg(String langg)  
 {  
 **if** (langg != **null**)  
 {  
 **lang1**=langg;  
 }  
 **else** {**lang1**=*lang*;}  
 }  
 **public void** setCat(String cat)  
 {  
 **if** (!TextUtils.*isEmpty*(cat)) {  
 **category**=cat.toLowerCase();  
  
 **txt\_cat**.setText(**"#"** + cat);  
  
 }  
 **else** {**txt\_cat**.setText(**""**);}  
 }  
 **public void** setVideo(String video)  
 {**fwd\_video\_file**= video;}  
  
 **public void** setVid\_size(Long vid\_size)  
 {  
  
 **vidoe\_size**=vid\_size;  
 }  
 **public void** setPostkey(String postkey) {  
 **post\_key** = postkey;  
  
 }  
  
 **public void** setTime(Context ctx, Long time) {  
 *//TextView txt\_datetime;* **if** (time == **null**) {  
 *//System.out.print(time);* } **else** {  
 **imagefilename**= **new** java.text.SimpleDateFormat(**"yyyyMMddHHmmss"**, Locale.***US***).format(**new** java.util.Date(time));  
 **vidfilename**=**imagefilename**;  
 String date = **new** java.text.SimpleDateFormat(**"dd/MM/yyyy HH:mm:ss"**, Locale.***US***).format(**new** java.util.Date(time));  
  
  
 **txt\_time**.setText(timeInFormat(time));  
 **timed**= date;**timed1**=time;  
 }  
 }  
 **private** String timeInFormat(Long time) {  
 **long** diff = Math.*abs*(time - System.*currentTimeMillis*());  
 **long** ddiff;  
 **long** hdiff;  
 **long** mdiff;  
 ddiff= (**long**) Math.*floor*(diff/(1000\*60\*60\*24));  
 **long** diff1 = diff - (ddiff \* 1000 \* 60 \* 60 \* 24);  
  
 hdiff= (**long**) Math.*floor*(diff1/(1000\*60\*60));  
 **long** diff2 = diff1 - (hdiff\*1000\*60\*60);  
  
 mdiff= (**long**) Math.*floor*(diff2/(1000\*60));  
 **if**(ddiff >= 1)  
 { **return** ddiff+**"d "**+hdiff+**"h "**+mdiff+**"m"**;}  
 **else if** (hdiff >=1){ **return** hdiff+**"h "**+mdiff+**"m"**;}  
 **else** {**return** mdiff+**"m"**;}  
 }  
  
 **public void** setMsg(String msg)  
 {  
 **fwd\_post\_msg**=msg;  
 **if** (!TextUtils.*isEmpty*(msg))  
 {  
 String findhashtag = msg.trim();  
 Spannable spannable = **new** SpannableString(msg);  
 Pattern pattern = Pattern.*compile*(**"#\\w+"**);  
 Matcher matcher = pattern.matcher(findhashtag);  
  
 Pattern pattern1 = Pattern.*compile*(**"@\\w+"**);  
 Matcher matcher1=pattern1.matcher(findhashtag);  
 **while** (matcher.find())  
 {  
 ClickableSpan clickableSpan= **new** ClickableSpan() {  
 @Override  
 **public void** onClick(View widget) {  
 TextView tv =(TextView)widget;  
 Spanned s = (Spanned) tv.getText();  
 **int** start = s.getSpanStart(**this**);  
 **int** end = s.getSpanEnd(**this**);  
 Intent intent= **new** Intent(**mview**.getContext(),Xpress\_category.**class**);  
 intent.putExtra(**"cat"**,s.subSequence(start+1,end).toString().toLowerCase());  
 **mview**.getContext().startActivity(intent);  
 }  
 @Override  
 **public void** updateDrawState(TextPaint ds) {  
 **super**.updateDrawState(ds);  
 ds.setUnderlineText(**false**);  
 ds.setColor(**mview**.getResources().getColor(R.color.***blue\_light***));  
 }  
 };  
 **if** (matcher.end() - matcher.start() >1) {  
 spannable.setSpan(clickableSpan, matcher.start(), matcher.end(), Spanned.***SPAN\_EXCLUSIVE\_EXCLUSIVE***);  
 }  
 }  
  
 **while** (matcher1.find())  
 {  
 ClickableSpan clickableSpan= **new** ClickableSpan() {  
 @Override  
 **public void** onClick(View widget) {  
 TextView tv =(TextView)widget;  
 Spanned s = (Spanned) tv.getText();  
 **int** start = s.getSpanStart(**this**);  
 **int** end = s.getSpanEnd(**this**);  
 Intent intent= **new** Intent(**mview**.getContext(), Xpress\_buddy\_info.**class**);  
 intent.putExtra(**"buddyid"**,s.subSequence(start+1,end).toString().toLowerCase());  
 **mview**.getContext().startActivity(intent);  
 }  
 @Override  
 **public void** updateDrawState(TextPaint ds) {  
 **super**.updateDrawState(ds);  
 ds.setUnderlineText(**false**);  
 ds.setColor(**mview**.getResources().getColor(R.color.***blue\_light***));  
 }  
 };  
 **if** (matcher1.end() - matcher1.start() >1) {  
 spannable.setSpan(clickableSpan, matcher1.start(), matcher1.end(), Spanned.***SPAN\_EXCLUSIVE\_EXCLUSIVE***);  
 }  
 }  
  
 **txt\_msg**.setText(spannable);  
 **txt\_msg**.setMovementMethod(LinkMovementMethod.*getInstance*());  
 }  
 **else** {**txt\_msg**.setText(**""**);}  
 *//to increase value of views* **if** (**post\_key** != **null**) {  
 IncreaseViews(**post\_key**);  
 }  
 }  
 **public void** setLikes(Long likes) {  
 **try** {  
 **if** (likes == **null** )  
 {**txt\_like**.setText(String.*valueOf*(0));}  
 **else** {  
 **txt\_like**.setText(String.*valueOf*(likes));  
 }  
 }**catch** (Exception ex)  
 {  
  
 }  
 }  
 **public void** setWacount(Long wacount){  
 **try**{  
 **if**(wacount ==**null**){**txt\_wa\_count**.setText(String.*valueOf*(0));}  
 **else** {**txt\_wa\_count**.setText(String.*valueOf*(wacount));}  
 }**catch** (Exception ex){}  
 }  
 **public void** setComments(Long comments)  
 {  
 **try** {  
 **if** (comments == **null** ) {  
 **txt\_comments**.setText(String.*valueOf*(0));  
 } **else** {  
 **txt\_comments**.setText(String.*valueOf*(comments));  
 }  
 }**catch** (Exception ex)  
 {}  
 }  
 **public void** setViews(Long views) {  
 **try** {  
 **if** (views == **null** )  
 {**txt\_views**.setText(**mview**.getResources().getString(R.string.***views***)+**" "**+String.*valueOf*(0));}  
 **else** {  
 **txt\_views**.setText(**mview**.getResources().getString(R.string.***views***)+**" "**+String.*valueOf*(views));  
 }  
 }**catch** (Exception ex)  
 {  
  
 }  
 }  
  
 **public void** setImage(**final** Context ctx,**final** String image)  
 {  
**if** (image == **null** || image.equals(**""**))  
 {**imageView**.setImageResource(0);  
 **fwd\_post\_image**=**""**;  
 }  
 **else** {  
 **fwd\_post\_image** = image;  
  
 **imageView**.setVisibility(View.***VISIBLE***);  
  
 **if** (**mview**.getContext() != **null**)  
 {  
  
 Glide.*with*(ctx).load(image)  
 .apply(RequestOptions.*overrideOf*(450,600).centerInside().diskCacheStrategy(DiskCacheStrategy.***RESOURCE***)).thumbnail(0.01f).listener(**new** RequestListener<Drawable>() {  
 @Override  
 **public boolean** onLoadFailed(@Nullable GlideException e, Object model, Target<Drawable> target, **boolean** isFirstResource) {  
 **vbar**.setVisibility(View.***GONE***);  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onResourceReady(Drawable resource, Object model, Target<Drawable> target, DataSource dataSource, **boolean** isFirstResource) {  
 **vbar**.setVisibility(View.***GONE***);  
 **return false**;  
 }  
 }).into(**imageView**);  
 }  
 *//to increase value of views* **if** (**post\_key** != **null**) {  
 IncreaseViews(**post\_key**);  
 }  
  
 }  
 }  
  
  
  
 **public void** setProf\_pic(**final** Context ctx,**final** String prof\_pic)  
 {  
 **if** (prof\_pic == **null** || prof\_pic.equals(**""**))  
 {  
 *//img\_profile\_pic.setImageResource(R.mipmap.placeholder);* Glide.*with*(ctx).load(R.mipmap.***placeholder***).apply(RequestOptions.*circleCropTransform*()).into(**img\_profile\_pic**);  
 }  
 **else** {**profile\_pic**= prof\_pic;  
  
 **if** (**mview**.getContext() != **null**) {  
 Glide.*with*(ctx).load(prof\_pic)  
 .apply(RequestOptions.*overrideOf*(100, 100).fitCenter().circleCrop().diskCacheStrategy(DiskCacheStrategy.***RESOURCE***).placeholder(R.mipmap.***placeholder***))  
 .thumbnail(0.01f).into(**img\_profile\_pic**);  
 }}  
 }  
 **public void** setThumbnail(**final** Context ctx, **final** String thumbnail) {  
  
 **if** (thumbnail == **null** || thumbnail.equals(**""**)) {  
 **if** (**fwd\_post\_image**.equals(**""**))  
 {  
 **imageView**.setImageResource(0);  
 **fwd\_thumbnail**=**""**;  
 **vbar**.setVisibility(View.***GONE***);  
 }  
 **else** {  
 **fwd\_thumbnail**=**""**;  
 }  
 } **else** {  
 **fwd\_thumbnail**= thumbnail;  
  
 **if** (**mview**.getContext() != **null**)  
 {  
  
 Glide.*with*(ctx).load(thumbnail)  
 .apply(RequestOptions.*overrideOf*(450,225).centerCrop().diskCacheStrategy(DiskCacheStrategy.***RESOURCE***)).thumbnail(0.01f).listener(**new** RequestListener<Drawable>() {  
 @Override  
 **public boolean** onLoadFailed(@Nullable GlideException e, Object model, Target<Drawable> target, **boolean** isFirstResource) {  
 **vbar**.setVisibility(View.***GONE***);  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onResourceReady(Drawable resource, Object model, Target<Drawable> target, DataSource dataSource, **boolean** isFirstResource) {  
 **vbar**.setVisibility(View.***GONE***);  
 **return false**;  
 }  
 }).into(**imageView**);  
 }}  
 }  
 *//to save image into local storage* **private void** saveImage(**final** String fwd\_post\_image, **final** String buddyidd, **final** String imagefilename) {  
 **if** (fwd\_post\_image != **null** && !fwd\_post\_image.isEmpty()) {  
  
 *//Picasso.with(mview.getContext()).load(fwd\_post\_image).into(getTarget(fwd\_post\_image, buddyidd, timed));* **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 StorageReference filerefer;  
 FirebaseStorage filestore;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReferenceFromUrl(fwd\_post\_image);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReferenceFromUrl(fwd\_post\_image);  
  
 }  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(), **"/Hibuddy/Images/xpressions/"**);  
 **if** (!rootpath.exists()) {  
 rootpath.mkdirs();  
 }  
 **final** File localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 filerefer.getFile(localFile);  
 }  
 }).start();  
  
 }  
 }  
  
  
  
 @Override  
 **public void** onClick(View view) {  
 **if** (view==**btn\_like**)  
 {  
 Drawable top = **mview**.getResources().getDrawable(R.drawable.***liked***);  
 **btn\_like**.setCompoundDrawablesWithIntrinsicBounds(**null**, top , **null**, **null**);  
 **btn\_like**.setEnabled(**false**);  
 **if** (**lang1** != **null**) {  
 *mdatabase*.child(**"xpress\_likes"**).child(**post\_key**).child(**uidd**).setValue(**uidd**);  
 **firestore**.collection(**"xpression"**).document(**post\_key**).update(**"likes"**, FieldValue.*increment*(1));  
  
  
 }  
 }  
  
 **if** (view == **btn\_comment**)  
 {  
 **if** (**fwd\_thumbnail** == **null** || **fwd\_thumbnail**.equals(**""**))  
 { *//Log.i("tttt",fwd\_post\_image);* saveImage(**fwd\_post\_image**, **buddyidd**, **imagefilename**);*//to save image local disc* Intent intent= **new** Intent(**mview**.getContext(),Xpress\_comment.**class**);  
 intent.putExtra(**"buddyid"**,**buddyidd**);  
 intent.putExtra(**"profile\_pic"**,**profile\_pic**);  
 intent.putExtra(**"time"**,**timed**);  
 intent.putExtra(**"msg"**,**fwd\_post\_msg**);  
 intent.putExtra(**"image"**,**fwd\_post\_image**);  
 intent.putExtra(**"thumbnail"**,**""**);  
 intent.putExtra(**"key"**,**post\_key**);  
 intent.putExtra(**"cat"**,**category**);  
 view.getContext().startActivity(intent);  
  
 }  
 **else if** (**fwd\_thumbnail** != **null** && !**fwd\_thumbnail**.equals(**""**))  
 { *//Log.i("ttt",fwd\_thumbnail);* Intent intent= **new** Intent(**mview**.getContext(),Xpress\_comment.**class**);  
 intent.putExtra(**"buddyid"**,**buddyidd**);  
 intent.putExtra(**"profile\_pic"**,**profile\_pic**);  
 intent.putExtra(**"time"**,**timed**);  
 intent.putExtra(**"msg"**,**fwd\_post\_msg**);  
 intent.putExtra(**"image"**,**""**);  
 intent.putExtra(**"thumbnail"**,**fwd\_thumbnail**);  
 intent.putExtra(**"key"**,**post\_key**);  
 intent.putExtra(**"cat"**,**category**);  
 view.getContext().startActivity(intent);  
 }  
  
  
 }  
 **if** (view==**imageView**)  
 {  
 **if** (**fwd\_post\_image** != **null** && !**fwd\_post\_image**.isEmpty()) {  
 saveImage(**fwd\_post\_image**, **buddyidd**, **imagefilename**);*//to save image local disc* }  
 }  
 **if** (view==**btn\_play**)  
 {  
 *//to increase value of views* **if** (**post\_key** != **null**) {  
 IncreaseViews(**post\_key**);  
 }  
  
 }  
 **if** (view==**btn\_delete**)  
 {  
 *//introduced where dialog will come from the bottom* **final** BottomSheetDialog mBottomSheetDialog = **new** BottomSheetDialog(**mview**.getContext());  
 LayoutInflater layoutInflater=LayoutInflater.*from*(**mview**.getContext());  
 **final** View sheetView = layoutInflater.inflate(R.layout.***bottom\_sheet\_dialog***, **null**);  
 mBottomSheetDialog.setContentView(sheetView);  
 LinearLayout li\_share = sheetView.findViewById(R.id.***lay\_share***);  
 LinearLayout li\_share\_bandhoo = sheetView.findViewById(R.id.***lay\_share\_bandhoo***);  
 LinearLayout li\_delete = sheetView.findViewById(R.id.***lay\_delete***);  
 LinearLayout li\_treasure = sheetView.findViewById(R.id.***lay\_treasure***);  
 LinearLayout li\_report = sheetView.findViewById(R.id.***lay\_report***);  
  
 li\_share.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//Toast.makeText(mview.getContext(), "hi", Toast.LENGTH\_SHORT).show();* **if**(mBottomSheetDialog.isShowing())  
 {  
 mBottomSheetDialog.dismiss();}  
 Intent sendIntent = **new** Intent();  
 sendIntent.setAction(Intent.***ACTION\_SEND***);  
 sendIntent.putExtra(Intent.***EXTRA\_TEXT***, **fwd\_post\_msg**+**'\n'**+**"https://bandhoo.in/onepost/"**+**post\_key**);  
 sendIntent.setType(**"text/plain"**);  
 **mview**.getContext().startActivity(Intent.*createChooser*(sendIntent, **mview**.getContext().getResources().getString(R.string.***share***)));  
  
 }  
 });  
 li\_share\_bandhoo.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!TextUtils.*isEmpty*(**post\_key**))  
 {  
  
 **if**(mBottomSheetDialog.isShowing())  
 {  
 mBottomSheetDialog.dismiss();}  
 Intent intent = **new** Intent(**mview**.getContext(), fwd\_post\_screen.**class**);  
 intent.setAction(Intent.***ACTION\_MAIN***);*//needed in fwd.post.screen to decide from where intent has come.* intent.putExtra(**"from\_xpression"**,**"true"**);  
 intent.putExtra(**"key\_post"**,**post\_key**);  
 **mview**.getContext().startActivity(intent);  
  
 }  
  
 }  
 });  
 li\_delete.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(mBottomSheetDialog.isShowing())  
 {  
 mBottomSheetDialog.dismiss();}  
 **if** (!TextUtils.*isEmpty*(**buddy\_uid**) && **uidd**.equals(**buddy\_uid**))  
 {  
 showdialog();}  
 **else** {  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***cannotdelete***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
 });  
 li\_report.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(mBottomSheetDialog.isShowing())  
 {  
 mBottomSheetDialog.dismiss();}  
 showdialogReport(**post\_key**);  
 }  
 });  
 li\_treasure.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(mBottomSheetDialog.isShowing())  
 {  
 mBottomSheetDialog.dismiss();}  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 DatabaseReference xdata = *mdatabase*.child(**"xtreasure"**).child(**uidd**);  
 Map newmsg = **new** HashMap();  
 newmsg.put(**"msg"**,**fwd\_post\_msg**);  
 newmsg.put(**"image"**, **fwd\_post\_image**);  
 newmsg.put(**"video"**, **fwd\_video\_file**);  
 newmsg.put(**"buddyid"**, **buddyidd**);  
 newmsg.put(**"time"**, **timed1**);  
 newmsg.put(**"postkey"**, **post\_key**);  
 newmsg.put(**"thumbnail"**, **fwd\_thumbnail**);  
 newmsg.put(**"vid\_size"**, **vidoe\_size**);  
 newmsg.put(**"cat"**,**category**);  
 xdata.child(**post\_key**).updateChildren(newmsg, **new** DatabaseReference.CompletionListener() {  
 @Override  
 **public void** onComplete(@Nullable DatabaseError databaseError, @NonNull DatabaseReference databaseReference) {  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***savedtotreasure***), Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 }  
 }).start();  
  
 }  
 });  
 mBottomSheetDialog.show();  
  
  
  
 }  
 **if** (view == **img\_profile\_pic**)  
 {  
 **if** (!TextUtils.*isEmpty*(**buddy\_uid**)) {  
 Intent intent = **new** Intent(**mview**.getContext(), Xpress\_buddy\_info.**class**);  
 intent.putExtra(**"buddy\_uid"**, **buddy\_uid**);  
 intent.putExtra(**"buddyid"**,**buddyidd**);  
 **mview**.getContext().startActivity(intent);  
 }  
  
 }  
 **if** (view == **txt\_buddyid**)  
 {  
 **if** (!TextUtils.*isEmpty*(**buddy\_uid**)) {  
 Intent intent = **new** Intent(**mview**.getContext(), Xpress\_buddy\_info.**class**);  
 intent.putExtra(**"buddy\_uid"**, **buddy\_uid**);  
 intent.putExtra(**"buddyid"**,**buddyidd**);  
 **mview**.getContext().startActivity(intent);  
 }  
  
 }  
 **if** (view == **txt\_cat**)  
 {  
 **if** (!TextUtils.*isEmpty*(**category**))  
 {  
 Intent intent= **new** Intent(**mview**.getContext(),Xpress\_category.**class**);  
 intent.putExtra(**"cat"**,**category**);  
 **mview**.getContext().startActivity(intent);  
  
 }  
  
 }  
 **if** (view==**btn\_save**)  
 {  
 **if** (**fwd\_video\_file** != **null** && **fwd\_thumbnail** != **null** && !**fwd\_video\_file**.isEmpty() && !**fwd\_thumbnail**.isEmpty())  
 {  
 Random rand = **new** Random();  
 **final int** r = rand.nextInt(10000);*//r will be random int between 0 and 4* **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 downloadVideo(**fwd\_video\_file**,**buddyidd**,**vidfilename**,r);  
 *//r will be different for every click and so notification will be diiferent for every click  
 //it will be needed if user clicks multiple downloads .. then they will not be overridden* }  
 }).start();  
  
 }  
 **if** (**fwd\_post\_image** != **null** && !**fwd\_post\_image**.isEmpty()) {  
  
 Random rand = **new** Random();  
 **final int** r = rand.nextInt(10000);*//r will be random int between 0 and 4* downloadImage(**fwd\_post\_image**,**buddyidd**,**imagefilename**,r);  
 }  
 }  
 **if** (view == **btn\_wa**)  
 {  
 *//to increase no of wa share* **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 increaseCount(**post\_key**);  
 }  
 }).start();  
 **if** (**fwd\_video\_file** != **null** && **fwd\_thumbnail** != **null** && !**fwd\_video\_file**.isEmpty() && !**fwd\_thumbnail**.isEmpty())  
 {  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/videos/"**);  
 File localFile = **new** File(rootpath,**"vid\_"**+**buddyidd**+**"\_"**+**vidfilename**+**".mp4"**);  
 **if** (localFile.length()>0)  
 {shareVideotoWA(localFile);}  
 **else** {  
 **if** (!isConnectedtoInternet(**mview**.getContext()))*//to check internet connection* {  
  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***internet***), Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***ftxmp***),Toast.***LENGTH\_LONG***).show();  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 getVideoforUri(**fwd\_video\_file**,**buddyidd**,**vidfilename**);  
  
 }  
 }).start();  
  
 }  
 }  
 **else if** (**fwd\_post\_image** != **null** && !**fwd\_post\_image**.isEmpty())  
 {  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/Images/xpressions/"**);  
 File localFile = **new** File(rootpath, **buddyidd**+**"\_"**+**imagefilename** + **".jpeg"**);  
 **if** (localFile.length() >0)  
 {shareImagetoWA(localFile);  
 *//Log.i("ssd","sorr");* }  
 **else** {  
 **if** (!isConnectedtoInternet(**mview**.getContext()))*//to check internet connection* {  
  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***internet***), Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***ftxmp1***),Toast.***LENGTH\_SHORT***).show();  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 getImageforUri(**fwd\_post\_image**,**buddyidd**,**imagefilename**);  
 }  
 }).start();  
  
  
 }  
 }  
 **else if** (**fwd\_post\_msg**!=**null** && !**fwd\_post\_msg**.isEmpty())  
 {  
 Intent intent= **new** Intent();  
 intent.setAction(Intent.***ACTION\_SEND***);  
 *// intent.putExtra(Intent.EXTRA\_TEXT,fwd\_post\_msg);  
 // intent.setType("text/plain");* intent.putExtra(Intent.***EXTRA\_TEXT***,**fwd\_post\_msg**+**"\n"**+**"Bandhoo"**+**"\n"**+**mview**.getResources().getString(R.string.***invite\_friends\_message***)+**"\n"**+ **"https://play.google.com/store/apps/details?id=xyz.lifeissimple.hibuddy"**);  
 intent.setType(**"text/plain"**);  
 intent.setPackage(**"com.whatsapp"**);  
 **try** {  
 **mview**.getContext().startActivity(intent);  
 }**catch** (android.content.ActivityNotFoundException ex) {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***ftxmp2***),Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 **else** {}  
  
  
 }  
 **if** (view ==**btnfollow**){  
 **if** (!isConnectedtoInternet(**mview**.getContext()))*//to check internet connection* { Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***internet***), Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **else** {  
 **btnfollow**.setText(**"✔"**);  
 **btnfollow**.setEnabled(**false**);  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *mdatabase*.child(**"xfollowers"**).child(**buddyidd**).child(**uidd**).setValue(**uidd**);  
 *//DatabaseReference mdata = mdatabase.child("xfollow").child(uid).child(buddy\_uid);* **final long** time = Calendar.*getInstance*().getTimeInMillis();  
 **final** Map map = **new** HashMap();  
 map.put(**"buddyid"**, **buddyidd**);  
 map.put(**"pro\_pic"**,**profile\_pic** );  
 map.put(**"time"**,time);  
 map.put(**"uid"**,**buddy\_uid**);  
  
 WriteBatch writeBatch= **firestore**.batch();  
 DocumentReference d1 =**firestore**.collection(**"followings"**).document(**uidd**).collection(**"user\_followings"**).document(**buddy\_uid**);  
 writeBatch.set(d1,map);  
 DocumentReference d2 =**firestore**.collection(**"users"**).document(**buddy\_uid**);  
 writeBatch.update(d2,**"followers\_count"**, FieldValue.*increment*(1));  
 DocumentReference d3=**firestore**.collection(**"followers"**).document(**buddy\_uid**).collection(**"user\_followers"**).document(**uidd**);  
 *//follow karne wale ka data for future use* Map map1= **new** HashMap();  
 map1.put(**"uid"**,**uidd**);  
 **if** (*mauth*.getCurrentUser().getPhotoUrl() != **null**){map1.put(**"pro\_pic"**,*mauth*.getCurrentUser().getPhotoUrl().toString());}  
 **else** { map1.put(**"pro\_pic"**,**""**);}  
 map1.put(**"name"**,*mauth*.getCurrentUser().getDisplayName());  
 map1.put(**"time"**,time);  
 writeBatch.set(d3,map1);  
 writeBatch.commit().addOnCompleteListener(**new** OnCompleteListener<Void>() {  
 @Override  
 **public void** onComplete(@NonNull Task<Void> task) {  
 *//add post into wall when some one start follow someone* **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 addexistingPost(**buddy\_uid**,**uidd**);  
 }  
 }).start();  
 }  
 });  
  
  
  
 }  
 }).start();  
 }  
 }  
 **if** (view==**btn\_repost**)  
 {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 increaseRepostCount(**post\_key**);  
 }  
 }).start();  
 Intent intent= **new** Intent(**mview**.getContext(),Repost.**class**);  
 intent.putExtra(**"cat"**,**category**);  
 intent.putExtra(**"msg"**,**fwd\_post\_msg**);  
 intent.putExtra(**"image"**, **fwd\_post\_image**);  
 intent.putExtra(**"video"**, **fwd\_video\_file**);  
 intent.putExtra(**"uid"**, **uidd**);  
 intent.putExtra(**"thumbnail"**, **fwd\_thumbnail**);  
 intent.putExtra(**"vid\_size"**, **vidoe\_size**);  
 intent.putExtra(**"cat"**,**category**);  
 intent.putExtra(**"sender\_id"**,**buddyidd**);  
  
  
 **mview**.getContext().startActivity(intent);  
  
 }  
 }  
  
 **private void** addexistingPost(String buddy\_uid, **final** String uid) {  
 **firestore**.collection(**"xpression"**).whereEqualTo(**"uid"**,buddy\_uid).orderBy(**"time"**, Query.Direction.***DESCENDING***).limit(5).get()  
 .addOnCompleteListener(**new** OnCompleteListener<QuerySnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<QuerySnapshot> task) {  
 **if** (task.isSuccessful())  
 {  
 **for** (QueryDocumentSnapshot documentSnapshot : task.getResult())  
 {  
 *// Log.i("kkkk", (String) documentSnapshot.get("postkey"));* Map map= **new** HashMap();  
 map.put(**"postkey"**,FieldValue.*arrayUnion*((String)documentSnapshot.get(**"postkey"**)));  
 **firestore**.collection(**"users\_wall"**).document(uid).set(map,SetOptions.*merge*());  
 }  
 }  
 }  
 });  
 }  
  
 **private void** increaseRepostCount(String postkey) {  
 **if** (**lang1** != **null**) {  
 **firestore**.collection(**"xpression"**).document(postkey).update(**"repostno"**, FieldValue.*increment*(1));  
 }  
 }  
 **private void** increaseCount(**final** String postkey) {  
*// if (category != null) {  
// wacountToCategory(postkey, category);//like for category and hashtags  
// }* **if** (**lang1** != **null**) {  
 **firestore**.collection(**"xpression"**).document(postkey).update(**"wacount"**, FieldValue.*increment*(1));  
  
 }  
 }  
  
 **private boolean** isConnectedtoInternet(Context context) {  
 ConnectivityManager connectivity = (ConnectivityManager) context.getSystemService(Context.***CONNECTIVITY\_SERVICE***);  
 **if** (connectivity != **null**) {  
 NetworkInfo info = connectivity.getActiveNetworkInfo();  
 **if** (info != **null**) {  
 **if** (info.getType() == ConnectivityManager.***TYPE\_WIFI***) {  
 **return true**;  
 } **else if** (info.getType() == ConnectivityManager.***TYPE\_MOBILE***) {  
 **return true**;  
 }  
 }  
 }  
 **return false**;  
 }  
  
  
  
 **private void** IncreaseViews(**final** String post\_key) {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **final** String date = **new** java.text.SimpleDateFormat(**"ddMMyyyy"**, Locale.*getDefault*()).format(**new** Date());  
  
 *mdatabase*.child(**"xpress\_views"**).child(post\_key).child(**uidd**).addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **if** (dataSnapshot.exists())  
 {  
 **if** (dataSnapshot.child(date).exists())  
 {}  
 **else** {  
 **if** (**lang1** != **null**) {  
 **firestore**.collection(**"xpression"**).document(post\_key).update(**"views"**, FieldValue.*increment*(1));  
 *mdatabase*.child(**"xpress\_views"**).child(post\_key).child(**uidd**).child(date).setValue(**true**);*//only one view per day can be increased* }  
  
 }  
  
 }  
 **else** {  
 **firestore**.collection(**"xpression"**).document(post\_key).update(**"views"**, FieldValue.*increment*(1));  
 *mdatabase*.child(**"xpress\_views"**).child(post\_key).child(**uidd**).child(date).setValue(**true**);*//only one view per day can be increased* }  
 }  
  
 @Override  
 **public void** onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
  
 }  
 }).start();  
 }  
  
  
 **private void** getVideoforUri(**final** String fwd\_video\_file, **final** String buddyidd, **final** String vidfilename) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }  
 filerefer.getDownloadUrl().addOnSuccessListener(**new** OnSuccessListener<Uri>() {  
 @Override  
 **public void** onSuccess(Uri uri) {  
 newVideo(buddyidd,vidfilename);*//for changed video after logo* }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 oldVideo(fwd\_video\_file,buddyidd,vidfilename);*//earlier one* }  
 });  
  
  
 }  
  
 **private void** newVideo(String buddyidd, String vidfilename) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }  
  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(), **"/Hibuddy/videos/"**);  
 **if** (!rootpath.exists()) {  
 rootpath.mkdirs();  
 }  
  
 **final** File localFile = **new** File(rootpath, **"vid\_"** + buddyidd + **"\_"** + vidfilename + **".mp4"**);  
 filerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Handler handler = **new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***add\_gp2***), Toast.***LENGTH\_SHORT***).show();  
  
  
 }  
 });  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 shareVideotoWA(localFile);  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler = **new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
  
  
 }  
 });  
 }  
 **private void** oldVideo(String fwd\_video\_file, String buddyidd, String vidfilename) {  
 StorageReference filerefer;  
 FirebaseStorage filestore;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReferenceFromUrl(fwd\_video\_file);  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReferenceFromUrl(fwd\_video\_file);  
 *//Log.i("rtf",filerefer.toString());* }  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(), **"/Hibuddy/videos/"**);  
 **if** (!rootpath.exists()) {  
 rootpath.mkdirs();  
 }  
  
 **final** File localFile = **new** File(rootpath, **"vid\_"** + buddyidd + **"\_"** + vidfilename + **".mp4"**);  
 filerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Handler handler = **new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(), **mview**.getResources().getString(R.string.***add\_gp2***), Toast.***LENGTH\_SHORT***).show();  
  
  
 }  
 });  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 shareVideotoWA(localFile);  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler = **new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
  
  
 }  
 });  
 }  
  
 **private void** shareVideotoWA(File localFile) {  
 Uri uri;  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***N***)  
 { uri = FileProvider.*getUriForFile*(**mview**.getContext(),**"xyz.lifeissimple.hibuddy.fileProvider"**,localFile);}  
 **else** {uri=Uri.*fromFile*(localFile);}  
 *//Log.i("ssdp",uri.toString());* Intent intent= **new** Intent();  
 intent.setAction(Intent.***ACTION\_SEND***);  
 intent.putExtra(Intent.***EXTRA\_STREAM***,uri);  
 intent.putExtra(Intent.***EXTRA\_TEXT***,**fwd\_post\_msg**+**"\n"**+**"Bandhoo"**+**"\n"**+**mview**.getResources().getString(R.string.***invite\_friends\_message***)+**"\n"**+ **"https://play.google.com/store/apps/details?id=xyz.lifeissimple.hibuddy"**);  
 intent.setType(**"text/plain"**);  
 intent.setType(**"video/mp4"**);  
 intent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  
 intent.setPackage(**"com.whatsapp"**);  
 **try** {  
 **mview**.getContext().startActivity(intent);  
 }**catch** (android.content.ActivityNotFoundException ex) {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***ftxmp2***),Toast.***LENGTH\_SHORT***).show();}  
  
 }  
 **private void** getImageforUri(String fwd\_post\_image, **final** String buddyidd, **final** String imagefilename) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 }  
 filerefer.getDownloadUrl().addOnSuccessListener(**new** OnSuccessListener<Uri>() {  
 @Override  
 **public void** onSuccess(Uri uri) {  
 newImage(buddyidd,imagefilename);*//for changed image after logo* }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 oldImage(buddyidd,imagefilename);*//earlier one* }  
 });  
  
  
  
 }  
 **private void** newImage(**final** String buddyidd, **final** String imagefilename) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 }  
  
 **final** File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/Images/xpressions/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
 *//below to know the type of file ... if it is gif or jpeg* **final** StorageReference finalFilerefer = filerefer;  
 **final** StorageReference finalFilerefer1 = filerefer;  
 filerefer.getMetadata().addOnSuccessListener(**new** OnSuccessListener<StorageMetadata>() {  
 @Override  
 **public void** onSuccess(StorageMetadata storageMetadata) {  
 String extension = storageMetadata.getContentType();  
 *//Log.i("dfg",extension);* **final** File localFile;  
 **if** (extension.equals(**"image/gif"**))  
 {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".gif"**);  
 }  
 **else** {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 }  
  
 finalFilerefer1.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***add\_gp2***),Toast.***LENGTH\_SHORT***).show();  
  
  
 }  
 });  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 **if** (localFile.length()>0) {  
 shareImagetoWA(localFile);  
 }  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
 }  
 });  
  
  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **final** File localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 finalFilerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***add\_gp2***),Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 **if** (localFile.length()>0) {  
 shareImagetoWA(localFile);  
 }  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
 }  
 });  
  
 }  
 });  
  
 }  
 **private void** oldImage(**final** String buddyidd, **final** String imagefilename) {  
 StorageReference filerefer;  
 FirebaseStorage filestore;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReferenceFromUrl(**fwd\_post\_image**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReferenceFromUrl(**fwd\_post\_image**);  
  
 }  
 **final** File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/Images/xpressions/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
 *//below to know the type of file ... if it is gif or jpeg* **final** StorageReference finalFilerefer = filerefer;  
 filerefer.getMetadata().addOnSuccessListener(**new** OnSuccessListener<StorageMetadata>() {  
 @Override  
 **public void** onSuccess(StorageMetadata storageMetadata) {  
 String extension = storageMetadata.getContentType();  
 *//Log.i("dfg",extension);* **final** File localFile;  
 **if** (extension.equals(**"image/gif"**))  
 {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".gif"**);  
 }  
 **else** {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 }  
  
 finalFilerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***add\_gp2***),Toast.***LENGTH\_SHORT***).show();  
  
  
 }  
 });  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 **if** (localFile.length()>0) {  
 shareImagetoWA(localFile);  
 }  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
 }  
 });  
  
  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **final** File localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 finalFilerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***add\_gp2***),Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
  
  
 }  
 }).addOnCompleteListener(**new** OnCompleteListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onComplete(@NonNull Task<FileDownloadTask.TaskSnapshot> task) {  
 **if** (localFile.length()>0) {  
 shareImagetoWA(localFile);  
 }  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **final double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 Handler handler =**new** Handler();  
 handler.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 **txt\_wa\_perc**.setText(String.*valueOf*((**int**) progress) + **"%"**);  
  
 }  
 });  
 }  
 });  
  
 }  
 });  
  
 }  
  
 **private void** shareImagetoWA(File localFile) {  
 Uri uri;  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***N***)  
 { uri = FileProvider.*getUriForFile*(**mview**.getContext(),**"xyz.lifeissimple.hibuddy.fileProvider"**,localFile);}  
 **else** {uri=Uri.*fromFile*(localFile);}  
 *//Log.i("ssdp",uri.toString());* Intent intent= **new** Intent();  
 intent.setAction(Intent.***ACTION\_SEND***);  
 intent.putExtra(Intent.***EXTRA\_STREAM***,uri);  
 intent.putExtra(Intent.***EXTRA\_TEXT***,**fwd\_post\_msg**+**"\n"**+**"Bandhoo"**+**"\n"**+**mview**.getResources().getString(R.string.***invite\_friends\_message***)+**"\n"**+ **"https://play.google.com/store/apps/details?id=xyz.lifeissimple.hibuddy"**);  
 intent.setType(**"text/plain"**);  
 intent.setType(**"image/jpeg"**);  
 intent.addFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  
 intent.setPackage(**"com.whatsapp"**);  
 **try** {  
 **mview**.getContext().startActivity(intent);  
 }**catch** (android.content.ActivityNotFoundException ex) {  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***ftxmp2***),Toast.***LENGTH\_SHORT***).show();}  
  
  
 }  
  
  
 **private void** downloadImage(String fwd\_post\_image, **final** String buddyidd, **final** String imagefilename, **final int** r) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 }  
  
 filerefer.getDownloadUrl().addOnSuccessListener(**new** OnSuccessListener<Uri>() {  
 @Override  
 **public void** onSuccess(Uri uri) {  
 downloadnewImage(buddyidd,imagefilename,r);*//for changed image after logo* }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 downloadoldImage(buddyidd,imagefilename,r);*//earlier one* }  
 });  
  
  
 }  
  
 **private void** downloadnewImage(**final** String buddyidd, **final** String imagefilename, **final int** r) {  
 *//FirebaseStorage filestore = FirebaseStorage.getInstance();* FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_image/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 }  
  
 **final** File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/Images/xpressions/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
  
  
 **final** StorageReference finalFilerefer = filerefer;  
 **final** StorageReference finalFilerefer1 = filerefer;  
 filerefer.getMetadata().addOnSuccessListener(**new** OnSuccessListener<StorageMetadata>() {  
 @Override  
 **public void** onSuccess(StorageMetadata storageMetadata) {  
 String extension = storageMetadata.getContentType();  
 *//Log.i("dfg",extension);* **final** File localFile;  
 **if** (extension.equals(**"image/gif"**))  
 {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".gif"**);  
 }  
 **else** {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 }  
 downloadfoto(localFile, finalFilerefer, r);  
  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 File localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 downloadfoto(localFile, finalFilerefer1, r);  
  
 }  
 });  
 }  
 **private void** downloadoldImage(**final** String buddyidd, **final** String imagefilename,**final int** r) {  
 StorageReference filerefer;  
 FirebaseStorage filestore;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**);  
 filerefer = filestore.getReferenceFromUrl(**fwd\_post\_image**);  
  
 }**catch** (Exception ex)  
 {  
 filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReferenceFromUrl(**fwd\_post\_image**);  
  
 }  
 **final** File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/Images/xpressions/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
  
  
 **final** StorageReference finalFilerefer = filerefer;  
 filerefer.getMetadata().addOnSuccessListener(**new** OnSuccessListener<StorageMetadata>() {  
 @Override  
 **public void** onSuccess(StorageMetadata storageMetadata) {  
 String extension = storageMetadata.getContentType();  
 *//Log.i("dfg",extension);* **final** File localFile;  
 **if** (extension.equals(**"image/gif"**))  
 {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".gif"**);  
 }  
 **else** {  
 localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 }  
 downloadfoto(localFile, finalFilerefer, r);  
  
  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 File localFile = **new** File(rootpath, buddyidd + **"\_"** + imagefilename + **".jpeg"**);  
 downloadfoto(localFile,finalFilerefer, r);  
  
 }  
 });  
  
 }  
  
  
 **private void** downloadfoto(**final** File localFile, StorageReference filerefer, **final int** r) {  
 *//final File localFile = new File(rootpath, buddyidd+"\_"+imagefilename + ".jpeg");* filerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 String prog= String.*format*(**"%.2f"**,progress);  
 *//prepare pending Intent when notificatio will be clicked* Intent intent= **new** Intent(Intent.***ACTION\_VIEW***);  
 String channelid = **buddyidd**+**"\_"**+**imagefilename**;  
 *// below code to apply FileProvider to open file. It is needed when sdk >23, ie for android 7.0* **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***N***)  
 {  
 intent.setFlags(Intent.***FLAG\_GRANT\_READ\_URI\_PERMISSION***);  
 Uri uri = FileProvider.*getUriForFile*(**mview**.getContext(),**"xyz.lifeissimple.hibuddy.fileProvider"**,localFile);  
 intent.setDataAndType(uri,**"image/\*"**);  
  
 }  
 **else** {  
 Uri uri = Uri.*fromFile*(localFile);  
 intent.setDataAndType(uri,**"image/\*"**);  
 }  
  
  
  
 PendingIntent pintent = PendingIntent.*getActivity*(**mview**.getContext(), r,  
 intent, 0);  
  
  
  
 NotificationCompat.Builder mbuilder = **new** NotificationCompat.Builder(**mview**.getContext(),channelid)  
 .setSmallIcon(android.R.drawable.***stat\_sys\_download***)  
 .setContentTitle(**mview**.getResources().getString(R.string.***ftxmp3***))  
 .setContentText(prog+**"%"**)  
 .setContentIntent(pintent)  
 .setAutoCancel(**true**);  
  
 NotificationManager notificationManager = (NotificationManager) **mview**.getContext().getSystemService(Context.***NOTIFICATION\_SERVICE***);  
 *//below needed for android 8.0* String channelname = **"Hibuddy\_Images"**;  
 **int** importance = NotificationManager.***IMPORTANCE\_HIGH***;  
 **if**(Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***O***){  
 NotificationChannel channel =**new** NotificationChannel(channelid,channelname,importance);  
 **if** (notificationManager != **null**) {  
 notificationManager.createNotificationChannel(channel);}  
 }  
 **if** (notificationManager != **null**) {  
 notificationManager.notify(r, mbuilder.build());  
 }  
  
 }  
 }).addOnSuccessListener(**new** OnSuccessListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {  
  
 }  
 });  
 }  
  
 **private void** downloadVideo(**final** String fwd\_video\_file, **final** String buddyidd, **final** String vidfilename, **final int** r) {  
 FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }  
  
 filerefer.getDownloadUrl().addOnSuccessListener(**new** OnSuccessListener<Uri>() {  
 @Override  
 **public void** onSuccess(Uri uri) {  
 downloadnewVideo(buddyidd,vidfilename,r);*//for changed video after logo* }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 downloadoldVideo(fwd\_video\_file,buddyidd,vidfilename,r);*//earlier one* }  
 });  
 }  
  
 **private void** downloadnewVideo(String buddyidd, String vidfilename, **final int** r) {  
 *// FirebaseStorage filestore = FirebaseStorage.getInstance();* FirebaseStorage filestore;  
 StorageReference filerefer;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReference().child(**"xpress\_video/"**+buddyidd+**"/changed/"**+**post\_key**);  
  
 }  
  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/videos/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
  
 **final** File localFile = **new** File(rootpath,**"vid\_"**+buddyidd+**"\_"**+vidfilename+**".mp4"**);  
 filerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
  
  
 }  
 }).addOnSuccessListener(**new** OnSuccessListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 String prog= String.*format*(**"%.2f"**,progress);  
 *//prepare pending Intent when notificatio will be clicked* Intent intent= **new** Intent(**mview**.getContext(),video\_fullscreen.**class**);  
 intent.putExtra(**"video"**, localFile.toString());  
  
 String channelid=**"xvid"**+System.*currentTimeMillis*();  
 **int** id = (**int**) System.*currentTimeMillis*();  
  
  
 PendingIntent pintent = PendingIntent.*getActivity*(**mview**.getContext(), id,  
 intent, 0);  
  
  
  
 NotificationCompat.Builder mbuilder = **new** NotificationCompat.Builder(**mview**.getContext(),channelid)  
 .setSmallIcon(android.R.drawable.***stat\_sys\_download***)  
 .setContentTitle(**mview**.getResources().getString(R.string.***ftxmp4***))  
 .setContentText(prog+**"%"**)  
 .setContentIntent(pintent)  
 .setAutoCancel(**true**);  
  
  
 NotificationManager notificationManager = (NotificationManager) **mview**.getContext().getSystemService(Context.***NOTIFICATION\_SERVICE***);  
 *//below needed for android 8.0* String channelname = **"Hibuddy\_video"**;  
 **int** importance = NotificationManager.***IMPORTANCE\_HIGH***;  
 **if**(Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***O***){  
 NotificationChannel channel =**new** NotificationChannel(channelid,channelname,importance);  
 **if** (notificationManager != **null**) {  
 notificationManager.createNotificationChannel(channel);}  
 }  
  
 **if** (notificationManager != **null**) {  
  
 notificationManager.notify(r, mbuilder.build());  
 }  
  
  
 }  
 });  
 }  
  
 **private void** downloadoldVideo(String fwd\_video\_file, String buddyidd, String vidfilename, **final int** r) {  
 StorageReference filerefer;  
 FirebaseStorage filestore;  
 **try** {  
 filestore = FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**);  
 filerefer = filestore.getReferenceFromUrl(fwd\_video\_file);  
  
 *//Log.i("rtf",filerefer.toString());* }**catch** (Exception ex)  
 { filestore = FirebaseStorage.*getInstance*();  
 filerefer = filestore.getReferenceFromUrl(fwd\_video\_file);  
 *//Log.i("rtf",filerefer.toString());* }  
 File rootpath = **new** File(Environment.*getExternalStorageDirectory*(),**"/Hibuddy/videos/"**);  
 **if** (!rootpath.exists())  
 {  
 rootpath.mkdirs();  
 }  
  
 **final** File localFile = **new** File(rootpath,**"vid\_"**+buddyidd+**"\_"**+vidfilename+**".mp4"**);  
 filerefer.getFile(localFile).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
  
  
 }  
 }).addOnSuccessListener(**new** OnSuccessListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {  
  
 }  
 }).addOnProgressListener(**new** OnProgressListener<FileDownloadTask.TaskSnapshot>() {  
 @Override  
 **public void** onProgress(FileDownloadTask.TaskSnapshot taskSnapshot) {  
 **double** progress = (100.0 \* taskSnapshot.getBytesTransferred()) / taskSnapshot.getTotalByteCount();  
 String prog= String.*format*(**"%.2f"**,progress);  
 *//prepare pending Intent when notificatio will be clicked* Intent intent= **new** Intent(**mview**.getContext(),video\_fullscreen.**class**);  
 intent.putExtra(**"video"**, localFile.toString());  
  
 String channelid=**"xvid"**+System.*currentTimeMillis*();  
 **int** id = (**int**) System.*currentTimeMillis*();  
  
  
 PendingIntent pintent = PendingIntent.*getActivity*(**mview**.getContext(), id,  
 intent, 0);  
  
  
  
 NotificationCompat.Builder mbuilder = **new** NotificationCompat.Builder(**mview**.getContext(),channelid)  
 .setSmallIcon(android.R.drawable.***stat\_sys\_download***)  
 .setContentTitle(**mview**.getResources().getString(R.string.***ftxmp4***))  
 .setContentText(prog+**"%"**)  
 .setContentIntent(pintent)  
 .setAutoCancel(**true**);  
  
  
 NotificationManager notificationManager = (NotificationManager) **mview**.getContext().getSystemService(Context.***NOTIFICATION\_SERVICE***);  
 *//below needed for android 8.0* String channelname = **"Hibuddy\_video"**;  
 **int** importance = NotificationManager.***IMPORTANCE\_HIGH***;  
 **if**(Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***O***){  
 NotificationChannel channel =**new** NotificationChannel(channelid,channelname,importance);  
 **if** (notificationManager != **null**) {  
 notificationManager.createNotificationChannel(channel);}  
 }  
  
 **if** (notificationManager != **null**) {  
  
 notificationManager.notify(r, mbuilder.build());  
 }  
  
  
 }  
 });  
 }  
  
  
  
  
 **private void** showdialog() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mview**.getContext());  
 builder.setMessage(**mview**.getResources().getString(R.string.***ftxmp5***));  
 builder.setPositiveButton(**mview**.getResources().getString(R.string.***yes***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 *//mdatabase.child("xpress").child(post\_key).removeValue();* **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *// xpressdatabase.child(post\_key).removeValue();* **firestore**.collection(**"xpression"**).document(**post\_key**).delete();  
 *mdatabase*.child(**"xpress\_views"**).child(**post\_key**).removeValue();  
 StorageReference st= FirebaseStorage.*getInstance*(**"gs://bandhoo-xvideos"**).getReference();  
 StorageReference stim= FirebaseStorage.*getInstance*(**"gs://bandhoo-ximages"**).getReference();  
 FirebaseStorage.*getInstance*().getReference().child(**"xpress\_video"**).child(**buddyidd**).child(**buddyidd**+**post\_key**).delete();*//old videos bucket* st.child(**"xpress\_video"**).child(**buddyidd**).child(**buddyidd**+**post\_key**).delete();  
 st.child(**"xpress\_thumbnail"**).child(**buddyidd**).child(**buddyidd**+**post\_key**).delete();  
 stim.child(**"xpress\_image"**).child(**buddyidd**).child(**buddyidd**+**post\_key**).delete();  
  
 }  
 }).start();  
 }  
 }  
  
 );  
 builder.setNegativeButton(**mview**.getResources().getString(R.string.***cancel***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
  
 }  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 }  
 **private void** showdialogReport(**final** String post\_key) {  
 **final** String [] reports = **new** String[]{**mview**.getResources().getString(R.string.***report1***),**mview**.getResources().getString(R.string.***report2***),**mview**.getResources().getString(R.string.***report3***)};  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mview**.getContext());  
 builder.setCancelable(**false**);  
 builder.setTitle(**mview**.getResources().getString(R.string.***selectreport***));  
 builder.setItems(reports, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 *mdatabase*.child(**"reports"**).child(post\_key).child(**uidd**).setValue(reports[which]);  
 Toast.*makeText*(**mview**.getContext(),**mview**.getResources().getString(R.string.***reportsuccess***),Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 builder.setNegativeButton(**mview**.getResources().getString(R.string.***cancel***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
 builder.show();  
 }  
 }  
  
  
}