activity_dashboard.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".DashboardActivity">

    <FrameLayout
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
    </FrameLayout>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/menu_nav">
    </com.google.android.material.bottomnavigation.BottomNavigationView>
</LinearLayout>
```

fragment_home.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HomeFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/postsRecyclerview"/>

</FrameLayout>
```

fragment_profile.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#F1EDED"
tools:context=".ProfileFragment">

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/coverIv"
            android:layout_width="match_parent"
            android:layout_height="180dp"
            android:scaleType="fitXY"
            android:background="@color/colorPrimaryDark">

        </ImageView>

        <LinearLayout
            android:id="@+id/linearLayout"
            android:layout_width="match_parent"
            android:layout_height="120dp"
            android:orientation="horizontal"
            android:layout_marginTop="100dp">

            <ImageView
                android:id="@+id/avatarIv"
                android:layout_marginStart="20dp"
                android:background="@color/colorPrimary"
                android:src="@drawable/ic_default_img_white"
                android:padding="5dp"
                android:layout_width="120dp"
                android:layout_height="120dp"
                android:layout_marginLeft="20dp"/>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:background="#77000000">

                <TextView
                    android:id="@+id/nameTv"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="5dp"
                    android:layout_marginTop="5dp"
```

```
                        android:textSize="25sp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                    <TextView
                        android:id="@+id/emailTv"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                    <TextView
                        android:id="@+id/phoneTv"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                </LinearLayout>

            </LinearLayout>

            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/recyclerview_posts"
                android:layout_below="@+id/linearLayout"
                android:layout_width="match_parent"
                android:layout_height="match_parent"/>

        </RelativeLayout>

    </ScrollView>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:src="@drawable/ic_edit_white"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_margin="10dp"
        android:layout_alignParentRight="true"/>

</RelativeLayout>
```

fragment_users.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UsersFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/users_recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

    </androidx.recyclerview.widget.RecyclerView>

</FrameLayout>
```

fragment_chat_list.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".ChatListFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        tools:listitem="@layout/row_chatlist"
        app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"/>

</FrameLayout>
```

activity_add_post.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AddPostActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:orientation="vertical">

        <EditText
            android:id="@+id/pTitleEt"
            android:background="@drawable/edittextstyle"
            android:padding="10dp"
            android:hint="Enter Title"
            android:singleLine="true"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <ImageView
            android:id="@+id/pImageIv"
            android:background="@color/colorPrimary"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minHeight="200dp"/>

        <EditText
            android:id="@+id/pDescriptionEt"
            android:background="@drawable/edittextstyle"
            android:padding="10dp"
            android:hint="Enter Title"
            android:inputType="textCapSentences|textMultiLine"
            android:minHeight="120dp"
            android:gravity="start"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/pUploadBtn"
            style="@style/Widget.AppCompat.Button.Colored"
            android:text="Upload"
            android:layout_gravity="end"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

    </LinearLayout>

</ScrollView>
```

activity_chat.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F1EDED"
    tools:context=".ChatActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?android:attr/actionBarSize"
        android:background="@color/colorPrimaryDark"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:id="@+id/profileIv"
            android:layout_width="35dp"
            android:layout_height="35dp"
            android:scaleType="centerCrop"
            android:layout_marginRight="4dp"
            android:src="@drawable/ic_default_img_white"
            app:c_background_color="@color/colorPrimaryDark"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginStart="20dp"
            android:layout_marginLeft="20dp"
            android:gravity="center">

            <TextView
                android:id="@+id/nameIv"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="His Name"
                android:textColor="@color/colorWhite"
                android:textSize="18sp"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/userStatusTv"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="online"
                android:textColor="@color/colorWhite"
                android:textStyle="bold"/>

        </LinearLayout>

    </androidx.appcompat.widget.Toolbar>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/chat_recyclerView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/toolbar"
        android:layout_above="@+id/chatLayout"/>

        <LinearLayout
            android:id="@+id/chatLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:background="@color/colorWhite"
            android:gravity="center"
            android:orientation="horizontal">

            <EditText
                android:id="@+id/messageEt"
                android:layout_weight="1"
                android:background="@null"
                android:hint="Start typing..."
                android:padding="15dp"
                android:inputType="textCapSentences|textMultiLine"
                android:layout_width="0dp"
                android:layout_height="wrap_content" />

            <ImageButton
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:id="@+id/sendBtn"
                android:background="@null"
                android:src="@drawable/ic_send"/>

        </LinearLayout>

</RelativeLayout>
```

activity_post_detail.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PostDetailActivity">

    <androidx.core.widget.NestedScrollView
        android:layout_above="@id/commentsLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

```
app:cardCornerRadius="3dp"
app:cardElevation="3dp"
app:contentPadding="5dp"
app:cardUseCompatPadding="true"
app:cardBackgroundColor="@color/colorWhite">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <LinearLayout
        android:id="@+id/profileLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:id="@+id/uPictureIv"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_default_img"
            android:layout_width="50dp"
            android:layout_height="50dp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_weight="1">

            <TextView
                android:id="@+id/uNameTv"
                android:text="Name"
                android:textColor="@color/colorBlack"
                android:textSize="20sp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/pTimeTv"
                android:text="33 min"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </LinearLayout>

        <ImageButton
            android:id="@+id/moreBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
        android:background="@null"
        android:src="@drawable/ic_more_black"/>

</LinearLayout>

<TextView
    android:id="@+id/pTitleTv"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:autoLink="all"/>

<TextView
    android:id="@+id/pDescriptionTv"
    android:text="The description of the post"
    android:textColor="@color/colorBlack"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:autoLink="all"/>

<ImageView
    android:id="@+id/pImageIv"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="@color/colorWhite"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/pLikesTv"
        android:text="1.2k Accepted"
        android:layout_weight="1"
        android:textColor="@color/colorPrimary"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/pCommentsTv"
        android:text="1.2k Comments"
        android:layout_weight="1"
        android:textAlignment="textEnd"
        android:gravity="end"
        android:textColor="@color/colorPrimary"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
        </LinearLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="#F5F0F0"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center">

            <Button
                android:id="@+id/likeBtn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Accept"
                android:drawableLeft="@drawable/ic_accept_black"
                android:drawableStart="@drawable/ic_accept_black"
                android:drawablePadding="5dp"
                android:background="@color/colorWhite"/>

            <Button
                android:id="@+id/shareBtn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Share"
                android:drawableLeft="@drawable/ic_share_black"
                android:drawableStart="@drawable/ic_share_black"
                android:drawablePadding="5dp"
                android:background="@color/colorWhite"/>

        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>

<TextView
    android:text="Comments"
    android:textColor="@color/colorBlack"
    android:drawableStart="@drawable/ic_comment_black"
    android:drawablePadding="5dp"
    android:padding="2dp"
    android:drawableLeft="@drawable/ic_comment_black"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/recyclerView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"/>
        </LinearLayout>

    </androidx.core.widget.NestedScrollView>

    <RelativeLayout
        android:id="@+id/commentsLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:id="@+id/cAvatarIv"
            android:src="@drawable/ic_default_img"/>

        <EditText
            android:id="@+id/commentEt"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_toEndOf="@+id/cAvatarIv"
            android:layout_toStartOf="@+id/sendBtn"
            android:background="@color/colorWhite"
            android:padding="10dp"
            android:hint="Enter comment..."
            android:inputType="textCapSentences|textMultiLine"
            android:layout_centerVertical="true"
            android:layout_margin="1dp"
            android:layout_toRightOf="@+id/cAvatarIv"
            android:layout_toLeftOf="@+id/sendBtn"/>

        <ImageButton
            android:layout_centerVertical="true"
            android:background="@color/colorWhite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/sendBtn"
            android:layout_alignParentEnd="true"
            android:src="@drawable/ic_send"
            android:layout_alignParentRight="true"/>
    </RelativeLayout>

</RelativeLayout>
```

activity_there_profile.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F1EDED"
    tools:context=".ThereProfileActivity">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/coverIv"
                android:layout_width="match_parent"
                android:layout_height="180dp"
                android:scaleType="fitXY"
                android:background="@color/colorPrimaryDark">

            </ImageView>

            <LinearLayout
                android:id="@+id/linearLayout"
                android:layout_width="match_parent"
                android:layout_height="120dp"
                android:orientation="horizontal"
                android:layout_marginTop="100dp">

                <ImageView
                    android:id="@+id/avatarIv"
                    android:layout_marginStart="20dp"
                    android:background="@color/colorPrimary"
                    android:src="@drawable/ic_default_img_white"
                    android:padding="5dp"
                    android:layout_width="120dp"
                    android:layout_height="120dp"
                    android:layout_marginLeft="20dp"/>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">

                    <TextView
                        android:id="@+id/nameTv"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:layout_marginTop="5dp"
                        android:textSize="25sp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                    <TextView
                        android:id="@+id/emailTv"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                    <TextView
                        android:id="@+id/phoneTv"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="5dp"
                        android:layout_marginLeft="5dp"
                        android:textColor="@color/colorWhite"/>

                </LinearLayout>

            </LinearLayout>

            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/recyclerview_posts"
                android:layout_below="@+id/linearLayout"
                android:layout_width="match_parent"
                android:layout_height="match_parent"/>

        </RelativeLayout>

</ScrollView>
```

row_chat_left.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/messageLayout"
    android:padding="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <com.blogspot.atifsoftwares.circularimageview.CircularImageView
```

```
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:id="@+id/profileIv"
        app:c_border_color="@null"
        android:src="@drawable/ic_default_img"/>

    <TextView
        android:id="@+id/messageTv"
        android:layout_weight="1"
        android:textSize="16sp"
        android:textColor="@color/colorBlack"
        android:background="@drawable/bg_receiver"
        android:padding="15dp"
        android:text="His Message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/timeTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="02/03/2019 05:59PM"
            android:textColor="@color/colorBlack"
            android:textSize="12sp"/>

    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/isSeenTv"
        android:visibility="gone"
        android:gravity="end"
        android:textAlignment="textEnd"
        android:text="delivered"/>

</LinearLayout>
```

row_chat_right.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/messageLayout"
    android:padding="10dp">

    <LinearLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:id="@+id/profileIv"
            app:c_border_color="@null"
            android:visibility="gone"
            android:src="@drawable/ic_default_img"/>

        <TextView
            android:id="@+id/timeTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="02/03/2019 05:59PM"
            android:textColor="@color/colorBlack"
            android:textSize="12sp"/>

        <TextView
            android:id="@+id/messageTv"
            android:layout_weight="1"
            android:textSize="16sp"
            android:textColor="@color/colorBlack"
            android:background="@drawable/bg_sender"
            android:padding="15dp"
            android:text="His Message"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/isSeenTv"
        android:gravity="end"
        android:textAlignment="textEnd"
        android:text="delivered"/>

</LinearLayout>
```

row_chatlist.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:contentPadding="3dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:id="@+id/profileIv"
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:src="@drawable/ic_default_img"/>

        <ImageView
            android:id="@+id/onlineStatusIv"
            android:layout_width="25dp"
            android:layout_height="25dp" />

        <TextView
            android:id="@+id/nameTv"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="His Name"
            android:layout_marginStart="4dp"
            android:layout_marginLeft="3dp"
            android:textColor="@color/colorBlack"
            android:textSize="18sp"
            android:layout_toEndOf="@+id/profileIv"
            android:layout_toRightOf="@+id/profileIv"/>

        <TextView
            android:id="@+id/lastMessageTv"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="The last message"
            android:maxLines="2"
            android:layout_marginTop="3dp"
            android:layout_marginStart="4dp"
            android:layout_marginLeft="4dp"
            android:layout_toEndOf="@+id/profileIv"
            android:layout_toRightOf="@+id/profileIv"
            android:layout_below="@+id/nameTv"/>

    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

row_comments.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="1dp"
    app:contentPadding="2dp"
    app:cardBackgroundColor="@color/colorWhite">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:id="@+id/avatarIv"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginEnd="5dp"
            android:layout_marginRight="5dp"
            android:src="@drawable/ic_default_img"/>

        <TextView
            android:id="@+id/nameTv"
            android:layout_toRightOf="@+id/avatarIv"
            android:text="Bhargav"
            android:textColor="@color/colorBlack"
            android:textSize="16sp"
            android:textStyle="bold"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/commentTv"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/nameTv"
            android:layout_toEndOf="@+id/avatarIv"
            android:text="This is actual comment"
            android:layout_toRightOf="@+id/avatarIv"
            android:textColor="@color/colorBlack"/>

        <TextView
            android:id="@+id/timeTv"
            android:text="12/06/2019"
            android:layout_below="@+id/commentTv"
            android:layout_toEndOf="@+id/avatarIv"
            android:layout_toRightOf="@+id/avatarIv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

row_posts.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="3dp"
    app:cardElevation="3dp"
    app:contentPadding="5dp"
    app:cardUseCompatPadding="true"
    app:cardBackgroundColor="@color/colorWhite">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:id="@+id/profileLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical">

            <com.blogspot.atifsoftwares.circularimageview.CircularImageView
                android:id="@+id/uPictureIv"
                android:scaleType="centerCrop"
                android:src="@drawable/ic_default_img"
                android:layout_width="50dp"
                android:layout_height="50dp" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:layout_weight="1">

                <TextView
                    android:id="@+id/uNameTv"
                    android:text="Name"
                    android:textColor="@color/colorBlack"
                    android:textSize="20sp"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content" />
                <TextView
                    android:id="@+id/pTimeTv"
                    android:text="33 min"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content" />

            </LinearLayout>

            <ImageButton
                android:id="@+id/moreBtn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@null"
                android:src="@drawable/ic_more_black"/>

        </LinearLayout>

        <TextView
            android:id="@+id/pTitleTv"
            android:textSize="16sp"
            android:textStyle="bold"
            android:text="Title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autoLink="all"/>

        <TextView
            android:id="@+id/pDescriptionTv"
            android:text="The description of the post"
            android:textColor="@color/colorBlack"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:autoLink="all"/>

        <ImageView
            android:id="@+id/pImageIv"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:background="@color/colorWhite"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/pLikesTv"
                android:text="1.2k Accepted"
                android:layout_weight="1"
                android:textColor="@color/colorPrimary"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/pCommentsTv"
                android:text="1.2k Comments"
                android:layout_weight="1"
                android:textAlignment="textEnd"
                android:gravity="end"
                android:textColor="@color/colorPrimary"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </LinearLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="#F5F0F0"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center">

            <Button
                android:id="@+id/likeBtn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Accept"
                android:drawableLeft="@drawable/ic_accept_black"
                android:drawableStart="@drawable/ic_accept_black"
                android:drawablePadding="5dp"
                android:background="@color/colorWhite"/>

            <Button
                android:id="@+id/commentBtn"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Comment"
                android:drawableLeft="@drawable/ic_comment_black"
                android:drawableStart="@drawable/ic_comment_black"
                android:drawablePadding="5dp"
                android:background="@color/colorWhite"/>

            <Button
                android:id="@+id/shareBtn"
                android:layout_width="wrap_content"
```

```
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Share"
                android:drawableLeft="@drawable/ic_share_black"
                android:drawableStart="@drawable/ic_share_black"
                android:drawablePadding="5dp"
                android:background="@color/colorWhite"/>

        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>
```

row_users.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardUseCompatPadding="true"
    app:cardCornerRadius="2dp"
    app:cardElevation="2dp"
    app:cardBackgroundColor="@color/colorWhite"
    app:contentPadding="3dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:orientation="horizontal">

        <com.blogspot.atifsoftwares.circularimageview.CircularImageView
            android:id="@+id/avatarIv"
            android:src="@drawable/ic_default_img"
            android:layout_marginRight="5dp"
            android:layout_width="70dp"
            android:layout_height="70dp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:id="@+id/nameTv"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Person Name"
                android:textSize="18sp"
                android:textColor="@color/colorBlack"/>
            <TextView
                android:id="@+id/emailTv"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Person Email"
                android:textColor="@color/colorBlack"/>
        </LinearLayout>

    </LinearLayout>

</androidx.cardview.widget.CardView>
```

Java Folder:

DashboardActivity.java:

```
package com.bhargav.verifyproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentTransaction;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.MenuItem;

import com.bhargav.verifyproject.notifications.Token;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;

public class  DashboardActivity extends AppCompatActivity {

    FirebaseAuth firebaseAuth;

    ActionBar actionBar;

    String mUID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        actionBar = getSupportActionBar();
        actionBar.setTitle("Profile");

        firebaseAuth = FirebaseAuth.getInstance();

        BottomNavigationView navigationView = findViewById(R.id.navigation);
        navigationView.setOnNavigationItemSelectedListener(selectedListener);

        actionBar.setTitle("Home");
        HomeFragment fragment1 = new HomeFragment();
        FragmentTransaction ft1 = getSupportFragmentManager().beginTransaction();
        ft1.replace(R.id.content,fragment1,"");
        ft1.commit();

        checkUserStatus();
    }

    @Override
    protected void onResume() {
        checkUserStatus();
        super.onResume();
    }

    public void updateToken(String token) {
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Tokens");
        Token mToken = new Token(token);
        ref.child(mUID).setValue(mToken);
    }

    private BottomNavigationView.OnNavigationItemSelectedListener selectedListener =
            new BottomNavigationView.OnNavigationItemSelectedListener() {
                @Override
                public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
                    switch (menuItem.getItemId()) {
                        case R.id.nav_home:
                            actionBar.setTitle("Home");
                            HomeFragment fragment1 = new HomeFragment();
                            FragmentTransaction ft1 = getSupportFragmentManager().beginTransaction();
                            ft1.replace(R.id.content,fragment1,"");
                            ft1.commit();
                            return true;
                        case R.id.nav_profile:
                            actionBar.setTitle("Profile");
                            ProfileFragment fragment2 = new ProfileFragment();
                            FragmentTransaction ft2 = getSupportFragmentManager().beginTransaction();
                            ft2.replace(R.id.content,fragment2,"");
                            ft2.commit();
                            return true;
                        case R.id.nav_users:
                            actionBar.setTitle("Users");
                            UsersFragment fragment3 = new UsersFragment();
                            FragmentTransaction ft3 = getSupportFragmentManager().beginTransaction();
                            ft3.replace(R.id.content,fragment3,"");
                            ft3.commit();
                            return true;
                        case R.id.nav_chat:
                            actionBar.setTitle("Chats");
                            ChatListFragment fragment4 = new ChatListFragment();
                            FragmentTransaction ft4 = getSupportFragmentManager().beginTransaction();
                            ft4.replace(R.id.content,fragment4,"");
                            ft4.commit();
                            return true;
                    }
                    return false;
                }
            };

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {
            mUID = user.getUid();

            SharedPreferences sp = getSharedPreferences("SP_USER", MODE_PRIVATE);
            SharedPreferences.Editor editor = sp.edit();
            editor.putString("Current_USERID", mUID);
            editor.apply();

            updateToken(FirebaseInstanceId.getInstance().getToken());
        }
        else {
            startActivity(new Intent(DashboardActivity.this,MainActivity.class));
            finish();
        }
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        finish();
    }

    protected void onStart() {
        checkUserStatus();
        super.onStart();
    }


}
```

HomeFragment.java:

```java
package com.bhargav.verifyproject;


import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.view.MenuItemCompat;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SearchView;
import android.widget.Toast;

import com.bhargav.verifyproject.adapters.AdapterPosts;
import com.bhargav.verifyproject.models.ModelPost;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.ValueEventListener;

import java.util.ArrayList;
import java.util.List;


/**
 * A simple {@link Fragment} subclass.
 */
public class HomeFragment extends Fragment {

    FirebaseAuth firebaseAuth;

    RecyclerView recyclerView;
    List<ModelPost> postList;
    AdapterPosts adapterPosts;

    public HomeFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_home, container, false);

        firebaseAuth = FirebaseAuth.getInstance();

        recyclerView = view.findViewById(R.id.postsRecyclerview);
        LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());

        layoutManager.setStackFromEnd(true);
        layoutManager.setReverseLayout(true);

        recyclerView.setLayoutManager(layoutManager);

        postList = new ArrayList<>();

        loadPosts();

        return view;

    }

    private void loadPosts() {
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost modelPost = ds.getValue(ModelPost.class);
                    postList.add(modelPost);
                    adapterPosts = new AdapterPosts(getActivity(), postList);
                    recyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(getActivity(),""+databaseError.getMessage(),
  Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void searchPosts(final String searchQuery) {
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost modelPost = ds.getValue(ModelPost.class);

                    if(modelPost.getpTitle().toLowerCase().contains(searchQuery.toLowerCase()) ||
                            modelPost.getpDescr().toLowerCase().contains(searchQuery.toLowerCase())) {
                        postList.add(modelPost);
                    }
                    adapterPosts = new AdapterPosts(getActivity(), postList);
                    recyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(getActivity(),""+databaseError.getMessage(),
   Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {

        }
        else {
            startActivity(new Intent(getActivity(),MainActivity.class));
            getActivity().finish();
        }
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        setHasOptionsMenu(true);
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu_main,menu);

        MenuItem item = menu.findItem(R.id.action_search);
        SearchView searchView = (SearchView) MenuItemCompat.getActionView(item);

        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchPosts(s);
                }
                else {
                    loadPosts();
                }
                return false;
            }

            @Override
            public boolean onQueryTextChange(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchPosts(s);
                }
                else {
                    loadPosts();
                }
                return false;
            }
        });

        super.onCreateOptionsMenu(menu, inflater);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.action_logout) {
            firebaseAuth.signOut();
            checkUserStatus();
        }
        if(id == R.id.action_add_post) {
            startActivity(new Intent(getActivity(), AddPostActivity.class));
        }
        return super.onOptionsItemSelected(item);
    }

}
```

ProfileFragment.java:

```
package com.bhargav.verifyproject;


import android.Manifest;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
```

```java
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.content.ContextCompat;
import androidx.core.view.MenuItemCompat;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.provider.MediaStore;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.SearchView;
import android.widget.TextView;
import android.widget.Toast;

import com.bhargav.verifyproject.adapters.AdapterPosts;
import com.bhargav.verifyproject.models.ModelPost;
import com.bhargav.verifyproject.notifications.Data;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

import java.security.Key;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import static android.app.Activity.RESULT_OK;
import static com.google.firebase.storage.FirebaseStorage.getInstance;


/**
 * A simple {@link Fragment} subclass.
 */
public class ProfileFragment extends Fragment {

    FirebaseAuth firebaseAuth;
    FirebaseUser user;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;

    StorageReference storageReference;

    String storagePath = "Users_Profile_Cover_Imags/";

    ImageView avatarIv,coverIv;
    TextView nameTv,emailTv,phoneTv;
    FloatingActionButton fab;
    RecyclerView postRecyclerView;

    ProgressDialog pd;

    private static final int CAMERA_REQUEST_CODE = 100;
    private static final int STORAGE_REQUEST_CODE = 200;
    private static final int IMAGE_PICK_GALLERY_CODE = 300;
    private static final int IMAGE_PICK_CAMERA_CODE = 400;

    String cameraPermissions[];
    String storagePermissions[];

    List<ModelPost> postList;
    AdapterPosts adapterPosts;
    String uid;

    Uri image_uri;

    String profileOrCoverPhoto;

    public ProfileFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_profile,container,false);

        firebaseAuth = FirebaseAuth.getInstance();
        user = firebaseAuth.getCurrentUser();
        firebaseDatabase = FirebaseDatabase.getInstance();
        databaseReference = firebaseDatabase.getReference("Users");
        storageReference = getInstance().getReference();

        cameraPermissions = new String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE};
        storagePermissions = new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE};

        avatarIv = view.findViewById(R.id.avatarIv);
        coverIv = view.findViewById(R.id.coverIv);
        nameTv = view.findViewById(R.id.nameTv);
        emailTv = view.findViewById(R.id.emailTv);
        phoneTv = view.findViewById(R.id.phoneTv);
        fab = view.findViewById(R.id.fab);
        postRecyclerView = view.findViewById(R.id.recyclerview_posts);

        pd = new ProgressDialog(getActivity());

        Query query = databaseReference.orderByChild("email").equalTo(user.getEmail());
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot ds : dataSnapshot.getChildren()) {
                    String name = ""+ds.child("name").getValue();
                    String email = ""+ds.child("email").getValue();
                    String phone = ""+ds.child("phone").getValue();
                    String image = ""+ds.child("image").getValue();
                    String cover = ""+ds.child("cover").getValue();

                    nameTv.setText(name);
                    emailTv.setText(email);
                    phoneTv.setText(phone);

                    try {
                        Picasso.get().load(image).into(avatarIv);
                    }
                    catch(Exception e) {
                        Picasso.get().load(R.drawable.ic_default_img_white).into(avatarIv);
                    }

                    try {
                        Picasso.get().load(cover).into(coverIv);
                    }
                    catch(Exception e) {

                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showEditProfileDialog();
            }
        });

        postList = new ArrayList<>();

        checkUserStatus();
        loadMyPosts();

        return view;
    }

    private void loadMyPosts() {

        LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());
        layoutManager.setStackFromEnd(true);
        layoutManager.setReverseLayout(true);
        postRecyclerView.setLayoutManager(layoutManager);
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        Query query = ref.orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost myPosts = ds.getValue(ModelPost.class);
                    postList.add(myPosts);
                    adapterPosts = new AdapterPosts(getActivity(), postList);
                    postRecyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(getActivity(),""+databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
    private void searchMyPosts(final String searchQuery) {

        LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());
        layoutManager.setStackFromEnd(true);
        layoutManager.setReverseLayout(true);
        postRecyclerView.setLayoutManager(layoutManager);
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        Query query = ref.orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost myPosts = ds.getValue(ModelPost.class);

                    if(myPosts.getpTitle().toLowerCase().contains(searchQuery.toLowerCase()) ||
                    myPosts.getpDescr().toLowerCase().contains(searchQuery.toLowerCase())) {
                        postList.add(myPosts);
                    }

                    postList.add(myPosts);
                    adapterPosts = new AdapterPosts(getActivity(), postList);
                    postRecyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(getActivity(),""+databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private boolean checkStoragePermission() {
        boolean result = ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.WRITE_EXTERNAL_STORAGE)
                == (PackageManager.PERMISSION_GRANTED);
        return result;
    }

    private void requestStoragePermission() {
        requestPermissions(storagePermissions, STORAGE_REQUEST_CODE);
    }

    private boolean checkCameraPermission() {
        boolean result = ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.CAMERA)
                == (PackageManager.PERMISSION_GRANTED);
        boolean result1 = ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.CAMERA)
```

```
            == (PackageManager.PERMISSION_GRANTED);
    return result && result1;
}

private void requestCameraPermission() {
    requestPermissions(cameraPermissions, CAMERA_REQUEST_CODE);
}

private void showEditProfileDialog() {
    String options[] = {"Edit Profile Picture","Edit Cover Photo","Edit Name","Edit Phone"};
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    builder.setTitle("Choose Action");
    builder.setItems(options, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            if(which == 0) {
                pd.setMessage("Updating Profile Picture");
                profileOrCoverPhoto = "image";
                showImagePicDialog();
            }
            else if(which == 1) {
                pd.setMessage("Updating Cover Picture");
                profileOrCoverPhoto = "cover";
                showImagePicDialog();
            }
            else if(which == 2) {
                pd.setMessage("Updating Name");
                showNamePhoneUpdateDialog("name");
            }
            else if(which == 3) {
                pd.setMessage("Updating Phone");
                showNamePhoneUpdateDialog("phone");
            }
        }
    });
    builder.create().show();
}

private void showNamePhoneUpdateDialog(final String key) {
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    builder.setTitle("Update "+ key);

    LinearLayout linearLayout = new LinearLayout(getActivity());
    linearLayout.setOrientation(LinearLayout.VERTICAL);
    linearLayout.setPadding(10,10,10,10);

    final EditText editText = new EditText(getActivity());
    editText.setHint("Enter "+key);
    linearLayout.addView(editText);

    builder.setView(linearLayout);
```

```
        builder.setPositiveButton("Update", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                final String value = editText.getText().toString().trim();
                if(!TextUtils.isEmpty(value)) {
                    pd.show();
                    HashMap<String, Object> result = new HashMap<>();
                    result.put(key,value);

                    databaseReference.child(user.getUid()).updateChildren(result)
                            .addOnSuccessListener(new OnSuccessListener<Void>() {
                                @Override
                                public void onSuccess(Void aVoid) {
                                    pd.dismiss();
                                    Toast.makeText(getActivity(),"Updated...",Toast.LENGTH_SHORT).show();
                                }
                            })
                            .addOnFailureListener(new OnFailureListener() {
                                @Override
                                public void onFailure(@NonNull Exception e) {
                                    pd.dismiss();

Toast.makeText(getActivity(),""+e.getMessage(),Toast.LENGTH_SHORT).show();
                                }
                            });

                    if (key.equals("name")) {
                        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
                        Query query = ref.orderByChild("uid").equalTo(uid);
                        query.addValueEventListener(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                for(DataSnapshot ds: dataSnapshot.getChildren()) {
                                    String child = ds.getKey();
                                    dataSnapshot.getRef().child(child).child("uName").setValue(value);
                                }
                            }

                            @Override
                            public void onCancelled(@NonNull DatabaseError databaseError) {

                            }
                        });

                        ref.addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                    String child = ds.getKey();
                                    if (dataSnapshot.child(child).hasChild("Comments")) {
                                        String child1 = ""+dataSnapshot.child(child).getKey();
                                        Query child2 =
FirebaseDatabase.getInstance().getReference("Posts").child(child1).child("Comments").orderByChild(
"uid").equalTo(uid);
                                        child2.addValueEventListener(new ValueEventListener() {
                                            @Override
                                            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                                    String child = ds.getKey();
                                                    dataSnapshot.getRef().child(child).child("uName").setValue(value);
                                                }
                                            }

                                            @Override
                                            public void onCancelled(@NonNull DatabaseError databaseError) {

                                            }
                                        });
                                    }
                                }
                            }

                            @Override
                            public void onCancelled(@NonNull DatabaseError databaseError) {

                            }
                        });
                    }
                }
                else {
                    Toast.makeText(getActivity(),"Please enter "+key,Toast.LENGTH_SHORT).show();
                }
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });

        builder.create().show();
    }

    private void showImagePicDialog() {

        String options[] = {"Camera","Gallery"};
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("Pick Image From");
        builder.setItems(options, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                if(which == 0) {
                    if(!checkCameraPermission()) {
                        requestCameraPermission();
                    }
                    else {
                        pickFromCamera();
                    }
                }
                else if(which == 1) {
                    if(!checkStoragePermission()) {
                        requestStoragePermission();
                    }
                    else {
                        pickFromGallery();
                    }
                }
            }
        });
        builder.create().show();
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        switch(requestCode) {
            case CAMERA_REQUEST_CODE:{
                if(grantResults.length > 0) {
                    boolean cameraAccepted= grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                    boolean writeStorageAccepted= grantResults[1] ==
PackageManager.PERMISSION_GRANTED;
                    if(cameraAccepted && writeStorageAccepted) {
                        pickFromCamera();
                    }
                    else {
                        Toast.makeText(getActivity(), "Please enable camera & storage
permmission",Toast.LENGTH_SHORT).show();
                    }
                }
            }
            break;
            case STORAGE_REQUEST_CODE:{
                if(grantResults.length > 0) {
                    boolean writeStorageAccepted= grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                    if(writeStorageAccepted) {
                        pickFromGallery();
                    }
                    else {
                        Toast.makeText(getActivity(), "Please enable storage
permmission",Toast.LENGTH_SHORT).show();
                    }
                }
            }
            break;
        }
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        if(resultCode == RESULT_OK) {
            if(requestCode == IMAGE_PICK_GALLERY_CODE) {
                image_uri = data.getData();
                uploadProfileCoverPhoto(image_uri);
            }
            if(requestCode == IMAGE_PICK_CAMERA_CODE) {
                uploadProfileCoverPhoto(image_uri);
            }
        }
        super.onActivityResult(requestCode, resultCode, data);
    }

    private void uploadProfileCoverPhoto(final Uri uri) {

        pd.show();

        String filePathAndName = storagePath+ ""+ profileOrCoverPhoto +"_"+ user.getUid();

        StorageReference storageReference2nd = storageReference.child(filePathAndName);
        storageReference2nd.putFile(uri)
                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                    @Override
                    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
                        while (!uriTask.isSuccessful());
                        final Uri downloadUri = uriTask.getResult();

                        if(uriTask.isSuccessful()) {
                            HashMap<String,Object> results = new HashMap<>();
                            results.put(profileOrCoverPhoto,downloadUri.toString());

                            databaseReference.child(user.getUid()).updateChildren(results)
                                    .addOnSuccessListener(new OnSuccessListener<Void>() {
                                        @Override
                                        public void onSuccess(Void aVoid) {
                                            pd.dismiss();
                                            Toast.makeText(getActivity(),"Image
Updated...",Toast.LENGTH_SHORT).show();
                                        }
                                    }).addOnFailureListener(new OnFailureListener() {
                                @Override
                                public void onFailure(@NonNull Exception e) {
                                    pd.dismiss();
                                    Toast.makeText(getActivity(),"Error Updating
Image...",Toast.LENGTH_SHORT).show();
                                }
                            });

                            if (profileOrCoverPhoto.equals("image")) {
                                DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
                                Query query = ref.orderByChild("uid").equalTo(uid);
                                query.addValueEventListener(new ValueEventListener() {
                                    @Override
                                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                        for(DataSnapshot ds: dataSnapshot.getChildren()) {
                                            String child = ds.getKey();

dataSnapshot.getRef().child(child).child("uDp").setValue(downloadUri.toString());
                                        }
                                    }

                                    @Override
                                    public void onCancelled(@NonNull DatabaseError databaseError) {

                                    }
                                });

                                ref.addListenerForSingleValueEvent(new ValueEventListener() {
                                    @Override
                                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                        for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                            String child = ds.getKey();
                                            if (dataSnapshot.child(child).hasChild("Comments")) {
                                                String child1 = ""+dataSnapshot.child(child).getKey();
                                                Query child2 =
FirebaseDatabase.getInstance().getReference("Posts").child(child1).child("Comments").orderByChild(
"uid").equalTo(uid);
                                                child2.addValueEventListener(new ValueEventListener() {
                                                    @Override
                                                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                                        for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                                            String child = ds.getKey();

dataSnapshot.getRef().child(child).child("uDp").setValue(downloadUri.toString());
                                                        }
                                                    }

                                                    @Override
                                                    public void onCancelled(@NonNull DatabaseError databaseError) {

                                                    }
                                                });
                                            }
                                        }
                                    }

                                    @Override
                                    public void onCancelled(@NonNull DatabaseError databaseError) {

                                    }
                                });
                            }
                        }
                        else {
                            pd.dismiss();
                            Toast.makeText(getActivity(),"Some error
occured",Toast.LENGTH_SHORT).show();
                        }
                    }
                }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                pd.dismiss();;
                Toast.makeText(getActivity(),e.getMessage(),Toast.LENGTH_SHORT).show();
            }
        });

    }

    private void pickFromCamera() {
        ContentValues values = new ContentValues();
        values.put(MediaStore.Images.Media.TITLE,"Temp Pic");
        values.put(MediaStore.Images.Media.DESCRIPTION,"Temp Description");

        image_uri =
getActivity().getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
values);

        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, image_uri);
        startActivityForResult(cameraIntent, IMAGE_PICK_CAMERA_CODE);
    }

    private void pickFromGallery() {
        Intent galleryIntent = new Intent(Intent.ACTION_PICK);
        galleryIntent.setType("image/*");
        startActivityForResult(galleryIntent, IMAGE_PICK_GALLERY_CODE);
    }

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {
            uid = user.getUid();
        }
        else {
            startActivity(new Intent(getActivity(),MainActivity.class));
            getActivity().finish();
        }
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        setHasOptionsMenu(true);
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu_main,menu);

        MenuItem item = menu.findItem(R.id.action_search);
        SearchView searchView = (SearchView) MenuItemCompat.getActionView(item);
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchMyPosts(s);
                }
                else {
                    loadMyPosts();
                }
                return false;
            }

            @Override
            public boolean onQueryTextChange(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchMyPosts(s);
                }
                else {
                    loadMyPosts();
                }
                return false;
            }
        });

        super.onCreateOptionsMenu(menu, inflater);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.action_logout) {
            firebaseAuth.signOut();
```

```
            checkUserStatus();
        }
        if (id == R.id.action_add_post) {
            startActivity(new Intent(getActivity(),AddPostActivity.class));
        }
        return super.onOptionsItemSelected(item);
    }
}
```

UsersFragment.java:

```
package com.bhargav.verifyproject;


import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.view.MenuItemCompat;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SearchView;

import com.bhargav.verifyproject.adapters.AdapterUsers;
import com.bhargav.verifyproject.models.ModelUser;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.ValueEventListener;

import java.util.ArrayList;
import java.util.List;


/**
 * A simple {@link Fragment} subclass.
 */
```

```
public class UsersFragment extends Fragment {

    RecyclerView recyclerView;
    AdapterUsers adapterUsers;
    List<ModelUser> userList;

    FirebaseAuth firebaseAuth;


    public UsersFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_users, container, false);

        firebaseAuth = FirebaseAuth.getInstance();

        recyclerView = view.findViewById(R.id.users_recyclerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));

        userList = new ArrayList<>();

        getAllUsers();
        return view;
    }

    private void getAllUsers() {
        final FirebaseUser fUser = FirebaseAuth.getInstance().getCurrentUser();
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Users");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                userList.clear();
                for(DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelUser modelUser = ds.getValue(ModelUser.class);
                    if(!modelUser.getUid().equals(fUser.getUid())) {
                        userList.add(modelUser);
                    }

                    adapterUsers = new AdapterUsers(getActivity(), userList);
                    recyclerView.setAdapter(adapterUsers);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void searchUsers(final String query) {
        final FirebaseUser fUser = FirebaseAuth.getInstance().getCurrentUser();
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Users");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                userList.clear();
                for(DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelUser modelUser = ds.getValue(ModelUser.class);
                    if(!modelUser.getUid().equals(fUser.getUid())) {

                        if(modelUser.getName().toLowerCase().contains(query.toLowerCase()) ||
                        modelUser.getEmail().toLowerCase().contains(query.toLowerCase())) {
                            userList.add(modelUser);
                        }
                        userList.add(modelUser);
                    }

                    adapterUsers = new AdapterUsers(getActivity(), userList);
                    adapterUsers.notifyDataSetChanged();
                    recyclerView.setAdapter(adapterUsers);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {

        }
        else {
            startActivity(new Intent(getActivity(),MainActivity.class));
            getActivity().finish();
        }
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        setHasOptionsMenu(true);
```

```
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu_main,menu);

        menu.findItem(R.id.action_add_post).setVisible(false);

        MenuItem item = menu.findItem(R.id.action_search);
        SearchView searchView = (SearchView) MenuItemCompat.getActionView(item);
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                if(!TextUtils.isEmpty(s.trim())) {
                    searchUsers(s);
                }
                else {
                    getAllUsers();
                }
                return false;
            }

            @Override
            public boolean onQueryTextChange(String s) {
                if(!TextUtils.isEmpty(s.trim())) {
                    searchUsers(s);
                }
                else {
                    getAllUsers();
                }
                return false;
            }
        });

        super.onCreateOptionsMenu(menu, inflater);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.action_logout) {
            firebaseAuth.signOut();
            checkUserStatus();
        }
        return super.onOptionsItemSelected(item);
    }

}
```

ChatlistFragment.java:

```java
package com.bhargav.verifyproject;


import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.view.MenuItemCompat;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.RecyclerView;

import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.SearchView;

import com.bhargav.verifyproject.adapters.AdapterChatlist;
import com.bhargav.verifyproject.models.ModelChat;
import com.bhargav.verifyproject.models.ModelChatlist;
import com.bhargav.verifyproject.models.ModelUser;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.ValueEventListener;

import java.util.ArrayList;
import java.util.List;


/**
 * A simple {@link Fragment} subclass.
 */
public class ChatListFragment extends Fragment {

    FirebaseAuth firebaseAuth;

    RecyclerView recyclerView;
    List<ModelChatlist> chatlistList;
    List<ModelUser> userList;
    DatabaseReference reference;
    FirebaseUser currentUser;
    AdapterChatlist adapterChatlist;
```

```
public ChatListFragment() {
    // Required empty public constructor
}


@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view  = inflater.inflate(R.layout.fragment_chat_list, container, false);

    firebaseAuth = FirebaseAuth.getInstance();

    currentUser = FirebaseAuth.getInstance().getCurrentUser();

    recyclerView = view.findViewById(R.id.recyclerView);

    chatlistList = new ArrayList<>();
    reference = FirebaseDatabase.getInstance().getReference("Chatlist").child(currentUser.getUid());
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            chatlistList.clear();
            for (DataSnapshot ds: dataSnapshot.getChildren()) {
                ModelChatlist chatlist = ds.getValue(ModelChatlist.class);
                chatlistList.add(chatlist);
            }
            loadChats();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });

    return view;
}

private void loadChats() {
    userList = new ArrayList<>();
    reference = FirebaseDatabase.getInstance().getReference("Users");
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            userList.clear();
            for (DataSnapshot ds: dataSnapshot.getChildren()) {
                ModelUser user = ds.getValue(ModelUser.class);
                for (ModelChatlist chatlist: chatlistList) {
                    if (user.getUid() != null && user.getUid().equals(chatlist.getId())) {
                        userList.add(user);
                        break;
                    }
                }

                adapterChatlist = new AdapterChatlist(getContext(), userList);
                recyclerView.setAdapter(adapterChatlist);
                for (int i=0; i<userList.size(); i++) {
                    lastMessage(userList.get(i).getUid());
                }
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}

private void lastMessage(final String userId) {
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Chats");
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            String theLastMessage = "default";
            for (DataSnapshot ds: dataSnapshot.getChildren()) {
                ModelChat chat = ds.getValue(ModelChat.class);
                if (chat==null) {
                    continue;
                }
                String sender = chat.getSender();
                String receiver = chat.getReceiver();
                if (sender == null || receiver == null) {
                    continue;
                }
                if (chat.getReceiver().equals(currentUser.getUid()) &&
                        chat.getSender().equals(userId) ||
                        chat.getReceiver().equals(userId) &&
                                chat.getSender().equals(currentUser.getUid())) {
                    theLastMessage = chat.getMessage();

                }
            }
            adapterChatlist.setLastMessageMap(userId, theLastMessage);
            adapterChatlist.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
    });
}

private void checkUserStatus() {
    FirebaseUser user = firebaseAuth.getCurrentUser();
    if(user != null) {

    }
    else {
        startActivity(new Intent(getActivity(),MainActivity.class));
        getActivity().finish();
    }
}

@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    setHasOptionsMenu(true);
    super.onCreate(savedInstanceState);
}

@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.menu_main,menu);

    menu.findItem(R.id.action_add_post).setVisible(false);

    super.onCreateOptionsMenu(menu, inflater);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    int id = item.getItemId();
    if(id == R.id.action_logout) {
        firebaseAuth.signOut();
        checkUserStatus();
    }
    return super.onOptionsItemSelected(item);
}

}
```

AddPostActivity.java:

```
package com.bhargav.verifyproject;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

import java.io.ByteArrayOutputStream;
import java.util.HashMap;

public class AddPostActivity extends AppCompatActivity {

    FirebaseAuth firebaseAuth;
    DatabaseReference userDbRef;

    ActionBar actionBar;
    private static final int CAMERA_REQUEST_CODE = 100;
    private static final int STORAGE_REQUEST_CODE = 200;

    private static final int IMAGE_PICK_CAMERA_CODE = 300;
    private static final int IMAGE_PICK_GALLERY_CODE = 400;

    String[] cameraPermissions;
    String[] storagePermissions;

    EditText titleEt, descriptionEt;
    ImageView imageIv;
    Button uploadBtn;

    String name, email, uid, dp;

    String editTitle, editDescription, editImage;

    Uri image_rui = null;

    ProgressDialog pd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_post);

        actionBar = getSupportActionBar();
        actionBar.setTitle("Add New Post");

        actionBar.setDisplayShowHomeEnabled(true);
        actionBar.setDisplayHomeAsUpEnabled(true);

        cameraPermissions = new String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE};
        storagePermissions = new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE};

        pd = new ProgressDialog(this);

        firebaseAuth = FirebaseAuth.getInstance();
        checkUserStatus();

        titleEt = findViewById(R.id.pTitleEt);
        descriptionEt = findViewById(R.id.pDescriptionEt);
        imageIv = findViewById(R.id.pImageIv);
        uploadBtn = findViewById(R.id.pUploadBtn);

        Intent intent = getIntent();
        final String isUpdateKey = ""+intent.getStringExtra("key");
        final String editPostId = ""+intent.getStringExtra("editPostId");

        if (isUpdateKey.equals("editPost")) {
            actionBar.setTitle("Update Post");
            uploadBtn.setText("Update");
            loadPostData(editPostId);
        }
        else {
            actionBar.setTitle("Add New Post");
            uploadBtn.setText("Upload");
        }

        actionBar.setSubtitle(email);

        userDbRef = FirebaseDatabase.getInstance().getReference("Users");
        Query query = userDbRef.orderByChild("email").equalTo(email);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    name = ""+ ds.child("name").getValue();
                    email = ""+ ds.child("email").getValue();
                    dp = ""+ ds.child("image").getValue();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

        imageIv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showImagePickDialog();
            }
        });

        uploadBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String title = titleEt.getText().toString().trim();
                String description = descriptionEt.getText().toString().trim();
                if(TextUtils.isEmpty(title)) {
                    Toast.makeText(AddPostActivity.this,"Enter Title...",Toast.LENGTH_SHORT).show();
                    return;
                }
                if(TextUtils.isEmpty(description)) {
                    Toast.makeText(AddPostActivity.this, "Enter
description...",Toast.LENGTH_SHORT).show();
                    return;
                }
                if (isUpdateKey.equals("editPost")) {
                    beginUpdate(title, description, editPostId);
                }
                else {
                    uploadData(title, description);
                }
            }
        });
    }

    private void beginUpdate(String title, String description, String editPostId) {
        pd.setMessage("Updating Post...");
        pd.show();

        if(!editImage.equals("noImage")) {
            updateWasWithImage(title, description, editPostId);
        }
        else if(imageIv.getDrawable() != null){
            updateWithNowImage(title, description, editPostId);
        }
        else {
            updateWithoutImage(title, description, editPostId);
        }
    }

    private void updateWithoutImage(String title, String description, String editPostId) {
        HashMap<String, Object> hashMap = new HashMap<>();

        hashMap.put("uid",uid);
        hashMap.put("uName",name);
        hashMap.put("uEmail",email);
        hashMap.put("uDp",dp);
        hashMap.put("pTitle",title);
        hashMap.put("pDescr",description);
        hashMap.put("pImage","noImage");

        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        ref.child(editPostId)
                .updateChildren(hashMap)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        pd.dismiss();
                        Toast.makeText(AddPostActivity.this,"Updated...",Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        pd.dismiss();
                        Toast.makeText(AddPostActivity.this,""+e.getMessage(),
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void updateWithNowImage(final String title, final String description, final String editPostId)
{

        String timeStamp = String.valueOf(System.currentTimeMillis());
        String filePathAndName = "Posts/"+ "post_"+timeStamp;

        Bitmap bitmap = ((BitmapDrawable)imageIv.getDrawable()).getBitmap();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.PNG,100,baos);
        byte[] data = baos.toByteArray();

        StorageReference ref = FirebaseStorage.getInstance().getReference().child(filePathAndName);
        ref.putBytes(data)
                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                    @Override
                    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
                        while (!uriTask.isSuccessful());

                        String downloadUri = uriTask.getResult().toString();
                        if (uriTask.isSuccessful()) {
                            HashMap<String, Object> hashMap = new HashMap<>();

                            hashMap.put("uid",uid);
                            hashMap.put("uName",name);
                            hashMap.put("uEmail",email);
                            hashMap.put("uDp",dp);
                            hashMap.put("pTitle",title);
                            hashMap.put("pDescr",description);
                            hashMap.put("pImage",downloadUri);

                            DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
                            ref.child(editPostId)
                                    .updateChildren(hashMap)
                                    .addOnSuccessListener(new OnSuccessListener<Void>() {
                                        @Override
                                        public void onSuccess(Void aVoid) {
                                            pd.dismiss();

Toast.makeText(AddPostActivity.this,"Updated...",Toast.LENGTH_SHORT).show();
                                        }
                                    })
                                    .addOnFailureListener(new OnFailureListener() {
                                        @Override
                                        public void onFailure(@NonNull Exception e) {
                                            pd.dismiss();
                                            Toast.makeText(AddPostActivity.this,""+e.getMessage(),
Toast.LENGTH_SHORT).show();
                                        }
                                    });
                        }
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        pd.dismiss();

Toast.makeText(AddPostActivity.this,""+e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void updateWasWithImage(final String title, final String description, final String editPostId) {
        StorageReference mPictureRef = FirebaseStorage.getInstance().getReferenceFromUrl(editImage);
        mPictureRef.delete()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        String timeStamp = String.valueOf(System.currentTimeMillis());
                        String filePathAndName = "Posts/"+ "post_"+timeStamp;

                        Bitmap bitmap = ((BitmapDrawable)imageIv.getDrawable()).getBitmap();
                        ByteArrayOutputStream baos = new ByteArrayOutputStream();
                        bitmap.compress(Bitmap.CompressFormat.PNG,100,baos);
                        byte[] data = baos.toByteArray();

                        StorageReference ref =
FirebaseStorage.getInstance().getReference().child(filePathAndName);
                        ref.putBytes(data)
                                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                                    @Override
                                    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                                        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
                                        while (!uriTask.isSuccessful());

                                        String downloadUri = uriTask.getResult().toString();
                                        if (uriTask.isSuccessful()) {
                                            HashMap<String, Object> hashMap = new HashMap<>();

                                            hashMap.put("uid",uid);
                                            hashMap.put("uName",name);
                                            hashMap.put("uEmail",email);
                                            hashMap.put("uDp",dp);
                                            hashMap.put("pTitle",title);
                                            hashMap.put("pDescr",description);
                                            hashMap.put("pImage",downloadUri);
                                            DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("Posts");
                                            ref.child(editPostId)
                                                    .updateChildren(hashMap)
                                                    .addOnSuccessListener(new OnSuccessListener<Void>() {
                                                        @Override
                                                        public void onSuccess(Void aVoid) {
                                                            pd.dismiss();

Toast.makeText(AddPostActivity.this,"Updated...",Toast.LENGTH_SHORT).show();
                                                        }
                                                    })
                                                    .addOnFailureListener(new OnFailureListener() {
                                                        @Override
                                                        public void onFailure(@NonNull Exception e) {
                                                            pd.dismiss();
                                                            Toast.makeText(AddPostActivity.this,""+e.getMessage(),
Toast.LENGTH_SHORT).show();
                                                        }
                                                    });
                                        }
                                    }
                                })
                                .addOnFailureListener(new OnFailureListener() {
                                    @Override
                                    public void onFailure(@NonNull Exception e) {
                                        pd.dismiss();

Toast.makeText(AddPostActivity.this,""+e.getMessage(),Toast.LENGTH_SHORT).show();
                                    }
                                });
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        pd.dismiss();

Toast.makeText(AddPostActivity.this,""+e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void loadPostData(String editPostId) {
        DatabaseReference reference = FirebaseDatabase.getInstance().getReference("Posts");

        Query fquery = reference.orderByChild("pId").equalTo(editPostId);
        fquery.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
```

```
            for (DataSnapshot ds: dataSnapshot.getChildren()) {
                editTitle = ""+ds.child("pTitle").getValue();
                editDescription = ""+ds.child("pDescr").getValue();
                editImage = ""+ds.child("pImage").getValue();

                titleEt.setText(editTitle);
                descriptionEt.setText(editDescription);

                if (!editImage.equals("noImage")) {
                    try {
                        Picasso.get().load(editImage).into(imageIv);
                    }
                    catch (Exception e) {

                    }
                }
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}

private void uploadData(final String title, final String description) {

    pd.setMessage("Publishing post...");
    pd.show();

    final String timeStamp = String.valueOf(System.currentTimeMillis());

    String filePathAndName = "Posts/" + "post_" + timeStamp;

    if (imageIv.getDrawable() != null) {

        Bitmap bitmap = ((BitmapDrawable)imageIv.getDrawable()).getBitmap();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.PNG,100,baos);
        byte[] data = baos.toByteArray();

        StorageReference ref = FirebaseStorage.getInstance().getReference().child(filePathAndName);
        ref.putBytes(data)
                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                    @Override
                    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
                        while (!uriTask.isSuccessful());
                        String downloadUri = uriTask.getResult().toString();
                        if(uriTask.isSuccessful()) {
                                HashMap<Object, String> hashMap = new HashMap<>();
                                hashMap.put("uid", uid);
                                hashMap.put("uName", name);
                                hashMap.put("uEmail", email);
                                hashMap.put("uDp", dp);
                                hashMap.put("pId", timeStamp);
                                hashMap.put("pTitle", title);
                                hashMap.put("pDescr", description);
                                hashMap.put("pImage", downloadUri);
                                hashMap.put("pTime", timeStamp);
                                hashMap.put("pLikes","0");
                                hashMap.put("pComments","0");

                                DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
                                ref.child(timeStamp).setValue(hashMap)
                                        .addOnSuccessListener(new OnSuccessListener<Void>() {
                                            @Override
                                            public void onSuccess(Void aVoid) {
                                                pd.dismiss();
                                                Toast.makeText(AddPostActivity.this,"Post
published",Toast.LENGTH_SHORT).show();
                                                titleEt.setText("");
                                                descriptionEt.setText("");
                                                imageIv.setImageURI(null);
                                                image_rui = null;
                                            }
                                        })
                                        .addOnFailureListener(new OnFailureListener() {
                                            @Override
                                            public void onFailure(@NonNull Exception e) {
                                                pd.dismiss();
                                                Toast.makeText(AddPostActivity.this, "" +
e.getMessage(),Toast.LENGTH_SHORT).show();
                                            }
                                        });
                            }
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            pd.dismiss();
                            Toast.makeText(AddPostActivity.this,"" +
e.getMessage(),Toast.LENGTH_SHORT).show();
                        }
                    });
        }
        else {
            HashMap<Object, String> hashMap = new HashMap<>();
            hashMap.put("uid", uid);
            hashMap.put("uName", name);
            hashMap.put("uEmail", email);
            hashMap.put("uDp", dp);
            hashMap.put("pId", timeStamp);
            hashMap.put("pTitle", title);
            hashMap.put("pDescr", description);
            hashMap.put("pImage", "noImage");
            hashMap.put("pTime", timeStamp);
            hashMap.put("pLikes","0");
            hashMap.put("pComments","0");

            DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
            ref.child(timeStamp).setValue(hashMap)
                    .addOnSuccessListener(new OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            pd.dismiss();
                            Toast.makeText(AddPostActivity.this,"Post
published",Toast.LENGTH_SHORT).show();
                            titleEt.setText("");
                            descriptionEt.setText("");
                            imageIv.setImageURI(null);
                            image_rui = null;
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            pd.dismiss();
                            Toast.makeText(AddPostActivity.this, "" +
e.getMessage(),Toast.LENGTH_SHORT).show();
                        }
                    });
        }
    }

    private void showImagePickDialog() {
        String[] options = {"Camera", "Gallery"};

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Choose Image from");

        builder.setItems(options, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                if(which==0){
                    if(!checkCameraPermission()) {
                        requestCameraPermission();
                    }
                    else {
                        pickFromCamera();
                    }
                }
                if(which==1){
                    if(!checkStoragePermission()) {
                        requestStoragePermission();
                    }
                    else {
                        pickFromGallery();
                    }
                }
            }
        });
        builder.create().show();
    }

    private void pickFromGallery() {

        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType("image/*");
        startActivityForResult(intent, IMAGE_PICK_GALLERY_CODE);
    }

    private void pickFromCamera() {

        ContentValues cv = new ContentValues();
        cv.put(MediaStore.Images.Media.TITLE,"Temp Pick");
        cv.put(MediaStore.Images.Media.DESCRIPTION,"Temp Descr");
        image_rui =
getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, cv);

        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, image_rui);
        startActivityForResult(intent, IMAGE_PICK_CAMERA_CODE);
    }

    private boolean checkStoragePermission() {
        boolean result = ContextCompat.checkSelfPermission(this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
        return result;
    }

    private void requestStoragePermission() {
        ActivityCompat.requestPermissions(this, storagePermissions, STORAGE_REQUEST_CODE);
    }

    private boolean checkCameraPermission() {
        boolean result = ContextCompat.checkSelfPermission(this,
                Manifest.permission.CAMERA) == (PackageManager.PERMISSION_GRANTED);
        boolean result1 = ContextCompat.checkSelfPermission(this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
```

```
    return result && result1;
}

private void requestCameraPermission() {
    ActivityCompat.requestPermissions(this, cameraPermissions, CAMERA_REQUEST_CODE);
}

@Override
protected void onStart() {
    super.onStart();
    checkUserStatus();
}

@Override
protected void onResume() {
    super.onResume();
    checkUserStatus();
}

private void checkUserStatus() {
    FirebaseUser user = firebaseAuth.getCurrentUser();
    if(user != null) {
        email = user.getEmail();
        uid = user.getUid();
    }
    else {
        startActivity(new Intent(AddPostActivity.this,MainActivity.class));
        finish();
    }
}

@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return super.onSupportNavigateUp();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main,menu);
    menu.findItem(R.id.action_add_post).setVisible(false);
    menu.findItem(R.id.action_search).setVisible(false);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    int id = item.getItemId();
    if(id == R.id.action_logout) {
        firebaseAuth.signOut();
```

```
            checkUserStatus();
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
            case CAMERA_REQUEST_CODE: {
                if (grantResults.length>0) {
                    boolean cameraAccepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                    boolean storageAccepted = grantResults[1] ==
PackageManager.PERMISSION_GRANTED;
                    if(cameraAccepted && storageAccepted) {
                        pickFromCamera();
                    }
                    else {
                        Toast.makeText(this, "Camera & Storage both permissions are neccessary...",
Toast.LENGTH_SHORT).show();
                    }
                }
                else {
                }
            }
            break;
            case STORAGE_REQUEST_CODE: {
                if(grantResults.length>0) {
                    boolean storageAccepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                    if(storageAccepted) {
                        pickFromGallery();
                    }
                    else {
                        Toast.makeText(this, "Storage permissions neccessary...",
Toast.LENGTH_SHORT).show();
                    }
                }
                else {

                }
            }
            break;
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        if(resultCode == RESULT_OK) {
            if(requestCode == IMAGE_PICK_GALLERY_CODE) {
                image_rui = data.getData();
                imageIv.setImageURI(image_rui);
            }
            else if(requestCode == IMAGE_PICK_CAMERA_CODE) {
                imageIv.setImageURI(image_rui);
            }
        }

        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

ChatActivity.java:

```
package com.bhargav.verifyproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.text.format.DateFormat;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bhargav.verifyproject.adapters.AdapterChat;
import com.bhargav.verifyproject.models.ModelChat;
import com.bhargav.verifyproject.models.ModelUser;
import com.bhargav.verifyproject.notifications.APIService;
import com.bhargav.verifyproject.notifications.Client;
import com.bhargav.verifyproject.notifications.Data;
import com.bhargav.verifyproject.notifications.Response;
import com.bhargav.verifyproject.notifications.Sender;
import com.bhargav.verifyproject.notifications.Token;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
```

```java
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.squareup.picasso.Picasso;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;

import retrofit2.Call;
import retrofit2.Callback;


public class ChatActivity extends AppCompatActivity {

    Toolbar toolbar;
    RecyclerView recyclerView;
    ImageView profieTv;
    TextView nameIv,userStatusTv;
    EditText messageEt;
    ImageButton sendBtn;

    FirebaseAuth firebaseAuth;

    FirebaseDatabase firebaseDatabase;
    DatabaseReference usersDbRef;

    ValueEventListener seenListener;
    DatabaseReference userRefForSeen;

    List<ModelChat> chatList;
    AdapterChat adapterChat;

    String hisUid;
    String myUid;
    String hisImage;

    APIService apiService;
    boolean notify = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_chat);

        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
```

```
toolbar.setTitle("");
recyclerView = findViewById(R.id.chat_recyclerView);
profieTv = findViewById(R.id.profileIv);
nameIv = findViewById(R.id.nameIv);
userStatusTv = findViewById(R.id.userStatusTv);
messageEt = findViewById(R.id.messageEt);
sendBtn = findViewById(R.id.sendBtn);

LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this);
linearLayoutManager.setStackFromEnd(true);

recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(linearLayoutManager);

apiService = Client.getRetrofit("https://fcm.googleapis.com/").create(APIService.class);

Intent intent = getIntent();
hisUid = intent.getStringExtra("hisUid");

firebaseAuth = FirebaseAuth.getInstance();

firebaseDatabase = FirebaseDatabase.getInstance();
usersDbRef = firebaseDatabase.getReference("Users");

Query userQuery = usersDbRef.orderByChild("uid").equalTo(hisUid);
userQuery.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for(DataSnapshot ds: dataSnapshot.getChildren()) {
            String name =""+ ds.child("name").getValue();
            hisImage =""+ ds.child("image").getValue();
            String typingStatus =""+ ds.child("typingTo").getValue();

            if(typingStatus.equals(myUid)) {
                userStatusTv.setText("typing...");
            }
            else {
                String onlineStatus = ""+ ds.child("onlineStatus").getValue();

                if(onlineStatus.equals("online")) {
                    userStatusTv.setText(onlineStatus);
                }
                else {
                    Calendar cal = Calendar.getInstance(Locale.ENGLISH);
                    cal.setTimeInMillis(Long.parseLong(onlineStatus));
                    String dateTime = DateFormat.format("dd/MM/yyyy hh:mm aa", cal).toString();
                    userStatusTv.setText("Last seen at: "+ dateTime);
                }
            }

            nameIv.setText(name);
```

```
                    try {
Picasso.get().load(hisImage).placeholder(R.drawable.ic_default_img_white).into(profieTv);
                    }
                    catch (Exception e) {
                        Picasso.get().load(R.drawable.ic_default_img_white).into(profieTv);

                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

        sendBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                notify = true;
                String message = messageEt.getText().toString().trim();
                if(TextUtils.isEmpty(message)) {
                    Toast.makeText(ChatActivity.this,"Cannot send the empty
message...",Toast.LENGTH_SHORT).show();
                }
                else {
                    sendMessage(message);
                }
                messageEt.setText("");
            }
        });

        messageEt.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if(s.toString().trim().length() == 0) {
                    checkTypingStatus("noOne");
                }
                else {
                    checkTypingStatus(hisUid);
                }
            }

            @Override
            public void afterTextChanged(Editable s) {
            }
        });

        readMessages();

        seenMessage();

    }

    private void seenMessage() {
        userRefForSeen = FirebaseDatabase.getInstance().getReference("Chats");
        seenListener = userRefForSeen.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelChat chat = ds.getValue(ModelChat.class);
                    if(chat.getReceiver().equals(myUid) && chat.getSender().equals(hisUid)) {
                        HashMap<String, Object> hasSeenHashMap = new HashMap<>();
                        hasSeenHashMap.put("isSeen", true);
                        ds.getRef().updateChildren(hasSeenHashMap);
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void readMessages() {
        chatList = new ArrayList<>();
        DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference("Chats");
        dbRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                chatList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelChat chat = ds.getValue(ModelChat.class);
                    if(chat.getReceiver().equals(myUid) && chat.getSender().equals(hisUid) ||
                            chat.getReceiver().equals(hisUid) && chat.getSender().equals(myUid)) {
                        chatList.add(chat);
                    }

                    adapterChat = new AdapterChat(ChatActivity.this,chatList,hisImage);
                    adapterChat.notifyDataSetChanged();
                    recyclerView.setAdapter(adapterChat);
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void sendMessage(final String message) {
        DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference();

        String timestamp = String.valueOf(System.currentTimeMillis());

        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("sender",myUid);
        hashMap.put("receiver", hisUid);
        hashMap.put("message", message);
        hashMap.put("timestamp", timestamp);
        hashMap.put("isSeen", false);
        databaseReference.child("Chats").push().setValue(hashMap);

        final DatabaseReference database =
FirebaseDatabase.getInstance().getReference("Users").child(myUid);
        database.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                ModelUser user = dataSnapshot.getValue(ModelUser.class);
                if(notify) {
                    senNotification(hisUid, user.getName(), message);
                }
                notify = false;
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

        final DatabaseReference chatRef1 = FirebaseDatabase.getInstance().getReference("Chatlist")
                .child(myUid)
                .child(hisUid);

        chatRef1.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (!dataSnapshot.exists()) {
                    chatRef1.child("id").setValue(hisUid);
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

        final DatabaseReference chatRef2 = FirebaseDatabase.getInstance().getReference("Chatlist")
                .child(hisUid)
                .child(myUid);

        chatRef2.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (!dataSnapshot.exists()) {
                    chatRef2 .child("id").setValue(myUid);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void senNotification(final String hisUid, final String name, final String message) {

        DatabaseReference allTokens = FirebaseDatabase.getInstance().getReference("Tokens");
        Query query = allTokens.orderByKey().equalTo(hisUid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    Token token = ds.getValue(Token.class);
                    Data data = new Data(myUid, name+":"+message, "New Message", hisUid,
R.drawable.ic_default_img);

                    Sender sender = new Sender(data, token.getToken());
                    apiService.sendNotification(sender)
                            .enqueue(new Callback<Response>() {
                                @Override
                                public void onResponse(Call<Response> call, retrofit2.Response<Response>
response) {
                                    Toast.makeText(ChatActivity.this,""+response.message(),
Toast.LENGTH_SHORT).show();
                                }

                                @Override
                                public void onFailure(Call<Response> call, Throwable t) {

                                }
                            });
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {
            myUid = user.getUid();
        }
        else {
            startActivity(new Intent(ChatActivity.this,MainActivity.class));
            finish();
        }
    }

    private void checkOnlineStatus(String status) {
        DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference("Users").child(myUid);
        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("onlineStatus", status);

        dbRef.updateChildren(hashMap);
    }

    private void checkTypingStatus(String typing) {
        DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference("Users").child(myUid);
        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("typingTo", typing);

        dbRef.updateChildren(hashMap);
    }

    @Override
    protected void onStart() {
        checkUserStatus();
        checkOnlineStatus("online");
        super.onStart();
    }

    @Override
    protected void onPause() {
        super.onPause();

        String timestamp = String.valueOf(System.currentTimeMillis());
        checkOnlineStatus(timestamp);
        checkTypingStatus("noOne");

        userRefForSeen.removeEventListener(seenListener);
    }

    @Override
    protected void onResume() {
        checkOnlineStatus("online");
        super.onResume();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main,menu);
        menu.findItem(R.id.action_search).setVisible(false);
        menu.findItem(R.id.action_add_post).setVisible(false);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {

        int id = item.getItemId();
        if(id == R.id.action_logout) {
            firebaseAuth.signOut();
            checkUserStatus();
        }

        return super.onOptionsItemSelected(item);
    }
}
```

PostDetailActivity.java:

```
package com.bhargav.verifyproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.PopupMenu;
import androidx.core.content.FileProvider;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.text.format.DateFormat;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.bhargav.verifyproject.adapters.AdapterComments;
import com.bhargav.verifyproject.models.ModelComment;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.squareup.picasso.Picasso;

import java.io.File;
import java.io.FileOutputStream;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;

public class PostDetailActivity extends AppCompatActivity {

    String hisUid,myUid,myEmail,myName,myDp,
    postId,pLikes,hisDp,hisName,pImage;

    boolean mProcessComment = false;
    boolean mProcessLike = false;

    ProgressDialog pd;

    ImageView uPictureIv, pImageIv;
    TextView uNameTv, pTimeTiv, pTitleTv, pDescriptionTv, pLikesTv, pCommentsTv;
```

```
ImageButton moreBtn;
Button likeBtn, shareBtn;
LinearLayout profileLayout;
RecyclerView recyclerView;

List<ModelComment> commentList;
AdapterComments adapterComments;

EditText commentEt;
ImageButton sendBtn;
ImageView cAvatarIv;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_post_detail);

    ActionBar actionBar = getSupportActionBar();
    actionBar.setTitle("Post Detail");
    actionBar.setDisplayShowHomeEnabled(true);
    actionBar.setDisplayHomeAsUpEnabled(true);

    Intent intent = getIntent();
    postId = intent.getStringExtra("postId");

    uPictureIv = findViewById(R.id.uPictureIv);
    pImageIv = findViewById(R.id.pImageIv);
    uNameTv = findViewById(R.id.uNameTv);
    pTimeTiv = findViewById(R.id.pTimeTv);
    pTitleTv = findViewById(R.id.pTitleTv);
    pDescriptionTv = findViewById(R.id.pDescriptionTv);
    pLikesTv = findViewById(R.id.pLikesTv);
    pCommentsTv = findViewById(R.id.pCommentsTv);
    moreBtn = findViewById(R.id.moreBtn);
    likeBtn = findViewById(R.id.likeBtn);
    shareBtn = findViewById(R.id.shareBtn);
    profileLayout = findViewById(R.id.profileLayout);
    recyclerView = findViewById(R.id.recyclerView);

    commentEt = findViewById(R.id.commentEt);
    sendBtn = findViewById(R.id.sendBtn);
    cAvatarIv = findViewById(R.id.cAvatarIv);

    loadPostInfo();

    checkUserStatus();

    loadUserInfo();

    setLikes();
```

```
        actionBar.setSubtitle("SignedIn as: " +myEmail);

        loadComments();

        sendBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                postComment();
            }
        });

        likeBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                likePost();
            }
        });

        moreBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showMoreOptions();
            }
        });

        shareBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String pTitle = pTitleTv.getText().toString().trim();
                String pDescription = pDescriptionTv.getText().toString().trim();

                BitmapDrawable bitmapDrawable = (BitmapDrawable)pImageIv.getDrawable();
                if (bitmapDrawable == null) {
                    shareTextOnly(pTitle, pDescription);
                }
                else {
                    Bitmap bitmap = bitmapDrawable.getBitmap();
                    shareImageAndText(pTitle, pDescription, bitmap);
                }
            }
        });
    }

    private void shareTextOnly(String pTitle, String pDescription) {
        String shareBody = pTitle +"\n"+ pDescription;

        Intent sIntent = new Intent(Intent.ACTION_SEND);
        sIntent.setType("text/plain");
        sIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject Here");
        sIntent.putExtra(Intent.EXTRA_TEXT, shareBody);
        startActivity(Intent.createChooser(sIntent, "Shaer Via"));
    }

    private void shareImageAndText(String pTitle, String pDescription, Bitmap bitmap) {
        String shareBody = pTitle +"\n"+ pDescription;

        Uri uri = saveImageToShare(bitmap);

        Intent sIntent = new Intent(Intent.ACTION_SEND);
        sIntent.putExtra(Intent.EXTRA_STREAM, uri);
        sIntent.putExtra(Intent.EXTRA_TEXT, shareBody);
        sIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject Here");
        sIntent.setType("image/png");
        startActivity(Intent.createChooser(sIntent, "Share Via"));
    }

    private Uri saveImageToShare(Bitmap bitmap) {
        File imageFolder = new File(getCacheDir(), "images");
        Uri uri = null;
        try {
            imageFolder.mkdirs();
            File file = new File(imageFolder, "shared_image.png");
            FileOutputStream stream = new FileOutputStream(file);

            bitmap.compress(Bitmap.CompressFormat.PNG,90, stream);
            stream.flush();
            stream.close();
            uri = FileProvider.getUriForFile(this,"com.bhargav.verifyproject.fileprovider",file);
        }
        catch (Exception e) {
            Toast.makeText(this, ""+e.getMessage(),Toast.LENGTH_SHORT).show();
        }
        return uri;
    }

    private void loadComments() {
        LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(layoutManager);
        commentList = new ArrayList<>();
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("Posts").child(postId).child("Comments");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                commentList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelComment modelComment = ds.getValue(ModelComment.class);
                    commentList.add(modelComment);
                    adapterComments = new AdapterComments(getApplicationContext(), commentList,
myUid, postId);
                    recyclerView.setAdapter(adapterComments);
                }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}

private void showMoreOptions() {
    PopupMenu popupMenu = new PopupMenu(this, moreBtn, Gravity.END);
    if (hisUid.equals(myUid)) {
        popupMenu.getMenu().add(Menu.NONE,0,0,"Delete");
        popupMenu.getMenu().add(Menu.NONE,1,0,"Edit");
    }
    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem menuItem) {
            int id = menuItem.getItemId();
            if (id==0) {
                beginDelete();
            }
            else if (id==1) {
                Intent intent = new Intent(PostDetailActivity.this, AddPostActivity.class);
                intent.putExtra("key","editPost");
                intent.putExtra("editPostId",postId);
                startActivity(intent);
            }
            return false;
        }
    });
    popupMenu.show();
}

private void beginDelete() {

    if (pImage.equals("noImage")) {
        deleteWithoutImage();
    }
    else {
        deleteWithImage();
    }
}

private void deleteWithImage() {
    final ProgressDialog pd = new ProgressDialog(this);
    pd.setMessage("Deleting...");
    StorageReference picRef = FirebaseStorage.getInstance().getReferenceFromUrl(pImage);
    picRef.delete()
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
```

```
                    public void onSuccess(Void aVoid) {
                        Query fquery =
FirebaseDatabase.getInstance().getReference("Posts").orderByChild("pId").equalTo(postId);
                        fquery.addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                    ds.getRef().removeValue();
                                }
                                Toast.makeText(PostDetailActivity.this, "Deleted
Successfully",Toast.LENGTH_SHORT).show();
                                pd.dismiss();
                            }

                            @Override
                            public void onCancelled(@NonNull DatabaseError databaseError) {

                            }
                        });
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        pd.dismiss();
                        Toast.makeText(PostDetailActivity.this,""+e.getMessage(),
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void deleteWithoutImage() {

        final ProgressDialog pd = new ProgressDialog(this);
        pd.setMessage("Deleting...");

        Query fquery =
FirebaseDatabase.getInstance().getReference("Posts").orderByChild("pId").equalTo(postId);
        fquery.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ds.getRef().removeValue();
                }
                Toast.makeText(PostDetailActivity.this, "Deleted
Successfully",Toast.LENGTH_SHORT).show();
                pd.dismiss();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
            }
        });
    }

    private void setLikes() {
        final DatabaseReference likesRef = FirebaseDatabase.getInstance().getReference().child("Likes");

        likesRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (dataSnapshot.child(postId).hasChild(myUid)) {
                    likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_liked,0,0,0);
                    likeBtn.setText("Accepted");
                }
                else {
                    likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_accept_black,0,0,0);
                    likeBtn.setText("Accept");
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void likePost() {

        mProcessLike = true;
        final DatabaseReference likesRef = FirebaseDatabase.getInstance().getReference().child("Likes");
        final DatabaseReference postsRef = FirebaseDatabase.getInstance().getReference().child("Posts");
        likesRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (mProcessLike) {
                    if (dataSnapshot.child(postId).hasChild(myUid)) {
                        postsRef.child(postId).child("pLikes").setValue(""+(Integer.parseInt(pLikes)-1));
                        likesRef.child(postId).child(myUid).removeValue();
                        mProcessLike = false;
                    }
                    else {
                        postsRef.child(postId).child("pLikes").setValue(""+(Integer.parseInt(pLikes)+1));
                        likesRef.child(postId).child(myUid).setValue("Accepted");
                        mProcessLike = false;
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void postComment() {
        pd = new ProgressDialog(this);
        pd.setMessage("Adding Comment...");
        String comment = commentEt.getText().toString().trim();
        if (TextUtils.isEmpty(comment)) {
            Toast.makeText(this, "Comment is empty...",Toast.LENGTH_SHORT).show();
            return;
        }

        String timeStamp = String.valueOf(System.currentTimeMillis());
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("Posts").child(postId).child("Comments");

        HashMap<String, Object> hashMap = new HashMap<>();

        hashMap.put("cId", timeStamp);
        hashMap.put("comment", comment);
        hashMap.put("timestamp", timeStamp);
        hashMap.put("uid", myUid);
        hashMap.put("uEmail", myEmail);
        hashMap.put("uDp", myDp);
        hashMap.put("uName", myName);

        ref.child(timeStamp).setValue(hashMap)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        pd.dismiss();
                        Toast.makeText(PostDetailActivity.this,"Comment Added...",
Toast.LENGTH_SHORT).show();
                        commentEt.setText("");
                        updateCommentCount();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        pd.dismiss();

Toast.makeText(PostDetailActivity.this,""+e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void updateCommentCount() {
        mProcessComment = true;
        final DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts").child(postId);
        ref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (mProcessComment) {
                    String comments = ""+ dataSnapshot.child("pComments").getValue();
                    int newCommentVal = Integer.parseInt(comments) + 1;
                    ref.child("pComments").setValue(""+newCommentVal);
                    mProcessComment = false;
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void loadUserInfo() {
        Query myRef = FirebaseDatabase.getInstance().getReference("Users");
        myRef.orderByChild("uid").equalTo(myUid).addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    myName = ""+ds.child("name").getValue();
                    myDp = ""+ds.child("image").getValue();

                    try {
                        Picasso.get().load(myDp).placeholder(R.drawable.ic_default_img).into(cAvatarIv);
                    }
                    catch(Exception e) {
                        Picasso.get().load(R.drawable.ic_default_img).into(cAvatarIv);
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void loadPostInfo() {
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        Query query = ref.orderByChild("pId").equalTo(postId);
        query.addValueEventListener(new ValueEventListener() {
```

```
@Override
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
    for (DataSnapshot ds: dataSnapshot.getChildren()) {
        String pTitle = ""+ds.child("pTitle").getValue();
        String pDescr = ""+ds.child("pDescr").getValue();
        pLikes = ""+ds.child("pLikes").getValue();
        String pTimeStamp = ""+ds.child("pTime").getValue();
        pImage = ""+ds.child("pImage").getValue();
        hisDp = ""+ds.child("uDp").getValue();
        hisUid = ""+ds.child("uid").getValue();
        String uEmail = ""+ds.child("uEmail").getValue();
        hisName = ""+ds.child("uName").getValue();
        String commentCount = ""+ds.child("pComments").getValue();

        Calendar calendar = Calendar.getInstance(Locale.getDefault());
        calendar.setTimeInMillis(Long.parseLong(pTimeStamp));
        String pTime = DateFormat.format("dd/MM/yyyy hh:mm aa", calendar).toString();

        pTitleTv.setText(pTitle);
        pDescriptionTv.setText(pDescr);
        pLikesTv.setText(pLikes + " Accept");
        pTimeTiv.setText(pTime);
        pCommentsTv.setText(commentCount +" Comments");
        uNameTv.setText(hisName);

        if (pImage.equals("noImage")) {
            pImageIv.setVisibility(View.GONE);
        }
        else {
            pImageIv.setVisibility(View.VISIBLE);
            try {
                Picasso.get().load(pImage).into(pImageIv);
            }
            catch (Exception e) {

            }
        }

        try {
            Picasso.get().load(hisDp).placeholder(R.drawable.ic_default_img).into(uPictureIv);
        }
        catch (Exception e) {
            Picasso.get().load(R.drawable.ic_default_img).into(uPictureIv);
        }



    }
}

@Override
```

```
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void checkUserStatus() {
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        if (user!= null) {
            myEmail = user.getEmail();
            myUid = user.getUid();
        }
        else {
            startActivity(new Intent(PostDetailActivity.this,MainActivity.class));
            finish();
        }
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main,menu);
        menu.findItem(R.id.action_add_post).setVisible(false);
        menu.findItem(R.id.action_search).setVisible(false);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_logout) {
            FirebaseAuth.getInstance().signOut();
            checkUserStatus();
        }
        return super.onOptionsItemSelected(item);
    }
}
```

ThereProfileActivity.java:

```
package com.bhargav.verifyproject;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.MenuItemCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.SearchView;
import android.widget.TextView;
import android.widget.Toast;

import com.bhargav.verifyproject.adapters.AdapterPosts;
import com.bhargav.verifyproject.models.ModelPost;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.squareup.picasso.Picasso;

import java.util.ArrayList;
import java.util.List;

public class ThereProfileActivity extends AppCompatActivity {

    FirebaseAuth firebaseAuth;

    ImageView avatarIv,coverIv;
    TextView nameTv,emailTv,phoneTv;
    RecyclerView postRecyclerView;

    List<ModelPost> postList;
    AdapterPosts adapterPosts;
    String uid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_there_profile);

        ActionBar actionBar = getSupportActionBar();
        actionBar.setTitle("Profile");
        actionBar.setDisplayShowHomeEnabled(true);
        actionBar.setDisplayHomeAsUpEnabled(true);
        avatarIv = findViewById(R.id.avatarIv);
        coverIv = findViewById(R.id.coverIv);
        nameTv = findViewById(R.id.nameTv);
        emailTv = findViewById(R.id.emailTv);
        phoneTv = findViewById(R.id.phoneTv);

        postRecyclerView = findViewById(R.id.recyclerview_posts);

        firebaseAuth = FirebaseAuth.getInstance();

        Intent intent = getIntent();
        uid = intent.getStringExtra("uid");

        Query query =
FirebaseDatabase.getInstance().getReference("Users").orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot ds : dataSnapshot.getChildren()) {
                    String name = ""+ds.child("name").getValue();
                    String email = ""+ds.child("email").getValue();
                    String phone = ""+ds.child("phone").getValue();
                    String image = ""+ds.child("image").getValue();
                    String cover = ""+ds.child("cover").getValue();

                    nameTv.setText(name);
                    emailTv.setText(email);
                    phoneTv.setText(phone);

                    try {
                        Picasso.get().load(image).into(avatarIv);
                    }
                    catch(Exception e) {
                        Picasso.get().load(R.drawable.ic_default_img_white).into(avatarIv);
                    }

                    try {
                        Picasso.get().load(cover).into(coverIv);
                    }
                    catch(Exception e) {

                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
        postList = new ArrayList<>();

        checkUserStatus();
        loadHistPosts();

    }

    private void loadHistPosts() {
        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
        layoutManager.setStackFromEnd(true);
        layoutManager.setReverseLayout(true);
        postRecyclerView.setLayoutManager(layoutManager);
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        Query query = ref.orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost myPosts = ds.getValue(ModelPost.class);
                    postList.add(myPosts);
                    adapterPosts = new AdapterPosts(ThereProfileActivity.this, postList);
                    postRecyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(ThereProfileActivity.this,""+databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void searchHistPosts(final String searchQuery) {
        LinearLayoutManager layoutManager = new LinearLayoutManager(ThereProfileActivity.this);
        layoutManager.setStackFromEnd(true);
        layoutManager.setReverseLayout(true);
        postRecyclerView.setLayoutManager(layoutManager);
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts");
        Query query = ref.orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                postList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ModelPost myPosts = ds.getValue(ModelPost.class);

                    if(myPosts.getpTitle().toLowerCase().contains(searchQuery.toLowerCase()) ||
                            myPosts.getpDescr().toLowerCase().contains(searchQuery.toLowerCase())) {
                        postList.add(myPosts);
                    }

                    postList.add(myPosts);
                    adapterPosts = new AdapterPosts(ThereProfileActivity.this, postList);
                    postRecyclerView.setAdapter(adapterPosts);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {
                Toast.makeText(ThereProfileActivity.this,""+databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void checkUserStatus() {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if(user != null) {
        }
        else {
            startActivity(new Intent(ThereProfileActivity.this,MainActivity.class));
            finish();
        }
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return super.onSupportNavigateUp();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        menu.findItem(R.id.action_add_post).setVisible(true);

        MenuItem item = menu.findItem(R.id.action_search);
        SearchView searchView = (SearchView) MenuItemCompat.getActionView(item);
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchHistPosts(s);
                }
                else {
                    loadHistPosts();
                }
                return false;
            }
            @Override
            public boolean onQueryTextChange(String s) {
                if(!TextUtils.isEmpty(s)) {
                    searchHistPosts(s);
                }
                else {
                    loadHistPosts();
                }
                return false;
            }
        });

        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.action_logout) {
            firebaseAuth.signOut();
            checkUserStatus();
        }
        return super.onOptionsItemSelected(item);
    }

}
```

Adapters in Java Folder (.java):

AdapterChat.java:

```
package com.bhargav.verifyproject.adapters;

import android.content.Context;
import android.content.DialogInterface;
import android.text.format.DateFormat;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.recyclerview.widget.RecyclerView;

import com.bhargav.verifyproject.R;
import com.bhargav.verifyproject.models.ModelChat;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
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.squareup.picasso.Picasso;

import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;

public class AdapterChat extends RecyclerView.Adapter<AdapterChat.MyHolder> {

    private static final int MSG_TYPE_LEFT = 0;
    private static final int MSG_TYPE_RIGHT = 1;
    Context context;
    List<ModelChat> chatList;
    String imageUrl;

    FirebaseUser fUser;

    public AdapterChat(Context context, List<ModelChat> chatList, String imageUrl) {
        this.context = context;
        this.chatList = chatList;
        this.imageUrl = imageUrl;
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        if(i==MSG_TYPE_RIGHT) {
            View view = LayoutInflater.from(context).inflate(R.layout.row_chat_right, viewGroup, false);
            return new MyHolder(view);
        }
        else {
            View view = LayoutInflater.from(context).inflate(R.layout.row_chat_left, viewGroup, false);
            return new MyHolder(view);
        }
    }

    @Override
    public void onBindViewHolder(@NonNull MyHolder myHolder, final int i) {
        String message = chatList.get(i).getMessage();
        String timeStamp = chatList.get(i).getTimestamp();

        Calendar cal = Calendar.getInstance(Locale.ENGLISH);
        cal.setTimeInMillis(Long.parseLong(timeStamp));
        String dateTime = DateFormat.format("dd/MM/yyyy hh:mm aa", cal).toString();

        myHolder.messageTv.setText(message);
        myHolder.timeTv.setText(dateTime);
        try {
            Picasso.get().load(imageUrl).into(myHolder.profileIv);
        }
        catch(Exception e) {

        }

        myHolder.messageLAyout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AlertDialog.Builder builder = new AlertDialog.Builder(context);
                builder.setTitle("Delete");
                builder.setMessage("Are you sure to delete message?");
                builder.setPositiveButton("Delete", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        deleteMessage(i);
                    }
                });
                builder.setNegativeButton("No", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                    }
                });
                builder.create().show();
            }
        });

        if (i==chatList.size()-1) {
            if (chatList.get(i).isSeen()) {
                myHolder.isSeenTv.setText("Seen");
            }
            else {
                myHolder.isSeenTv.setText("Delivered");
            }
        }
        else {
            myHolder.isSeenTv.setVisibility(View.GONE);
        }
    }

    private void deleteMessage(int position) {
        final String myUID = FirebaseAuth.getInstance().getCurrentUser().getUid();
        String msgTimeStamp = chatList.get(position).getTimestamp();
        DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference("Chats");
        Query query = dbRef.orderByChild("timestamp").equalTo(msgTimeStamp);
        query.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot ds: dataSnapshot.getChildren()){
                    if(ds.child("sender").getValue().equals(myUID)){
                        //ds.getRef().removeValue();
                        HashMap<String, Object> hashMap = new HashMap<>();
                        hashMap.put("message","This message was deleted...");
                        ds.getRef().updateChildren(hashMap);
                        Toast.makeText(context,"message deleted...",Toast.LENGTH_SHORT).show();
                    }
                    else {
                        Toast.makeText(context,"You can delete only your
messages...",Toast.LENGTH_SHORT).show();
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    @Override
    public int getItemCount() {
        return chatList.size();
    }

    @Override
    public int getItemViewType(int position) {
        fUser = FirebaseAuth.getInstance().getCurrentUser();
        if(chatList.get(position).getSender().equals(fUser.getUid())) {
            return MSG_TYPE_RIGHT;
        }
        else {
            return MSG_TYPE_LEFT;
        }
    }

    class MyHolder extends RecyclerView.ViewHolder {

        ImageView profileIv;
        TextView messageTv, timeTv, isSeenTv;
        LinearLayout messageLAyout;

        public MyHolder(@NonNull View itemView) {
            super(itemView);

            profileIv = itemView.findViewById(R.id.profileIv);
            messageTv = itemView.findViewById(R.id.messageTv);
            timeTv = itemView.findViewById(R.id.timeTv);
            isSeenTv = itemView.findViewById(R.id.isSeenTv);
            messageLAyout = itemView.findViewById(R.id.messageLayout);
        }
    }
}
```

AdapterChatlist.java:

```
package com.bhargav.verifyproject.adapters;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bhargav.verifyproject.ChatActivity;
import com.bhargav.verifyproject.R;
import com.bhargav.verifyproject.models.ModelUser;
import com.squareup.picasso.Picasso;

import java.util.HashMap;
import java.util.List;

public class AdapterChatlist extends RecyclerView.Adapter<AdapterChatlist.MyHolder> {

    Context context;
    List<ModelUser> userList;
    private HashMap<String , String > lastMessageMap;

    public AdapterChatlist(Context context, List<ModelUser> userList) {
        this.context = context;
        this.userList = userList;
        lastMessageMap = new HashMap<>();
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View view = LayoutInflater.from(context).inflate(R.layout.row_chatlist, viewGroup, false);
        return new MyHolder(view);
    }
```

```
@Override
public void onBindViewHolder(@NonNull MyHolder myHolder, int i) {

    final String hisUid = userList.get(i).getUid();
    String userImage = userList.get(i).getImage();
    String userName = userList.get(i).getName();
    String lastMessage = lastMessageMap.get(hisUid);

    myHolder.nameTv.setText(userName);
    if (lastMessage==null || lastMessage.equals("default")) {
        myHolder.lastMessageTv.setVisibility(View.GONE);
    }
    else {
        myHolder.lastMessageTv.setVisibility(View.VISIBLE);
        myHolder.lastMessageTv.setText(lastMessage);
    }

    try {
        Picasso.get().load(userImage).placeholder(R.drawable.ic_default_img).into(myHolder.profileIv);
    }
    catch(Exception e) {
        Picasso.get().load(R.drawable.ic_default_img).into(myHolder.profileIv);
    }

    if (userList.get(i).getOnlineStatus().equals("online")) {
        myHolder.onlineStatusIv.setImageResource(R.drawable.circle_online);
    }
    else {
        myHolder.onlineStatusIv.setImageResource(R.drawable.circle_offline);
    }

    myHolder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(context, ChatActivity.class);
            intent.putExtra("hisUid", hisUid);
            context.startActivity(intent);
        }
    });
}

public void setLastMessageMap(String userId, String lastMessage) {
    lastMessageMap.put(userId, lastMessage);
}

@Override
public int getItemCount() {
    return userList.size();
}

class MyHolder extends RecyclerView.ViewHolder {
        ImageView profileIv, onlineStatusIv;
        TextView nameTv, lastMessageTv;

        public MyHolder(@NonNull View itemView) {
            super(itemView);

            profileIv = itemView.findViewById(R.id.profileIv);
            onlineStatusIv = itemView.findViewById(R.id.onlineStatusIv);
            nameTv = itemView.findViewById(R.id.nameTv);
            lastMessageTv = itemView.findViewById(R.id.lastMessageTv);
        }
    }
}
```

AdapterComments.java:

```
package com.bhargav.verifyproject.adapters;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.text.format.DateFormat;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bhargav.verifyproject.R;
import com.bhargav.verifyproject.models.ModelComment;
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.ValueEventListener;
import com.squareup.picasso.Picasso;

import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class AdapterComments extends RecyclerView.Adapter<AdapterComments.MyHolder> {

    Context context;
    List<ModelComment> commentList;
    String myUid,postId;

    public AdapterComments(Context context, List<ModelComment> commentList, String myUid, String
postId) {
        this.context = context;
        this.commentList = commentList;
        this.myUid = myUid;
        this.postId = postId;
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {

        View view = LayoutInflater.from(context).inflate(R.layout.row_comments, viewGroup,false);
        return new MyHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyHolder myHolder, int i) {
        final String uid = commentList.get(i).getUid();
        String name = commentList.get(i).getuName();
        String email = commentList.get(i).getuEmail();
        String image = commentList.get(i).getuDp();
        final String cid = commentList.get(i).getcId();
        String comment = commentList.get(i).getComment();
        String timestamp = commentList.get(i).getTimestamp();

        Calendar calendar = Calendar.getInstance(Locale.getDefault());
        calendar.setTimeInMillis(Long.parseLong(timestamp));
        String pTime = DateFormat.format("dd/MM/yyyy hh:mm aa", calendar).toString();

        myHolder.nameTv.setText(name);
        myHolder.commentTv.setText(comment);
        myHolder.timeTv.setText(pTime);

        try {
            Picasso.get().load(image).placeholder(R.drawable.ic_default_img).into(myHolder.avatarIv);
        }
        catch(Exception e) {}

        myHolder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (myUid.equals(uid)) {
                    AlertDialog.Builder builder = new AlertDialog.Builder(v.getRootView().getContext());
                    builder.setTitle("Delete");
                    builder.setMessage("Are you sure to delete this comment?");
                    builder.setPositiveButton("Delete", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            deleteComment(cid);
                        }
                    });
                    builder.setNegativeButton("Cacnel", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    });
                    builder.create().show();
                }
                else {
                    Toast.makeText(context,"Can't delete Others's comment...",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    private void deleteComment(String cid) {

        final DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Posts").child(postId);
        ref.child("Comments").child(cid).removeValue();

        ref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                String comments = ""+ dataSnapshot.child("pComments").getValue();
                int newCommentVal = Integer.parseInt(comments) - 1;
                ref.child("pComments").setValue(""+newCommentVal);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });

    }

    @Override
    public int getItemCount() {
        return commentList.size();
    }

    class MyHolder extends RecyclerView.ViewHolder {
        ImageView avatarIv;
        TextView nameTv, commentTv, timeTv;

        public MyHolder(@NonNull View itemView) {
            super(itemView);
            avatarIv = itemView.findViewById(R.id.avatarIv);
            nameTv = itemView.findViewById(R.id.nameTv);
            commentTv = itemView.findViewById(R.id.commentTv);
            timeTv = itemView.findViewById(R.id.timeTv);
        }
    }
}
```

AdapterPosts.java:

```
package com.bhargav.verifyproject.adapters;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.text.format.DateFormat;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.widget.PopupMenu;
import androidx.core.content.FileProvider;
import androidx.recyclerview.widget.RecyclerView;

import com.bhargav.verifyproject.AddPostActivity;
import com.bhargav.verifyproject.PostDetailActivity;
import com.bhargav.verifyproject.R;
import com.bhargav.verifyproject.ThereProfileActivity;
import com.bhargav.verifyproject.models.ModelPost;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
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;
```

```java
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.squareup.picasso.Picasso;

import java.io.File;
import java.io.FileOutputStream;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class AdapterPosts extends RecyclerView.Adapter<AdapterPosts.MyHolder> {

    Context context;
    List<ModelPost> postList;

    String myUid;

    private DatabaseReference likesRef;
    private DatabaseReference postsRef;

    boolean mProcessLike = false;

    public AdapterPosts(Context context, List<ModelPost> postList) {
        this.context = context;
        this.postList = postList;
        myUid = FirebaseAuth.getInstance().getCurrentUser().getUid();
        likesRef = FirebaseDatabase.getInstance().getReference().child("Likes");
        postsRef = FirebaseDatabase.getInstance().getReference().child("Posts");
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View view = LayoutInflater.from(context).inflate(R.layout.row_posts, viewGroup, false);
        return new MyHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull final MyHolder myHolder, final int i) {

        final String uid = postList.get(i).getUid();
        String uEmail = postList.get(i).getuEmail();
        String uName = postList.get(i).getuName();
        String uDp = postList.get(i).getuDp();
        final String pId = postList.get(i).getpId();
        final String pTitle = postList.get(i).getpTitle();
        final String pDescription = postList.get(i).getpDescr();
        final String pImage = postList.get(i).getpImage();
        String pTimeStamp = postList.get(i).getpTime();
        String pLikes = postList.get(i).getpLikes();
        String pComments = postList.get(i).getpComments();

        Calendar calendar = Calendar.getInstance(Locale.getDefault());
        calendar.setTimeInMillis(Long.parseLong(pTimeStamp));
        String pTime = DateFormat.format("dd/MM/yyyy hh:mm aa", calendar).toString();

        myHolder.uNameTv.setText(uName);
        myHolder.pTimeTv.setText(pTime);
        myHolder.pTitleTv.setText(pTitle);
        myHolder.pDescriptionTv.setText(pDescription);
        myHolder.pLikesTv.setText(pLikes +" Accept");
        myHolder.pCommentsTv.setText(pComments + " Comments");

        setLikes(myHolder, pId);

        try {
            Picasso.get().load(uDp).placeholder(R.drawable.ic_default_img).into(myHolder.uPictureIv);
        }
        catch (Exception e) {

        }

        if (pImage.equals("noImage")) {
            myHolder.pImageIv.setVisibility(View.GONE);
        }
        else {
            myHolder.pImageIv.setVisibility(View.VISIBLE);
            try {
                Picasso.get().load(pImage).into(myHolder.pImageIv);
            }
            catch (Exception e) {

            }
        }

        myHolder.moreBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showMoreOptions(myHolder.moreBtn, uid, myUid, pId, pImage);
            }
        });

        myHolder.likeBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final int pLikes = Integer.parseInt(postList.get(i).getpLikes());
                mProcessLike = true;
                final String postIde = postList.get(i).getpId();
                likesRef.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        if (mProcessLike) {
                            if (dataSnapshot.child(postIde).hasChild(myUid)) {
                                postsRef.child(postIde).child("pLikes").setValue(""+(pLikes-1));
                                likesRef.child(postIde).child(myUid).removeValue();
                                mProcessLike = false;
                            }
                            else {
                                postsRef.child(postIde).child("pLikes").setValue(""+(pLikes+1));
                                likesRef.child(postIde).child(myUid).setValue("Liked");
                                mProcessLike = false;
                            }
                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });
            }
        });

        myHolder.commentBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context, PostDetailActivity.class);
                intent.putExtra("postId",pId);
                context.startActivity(intent);
            }
        });

        myHolder.shareBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                BitmapDrawable bitmapDrawable = (BitmapDrawable)myHolder.pImageIv.getDrawable();
                if (bitmapDrawable == null) {
                    shareTextOnly(pTitle, pDescription);
                }
                else {
                    Bitmap bitmap = bitmapDrawable.getBitmap();
                    shareImageAndText(pTitle, pDescription, bitmap);
                }
            }
        });

        myHolder.profileLayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context, ThereProfileActivity.class);
                intent.putExtra("uid",uid);
                context.startActivity(intent);
                }
            });

        }

        private void shareTextOnly(String pTitle, String pDescription) {
            String shareBody = pTitle +"\n"+ pDescription;

            Intent sIntent = new Intent(Intent.ACTION_SEND);
            sIntent.setType("text/plain");
            sIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject Here");
            sIntent.putExtra(Intent.EXTRA_TEXT, shareBody);
            context.startActivity(Intent.createChooser(sIntent, "Shaer Via"));
        }

        private void shareImageAndText(String pTitle, String pDescription, Bitmap bitmap) {
            String shareBody = pTitle +"\n"+ pDescription;

            Uri uri = saveImageToShare(bitmap);

            Intent sIntent = new Intent(Intent.ACTION_SEND);
            sIntent.putExtra(Intent.EXTRA_STREAM, uri);
            sIntent.putExtra(Intent.EXTRA_TEXT, shareBody);
            sIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject Here");
            sIntent.setType("image/png");
            context.startActivity(Intent.createChooser(sIntent, "Share Via"));
        }

        private Uri saveImageToShare(Bitmap bitmap) {
            File imageFolder = new File(context.getCacheDir(), "images");
            Uri uri = null;
            try {
                imageFolder.mkdirs();
                File file = new File(imageFolder, "shared_image.png");
                FileOutputStream stream = new FileOutputStream(file);

                bitmap.compress(Bitmap.CompressFormat.PNG,90, stream);
                stream.flush();
                stream.close();
                uri = FileProvider.getUriForFile(context,"com.bhargav.verifyproject.fileprovider",file);
            }
            catch (Exception e) {
                Toast.makeText(context, ""+e.getMessage(),Toast.LENGTH_SHORT).show();
            }
            return uri;
        }

        private void setLikes(final MyHolder holder, final String postKey) {
            likesRef.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                    if (dataSnapshot.child(postKey).hasChild(myUid)) {
                        holder.likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_liked,0,0,0);
                        holder.likeBtn.setText("Accepted");
                    }
                    else {

    holder.likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_accept_black,0,0,0);
                        holder.likeBtn.setText("Accept");
                    }
                }

                @Override
                public void onCancelled(@NonNull DatabaseError databaseError) {

                }
            });
        }

        private void showMoreOptions(ImageButton moreBtn, String uid, String myUid, final String pId,
    final String pImage) {
            PopupMenu popupMenu = new PopupMenu(context, moreBtn, Gravity.END);
            if (uid.equals(myUid)) {
                popupMenu.getMenu().add(Menu.NONE,0,0,"Delete");
                popupMenu.getMenu().add(Menu.NONE,1,0,"Edit");
            }
            popupMenu.getMenu().add(Menu.NONE,2,0,"View Detail");
            popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                @Override
                public boolean onMenuItemClick(MenuItem menuItem) {
                    int id = menuItem.getItemId();
                    if (id==0) {
                        beginDelete(pId, pImage);
                    }
                    else if (id==1) {
                        Intent intent = new Intent(context, AddPostActivity.class);
                        intent.putExtra("key","editPost");
                        intent.putExtra("editPostId",pId);
                        context.startActivity(intent);
                    }
                    else if(id==2) {
                        Intent intent = new Intent(context, PostDetailActivity.class);
                        intent.putExtra("postId",pId);
                        context.startActivity(intent);
                    }
                    return false;
                }
            });
            popupMenu.show();
        }

        private void beginDelete(String pId, String pImage) {
            if (pImage.equals("noImage")) {
                deleteWithoutImage(pId);
            }
            else {
                deleteWithImage(pId, pImage);
            }
        }

        private void deleteWithImage(final String pId, String pImage) {
            final ProgressDialog pd = new ProgressDialog(context);
            pd.setMessage("Deleting...");
            StorageReference picRef = FirebaseStorage.getInstance().getReferenceFromUrl(pImage);
            picRef.delete()
                    .addOnSuccessListener(new OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            Query fquery =
    FirebaseDatabase.getInstance().getReference("Posts").orderByChild("pId").equalTo(pId);
                            fquery.addListenerForSingleValueEvent(new ValueEventListener() {
                                @Override
                                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                    for (DataSnapshot ds: dataSnapshot.getChildren()) {
                                        ds.getRef().removeValue();
                                    }
                                    Toast.makeText(context, "Deleted Successfully",Toast.LENGTH_SHORT).show();
                                    pd.dismiss();
                                }

                                @Override
                                public void onCancelled(@NonNull DatabaseError databaseError) {

                                }
                            });
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            pd.dismiss();
                            Toast.makeText(context,""+e.getMessage(), Toast.LENGTH_SHORT).show();
                        }
                    });
        }

        private void deleteWithoutImage(String pId) {

            final ProgressDialog pd = new ProgressDialog(context);
            pd.setMessage("Deleting...");

            Query fquery =
    FirebaseDatabase.getInstance().getReference("Posts").orderByChild("pId").equalTo(pId);
```

```
        fquery.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    ds.getRef().removeValue();
                }
                Toast.makeText(context, "Deleted Successfully",Toast.LENGTH_SHORT).show();
                pd.dismiss();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    @Override
    public int getItemCount() {
        return postList.size();
    }

    class MyHolder extends RecyclerView.ViewHolder {

        ImageView uPictureIv, pImageIv;
        TextView uNameTv, pTimeTv, pTitleTv, pDescriptionTv, pLikesTv, pCommentsTv;
        ImageButton moreBtn;
        Button likeBtn, commentBtn, shareBtn;
        LinearLayout profileLayout;

        public MyHolder(@NonNull View itemView) {
            super(itemView);

            uPictureIv = itemView.findViewById(R.id.uPictureIv);
            pImageIv = itemView.findViewById(R.id.pImageIv);
            uNameTv = itemView.findViewById(R.id.uNameTv);
            pTimeTv = itemView.findViewById(R.id.pTimeTv);
            pTitleTv = itemView.findViewById(R.id.pTitleTv);
            pDescriptionTv = itemView.findViewById(R.id.pDescriptionTv);
            pLikesTv = itemView.findViewById(R.id.pLikesTv);
            pCommentsTv = itemView.findViewById(R.id.pCommentsTv);
            moreBtn = itemView.findViewById(R.id.moreBtn);
            likeBtn = itemView.findViewById(R.id.likeBtn);
            commentBtn = itemView.findViewById(R.id.commentBtn);
            shareBtn = itemView.findViewById(R.id.shareBtn);
            profileLayout = itemView.findViewById(R.id.profileLayout);
        }
    }
}
```

AdapterUsers.java:

```java
package com.bhargav.verifyproject.adapters;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bhargav.verifyproject.ChatActivity;
import com.bhargav.verifyproject.R;
import com.bhargav.verifyproject.ThereProfileActivity;
import com.bhargav.verifyproject.models.ModelUser;
import com.squareup.picasso.Picasso;

import java.util.List;

public class AdapterUsers extends RecyclerView.Adapter<AdapterUsers.MyHolder>{

    Context context;
    List<ModelUser> userList;

    public AdapterUsers(Context context, List<ModelUser> userList) {
        this.context = context;
        this.userList = userList;
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {

        View view = LayoutInflater.from(context).inflate(R.layout.row_users, viewGroup,false);
        return new MyHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyHolder myHolder, int i) {

        final String hisUID = userList.get(i).getUid();
        String userImage = userList.get(i).getImage();
        String userName = userList.get(i).getName();
        final String userEmail = userList.get(i).getEmail();

        myHolder.mNameTv.setText(userName);
        myHolder.mEmailTv.setText(userEmail);
        try {
            Picasso.get().load(userImage)
                    .placeholder(R.drawable.ic_default_img)
                    .into(myHolder.mAvatarIv);
        }
        catch(Exception e) {

        }

        myHolder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                AlertDialog.Builder builder = new AlertDialog.Builder(context);
                builder.setItems(new String[]{"Profile", "Chat"}, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        if(which==0) {
                            Intent intent = new Intent(context, ThereProfileActivity.class);
                            intent.putExtra("uid",hisUID);
                            context.startActivity(intent);
                        }
                        if(which==1) {
                            Intent intent = new Intent(context, ChatActivity.class);
                            intent.putExtra("hisUid", hisUID);
                            context.startActivity(intent);
                        }

                    }
                });
                builder.create().show();
            }
        });

    }

    @Override
    public int getItemCount() {
        return userList.size();
    }

    class MyHolder extends RecyclerView.ViewHolder {

        ImageView mAvatarIv;
        TextView mNameTv, mEmailTv;

        public MyHolder(@NonNull View itemView) {
            super(itemView);

            mAvatarIv = itemView.findViewById(R.id.avatarIv);
            mNameTv = itemView.findViewById(R.id.nameTv);
            mEmailTv = itemView.findViewById(R.id.emailTv);
        }
    }
}
```

Models in Java Folder (.java):

ModelChat.java:

```
package com.bhargav.verifyproject.models;

public class ModelChat {

    String message, receiver, sender, timestamp;
    boolean isSeen;

    public ModelChat() {
    }

    public ModelChat(String message, String receiver, String sender, String timestamp, boolean isSeen) {
        this.message = message;
        this.receiver = receiver;
        this.sender = sender;
        this.timestamp = timestamp;
        this.isSeen = isSeen;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getReceiver() {
        return receiver;
    }

    public void setReceiver(String receiver) {
        this.receiver = receiver;
    }

    public String getSender() {
        return sender;
    }

    public void setSender(String sender) {
        this.sender = sender;
```

```java
    }

    public String getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(String timestamp) {
        this.timestamp = timestamp;
    }

    public boolean isSeen() {
        return isSeen;
    }

    public void setSeen(boolean seen) {
        isSeen = seen;
    }
}
```

ModelChatlist.java:

```java
package com.bhargav.verifyproject.models;

public class ModelChatlist {

    String id;

    public ModelChatlist() {
    }

    public ModelChatlist(String id) {
        this.id = id;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
}
```

ModelComment.java:

```java
package com.bhargav.verifyproject.models;

public class ModelComment {

    String cId, comment, timestamp, uid, uEmail, uDp, uName;
    public ModelComment() {
    }

    public ModelComment(String cId, String comment, String timestamp, String uid, String uEmail,
String uDp, String uName) {
        this.cId = cId;
        this.comment = comment;
        this.timestamp = timestamp;
        this.uid = uid;
        this.uEmail = uEmail;
        this.uDp = uDp;
        this.uName = uName;
    }

    public String getcId() {
        return cId;
    }

    public void setcId(String cId) {
        this.cId = cId;
    }

    public String getComment() {
        return comment;
    }

    public void setComment(String comment) {
        this.comment = comment;
    }

    public String getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(String timestamp) {
        this.timestamp = timestamp;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }

    public String getuEmail() {
        return uEmail;
    }
    public void setuEmail(String uEmail) {
        this.uEmail = uEmail;
    }

    public String getuDp() {
        return uDp;
    }

    public void setuDp(String uDp) {
        this.uDp = uDp;
    }

    public String getuName() {
        return uName;
    }

    public void setuName(String uName) {
        this.uName = uName;
    }
}
```

ModelPost.java:

```
package com.bhargav.verifyproject.models;

public class ModelPost {

    String pId, pTitle, pDescr, pLikes,pComments, pImage, pTime, uid, uEmail, uDp, uName;

    public ModelPost() {
    }

    public ModelPost(String pId, String pTitle, String pDescr, String pLikes, String pComments, String
pImage, String pTime, String uid, String uEmail, String uDp, String uName) {
        this.pId = pId;
        this.pTitle = pTitle;
        this.pDescr = pDescr;
        this.pLikes = pLikes;
        this.pComments = pComments;
        this.pImage = pImage;
        this.pTime = pTime;
        this.uid = uid;
        this.uEmail = uEmail;
        this.uDp = uDp;
        this.uName = uName;
    }

    public String getpId() {
        return pId;
    }
```

```
public void setpId(String pId) {
    this.pId = pId;
}

public String getpTitle() {
    return pTitle;
}

public void setpTitle(String pTitle) {
    this.pTitle = pTitle;
}

public String getpDescr() {
    return pDescr;
}

public void setpDescr(String pDescr) {
    this.pDescr = pDescr;
}

public String getpLikes() {
    return pLikes;
}

public void setpLikes(String pLikes) {
    this.pLikes = pLikes;
}

public String getpComments() {
    return pComments;
}

public void setpComments(String pComments) {
    this.pComments = pComments;
}

public String getpImage() {
    return pImage;
}

public void setpImage(String pImage) {
    this.pImage = pImage;
}

public String getpTime() {
    return pTime;
}

public void setpTime(String pTime) {
    this.pTime = pTime;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }

    public String getuEmail() {
        return uEmail;
    }

    public void setuEmail(String uEmail) {
        this.uEmail = uEmail;
    }

    public String getuDp() {
        return uDp;
    }

    public void setuDp(String uDp) {
        this.uDp = uDp;
    }

    public String getuName() {
        return uName;
    }

    public void setuName(String uName) {
        this.uName = uName;
    }
}
```

ModelUser.java:

```
package com.bhargav.verifyproject.models;

public class ModelUser {

    String name, email, search, phone, image, cover, uid, onlineStatus, typingTo;

    public ModelUser() {
    }

    public ModelUser(String name, String email, String search, String phone, String image, String cover,
String uid, String onlineStatus, String typingTo) {
        this.name = name;
        this.email = email;
    this.search = search;
    this.phone = phone;
    this.image = image;
    this.cover = cover;
    this.uid = uid;
    this.onlineStatus = onlineStatus;
    this.typingTo = typingTo;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getSearch() {
    return search;
}

public void setSearch(String search) {
    this.search = search;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public String getImage() {
    return image;
}

public void setImage(String image) {
    this.image = image;
}

public String getCover() {
    return cover;
    }

    public void setCover(String cover) {
        this.cover = cover;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }

    public String getOnlineStatus() {
        return onlineStatus;
    }

    public void setOnlineStatus(String onlineStatus) {
        this.onlineStatus = onlineStatus;
    }

    public String getTypingTo() {
        return typingTo;
    }

    public void setTypingTo(String typingTo) {
        this.typingTo = typingTo;
    }
}
```

Notifications in Java Folder (.java):

APIService.java (Interface):

```
package com.bhargav.verifyproject.notifications;

import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.Headers;
import retrofit2.http.POST;

public interface APIService {

    @Headers({
            "Content-Type:application/json",

"Authorization:key=AAAAxwMsWFE:APA91bEeThcontIWyGzXBd__aCZsBiwBSoWYcscUihH
QEzWP_-QeJunW-c_WcbuFSwNFEpHZAsVVcMJASkzwJ-
J2cSIq6CgsgY6QD09Bgaz6Z0T1D_c7nOgkO9h3JpZtwMWTZxNVj9DU"
    })
    @POST("fcm/send")
    Call<Response> sendNotification(@Body Sender body);
}
```

Client.java:

```
package com.bhargav.verifyproject.notifications;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class Client {

    private static Retrofit retrofit = null;

    public static Retrofit getRetrofit(String url) {
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                    .baseUrl(url)
                    .addConverterFactory(GsonConverterFactory.create())
                    .build();
        }
        return retrofit;
    }
}
```

Data.java:

```
package com.bhargav.verifyproject.notifications;

public class Data {

    private String user, body, title, sent;
    private Integer icon;

    public Data() {
    }

    public Data(String user, String body, String title, String sent, Integer icon) {
        this.user = user;
        this.body = body;
        this.title = title;
        this.sent = sent;
        this.icon = icon;
    }

    public String getUser() {
        return user;
    }
    public void setUser(String user) {
        this.user = user;
    }

    public String getBody() {
        return body;
    }

    public void setBody(String body) {
        this.body = body;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getSent() {
        return sent;
    }

    public void setSent(String sent) {
        this.sent = sent;
    }

    public Integer getIcon() {
        return icon;
    }

    public void setIcon(Integer icon) {
        this.icon = icon;
    }
}
```

FirebaseMessaging.java:

```
package com.bhargav.verifyproject.notifications;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;

import androidx.core.app.NotificationCompat;

import com.bhargav.verifyproject.ChatActivity;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

public class FirebaseMessaging extends FirebaseMessagingService {

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);

        SharedPreferences sp = getSharedPreferences("SP_USER", MODE_PRIVATE);
        String savedCurrentUser = sp.getString("Current_USERID", "None");
        String sent = remoteMessage.getData().get("sent");
        String user = remoteMessage.getData().get("user");
        FirebaseUser fUser = FirebaseAuth.getInstance().getCurrentUser();
        if (fUser != null && sent.equals(fUser.getUid())) {
            if (!savedCurrentUser.equals(user)) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                    sendOAndAboveNotifications(remoteMessage);
                }
                else {
                    sendNormalNotification(remoteMessage);
                }
            }
        }
    }

    private void sendNormalNotification(RemoteMessage remoteMessage) {
        String user = remoteMessage.getData().get("user");
        String icon = remoteMessage.getData().get("icon");
        String title = remoteMessage.getData().get("title");
        String body = remoteMessage.getData().get("body");

        RemoteMessage.Notification notification = remoteMessage.getNotification();
        int i = Integer.parseInt(user.replaceAll("[\\D]",""));
        Intent intent = new Intent(this, ChatActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("hisUid", user);
        intent.putExtras(bundle);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pIntent = PendingIntent.getActivity(this, i, intent,
PendingIntent.FLAG_ONE_SHOT);

        Uri defSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
                .setSmallIcon(Integer.parseInt(icon))
                .setContentText(body)
                .setContentTitle(title)
                .setAutoCancel(true)
                .setSound(defSoundUri)
                .setContentIntent(pIntent);

        NotificationManager notificationManager =
(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
        int j = 0;
        if (i>0) {
            j = i;
        }
        notificationManager.notify(j,builder.build());
    }

    private void sendOAndAboveNotifications(RemoteMessage remoteMessage) {
        String user = remoteMessage.getData().get("user");
        String icon = remoteMessage.getData().get("icon");
        String title = remoteMessage.getData().get("title");
        String body = remoteMessage.getData().get("body");

        RemoteMessage.Notification notification = remoteMessage.getNotification();
        int i = Integer.parseInt(user.replaceAll("[\\D]",""));
        Intent intent = new Intent(this, ChatActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("hisUid", user);
        intent.putExtras(bundle);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pIntent = PendingIntent.getActivity(this, i, intent,
PendingIntent.FLAG_ONE_SHOT);

        Uri defSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);

        OreoAndAboveNotification notification1 = new OreoAndAboveNotification(this);
        Notification.Builder builder = notification1.getONotifications(title,body,pIntent, defSoundUri,icon);

        int j = 0;
        if (i>0) {
            j = i;
        }
        notification1.getManager().notify(j,builder.build());
    }
}
```

FirebaseService.java:

```
package com.bhargav.verifyproject.notifications;
```

```
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

public class FirebaseService extends FirebaseInstanceIdService {

    @Override
    public void onTokenRefresh() {
        super.onTokenRefresh();
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        String tokenRefresh = FirebaseInstanceId.getInstance().getToken();
        if (user != null) {
            updateToken(tokenRefresh);
        }
    }

    private void updateToken(String tokenRefresh) {
        FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Tokens");
        Token token = new Token(tokenRefresh);
        ref.child(user.getUid()).setValue(token);
    }
}
```

OreoAndAboveNotification.java:

```
package com.bhargav.verifyproject.notifications;

import android.annotation.TargetApi;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.ContextWrapper;
import android.net.Uri;
import android.os.Build;

public class OreoAndAboveNotification extends ContextWrapper {

    private static final String ID = "some_id";
    private static final String NAME = "VerifyProject";

    private NotificationManager notificationManager;

    public OreoAndAboveNotification(Context base) {
        super(base);
        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.O) {
            createChannel();
        }
    }

    @TargetApi(Build.VERSION_CODES.O)
    private void createChannel() {
        NotificationChannel notificationChannel = new NotificationChannel(ID, NAME,
NotificationManager.IMPORTANCE_DEFAULT);
        notificationChannel.enableLights(true);
        notificationChannel.enableVibration(true);
        notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PRIVATE);
        getManager().createNotificationChannel(notificationChannel);
    }

    public NotificationManager getManager() {
        if(notificationManager == null) {
            notificationManager =
(NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
        }
        return notificationManager;
    }

    @TargetApi(Build.VERSION_CODES.O)
    public Notification.Builder getONotifications(String title,
                                                  String body,
                                                  PendingIntent pIntennt,
                                                  Uri soundUri,
                                                  String icon) {
        return new Notification.Builder(getApplicationContext(), ID)
                .setContentIntent(pIntennt)
                .setContentTitle(title)
                .setContentText(body)
                .setSound(soundUri)
                .setAutoCancel(true)
                .setSmallIcon(Integer.parseInt(icon));
    }
}
```

Response.java:

```
package com.bhargav.verifyproject.notifications;

public class Response {

    private String success;
}
```

Sender.java:

```
package com.bhargav.verifyproject.notifications;

public class Sender {

    private Data data;
    private String to;

    public Sender() {
    }

    public Sender(Data data, String to) {
        this.data = data;
        this.to = to;
    }

    public Data getData() {
        return data;
    }

    public void setData(Data data) {
        this.data = data;
    }

    public String getTo() {
        return to;
    }

    public void setTo(String to) {
        this.to = to;
    }
}
```

Token.java:

```
package com.bhargav.verifyproject.notifications;

public class Token {

    String token;

    public Token() {
    }

    public Token(String token) {
        this.token = token;
    }

    public String getToken() {
        return token;
    }
    public void setToken(String token) {
        this.token = token;
    }
}
```