

KH.Eftekharul Haque

ID: 162-15-8201

Project: chatapp (whats app clone)

## All codes:

### MainActivity.java:

```
public class MainActivity extends AppCompatActivity {

    CircleImageView profile_image;
    TextView username;

    FirebaseUser firebaseUser;
    DatabaseReference reference;

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("");

        profile_image = findViewById(R.id.profile_image);
        username = findViewById(R.id.username);

        firebaseUser = FirebaseAuth.getInstance().getCurrentUser();
        reference =
        FirebaseDatabase.getInstance().getReference("Users").child(firebaseUser.getId());

        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                User user = dataSnapshot.getValue(User.class);
                username.setText(user.getUsername());
                if (user.getImageURL().equals("default")){
                    profile_image.setImageResource(R.mipmap.ic_launcher);
                } else {

                    //change this

                    Glide.with(getApplicationContext()).load(user.getImageURL()).into(profile_image);
                }
            }
        })
    }
}
```

```

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

        }
    });

    final TabLayout tabLayout = findViewById(R.id.tab_layout);
    final ViewPager viewPager = findViewById(R.id.view_pager);

    reference = FirebaseDatabase.getInstance().getReference("Chats");
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            ViewPagerAdapter viewPagerAdapter = new
ViewPagerAdapter(getSupportFragmentManager());
            int unread = 0;
            for (DataSnapshot snapshot : dataSnapshot.getChildren()){
                Chat chat = snapshot.getValue(Chat.class);
                if (chat.getReceiver().equals(firebaseUser.getUid()) &&
!chat.isIsseen()){
                    unread++;
                }
            }

            if (unread == 0){
                viewPagerAdapter.addFragment(new ChatsFragment(), "Chats");
            } else {
                viewPagerAdapter.addFragment(new ChatsFragment(), "("+unread+"
Chats");
            }

            viewPagerAdapter.addFragment(new UsersFragment(), "Users");
            viewPagerAdapter.addFragment(new ProfileFragment(), "Profile");

            viewPager.setAdapter(viewPagerAdapter);

            tabLayout.setupWithViewPager(viewPager);

        }

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

        }
    });

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

```

```

    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){

            case R.id.Logout:
                FirebaseAuth.getInstance().signOut();
                // change this code beacuse your app will crash
                startActivity(new Intent(MainActivity.this,
StartActivity.class)).setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP));
                return true;
            }

            return false;
        }

        class ViewPagerAdapter extends FragmentPagerAdapter {

            private ArrayList<Fragment> fragments;
            private ArrayList<String> titles;

            ViewPagerAdapter(FragmentManager fm){
                super(fm);
                this.fragments = new ArrayList<>();
                this.titles = new ArrayList<>();
            }

            @Override
            public Fragment getItem(int position) {
                return fragments.get(position);
            }

            @Override
            public int getCount() {
                return fragments.size();
            }

            public void addFragment(Fragment fragment, String title){
                fragments.add(fragment);
                titles.add(title);
            }

            // Ctrl + O

            @Nullable
            @Override
            public CharSequence getPageTitle(int position) {
                return titles.get(position);
            }
        }

        private void status(String status){
            reference =
            FirebaseDatabase.getInstance().getReference("Users").child(firebaseUser.getUid());

```

```

        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("status", status);

        reference.updateChildren(hashMap);
    }

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

    @Override
    protected void onPause() {
        super.onPause();
        status("offline");
    }
}

```

## LoginActivity.java:

```

public class LoginActivity extends AppCompatActivity {

    MaterialEditText email, password;
    Button btn_login;

    FirebaseAuth auth;
    TextView forgot_password;

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Login");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        auth = FirebaseAuth.getInstance();

        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
        btn_login = findViewById(R.id.btn_login);
        forgot_password = findViewById(R.id.forgot_password);

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

```



```

ImageButton btn_send;
EditText text_send;

MessageAdapter messageAdapter;
List<Chat> mchat;

RecyclerView recyclerView;

Intent intent;

ValueEventListener seenListener;

String userid;

APIService apiService;

boolean notify = false;

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

    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setTitle("");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    toolbar.setNavigationOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // and this
            startActivity(new Intent(MessageActivity.this,
MainActivity.class).setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP));
        }
    });

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

    recyclerView = findViewById(R.id.recycler_view);
    recyclerView.setHasFixedSize(true);
    LinearLayoutManager layoutManager = new
LinearLayoutManager(getApplicationContext());
    layoutManager.setStackFromEnd(true);
    recyclerView.setLayoutManager(layoutManager);

    profile_image = findViewById(R.id.profile_image);
    username = findViewById(R.id.username);
    btn_send = findViewById(R.id.btn_send);
    text_send = findViewById(R.id.text_send);

    intent = getIntent();
    userid = intent.getStringExtra("userid");
    fuser = FirebaseAuth.getInstance().getCurrentUser();

```

```

        btn_send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                notify = true;
                String msg = text_send.getText().toString();
                if (!msg.equals("")){
                    sendMessage(fuser.getUid(), userid, msg);
                } else {
                    Toast.makeText(MessageActivity.this, "You can't send empty
message", Toast.LENGTH_SHORT).show();
                }
                text_send.setText("");
            }
        });

        reference =
        FirebaseDatabase.getInstance().getReference("Users").child(userid);

        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                User user = dataSnapshot.getValue(User.class);
                username.setText(user.getUsername());
                if (user.getImageURL().equals("default")){
                    profile_image.setImageResource(R.mipmap.ic_launcher);
                } else {
                    //and this
                }

                Glide.with(getApplicationContext()).load(user.getImageURL()).into(profile_image);
            }

            readMesagges(fuser.getUid(), userid, user.getImageURL());
        }

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

        }
    });

    seenMessage(userid);
}

private void seenMessage(final String userid){
    reference = FirebaseDatabase.getInstance().getReference("Chats");
    seenListener = reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()){
                Chat chat = snapshot.getValue(Chat.class);
                if (chat.getReceiver().equals(fuser.getUid()) &&
chat.getSender().equals(userid)){
                    HashMap<String, Object> hashMap = new HashMap<>();

```

```

        hashMap.put("isseen", true);
        snapshot.getRef().updateChildren(hashMap);
    }
}

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

});
}

private void sendMessage(String sender, final String receiver, String message){

    DatabaseReference reference = FirebaseDatabase.getInstance().getReference();

    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("sender", sender);
    hashMap.put("receiver", receiver);
    hashMap.put("message", message);
    hashMap.put("isseen", false);

    reference.child("Chats").push().setValue(hashMap);

    // add user to chat fragment
    final DatabaseReference chatRef =
    FirebaseDatabase.getInstance().getReference("Chatlist")
        .child(fuser.getUid())
        .child(userid);

    chatRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (!dataSnapshot.exists()){
                chatRef.child("id").setValue(userid);
            }
        }

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

    final DatabaseReference chatRefReceiver =
    FirebaseDatabase.getInstance().getReference("Chatlist")
        .child(userid)
        .child(fuser.getUid());
    chatRefReceiver.child("id").setValue(fuser.getUid());

    final String msg = message;

    reference =

```



```

FirebaseDatabase.getInstance().getReference("Users").child(fuser.getId());
reference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        User user = dataSnapshot.getValue(User.class);
        if (notify) {
            sendNotifiaction(receiver, user.getUsername(), msg);
        }
        notify = false;
    }

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

    }
});
}

private void sendNotifiaction(String receiver, final String username, final
String message){
    DatabaseReference tokens =
FirebaseDatabase.getInstance().getReference("Tokens");
    Query query = tokens.orderByKey().equalTo(receiver);
    query.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()){
                Token token = snapshot.getValue(Token.class);
                Data data = new Data(fuser.getId(), R.mipmap.ic_launcher,
username+": "+message, "New Message",
userid);

                Sender sender = new Sender(data, token.getToken());

                apiService.sendNotification(sender)
                    .enqueue(new Callback<MyResponse>() {
                        @Override
                        public void onResponse(Call<MyResponse> call,
Response<MyResponse> response) {
                            if (response.code() == 200){
                                if (response.body().success != 1){
                                    Toast.makeText(MessageActivity.this,
"Failed!", Toast.LENGTH_SHORT).show();
                                }
                            }
                        }

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

                        }
                    });
            }
        }
    });
}
}

```

```

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

        }

    });
}

private void readMesagges(final String myid, final String userid, final String
imageurl){
    mchat = new ArrayList<>();

    reference = FirebaseDatabase.getInstance().getReference("Chats");
    reference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            mchat.clear();
            for (DataSnapshot snapshot : dataSnapshot.getChildren()){
                Chat chat = snapshot.getValue(Chat.class);
                if (chat.getReceiver().equals(myid) &&
chat.getSender().equals(userid) ||
                chat.getReceiver().equals(userid) &&
chat.getSender().equals(myid)){
                    mchat.add(chat);
                }

                messageAdapter = new MessageAdapter(MessageActivity.this, mchat,
imageurl);
                recyclerView.setAdapter(messageAdapter);
            }
        }

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

        }

    });
}

private void currentUser(String userid){
    SharedPreferences.Editor editor = getSharedPreferences("PREFS",
MODE_PRIVATE).edit();
    editor.putString("currentuser", userid);
    editor.apply();
}

private void status(String status){
    reference =
FirebaseDatabase.getInstance().getReference("Users").child(fuser.getId());

    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("status", status);

    reference.updateChildren(hashMap);
}

```

```

@Override
protected void onResume() {
    super.onResume();
    status("online");
    currentUser(userid);
}

@Override
protected void onPause() {
    super.onPause();
    reference.removeEventListener(seenListener);
    status("offline");
    currentUser("none");
}
}

```

## RegisterActivity:

```

public class RegisterActivity extends AppCompatActivity {

    MaterialEditText username, email, password;
    Button btn_register;

    FirebaseAuth auth;
    DatabaseReference reference;

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Register");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        username = findViewById(R.id.username);
        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
        btn_register = findViewById(R.id.btn_register);

        auth = FirebaseAuth.getInstance();

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

```

```

        String txt_username = username.getText().toString();
        String txt_email = email.getText().toString();
        String txt_password = password.getText().toString();

        if (TextUtils.isEmpty(txt_username) || TextUtils.isEmpty(txt_email)
|| TextUtils.isEmpty(txt_password)){
            Toast.makeText(RegisterActivity.this, "All fields are required",
Toast.LENGTH_SHORT).show();
        } else if (txt_password.length() < 6 ){
            Toast.makeText(RegisterActivity.this, "password must be at least
6 characters", Toast.LENGTH_SHORT).show();
        } else {
            register(txt_username, txt_email, txt_password);
        }
    }
});
}

private void register(final String username, String email, String password){

    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    FirebaseUser firebaseUser = auth.getCurrentUser();
                    assert firebaseUser != null;
                    String userid = firebaseUser.getId();

                    reference =
FirebaseDatabase.getInstance().getReference("Users").child(userid);

                    HashMap<String, String> hashMap = new HashMap<>();
                    hashMap.put("id", userid);
                    hashMap.put("username", username);
                    hashMap.put("imageUrl", "default");
                    hashMap.put("status", "offline");
                    hashMap.put("search", username.toLowerCase());

                    reference.setValue(hashMap).addOnCompleteListener(new
OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()){
                                Intent intent = new
Intent(RegisterActivity.this, MainActivity.class);

                                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
                                startActivity(intent);
                                finish();
                            }
                        }
                    });
                } else {
                    Toast.makeText(RegisterActivity.this, "You can't register

```

```

        with this email or password", Toast.LENGTH_SHORT).show();
    }
    });
}
}

```

## ResetPasswordActivity:

```

public class ResetPasswordActivity extends AppCompatActivity {

    EditText send_email;
    Button btn_reset;

    FirebaseAuth firebaseAuth;

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Reset Password");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        send_email = findViewById(R.id.send_email);
        btn_reset = findViewById(R.id.btn_reset);

        firebaseAuth = FirebaseAuth.getInstance();

        btn_reset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String email = send_email.getText().toString();

                if (email.equals("")){
                    Toast.makeText(ResetPasswordActivity.this, "All fields are
required!", Toast.LENGTH_SHORT).show();
                } else {

                    firebaseAuth.sendPasswordResetEmail(email).addOnCompleteListener(new
OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()){
                                Toast.makeText(ResetPasswordActivity.this, "Please
check you Email", Toast.LENGTH_SHORT).show();
                                startActivity(new Intent(ResetPasswordActivity.this,

```

```

LoginActivity.class));
                                } else {
                                    String error = task.getException().getMessage();
                                    Toast.makeText(ResetPasswordActivity.this, error,
Toast.LENGTH_SHORT).show();
                                }
                            }
                        });
                    }
                });
            }
        }
    }
}

```

## StartActivity:

```
public class StartActivity extends AppCompatActivity {

    Button login, register;

    FirebaseAuth firebaseUser;

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

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

        //check if user is null
        if (firebaseUser != null){
            Intent intent = new Intent(StartActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
    }

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

        login = findViewById(R.id.login);
        register = findViewById(R.id.register);

        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(StartActivity.this, LoginActivity.class));
            }
        });
    }
}
```

```
        }  
    });  
  
    register.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            startActivity(new Intent(StartActivity.this,  
RegisterActivity.class));  
        }  
    });  
}  
}
```