```java
package com.example.covid_19alertapp.activities;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.widget.Toolbar;

import com.example.covid_19alertapp.R;
import com.example.covid_19alertapp.extras.Constants;
import com.example.covid_19alertapp.models.UserInfoData;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;
import com.google.firebase.auth.UserInfo;
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 static com.example.covid_19alertapp.activities.SignUpActivity.verification;


public class VerificationPageActivity extends AppCompatActivity {

    Toolbar toolbar;
    Button homeButton,confirmButton,editNumberButton;
    TextView textViewResendOTP;
    EditText digit1,digit2,digit3,digit4,digit5,digit6;
    String verificationCode,uid;
    FirebaseAuth auth;
    SharedPreferences sp,userInfoCheck,signUpSp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_verification_page);



        homeButton = findViewById(R.id.home_button_verification_page);
        toolbar = findViewById(R.id.verification_page_toolbar);

        digit1=findViewById(R.id.editTextDigit1);
        digit2=findViewById(R.id.editTextDigit2);
        digit3=findViewById(R.id.editTextDigit3);
        digit4=findViewById(R.id.editTextDigit4);
        digit5=findViewById(R.id.editTextDigit5);
        digit6=findViewById(R.id.editTextDigit6);

        auth=FirebaseAuth.getInstance();

        setSupportActionBar(toolbar);

        sp = getSharedPreferences("verify",MODE_PRIVATE);
        userInfoCheck=getSharedPreferences("info",MODE_PRIVATE);
        signUpSp =
getSharedPreferences(Constants.USER_LOGIN_INFO_SHARED_PREFERENCES,MODE_PRIVATE);

        if(sp.getBoolean("logged",false)){
            if(userInfoCheck.getBoolean(Constants.user_exists_preference,false)) {
                GoToMainActivity();
                finish();
```

```
    }
    else
        checkIfUserInfoExist();

}

homeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SignUpActivity.ISRETURNEDFROMVERLAYOUT = true;
        startActivity(new Intent(getApplicationContext(), SignUpActivity.class));
        finish();
    }
});



((EditText)findViewById(R.id.editTextDigit1)).setCursorVisible(false);
findViewById(R.id.editTextDigit1).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ((EditText) findViewById(R.id.editTextDigit1)).setCursorVisible(true); }});
((EditText)findViewById(R.id.editTextDigit1)).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(((EditText)findViewById(R.id.editTextDigit1)).getText().toString().length()==1)
        {
            findViewById(R.id.editTextDigit2).clearFocus();
            findViewById(R.id.editTextDigit2).requestFocus();
            ((EditText) findViewById(R.id.editTextDigit2)).setCursorVisible(true);
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});
((EditText)findViewById(R.id.editTextDigit2)).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(((EditText)findViewById(R.id.editTextDigit2)).getText().toString().length()==1)
        {
            findViewById(R.id.editTextDigit3).clearFocus();
            findViewById(R.id.editTextDigit3).requestFocus();
            ((EditText) findViewById(R.id.editTextDigit3)).setCursorVisible(true);
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});
((EditText)findViewById(R.id.editTextDigit3)).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(((EditText)findViewById(R.id.editTextDigit3)).getText().toString().length()==1)
        {
            findViewById(R.id.editTextDigit4).clearFocus();
            findViewById(R.id.editTextDigit4).requestFocus();
            ((EditText) findViewById(R.id.editTextDigit4)).setCursorVisible(true);
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});
((EditText)findViewById(R.id.editTextDigit4)).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(((EditText)findViewById(R.id.editTextDigit4)).getText().toString().length()==1)
        {
            findViewById(R.id.editTextDigit5).clearFocus();
            findViewById(R.id.editTextDigit5).requestFocus();
            ((EditText) findViewById(R.id.editTextDigit5)).setCursorVisible(true);
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});
((EditText)findViewById(R.id.editTextDigit5)).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(((EditText)findViewById(R.id.editTextDigit5)).getText().toString().length()==1)
        {
            findViewById(R.id.editTextDigit6).clearFocus();
            findViewById(R.id.editTextDigit6).requestFocus();
            ((EditText) findViewById(R.id.editTextDigit6)).setCursorVisible(true);
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});
((EditText)findViewById(R.id.editTextDigit6)).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(((EditText)findViewById(R.id.editTextDigit6)).getText().toString().length()==1)
        {
            ((EditText) findViewById(R.id.editTextDigit6)).setCursorVisible(false);
            findViewById(R.id.btn_continue).clearFocus();
            findViewById(R.id.btn_continue).requestFocus();
            hideSoftInput();
        }
    }
    @Override
    public void afterTextChanged(Editable s) { }
});



textViewResendOTP = findViewById(R.id.TextViewResendOTP);
textViewResendOTP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Toast.makeText(getApplicationContext(),"RESENDING OTP",Toast.LENGTH_SHORT).show();
        textViewResendOTP.setEnabled(false);
        textViewResendOTP.setTextColor(getResources().getColor(R.color.colorInactive));
        ToggleResendTextView(textViewResendOTP);
```

```
                //Write ResendOTP Function Here


            }
        });


        ToggleResendTextView(textViewResendOTP);
        confirmButton = findViewById(R.id.btn_continue);
        editNumberButton = findViewById(R.id.btn_change_number);
        confirmButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Write OTP Submission Function Here

verificationCode=digit1.getText().toString().trim()+""+digit2.getText().toString().trim()+""+digit3.getText().toString().trim()+""+digit4.getText().toString().trim()+""+digit5.getText().toString().trim()+""+digit6.getText().toString().trim();
                System.out.println(verificationCode+" sdf"+ digit1.getText().toString());
                verify(verificationCode);




            }
        });



        editNumberButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(),SignUpActivity.class));
                SignUpActivity.ISRETURNEDFROMVERLAYOUT = true;
                signUpSp.edit().putBoolean(Constants.user_login_state_shared_preference,false).apply();
                finish();

            }
        });

    }

    //Methods
    public void hideSoftInput() {
        View view1 = this.getCurrentFocus();
        if(view1!= null){
            InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(view1.getWindowToken(), 0);
        }
    }
    public void ToggleResendTextView(final TextView textView)
    {
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
```

```
        textView.setEnabled(true);
        textView.setTextColor(getResources().getColor(R.color.colorActive));
      }
    }, 20000);
  }

  public void verify(String verificationCode){
    System.out.println(verification+" verify");
    verfyPhoneNumber(verification,verificationCode);
  }

  private void verfyPhoneNumber(String verification, String entereredCodeString) {
    System.out.println(verification+" credential "+entereredCodeString);
    PhoneAuthCredential phoneAuthCredential=
PhoneAuthProvider.getCredential(verification,entereredCodeString);
    signInWithPhoneAuthCredential(phoneAuthCredential);
  }



  private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
          @Override
          public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
              // Sign in success, update UI with the signed-in user's information
}

              //Log.d(TAG, "signInWithCredential:success");
              //System.out.println("Successful");
              FirebaseUser user = task.getResult().getUser();
              uid= FirebaseAuth.getInstance().getCurrentUser().getUid();
              if(userInfoCheck.getBoolean(Constants.user_exists_preference,false)) {
                  GoToMainActivity();
                  finish();
              }
              else
                  checkIfUserInfoExist();

              sp.edit().putBoolean("logged",true).apply();
              Toast.makeText(getApplicationContext(),"User Signed In
Successfully",Toast.LENGTH_SHORT).show();

            } else {
              //System.out.println(task.getException()+" task exception");
              Toast.makeText(getApplicationContext(),"Please use the valid
code",Toast.LENGTH_SHORT).show();
              // Sign in failed, display a message and update the UI


            }
          }
        });
  }
```

```
  public void checkIfUserInfoExist(){

    FirebaseDatabase database = FirebaseDatabase.getInstance();

    uid=FirebaseAuth.getInstance().getUid();

    DatabaseReference ref = database.getReference().child("UserInfo").child(uid);

    ValueEventListener valueEventListener = new ValueEventListener() {

      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
        if(dataSnapshot.exists()){

          UserInfoData user = dataSnapshot.getValue(UserInfoData.class);

          userInfoCheck.edit().putString(Constants.username_preference, user.getName()).apply();
          userInfoCheck.edit().putString(Constants.user_dob_preference, user.getDob()).apply();

          userInfoCheck.edit().putString(Constants.user_home_address_preference,
user.getHomeAddress()).apply();

          userInfoCheck.edit().putString(Constants.user_home_address_latlng_preference,
user.getHomeLatLng()).apply();
```

```
                    userInfoCheck.edit().putString(Constants.uid_preference,uid).apply();

                    userInfoCheck.edit().putString(Constants.user_phone_no_preference,
user.getContactNumber()).apply();


                    userInfoCheck.edit().putBoolean(Constants.user_exists_preference,true).apply();




if(String.valueOf(dataSnapshot.child(Constants.userInfo_node_workAddress).getValue())!=null) {


                        userInfoCheck.edit().putString(Constants.user_work_address_preference,
user.getWorkAddress()).apply();

                        userInfoCheck.edit().putString(Constants.user_work_address_latlng_preference,
user.getWorkLatLng()).apply();

                    }


                    GoToMainActivity();

                }

                else {


                    GotoUserInfoFormActivity();

                }


                finish();


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

            }
        };
        ref.addListenerForSingleValueEvent(valueEventListener);


    }

    public void GoToMainActivity(){

        startActivity(new Intent(getApplicationContext(), MenuActivity.class));
    }
    public void GotoUserInfoFormActivity(){

        startActivity(new Intent(getApplicationContext(), UserInfoFormActivity.class));
    }
}
```