



BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng

Bài tập buổi 8:

Lập trình ứng dụng di động với chức năng theo yêu cầu và bật ProGuard

GV: Nghi Hoàng Khoa

Ngày báo cáo: 10/05/2023

Nhóm: Pengu

1. THÔNG TIN CHUNG:

Lớp: NT213.N21.ANTT

STT	Họ và tên	MSSV	Email
1	Phạm Phúc Đức	20520162	20520162@gm.uit.edu.vn
2	Nguyễn Hoàng Phúc	20520277	20520277@gm.uit.edu.vn
3	Nguyễn Đức Tấn	20520751	20520751@gm.uit.edu.vn
4	Nguyễn Nhật Hiếu Trung	20520830	20520830@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Lập trình ứng dụng di động có chức năng Đăng ký/Login/logout tài khoản	100%	Cả nhóm
2	Sử dụng ProGuard cho ứng dụng Android	100%	Cả nhóm

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Lập trình ứng dụng di động có chức năng Đăng ký/Login/logout tài khoản

1.1. Với database là SQLite với dữ liệu được lưu tại máy android, chúng ta có file DatabaseHelper.java:

```
public class DatabaseHelper extends SQLiteOpenHelper {  
    public static final String dbName = "SignLog.db";  
    public DatabaseHelper(@Nullable Context context) {  
        super(context, "SignLog.db", null, 1);  
    }  
    @Override  
    public void onCreate(SQLiteDatabase MyDatabase) {  
        MyDatabase.execSQL("create Table users(username TEXT primary key, password TEXT)");  
    }  
    @Override  
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {  
        MyDB.execSQL("drop Table if exists users");  
    }  
    public Boolean insertData(String username, String password){  
        SQLiteDatabase MyDatabase = this.getWritableDatabase();  
        ContentValues contentValues = new ContentValues();  
        contentValues.put("username", username);  
        contentValues.put("password", password);  
        long result = MyDatabase.insert("users", null, contentValues);  
        if (result == -1) {  
            return false;  
        } else {  
            return true;  
        }  
    }  
}
```

```
}

public Boolean checkUser(String username){

    SQLiteDatabase MyDatabase = this.getWritableDatabase();

    Cursor cursor = MyDatabase.rawQuery("Select * from users where username = ?", new String[]{username});

    if(cursor.getCount() > 0) {

        return true;

    }else {

        return false;

    }

}

public Boolean checkUserPassword(String username, String password){

    SQLiteDatabase MyDatabase = this.getWritableDatabase();

    Cursor cursor = MyDatabase.rawQuery("Select * from users where username = ? and password = ?", new String[]{username, password});

    if (cursor.getCount() > 0) {

        return true;

    }else {

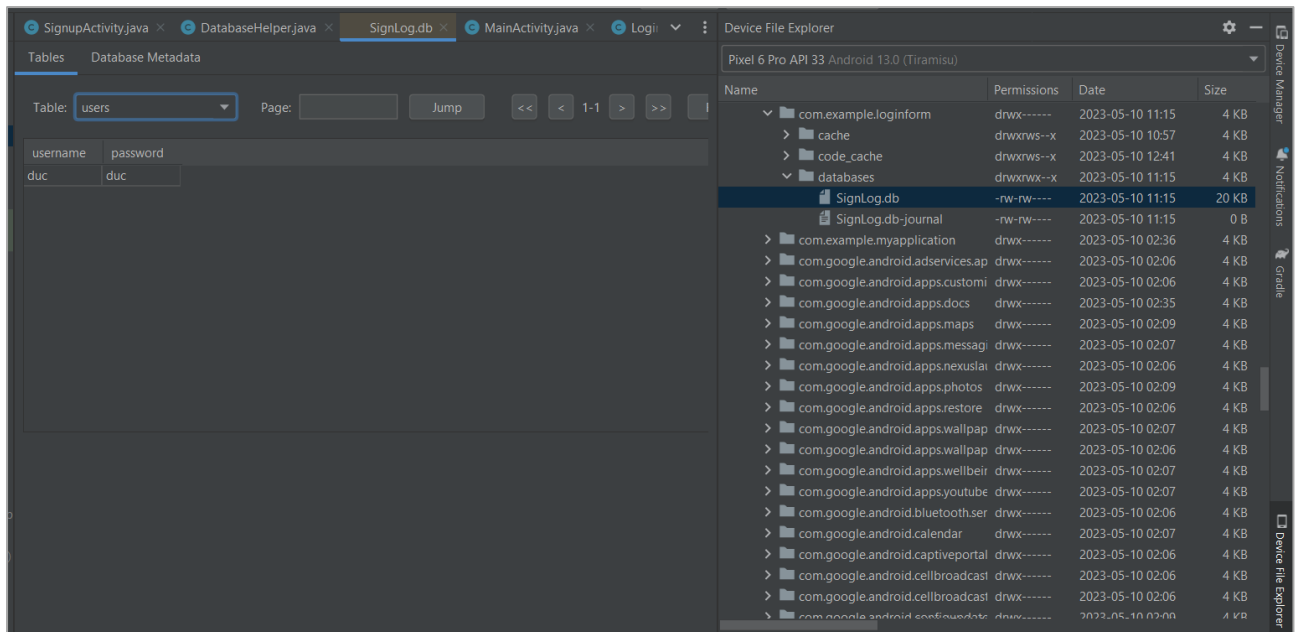
        return false;

    }

}

}
```

Các thao tác sẽ được thực hiện trên file SignUp.db được lưu trữ trên máy android, với đường dẫn là `data/data/<tên package>/databases/`:



Hình 1: Vị trí lưu thông tin đăng nhập trên máy

Trong đó, DatabaseHelper.java sẽ giúp chúng ta thao tác với dữ liệu khi thực hiện đăng kí (thêm mới với tài khoản đăng kí hợp lệ) và đồng thời cũng giúp thực hiện kiểm tra tài khoản khớp với dữ liệu trong database với hàm **checkUser** và **checkUserPassword**.

- Việc thêm dữ liệu mới vào database khi người dùng thực hiện đăng kí, chúng ta sẽ có bước kiểm tra dữ liệu nhập vào với hàm **insert()**

```
public long insert(String table, String nullColumnHack, ContentValues values) {
    try {
        return insertWithOnConflict(table, nullColumnHack, values,
CONFLICT_NONE);
    } catch (SQLException e) {
        Log.e(TAG, "Error inserting " + values, e);
        return -1;
    }
}
```

Nếu như người dùng nhập vào tài khoản không hợp lệ -> hàm này sẽ trả về giá trị -1 và hàm insertData() trong DatabaseHelper.java sẽ huỷ việc tạo tài khoản này.

- Kiểm tra dữ liệu: khi người dùng đăng nhập, hệ thống sẽ thực hiện query tới database để tìm tới user trùng khớp và tiếp đến kiểm tra nếu mật khẩu nhập vào trùng khớp với user đó

Nếu tìm ra dữ liệu hợp lệ **cursor.getCount() > 0** kết quả sẽ trả về **true**.

1.2. Trang SignupActivity với SignupActivity.java:

```
public class SignupActivity extends AppCompatActivity {

    ActivitySignupBinding binding;

    DatabaseHelper databaseHelper;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        binding = ActivitySignupBinding.inflate(getLayoutInflater());

        setContentView(binding.getRoot());

        databaseHelper = new DatabaseHelper(this);

        binding.registerBtn.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                String user = binding.username.getText().toString();

                String password = binding.password.getText().toString();

                String confirmPassword = binding.repassword.getText().toString();

                if(user.equals("")||password.equals("")||confirmPassword.equals(""))

                    Toast.makeText(SignupActivity.this, "All fields are mandatory", Toast.LENGTH_SHORT).show();

                else{

                    if(password.equals(confirmPassword)){

                        Boolean checkUsername = databaseHelper.checkUser(user);

                        if(checkUsername == false){

                            Boolean insert = databaseHelper.insertData(user, password);

                            if(insert == true){

                                Toast.makeText(SignupActivity.this, "Signup Successfully!", Toast.LENGTH_SHORT).show();
```

```
        Intent intent = new Intent(getApplicationContext(), LoginActivity.class);

        startActivity(intent);

    }else{

        Toast.makeText(SignupActivity.this, "Signup Failed!", Toast.LENGTH_SHORT).show();

    }

}

else{

    Toast.makeText(SignupActivity.this, "User already exists! Please login",
Toast.LENGTH_SHORT).show();

    }

}else{

    Toast.makeText(SignupActivity.this, "Invalid Password!", Toast.LENGTH_SHORT).show();

    }

}

}

});

binding.loginBtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Intent intent = new Intent(SignupActivity.this, LoginActivity.class);

        startActivity(intent);

    }

});

}

}
```

Trong đó, hàm `onClick()` sẽ chứa câu lệnh `if` để kiểm tra dữ liệu nhập vào cần phải khác rỗng, nếu 1 trong 3 trường Username, Password, Confirmpassword bị thiếu, người dùng sẽ nhận được thông báo *"All fields are mandatory"*



Hình 2: Thông báo hiện ra

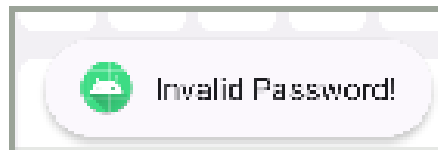
Tiếp đến, khi mọi thông tin đã được nhập đầy đủ, hàm `checkUser()` từ `DatabaseHelper` sẽ được gọi lại để kiểm tra username đã tồn tại trong database hay chưa, nếu chưa thì user mới sẽ được thêm vào đồng thời hiện thông báo *"Signup Successfully!"*



Hình 3: Thông báo đăng kí

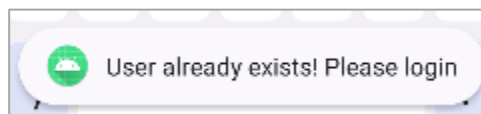
Tương tự:

- Khi mật khẩu xác nhận khác với mật khẩu đã nhập:



Hình 4: Sai mật khẩu xác nhận

- Khi tạo username đã có trong database:



Hình 5: Username đã tồn tại

1.3. Trang đăng nhập với `LoginActivity.java`:

```
public class LoginActivity extends AppCompatActivity {  
    ActivityLoginBinding binding;  
    DatabaseHelper databaseHelper;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);

binding = ActivityLoginBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

databaseHelper = new DatabaseHelper(this);

binding.loginBtn1.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        String username = binding.username1.getText().toString();

        String password = binding.password1.getText().toString();

        if(username.equals("")||password.equals(""))

            Toast.makeText(LoginActivity.this, "All fields are mandatory", Toast.LENGTH_SHORT).show();

        else{

            Boolean checkCredentials = databaseHelper.checkUserPassword(username, password);

            if(checkCredentials == true){

                Toast.makeText(LoginActivity.this, "Login Successfully!", Toast.LENGTH_SHORT).show();

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

                startActivity(intent);

            }else{

                Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LENGTH_SHORT).show();

            }

        }

    }

});

binding.registerBtn1.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        Intent intent = new Intent(LoginActivity.this, SignupActivity.class);

        startActivity(intent);

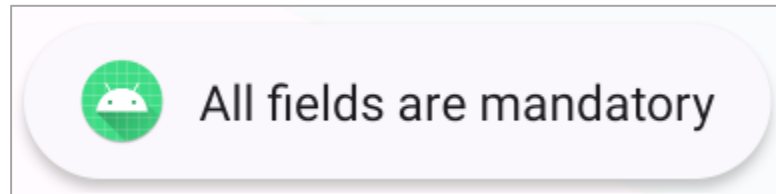
    }

});
```



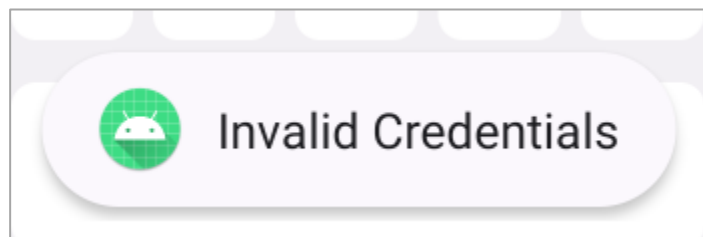
```
});  
}  
}
```

Tương tự như trên, mọi trường cần phải khác rỗng:



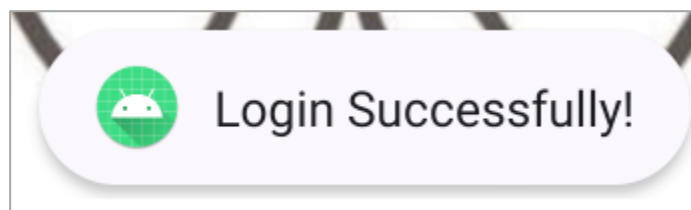
Hình 6: Các trường cần được nhập đầy đủ

Hàm checkUserPassword trong DatabaseHelper được gọi lại để kiểm tra dữ liệu đăng nhập, kết quả sẽ được gán vào checkCredentials, nếu giá trị là **false** – đăng nhập sai:



Hình 7: Đăng nhập sai

Khi đăng nhập hợp lệ, chúng ta sẽ nhận được thông báo và chuyển hướng đến trang chủ (đồng thời có thể logout):



Hình 8: Đăng nhập thành công

1.4. Trang chủ sẽ xử lý thao tác nút Log out, MainActivity.java:

```
public class MainActivity extends AppCompatActivity {
```

```
    Button logout;
```

```
@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    new Handler().postDelayed(new Runnable() {

        @Override

        public void run() {

            startActivity(new Intent(getApplicationContext(), LoginActivity.class));

        }

    },60000);

    logout = (Button) findViewById(R.id.logoutBtn);

    logout.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            Intent intent = new Intent(getApplicationContext(), LoginActivity.class);

            startActivity(intent);

        }

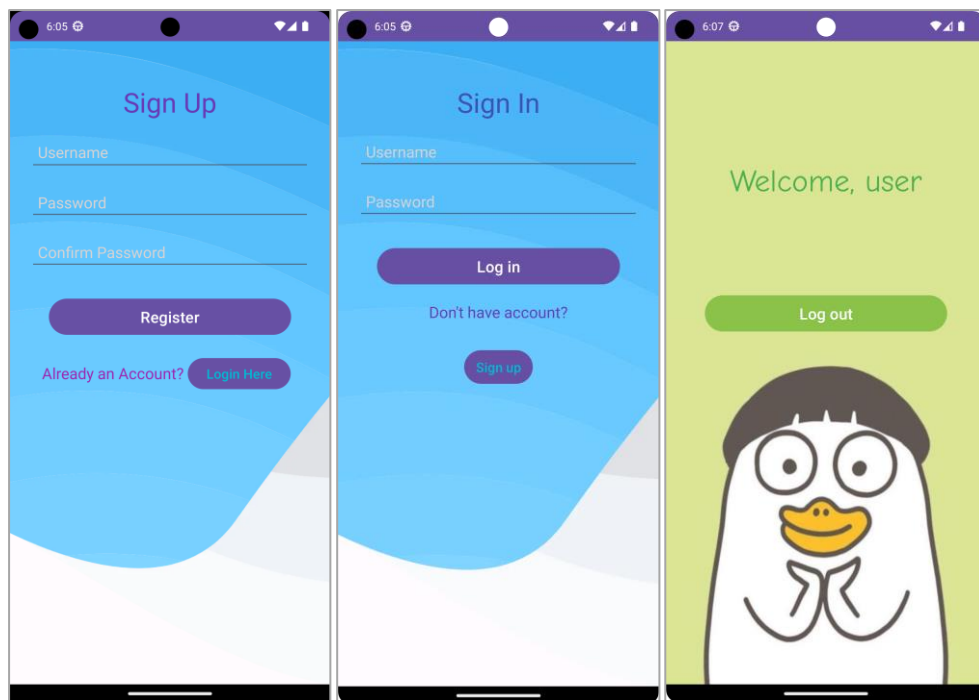
    });

}
```

Trang chủ sẽ thực hiện chuyển hướng tự động về trang **LoginActivity** (với hàm **startActivity()**) sau 60000 milliseconds (1 phút) trong hàm **run()**

Người dùng có thể chủ động quay trở về trang Login khi thực hiện bấm vào nút Log out với cơ chế tương tự trong hàm **onClick()**

1.5. Xây dựng giao diện cho các trang: SignUp – SignIn – Home (SignOut)

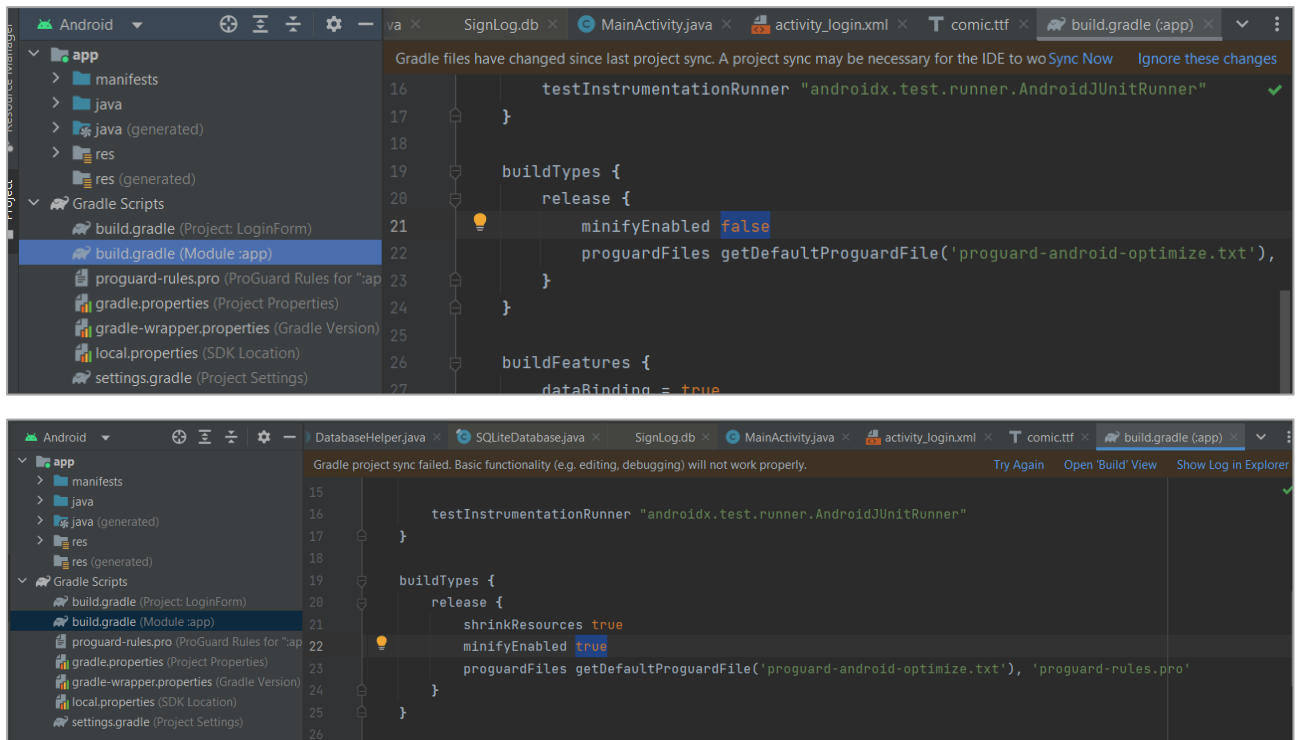


Hình 9: Giao diện các trang

Note: Trang web cuối có sử dụng font chữ **Comic Neue Regular**.

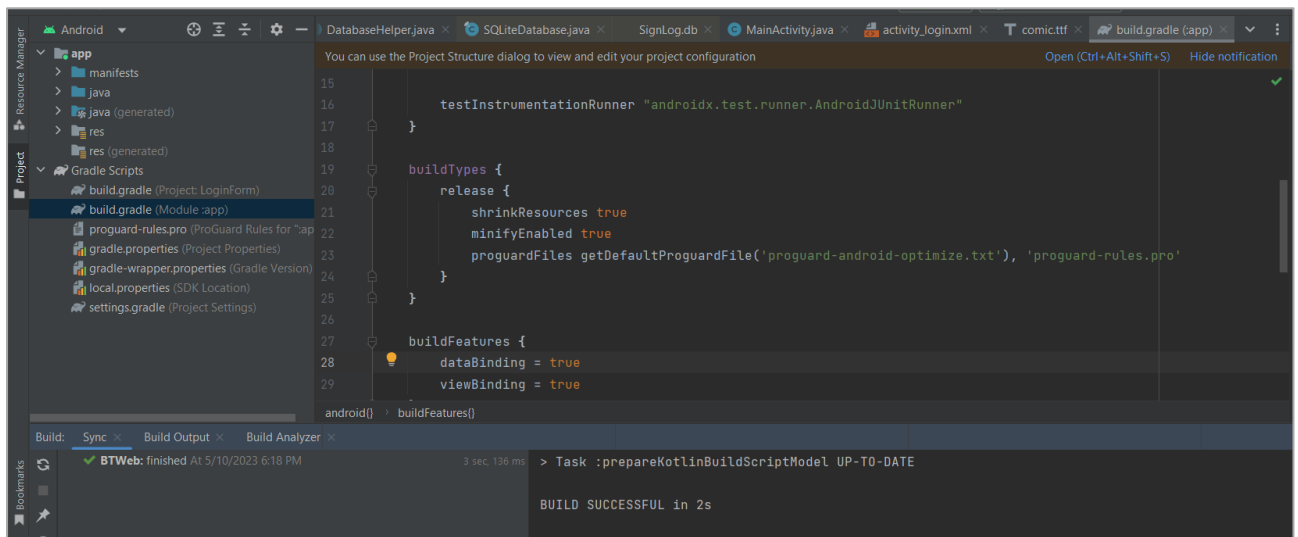
2. Sử dụng ProGuard cho ứng dụng Android

Bật ProGuard bằng cách truy cập vào file build.gradle và thực hiện thay đổi:



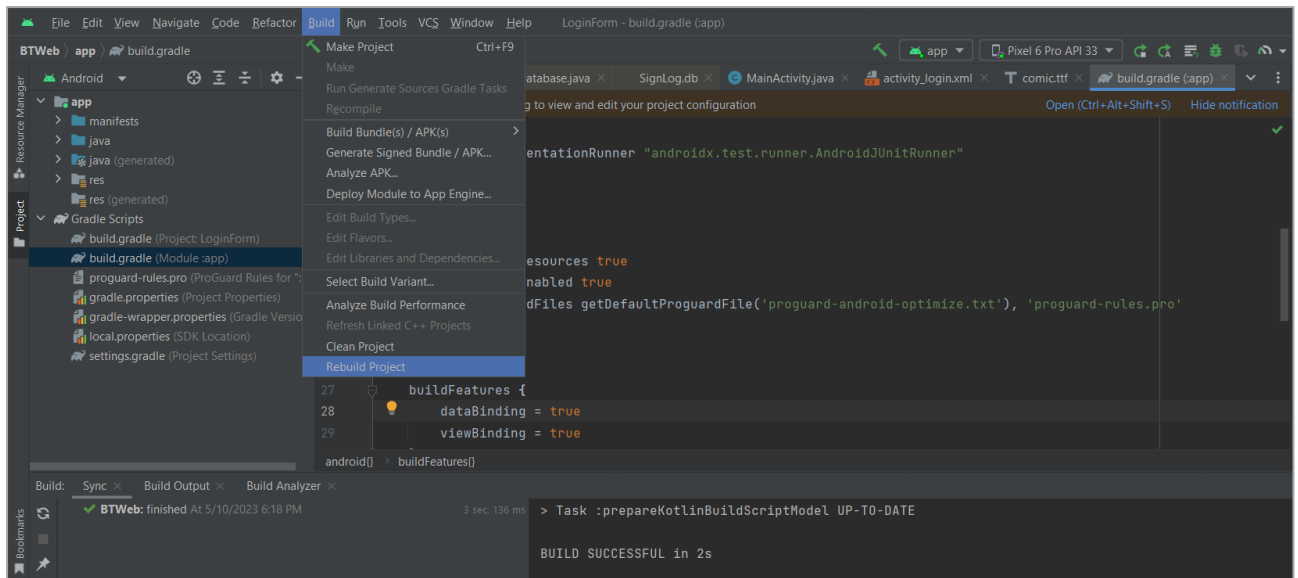
Hình 10: Bật ProGuard và thêm các tùy chọn

Lưu lại cài đặt bằng cách bấm Sync Now:



Hình 11: Lưu lại cài đặt thành công

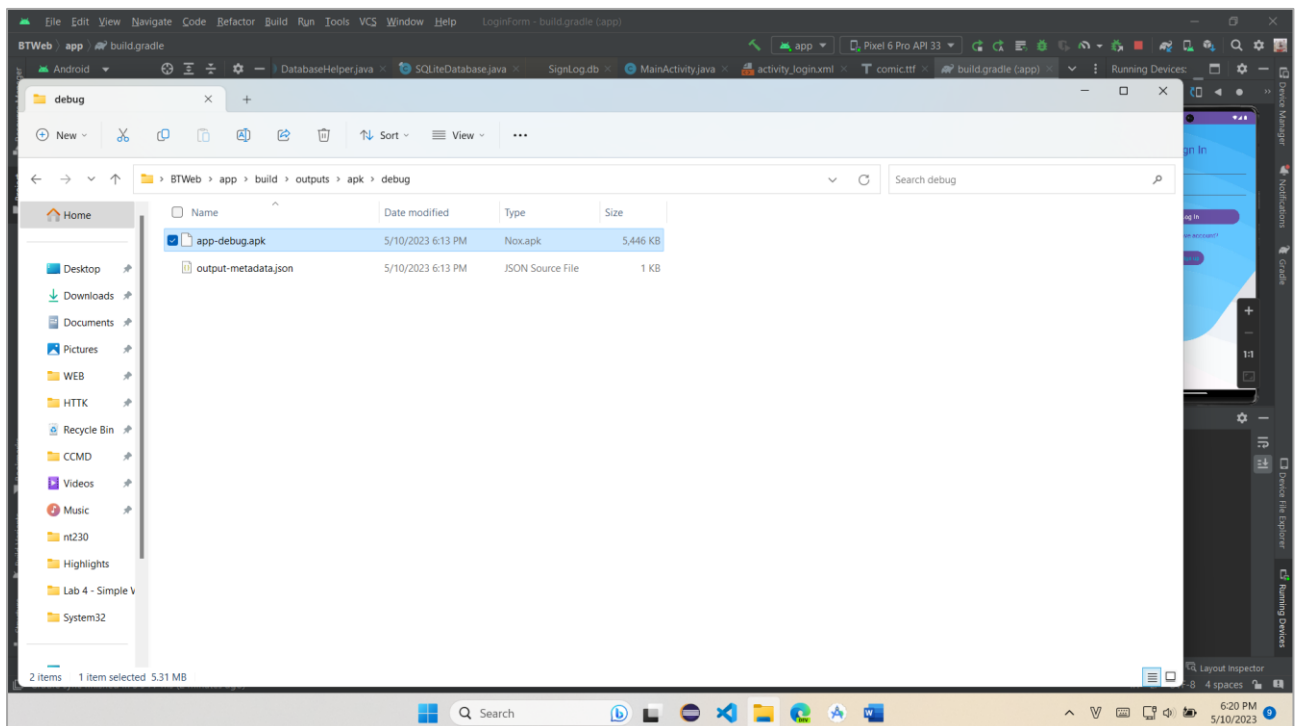
Build file apk:



Hình 12: Rebuild app

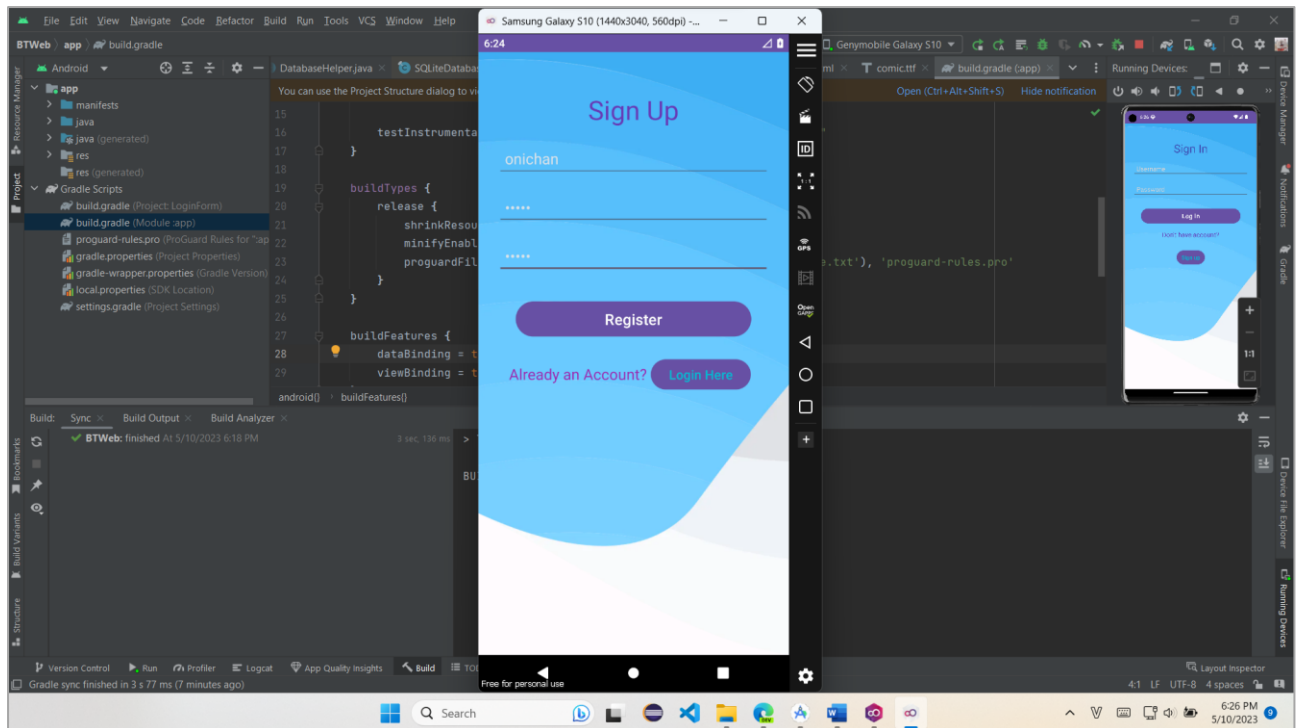
Vị trí lưu của app sau khi build sẽ là:

C:\Users\Pengu\Desktop\BTWeb\app\build\outputs\apk\debug

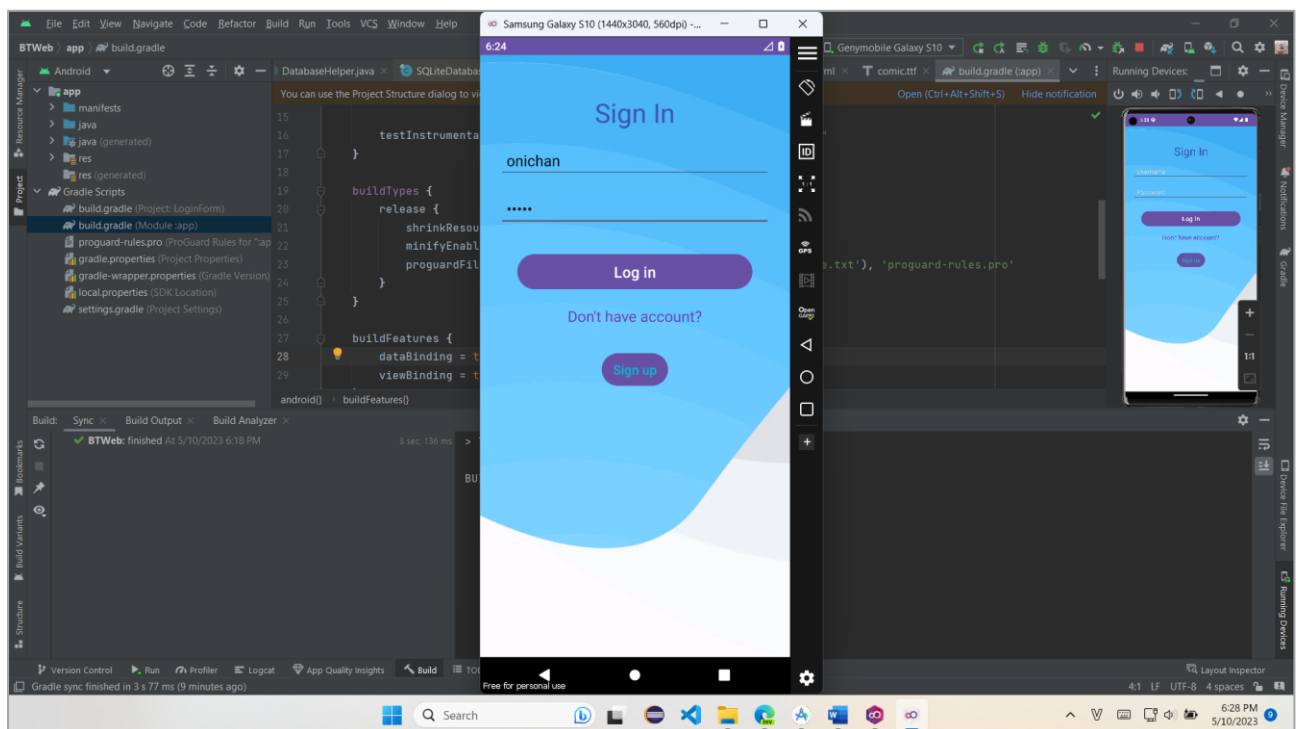


Hình 13: Build app thành công

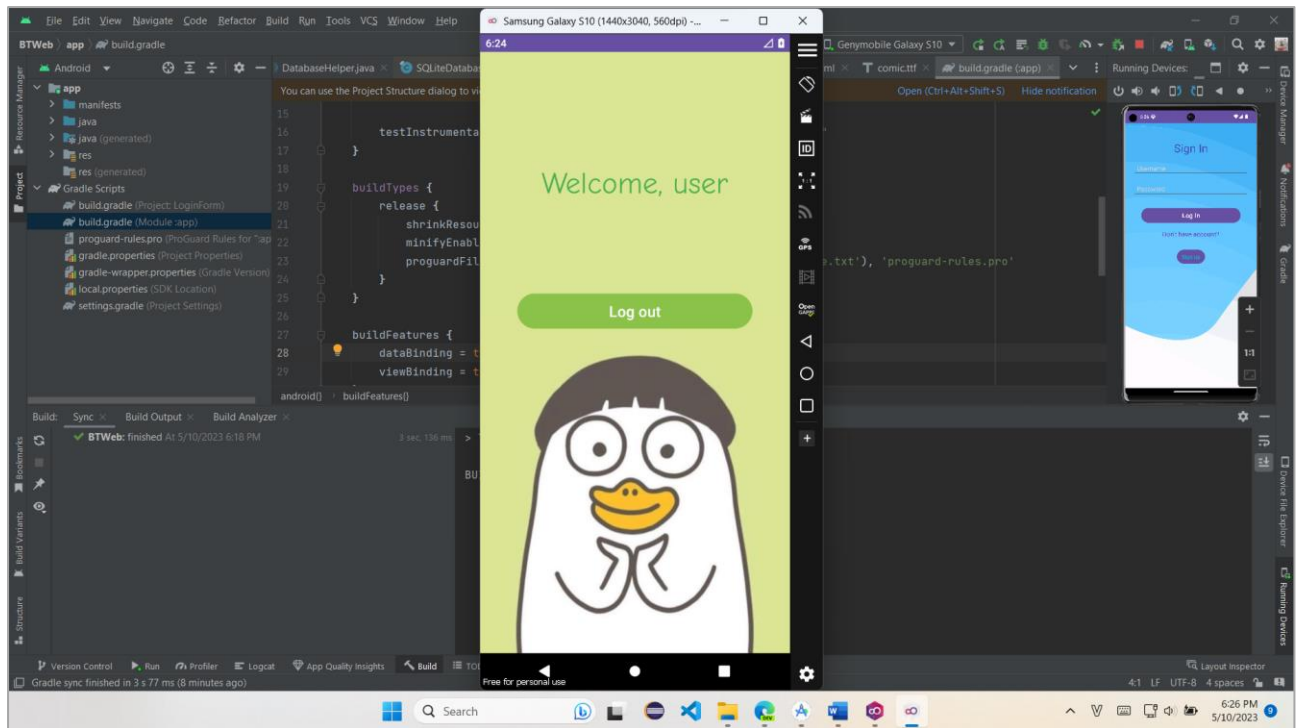
Thực hiện test trên thiết bị khác:



Hình 14: Đăng kí



Hình 15: Đăng nhập



Hình 16: Trang chủ khi đăng nhập thành công

HẾT