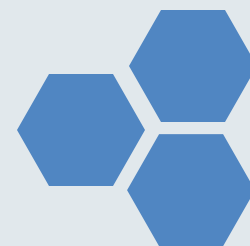




**Ví dụ Video Short with FireBase and API
Server kết hợp ViewPager2 trong Android**



- Bước 1: Thêm thư viện
- Bước 2: Thiết kế giao diện
- Bước 3: Khởi tạo database FireBase và Store
- Bước 4: Viết Adapter
- Bước 5: Viết hàm trong MainActivity

□ Bước 1: Thêm thư viện

- Mở file build.gradle (Module:app) thêm vào phần dependency

```
implementation 'com.google.android.gms:play-services-auth:20.4.1'
/**firebase dependencies*/
implementation 'com.google.firebase:firebase-auth:21.1.0'
implementation 'com.google.firebase:firebase-database:20.1.0'
implementation platform('com.google.firebase:firebase-bom:31.2.3')
implementation 'com.google.firebase:firebase-database-ktx'
implementation 'com.google.firebase:firebase-firestore:24.4.5'
implementation 'com.google.firebase:firebase-storage:20.1.0'
implementation 'com.firebaseui:firebase-ui-database:8.0.1'
```

- Mở file build.gradle (Module:app) thêm vào phần plugins

```
plugins {
    id 'com.google.gms.google-services'
}
```

- Bước 1: Thêm thư viện
- Mở file setting.gradle (Project setting) thêm vào

```
buildscript {
    repositories {
        // Make sure that you have the following two repositories
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
    dependencies {
        // Add the dependency for the Google services Gradle plugin
        classpath 'com.google.gms:google-services:4.3.15'
    }
}
```

- Bước 1: Thêm thư viện
- Mở file AndroidManifest.xml và thêm

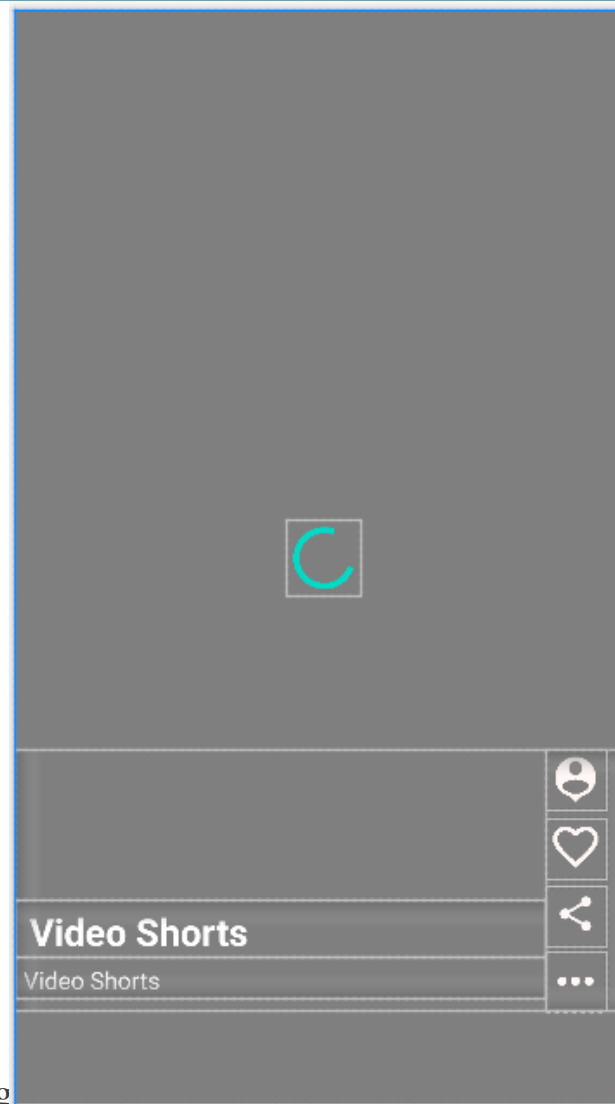

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```
- Mở file AndroidManifest.xml và thêm vào thẻ mở Application


```
android:usesCleartextTraffic="true"
```

□ Bước 2: Thiết kế giao diện

```

activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".Activities.MainActivity">
9
10     <androidx.viewpager2.widget.ViewPager2
11         android:id="@+id/vpager"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         android:orientation="vertical"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintLeft_toLeftOf="parent"
17         app:layout_constraintRight_toRightOf="parent"
18         app:layout_constraintTop_toTopOf="parent" />
19 </androidx.constraintlayout.widget.ConstraintLayout>
    
```



□ Bước 2: Thiết kế giao diện

```
single_video_row.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     xmlns:app="http://schemas.android.com/apk/res-auto"
7     android:background="@color/black">
8     <VideoView
9         android:id="@+id/videoView"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        app:layout_constraintBottom_toBottomOf="parent"
13        app:layout_constraintTop_toTopOf="parent"
14        app:layout_constraintEnd_toEndOf="parent"
15        app:layout_constraintStart_toStartOf="parent"
16        app:layout_constraintHorizontal_bias="0.0"
17        app:layout_constraintVertical_bias="1.0"
18    />
19    <ProgressBar
20        android:id="@+id/videoProgressBar"
21        app:layout_constraintBottom_toBottomOf="parent"
22        app:layout_constraintStart_toStartOf="parent"
23        app:layout_constraintTop_toTopOf="parent"
24        app:layout_constraintEnd_toEndOf="parent"
25        android:layout_width="50dp"
26        android:layout_height="50dp"/>
27    <LinearLayout
```

```
single_video_row.xml
28        android:id="@+id/linearLayout"
29        android:orientation="horizontal"
30        app:layout_constraintBottom_toBottomOf="parent"
31        app:layout_constraintEnd_toEndOf="parent"
32        app:layout_constraintHorizontal_bias="1.0"
33        app:layout_constraintStart_toStartOf="parent"
34        app:layout_constraintTop_toTopOf="parent"
35        app:layout_constraintVertical_bias="0.888"
36        android:layout_width="match_parent"
37        android:layout_height="wrap_content">
38    <LinearLayout
39        android:layout_weight="1"
40        android:orientation="vertical"
41        android:gravity="bottom"
42        android:layout_gravity="bottom"
43        android:layout_marginBottom="10dp"
44        android:layout_width="0dp"
45        android:layout_height="wrap_content">
46    <TextView
47        android:id="@+id/textVideoTitle"
48        android:paddingStart="9dp"
49        android:paddingTop="5dp"
50        android:paddingEnd="5dp"
51        android:text="Video Shorts"
52        android:textColor="@color/white"
53        android:textSize="25sp"
54        android:textStyle="bold"
55        android:shadowDx="0"
56        android:shadowDy="0"
```

□ Bước 2: Thiết kế giao diện

single_video_row.xml

```

57         android:shadowRadius="15"
58         android:layout_width="match_parent"
59         android:layout_height="wrap_content"/>
60     <TextView
61         android:id="@+id/textVideoDescription"
62         android:layout_width="match_parent"
63         android:layout_height="wrap_content"
64         android:paddingStart="5dp"
65         android:paddingTop="5dp"
66         android:paddingEnd="5dp"
67         android:shadowDx="0"
68         android:shadowDy="0"
69         android:shadowRadius="15"
70         android:text="Video Shorts"
71         android:textColor="#fff"
72         android:textSize="16sp" />
73     </LinearLayout>
74     <LinearLayout
75         android:orientation="vertical"
76         android:layout_gravity="end"
77         android:gravity="end"
78         android:layout_width="wrap_content"
79         android:layout_height="wrap_content">
80         <ImageView
81             android:id="@+id/imPerson"
82             android:src="@drawable/ic_person_pin"
83             android:layout_gravity="end"
84             android:padding="2dp"
85             android:layout_marginEnd="18dp"

```

single_video_row.xml

```

86         android:layout_width="40dp"
87         android:layout_height="40dp" />
88     <ImageView
89         android:id="@+id/favorites"
90         android:src="@drawable/ic_favorite"
91         android:layout_gravity="end"
92         android:padding="2dp"
93         android:layout_marginTop="5dp"
94         android:layout_marginEnd="18dp"
95         android:layout_width="40dp"
96         android:layout_height="40dp" />
97     <ImageView
98         android:id="@+id/imShare"
99         android:src="@drawable/ic_share"
100        android:layout_gravity="end"
101        android:padding="6dp"
102        android:layout_marginTop="5dp"
103        android:layout_marginEnd="18dp"
104        android:layout_width="40dp"
105        android:layout_height="40dp" />
106     <ImageView
107         android:id="@+id/imMore"
108         android:src="@drawable/ic_more"
109         android:layout_gravity="end"
110         android:padding="2dp"
111         android:layout_marginTop="5dp"
112         android:layout_marginEnd="18dp"
113         android:layout_width="40dp"
114         android:layout_height="40dp" />

```

115 </LinearLayout>

116 </LinearLayout>

117 </androidx.constraintlayout.widget.ConstraintLayout>

115 </LinearLayout>

116 </LinearLayout>

117 </androidx.constraintlayout.widget.ConstraintLayout>

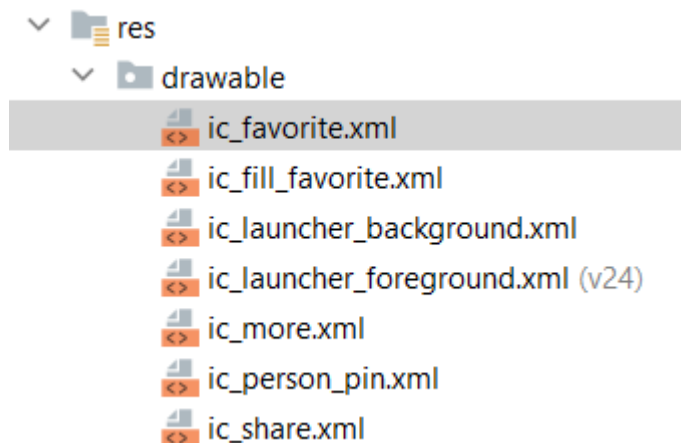
115 </LinearLayout>

116 </LinearLayout>

117 </androidx.constraintlayout.widget.ConstraintLayout>

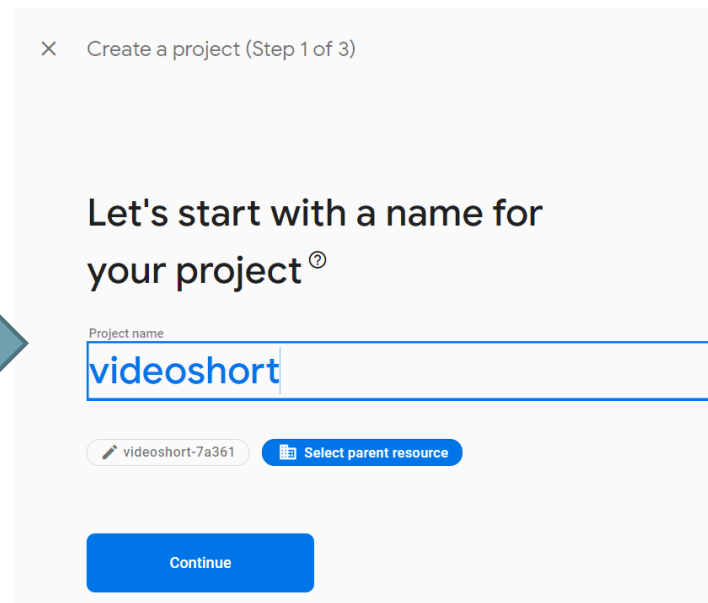
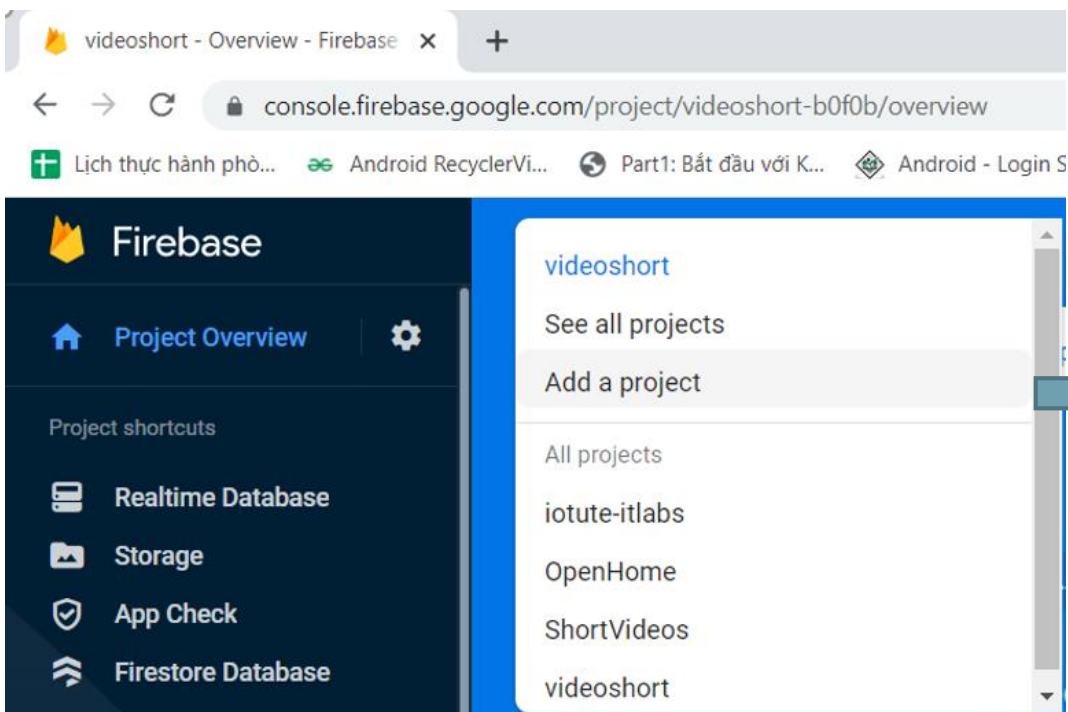
□ Bước 2: Thiết kế giao diện

- Các nút favorite, share,... các em dùng chức năng Vector Asset để tạo nhé.



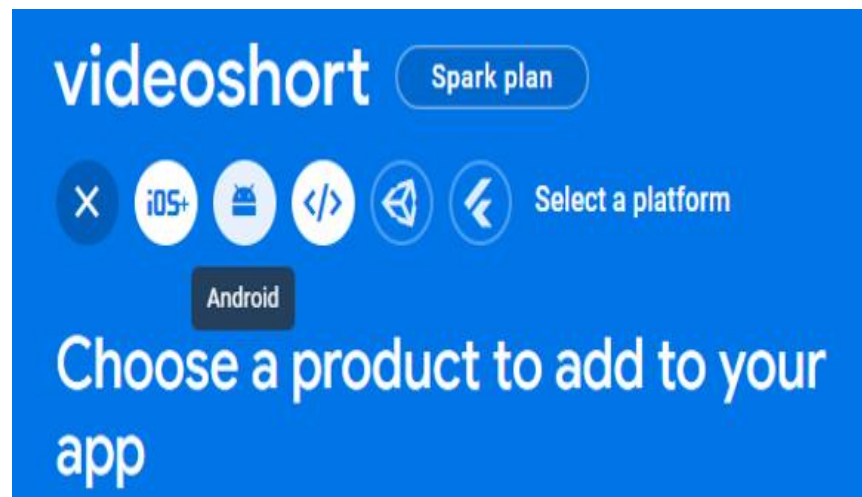
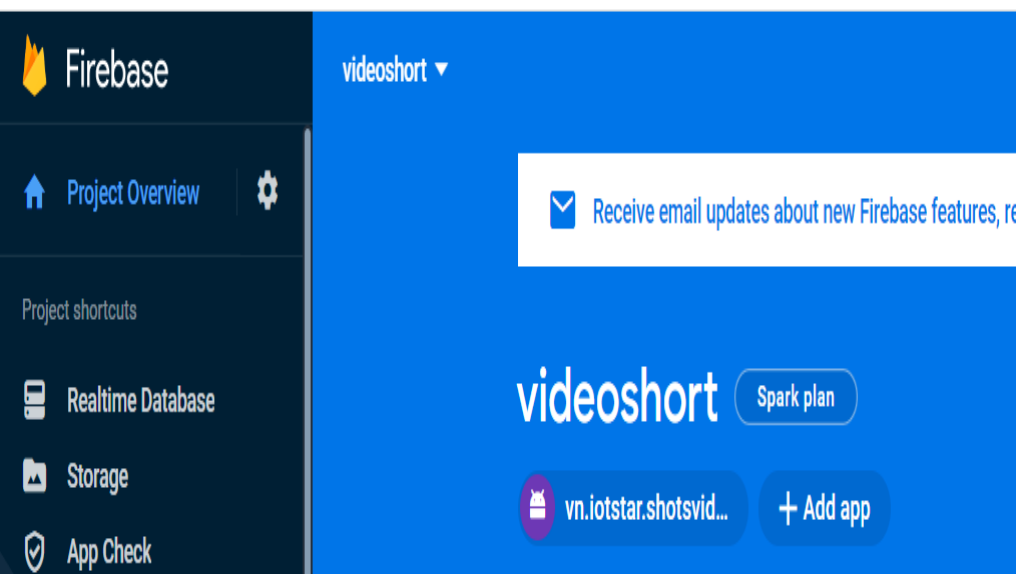
□ Bước 3: Khởi tạo database FireBase và Store

Đăng nhập vào trang : <https://console.firebase.google.com/> tạo 01 project -> Add a project thực hiện theo hướng dẫn 03 bước để có 01 project.



□ Bước 3: Khởi tạo database FireBase và Storage

Bạn tiến hành tạo App và chọn app thuộc IOS hay Android như hình



□ Bước 3: Khởi tạo database FireBase và Storage

Thực hiện 05 bước để cấu hình app của mình: bước 1: copy tên trong namespace past vào bước 1.

× Add Firebase to your Android app


✓ Register app
Android package name: vn.iotstar.shotsvideo

2 Download and then add config file

Download google-services.json


Switch to the Project view in Android Studio to see your project root directory.


Move your downloaded google-services.json file into your module (app-level) root directory.



google-services.json

Next




build.gradle (:app) ×

You can use the Project Structure dialog to view and edit your project configuration

```

3      id 'org.jetbrains.kotlin.android'
4      id 'com.google.gms.google-services'
5
6  }
7
8  android {
9      namespace 'vn.iotstar.shotsvideo'
10     compileSdk 33
11
12     defaultConfig {
13         applicationId "vn.iotstar.shotsvideo"
14         minSdk 22
15         targetSdk 33
16         versionCode 1
17         versionName "1.0"
18     }

```

□ Bước 3: Khởi tạo database FireBase và Storage

Thực hiện 05 bước để cấu hình app của mình: bước 2:tải file google-service.json bỏ vào thư mục app của project mình trên Android như hình

Bấm next để qua bước 3, 4,5 (lưu ý bước 3, 4 đã làm ở trên rồi)

× Add Firebase to your Android app

✓ Register app
Android package name: vn.lotstar.shotsvideo

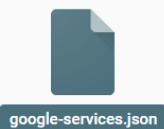
2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

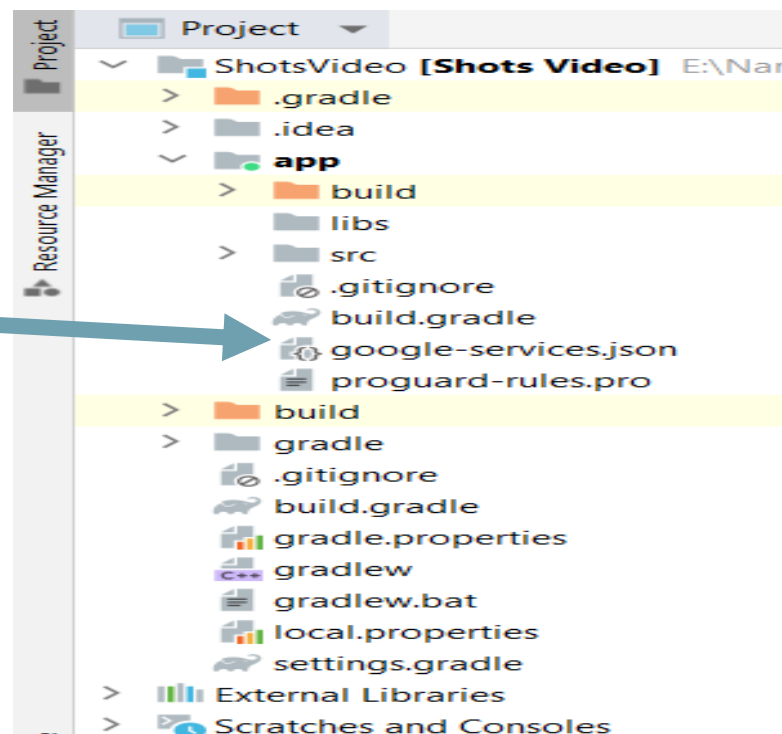
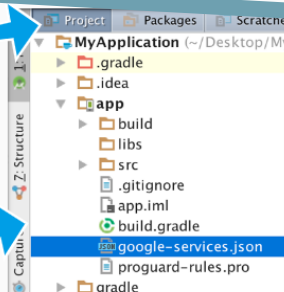
Download google-services.json

Switch to the Project view in Android Studio to see your project root directory.

Move your downloaded google-services.json file into your module (app-level) root directory.



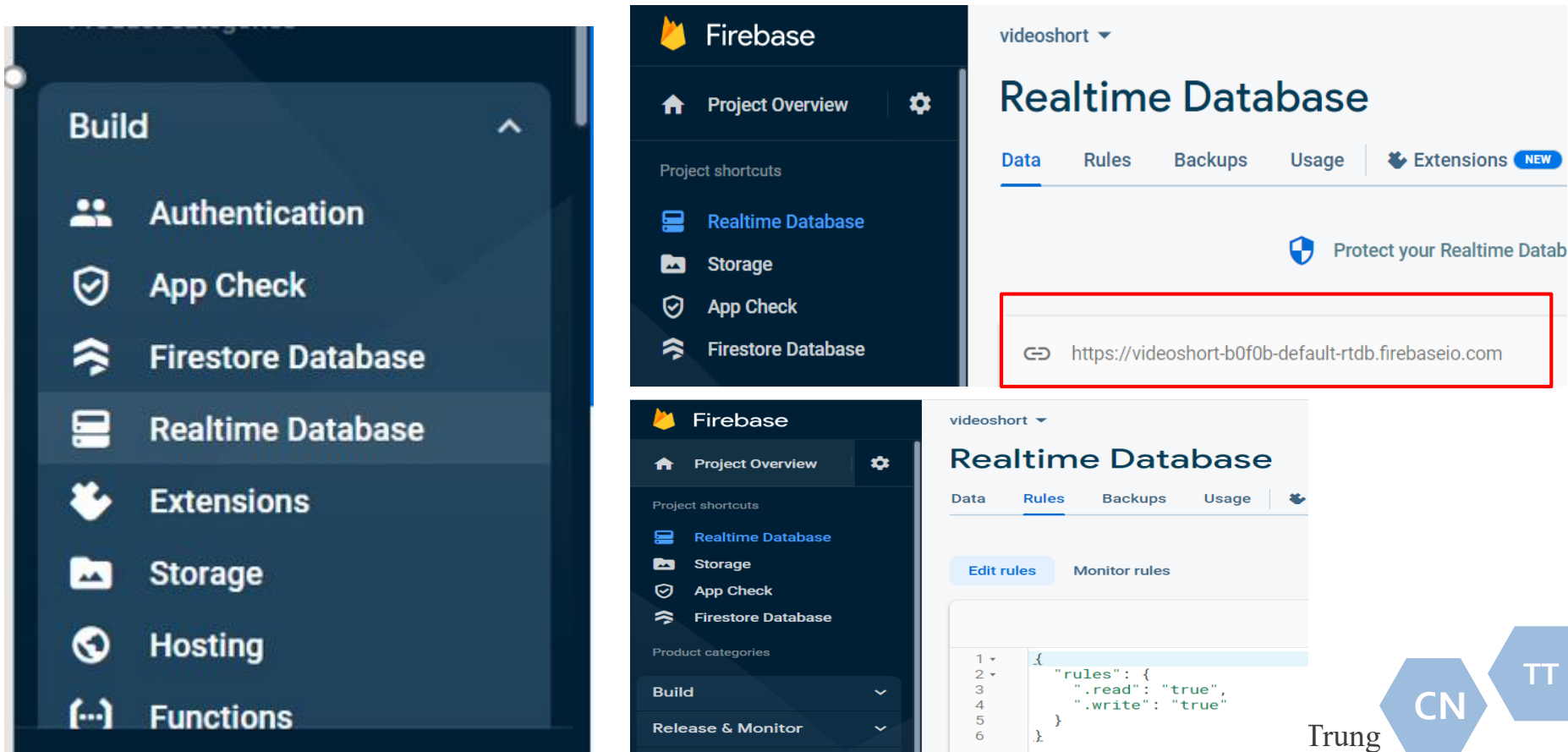
Next



□ Bước 3: Khởi tạo database FireBase và Storage

Chọn Build và Chọn Realtime Database để thêm Database vào project của mình

Chọn Thẻ Rules để cài quyền truy cập database như hình dưới



The first screenshot shows the 'Build' menu in the Firebase console, with 'Realtime Database' highlighted. The second screenshot shows the 'Project Overview' page with 'Realtime Database' selected in the 'Project shortcuts' list. The third screenshot shows the 'Realtime Database' page with the 'Rules' tab selected, displaying the default rules for the database.

Build

- Authentication
- App Check
- Firestore Database
- Realtime Database**
- Extensions
- Storage
- Hosting
- Functions

Firestore

Project Overview

Project shortcuts

- Realtime Database
- Storage
- App Check
- Firestore Database

Product categories

- Build
- Release & Monitor

videoshort

Realtime Database

Data Rules Backups Usage Extensions NEW

Protect your Realtime Database

<https://videoshort-b0f0b-default-rtdb.firebaseio.com>

Realtime Database

Data Rules Backups Usage

Edit rules Monitor rules

```

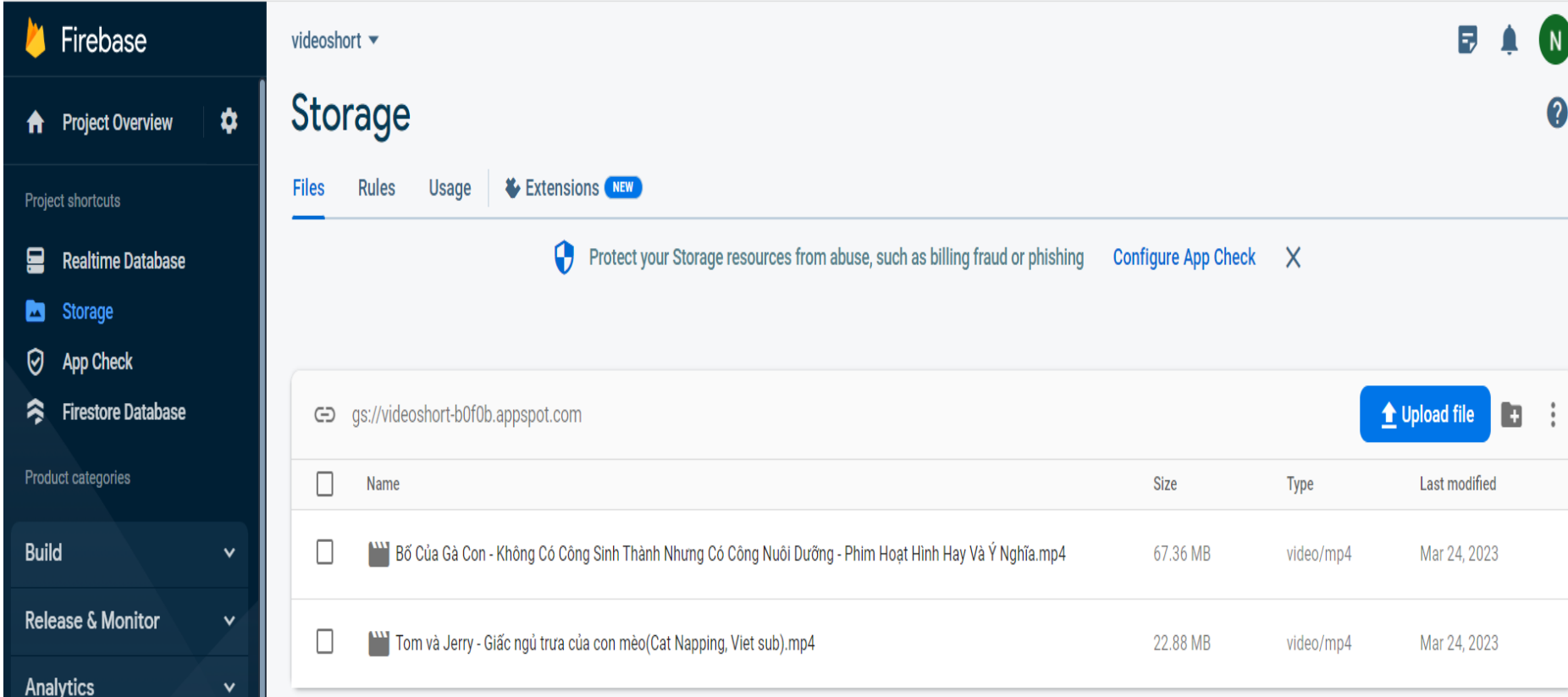
1  {
2  "rules": {
3    ".read": "true",
4    ".write": "true"
5  }
6  }

```

Trung CN TT

□ Bước 3: Khởi tạo database FireBase và Storage

Chọn Build và Chọn Storage để thêm Storage vào project của mình. Sau đó upload video .mp4 lên Storage như hình



The screenshot shows the Firebase Storage console for a project named 'videoshort'. The left sidebar contains the Firebase logo and navigation links: Project Overview, Project shortcuts (Realtime Database, Storage, App Check, Firestore Database), and Product categories (Build, Release & Monitor, Analytics). The main area displays the 'Storage' tab with a warning message: 'Protect your Storage resources from abuse, such as billing fraud or phishing'. Below this, there is a link to 'Configure App Check' and an 'Upload file' button. A table lists the uploaded files:

Name	Size	Type	Last modified
Bố Của Gà Con - Không Có Công Sinh Thành Nhưng Có Công Nuôi Dưỡng - Phim Hoạt Hình Hay Và Ý Nghĩa.mp4	67.36 MB	video/mp4	Mar 24, 2023
Tom và Jerry - Giấc ngủ trưa của con mèo(Cat Napping, Viet sub).mp4	22.88 MB	video/mp4	Mar 24, 2023

□ Bước 3: Khởi tạo database FireBase và Storage

Nhấp chuột vào file đã upload, bên phải xuất hiện Token và nhấp chuột vào mã token để lấy đường dẫn file.

Vào Realtime Database để thêm bảng dữ liệu truy vấn như sau, url chính là đường dẫn token

File location

Storage location

gs://videoshort-b0f0b.appspot.com/Tom và Jerry - Giấc ngủ trưa của con mèo(Cat Napping, Viet sub).mp4

Access token [Revoke](#)

14a80f16-1124-4abd-8095-1df3f03da1af

[Create](#)

Click to copy download URL with token

□ Bước 4: Tạo Model và Viết Adapter

```
import java.io.Serializable;

public class Video1Model implements Serializable {
    private String title;
    private String desc;
    private String url;
```

Các biến trùng tên với các key đã tạo trên Firebase nhé

□ Bước 4: Tạo Model và Viết Adapter

```
public class VideosFireBaseAdapter extends FirebaseRecyclerAdapter<Video1Model, VideosFireBaseAdapter.MyHolder>{

    private boolean isFav = false;

    public VideosFireBaseAdapter(@NonNull FirebaseRecyclerOptions<Video1Model> options) {
        super(options);
    }

    public class MyHolder extends RecyclerView.ViewHolder{

        private VideoView videoView;
        private ProgressBar videoProgressBar;
        private TextView textVideoTitle;
        private TextView textVideoDescription;
        private ImageView imPerson, favorites, imShare, imMore;
        public MyHolder(@NonNull View itemView) {
            super(itemView);
            videoView = itemView.findViewById(R.id.videoView);
            videoProgressBar = itemView.findViewById(R.id.videoProgressBar);
            textVideoTitle = itemView.findViewById(R.id.textVideoTitle);
            textVideoDescription = itemView.findViewById(R.id.textVideoDescription);
            imPerson = itemView.findViewById(R.id.imPerson);
            favorites = itemView.findViewById(R.id.favorites);
            imShare = itemView.findViewById(R.id.imShare);
            imMore = itemView.findViewById(R.id.imMore);
        }
    }
}
```

□ Bước 4: Tạo Model và Viết Adapter

```
@NonNull
@Override
public MyHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.single_video_row, parent, attachToRoot: false);
    return new MyHolder(view);
}
```

```
@Override
protected void onBindViewHolder(@NonNull MyHolder holder, int position, @NonNull Video1Model model) {
    //Video1Model videoModel = videoList.get(position);
    holder.textVideoTitle.setText(model.getTitle());
    holder.textVideoDescription.setText(model.getDesc());
    //holder.videoView.setVideoPath(videoModel.getUrl());
    holder.videoView.setVideoURI(Uri.parse(model.getUrl()));
    holder.videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {
            holder.videoProgressBar.setVisibility(View.GONE);
            mp.start();
            float videoRatio = mp.getVideoWidth() / (float) mp.getVideoHeight();
            float screenRatio = holder.videoView.getWidth() / (float) holder.videoView.getHeight();
            float scale = videoRatio / screenRatio;
            if(scale >= 1f){
                holder.videoView.setScaleX(scale);
            }else{
                holder.videoView.setScaleY(1f/scale);
            }
        }
    });
}
```

```
holder.videoView.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mp) {
        mp.start();
    }
});
holder.favorites.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!isFav){
            holder.favorites.setImageResource(R.drawable.ic_fill_favorite);
            isFav = true;
        }
        else{
            holder.favorites.setImageResource(R.drawable.ic_favorite);
            isFav = false;
        }
    }
});
}
```

□ Bước 5: Viết hàm trong MainActivity

```
public class VideoShortFireBaseActivity extends AppCompatActivity {
    private ViewPager2 viewPager2;
    private VideosFireBaseAdapter videosAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_vs_firebase);
        viewPager2 = findViewById(R.id.vpager);
        getVideos();
    }

    private void getVideos() {
        /**set database*/
        DatabaseReference mDataBase = FirebaseDatabase.getInstance().getReference(path: "videos");
        FirebaseRecyclerOptions<Video1Model> options = new FirebaseRecyclerOptions.Builder<Video1Model>()
            .setQuery(mDataBase, Video1Model.class).build();
        /**set adapter*/
        videosAdapter = new VideosFireBaseAdapter(options);
        viewPager2.setOrientation(ViewPager2.ORIENTATION_VERTICAL);
        viewPager2.setAdapter(videosAdapter);
    }

    @Override
    protected void onStart() {
        super.onStart();
        videosAdapter.startListening();
    }
}
```

Tên bảng trên Firebase

```
@Override
protected void onStop() {
    super.onStop();
    videosAdapter.stopListening();
}

@Override
protected void onResume() {
    super.onResume();
    videosAdapter.notifyDataSetChanged();
}
```

□ Bước 1: Thêm thư viện

Mở file build.gradle (Module: app) thêm vào dependency

// Network & Retrofit

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

//Gson

implementation 'com.google.code.gson:gson:2.10.1'

□ Mở file AndroidManifest.xml và thêm

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

□ Mở file AndroidManifest.xml và thêm vào thẻ mở Application

android:usesCleartextTraffic="true"

- Bước 2: Thiết kế giao diện giống bước 2 ví dụ 1 nhé
- Bước 3: Tạo Model

```
import java.io.Serializable;
```

```
public class VideoModel implements Serializable {
    private int id;
    private String title;
    private String description;
    private String url;

```

```
    public class MessageVideoModel implements Serializable {
        private boolean success;
        private String message;
        private List<VideoModel> result;
    }
}
```

□ Bước 4: Viết Adapter

```
public class VideosAdapter extends RecyclerView.Adapter<VideosAdapter.MyViewHolder> {
    private Context context;
    private List<VideoModel> videoList;
    private boolean isFav = false;

    public VideosAdapter(Context context, List<VideoModel> videoList) {
        this.context = context;
        this.videoList = videoList;
    }

    @NonNull
    @Override
    public VideosAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.single_video_row, parent, attachToRoot: false);
        return new MyViewHolder(view);
    }
}
```

□ Bước 4: Viết Adapter

```
@Override
public void onBindViewHolder(@NonNull VideosAdapter.MyViewHolder holder, int position) {
    VideoModel videoModel = videoList.get(position);
    holder.textVideoTitle.setText(videoModel.getTitle());
    holder.textVideoDescription.setText(videoModel.getDescription());
    //holder.videoView.setVideoPath(videoModel.getUrl());
    holder.videoView.setVideoURI(Uri.parse(videoModel.getUrl()));
    holder.videoView.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {
            holder.videoProgressBar.setVisibility(View.GONE);
            mp.start();
            float videoRatio = mp.getVideoWidth() / (float) mp.getVideoHeight();
            float screenRatio = holder.videoView.getWidth() / (float) holder.videoView.getHeight();
            float scale = videoRatio / screenRatio;
            if (scale >= 1f) {
                holder.videoView.setScaleX(scale);
            } else {
                holder.videoView.setScaleY(1f / scale);
            }
        }
    });
    holder.videoView.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
        @Override
        public void onCompletion(MediaPlayer mp) {
            holder.videoView.setMediaController(new MediaController(context));
            holder.videoView.requestFocus();
            mp.start();
        }
    });
};
```


□ Bước 3: Viết Adapter

```
holder.favorites.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!isFav){
            holder.favorites.setImageResource(R.drawable.ic_fill_favorite);
            isFav = true;
        }
        else{
            holder.favorites.setImageResource(R.drawable.ic_favorite);
            isFav = false;
        }
    }
});
}
```

□ Bước 3: Viết Adapter

```
@Override
public int getItemCount() {
    if (videoList != null){
        return videoList.size();
    }
    return 0;
}

public class MyViewHolder extends RecyclerView.ViewHolder{

    private VideoView videoView;
    private ProgressBar videoProgressBar;
    private TextView textVideoTitle;
    private TextView textVideoDescription;
    private ImageView imPerson, favorites, imShare, imMore;
    public MyViewHolder(@NonNull View itemView) {
        super(itemView);
        videoView = itemView.findViewById(R.id.videoView);
        videoProgressBar = itemView.findViewById(R.id.videoProgressBar);
        textVideoTitle = itemView.findViewById(R.id.textVideoTitle);
        textVideoDescription = itemView.findViewById(R.id.textVideoDescription);
        imPerson = itemView.findViewById(R.id.imPerson);
        favorites = itemView.findViewById(R.id.favorites);
        imShare = itemView.findViewById(R.id.imShare);
        imMore = itemView.findViewById(R.id.imMore);
    }
}
```

□ Bước 5: RetrofitClient gọi API

```
public interface APIService {

    Gson gson = new GsonBuilder().setDateFormat("yyyy MM dd HH:mm:ss").create();

    APIService servieapi = new Retrofit.Builder()
        .baseUrl("http://app.iotstar.vn/appfoods/") Retrofit.Builder
        .addConverterFactory(GsonConverterFactory.create(gson))
        .build() Retrofit
        .create(APIService.class);

    @GET("getvideos.php")
    Call<MessageVideoModel> getVideos();
}
```

□ Bước 6: Viết hàm trong MainActivity

```
public class MainActivity extends AppCompatActivity {
    private ViewPager2 viewPager2;
    private VideosAdapter videosAdapter;
    private List<VideoModel> list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_main);
        viewPager2 = findViewById(R.id.vpager);
        list = new ArrayList<>();
        getVideos();
    }
    private void getVideos() {
        ApiService.servieapi.getVideos().enqueue(new Callback<MessageVideoModel>() {
            @Override
            public void onResponse(Call<MessageVideoModel> call, Response<MessageVideoModel> response) {
                list = response.body().getResult();
                //Log.d("TAG", list.get(0).getUrl());
                videosAdapter = new VideosAdapter(getApplicationContext(), list);
                viewPager2.setOrientation(ViewPager2.ORIENTATION_VERTICAL);
                viewPager2.setAdapter(videosAdapter);
            }
            @Override
            public void onFailure(Call<MessageVideoModel> call, Throwable t) {
                Log.d( tag: "TAG", t.getMessage());
            }
        });
    }
}
```

- Nguyễn Hữu Trung
- 0908617108
- trungnh@hcmute.edu.vn
- utex.hcmute.edu.vn