TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

****

BÀI TẬP LỚN ANDROI

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

THÁI NGUYÊN 2024-2025

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

****

ĐỒ ÁN MÔN HỌC PHÂN TÍCH THIẾT KẾ HỆ THỐNG

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

|  |  |  |
| --- | --- | --- |
| GIÁO VIÊN HƯỚNG DẪN | : | ĐÕ DUY CỐP |
| HỌ VÀ TÊN SINH VIÊN | : | NGUYỄN VĂN SONG |
| LỚP | : | K57KMT |
| MSSV | : | K215480106043 |

THÁI NGUYÊN 2024-2025

|  |  |
| --- | --- |
| TRƯỜNG ĐHKTCN | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM |
| KHOA ĐIỆN TỬ | *Độc lập - Tự do - Hạnh phúc* |

Sinh viên: Nguyễn Văn Song

Lớp: K57.KMT.01 Khoá: 2021- 2025

Bộ môn: Công Nghệ Thông Tin

Giáo viên hướng dẫn: Đỗ Duy Cốp

1. Bài Tập Lớn

2. Nội dung các phần thuyết minh và tính toán

* Khảo sát, phân tích hiện trạng của hệ thông
* Thiết kế chương trình

3. Các sản phẩm, kết quả :

* Thuyết minh báo cáo
* Demo phần mềm

4. Ngày giao nhiệm vụ: 08/01/2025

5. Ngày hoàn thành nhiệm vụ: 5/03/2025

|  |  |  |
| --- | --- | --- |
| BCN KHOA | TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN |
| *(Ký và ghi rõ họ tên)* | *(Ký và ghi rõ họ tên)* | *(Ký và ghi rõ họ tên)* |

|  |  |
| --- | --- |
| TRƯỜNG ĐHKTCN | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM |
| KHOA ĐIỆN TỬ | *Độc lập – Tự do – Hạnh phúc* |

**Sinh viên:** Nguyễn Văn Song

**Lớp:** K57KMT

**GVHD:** Đỗ Duy Cốp

Bài Tập Lớn

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Xếp loại:……………… Điểm :………

Thái Nguyên, ngày….tháng…..năm 20....

GIÁO VIÊN HƯỚNG DẪN

# MỤC LỤC

Contents

[MỤC LỤC 4](#_Toc191326469)

[LỜI CAM ĐOAN 6](#_Toc191326470)

[LỜI NÓI ĐẦU 7](#_Toc191326471)

[BÀI TẬP 1 8](#_Toc191326472)

[1. Phân Tích 8](#_Toc191326473)

[2. Các Bước Làm: Ảnh Chụp (S) + Mô Tả 9](#_Toc191326474)

[3. Nhận Xét Kết Quả 13](#_Toc191326475)

[BÀI TẬP 2 14](#_Toc191326476)

[1. Phân tích yêu cầu 14](#_Toc191326477)

[2. Các Bước Làm: Ảnh Chụp (S) + Mô Tả 14](#_Toc191326478)

[Bước 1: Tạo Giao Diện Người Dùng 14](#_Toc191326479)

[1.Giao diện chính (activity\_main.xml): 14](#_Toc191326480)

[2.Giao diện sau khi thông báo được nhấn (activity\_after\_notification.xml): 15](#_Toc191326481)

[Bước 3: Tạo Kênh Thông Báo 16](#_Toc191326482)

[1.Tạo kênh thông báo: 16](#_Toc191326483)

[Bước 4: Kiểm Tra Quyền và Gửi Thông Báo 17](#_Toc191326484)

[1.Kiểm tra quyền: 17](#_Toc191326485)

[2.Gửi thông báo: 17](#_Toc191326486)

[Bước 5: Xử Lý Kết Quả Cấp Quyền 18](#_Toc191326487)

[1.Yêu cầu cấp quyền: 18](#_Toc191326488)

[2.Xử lý kết quả yêu cầu quyền: 18](#_Toc191326489)

[Bước 6: Kiểm Tra và Gửi Thông Báo 19](#_Toc191326490)

[1.Nhấn nút và gửi thông báo: 19](#_Toc191326491)

[3. Nhận Xét Kết Quả 19](#_Toc191326492)

[BÀI TẬP 3 : MITAPP 20](#_Toc191326493)

[1. Mục tiêu Ứng Dụng 20](#_Toc191326494)

[2. Phân Tích Các Bước Thực Hiện 20](#_Toc191326495)

[Bước 1: Tạo Dự Án Mới 20](#_Toc191326496)

[Bước 2: Thiết Kế Giao Diện Giới Thiệu Thông Tin Cá Nhân 20](#_Toc191326497)

[Bước 3: Xử Lý Logic Phát Nhạc 21](#_Toc191326498)

[Bước 4: Thiết Kế Màn Hình Giải Toán 22](#_Toc191326499)

[Bước 5: Xử Lý Logic Giải Toán 23](#_Toc191326500)

[Bước 6: Điều Hướng Giữa Các Màn Hình 24](#_Toc191326501)

[Bước 7: Kiểm Tra và Hoàn Thiện 24](#_Toc191326502)

[3. Kết Luận 25](#_Toc191326503)

# LỜI CAM ĐOAN

Tôi xin cam đoan bài tập lớn này là công trình nghiên cứu của riêng tôi. Các số liệu sử dụng trong luận văn là trung thực. Các kết quả nghiên cứu được trình bày trong đồ án chưa từng được công bố tại bất kỳ công trình nào khác.

Tên sinh viên

# LỜI NÓI ĐẦU

Android Studio là môi trường phát triển tích hợp (IDE) chính thức được Google phát triển dành riêng cho việc tạo ra ứng dụng Android. Được xây dựng trên nền tảng IntelliJ IDEA, Android Studio cung cấp một bộ công cụ mạnh mẽ và dễ sử dụng, giúp các nhà phát triển thiết kế, kiểm tra và tối ưu hóa ứng dụng Android một cách hiệu quả và thuận tiện.

Android Studio được trang bị tất cả các công cụ mà bạn cần để phát triển một ứng dụng Android từ những bước đầu tiên đến khi hoàn thiện, cho phép bạn tập trung vào sáng tạo và nâng cao chất lượng sản phẩm. Nhờ đó, Android Studio là lựa chọn tối ưu cho các lập trình viên Android từ người mới bắt đầu cho đến những chuyên gia.

Với sự phát triển mạnh mẽ của hệ sinh thái Android, việc sử dụng Android Studio sẽ giúp bạn xây dựng các ứng dụng không chỉ đáp ứng nhu cầu hiện tại mà còn dễ dàng mở rộng và cập nhật trong tương lai.

# BÀI TẬP 1

## 1. Phân Tích

Ứng dụng này là một ứng dụng đơn giản để dự đoán số tiền mà người dùng có thể kiếm được dựa trên tên và số tiền nhập vào. Dưới đây là các chức năng và các bước bạn cần thực hiện để phát triển ứng dụng:

Mục tiêu:

Ứng dụng sẽ yêu cầu người dùng nhập tên và số tiền. Sau đó, khi nhấn nút "Bói ngay", ứng dụng sẽ hiển thị kết quả dự đoán dưới dạng một thông báo (TextView).

Các bước thực hiện:

1. Tạo giao diện người dùng (UI) trong activity\_main.xml.
   * EditText để nhập tên và số tiền.
   * Button để kích hoạt dự đoán.
   * TextView để hiển thị kết quả dự đoán.
2. Viết mã xử lý logic trong MainActivity.java.
   * Lấy dữ liệu từ người dùng (tên và số tiền).
   * Xử lý kết quả dự đoán.
   * Hiển thị kết quả dự đoán trong TextView.
3. Kiểm tra ứng dụng trên thiết bị giả lập hoặc điện thoại thật.

## 2. Các Bước Làm: Ảnh Chụp (S) + Mô Tả

Bước 1: Tạo Giao Diện Người Dùng

Ứng dụng Android này có giao diện đơn giản với các thành phần sau:

* EditText: Nhập tên người dùng.
* EditText: Nhập số tiền.
* Button: Khi nhấn vào sẽ hiển thị kết quả "bói toán".
* TextView: Hiển thị kết quả dự đoán.

<!-- activity\_main.xml -->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<EditText

android:id="@+id/edtName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Nhập tên bạn vào"

android:inputType="textPersonName"

android:padding="8dp" />

<EditText

android:id="@+id/edtIncome"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Nhập số tiền"

android:inputType="number"

android:padding="8dp" />

<Button

android:id="@+id/btnPredict"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Bói ngay"

android:layout\_gravity="center\_horizontal"

android:padding="8dp"

android:marginTop="16dp" />

<TextView

android:id="@+id/txtResult"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=""

android:padding="8dp"

android:textSize="16sp"

android:layout\_marginTop="16dp" />

</LinearLayout>

Bước 2: Xử Lý Logic trong MainActivity.java

public class MainActivity extends AppCompatActivity {

    private EditText edtName, edtIncome;

    private Button btnPredict;

    private TextView txtResult;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        // Ánh xạ các thành phần giao diện

        edtName = findViewById(R.id.edtName);

        edtIncome = findViewById(R.id.edtIncome);

        btnPredict = findViewById(R.id.btnPredict);

        txtResult = findViewById(R.id.txtResult);

        // Xử lý sự kiện khi nhấn nút Bói ngay

        btnPredict.setOnClickListener(v -> {

            String name = edtName.getText().toString();

            String income = edtIncome.getText().toString();

            // Kiểm tra dữ liệu người dùng nhập vào

            if (name.isEmpty() || income.isEmpty()) {

                txtResult.setText("Vui lòng nhập đầy đủ thông tin!");

            } else {

                int incomeValue = Integer.parseInt(income);

                // Logic dự đoán (ví dụ: tính toán đơn giản)

                String prediction = "Chào " + name + ", số tiền của bạn là " + incomeValue + " VNĐ.";

                if (incomeValue > 1000000) {

                    prediction += " Bạn có khả năng giàu có trong năm nay!";

                } else {

                    prediction += " Hãy tiếp tục nỗ lực để đạt được mục tiêu!";

                }

                // Hiển thị kết quả

                txtResult.setText(prediction);

            }

        });

    }

}

Bước 3: Chạy Ứng Dụng

* Sau khi viết mã, nhấn nút Run (hoặc Shift + F10) để chạy ứng dụng trên giả lập hoặc thiết bị thực tế.

## 3. Nhận Xét Kết Quả

* Hoàn Thành: Ứng dụng sẽ lấy tên và số tiền từ người dùng, thực hiện một logic dự đoán đơn giản và hiển thị kết quả trong TextView. Các thao tác diễn ra suôn sẻ và không có lỗi xảy ra.
* Đánh Giá: Ứng dụng đã hoàn thành với giao diện đơn giản và dễ sử dụng. Tuy nhiên, bạn có thể nâng cấp thêm các tính năng khác như kiểm tra nhập liệu hợp lệ, tối ưu hóa giao diện người dùng hoặc cải tiến logic dự đoán.

# BÀI TẬP 2

## 1. Phân tích yêu cầu

* Gửi thông báo khi người dùng nhấn nút "Send Notification".
* Đảm bảo người dùng có quyền gửi thông báo (từ Android 13 trở lên).
* Chuyển người dùng đến một màn hình mới sau khi họ nhấn vào thông báo.

## 2. Các Bước Làm: Ảnh Chụp (S) + Mô Tả

## Bước 1: Tạo Giao Diện Người Dùng

### 1.Giao diện chính (activity\_main.xml):

* + Tạo một giao diện chứa một Button để gửi thông báo. Bạn sẽ sử dụng ConstraintLayout hoặc LinearLayout để căn chỉnh các phần tử.
  + Giao diện chính sẽ có một nút bấm "Send Notification", khi nhấn vào nút này, ứng dụng sẽ gửi thông báo cho người dùng.

Code:

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white">

<Button

android:id="@+id/btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Send Notification"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

### 2.Giao diện sau khi thông báo được nhấn (activity\_after\_notification.xml):

* + Tạo giao diện cho màn hình hiển thị khi người dùng nhấn vào thông báo, ví dụ một TextView với nội dung "Welcome To GeeksforGeeks".

Code:

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:background="@color/white">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Welcome To GeeksforGeeks"

android:textSize="15sp"

android:textStyle="bold" />

</LinearLayout>

## Bước 3: Tạo Kênh Thông Báo

### 1.Tạo kênh thông báo:

* + Kênh thông báo chỉ cần thiết trên Android 8.0 (API level 26) và các phiên bản mới hơn. Vì vậy, ta cần tạo kênh thông báo để các thông báo có thể được hiển thị.

Code:

private void createNotificationChannel() {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

NotificationChannel notificationChannel = new NotificationChannel(

channelId, description, NotificationManager.IMPORTANCE\_HIGH

);

notificationChannel.enableLights(true);

notificationChannel.setLightColor(Color.GREEN);

notificationChannel.enableVibration(true);

NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);

if (notificationManager != null) {

notificationManager.createNotificationChannel(notificationChannel);

}

}

}

## Bước 4: Kiểm Tra Quyền và Gửi Thông Báo

### 1.Kiểm tra quyền:

* + Trước khi gửi thông báo, ứng dụng phải kiểm tra quyền POST\_NOTIFICATIONS trên Android 13 (API level 33) và các phiên bản mới hơn. Nếu chưa có quyền, yêu cầu người dùng cấp quyền.

### 2.Gửi thông báo:

* + Khi người dùng nhấn nút "Send Notification", ứng dụng sẽ gửi thông báo với PendingIntent để mở màn hình AfterNotification.

Code:

private void sendNotification() {

Intent intent = new Intent(this, AfterNotification.class);

PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG\_UPDATE\_CURRENT);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, channelId)

.setSmallIcon(R.drawable.gfg\_logo)

.setContentIntent(pendingIntent)

.setAutoCancel(true)

.setPriority(NotificationCompat.PRIORITY\_HIGH);

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);

notificationManager.notify(notificationId, builder.build());

}

## Bước 5: Xử Lý Kết Quả Cấp Quyền

### 1.Yêu cầu cấp quyền:

* + Khi người dùng chưa cấp quyền gửi thông báo, ứng dụng sẽ yêu cầu người dùng cấp quyền POST\_NOTIFICATIONS thông qua phương thức requestPermissions().

### 2.Xử lý kết quả yêu cầu quyền:

* + Nếu quyền được cấp, gửi thông báo ngay lập tức. Nếu không được cấp, hiển thị thông báo lỗi.

Code:

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

if (requestCode == REQUEST\_NOTIFICATION\_PERMISSION) {

if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

sendNotification();

} else {

Toast.makeText(this, "Permission denied! Cannot send notifications.", Toast.LENGTH\_SHORT).show();

}

}

}

## Bước 6: Kiểm Tra và Gửi Thông Báo

### 1.Nhấn nút và gửi thông báo:

* + Khi nút "Send Notification" được nhấn, ứng dụng sẽ thực hiện kiểm tra quyền, nếu quyền đã có, thông báo sẽ được gửi đi.

## 3. Nhận Xét Kết Quả

Sau khi hoàn thành các bước trên, ứng dụng của bạn sẽ có khả năng:

* Kiểm tra quyền thông báo.
* Gửi thông báo với kênh riêng biệt.
* Điều hướng người dùng đến màn hình mới khi nhấn vào thông báo.

Nhận xét:

* Quy trình đã được thực hiện đầy đủ, ứng dụng hoạt động ổn định với các chức năng chính như gửi thông báo và chuyển hướng người dùng khi nhấn vào thông báo.
* Tuy nhiên, có thể cải thiện thêm phần giao diện và thêm các tính năng bổ sung như tuỳ chọn âm thanh cho thông báo hoặc hiển thị ảnh trong thông báo.

# BÀI TẬP 3 : MITAPP

## 1. Mục tiêu Ứng Dụng

Ứng dụng sẽ thực hiện 3 chức năng chính:

1. Giới thiệu thông tin cá nhân (chứa ảnh và nhạc MP3).
2. Giải toán đơn giản cho học sinh từ lớp 1 đến lớp 9.
3. Điều hướng giữa các màn hình (giới thiệu và giải toán).

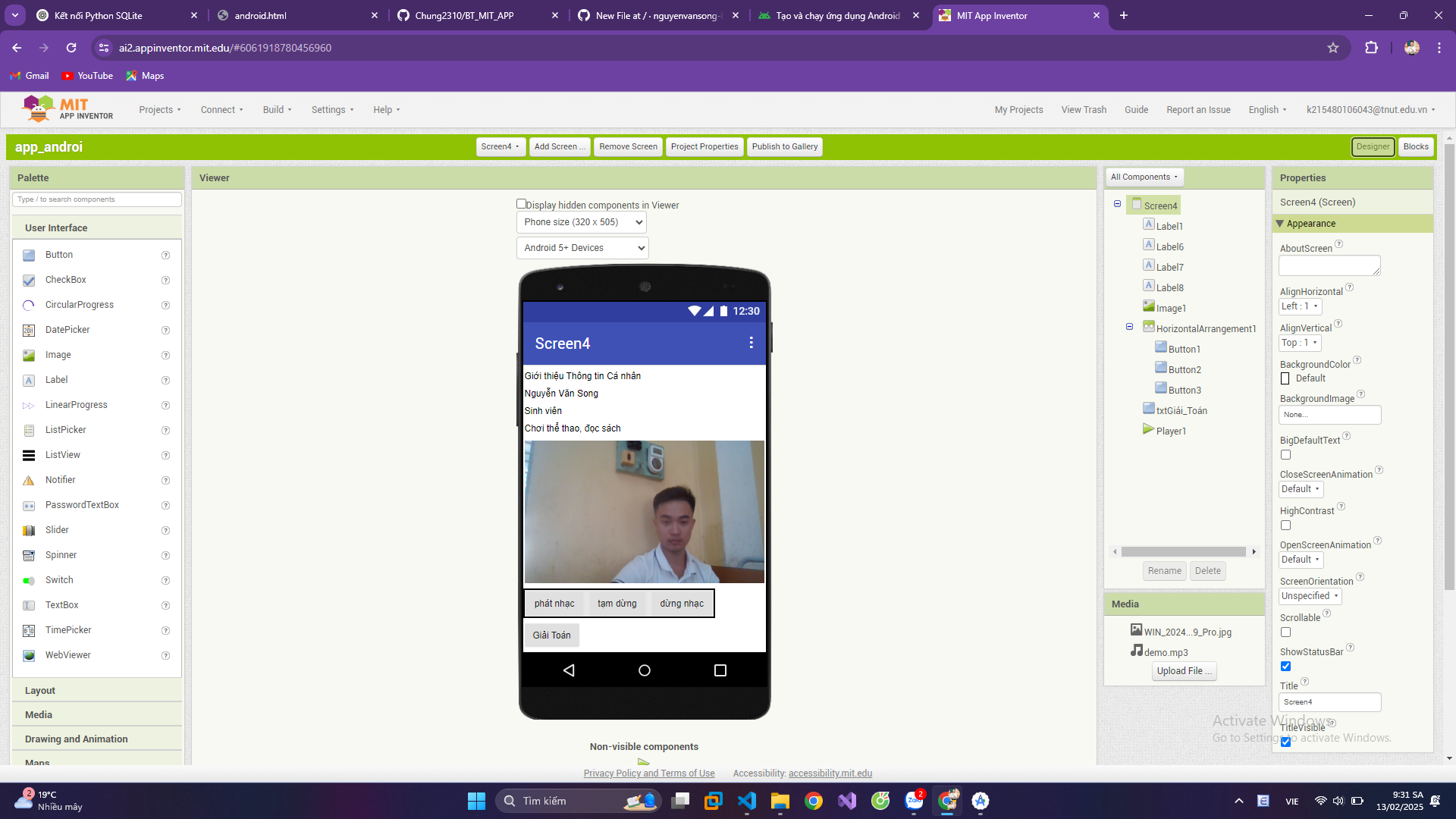
## 2. Phân Tích Các Bước Thực Hiện

## Bước 1: Tạo Dự Án Mới

* Mục tiêu: Bắt đầu bằng việc tạo một dự án mới trong MIT App Inventor.
* Cách làm: Đăng nhập vào MIT App Inventor, tạo một dự án mới và đặt tên cho nó (ví dụ "PersonalInfoApp").
* Mô tả: Đây là bước cơ bản để tạo ra môi trường phát triển ứng dụng. Sau khi tạo xong, bạn sẽ có màn hình chính để bắt đầu thêm các thành phần và logic.

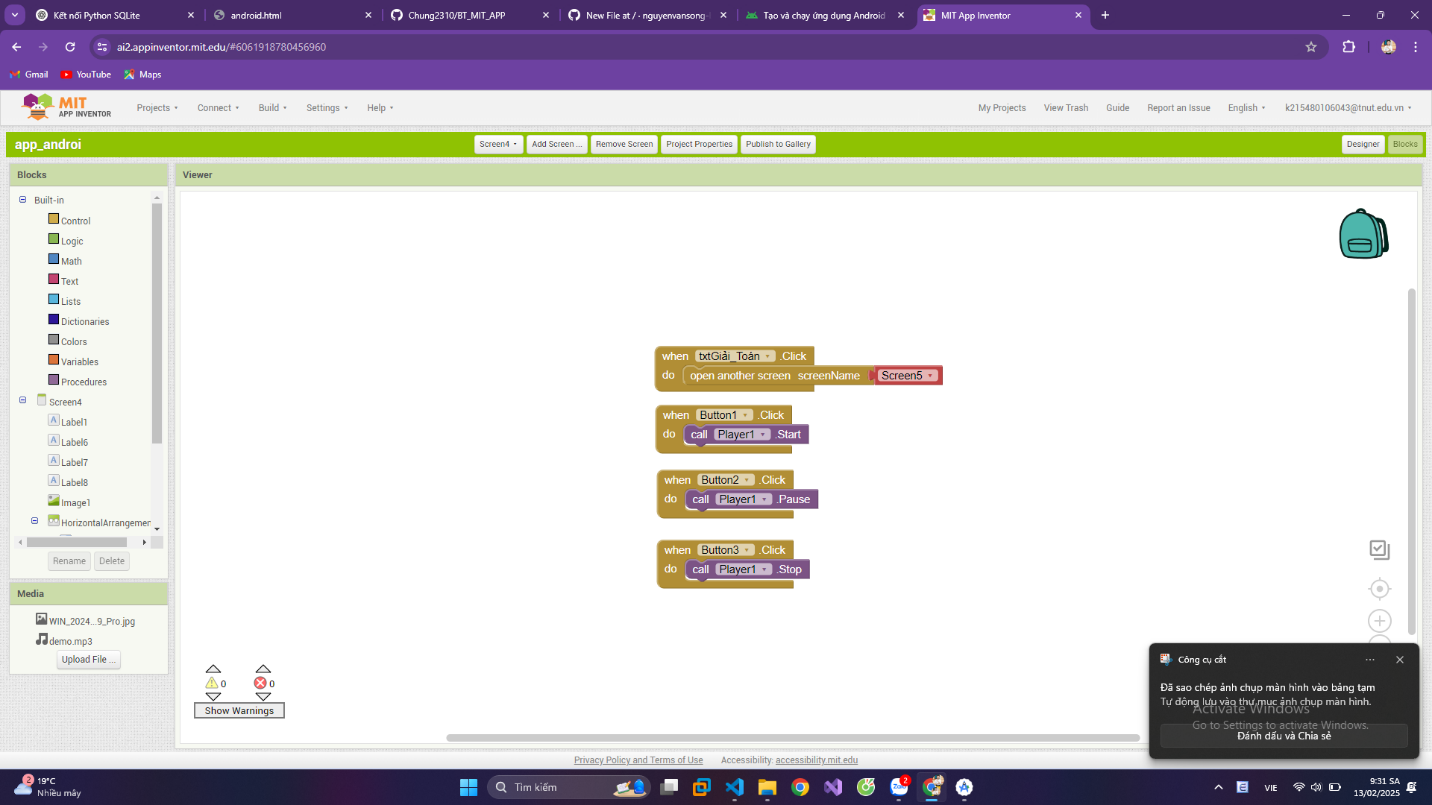
## Bước 2: Thiết Kế Giao Diện Giới Thiệu Thông Tin Cá Nhân

* Mục tiêu: Hiển thị thông tin cá nhân, bao gồm ảnh và nhạc MP3.
* Cách làm:
  + Kéo Image vào màn hình để hiển thị ảnh cá nhân.
  + Kéo Label vào màn hình để hiển thị tên và mô tả.
  + Kéo Button để tạo nút phát nhạc.
  + Tải file MP3 vào phần Media của dự án và sử dụng thành phần Player để phát nhạc khi nhấn nút.
* Mô tả: Đây là màn hình giới thiệu, nơi người dùng có thể xem thông tin cá nhân và nghe nhạc.



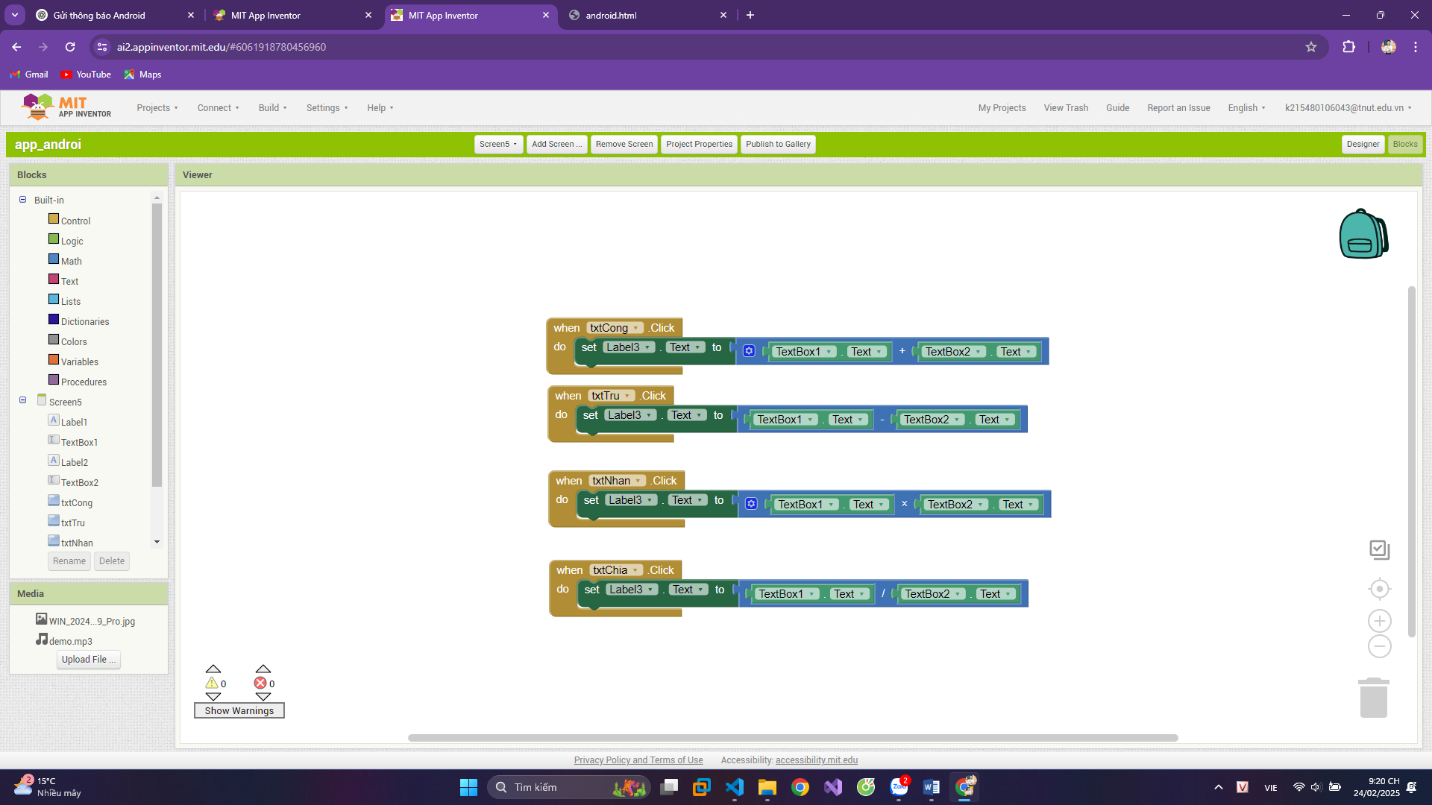
## Bước 3: Xử Lý Logic Phát Nhạc

* Mục tiêu: Khi người dùng nhấn nút, ứng dụng sẽ phát nhạc.
* Cách làm:
  + Sử dụng thành phần Player trong MIT App Inventor để phát nhạc MP3.
  + Tạo một block để điều khiển hành động phát nhạc khi nút được nhấn.
* Mô tả: Đảm bảo ứng dụng có khả năng phát nhạc ngay khi người dùng nhấn nút phát nhạc. Đây là một tính năng giải trí trong ứng dụng.



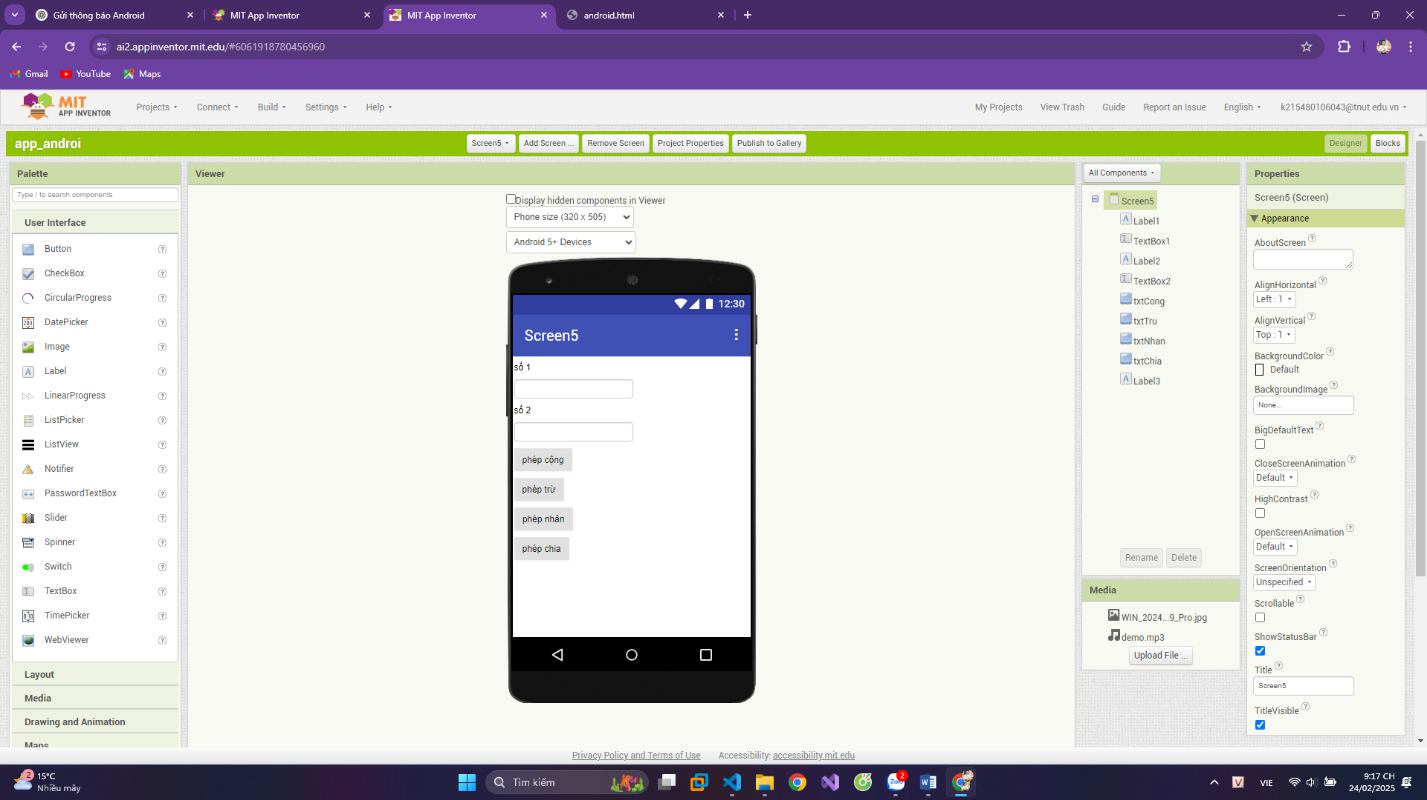
## Bước 4: Thiết Kế Màn Hình Giải Toán

* Mục tiêu: Cho phép người dùng giải các bài toán đơn giản từ lớp 1 đến lớp 9.
* Cách làm:
  + Kéo Label vào màn hình để hiển thị câu hỏi (ví dụ: 5 + 3 = ?).
  + Kéo TextBox để người dùng nhập câu trả lời.
  + Kéo Button để kiểm tra câu trả lời và hiển thị kết quả (đúng/sai).
* Mô tả: Đây là nơi học sinh có thể thử giải các phép toán cơ bản và kiểm tra kết quả của mình.



## Bước 5: Xử Lý Logic Giải Toán

* Mục tiêu: Kiểm tra câu trả lời người dùng nhập và hiển thị kết quả đúng/sai.
* Cách làm:
  + Sử dụng if-else block để so sánh câu trả lời của người dùng với kết quả đúng.
  + Tạo một Label để hiển thị thông báo "Đúng rồi!" hoặc "Sai rồi, thử lại nhé".
* Mô tả: Logic này sẽ giúp ứng dụng hoạt động như một công cụ kiểm tra toán học cho người dùng, đặc biệt là học sinh.



## Bước 6: Điều Hướng Giữa Các Màn Hình

* Mục tiêu: Chuyển giữa màn hình giới thiệu và màn hình giải toán.
* Cách làm:
  + Sử dụng block open another screen để chuyển từ màn hình giới thiệu sang màn hình giải toán.
  + Tạo thêm nút quay lại màn hình giới thiệu.
* Mô tả: Điều hướng giữa các màn hình sẽ tạo cảm giác mượt mà cho người dùng, giúp họ dễ dàng chuyển đổi giữa các tính năng của ứng dụng.

## Bước 7: Kiểm Tra và Hoàn Thiện

* Mục tiêu: Kiểm tra ứng dụng và sửa lỗi nếu có.
* Cách làm:
  + Kiểm tra ứng dụng bằng cách sử dụng AI2 Companion trên điện thoại.
  + Tinh chỉnh giao diện và sửa lỗi nếu có.
* Mô tả: Đây là bước cuối cùng trước khi hoàn thiện ứng dụng. Bạn cần đảm bảo mọi chức năng hoạt động đúng đắn và giao diện hiển thị đẹp mắt.

## 3. Kết Luận

Kết quả cuối cùng của ứng dụng này sẽ là một ứng dụng đơn giản nhưng đầy đủ chức năng:

1. Giới thiệu thông tin cá nhân với ảnh và nhạc MP3.
2. Giải toán đơn giản cho học sinh từ lớp 1 đến lớp 9.
3. Điều hướng giữa các màn hình để người dùng có thể dễ dàng chuyển giữa các phần.

Ứng dụng này giúp người dùng giải trí và học tập hiệu quả thông qua các tính năng phát nhạc và giải toán.

