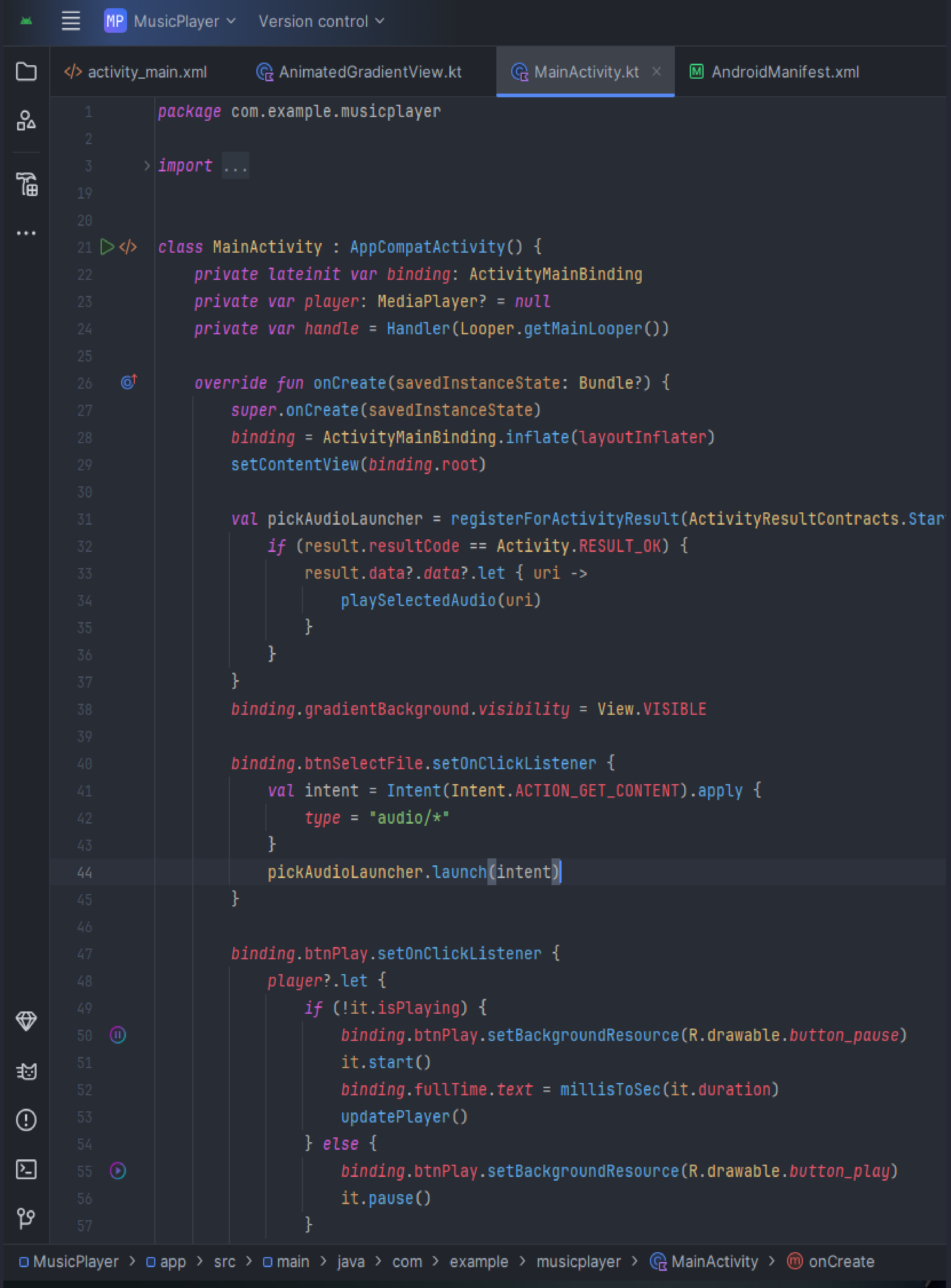


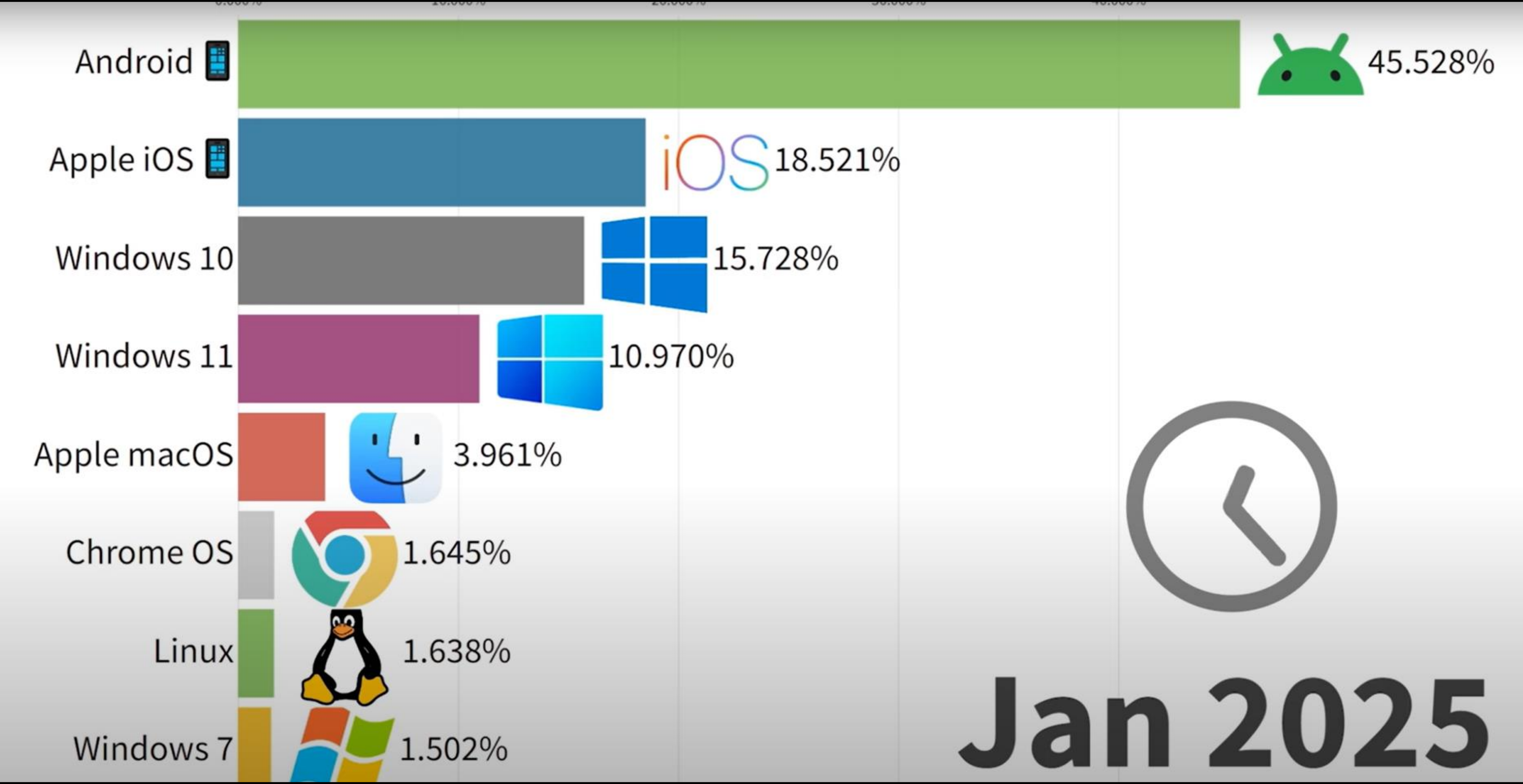
```kotlin
package com.example.musicplayer

import ...


class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private var player: MediaPlayer? = null
    private var handle = Handler(Looper.getMainLooper())

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val pickAudioLauncher = registerForActivityResult(ActivityResultContracts.Star
            if (result.resultCode == Activity.RESULT_OK) {
                result.data?.data?.let { uri ->
                    playSelectedAudio(uri)
                }
            }
        }
        binding.gradientBackground.visibility = View.VISIBLE

        binding.btnSelectFile.setOnClickListener {
            val intent = Intent(Intent.ACTION_GET_CONTENT).apply {
                type = "audio/*"
            }
            pickAudioLauncher.launch(intent)
        }

        binding.btnPlay.setOnClickListener {
            player?.let {
                if (!it.isPlaying) {
                    binding.btnPlay.setBackgroundResource(R.drawable.button_pause)
                    it.start()
                    binding.fullTime.text = millisToSec(it.duration)
                    updatePlayer()
                } else {
                    binding.btnPlay.setBackgroundResource(R.drawable.button_play)
                    it.pause()
                }
```

By Amir Erfan Eslamikia

Overview of Android Studio

Android Studio is the official **Integrated Development Environment (IDE)** for Android app development, based on IntelliJ IDEA. It provides a powerful **graphical user interface (GUI)** for building Android applications efficiently.

Android Studio simplifies app development by offering tools for debugging, profiling, and performance monitoring. It supports multiple programming languages, including **Kotlin** and **Java**, making it flexible for developers.


0.000%
10.000%
20.000%
30.000%
40.000%
Android 📱
45.528%
Apple iOS 📱
iOS
18.521%
Windows 10
15.728%
Windows 11
10.970%
Apple macOS
3.961%
Chrome OS
1.645%
Linux
1.638%
Windows 7
1.502%
Jan 2025

Key Features of Android Studio

Gradle-based Build System

Automates the build process and manages dependencies.

Intelligent Code Editor

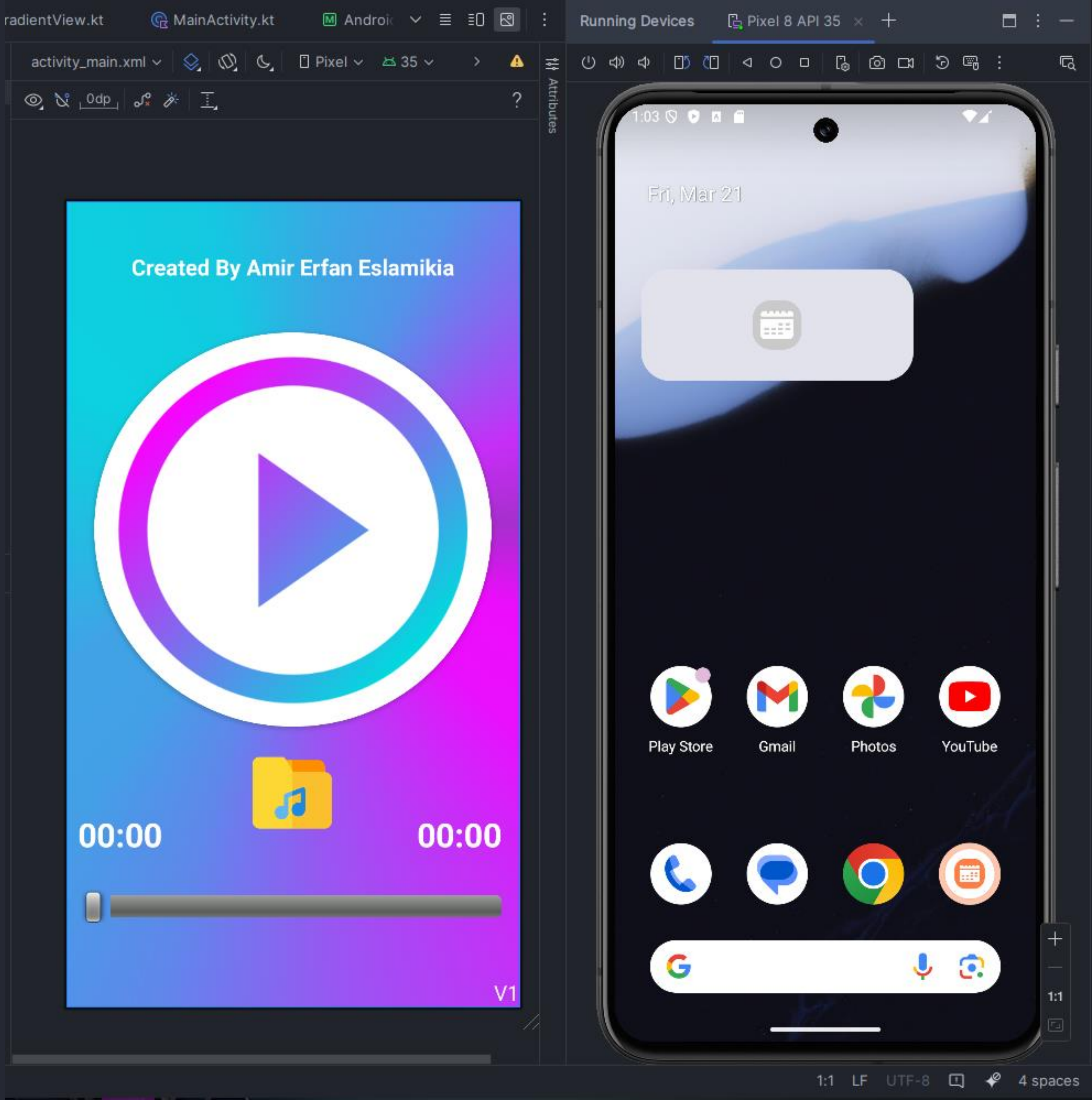
Offers smart code completion and real-time error detection.

Emulator

A built-in virtual device to test applications without a physical phone.

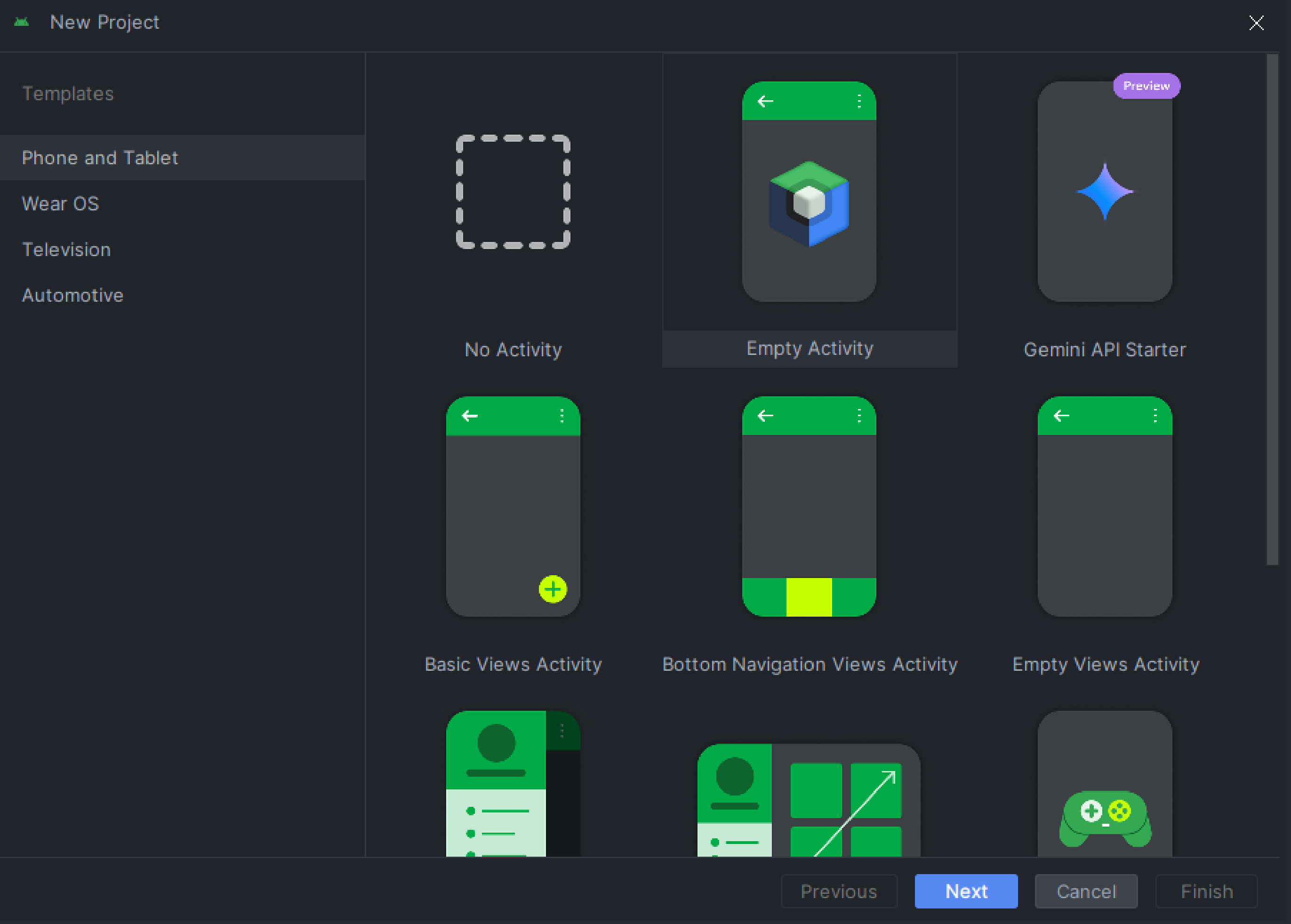
Layout Editor

Provides a **drag-and-drop** interface for designing UI elements.



Steps to Develop an Android App

1. **Install Android Studio** and set up the **SDK**.
2. Create a **new project** and choose a template.
3. Design the **user interface (UI)** using XML or the Layout Editor.
4. Write the application logic using **Kotlin** or Java.
5. Test the app using the **Emulator** or a real device.
6. Debug and optimize the code for better performance.
7. Publish the app on the **Google Play Store Or Bazar**.



Deployment and Publishing

Sign Your App

Generate signed APK or App Bundle with your unique digital key.

Test Distribution

Use internal testing tracks to share with testers before release.

Google Play Upload

Submit your app bundle through the Google Play Console.

Maintain & Update

Release updates using version control and CI/CD pipelines.

Coding in Android Studio

Programming Languages

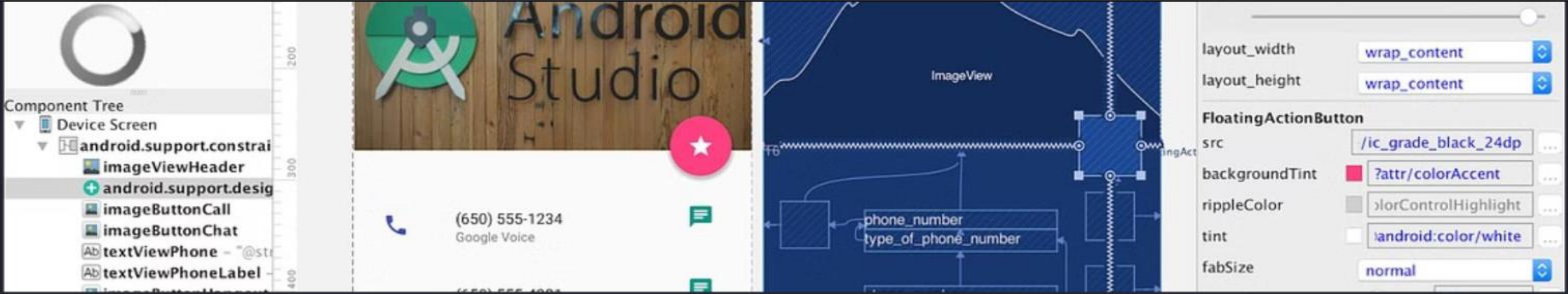
Write apps in Java or Kotlin. Google now recommends Kotlin as the preferred language.

- Auto-completion
- Syntax highlighting
- Error detection

Layout Design

Create user interfaces with XML. Each UI element is defined as an XML tag.

- Constraint layouts
- Material design components
- Resource management



layout_width wrap_content
layout_height wrap_content
src /ic_grade_black_24dp
backgroundTint ?attr/colorAccent
rippleColor ›lorControlHighlight
tint ›android:color/white
fabSize normal

Visual Design Tools

Layout Editor

Drag-and-drop UI components to design screens visually without writing XML code.

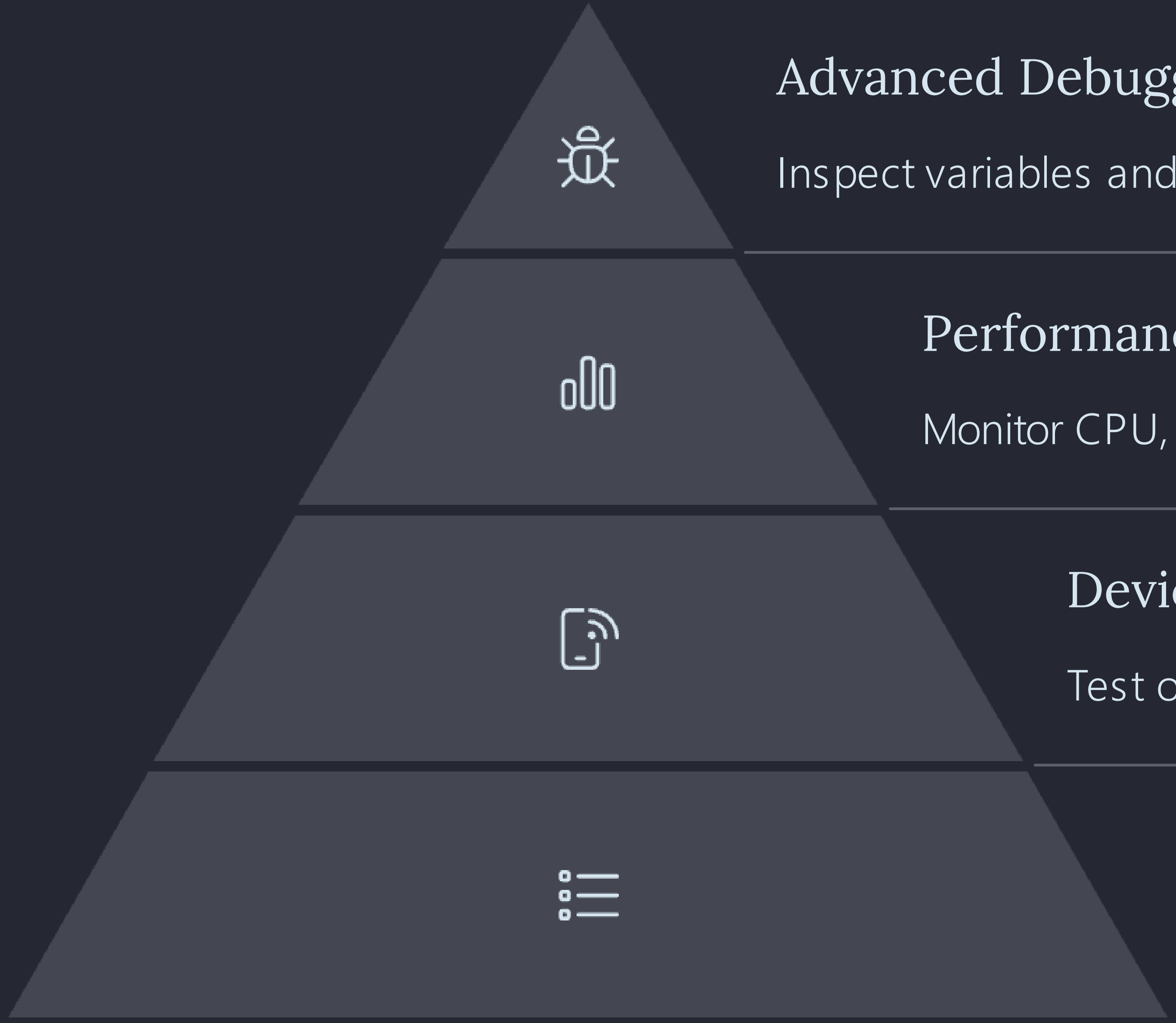
Multi-Preview

Test layouts on different screen sizes and orientations simultaneously.

Theme Editor

Customize colors, styles, and visual attributes for a consistent app appearance.

Testing and Debugging



Advanced Debugging

Inspect variables and evaluate expressions during runtime

Performance Profilers

Monitor CPU, memory, network, and battery usage

Device Testing

Test on emulators or connected physical devices

Logcat System

View detailed system logs for troubleshooting

Why Android Studio Matters

Industry Standard

Recognized as the definitive tool for Android development worldwide. Most job listings require this skill.

Constant Evolution

Regular updates bring new features and improvements. Google maintains it to work with the latest Android versions.

Powerful Community

Vast resources including tutorials, forums, and plugins. Solutions exist for almost any development challenge.

Key Vocabulary

- **Integrated Development Environment (IDE)** – A software application for programming.
- **Graphical User Interface (GUI)** – A visual interface for user interaction.
- **Drag-and-Drop** – A feature that allows users to move elements
- **SDK (Software Development Kit)** – A collection of tools, libraries, and documentation that developers use to build software.
- **Emulator** – A virtual device that simulates a real smartphone.
- **Gradle Build System** – A tool for automating the build process.
- **Layout Editor** – A feature for designing user interfaces visually.
- **Debugging** – The process of identifying and fixing errors in code.
- **Profiling** – Analyzing the performance of an application.
- **Dependencies** – External libraries or components required by a project.
- **Version Control** – A system for managing changes in code.
- **Refactoring** – The process of restructuring existing code without changing its functionality.
- **IntelliSense** – A code-completion feature that provides suggestions as you type.
- **Manifest File** – An XML file that contains essential information about an Android app, such as permissions and activities.
- **Google Play Store** – The official marketplace for Android apps.

Comprehension Questions

1. What is Android Studio, and what is its main purpose?

2. Name three key features of Android Studio and their functions.

3. Why is an emulator useful in Android development?

4. What are the main steps to develop an Android app using Android Studio?

5. How does version control help developers in Android Studio?

6. What is the role of the Android Manifest File, and why is it important?

7. How does Dependency Injection improve the modularity of an Android application?