Mobile Application Development

Muhammad Jamil

Mobile Application Development

A software development field that includes creating software for mobile devices like tablets and smartphones.

Why is it Important?

- Over **7.21 billion smartphone users** globally.
- Apps drive business, entertainment, healthcare, education, etc.

Mobile Operating Systems (OS) and their market share

Android (~70%) – Open-source, used by multiple manufacturers

iOS (~28%) – Closed ecosystem, premium market

Others (~2%) – HarmonyOS, KaiOS, etc.

History

Android (2008–Present)

iOS (2007–Present).

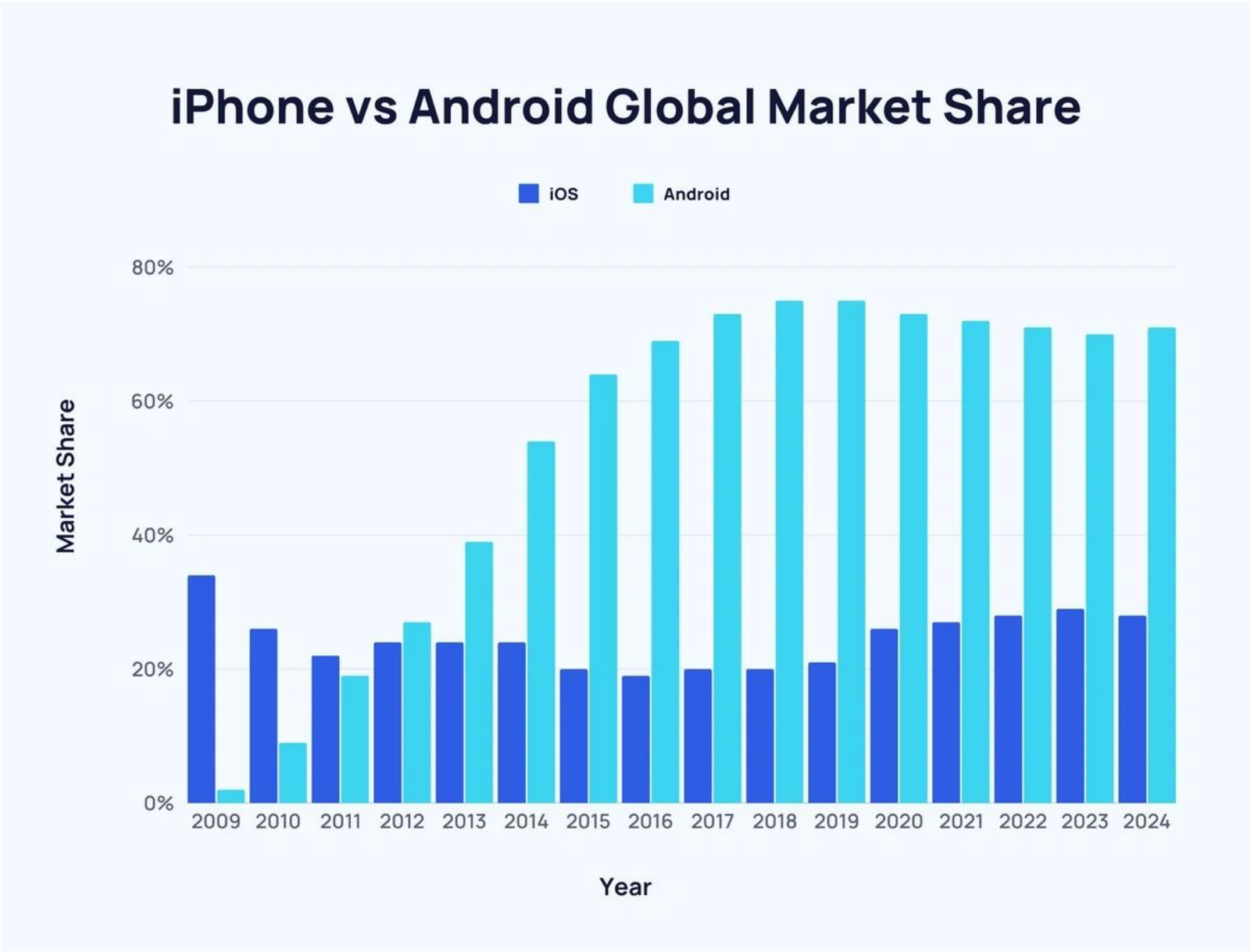
HarmonyOS (2019–Present).

Marketplace Of Android vs iOS

Android has largest market share

Globally: 70% of mobile OS

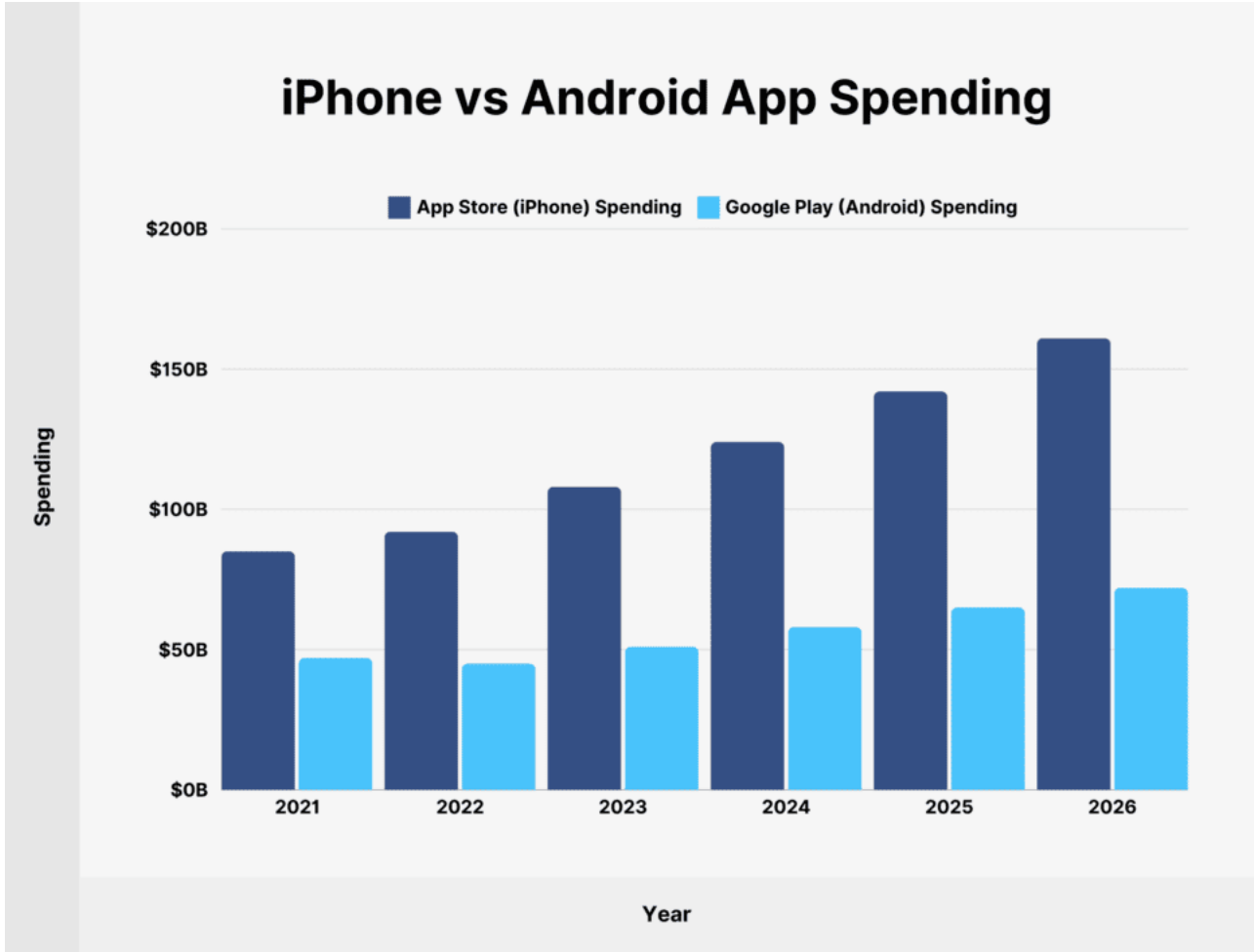
- Africa: 80%
- Asia: 84%
- Europe: 72%
- North America: 51%
- South America 88%



Marketplace of Android vs iOS

Despite Android's massive market share, iOS still makes more money

- iPhone is used by richer and more affluent users, and so, iPhone users are more likely to spend money on apps than Android users.



Android OS Overview

- Based on the Linux kernel
- Acquired by Google and later the Open Handset Alliance (OHA)

- **Features:** Open-source, customizable, supports various devices.

- **Versions:** Stock Android (Pixel), Custom ROMs (Samsung One UI, MIUI).

- **Devices:** Smartphones, Tablets, TVs, Wearables.

iOS Overview

- Based on Unix
- Created by Apple Inc. only for Apple devices (Not open-source)

- **Features:** Secure, optimized, controlled ecosystem.
- **Devices:** iPhone, iPad, Apple Watch, Apple TV.
- **App Store:** Strict guidelines, premium user base.

Types of Mobile App Development Technologies

1. **Native Development** – Built specifically for one OS.
2. **Cross-Platform** – Single codebase for multiple OS.
3. **Hybrid Apps** – Web-based apps in a native wrapper.
4. **Progressive Web Apps (PWA)** – Web apps behaving like mobile apps.

Native Development

Developed to target a single operating system, either iOS or Android OS in the OS's required programming language.

- **iOS:** Swift, Objective-C using Xcode.
- **Android:** Kotlin, Java using Android Studio.

- **Pros:** Best performance, Intuitive user experience, access to all device features.
- **Cons:** Time-consuming and costly: Requires separate development for each OS to target larger user base.

- Examples: Google Maps, Spotify, WhatsApp

Cross-Platform Development

Create apps using cross-platform frameworks, which use platform-specific SDKs (Android SDKs and iOS SDKs) from a unified API to target different operating systems

- **React Native by Meta** uses JavaScript as the programming language. (Instagram, Skype, Walmart, Airbnb)
- **Flutter by Google** uses Dart.

(Google Ads, My BMW App, New York Times)

- **Kotlin Multiplatform** uses Kotlin still using native UI and platform-specific APIs (Netflix, Yandex Apps)
- **.NET MAUI by Microsoft (formerly Xamarin)** uses C# and XAML. (Red-Point, SportsEngine)

- **Pros:** Low costs, Code reusability, Rapid development, Easier maintenance.
- **Cons:** Lower performance, Delayed platform features, Difficult integrations, Larger digital footprint, Limited UI consistency

Hybrid Apps

A hybrid application is just a web application with a lightweight native app "container" attached to it. With the help of this container, the hybrid application can benefit from native platform features and device hardware- like the calender, camera, push notification, pinch and spread functionality, and device hardware- that are not available to web application.

- **Ionic** – Uses web technologies with a native shell.

(MarketWatch, Sworkit)

- **Apache Cordova (PhoneGap)** – Wraps web apps into a native container.

(Wikipedia, Untappd)

- **Pros:** Faster Development, Cost-Effective, .
- **Cons:** Performance Issues, Limited Native Functionality

Progressive Web App

PWAs are web applications that behave like mobile apps. They run in a browser but can be installed on a device and work offline using caching mechanisms.

e.g. Twitter Lite, Pinterest , Starbucks

- **Pros:** No App Store Required, Offline Support, .
- **Cons:** Limited Native Access, Browser Dependent, Limited Push Notifications

Which platform to choose?

It depends upon lot of factors

Factor	Native	Cross-Platform	Hybrid	PWA
Performance	Best	Good	Moderate	Fast but browser-limited
User Experience (UX/UI)	Best	Good	Acceptable	Limited to browser UI
Access to Native Features	Full	Mostly Full	Limited via plugins	Minimal
Offline Support	Full	Full	Limited	Strong (Service Workers)
Development Cost	High	Medium	Medium	Low
Time to Market	Slow	Faster	Fast	Very Fast
Maintenance	Complex	Easier	Easier	Simplest
App Store Presence	Yes	Yes	Yes	No
SEO Visibility	No	No	No	Yes
Use Case Examples	Gaming, AR, VR, High-performance apps	Business apps, social media, finance apps	Content-heavy apps, enterprise apps	News, e-commerce, blogs, lightweight apps

Pre-requisites of this course

- Object Oriented Programming concepts
  - Class fundamentals
  - Inheritance concepts
  - Type casting, Access modifiers
- Core programming concepts
  - Arrays and operators
  - Control statements
- Concepts of designing using HTML/XML

Mobile app development issues/ challenges

- Supporting multiple screens
  - Multiple screen sizes
  - Screen resolutions
  - Screen orientations
- Compatibility
  - Run apps on older platform versions
- Structuring your code
- Designing According to need/market

Reference Video

Building a Mobile App in 2025: The BEST Technologies

- https://www.youtube.com/watch?v=NMb4RDpbRXs

Mobile App Development in 2025 - Choosing between Flutter, React Native and more

- https://www.youtube.com/watch?v=OzoYeouiaOA

Self Exercise

- Try installing android studio
- Setup Flutter SDKs