

**Câu 1: Hãy cho biết các nền tảng cho thiết bị di động thông minh hiện nay? Với mỗi nền tảng hãy cho biết đặc điểm, ưu và khuyết điểm.**

❖ **Android**

➤ Đặc điểm:

- Hệ điều hành mã nguồn mở, độ tùy biến cao.
- Cung cấp kho ứng dụng Google Play Store khổng lồ.
- Đa dạng các thiết bị từ bình dân đến cao cấp.
- Cập nhật phiên bản thường xuyên.

➤ Ưu điểm:

- Tùy biến cao: Người dùng có thể tùy chỉnh giao diện, cài đặt ứng dụng từ nhiều nguồn khác nhau.
- Kho ứng dụng đa dạng: Google Play Store có số lượng ứng dụng khổng lồ và miễn phí.
- Cộng đồng phát triển lớn mạnh: Nhờ mã nguồn mở, Android có một cộng đồng phát triển ứng dụng rất lớn.

➤ Nhược điểm:

- Phần mềm độc hại: Do độ mở, Android dễ bị tấn công bởi phần mềm độc hại.
- Cập nhật phân tán: Việc cập nhật Android trên các thiết bị khác nhau có thể không đồng đều.

❖ **iOS**

➤ Đặc điểm:

- Hệ điều hành đóng, được phát triển và quản lý bởi Apple.
- Tích hợp chặt chẽ với các thiết bị của Apple.
- Giao diện trực quan, dễ sử dụng.
- Bảo mật cao.

➤ Ưu điểm:

- Trải nghiệm người dùng mượt mà: iOS được tối ưu hóa để hoạt động tốt trên các thiết bị Apple.
- Bảo mật cao: Apple thường xuyên cập nhật các bản vá bảo mật cho iOS.

- Ứng dụng chất lượng cao: Các ứng dụng trên App Store thường có chất lượng cao hơn.
- Nhược điểm:
  - Đóng: Người dùng không thể tùy chỉnh hệ điều hành nhiều như Android.
  - Giá thành cao: Các thiết bị chạy iOS thường có giá cao hơn so với Android.

## Câu 2: Liệt kê các nền tảng phát triển ứng dụng di động phổ biến hiện nay và so sánh sự khác biệt chính giữa chúng.

Tiêu chí so sánh	Flutter (Dart)	React Native (JavaScript)	Xamarin (.NET)
Ngôn ngữ lập trình chính	Dart	JavaScript	C#
Môi trường phát triển	Flutter SDK, VS Code	React Native CLI, VS Code	Visual Studio
Ưu điểm	Phát triển đa nền tảng, giao diện đẹp, hiệu năng cao	Phát triển nhanh, giao diện gần native, cộng đồng lớn	Tích hợp với các công cụ .NET, chia sẻ mã nguồn giữa các nền tảng
Nhược điểm	Cộng đồng còn mới, một số tính năng chưa hoàn thiện	Hiệu năng có thể kém hơn native, kích thước ứng dụng lớn hơn	Cần máy cấu hình mạnh, cộng đồng nhỏ hơn so với các nền tảng khác
Phù hợp cho	Ứng dụng cần giao diện đẹp, hiệu năng cao, phát triển nhanh	Ứng dụng cần phát triển nhanh, giao diện gần native, ứng dụng MVP	Ứng dụng doanh nghiệp, ứng dụng muốn tận dụng lại mã nguồn .NET
Ví dụ ứng dụng nổi bật	Xianyu (Alibaba), Reflectly	Facebook, Instagram, Skype	Xamarin Forms samples

## Câu 3: Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng? So sánh với các nền tảng khác như React Native và Xamarin.

### ❖ Hiệu năng cao:

- **Widget:** Flutter sử dụng một hệ thống widget tùy chỉnh, cho phép tạo ra các giao diện phức tạp một cách mượt mà và hiệu quả.
- **Rendering:** Flutter có một engine rendering riêng, giúp đảm bảo hiệu năng cao và trải nghiệm người dùng mượt mà trên cả Android và iOS.

### ❖ Giao diện người dùng đẹp mắt:

- **Customizable:** Flutter cung cấp một bộ widget phong phú và có thể tùy chỉnh cao, cho phép các nhà phát triển tạo ra các ứng dụng với giao diện đẹp mắt và hiện đại.
- **Hot Reload:** Tính năng hot reload giúp các nhà phát triển nhanh chóng thấy được những thay đổi trong giao diện mà không cần biên dịch lại ứng dụng.

### ❖ Phát triển nhanh:

- **Dart:** Ngôn ngữ Dart được thiết kế để tạo ra các ứng dụng nhanh và hiệu quả.
- **Công cụ phát triển:** Flutter cung cấp một bộ công cụ phát triển mạnh mẽ, giúp các nhà phát triển nhanh chóng thiết kế và xây dựng ứng dụng.

### ❖ Cộng đồng lớn mạnh:

- **Hỗ trợ:** Flutter có một cộng đồng lớn và sôi động, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức.
- **Thư viện:** Có rất nhiều thư viện và package của bên thứ ba hỗ trợ cho Flutter.

### ❖ Bảng so sánh Flutter với React Native và Xamarin

Tính năng	Flutter	React Native	Xamarin
Hiệu năng	Rất cao, gần native	Tốt, nhưng có thể gặp một số vấn đề về hiệu năng trên các giao diện phức tạp	Tốt, nhưng có thể bị ảnh hưởng bởi các lớp abstraction
Giao diện người dùng	Rất tốt, tùy chỉnh cao	Tốt, nhưng có thể gặp một số hạn chế về giao diện trên một số nền tảng	Tốt, nhưng có thể không được hiện đại bằng các nền tảng khác
Phát triển	Nhanh, hot reload	Nhanh, hot reload	Tương đối chậm hơn so với Flutter và React Native
Cộng đồng	Lớn mạnh, phát triển nhanh	Rất lớn, lâu đời	Nhỏ hơn so với Flutter và React Native
Ngôn ngữ lập trình	Dart	JavaScript	C#

## Câu 4: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android và giải thích tại sao chúng lại được chọn.

### ❖ Kotlin:

- Ngôn ngữ chính thức của Android: Google đã chính thức công bố Kotlin là ngôn ngữ lập trình chính thức cho Android từ năm 2017.
- Tính năng hiện đại: Kotlin mang đến nhiều tính năng hiện đại như null safety, coroutines, function programming giúp code trở nên ngắn gọn, rõ ràng và ít lỗi hơn.
- Tương thích với Java: Kotlin có thể tương tác với code Java một cách dễ dàng, giúp các nhà phát triển Android chuyển đổi sang Kotlin một cách mượt mà.
- Cộng đồng lớn mạnh: Kotlin có một cộng đồng người dùng lớn và sôi động, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức.

### ❖ JavaScript:

- Phổ biến nhất: JavaScript là ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong phát triển web.
- Framework đa dạng: Có rất nhiều framework JavaScript như React Native, Ionic giúp phát triển ứng dụng di động một cách nhanh chóng và hiệu quả.
- Cộng đồng lớn: JavaScript có một cộng đồng người dùng khổng lồ, với rất nhiều tài liệu, thư viện và công cụ hỗ trợ.

#### ❖ C#

- Xamarin: C# được sử dụng rộng rãi trong phát triển ứng dụng di động đa nền tảng nhờ Xamarin. Xamarin cho phép chia sẻ một lượng lớn mã nguồn giữa các nền tảng khác nhau, giúp tiết kiệm thời gian và chi phí phát triển.
- .NET: C# là một phần của .NET, một framework phát triển ứng dụng mạnh mẽ và được hỗ trợ bởi Microsoft.

#### ❖ Dart

- Flutter: Dart là ngôn ngữ chính được sử dụng để phát triển ứng dụng với Flutter. Flutter là một SDK cho phép xây dựng các ứng dụng di động, web và desktop từ một cơ sở mã chung.
- Hiệu năng cao: Dart được biên dịch thành mã máy native, giúp các ứng dụng Flutter có hiệu năng cao và mượt mà.
- Giao diện người dùng đẹp mắt: Flutter cung cấp một bộ widget phong phú và có thể tùy chỉnh cao, giúp tạo ra các ứng dụng với giao diện đẹp mắt và hiện đại.

### **Câu 5: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS.**

Objective-C: Là ngôn ngữ lập trình gốc được sử dụng để phát triển các ứng dụng iOS đầu tiên.

Swift: Là ngôn ngữ lập trình chính của Apple, được giới thiệu như một ngôn ngữ lập trình hiện đại, an toàn và hiệu suất cao để thay thế cho Objective-C.

### **Câu 6: Hãy thảo luận về những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó.**

- Microsoft đã quá coi thường bộ công cụ Google
- Microsoft thay đổi quá chậm khi mà cả Android và iOS đã về đích
- Nokia không còn ưu tiên sử dụng Windows Phone

### **Câu 7: Khám phá các ngôn ngữ và công cụ để phát triển ứng dụng web trên thiết bị di động.**

Các ngôn ngữ lập trình chính:

#### ❖ JavaScript:

- Là ngôn ngữ cốt lõi của các ứng dụng web.
- Được hỗ trợ bởi tất cả các trình duyệt.
- Có nhiều framework và thư viện mạnh mẽ như React, Angular, Vue.js.

#### ❖ **TypeScript:**

- Là một siêu tập của JavaScript, cung cấp tính năng gõ tĩnh, giúp cải thiện khả năng bảo trì và phát hiện lỗi sớm.

#### ❖ **Dart:**

- Được sử dụng bởi Flutter, một framework UI đa nền tảng, cho phép xây dựng các ứng dụng di động chất lượng cao với giao diện người dùng đẹp mắt.

#### ❖ **Các công cụ và framework phổ biến:**

##### ❖ **React Native:**

- Sử dụng JavaScript và React để xây dựng các ứng dụng di động đa nền tảng (iOS và Android) với giao diện người dùng gần native.
- Cung cấp hiệu năng cao và khả năng tái sử dụng mã.

##### ❖ **Flutter:**

- Sử dụng Dart để xây dựng các ứng dụng di động, web và desktop từ một cơ sở mã chung.
- Nổi bật với hiệu năng cao, giao diện người dùng đẹp mắt và khả năng tùy chỉnh.

##### ❖ **Ionic:**

- Dựa trên Angular và Cordova, cho phép xây dựng các ứng dụng di động hybrid bằng HTML, CSS và JavaScript.
- Tích hợp sẵn nhiều thành phần UI và dễ dàng triển khai.

##### ❖ **Cordova:**

- Là một nền tảng mở, cho phép các nhà phát triển sử dụng các công nghệ web tiêu chuẩn để tạo ra các ứng dụng di động.

##### ❖ **PhoneGap:**

- Tương tự như Cordova, cung cấp một khung làm việc để xây dựng các ứng dụng di động đa nền tảng bằng HTML, CSS và JavaScript.

❖ **Các yếu tố cần cân nhắc khi chọn lựa:**

- ❖ **Nền tảng:** Nếu bạn muốn tập trung vào một nền tảng cụ thể (iOS, Android, hoặc cả hai), hãy chọn các công cụ và framework hỗ trợ tốt nhất cho nền tảng đó.
- ❖ **Hiệu năng:** Nếu ứng dụng của bạn yêu cầu hiệu năng cao, hãy cân nhắc các framework như Flutter hoặc React Native.
- ❖ **Giao diện người dùng:** Nếu bạn muốn tạo ra các ứng dụng với giao diện người dùng đẹp mắt và hiện đại, Flutter là một lựa chọn tốt.
- ❖ **Cộng đồng:** Một cộng đồng lớn mạnh sẽ cung cấp nhiều tài liệu, thư viện và hỗ trợ cho bạn.
- ❖ **Độ phức tạp của ứng dụng:** Nếu ứng dụng của bạn đơn giản, bạn có thể chọn Ionic hoặc Cordova. Nếu ứng dụng phức tạp hơn, React Native hoặc Flutter sẽ phù hợp hơn.

## **Câu 8: Nghiên cứu về nhu cầu nguồn nhân lực lập trình viên trên thiết bị di động hiện nay và những kỹ năng được yêu cầu nhiều nhất**

### **Các kỹ năng được yêu cầu nhiều nhất**

Dựa trên các thông tin tuyển dụng trên topdev.vn và các nền tảng tương tự, có thể rút ra một số kỹ năng mà các nhà tuyển dụng đang tìm kiếm ở các lập trình viên di động:

❖ **Ngôn ngữ lập trình:**

- **Swift:** Cho iOS
- **Kotlin:** Cho Android
- **JavaScript:** Cho các framework như React Native, Flutter
- **C#:** Cho Xamarin

❖ **Framework và công cụ:**

- **Xcode:** Môi trường phát triển chính thức cho iOS
- **Android Studio:** Môi trường phát triển chính thức cho Android
- **React Native:** Framework phát triển ứng dụng đa nền tảng
- **Flutter:** Framework UI đa nền tảng
- **Ionic:** Framework xây dựng ứng dụng di động hybrid

❖ **Các kiến thức khác:**

- **Kiến trúc ứng dụng:** MVC, MVVM
- **Quản lý cơ sở dữ liệu:** SQLite, Realm
- **Thiết kế giao diện người dùng:** UI/UX design
- **Kiểm thử và bảo mật ứng dụng:** Unit testing, security best practices
- **Version control:** Git
- **Cloud computing:** AWS, Google Cloud Platform

❖ **Kỹ năng mềm:**

- **Giao tiếp:** Có khả năng làm việc nhóm, trao đổi thông tin với các thành viên trong đội ngũ.
- **Giải quyết vấn đề:** Nhanh chóng tìm ra giải pháp cho các vấn đề phát sinh.
- **Học hỏi:** Luôn cập nhật kiến thức và công nghệ mới.
- **Quản lý thời gian:** Hoàn thành công việc đúng tiến độ.