

### Câu 1:

Các nền tảng phổ biến hiện nay bao gồm:

Android:

Đặc điểm: Là hệ điều hành mã nguồn mở của Google, có thị phần lớn nhất trên toàn cầu.

Ưu điểm:

Có cộng đồng lớn và nhiều tài liệu hỗ trợ.

Được nhiều nhà sản xuất thiết bị hỗ trợ, đa dạng về phần cứng.

Khả năng tùy biến cao.

Khuyết điểm:

Phân mảnh lớn do có nhiều phiên bản hệ điều hành và thiết bị khác nhau, dẫn đến khó khăn trong việc tối ưu hóa ứng dụng.

Bảo mật có thể không đồng đều giữa các thiết bị.

iOS:

Đặc điểm: Hệ điều hành độc quyền của Apple, chỉ có trên các thiết bị của Apple như iPhone, iPad.

Ưu điểm:

Trải nghiệm người dùng đồng nhất và mượt mà.

Bảo mật cao và có sự hỗ trợ lâu dài từ Apple.

Ít phân mảnh hơn, dễ tối ưu hóa ứng dụng.

Khuyết điểm:

Chỉ chạy trên thiết bị của Apple, hạn chế lựa chọn cho người dùng.

Chi phí phát triển và duy trì ứng dụng có thể cao hơn do yêu cầu phần cứng cụ thể.

### Câu 2

· Native (Android với Java/Kotlin và iOS với Swift/Objective-C):

· Đặc điểm: Ứng dụng được viết riêng cho từng nền tảng bằng ngôn ngữ tương ứng (Java/Kotlin cho Android, Swift/Objective-C cho iOS).

Ưu điểm: Hiệu suất tốt nhất, có quyền truy cập đầy đủ vào các API gốc của hệ điều hành.

Khuyết điểm: Phải phát triển riêng biệt cho từng nền tảng, tốn nhiều thời gian và chi phí.

· Flutter:

· Đặc điểm: Framework của Google cho phép phát triển ứng dụng đa nền tảng với ngôn ngữ Dart.

Ưu điểm: Hiệu suất gần với native, giao diện đẹp mắt và nhất quán nhờ các widget của Flutter.

Khuyết điểm: Cộng đồng còn nhỏ so với các nền tảng khác, các thư viện bên thứ ba chưa đa dạng.

· React Native:

· Đặc điểm: Framework của Facebook sử dụng JavaScript, có thể tạo ứng dụng đa nền tảng.

Ưu điểm: Cộng đồng lớn, dễ học với các lập trình viên JavaScript.

Khuyết điểm: Hiệu suất có thể kém hơn native trong một số trường hợp, yêu cầu cầu nối (bridge) để truy cập các API native.

· Xamarin:

· Đặc điểm: Framework của Microsoft, sử dụng ngôn ngữ C#.

Ưu điểm: Hỗ trợ đa nền tảng, có thể chia sẻ mã nguồn với các ứng dụng .NET khác.

Khuyết điểm: Hiệu suất có thể không bằng native, ứng dụng có thể nặng hơn do phải chạy qua một lớp trung gian.

### Câu 3

Lý do Flutter phổ biến:

Hiệu suất cao: Flutter biên dịch trực tiếp thành mã máy, giúp ứng dụng đạt hiệu suất gần như native.

Giao diện đẹp: Flutter có nhiều widget tùy chỉnh, giúp tạo giao diện nhất quán trên cả iOS và Android.

Hỗ trợ đa nền tảng: Bên cạnh iOS và Android, Flutter còn hỗ trợ phát triển cho web và desktop.

Được Google hỗ trợ: Điều này giúp Flutter có tiềm năng phát triển và cập nhật tốt trong tương lai.

So sánh với React Native và Xamarin:

React Native: Dùng JavaScript, phổ biến hơn với cộng đồng lập trình viên front-end. Tuy nhiên, React Native không biên dịch trực tiếp mà phải qua cầu nối, nên có thể không nhanh bằng Flutter.

Xamarin: Dùng C# và hỗ trợ tốt với các dự án của Microsoft. Xamarin dễ tích hợp vào hệ sinh thái .NET, nhưng thường kém phổ biến và ứng dụng thường nặng hơn so với Flutter.

#### Câu 4

· Java:

· Lý do: Java là ngôn ngữ truyền thống cho Android, có cộng đồng lớn, nhiều tài liệu và thư viện hỗ trợ.

· Đặc điểm: Dễ học, dễ bảo trì và có hỗ trợ tốt từ Google.

· Kotlin:

· Lý do: Google đã công nhận Kotlin là ngôn ngữ chính thức cho Android do cú pháp ngắn gọn, dễ bảo trì và hiệu suất cao hơn Java trong nhiều trường hợp.

· Đặc điểm: Tương thích hoàn toàn với Java, có các tính năng hiện đại giúp mã sạch và an toàn hơn.

· C++:

· Lý do: Được sử dụng trong các trường hợp cần hiệu suất cao, như ứng dụng game hoặc xử lý đồ họa nặng.

· Đặc điểm: Tốc độ xử lý nhanh, nhưng khó học và phức tạp hơn Java và Kotlin.

#### Câu 5

Các ngôn ngữ lập trình chính để phát triển ứng dụng trên iOS bao gồm:

##### Swift:

Đặc điểm: Được Apple phát triển và khuyến khích sử dụng cho các ứng dụng iOS hiện đại. Swift có cú pháp ngắn gọn, rõ ràng và an toàn hơn so với Objective-C.

Ưu điểm: Hiệu suất cao, dễ bảo trì, khả năng đọc tốt và giảm thiểu lỗi runtime.

Khuyết điểm: Swift là ngôn ngữ mới hơn, nên có thể gặp một số khó khăn khi làm việc với các framework cũ hoặc thư viện viết bằng Objective-C.

##### Objective-C:

Đặc điểm: Ngôn ngữ cũ của Apple dùng để phát triển ứng dụng trên iOS, có cú pháp phức tạp hơn Swift.

Ưu điểm: Đã tồn tại lâu đời và có một hệ sinh thái phong phú các thư viện và framework.

Khuyết điểm: Khó đọc và bảo trì hơn Swift, ít được khuyến khích sử dụng cho các ứng dụng mới.

#### Câu 6

Windows Phone đã phải đối mặt với nhiều thách thức, dẫn đến sự sụt giảm thị phần và cuối cùng là ngừng hoạt động. Một số thách thức chính bao gồm:

Thiếu ứng dụng và hệ sinh thái: Windows Phone không thu hút được nhiều nhà phát triển, dẫn đến kho ứng dụng thiếu phong phú so với Android và iOS. Điều này làm cho hệ điều hành này kém hấp dẫn với người dùng.

Cạnh tranh mạnh mẽ từ Android và iOS: Android và iOS đã có thị phần lớn và hệ sinh thái phong phú, làm cho Windows Phone khó cạnh tranh.

Chiến lược tiếp thị chưa hiệu quả: Microsoft không có chiến lược tiếp thị mạnh mẽ và nhất quán như Apple hoặc Google, làm giảm sức hút của Windows Phone trên thị trường.

Thiếu sự hỗ trợ từ các nhà sản xuất phần cứng: Rất ít nhà sản xuất chọn Windows Phone làm nền tảng chính cho thiết bị của họ. Điều này làm giảm sự đa dạng về thiết bị và khả năng tiếp cận người dùng.

#### Câu 7

Các ngôn ngữ và công cụ phổ biến để phát triển ứng dụng web trên thiết bị di động bao gồm:

HTML, CSS và JavaScript: Là nền tảng cơ bản để phát triển giao diện web. Các công nghệ này có thể được sử dụng cùng với các framework để tạo ra trải nghiệm tương tác trên di động.

##### Frameworks và Libraries:

React Native: Cho phép tạo ứng dụng đa nền tảng với JavaScript, có thể hoạt động như một ứng dụng native trên cả iOS và Android.

Flutter: Framework của Google cho phép phát triển ứng dụng đa nền tảng, tạo ra giao diện đẹp mắt và hiệu suất tốt.

Ionic: Sử dụng HTML, CSS, và JavaScript để xây dựng ứng dụng di động với giao diện native-like, phù hợp cho các ứng dụng hybrid.

Angular và Vue.js: Được sử dụng cho các ứng dụng web, có thể kết hợp với Ionic hoặc Cordova để tạo ra ứng dụng di động.

Công cụ hỗ trợ:

Cordova: Cho phép các ứng dụng web hoạt động như ứng dụng native bằng cách truy cập các tính năng phần cứng của thiết bị.

Expo: Một công cụ hỗ trợ phát triển và triển khai ứng dụng React Native nhanh chóng.

### Câu 8

Lập trình viên di động hiện nay là một trong những lĩnh vực có nhu cầu tuyển dụng cao nhờ sự phát triển mạnh mẽ của các thiết bị di động và ứng dụng di động. Những kỹ năng được yêu cầu nhiều nhất cho lập trình viên di động bao gồm:

1. **Thành thạo ngôn ngữ lập trình di động chính** như Swift (iOS), Kotlin và Java (Android), cũng như các công nghệ đa nền tảng như Flutter và React Native.
2. **Hiểu biết về UI/UX** để thiết kế giao diện người dùng hấp dẫn và thân thiện.
3. **Kiến thức về bảo mật** nhằm đảm bảo an toàn dữ liệu người dùng trên ứng dụng di động.
4. **Kỹ năng tối ưu hóa hiệu suất** giúp ứng dụng chạy mượt mà và tối ưu.
5. **Khả năng kiểm thử và tự động hóa** bằng các công cụ kiểm thử phổ biến như JUnit, Espresso (Android), XCTest (iOS).
6. **Kỹ năng làm việc với RESTful API và GraphQL** để kết nối ứng dụng với các dịch vụ server.
7. **Hiểu biết về phát triển đa nền tảng** để có thể tạo ra ứng dụng hoạt động trên nhiều hệ điều hành.
8. **Kỹ năng giao tiếp và làm việc nhóm**, hỗ trợ phối hợp với các bộ phận khác trong quy trình phát triển phần mềm.
9. **Tinh thần học hỏi và cập nhật công nghệ mới** để bắt kịp các xu hướng phát triển trong ngành.
10. **Khả năng phân tích và giải quyết vấn đề** nhằm đáp ứng các yêu cầu kỹ thuật phức tạp.
11. **Kỹ năng về DevOps và CI/CD** để tự động hóa quy trình phát triển, kiểm thử và triển khai ứng dụng.
12. **Kiến thức về cơ sở dữ liệu** như SQLite, Firebase hoặc Core Data để quản lý dữ liệu hiệu quả trên ứng dụng di động.