

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 công nghệ và công cụ phổ biến trong phát triển ứng dụng web di động hiện nay bao gồm những nền tảng cơ bản như HTML, CSS và JavaScript, giúp xây dựng giao diện và chức năng cho các ứng dụng web. Kết hợp với các framework như React Native, các lập trình viên có thể phát triển ứng dụng đa nền tảng hoạt động như ứng dụng native trên cả iOS và Android. Flutter, framework của Google, cũng cho phép tạo ra các ứng dụng đẹp mắt và hiệu suất cao, trong khi Ionic, sử dụng HTML, CSS và JavaScript, giúp phát triển các ứng dụng hybrid với giao diện gần giống ứng dụng native. Ngoài ra, các framework web như Angular và Vue.js có thể được tích hợp với các công cụ như Ionic hoặc Cordova để tạo ứng dụng di động. Cordova cho phép ứng dụng web truy cập các tính năng phần

cứng của thiết bị, mang lại trải nghiệm tương tự ứng dụng native. Expo là một công cụ hỗ trợ phát triển ứng dụng React Native nhanh chóng và hiệu quả. Các công cụ và công nghệ này giúp lập trình viên phát triển ứng dụng di động đa nền tảng một cách dễ dàng, đồng thời nâng cao hiệu suất và trải nghiệm người dùng.

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. **Kỹ năng lập trình cơ bản:** Lập trình viên di động cần vững các ngôn ngữ lập trình như Java, Kotlin (cho Android), Swift, Objective-C (cho iOS), hoặc Flutter, React Native (cho ứng dụng đa nền tảng).
2. **Kiến thức về các nền tảng di động:** Lập trình viên cần hiểu rõ về hệ điều hành di động, chẳng hạn như Android và iOS, cách hoạt động, giao diện người dùng, và các công cụ phát triển như Android Studio, Xcode.
3. **Phát triển giao diện người dùng (UI/UX):** Kiến thức về thiết kế giao diện người dùng và trải nghiệm người dùng, làm sao để thiết kế ứng dụng dễ sử dụng, mượt mà và hấp dẫn.
4. **Lập trình hướng đối tượng:** Kỹ năng lập trình hướng đối tượng rất quan trọng trong phát triển phần mềm di động, giúp tổ chức mã nguồn hiệu quả và dễ bảo trì.
5. **Quản lý dữ liệu và cơ sở dữ liệu:** Kỹ năng làm việc với cơ sở dữ liệu di động như SQLite, Room (Android), Core Data (iOS) hoặc các dịch vụ đám mây như Firebase.
6. **Kiến thức về mạng và API:** Lập trình viên di động cần hiểu về việc kết nối và giao tiếp giữa các ứng dụng và máy chủ thông qua RESTful APIs, JSON, HTTP, v.v.
7. **Tối ưu hóa hiệu suất:** Cải thiện hiệu suất ứng dụng di động như giảm thiểu mức tiêu thụ pin, cải thiện tốc độ tải trang và xử lý dữ liệu nhanh chóng.
8. **Kiến thức về kiểm thử và gỡ lỗi:** Kỹ năng kiểm thử ứng dụng và gỡ lỗi là rất quan trọng để đảm bảo chất lượng phần mềm, tránh lỗi phát sinh trong quá trình sử dụng.
9. **Kinh nghiệm làm việc với các công cụ kiểm tra và CI/CD:** Các công cụ như Jenkins, Fastlane, hoặc GitLab CI giúp tự động hóa quá trình kiểm tra và phát hành ứng dụng.

- 10.**Hiểu biết về bảo mật ứng dụng di động: Kiến thức về bảo mật, mã hóa và các vấn đề bảo mật liên quan đến dữ liệu người dùng trong ứng dụng di động rất quan trọng.
- 11.**Phát triển ứng dụng đa nền tảng: Ngoài việc phát triển riêng biệt cho Android và iOS, việc làm quen với các framework như React Native, Flutter giúp lập trình viên tạo ứng dụng chạy trên cả hai nền tảng mà không phải viết mã lại từ đầu.