

Câu 1:

Các nền tảng phổ biến:

- Android: Phát triển bởi Google, là hệ điều hành mã nguồn mở và linh hoạt nhất hiện nay, được phân phối tự do.
 - + Ưu điểm: Đa dạng thiết bị hỗ trợ, dễ dàng tùy biến.
 - + Nhược điểm: Cần tối ưu cho nhiều thiết bị, gây khó khăn cho lập trình viên.
- iOS: Hệ điều hành của Apple, chỉ sử dụng cho các thiết bị iPhone, iPad.
 - + Ưu điểm: Trải nghiệm mượt mà, hệ sinh thái nhất quán.
 - + Nhược điểm: Khó khăn trong tùy biến, chỉ sử dụng trên thiết bị Apple.
- BlackBerry: Phát triển bởi hãng Research in Motion (RIM). Cung cấp cho sản phẩm khả năng đa nhiệm, sử dụng chính trên những thiết bị có phương pháp nhập đặc biệt và thường là màn hình cảm ứng hoặc là Trackball.
 - + Ưu điểm: Cung cấp giải pháp tiện lợi và nhanh chóng cho người dùng cuối, kết hợp được các tệp tin của người dùng và nhà cung cấp, quy mô lớn, quản lý được sự thay đổi phức tạp của thị trường.
 - + Nhược điểm: Không thể quản trị trải nghiệm người dùng đầu cuối một cách hiệu quả, chi tiết và chuyên nghiệp như các mô hình nền tảng truyền thống.
- KaiOS: Tập trung vào các thiết bị cơ bản và giá rẻ.
 - + Ưu điểm: Tiêu tốn ít tài nguyên, phù hợp cho thiết bị thấp.
 - + Nhược điểm: Hạn chế về tính năng và không có hệ sinh thái phong phú.

Câu 2:

Các nền tảng phát triển ứng dụng di động phổ biến hiện nay

- Native Development (Android/iOS): Sử dụng Java/Kotlin cho Android và Swift/Objective-C cho iOS.
 - + Ưu điểm: Hiệu suất cao, tận dụng tối đa phần cứng.
 - + Nhược điểm: Cần phát triển riêng cho từng nền tảng.
- Flutter: Sử dụng Dart, hỗ trợ đa nền tảng, cung cấp UI gần như native.
 - + Ưu điểm: Cùng một codebase cho cả Android và iOS, hiệu suất tốt.
 - + Nhược điểm: Cộng đồng chưa lớn bằng các framework khác.

– React Native: Sử dụng JavaScript, một framework đa nền tảng được Facebook phát triển.

+ Ưu điểm: Cộng đồng lớn, hỗ trợ từ Facebook, nhiều thư viện.

+ Nhược điểm: Cần tối ưu hiệu suất để đạt mức như native.

– Xamarin: Sử dụng C#, hỗ trợ bởi Microsoft.

+ Ưu điểm: Tích hợp tốt với các dịch vụ của Microsoft.

+ Nhược điểm: Đòi hỏi cấu hình máy cao, hiệu suất chậm hơn.

Câu 3:

Những điều giúp Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng

- Hiệu suất gần với phát triển ứng dụng gốc đảm bảo hiệu suất nhanh và hiệu quả.
- Kết xuất nhanh, nhất quán và có thể tùy chỉnh.
- Phát triển ứng dụng đa nền tảng cho phép các nhà lập trình sử dụng một ngôn ngữ lập trình và một nền mã để xây dựng một ứng dụng cho nhiều nền tảng.

Sự khác biệt nền tảng Flutter đối với React Native, Xarmian

- Thời gian ứng dụng phát triển:
 - + Flutter: Có Framework mang lại hiệu suất ổn định, sử dụng mã gốc và các widget để đạt hiệu suất tốt.
 - + React Native: Framework hỗ trợ hiệu suất cực nhanh, cho phép viết mã bằng ngôn ngữ Native như Java hoặc Swift.
 - + Xarmian: Hiệu suất tốt và liên tục cải tiến, phù hợp cho tiêu chuẩn phát triển ứng dụng Native. Sự trợ giúp nền tảng Xarmian, giúp xây dựng số tính năng như form, thư viện giao diện người dùng UI đặc biệt.
- Hiệu năng phát triển:
 - + Tính năng “Hot Reload” trên nền tảng Flutter giúp tiết kiệm thời gian cho các developer. Có nhiều widget và API trong Flutter giúp build phát triển ứng dụng một cách nhanh chóng.
 - + Với nền tảng React Native, có thể linh hoạt làm việc trên bất kỳ trình soạn thảo văn bản / IDE nào mà nhà phát triển thích. Cung cấp một tính năng được xây dựng trước có tên “Hot Reload”, giúp hoàn thành dự án nhanh.
 - + Còn nền tảng Xarmian thì cho phép các developer triển khai cải tiến cho XAML.

Câu 4:

Các ngôn ngữ chính dùng phát triển ứng dụng Android

- Java: dễ sử dụng, bảo trì và có cả một cộng đồng lớn liên tục cải tiến ngôn ngữ.
- Kotlin: có mức độ tương đồng như Java, cú pháp ngắn gọn, hiệu suất ổn định, tính tương thích cao với Java.
- Angular JS: có khả năng tạo và phát triển ứng dụng đa nền tảng trong đó có cả Android.

Câu 5:

Các ngôn ngữ lập trình chính sử dụng phát triển ứng dụng trên iOS

- Swift: Cú pháp đơn giản, ít mã lỗi, tăng tốc độ phát triển dành cho các lập trình viên mới chuyên phát triển ứng dụng.
- Objective-C: Hiệu suất ổn định, dễ connect với API , nguồn tài nguyên dồi dào cho các nhà phát triển.
- JavaScript + React Native: Ngôn ngữ linh hoạt cho nhiều loại dự án, trải nghiệm người dùng phong phú.

Câu 6:

Những thách thức và nguyên lý sụp đổ của Windows Phone

- Ứng dụng: Kho ứng dụng ít, thiếu nhiều ứng dụng phổ biến.
- Hạn chế đối tác: Không tạo được điểm thu hút với các nhà phát triển.
- Không nắm bắt thị trường: Microsoft không duy trì được sức hút cũng như các kiểu mẫu mã mới đối với người dùng.

Câu 7:

- Ngôn ngữ phát triển: HTML, CSS, Java, JavaScript giúp hỗ trợ xây dựng giao diện, tính năng tương tác của ứng dụng đối với người dùng.
- Framework: Progressive Web Apps (PWAs) cung cấp môi trường phát triển ứng dụng.
- Công cụ xây dựng ứng dụng: Firebase và các công cụ tối ưu hóa giao diện.

Câu 8:

Các lập trình viên ứng dụng buộc phải thông thạo ngôn ngữ và các nền tảng mà họ xây dựng và phát triển ứng dụng như Swift, Kotlin, Dart, Objective-C,... và các nền tảng như Flutter, React Native hoặc Xamarin.

Cùng với đó các kỹ năng cần đáp ứng ở một lập trình viên phát triển ứng dụng bao gồm: tư duy logic giải quyết vấn đề, kỹ năng làm việc nhóm và giao tiếp, khả năng tự học và một nền tảng ngoại ngữ nhất định.