

Câu 1: Các nền tảng cho thiết bị di động thông minh hiện nay

1. Android:

- Ưu điểm: Mã nguồn mở, hỗ trợ nhiều thiết bị, kho ứng dụng phong phú.
- Nhược điểm: Phân mảnh, bảo mật thấp hơn.

2. iOS:

- Ưu điểm: Bảo mật cao, hiệu suất ổn định, hệ sinh thái khép kín.
- Nhược điểm: Chỉ dùng trên thiết bị Apple, chi phí cao.

3. HarmonyOS:

- Ưu điểm: Tích hợp tốt trong hệ sinh thái Huawei.
- Nhược điểm: Ít ứng dụng, hệ sinh thái còn mới.

4. KaiOS:

- Ưu điểm: Chi phí thấp, tối ưu cho điện thoại phổ thông.
- Nhược điểm: Hạn chế về ứng dụng và hiệu năng.

Câu 2:

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

1. Native Development (Android và iOS)

2. React Native

3. Flutter

4. Xamarin

5. Ionic

So sánh sự khác biệt chính:

Hiệu suất: Native > Flutter > React Native > Xamarin > Ionic.

Dễ phát triển và tái sử dụng mã: Ionic > React Native > Xamarin > Flutter > Native.

Tích hợp hệ thống: Native > Flutter > React Native > Xamarin > Ionic.

Câu 3:

Flutter phổ biến nhờ:

1. Hiệu năng cao: Flutter dùng Dart và Skia để render UI trực tiếp, cho hiệu suất mượt mà hơn so với React Native (cần cầu nối JavaScript) và Xamarin (lớp trung gian C#).

2. UI nhất quán: Widget tùy chỉnh, UI đồng nhất trên mọi nền tảng, trong khi React Native và Xamarin sử dụng thành phần native, đòi hỏi điều chỉnh riêng cho iOS/Android.

3. Hot Reload: Thay đổi mã thấy ngay lập tức, nhanh hơn React Native và Xamarin.

4. Đa nền tảng: Hỗ trợ iOS, Android, web, desktop (Windows, macOS, Linux), linh hoạt hơn React Native và Xamarin.

5. Thư viện phong phú: Hệ sinh thái rộng và cập nhật nhanh nhờ Google, hơn hẳn Xamarin và tương đương React Native.

Flutter mạnh về hiệu năng, giao diện tùy chỉnh và hỗ trợ nhiều nền tảng, vượt trội hơn React Native và Xamarin trong nhiều trường hợp.

Câu 4:

1. Java

Lý do chọn: Java cung cấp khả năng tương thích cao, hiệu năng tốt, và hỗ trợ mạnh mẽ từ cộng đồng lập trình viên.

2. Kotlin

Lý do chọn: Kotlin giúp giảm bớt boilerplate code (mã lặp lại), hỗ trợ tính năng an toàn với null (null safety), và dễ dàng tích hợp với các thư viện Java.

3. C++

Lý do chọn: C++ cho phép tối ưu hóa mã nguồn, mang lại hiệu suất cao và được hỗ trợ qua NDK (Native Development Kit).

4. Dart (Flutter)

Lý do chọn: Dart dễ học, cho phép phát triển ứng dụng đa nền tảng nhanh chóng và với hiệu suất gần như native.

Câu 5:

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

Objective-C

Swift

Objective-C++

Dart (Flutter)

Câu 6:

Windows Phone đối mặt với nhiều thách thức và nguyên nhân dẫn đến sự sụt giảm thị phần bao gồm:

1. Thiếu ứng dụng:

- Thách thức: Hệ sinh thái ứng dụng của Windows Phone rất hạn chế so với iOS và Android. Nhiều ứng dụng phổ biến không có sẵn hoặc được phát triển chậm hơn trên nền tảng này.

- Nguyên nhân: Các nhà phát triển chủ yếu tập trung vào iOS và Android, nơi có thị trường người dùng lớn hơn.

2. Thị phần nhỏ:

- Thách thức: Với thị phần hạn chế, Windows Phone không thể thu hút đủ người dùng và các nhà phát triển ứng dụng.

- Nguyên nhân: Người tiêu dùng thường ngại chuyển sang hệ điều hành mới nếu không có đủ ứng dụng và sự hỗ trợ từ các nhà phát triển.

3. Cạnh tranh mạnh mẽ:

- Thách thức: Android và iOS đã thống trị thị trường di động và có những lợi thế lớn về ứng dụng, tính năng, và sự đổi mới.

- Nguyên nhân: Windows Phone không thể cạnh tranh hiệu quả với Android và iOS, cả về phần mềm lẫn phần cứng.

4. Thiếu đổi mới và phát triển chậm:

- Thách thức: Mặc dù Windows Phone có giao diện Metro độc đáo, nhưng không có sự đổi mới mạnh mẽ trong suốt thời gian dài.

- Nguyên nhân: Microsoft không thể đáp ứng kịp sự thay đổi nhanh chóng của ngành công nghiệp di động, dẫn đến sự tụt lại so với các đối thủ.

5. Chính sách tiếp thị yếu:

- Thách thức: Microsoft không thể tạo ra chiến dịch tiếp thị đủ mạnh để thu hút người dùng và nhà phát triển.

- Nguyên nhân: Thiếu sự tập trung và chiến lược rõ ràng trong việc xây dựng thương hiệu Windows Phone.

Câu 7:

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

1. HTML, CSS, JavaScript:

- HTML: Cấu trúc của trang web.

- CSS: Thiết kế giao diện, tối ưu cho các màn hình nhỏ.

- JavaScript: Xử lý logic và tương tác động trên trang web.

2. Frameworks và Libraries:

- React (React Native): Phát triển ứng dụng web và di động với JavaScript.
- Vue.js: Framework JavaScript nhẹ, dễ học và sử dụng.
- Angular: Framework mạnh mẽ cho các ứng dụng web phức tạp.
- Bootstrap: Framework CSS giúp tạo giao diện responsive dễ dàng.

3. Progressive Web Apps (PWA):

- ****Service Workers, Web App Manifest****: Các công nghệ giúp ứng dụng web hoạt động như ứng dụng native, cung cấp trải nghiệm offline và cài đặt trên thiết bị.

4. Ionic:

- Công cụ phát triển ứng dụng hybrid cho cả web và di động, sử dụng HTML, CSS, và JavaScript.

5. Flutter Web:

- Sử dụng Dart để phát triển ứng dụng web responsive, hỗ trợ cả di động và web từ cùng một mã nguồn.

6. Xamarin (Web):

- Sử dụng C# và .NET để phát triển ứng dụng đa nền tảng, bao gồm cả web và di động.

Câu 8:

1.Kỹ năng về ngôn ngữ lập trình:

Swift (iOS), Kotlin (Android): Đây là hai ngôn ngữ chính được sử dụng để phát triển ứng dụng native trên iOS và Android.

Java: Vẫn được sử dụng phổ biến cho các ứng dụng Android.

Flutter (Dart) và React Native (JavaScript): Kỹ năng phát triển ứng dụng đa nền tảng ngày càng được ưa chuộng, giúp lập trình viên xây dựng ứng dụng cho cả iOS và Android từ cùng một mã nguồn.

2.Kỹ năng về UX/UI:

Thiết kế giao diện thân thiện với người dùng trên thiết bị di động và tối ưu hóa trải nghiệm người dùng là một yêu cầu quan trọng.

3.Kiến thức về kiến trúc ứng dụng:

Hiểu biết về các mẫu kiến trúc như MVVM, MVP, và khả năng tổ chức mã tốt giúp ứng dụng hoạt động ổn định và dễ bảo trì.

4.Tích hợp API và dịch vụ bên ngoài:

Kỹ năng tích hợp với các API RESTful, GraphQL, và các dịch vụ như Firebase, Google Maps, Payment Gateway rất cần thiết.

5. Bảo mật ứng dụng:

Hiểu biết về bảo mật trên thiết bị di động, như mã hóa dữ liệu, xác thực người dùng, bảo vệ thông tin cá nhân, và ngăn ngừa tấn công.

6. Tư duy phát triển đa nền tảng:

Với nhu cầu phát triển nhanh, lập trình viên có kiến thức về PWA, Flutter, và React Native để xây dựng ứng dụng đa nền tảng được đánh giá cao.

7. Kỹ năng DevOps:

Kiến thức về CI/CD, quản lý và triển khai ứng dụng di động lên các cửa hàng ứng dụng như App Store và Google Play.