

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

1. Android

- **Đặc điểm:** Mã nguồn mở, dựa trên Linux, được Google phát triển.
- **Ưu điểm:**
 - Cộng đồng phát triển lớn.
 - Tùy chỉnh cao, hỗ trợ đa dạng thiết bị.
 - Kho ứng dụng phong phú (Google Play).
- **Khuyết điểm:**
 - Phân mảnh hệ điều hành (nhiều phiên bản chạy song song).
 - Tính bảo mật thấp hơn iOS.

2. iOS

- **Đặc điểm:** Được phát triển bởi Apple, chạy trên các thiết bị như iPhone, iPad.
- **Ưu điểm:**
 - Hiệu năng ổn định, tối ưu hóa tốt cho phần cứng.
 - Bảo mật cao.
 - Giao diện và trải nghiệm người dùng mượt mà.
- **Khuyết điểm:**
 - Chi phí thiết bị cao.
 - Hệ sinh thái đóng, hạn chế tùy chỉnh.

3. HarmonyOS

- **Đặc điểm:** Nền tảng của Huawei, có khả năng tích hợp trên nhiều thiết bị (smartphone, TV, IoT).
- **Ưu điểm:**
 - Tối ưu hóa kết nối đa thiết bị.
 - Hỗ trợ ứng dụng Android thông qua dịch vụ HMS.
- **Khuyết điểm:**
 - Ứng dụng chưa phong phú.
 - Cộng đồng phát triển nhỏ.

4. KaiOS

- **Đặc điểm:** Dành cho các điện thoại phổ thông (feature phones).
- **Ưu điểm:**
 - Tiêu thụ tài nguyên thấp.
 - Giá thành thiết bị rẻ.

- **Khuyết điểm:**
 - Hạn chế về chức năng.
 - Ít ứng dụng so với Android và iOS.

Câu 2: Các nền tảng phát triển ứng dụng di động phổ biến hiện nay

1. Native

- **Đặc điểm:** Sử dụng ngôn ngữ gốc (Java/Kotlin cho Android, Swift/Objective-C cho iOS).
- **Ưu điểm:**
 - Hiệu suất cao, trải nghiệm mượt mà.
 - Tận dụng tối đa phần cứng.
- **Khuyết điểm:**
 - Phát triển tốn thời gian và chi phí (mỗi nền tảng phải có mã riêng).

2. Cross-platform

- **Ví dụ:** Flutter, React Native, Xamarin.
- **Ưu điểm:**
 - Viết một lần, chạy trên nhiều nền tảng.
 - Giảm thời gian và chi phí phát triển.
- **Khuyết điểm:**
 - Hiệu suất thấp hơn Native.
 - Hạn chế sử dụng tính năng riêng của từng nền tảng.

3. Progressive Web Apps (PWA)

- **Đặc điểm:** Ứng dụng web với trải nghiệm giống ứng dụng di động.
- **Ưu điểm:**
 - Dễ tiếp cận, không cần tải từ store.
 - Dễ bảo trì và cập nhật.
- **Khuyết điểm:**
 - Chạy phụ thuộc vào trình duyệt.
 - Hạn chế về tính năng phần cứng.

Câu 3: Flutter và so sánh với React Native, Xamarin

1. Flutter

- **Ưu điểm:**
 - Hiệu suất gần Native nhờ sử dụng Dart và trình biên dịch AOT.
 - UI linh hoạt với hệ thống widget tùy chỉnh.
 - Một mã nguồn duy nhất cho iOS, Android, Web, và Desktop.
- **Khuyết điểm:**
 - Kích thước ứng dụng lớn.
 - Cộng đồng chưa lớn bằng React Native.

2. React Native

- **Ưu điểm:**
 - Phát triển nhanh, dựa trên JavaScript.
 - Cộng đồng lớn, nhiều thư viện hỗ trợ.
- **Khuyết điểm:**
 - Hiệu suất thấp hơn Flutter.
 - Tích hợp phần cứng phức tạp.

3. Xamarin

- **Ưu điểm:**
 - Hỗ trợ tích hợp sâu với hệ điều hành.
 - Dựa trên C#, dễ dàng cho lập trình viên .NET.
- **Khuyết điểm:**
 - Chi phí cao (nếu dùng Visual Studio Enterprise).
 - Tài liệu và cộng đồng ít hơn React Native.

Câu 4: Các ngôn ngữ lập trình cho Android

- **Java:** Ngôn ngữ truyền thống, dễ học và hỗ trợ tốt.
- **Kotlin:** Ngôn ngữ chính thức hiện nay, cú pháp hiện đại, ít lỗi hơn Java.

Câu 5: Các ngôn ngữ lập trình cho iOS

- **Swift:** Nhanh, hiện đại, được Apple tối ưu hóa.
- **Objective-C:** Truyền thống nhưng dần bị Swift thay thế.

Câu 6: Thách thức của Windows Phone

- **Hệ sinh thái ứng dụng hạn chế:** Ít ứng dụng hơn so với Android, iOS.
- **Tích hợp chậm:** Chậm đổi mới và thiếu hỗ trợ từ các nhà phát triển.
- **Chiến lược kinh doanh yếu:** Không cạnh tranh tốt với Android và iOS.

Câu 7: Ngôn ngữ và công cụ phát triển ứng dụng web trên thiết bị di động

- **Ngôn ngữ:** JavaScript, HTML, CSS.
- **Công cụ:** React, Angular, Vue.js, Ionic.

Câu 8: Nhu cầu và kỹ năng lập trình viên di động

- **Nhu cầu:** Cao ở các công ty công nghệ, startup.
- **Kỹ năng được yêu cầu:**
 - Thành thạo Native hoặc Cross-platform.
 - Kỹ năng UI/UX.
 - Kiến thức về API và cơ sở dữ liệu.