

Họ tên: Khổng Minh Phúc

MSSV: 2180607887

Câu 1: Hiện nay có nhiều hệ điều hành cho di động: IOS, Android, HarmonyOS

...

Hệ điều hành	Đặc điểm	Ưu điểm	Khuyết điểm
IOS	<ul style="list-style-type: none">-Độc quyền.-AppStore là nền tảng duy nhất để phân phối và tải xuống các ứng dụng cho IOS.-Giao diện người dùng thân thiện.-Bảo mật mạnh mẽ: mã hóa dữ liệu, sinh trắc học.-Môi trường đóng	<ul style="list-style-type: none">-Tính ổn định cao-Bảo mật và quyền riêng tư mạnh mẽ-Hệ sinh thái Apple đồng nhất và tương thích cao giữa các thiết bị.-Bảo mật cao	<ul style="list-style-type: none">-Độc quyền và giá cả cao: Giới hạn đối tượng người dùng.-Hạn chế tùy biến: Hạn chế trong việc tùy biến hệ điều hành và cài đặt ứng dụng từ bên ngoài AppStore.-Quy trình phê duyệt ứng dụng nghiêm ngặt
Android	<ul style="list-style-type: none">-Mã nguồn mở.-Chiếm khoảng 70% thị phần toàn cầu.-Đa dạng về thiết bị và sản xuất.	<ul style="list-style-type: none">-Tùy biến cao-Dễ phân phối ứng dụng-Đa dạng về thiết bị và giá cả-Cộng đồng lớn	<ul style="list-style-type: none">-Phân mảnh về phiên bản-Vấn đề bảo mật-Khó tối ưu cho nhiều thiết bị
HarmonyOS	<ul style="list-style-type: none">-Hệ điều hành phân tán-Có thể chạy trên nhiều loại thiết bị khác nhau (Tablet, Điện thoại, Tivi).-Hỗ trợ đa nền tảng với ARK Compiler	<ul style="list-style-type: none">-Hiệu suất cao độ trễ thấp-Tính bảo mật cao nhờ microkernel-Khả năng kết nối chia sẻ liên mạch nhiều thiết bị-Tiêu thụ ít tài nguyên hệ thống	<ul style="list-style-type: none">-Hệ sinh thái ứng dụng còn hạn chế-Chủ yếu phổ biến tại thị trường Trung Quốc-Thiếu các dịch vụ Google-Đang trong giai đoạn phát triển ban đầu

3. Câu 2:Các ứng dụng phát triển di động phổ biến: Native Development, Cross-platform Development, Progressive Web Apps (PWA)

So sánh sự khác biệt

Native Development	Cross-platform Development	Progressive Web Apps
<ul style="list-style-type: none">-Công cụ riêng biệt (Android Studio, Xcode)-Không cần lớp trung gian-Tối ưu nhất (100%)-Truy cập toàn bộ APIs-Phát triển riêng biệt-Chi phí cao nhất (200%)-Look & feel chuẩn platform	<ul style="list-style-type: none">-Một công cụ cho nhiều nền tảng-Có lớp trung gian xử lý-Đạt 80-90% hiệu suất native-Truy cập hầu hết APIs quan trọng-Phát triển song song-Chi phí trung bình (150%)-Giao diện gần giống native	<ul style="list-style-type: none">-Công cụ phát triển web thông thường-Sử dụng web APIs-Đạt 60-70% hiệu suất native-Chỉ dùng được Web APIs-Phát triển một lần-Chi phí thấp nhất (100%)-Giao diện kiểu web

Câu 3: Flutter nổi bật nhờ hiệu suất cao, hot reload hiệu quả, và hệ thống widget linh hoạt. So với React Native và Xamarin, Flutter cung cấp trải nghiệm phát triển tốt hơn và hiệu suất cao hơn, mặc dù có thể còn thiếu một số tính năng và thư viện so với các framework trưởng thành hơn.

1. Điểm mạnh của Flutter:

Hiệu suất cao:

- Biên dịch trực tiếp ra mã native
- Tốc độ render nhanh nhờ Skia Engine
- Không cần bridge như React Native
- Khả năng chạy 60fps mượt mà

Hot Reload:

- Cập nhật code ngay lập tức
- Không mất trạng thái ứng dụng
- Tăng tốc độ phát triển
- Dễ dàng thử nghiệm UI

Widget độc đáo:

- Tất cả đều là widget
- Widget có thể tùy chỉnh hoàn toàn
- Tái sử dụng widget dễ dàng
- Giao diện nhất quán cross-platform

Dart - Ngôn ngữ lập trình:

- Dễ học với developer từ Java/JavaScript
- Hỗ trợ AOT và JIT compilation
- Hiệu suất cao
- Type-safe

Hỗ trợ từ Google:

- Tài liệu phong phú
- Cập nhật thường xuyên
- Cộng đồng lớn mạnh
- Nhiều package chất lượng

So sánh với React Native:

Flutter:

- Hiệu suất tốt hơn nhờ không dùng bridge
- UI nhất quán hơn giữa các nền tảng
- Công cụ dev mạnh mẽ hơn
- Hot reload ổn định hơn

React Native:

- Cộng đồng lớn hơn
- Nhiều thư viện hơn
- Dễ học hơn với web developer
- Flexible hơn trong styling

So sánh với Xamarin:

Flutter:

- Miễn phí hoàn toàn
- Hot reload nhanh hơn
- Kích thước ứng dụng nhỏ hơn
- UI đẹp và dễ tùy biến hơn

Xamarin:

- Tích hợp tốt với hệ sinh thái Microsoft
- Hỗ trợ C# và .NET
- UI native thực sự
- Trưởng thành hơn

- So sánh chi tiết ba nền tảng:

Ngôn ngữ:

- Flutter: Dart
- React Native: JavaScript/TypeScript
- Xamarin: C#

Hiệu suất:

- Flutter: Cao nhất
- React Native: Trung bình
- Xamarin: Khá cao

Học tập & Phát triển:

- Flutter: Dễ học, tài liệu tốt
- React Native: Dễ với web dev
- Xamarin: Phức tạp hơn

UI Components:

- Flutter: Custom widgets
- React Native: Native components
- Xamarin: Native UI

Kích thước ứng dụng:

- Flutter: Trung bình
- React Native: Nhỏ
- Xamarin: Lớn

Cộng đồng & Ecosystem:

- Flutter: Đang phát triển nhanh
- React Native: Lớn nhất
- Xamarin: Nhỏ hơn

Câu 4:

Ngôn ngữ lập trình	Lý do được chọn
Kotlin	<ul style="list-style-type: none"> -Null safety tích hợp sẵn -Cú pháp ngắn gọn, hiện đại -Giảm thiểu lỗi runtime -Tương thích ngược với Java -Hỗ trợ lập trình đa luồng tốt (Coroutines) -Hiệu suất tương đương Java -Tích hợp tốt với Android Studio
Python(Qpython/Kivy)	<ul style="list-style-type: none"> -Phát triển nhanh -Dễ học và sử dụng -Tốt cho prototype -Machine learning -Data processing
Java	<ul style="list-style-type: none"> -Độ ổn định cao -Cộng đồng lớn -Nhiều thư viện sẵn có -Dễ học, phổ biến -Hiệu suất tốt -Cross-platform
C/C++	<ul style="list-style-type: none"> -Tối ưu hiệu suất -Xử lý đồ họa phức tạp -Phát triển game -Tái sử dụng code C/C++ có sẵn

Câu 5: Các ngôn ngữ lập trình chính được sử dụng để phát triển phần mềm IOS: Swift, Objective-C, C/C++, React Native/JavaScript

Câu 6: Những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó

- Thách thức về Ecosystem:

- Thiếu ứng dụng

- Chất lượng ứng dụng:

- Thách thức về Thời điểm:

- Tham gia thị trường muộn

- Chiến lược không phù hợp:

- Thách thức về Kỹ thuật:

- Giới hạn phần cứng

- Vấn đề tương thích

Chiến lược kinh doanh:

- Thay đổi chiến lược liên tục
- Không nhất quán trong phát triển
- Marketing kém hiệu quả
- Đầu tư không đủ

B. Cạnh tranh:

- Android chiếm thị trường giá rẻ
- iOS độc quyền phân khúc cao cấp
- Không tìm được thị trường ngách
- Khó cạnh tranh về giá

C. Developer Relations:

- Không thu hút được developer
- Tools phát triển phức tạp
- Chi phí phát triển cao
- ROI thấp cho developer

5. Bài học kinh nghiệm:

A. Tầm quan trọng của ecosystem:

- Cần xây dựng hệ sinh thái trước
- Phải thu hút developer
- Đảm bảo có đủ ứng dụng phổ biến
- Chất lượng ứng dụng quan trọng

B. Timing và Execution:

- Thời điểm ra mắt quan trọng
- Cần chiến lược rõ ràng
- Thực thi phải nhất quán
- Phản ứng nhanh với thị trường

C. User Experience:

- UX phải vượt trội
- Tính năng độc đáo
- Giá trị cộng thêm rõ ràng
- Dễ dàng chuyển đổi

6. Tác động của thất bại:

A. Đến Microsoft:

- Mất thị phần mobile
- Ảnh hưởng đến chiến lược mobile-first
- Tồn kém về tài chính
- Mất vị thế trong mobile OS

B. Đến thị trường:

- Giảm cạnh tranh
- Android/iOS thống trị
- Ít sự lựa chọn cho user
- Tập trung quyền lực

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

1. Ngôn ngữ lập trình cơ bản:

- HTML5: Tạo cấu trúc và nội dung
- CSS3: Định dạng giao diện, responsive design
- JavaScript: Xử lý logic và tương tác

2. Các framework phát triển:

- React Native: Framework phổ biến của Facebook
- Flutter: Framework đa nền tảng của Google
- Ionic: Framework hybrid dựa trên Angular
- Vue Native: Phiên bản mobile của Vue.js

3. Công cụ và thư viện hữu ích:

- Bootstrap: Framework CSS responsive
- jQuery Mobile: Thư viện tối ưu cho mobile
- PWA (Progressive Web Apps): Cho phép web hoạt động như app native
- WebView: Hiển thị nội dung web trong app

4. Yếu tố cần chú ý:

- Responsive design cho nhiều kích thước màn hình
- Tối ưu hiệu năng và tốc độ tải
- Trải nghiệm người dùng trên mobile
- Khả năng hoạt động offline

Câu 8:

Hiện nay, nhu cầu nhân lực trong lĩnh vực lập trình ứng dụng di động vẫn tiếp tục tăng cao, được thúc đẩy bởi sự phát triển của smartphone và các nền tảng di động. Các công ty công nghệ, thương mại điện tử, y tế số, và giáo dục trực tuyến đều đang tìm kiếm các lập trình viên có chuyên môn sâu để phát triển các ứng dụng đáp ứng nhu cầu ngày càng cao của người tiêu dùng và doanh nghiệp.

Các kỹ năng quan trọng và yêu cầu

1. **Nền tảng công nghệ:** Lập trình viên di động cần nắm vững ít nhất một trong hai hệ điều hành chính là Android hoặc iOS. Đối với Android, cần thành thạo Java hoặc Kotlin; còn iOS đòi hỏi kỹ năng với Swift hoặc Objective-C.
2. **Công nghệ mới:** Ngoài các ngôn ngữ lập trình truyền thống, các kỹ năng về công nghệ như trí tuệ nhân tạo (AI), dữ liệu lớn (Big Data), điện toán đám mây, và bảo mật thông tin đang ngày càng được chú trọng. Kiến thức về blockchain và ứng dụng AI trong di động cũng là những điểm cộng lớn trong mắt nhà tuyển dụng.
3. **Phát triển đa nền tảng:** Các công cụ như React Native, Flutter, và Xamarin đang trở thành lựa chọn phổ biến để phát triển ứng dụng đa nền tảng, tiết kiệm thời gian và chi phí phát triển.
4. **Kỹ năng mềm và thực tiễn:** Ngoài khả năng lập trình, các công ty thường đánh giá cao những kỹ năng mềm như làm việc nhóm, quản lý dự án, và khả năng giao tiếp tốt. Việc có kinh nghiệm thực tế thông qua các dự án hoặc kỳ thực tập là yếu tố quan trọng để tạo ấn tượng với nhà tuyển dụng.