

Câu 1:

Android:

- Đặc điểm:
 - + Mã nguồn mở: Cho phép các nhà phát triển tùy chỉnh và xây dựng các phiên bản Android riêng của họ.
 - + Google Play Store: Nền tảng chính để phân phối và tải xuống các ứng dụng Android.
 - + Đa dạng thiết bị: Hỗ trợ nhiều loại thiết bị từ nhiều nhà sản xuất khác nhau, dẫn đến sự phong phú về thiết kế và giá cả.
 - + Tùy biến cao: Người dùng có thể tùy chỉnh giao diện, cài đặt các ứng dụng bên ngoài Google Play Store và thay đổi nhiều thiết lập hệ thống.
- Ưu điểm:
 - + Sự đa dạng về thiết bị và giá cả, phù hợp với nhiều đối tượng người dùng.
 - + Khả năng tùy biến và cài đặt ứng dụng linh hoạt.
 - + Cộng đồng phát triển mạnh mẽ và phong phú.
- Nhược điểm:
 - + Sự phân mảnh: Nhiều phiên bản Android cùng tồn tại làm khó khăn trong việc cập nhật và hỗ trợ.
 - + Bảo mật: Do tính mở và khả năng cài đặt ứng dụng từ nguồn không chính thống, nguy cơ bảo mật cao hơn.

iOS:

- Đặc điểm:
 - + Độc quyền: Chỉ có sẵn trên các thiết bị của Apple.
 - + AppStore: Nền tảng duy nhất để phân phối và tải xuống các ứng dụng cho iOS.
 - + Giao diện người dùng: Được thiết kế thân thiện và dễ sử dụng, với các ứng dụng tích hợp chất lượng cao.
 - + Bảo mật: Tính năng bảo mật mạnh mẽ, bao gồm mã hóa dữ liệu, bảo mật sinh trắc học (Face ID, Touch ID).
- Ưu điểm:
 - + Tính ổn định và hiệu năng cao.
 - + Bảo mật và bảo vệ quyền riêng tư mạnh mẽ.
 - + Hệ sinh thái Apple đồng nhất và tương thích cao giữa các thiết bị.

- Nhược điểm:
 - + 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.

Câu 2:

- Các nền tảng phổ biến hiện nay: Flutter, React native, Xamarin
- Điểm khác nhau:

Điểm khác nhau	Flutter	React Native	Xamarin
Ngôn ngữ	Dart	JavaScript + React	C# và .NET
Hiệu suất	Gần như native	Khá tốt nhưng có bridge	Tốt với Xamarin.iOS/Android
Ui	Widget tùy chỉnh	Sử dụng thành phần gốc	Xamarin.Forms hạn chế hơn
Cộng đồng	Tăng nhanh	Rất lớn	Nhỏ hơn
Thư viện hỗ trợ	Đang phát triển	Phong phú	Hạn chế hơn

Câu 3:

Hiệu suất cao và giao diện nhất quán: Flutter render giao diện trực tiếp và không phụ thuộc vào UI gốc, giúp hiệu suất gần như ứng dụng native và giao diện đồng nhất trên iOS và Android. Trong khi đó, React Native phụ thuộc vào cầu nối (bridge), còn Xamarin cũng phụ thuộc vào API gốc nhưng không đồng nhất bằng.

Hot Reload: Flutter hỗ trợ Hot Reload ổn định và mượt mà, giúp tăng tốc phát triển. React Native cũng có Hot Reload nhưng kém ổn định hơn; Xamarin hỗ trợ nhưng hơi chậm trên các dự án phức tạp.

Ngôn ngữ: Flutter dùng Dart, ngôn ngữ tối ưu cho giao diện; React Native dùng JavaScript (phổ biến nhưng dễ gặp lỗi runtime); Xamarin dùng C# (mạnh mẽ với .NET nhưng ít tùy biến cho UI).

Cộng đồng và tài liệu: Cộng đồng Flutter phát triển nhanh và được Google hỗ trợ mạnh. React Native có cộng đồng lớn nhất nhưng dễ gặp vấn đề tương thích. Xamarin có sự hỗ trợ từ Microsoft nhưng cộng đồng nhỏ hơn.

Hỗ trợ đa nền tảng mở rộng: Flutter hỗ trợ Android, iOS, web, desktop. React Native tập trung vào Android, iOS, với hỗ trợ web và desktop chưa hoàn thiện; Xamarin hỗ trợ iOS, Android và Windows nhưng thiếu nhất quán trên các nền tảng.

Câu 4:

Ngôn ngữ	Lý do được chọn
Java	Độ phổ biến cao, hỗ trợ hướng đối tượng tốt, bảo mật và dễ bảo trì.
Kotlin	Hiện đại, dễ đọc, tương thích với Java, null safety, và ưu tiên từ Google.
C++	Hiệu suất cao, hỗ trợ xử lý đồ họa, dùng cho mã native với Android NDK.
Dart	Tối ưu hóa hiệu suất với Flutter, hỗ trợ đa nền tảng, phát triển nhanh với Hot Reload.
JavaScript	Dễ tiếp cận với web developers, phát triển đa nền tảng với React Native, nhiều thư viện hỗ trợ.
Python	Dễ học, nhiều thư viện AI mạnh mẽ, thích hợp cho các dự án thử nghiệm.

Câu 5:

Ngôn ngữ	Lý do được chọn
Swift	Hiện đại, hiệu suất cao, hỗ trợ từ Apple, cú pháp an toàn và dễ đọc.
Objective-C	Tương thích với Swift, có sẵn nhiều thư viện, phù hợp với các ứng dụng cũ hoặc hiệu suất cao.
Dart	Đa nền tảng với Flutter, hiệu suất tốt, Hot Reload hỗ trợ phát triển nhanh.
JavaScript	Dễ tiếp cận với web developers, phát triển đa nền tảng với React Native, nhiều thư viện hỗ trợ.
C#	Dùng cho Xamarin, chia sẻ mã dễ dàng giữa iOS và Android, phù hợp với hệ sinh thái Microsoft.
Python	Dễ học, nhiều thư viện AI mạnh mẽ, thích hợp cho các ứng dụng thử nghiệm hoặc tích hợp AI.

Câu 6:

- **Thiếu ứng dụng:** Windows Phone thiếu nhiều ứng dụng phổ biến và cập nhật so với iOS và Android, khiến người dùng không có đủ lựa chọn và nhà phát triển cũng không hứng thú vì lợi nhuận dự kiến thấp.
- **Chậm cải tiến:** Hệ điều hành được cập nhật chậm chạp, giao diện không đổi mới đủ nhanh, khiến nó trở nên lỗi thời so với đối thủ.
- **Chiến lược kinh doanh thiếu nhất quán:** Microsoft thay đổi hướng đi nhiều lần, đặc biệt là sau khi mua Nokia, làm cho Windows Phone mất định hướng và khó cạnh tranh.
- **Hạn chế về phần cứng và đối tác:** Ít nhà sản xuất hỗ trợ Windows Phone, phần lớn chỉ có Nokia, khiến phần cứng không đa dạng và kém hấp dẫn.
- **Hệ sinh thái kém phát triển:** Windows Phone không kết nối tốt với các sản phẩm và dịch vụ khác của Microsoft, không tạo được hệ sinh thái thu hút người dùng.

Câu 7:

Ngôn ngữ	Lý do được chọn
Ngôn ngữ	HTML, CSS, JavaScript, TypeScript, Dart
Framework UI	React, Vue.js, Angular
Framework đa nền tảng	Flutter, React Native, Ionic, Apache Cordova, Xamarin
Công cụ PWA	Workbox, Webpack, Capacitor
Backend và API	Node.js, Firebase, GraphQL

Câu 8: mức lương: khoảng 10 tr