

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

✧ **Android (Google)**

*Đặc điểm:* Có thể tùy chỉnh rộng rãi và được nhiều nhà sản xuất thiết bị sử dụng.

*Ưu điểm:* Nhiều sự lựa chọn, tùy chỉnh mạnh mẽ, đa dạng ứng dụng, tính tùy biến cao, hỗ trợ phần cứng phong phú

*Khuyết điểm:* Bảo mật kém, tối ưu hóa chưa được tốt

✧ **iOS (Apple)**

*Đặc điểm:* Đơn giản, đồng nhất

*Ưu điểm:* Bảo mật, hiệu năng cao, tích hợp tốt với các thiết bị Apple

*Khuyết điểm:* Thiết lập hạn chế, thiếu linh hoạt, giá thành cao

✧ **BlackBerry OS**

*Đặc điểm:* bảo mật, nhiều tính năng hỗ trợ doanh nhân

*Ưu điểm:* Bảo mật cao, dịch vụ nhắn tin & email bảo mật và được mã hóa

*Khuyết điểm:* Thiếu ứng dụng, không phù hợp với xu hướng cảm ứng, khá lỗi thời

✧ **Windows Phone (Microsoft)**

*Đặc điểm:* Đơn giản, đồng nhất

*Ưu điểm:* Tích hợp tốt với các dịch vụ Microsoft

*Khuyết điểm:* Thiếu ứng dụng, hỗ trợ kém hơn so với Android và iOS

✧ **HarmonyOS (Huawei)**

*Đặc điểm:* Có thể chạy trên nhiều loại thiết bị như điện thoại, máy tính bảng, và thiết bị IoT.

*Ưu điểm:* đa nền tảng, Tùy biến tốt

*Khuyết điểm:* Hạn chế về ứng dụng, Phụ thuộc vào Huawei

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

Nền tảng phát triển	Hiệu năng	Độ phức tạp khi phát triển
Flutter	Cao	Trung bình
React Native	Trung bình	Trung bình
Xamarin	Cao	Cao
Ionic	Thấp	Thấp

**Câu 3:** Flutter trở thành một lựa chọn phổ biến trong việc phát triển ứng dụng đa nền tảng vì nhiều lý do như:

- *Khả năng phát triển đa nền tảng:* Flutter hỗ trợ đa nền tảng từ ứng dụng Android, iOS đến web, desktop (Windows, macOS, Linux) giúp cho các doanh nghiệp muốn tiết kiệm chi phí và thời gian phát triển
- *Hiệu suất cao:* Flutter được viết bằng ngôn ngữ Dart nên có hiệu suất cao và chạy mượt mà trên nhiều
- *Giao diện đồng nhất và linh hoạt:* Flutter cung cấp UI framework riêng (widgets), giúp các nhà phát triển xây dựng giao diện nhất quán trên cả

Android, iOS, web và desktop. Thư viện widget phong phú và dễ tùy biến của Flutter cho phép thiết kế giao diện đẹp mắt và gần như giống hệt trên mọi nền tảng, tạo ra trải nghiệm người dùng mượt mà và đồng nhất.

- *Hot-Reload*: giúp tăng tốc quá trình phát triển Hot-reload là tính năng nổi bật của Flutter, giúp nhà phát triển xem ngay lập tức các thay đổi trong mã nguồn mà không cần khởi động lại toàn bộ ứng dụng. Tính năng này giúp tiết kiệm thời gian, tăng tốc độ phát triển, và thử nghiệm nhanh chóng các ý tưởng thiết kế mới.
- *Cộng đồng và hệ sinh thái phong phú*: Flutter có cộng đồng phát triển lớn và ngày càng mở rộng, hỗ trợ nhiều tài liệu, tài nguyên, và plugin sẵn có. Google liên tục cập nhật Flutter với các tính năng mới, mở rộng sang các nền tảng khác như web và desktop, tăng thêm tính linh hoạt cho các dự án phát triển.
- *Hỗ trợ cho nhiều plugin và dễ tích hợp*: Flutter cung cấp nhiều plugin để kết nối với các tính năng phần cứng và hệ thống (như camera, vị trí, bộ nhớ). Ngoài ra, có thể viết mã native để tích hợp các tính năng phức tạp, giúp ứng dụng trở nên mạnh mẽ và dễ dàng tương tác với các API hệ thống.
- *Sự hỗ trợ từ Google*: Là sản phẩm của Google, Flutter nhận được sự đầu tư mạnh mẽ từ một trong những công ty công nghệ hàng đầu thế giới, giúp tăng độ tin cậy và cập nhật liên tục, tạo nên một cộng đồng phát triển ổn định và mạnh mẽ. nền tảng.

**Câu 4:** Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android:

- ✧ *Kotlin*: là ngôn ngữ chính thức cho Android từ năm 2017 và được Google khuyến khích sử dụng. Kotlin cải thiện trải nghiệm lập trình so với Java, cung cấp cú pháp ngắn gọn, hiện đại và các tính năng mạnh mẽ như xử lý null (null safety), lập trình hàm, và coroutine để quản lý các tác vụ không đồng bộ.
- ✧ *Java*: là ngôn ngữ chính thức đầu tiên của Android và là ngôn ngữ được sử dụng lâu nhất trong phát triển Android, giúp nó có cộng đồng lớn, tài liệu phong phú và thư viện mạnh mẽ.
- ✧ *C++*: được sử dụng thông qua Android NDK (Native Development Kit) khi cần xây dựng các phần của ứng dụng đòi hỏi hiệu suất cao, đặc biệt là các tác vụ xử lý đồ họa, âm thanh, hoặc tính toán phức tạp.
- ✧ *Dart (Flutter)*: là ngôn ngữ chính của Flutter, framework phát triển ứng dụng đa nền tảng của Google. Flutter cho phép phát triển ứng dụng Android với giao diện đẹp và hiệu suất tốt từ một mã nguồn duy nhất, hỗ trợ cả Android và iOS.
- ✧ *JavaScript*: được sử dụng thông qua các framework như React Native hoặc Ionic để phát triển ứng dụng đa nền tảng. Điều này cho phép các lập

trình viên JavaScript tận dụng kỹ năng web để phát triển ứng dụng Android và iOS từ một mã nguồn.

**Câu 5:** Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS:

- Swift,
- Objective-C,
- Dart (Flutter),
- JavaScript (React Native),
- C++.

**Câu 6:** Windows Phone đã phải đối mặt với nhiều thách thức trong cuộc cạnh tranh với Android và iOS. Những nguyên nhân dẫn đến sự sụt giảm thị phần và thất bại của Windows Phone như:

- Số lượng ứng dụng hạn chế và Thiếu sự hỗ trợ từ các nhà phát triển,
- Thiếu tính năng nổi bật để cạnh tranh,
- Cải tiến và cập nhật chậm trễ,
- Thiếu định hướng rõ ràng từ Microsoft,
- Android và iOS chiếm ưu thế lớn,
- Giao diện không phù hợp với thị hiếu người dùng,
- Thiếu sự đa dạng và hấp dẫn trong phần cứng.

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

- Ngôn ngữ:
  - HTML
  - CSS
  - JavaScript
  - Dart
- Công cụ & Framework:
  - React.js
  - Vue.js
  - Angular
  - SASS/SCSS
  - PWA (như Workbox và Lighthouse)
  - Jest, Cypress, và Postman

**Câu 8:** Nhu cầu về nguồn nhân lực lập trình viên trên thiết bị di động hiện nay đang tăng mạnh và có sự phát triển không ngừng, đặc biệt trong bối cảnh chuyển đổi số và sự phát triển của công nghệ di động.

- Kỹ năng lập trình với các ngôn ngữ phổ biến :

- ✧ *Java và Kotlin (cho Android)*: Đây là hai ngôn ngữ chính để phát triển ứng dụng Android. Kotlin ngày càng trở nên phổ biến do sự hỗ trợ tốt từ Google và tính linh hoạt cao.
- ✧ *Swift (cho iOS)*: Là ngôn ngữ chính để phát triển ứng dụng iOS, Swift giúp tối ưu hóa hiệu suất và dễ bảo trì ứng dụng.
- ✧ *Dart (cho Flutter)*: Với sự phát triển mạnh mẽ của Flutter, các lập trình viên có nhu cầu sử dụng Dart để phát triển ứng dụng đa nền tảng.

#### - Kinh nghiệm với Frameworks và Công cụ phát triển:

- ✧ *React Native*: Giúp phát triển ứng dụng di động đa nền tảng với JavaScript. Đây là một trong những công cụ phổ biến nhất cho việc phát triển ứng dụng di động.
- ✧ *Flutter*: Framework phát triển ứng dụng đa nền tảng từ Google, sử dụng Dart, đang ngày càng được ưa chuộng nhờ khả năng phát triển nhanh và hiệu suất cao.
- ✧ *Xcode và Android Studio*: Là các công cụ phát triển chính cho ứng dụng iOS và Android, lập trình viên cần phải thành thạo trong việc sử dụng các IDE này.

#### - Kiến thức về thiết kế giao diện người dùng (UI/UX):

- ✧ Các lập trình viên cần có khả năng phát triển giao diện người dùng mượt mà, trực quan và dễ sử dụng.
- ✧ Kỹ năng về material design (cho Android) và Human Interface Guidelines (cho iOS) rất quan trọng.
- ✧ Hiểu biết về responsive design và các phương pháp thiết kế giúp tối ưu hóa trải nghiệm người dùng trên nhiều kích thước màn hình.

#### - Kỹ năng quản lý dữ liệu và API

- ✧ Lập trình viên cần có kiến thức về các cơ sở dữ liệu như SQLite, Realm, và Firebase. Việc tích hợp API và xử lý dữ liệu trong thời gian thực là một kỹ năng quan trọng.
- ✧ RESTful APIs và GraphQL: Kỹ năng sử dụng các API này rất quan trọng để kết nối ứng dụng di động với server và cơ sở dữ liệu.

#### - Kiến thức về Kiểm thử và Debugging

- ✧ Kiểm thử là một kỹ năng không thể thiếu trong phát triển ứng dụng di động. Kỹ năng sử dụng các công cụ như JUnit, Espresso (Android), và XCTest (iOS) giúp đảm bảo chất lượng ứng dụng.

✧ Automated testing và CI/CD (Continuous Integration/Continuous Deployment): Kiến thức về tự động hóa quy trình phát triển giúp cải thiện hiệu suất làm việc và chất lượng sản phẩm.

- Hiểu biết về bảo mật di động: Bảo mật dữ liệu người dùng là một yếu tố rất quan trọng trong phát triển ứng dụng di động. Kiến thức về mã hóa dữ liệu, bảo mật API và các phương pháp phòng ngừa tấn công như SQL injection, XSS là cần thiết.

- Kỹ năng làm việc nhóm và giao tiếp: Phát triển ứng dụng di động thường phụ thuộc đến việc làm việc nhóm với các nhà thiết kế, lập trình viên backend, và các nhóm quản lý sản phẩm. Kỹ năng giao tiếp và làm việc nhóm là cực kỳ quan trọng để đảm bảo dự án hoàn thành đúng tiến độ và yêu cầu.

- Khả năng tối ưu hiệu suất: Các ứng dụng di động cần phải chạy nhanh và hiệu quả trên nhiều loại thiết bị. Kỹ năng tối ưu hóa mã nguồn và sử dụng bộ nhớ, cũng như quản lý tài nguyên hệ thống hiệu quả, là rất quan trọng.

- Kiến thức về công nghệ mới: 5G và AI/ML (Artificial Intelligence/Machine Learning): Với sự phát triển của các công nghệ mới, lập trình viên di động cần có khả năng tích hợp các công nghệ này vào ứng dụng để mang lại trải nghiệm người dùng tốt hơn.