

Bài làm

**Câu 1:** Hãy cho biết các nền tảng cho thiết bị di động thông minh hiện nay? Với mỗi nền tảng hãy cho biết đặc điểm, ưu và khuyết điểm.

Nền tảng	Đặc điểm	Ưu điểm	Khuyết điểm
Android	<ul style="list-style-type: none"> <li>- Là nền tảng mã nguồn mở, phát triển bởi Google, sử dụng ngôn ngữ Java và Kotlin.</li> <li>- Android có giao diện tùy chỉnh và hệ sinh thái ứng dụng đa dạng.</li> </ul>	<ul style="list-style-type: none"> <li>- Tính tùy biến cao, cho phép nhà sản xuất hoặc người dùng dễ dàng tùy chỉnh giao diện và chức năng.</li> <li>- Hỗ trợ nhiều phân khúc thiết bị từ giá rẻ đến cao cấp.</li> <li>- Kho ứng dụng Google Play phong phú với hàng triệu ứng dụng và game.</li> </ul>	<ul style="list-style-type: none"> <li>- Nhiều biến thể Android trên các thiết bị khác nhau, gây phân mảnh phần mềm và khó khăn cho nhà phát triển trong việc tối ưu ứng dụng.</li> <li>- Tình trạng bảo mật kém hơn iOS, dễ bị nhiễm phần mềm độc hại.</li> </ul>
IOS	<ul style="list-style-type: none"> <li>- Là nền tảng độc quyền của Apple, được phát triển cho các thiết bị iPhone, iPad và</li> </ul>	<ul style="list-style-type: none"> <li>- Hiệu suất cao và ổn định nhờ sự tối ưu giữa phần cứng và phần mềm do Apple kiểm soát.</li> </ul>	<ul style="list-style-type: none"> <li>- Tính tùy biến thấp hơn Android, người dùng không thể tùy chỉnh giao diện hoặc cài đặt</li> </ul>

	iPod Touch. - IOS tập trung vào trải nghiệm người dùng mượt mà và nhất quán.	- Bảo mật tốt, với các tính năng bảo mật cao và kiểm soát chặt chẽ trên App Store. - Cập nhật phần mềm đồng loạt cho tất cả các thiết bị.	các ứng dụng ngoài App Store. - Giá thiết bị IOS thường cao hơn so với các thiết bị Android cùng phân khúc.
Windows Phone	Trước đây là hệ điều hành của Microsoft cho các thiết bị di động, tuy nhiên đã bị ngừng phát triển vào năm 2019.	- Giao diện Metro UI độc đáo, dễ sử dụng. - Tính ổn định cao, tích hợp tốt với các dịch vụ của Microsoft như Office, OneDrive.	- Cộng đồng ứng dụng ít, không có nhiều ứng dụng so với Android và iOS. - Microsoft đã ngừng hỗ trợ, nên không có cập nhật mới.

**Câu 2:** Liệt kê các nền tảng phát triển ứng dụng di động phổ biến hiện nay và so sánh sự khác biệt chính giữa chúng.

**- React Native:**

- Ưu điểm:
  - ✓ Hiệu suất gần giống như ứng dụng native.
  - ✓ Cộng đồng lớn, dễ dàng tìm tài liệu và plugin.
  - ✓ Dễ dàng tích hợp với các ứng dụng web hiện có, thích hợp cho việc chia sẻ logic với các ứng dụng web qua JavaScript.
- Khuyết điểm:

- ✓ Cần các gói và plugin bên thứ ba để truy cập nhiều tính năng native, dễ gây lỗi hoặc không ổn định.
- ✓ Hiệu suất không hoàn toàn bằng ứng dụng native, đặc biệt với các tác vụ phức tạp.

#### - **Flutter:**

- Ưu điểm:
  - ✓ Giao diện nhất quán và đẹp mắt trên mọi nền tảng.
  - ✓ Hiệu suất cao, gần bằng với native nhờ vào các thành phần UI được render trực tiếp.
  - ✓ Hỗ trợ "hot reload" giúp tăng tốc phát triển và thử nghiệm.
- Khuyết điểm:
  - ✓ Kích thước ứng dụng lớn hơn so với native.
  - ✓ Dart là ngôn ngữ ít phổ biến, đòi hỏi người dùng phải học thêm nếu chưa biết.

#### - **Xamarin:**

- Ưu điểm:
  - ✓ Cho phép tái sử dụng nhiều mã nguồn, giúp tiết kiệm thời gian phát triển.
  - ✓ Dễ dàng tích hợp với hệ sinh thái Microsoft.
  - ✓ Hỗ trợ tốt với các ứng dụng doanh nghiệp.
- Khuyết điểm:
  - ✓ Hiệu suất thấp hơn so với React Native hoặc Flutter.
  - ✓ Kích thước ứng dụng có thể lớn, nhất là với ứng dụng dùng Xamarin.Forms.

**Câu 3:** Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng? So sánh với các nền tảng khác như React Native và Xamarin.

- Lý do Flutter trở nên phổ biến:

- **Hiệu năng cao:** Flutter sử dụng công cụ đồ họa Skia để render giao diện trực tiếp trên canvas, thay vì thông qua WebView (như trong Ionic) hoặc cầu nối JavaScript (như trong React Native). Điều này giúp giảm độ trễ và mang lại hiệu năng mượt mà, gần bằng ứng dụng native.
- **Giao diện nhất quán:** Flutter cung cấp các widget tùy chỉnh sẵn, không phụ thuộc vào thành phần UI của hệ điều hành, giúp giao diện đồng nhất trên cả Android và iOS.
- **"Hot Reload" nhanh chóng:** Tính năng này cho phép nhà phát triển thấy ngay thay đổi trong mã nguồn mà không cần phải chạy lại ứng dụng, giúp tăng tốc độ phát triển và thử nghiệm.
- **Cộng đồng và sự hỗ trợ từ Google:** Google tích cực hỗ trợ và cải tiến Flutter, cộng đồng phát triển lớn mạnh, có nhiều thư viện và plugin sẵn có.

- So sánh với React Native và Xamarin:

Tiêu chí	Flutter	React Native	Xamarin
Ngôn ngữ	Dart	JavaScript	C#
Hiệu năng	Cao (gần bằng native)	Tốt, nhưng phụ thuộc vào cầu nối JS	Tốt, nhưng không bằng Flutter
Giao diện	Widget tùy chỉnh nhất quán	Thành phần native	Thành phần native hoặc Xamarin.Forms
Khả năng tùy chỉnh	Rất linh hoạt và dễ dàng	Dễ dàng, nhưng cần plugin native	Tùy biến tốt, đặc biệt với ứng dụng doanh nghiệp
Tích hợp với	Tốt, nhưng đôi	Tốt, cần các	Tích hợp sâu với

<b>nền tảng gốc</b>	khi cần plugin	bridge JS-to-native	hệ sinh thái Microsoft
<b>Hot Reload</b>	Có, rất mượt	Có	Có, nhưng không mượt bằng Flutter
<b>Dung lượng ứng dụng</b>	Khá lớn	Trung bình	Lớn
<b>Cộng đồng và hỗ trợ</b>	Lớn và tăng mạnh	Lớn và ổn định	Tốt, nhưng đang chậm lại

**Câu 4:** Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android và giải thích tại sao chúng lại được chọn.

#### 1. Java

- Lý do chọn Java:

- Là ngôn ngữ truyền thống của Android: Java đã được sử dụng từ khi Android ra đời, là nền tảng cho hầu hết các thư viện, API và tài liệu Android.
- Cộng đồng lớn: Với lịch sử lâu đời, Java có cộng đồng lập trình viên lớn, dễ dàng tìm kiếm tài liệu và hỗ trợ kỹ thuật.
- Bảo trì và mở rộng dễ dàng: Java có tính chất hướng đối tượng, giúp việc bảo trì và mở rộng mã nguồn trở nên dễ dàng, là lựa chọn tốt cho các ứng dụng phức tạp.
- Hiệu suất tốt: Với máy ảo JVM, Java cung cấp hiệu suất ổn định, đáp ứng tốt các yêu cầu của ứng dụng Android.

#### 2. Kotlin

- Lý do chọn Kotlin:

- Được Google chính thức hỗ trợ: Năm 2017, Google chính thức công nhận Kotlin là ngôn ngữ phát triển Android, giúp các tính năng mới được cập nhật đầy đủ.
- Cú pháp ngắn gọn, hiện đại: Kotlin có cú pháp ngắn gọn, dễ hiểu, giảm thiểu các lỗi phổ biến trong Java và giúp tăng hiệu quả lập trình.
- Tương thích với Java: Kotlin có thể dễ dàng tích hợp với Java, cho phép các ứng dụng Java hiện có chuyển dần sang Kotlin mà không phải thay đổi toàn bộ mã.
- An toàn null: Kotlin xử lý các ngoại lệ liên quan đến null một cách an toàn, giảm nguy cơ lỗi và crash cho ứng dụng.

### 3. Dart (qua Flutter)

- Lý do chọn Dart:

- Tương thích với Flutter: Dart là ngôn ngữ chính của Flutter, framework đa nền tảng do Google phát triển, giúp tạo ứng dụng cho cả Android và iOS từ một mã nguồn duy nhất.
- Hiệu năng tốt: Dart sử dụng cơ chế biên dịch Ahead-of-Time (AOT) cho hiệu suất mượt mà, giúp ứng dụng đạt tốc độ gần với ứng dụng native.
- Hot Reload nhanh: Dart hỗ trợ tính năng hot reload của Flutter, giúp phát triển ứng dụng nhanh hơn.

### 4. JavaScript (qua React Native)

- Lý do chọn JavaScript:

- Được hỗ trợ qua React Native: JavaScript là ngôn ngữ chính của React Native, một framework đa nền tảng giúp xây dựng ứng dụng Android và iOS từ một mã nguồn duy nhất.
- Phù hợp cho ứng dụng web lẫn di động: Các công ty có ứng dụng web dễ dàng chuyển sang ứng dụng di động mà không cần viết lại toàn bộ mã.

- Cộng đồng rộng lớn: JavaScript có cộng đồng lớn và nhiều thư viện hỗ trợ, giúp phát triển ứng dụng nhanh chóng.

**Câu 5:** Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS.

- Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS bao gồm: Swift, Objective-C, C++, Dart (qua Flutter), Javascript (qua React Native).

**Câu 6:** Hãy thảo luận về những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó.

### **1. Thiếu hụt ứng dụng và sự hỗ trợ của nhà phát triển**

- Thiếu ứng dụng phổ biến: Một trong những yếu tố quyết định đến sự thành công của hệ điều hành di động là kho ứng dụng phong phú. Windows Phone không có đủ các ứng dụng phổ biến như Android và iOS, gây khó khăn trong việc thu hút người dùng. Một số ứng dụng thiết yếu như Instagram, Snapchat, và các ứng dụng Google không có trên nền tảng này, hoặc chỉ có phiên bản kém chất lượng, lỗi thời.
- Thiếu sự hỗ trợ từ các nhà phát triển: Các nhà phát triển không hào hứng với việc viết ứng dụng cho Windows Phone vì thị phần nhỏ của nó, dẫn đến vòng lặp khó khăn trong việc thu hút cả người dùng và nhà phát triển.

### **2. Thiếu sự đổi mới và phát triển từ Microsoft**

- Thay đổi chiến lược không nhất quán: Microsoft đã nhiều lần thay đổi chiến lược phát triển Windows Phone, khiến người dùng và nhà phát triển mất niềm tin vào nền tảng này. Ví dụ, việc chuyển từ Windows Phone 7 lên Windows Phone 8 không hỗ trợ đầy đủ các thiết bị cũ đã khiến một lượng người dùng trung thành rời bỏ.
- Tốc độ cập nhật chậm: Trong khi Android và iOS liên tục được cập nhật với các tính năng mới và cải thiện hiệu năng, Windows Phone

thường chậm trễ trong việc phát hành bản cập nhật, khiến nền tảng này kém hấp dẫn hơn.

### **3. Hệ sinh thái và thị trường hạn chế**

- Thiếu sự tích hợp tốt với các thiết bị khác: Microsoft đã không thể xây dựng một hệ sinh thái đa dạng và gắn kết như Apple hoặc Google. Trong khi Apple có thể kết nối liền mạch các thiết bị iPhone, iPad, và MacBook, Windows Phone không cung cấp được sự tiện lợi này.
- Sự cạnh tranh khốc liệt từ Android và iOS: Thị trường di động bị chi phối bởi hai nền tảng lớn là iOS và Android, khiến Windows Phone khó cạnh tranh. Các nhà sản xuất thiết bị cũng không quá mặn mà với Windows Phone, trong khi các sản phẩm Android thường có giá thành tốt hơn.

### **4. Thiếu khả năng tùy chỉnh và trải nghiệm người dùng hạn chế**

- Giao diện và khả năng tùy biến hạn chế: Mặc dù Windows Phone có giao diện Live Tiles độc đáo, nhưng lại hạn chế khả năng tùy biến so với Android. Điều này làm giảm sức hút đối với người dùng thích tự do tùy chỉnh điện thoại của mình.
- Tính năng kém phong phú: Windows Phone thiếu nhiều tính năng và ứng dụng hỗ trợ từ bên thứ ba so với các nền tảng đối thủ, làm giảm sức hấp dẫn của nó.

### **5. Chiến lược phân phối và marketing yếu**

- Chiến lược hợp tác kém: Microsoft không tạo được mối quan hệ chặt chẽ với các nhà sản xuất điện thoại di động ngoài Nokia. Sau khi Microsoft mua lại Nokia, một trong những nhà sản xuất lớn của nền tảng Windows Phone, sự phụ thuộc vào một đối tác duy nhất càng làm nền tảng này thêm khó cạnh tranh.
- Marketing kém hiệu quả: Các chiến dịch tiếp thị của Microsoft không đủ hấp dẫn và không đủ mạnh để cạnh tranh với các chiến dịch lớn của Apple và Google.



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

### **1. Ngôn ngữ Front-End (Giao diện người dùng)**

- HTML, CSS, JavaScript

- HTML: Ngôn ngữ đánh dấu cơ bản giúp xây dựng cấu trúc của trang web.
- CSS: Dùng để thiết kế giao diện người dùng, định dạng và tạo kiểu cho trang web.
- JavaScript: Ngôn ngữ lập trình cho phép bạn thêm tính năng động vào trang web, như xử lý sự kiện, tạo tương tác người dùng.

- Các thư viện và framework JavaScript

- React: Thư viện JavaScript phổ biến để xây dựng giao diện người dùng, đặc biệt với khả năng tái sử dụng component, rất hiệu quả khi phát triển ứng dụng web trên di động.
- Vue.js: Framework nhẹ và dễ học, dùng để xây dựng các ứng dụng giao diện người dùng mạnh mẽ với khả năng tối ưu hóa cho di động.
- Angular: Framework mạnh mẽ của Google, được sử dụng để phát triển các ứng dụng web phức tạp và dễ dàng tối ưu cho thiết bị di động.
- Svelte: Framework hiện đại giúp tạo ra các ứng dụng với hiệu suất cao, giảm tải cho trình duyệt và tối ưu hóa trải nghiệm di động.

### **2. Công cụ và Thư viện Mobile-First (Thiết kế cho di động)**

- Bootstrap: Framework CSS phổ biến, giúp xây dựng giao diện responsive (phù hợp với mọi kích thước màn hình) dễ dàng. Bootstrap hỗ trợ các yếu tố thiết kế di động ngay từ đầu.
- Tailwind CSS: Một framework CSS utility-first giúp bạn tạo ra các giao diện di động linh hoạt, dễ dàng tùy chỉnh theo nhu cầu của ứng dụng.

- Ionic: Framework xây dựng ứng dụng di động sử dụng HTML, CSS và JavaScript. Ionic cho phép phát triển ứng dụng web mobile-first và hỗ trợ tích hợp với các API native (tính năng gốc) như camera, GPS.
- Foundation: Một framework responsive mạnh mẽ và linh hoạt, phù hợp với việc phát triển ứng dụng web di động với nhiều tính năng tùy chỉnh.

### **3. Ngôn ngữ và Công cụ Back-End (Máy chủ)**

- Node.js: Là môi trường runtime JavaScript, cho phép phát triển các ứng dụng back-end sử dụng JavaScript. Node.js rất phổ biến cho việc xây dựng các API RESTful, giúp kết nối ứng dụng web với cơ sở dữ liệu và các dịch vụ khác.
- PHP: Một ngôn ngữ lập trình phổ biến cho phát triển web, thường được dùng để xây dựng ứng dụng web động. PHP có thể dễ dàng tích hợp với cơ sở dữ liệu như MySQL, PostgreSQL.
- Ruby on Rails: Framework back-end được viết bằng Ruby, rất phù hợp cho việc phát triển nhanh các ứng dụng web với cấu trúc rõ ràng và dễ duy trì.
- Python (Django, Flask): Python là một ngôn ngữ phát triển mạnh mẽ cho back-end. Django và Flask là hai framework Python phổ biến, giúp xây dựng ứng dụng web mạnh mẽ và dễ dàng mở rộng.
- Java (Spring Boot): Framework Java rất phổ biến trong việc xây dựng các ứng dụng web lớn và phức tạp. Spring Boot giúp phát triển các API hiệu quả, có thể tích hợp với nhiều dịch vụ và cơ sở dữ liệu.

### **4. Công cụ Phát triển Web Di Động (Mobile Web Development Tools)**

- Progressive Web Apps (PWA): Đây là một kiểu ứng dụng web giúp cung cấp trải nghiệm giống như ứng dụng di động thông qua các tính năng như offline và thông báo push. Bạn có thể xây dựng PWA bằng cách sử dụng HTML, CSS, và JavaScript.

- **React Native:** Một framework JavaScript do Facebook phát triển, cho phép bạn viết ứng dụng di động (iOS và Android) bằng cách sử dụng React. React Native sử dụng web technologies, nhưng mã được biên dịch thành mã native, giúp cải thiện hiệu suất và khả năng sử dụng các tính năng gốc.
- **Flutter:** Một framework của Google, mặc dù chủ yếu được dùng để phát triển ứng dụng di động native, nhưng cũng hỗ trợ phát triển ứng dụng web. Flutter cho phép bạn viết mã cho cả web và di động từ một codebase duy nhất.
- **PhoneGap/Cordova:** Đây là công cụ phát triển ứng dụng di động sử dụng HTML, CSS và JavaScript, cho phép bạn xây dựng ứng dụng mobile web và có thể chuyển đổi thành ứng dụng native sử dụng một nền tảng chung.

## **5. Công cụ kiểm tra và tối ưu hóa hiệu suất di động**

- **Lighthouse:** Công cụ mã nguồn mở của Google dùng để kiểm tra hiệu suất, khả năng truy cập và tối ưu hóa SEO cho ứng dụng web, đặc biệt là đối với các ứng dụng web di động.
- **Chrome DevTools:** Công cụ phát triển của Google Chrome cho phép bạn kiểm tra ứng dụng web trên các thiết bị di động giả lập, tối ưu hóa và debug mã nguồn.
- **BrowserStack:** Công cụ kiểm thử ứng dụng web trên nhiều thiết bị và trình duyệt khác nhau, giúp bạn kiểm tra tính tương thích với thiết bị di động.

## **6. Công cụ hỗ trợ quản lý và triển khai ứng dụng web di động**

- **GitHub:** Công cụ quản lý mã nguồn phổ biến, giúp quản lý và cộng tác trong việc phát triển ứng dụng web cho di động.
- **Netlify:** Nền tảng triển khai cho các ứng dụng web, bao gồm hỗ trợ cho Progressive Web Apps và các ứng dụng di động.

- Firebase: Dịch vụ của Google cung cấp backend-as-a-service, hỗ trợ xây dựng các ứng dụng web và di động, với các tính năng như cơ sở dữ liệu thời gian thực, xác thực người dùng và lưu trữ.

**Câu 8:** Nghiên cứu về nhu cầu nguồn nhân lực lập trình viên trên thiết bị di động hiện nay và những kỹ năng được yêu cầu nhiều nhất.

- Nhu cầu về lập trình viên ứng dụng di động vẫn rất mạnh mẽ khi việc sử dụng smartphone và các ứng dụng di động tiếp tục gia tăng. Các kỹ năng chính cần có cho lập trình viên di động vào năm 2024 bao gồm thành thạo các ngôn ngữ lập trình, kiến thức về các framework phát triển, và hiểu biết về các xu hướng di động hiện đại như AI, AR/VR và IoT.
- Đối với phát triển Android, lập trình viên cần thành thạo Java và Kotlin, với Kotlin là ngôn ngữ được ưa chuộng hơn nhờ cú pháp hiện đại. Đối với phát triển iOS, Swift là ngôn ngữ chính. Lập trình viên cũng cần quen thuộc với các công cụ như Android Studio cho Android và Xcode cho iOS.
- Các lập trình viên di động ngày càng yêu cầu có kỹ năng trong các framework đa nền tảng như Flutter và React Native. Flutter, ví dụ, cho phép lập trình viên viết mã một lần và triển khai trên nhiều nền tảng (iOS, Android, web và desktop), giúp tiết kiệm thời gian và tài nguyên. React Native cũng cho phép chức năng tương tự nhưng phổ biến hơn với việc tận dụng JavaScript. Xamarin là một công cụ đa nền tảng đáng chú ý khác, được ưa chuộng bởi các lập trình viên sử dụng C#.
- Các kỹ năng bổ sung như quản lý cơ sở dữ liệu, tích hợp API và thực hành bảo mật tốt nhất là rất quan trọng khi các ứng dụng di động xử lý dữ liệu nhạy cảm và yêu cầu giao tiếp thời gian thực. Hơn nữa, việc quen thuộc với các công nghệ mới nổi như Máy học và AR/VR đang trở nên ngày càng quan trọng.

- Ngoài các kỹ năng kỹ thuật, các kỹ năng mềm như giao tiếp, làm việc nhóm và khả năng thích ứng cũng rất được coi trọng, vì chúng giúp tăng cường hiệu quả hợp tác trong các nhóm phát triển và với khách hàng.