

## **Câu 1:**

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

- Android, IOS, Window Phone

Nền tảng Android:

- Đặc điểm: Là một hệ điều hành mở của Google dễ sử dụng
- Ưu điểm: Giao diện thân thiện, dễ cài đặt ứng dụng ngoài
- Nhược điểm: Không bảo mật cao

## **Câu 2:**

### **Nền Tảng Bản Địa (Native)**

- **Ưu điểm:**
  - Hiệu năng cao nhất
  - Truy cập đầy đủ vào các tính năng của thiết bị
  - Giao diện người dùng mượt mà, trải nghiệm người dùng tốt
- **Nhược điểm:**
  - Phải viết mã riêng cho từng nền tảng (iOS, Android)
  - Thời gian phát triển lâu hơn
  - Chi phí phát triển cao hơn
- **Công cụ:**
  - **iOS:** Xcode, Swift
  - **Android:** Android Studio, Kotlin/Java

### **Nền Tảng Đa Nền Tảng (Cross-Platform)**

- **Ưu điểm:**
  - Chỉ cần viết một lần mã, chạy được trên nhiều nền tảng
  - Tiết kiệm thời gian và chi phí phát triển
  - Dễ dàng cập nhật và bảo trì
- **Nhược điểm:**
  - Hiệu năng có thể kém hơn so với ứng dụng bản địa

- Khả năng truy cập các tính năng thiết bị có thể hạn chế
- Giao diện người dùng có thể không được tự nhiên trên từng nền tảng
- **Công cụ:**
  - **React Native:** Sử dụng JavaScript và React để xây dựng giao diện người dùng, sau đó biên dịch thành các thành phần native.
  - **Flutter:** Sử dụng ngôn ngữ Dart để xây dựng giao diện người dùng, biên dịch thành code native cho nhiều nền tảng.
  - **Xamarin:** Sử dụng C# và .NET để xây dựng ứng dụng, chia sẻ phần lớn mã giữa các nền tảng.
  - **Ionic:** Sử dụng web technologies (HTML, CSS, JavaScript) để xây dựng ứng dụng, sau đó đóng gói vào một ứng dụng native.

### Câu 3:

Những lý do khiến Flutter trở thành lựa chọn cho việc phát triển ứng dụng đa nền tảng:

- Flutter sử dụng một engine đồ họa riêng, giúp ứng dụng chạy mượt mà và nhanh chóng trên các thiết bị khác nhau.
- Với Flutter, bạn chỉ cần viết một lần mã nguồn và có thể biên dịch để chạy trên cả iOS và Android, thậm chí cả web và desktop.
- Flutter có một cộng đồng người dùng lớn mạnh, sẵn sàng hỗ trợ và chia sẻ kinh nghiệm.

Với các nền tảng như React Native và Xamarin sẽ có các so sánh như sau:

	Flutter	React Native	Xamarin
Ngôn ngữ	Dart	JavaScript	C#
Hiệu năng	Cao	Tốt	Tốt
Giao diện người dùng	Phong phú, tùy biến cao	Tốt	Tốt
Cộng đồng	Lớn, đang phát triển	Lớn	Nhỏ hơn
Thư viện	Đang phát triển	Rất nhiều	Nhiều
Tái sử dụng mã	Khá tốt	Tốt (với web)	Tốt (trong .NET ecosystem)

#### Câu 4:

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

- Kotlin, Java, C++

#### Tại sao các ngôn ngữ này được lựa chọn?

- **Hiệu năng:** Các ngôn ngữ trên đều có thể tạo ra các ứng dụng Android có hiệu năng cao, đáp ứng được các yêu cầu của người dùng.
- **Khả năng tương tác:** Chúng có thể tương tác với các phần code được viết bằng các ngôn ngữ khác, giúp bạn tận dụng lại code cũ hoặc tích hợp các thư viện bên thứ ba.
- **Cộng đồng lớn:** Cộng đồng người dùng lớn mạnh sẽ cung cấp nhiều tài liệu, thư viện và hỗ trợ cho bạn.
- **Hỗ trợ từ Google:** Việc được Google hỗ trợ giúp các ngôn ngữ này ngày càng phát triển và hoàn thiện.

#### Câu 5:

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

- Swift, Objective-C

#### Câu 6:

Những thách thức mà Windows Phone đã phải đối mặt:

#### Áp lực cạnh tranh khốc liệt:

- **iOS và Android quá mạnh:** iOS và Android đã xây dựng được hệ sinh thái ứng dụng khổng lồ và cộng đồng người dùng trung thành.
- **Khó thay đổi thói quen người dùng:** Việc thuyết phục người dùng chuyển từ iOS hoặc Android sang Windows Phone là rất khó.

#### Mất thời gian để bắt kịp:

- **Ra mắt muộn:** Windows Phone ra mắt thị trường khá muộn so với iOS và Android, khiến nó luôn đi sau đối thủ về các tính năng và công nghệ mới.

- **Cập nhật phần mềm chậm:** Việc cập nhật phần mềm cho các thiết bị Windows Phone thường chậm hơn so với các thiết bị iOS và Android, khiến người dùng cảm thấy không được hỗ trợ tốt.

### **Thị phần quá nhỏ:**

- **Khó thu hút nhà phát triển:** Với thị phần nhỏ bé, các nhà phát triển ứng dụng thường ưu tiên phát triển ứng dụng cho iOS và Android trước tiên, vì đây là hai nền tảng có lượng người dùng đông đảo nhất. Điều này dẫn đến kho ứng dụng của Windows Phone luôn thiếu hụt so với đối thủ.

## **Câu 7:**

### **Ngôn ngữ cơ bản:**

- **HTML:** Định dạng cấu trúc của trang web.
- **CSS:** Tạo kiểu dáng và bố cục cho các phần tử trên trang.
- **JavaScript:** Thêm tính tương tác và động cho trang web.

### **Các framework JavaScript:**

- **React Native:**
  - Dựa trên thư viện React của Facebook.
  - Cho phép xây dựng ứng dụng di động native (iOS và Android) bằng JavaScript.
  - Tái sử dụng được nhiều thành phần và logic từ các ứng dụng web.
- **Angular:**
  - Framework mạnh mẽ của Google, cung cấp một kiến trúc hoàn chỉnh để xây dựng các ứng dụng web lớn.
  - Có thể sử dụng để xây dựng các ứng dụng di động hybrid thông qua các công cụ như Ionic.
- **Vue.js:**
  - Framework nhẹ và linh hoạt, dễ học.
  - Cũng có thể sử dụng để xây dựng các ứng dụng di động hybrid.

### **Các công cụ xây dựng ứng dụng hybrid:**

- **Ionic:**

- Dựa trên Angular, Cordova và các công nghệ web khác.
- Cho phép xây dựng các ứng dụng di động hybrid từ một cơ sở mã duy nhất.
- **PhoneGap:**
  - Dựa trên Cordova, cung cấp một nền tảng để xây dựng các ứng dụng di động hybrid.
- **Flutter:**
  - Sử dụng ngôn ngữ Dart, cung cấp hiệu năng cao và giao diện người dùng đẹp mắt.
  - Có thể xây dựng ứng dụng di động native cho iOS và Android, cũng như các ứng dụng web.

## Câu 8:

- Nhu cầu nhân lực: Các công ty công nghệ đều cần lập trình viên di động để phát triển các ứng dụng phục vụ người dùng trên smartphone. Đặc biệt là những nhu cầu về y tế, nhà nước như căn cước công dân, tài khoản ngân hàng, các ứng dụng hỗ trợ phạt nguội khi tham gia giao thông.
- Kỹ năng được yêu cầu:
  - Kỹ năng ngôn ngữ lập trình di động: Nắm vững Swift, Objective-C (cho iOS) và Kotlin, Java (cho Android).
  - Kinh nghiệm với các công nghệ đa nền tảng: React Native, Flutter và Xamarin đang ngày càng được ưa chuộng để tối ưu hóa chi phí và thời gian phát triển ứng dụng.
  - Kiến thức về UI/UX: Thiết kế giao diện người dùng để các ứng dụng di động yêu cầu giao diện trực quan và thân thiện.