**Bài làm**

Giải thích ngắn gọn về 8 lộ trình trong khung làm việc Flutter:

1. **Storage (Lưu trữ):**

* SqLite: Đây là một cơ sở dữ liệu nhẹ dựa trên SQLite, cho phép bạn lưu trữ dữ liệu cục bộ trong ứng dụng.
* File System: Đây là khả năng tương tác với hệ thống tệp cục bộ của thiết bị, cho phép bạn đọc/ghi tệp.
* Shared Preferences: Thư viện cho phép lưu trữ các cặp giá trị khóa/giá trị trong bộ nhớ cục bộ dưới dạng các thiết lập ứng dụng.

1. **3rd Party Libraries (Thư viện bên thứ ba):**

Cung cấp một danh sách các thư viện bên thứ ba được sử dụng trong dự án Flutter, bao gồm HTTP request, quản lý dependency, tải hình ảnh, xử lý luồng dữ liệu (streams), quản lý trạng thái ứng dụng, tích hợp trình duyệt web và nhiều tính năng khác.

* http: Thư viện HTTP cho Flutter, cho phép bạn tương tác với các API và dịch vụ web.
* dio: Một thư viện HTTP mạnh mẽ, hỗ trợ nhiều tính năng như tải tệp, xử lý request đồng thời và hơn thế nữa.
* get\_it: Thư viện dependency injection giúp quản lý và cung cấp các đối tượng trong toàn bộ ứng dụng.
* cached\_network\_image: Cho phép tải hình ảnh từ mạng và lưu trữ trong bộ nhớ cache để tối ưu hiệu suất.
* font\_awesome\_flutter: Thư viện hỗ trợ sử dụng biểu tượng từ Font Awesome trong ứng dụng của bạn.
* rxdart: Thư viện Reactive Extensions (Rx) cho Dart, hỗ trợ xử lý dữ liệu dạng luồng (stream) và xử lý bất đồng bộ.
* SQFLite: Một thư viện SQLite đơn giản cho Flutter, giúp quản lý cơ sở dữ liệu cục bộ trong ứng dụng của bạn.
* bloc\_pattern: Một mô hình quản lý trạng thái ứng dụng (state management) dựa trên BLoC (Business Logic Component), giúp tách biệt logic và giao diện.
* flutter\_webview\_plugin: Cho phép tích hợp trình duyệt web trong ứng dụng của bạn thông qua các webview.

Các thư viện này cung cấp các giải pháp tiện ích để phát triển ứng dụng Flutter của bạn, giúp bạn tập trung vào chức năng cốt lõi mà không phải xây dựng lại các phần tổng quát đã có sẵn.

1. **Behavior Components (Các thành phần hành vi):**

* Media Playback: Hỗ trợ phát các phương tiện như âm thanh, video trong ứng dụng.
* Preference: Cho phép quản lý các thiết lập và tùy chọn ứng dụng.
* Permission: Hỗ trợ yêu cầu và quản lý quyền truy cập của ứng dụng đối với các tài nguyên như máy ảnh, vị trí, v.v.
* Local Notification: Cho phép gửi thông báo từ ứng dụng trên thiết bị.
* Push Notification: Cho phép gửi thông báo từ máy chủ đến ứng dụng.
* Download Manager: Quản lý việc tải xuống tệp trong ứng dụng của bạn.

1. **State Management (Quản lý trạng thái):**

* setState: Cách đơn giản nhất để quản lý trạng thái trong một StatefulWidget, thường được sử dụng cho các ứng dụng nhỏ.
* Provider: Thư viện state management của Flutter, cho phép chia sẻ trạng thái giữa các thành phần của ứng dụng.
* BloC: Mô hình quản lý trạng thái ứng dụng dựa trên luồng dữ liệu (streams), giúp tách biệt logic và giao diện.
* Redux: Thư viện state management dựa trên kiến thức từ kiến trúc Flux, được sử dụng để quản lý trạng thái ứng dụng phức tạp.
* MobX: Thư viện state management sử dụng kỹ thuật reactive programming, giúp tối ưu việc cập nhật trạng thái.

1. **Quality Assurance (QA) - Đảm bảo chất lượng:**

* Firebase App Distribution: Dịch vụ cung cấp khả năng phân phát ứng dụng dưới dạng phiên bản thử nghiệm cho các người tham gia kiểm thử trước khi phát hành chính thức.
* Crashlytics: Cung cấp theo dõi và báo cáo sự cố, lỗi, crash của ứng dụng để giúp nhóm phát triển nhanh chóng khắc phục.
* Analytics: Cho phép thu thập dữ liệu và phân tích hành vi người dùng để hiểu và cải thiện trải nghiệm người dùng.

1. **Version Control - Quản lý phiên bản:**

* Git: Hệ thống quản lý phiên bản phân tán, cho phép theo dõi thay đổi trong mã nguồn và làm việc song song với nhóm phát triển.
* Github, GitLab, Bitbucket: Nền tảng dịch vụ lưu trữ mã nguồn Git trực tuyến, giúp quản lý mã nguồn, hợp tác và xem xét mã nguồn.

1. **Firebase - Nền tảng phát triển ứng dụng:**

* Firebase Auth: Cung cấp quản lý xác thực người dùng thông qua email, số điện thoại, mạng xã hội.
* Firebase Database: Cơ sở dữ liệu thời gian thực lưu trữ dữ liệu và đồng bộ trạng thái ứng dụng giữa các thiết bị.
* Firebase Messaging: Hỗ trợ gửi thông báo đẩy và tin nhắn trong ứng dụng.
* Firebase Storage: Dịch vụ lưu trữ tệp, hình ảnh, video và dữ liệu liên quan.

1. **Native Integration - Tích hợp nền tảng gốc:**

* Xcode, Swift/Objective-C: Dùng cho phát triển ứng dụng iOS, với Xcode là IDE và Swift/Objective-C là ngôn ngữ lập trình.
* App Store, Apple Certification: Dành cho việc phát hành ứng dụng trên App Store của Apple, cần tuân thủ các quy định và đạt các yêu cầu kiểm tra chất lượng.
* Android Studio, Java/Kotlin: IDE và ngôn ngữ lập trình cho phát triển ứng dụng Android.
* Google Play Store, App Signing: Dành cho việc phát hành ứng dụng trên Google Play Store, bao gồm việc ký tên ứng dụng và tuân thủ quy định của Google.