### **CHƯƠNG TRÌNH ĐÀO TẠO THỰC TẬP SINH FLUTTER**

### **Tuần 1: Làm quen với Flutter và Dart**

**Ngày 1: Giới thiệu về Flutter và Dart**

* Giới thiệu về Flutter, Dart và lý do Flutter trở thành lựa chọn phổ biến cho phát triển ứng dụng di động.
* Cài đặt Flutter SDK, Dart SDK, Android Studio và các công cụ phát triển cần thiết.
* Tạo dự án Flutter đầu tiên và chạy ứng dụng mẫu.

**Ngày 2: Kiến thức cơ bản về Dart**

* Giới thiệu cú pháp của Dart: biến, kiểu dữ liệu, hàm, toán tử.
* Cấu trúc điều khiển: if, else, switch, for, while, do-while.
* Luyện tập các bài tập cơ bản bằng Dart.

**Ngày 3: Kiến thức nâng cao Dart**

* Làm quen với các đối tượng trong Dart: class, đối tượng, constructors, getters/setters.
* Làm việc với collection trong Dart: List, Set, Map.
* Các khái niệm về asynchronous programming: Future, async, await.

**Ngày 4: Giới thiệu về Flutter Widgets**

* Làm quen với Widget trong Flutter: StatelessWidget, StatefulWidget.
* Xây dựng giao diện đơn giản với Text, Container, Row, Column.
* Các phương thức layout cơ bản: padding, margin, alignment.

**Ngày 5: Các widget cơ bản**

* Tìm hiểu các widget cơ bản trong Flutter như Text, Image, Icon, Button.
* Học cách sử dụng các widget bố cục như Column, Row, Stack.
* Thực hành tạo giao diện đơn giản với các widget cơ bản.

**Ngày 6: Xử lý sự kiện và tương tác người dùng**

* Xử lý sự kiện người dùng: GestureDetector, InkWell.
* Làm việc với các widget form: TextField, Form, TextFormField.
* Xử lý các sự kiện nhập liệu và validation cơ bản.

**Ngày 7: Ôn tập và thực hành**

* Ôn tập lại các kiến thức đã học trong tuần.
* Thực hành xây dựng một ứng dụng nhỏ (ví dụ: ứng dụng nhập liệu đơn giản).

### **Tuần 2: Làm việc với Layout và Navigation**

**Ngày 8: Layout nâng cao**

* Tìm hiểu các widget bố cục nâng cao: ListView, GridView, SingleChildScrollView.
* Làm quen với Expanded, Flex, Spacer.
* Thực hành tạo giao diện ứng dụng sử dụng ListView và GridView.

**Ngày 9: Navigation trong Flutter**

* Giới thiệu về navigation trong Flutter.
* Sử dụng Navigator để chuyển đổi giữa các màn hình.
* Thực hành xây dựng ứng dụng với nhiều màn hình (screen navigation).

**Ngày 10: Điều hướng với routes và named routes**

* Học cách cấu hình routes trong ứng dụng Flutter.
* Làm việc với named routes và truyền tham số giữa các màn hình.
* Thực hành xây dựng một ứng dụng nhỏ sử dụng named routes.

**Ngày 11: Các widget nâng cao**

* Tìm hiểu về Drawer, BottomNavigationBar.
* Làm quen với TabBar và TabView.
* Thực hành xây dựng giao diện có Drawer và BottomNavigationBar.

**Ngày 12: Quản lý trạng thái (State Management)**

* Giới thiệu về quản lý trạng thái trong Flutter.
* Tìm hiểu về setState, InheritedWidget.
* Thực hành sử dụng setState để thay đổi trạng thái của ứng dụng.

**Ngày 13: Các cách quản lý trạng thái**

* Tìm hiểu về Provider, Riverpod, và các thư viện quản lý trạng thái khác.
* Cài đặt và cấu hình Provider trong dự án Flutter.
* Thực hành xây dựng ứng dụng có quản lý trạng thái sử dụng Provider.

**Ngày 14: Ôn tập và thực hành**

* Ôn lại các kiến thức về navigation và quản lý trạng thái.
* Thực hành xây dựng ứng dụng đa màn hình với khả năng quản lý trạng thái.

### **Tuần 3: Làm việc với API và Async Programming**

**Ngày 15: Giới thiệu về HTTP và làm việc với API**

* Giới thiệu về HTTP trong Flutter.
* Cài đặt thư viện http và làm việc với API RESTful.
* Tìm hiểu thư viện dio.
* Thực hành gọi API và xử lý dữ liệu trả về.

**Ngày 16: Làm việc với dữ liệu JSON**

* Làm quen với cách làm việc với dữ liệu JSON trong Flutter.
* Sử dụng json.decode và json.encode để chuyển đổi dữ liệu.
* Thực hành tải dữ liệu JSON từ API và hiển thị trong giao diện.

**Ngày 17: Quản lý trạng thái khi gọi API**

* Xử lý trạng thái tải dữ liệu (loading state, error state).
* Sử dụng FutureBuilder và StreamBuilder để hiển thị dữ liệu bất đồng bộ.
* Thực hành xây dựng ứng dụng gọi API và hiển thị kết quả.

**Ngày 18: Xử lý lỗi và debug trong Flutter**

* Các phương pháp xử lý lỗi trong Flutter.
* Sử dụng try-catch để xử lý lỗi khi gọi API.
* Cách debug ứng dụng Flutter với Flutter DevTools.

**Ngày 19: Lưu trữ dữ liệu cục bộ**

* Làm quen với SharedPreferences để lưu trữ dữ liệu đơn giản.
* Sử dụng SQLite hoặc Hive để lưu trữ dữ liệu phức tạp.
* Thực hành lưu trữ và truy xuất dữ liệu trong ứng dụng.

**Ngày 20: Cải thiện hiệu suất ứng dụng - Quản lý phiên bản và Git**

* Các kỹ thuật tối ưu hiệu suất trong Flutter.
* Giới thiệu về lazy loading, image caching, và các kỹ thuật khác.
* Thực hành tối ưu hóa ứng dụng bằng cách sử dụng các phương pháp trên.
* Giới thiệu về Git và GitHub.
* Quản lý mã nguồn với Git.
* Thực hành các lệnh Git cơ bản (commit, push, pull, branch).

**Ngày 21: Ôn tập và thực hành**

* Ôn lại kiến thức về API, Async programming, và lưu trữ dữ liệu.
* Thực hành xây dựng một ứng dụng sử dụng API và lưu trữ dữ liệu.

### **Tuần 4: Tìm hiểu kiến trúc BLoC và xây dựng ứng dụng.**

**Ngày 22-26: Tìm hiểu kiến trúc BLoc**

* Tìm hiểu kiến trúc BLoc.
* Thực hành xây dựng một ứng dụng áp dụng kiến trúc BLoc.
* Hoàn thành ứng dụng của riêng thực tập sinh (dự án nhỏ).
* Code review và feedback.
* Sửa lỗi và cải thiện ứng dụng.
* Thực hiện đánh giá cuối kỳ.
* Chỉnh sửa, tối ưu mã nguồn.
* Chuẩn bị báo cáo và demo kết quả.