## Tạo ứng dụng sử dụng GetX

**Đề bài**

Sử dụng getX để tạo form thông tin và hiển thị lên màn hình.

**Mục tiêu**

Sau bài học này các bạn có thể đi sâu vào các khái niệm lập trình Dart hơn và học cách sử dụng các gói Flutter mã nguồn mở. GetX là một giải pháp cực kỳ nhẹ và mạnh mẽ cho Flutter. Nó kết hợp quản lý nhà nước hiệu suất cao, tiêm phụ thuộc thông minh và quản lý tuyến đường một cách nhanh chóng và thực tế.

**Những gì bạn sẽ học**

* Cách kết hợp các thư viện mã nguồn mở vào dự án của bạn bằng Gói Flutter.
* Cách quản lý Widget tùy chỉnh trong Flutter
* Cách sử dụng TextField Widget để lấy thông tin đầu vào của người dùng.
* Cách sử dụng TextEditingController, Divider
* Cách sử dụng lớp trong Dart
* Cách sử dụng các hàm Dart.
* Cách đọc và sử dụng Tài liệu Flutter.

**Thực hành :**

1. **Tạo project**

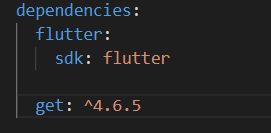
Sử dụng lệnh **flutter create** để tạo ứng dụng mới:

**flutter create getx**

**cd getx**

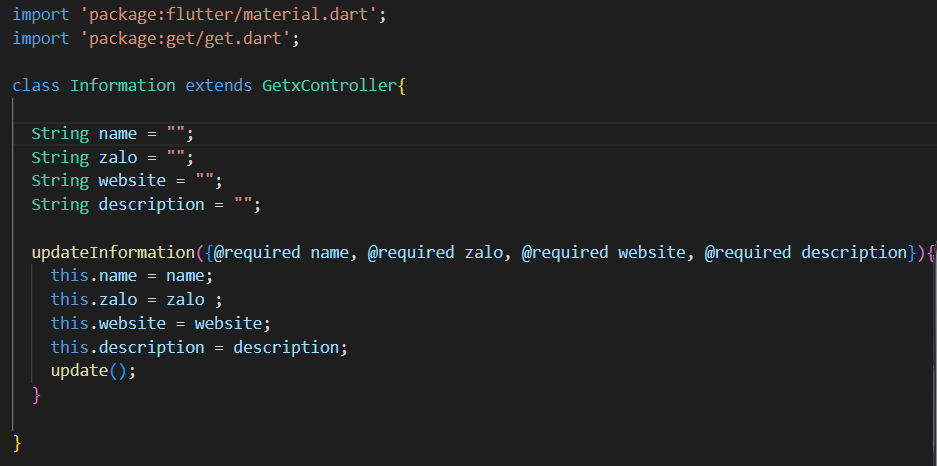
1. **Cài đặt thư viện cần thiết**

Mở file pubspec.yaml và thực hiện thêm thư viện **get**.



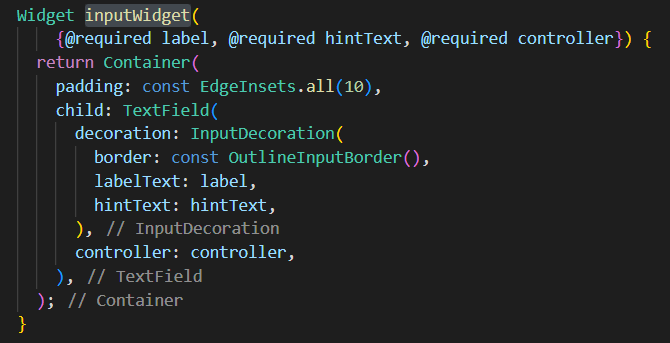
1. **Tạo lớp Information**

* Tạo lớp Information extends GetXController gồm các trường cần thiết.
* Hàm updateInformation để cập nhật giá trị cho các trường thông tin

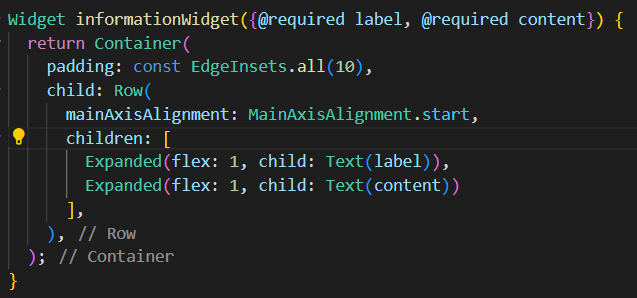
****

1. **Tạo các String và Widget cần thiết**

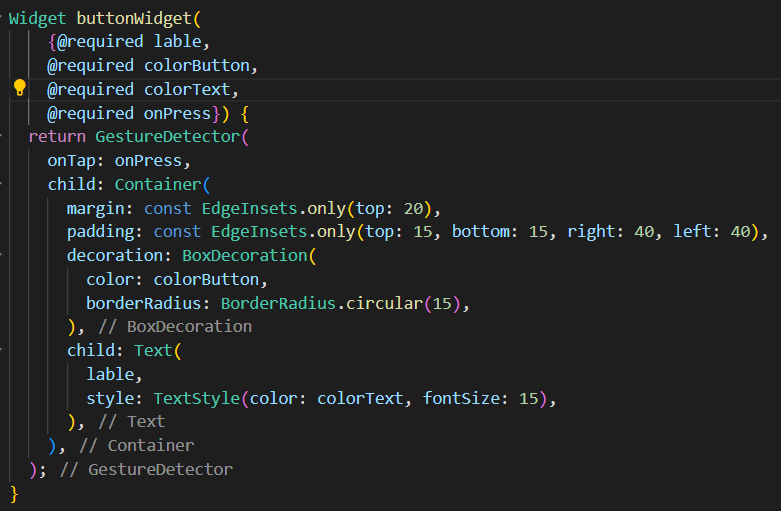
* Tạo inputWidget

****

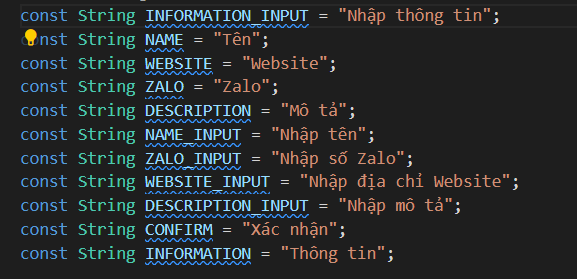
* Tạo informationWidget



* Tạo buttonWidget

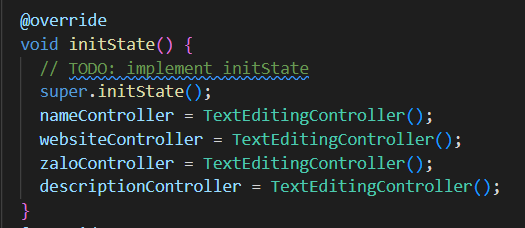


* Tạo một file string để quản lý chuỗi trong ứng dụng

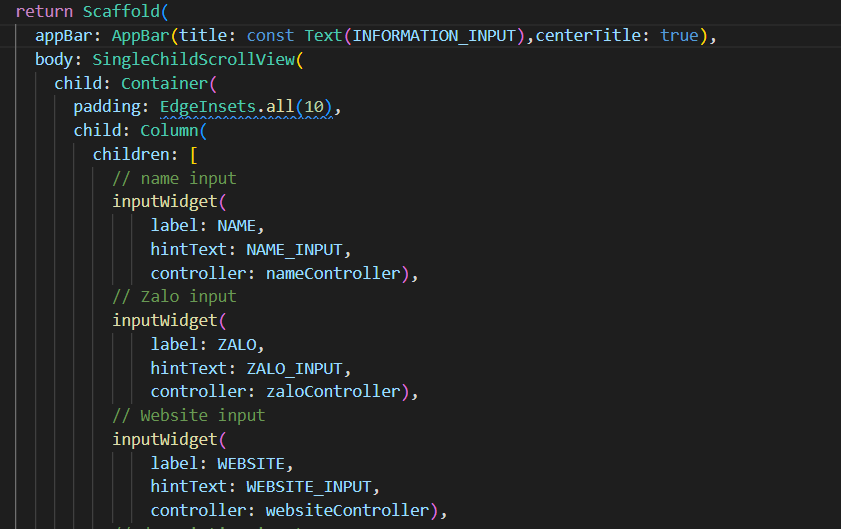


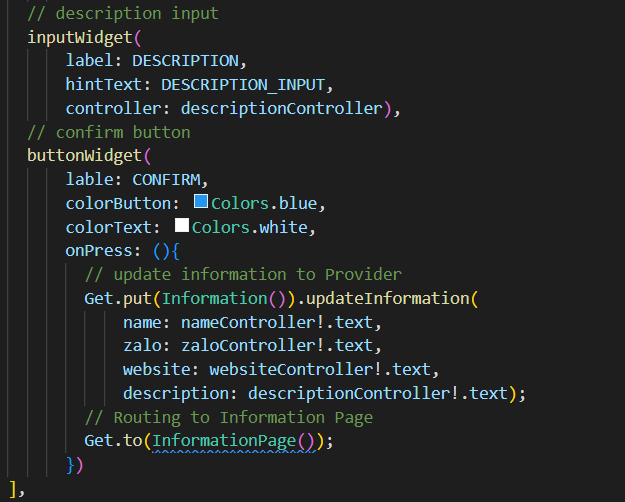
1. **Tạo trang input thông tin**

* Khởi tạo các biến TextEditingController để hỗ trợ nhập thông tin

****

* Tạo form nhập thông tin

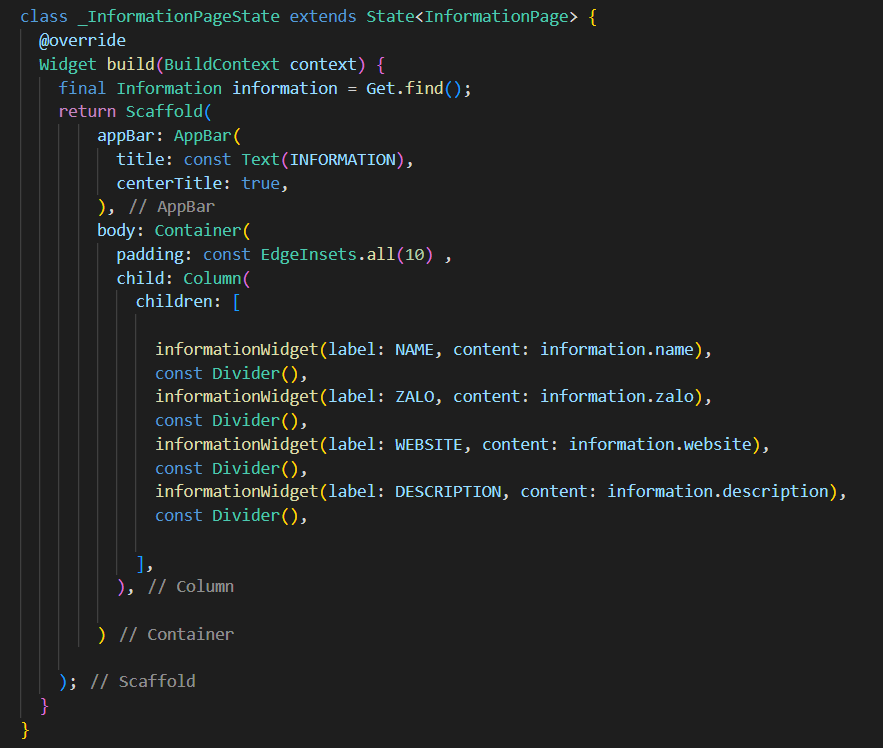




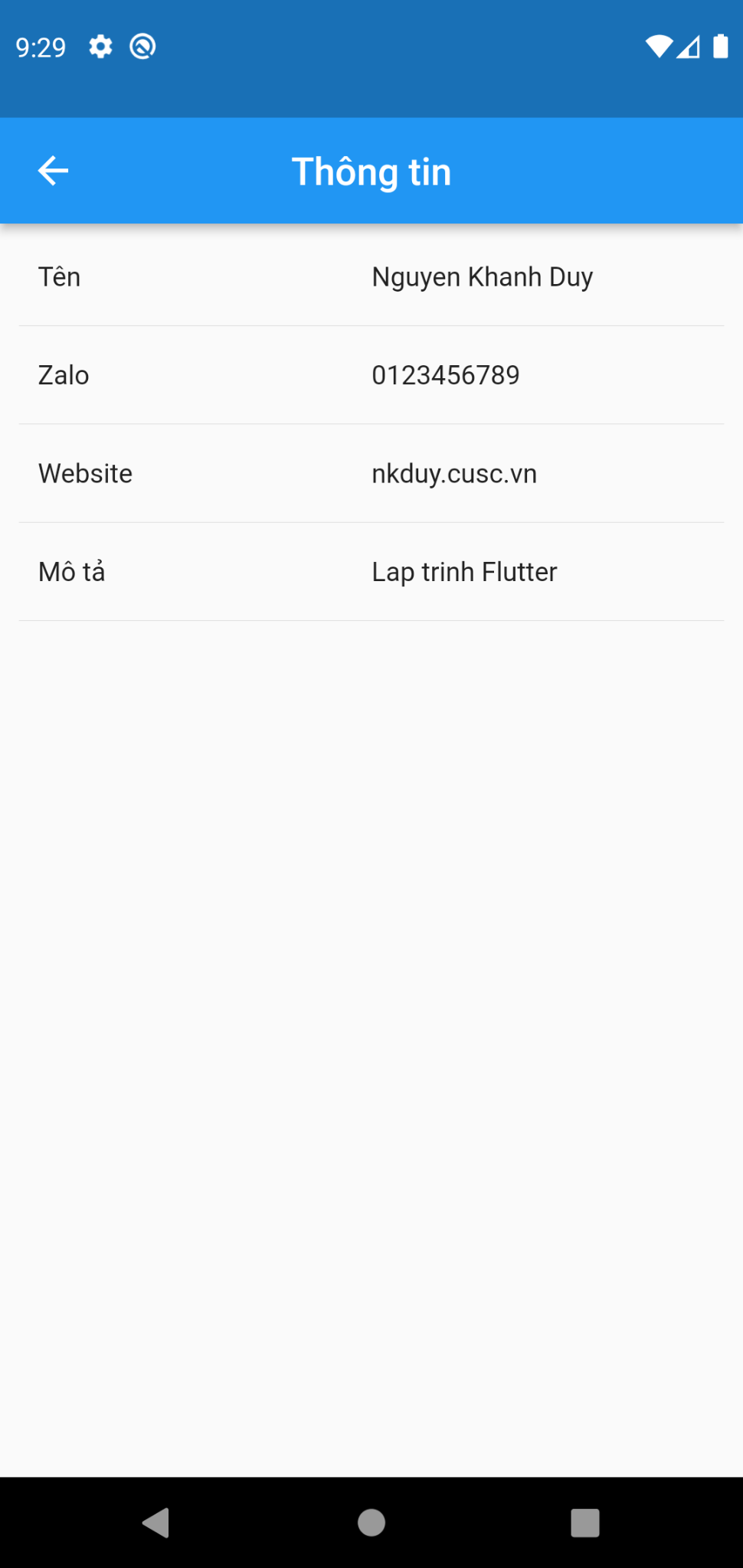
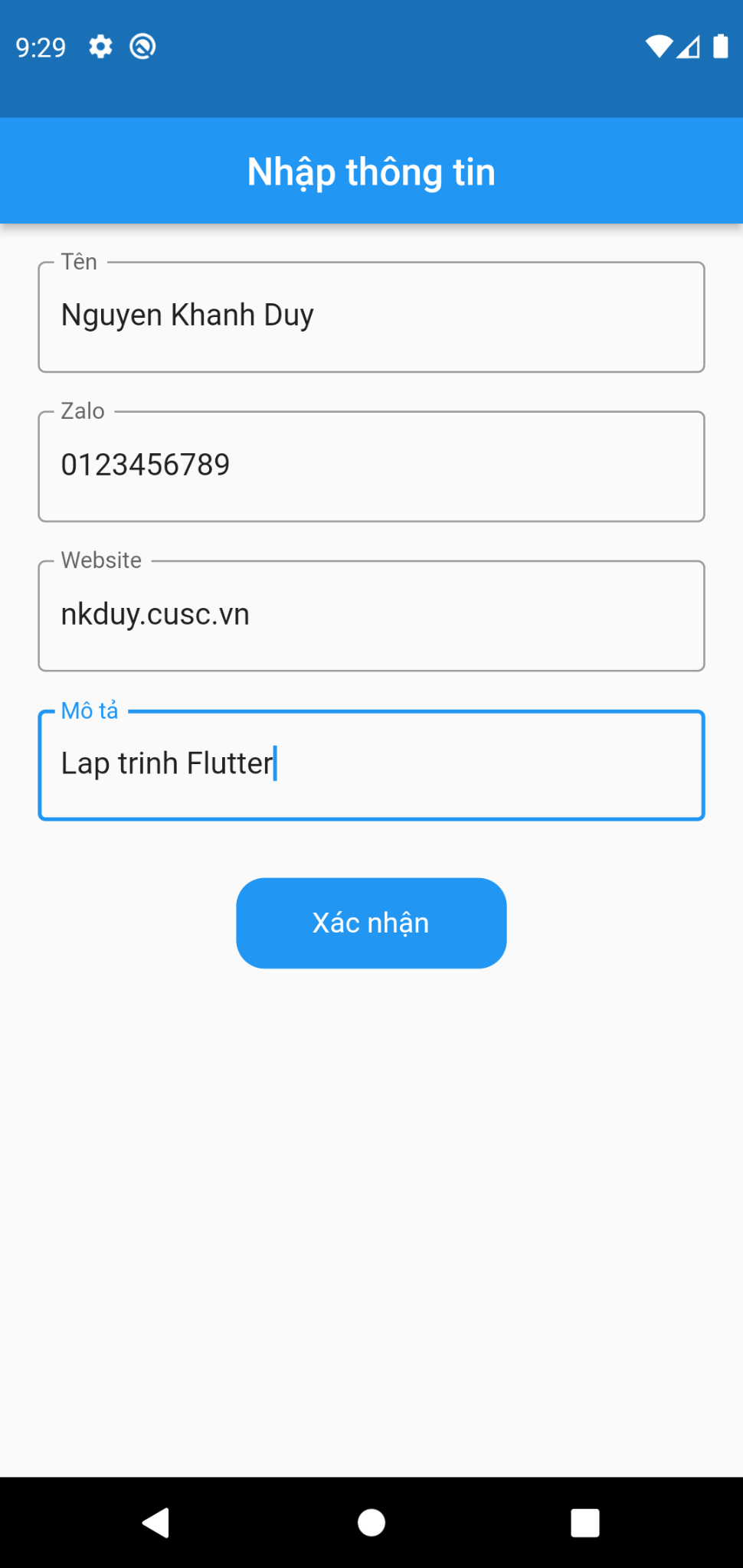
* Sử dụng Get.put để cập nhật các giá trị đã nhập vào model
* Sử dụng Get.to để chuyển sang màn hình hiển thị thông tin.

1. **Tạo trang output thông tin**

* Sử dụng Get.find để lấy thông tin đã nhập
* Sử dụng widget informationWidget để hiển thị các trường thông tin



**Kết quả**

****