**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**BÁO CÁO**

**LẬP TRÌNH ĐA NỀN TẢNG**

**CHỦ ĐỀ: TEXT VÀ STYLING TRONG FLUTTER**

**Người hướng dẫn: TS. Nguyễn Duy Nhật Viễn**

**Sinh viên thực hiện:**

**01. Nguyễn Thị Uyên Phương** **Lớp: 22KTMT1** **MSSV:**  106220231

**02. Lê Thị Hải Yến** **Lớp: 22KTMT2** **MSSV:**  106220278

**Đà Nẵng, 2025**

**THUYẾT MINH**

**BÁO CÁO**

**LẬP TRÌNH ĐA NỀN TẢNG**

**CHỦ ĐỀ: TEXT VÀ STYLING TRONG FLUTTER**

BẢNG PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | NHIỆM VỤ | KHỐI LƯỢNG |
| 01 | NGUYỄN THỊ UYÊN PHƯƠNG | Tìm hiểu kiến thức Text và Styling, soạn nội dung word | 50% |
| 02 | LÊ THỊ HẢI YẾN | Tìm hiểu kiến thức text và Styling, làm slide báo cáo | 50% |

 Link code github: <http://github/>...

**Mục lục**

…

**NỘI DUNG**

1. Giới thiệu chung:

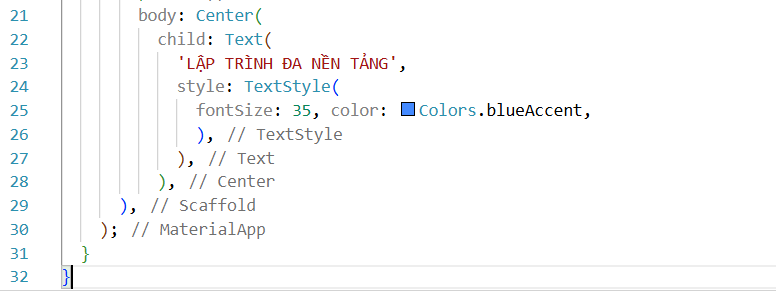
Trong Flutter, Text là một trong những widget cơ bản và quan trọng nhất, được dùng để hiển thị chuỗi ký tự trên giao diện người dùng. Mọi văn bản trong ứng dụng Flutter đều được hiển thị thông qua widget Text, đi kèm với thuộc tính TextStyle để định dạng kiểu chữ, kích thước, màu sắc, và nhiều yếu tố khác. Việc hiểu rõ cách sử dụng Text và TextStyle giúp lập trình viên tạo ra giao diện nhất quán và dễ đọc trong toàn bộ ứng dụng.

1. TextStyle và các thuộc tính định dạng:

Widget Text có thể đi kèm với đối tượng TextStyle để mô tả phong cách hiển thị văn bản. TextStyle cung cấp nhiều thuộc tính quan trọng, bao gồm:

* fontSize: xác định kích thước của chữ (đơn vị là pixel logic).  
  Ví dụ: fontSize: 20 sẽ làm văn bản lớn hơn mặc định.
* color: quy định màu chữ, có thể sử dụng từ thư viện Colors, ví dụ color: Colors.blue.

+ Code demo fontSize và Color:



+ Kết quả hiển thị:

A white background with blue text

AI-generated content may be incorrect.

* fontWeight: điều chỉnh độ đậm nhạt của chữ. Có các giá trị như FontWeight.bold, FontWeight.w500 (medium), FontWeight.normal,…
* fontStyle: xác định kiểu chữ thường hoặc nghiêng, thường dùng FontStyle.normal hoặc FontStyle.italic.

+ Code demo fontWeight và fontStyle:



+ Kết quả hiển thị:

A white background with black text

AI-generated content may be incorrect.

* letterSpacing: khoảng cách giữa các ký tự.
* wordSpacing: khoảng cách giữa các từ.
* decoration: cho phép thêm hiệu ứng gạch chân, gạch ngang hoặc gạch trên. Ví dụ: decoration: TextDecoration.underline.

+ Code demo letterSpacing, wordSpacing và decoration:

A computer screen shot of a code

AI-generated content may be incorrect.

+ Kết quả hiển thị:

A screenshot of a computer

AI-generated content may be incorrect.

* fontFamily: quy định kiểu chữ cụ thể l(Roboto, Times New Roman, v.v.).

Việc kết hợp các thuộc tính này giúp văn bản hiển thị hấp dẫn và dễ đọc hơn, phù hợp với phong cách thiết kế của toàn ứng dụng.

1. Text Overflow và Text Alignment:

Khi văn bản quá dài so với không gian hiển thị, Flutter cho phép kiểm soát hành vi tràn chữ thông qua thuộc tính overflow:

* TextOverflow.clip: cắt bớt phần chữ dư.
* TextOverflow.ellipsis: thêm dấu “…” khi chữ vượt quá giới hạn.
* TextOverflow.fade: làm mờ dần phần chữ bị tràn.

Bên cạnh đó, thuộc tính textAlign giúp căn chỉnh văn bản trong vùng hiển thị, với các giá trị như:

* TextAlign.left, TextAlign.right: căn trái hoặc phải.
* TextAlign.center: căn giữa.
* TextAlign.justify: căn đều hai bên lề.

Những thuộc tính này đặc biệt quan trọng khi thiết kế giao diện cho các thiết bị có kích thước màn hình khác nhau.

* Code demo:

A screenshot of a computer

AI-generated content may be incorrect.

* Kết quả hiển thị:

A screenshot of a text message

AI-generated content may be incorrect.

1. RichText và TextSpan:

Ngoài widget Text, Flutter còn cung cấp RichText – một widget mạnh mẽ cho phép hiển thị nhiều định dạng trong cùng một đoạn văn. RichText hoạt động dựa trên **TextSpan**, là các đoạn văn bản con có thể được định dạng độc lập.   
**Ví dụ:**

A screen shot of a computer program

AI-generated content may be incorrect.

* Kết quả:

A screenshot of a computer

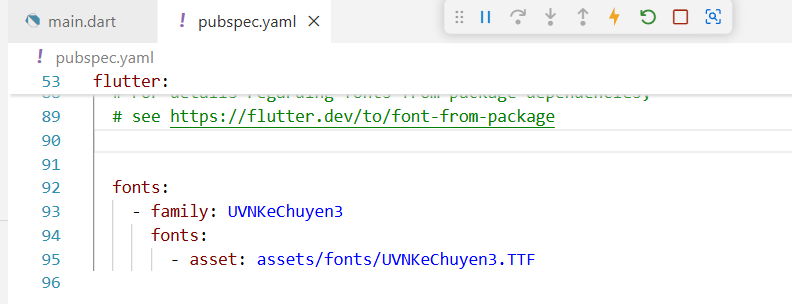
AI-generated content may be incorrect.

Điều này giúp bạn tạo tiêu đề hoặc đoạn giới thiệu có nhiều kiểu chữ trong cùng một dòng, rất hữu ích trong các ứng dụng cần nhấn mạnh một phần nội dung nhất định.

1. Custom Fonts và Theme

Flutter hỗ trợ sử dụng font tùy chỉnh (custom fonts) để đồng bộ phong cách thiết kế thương hiệu. Để thêm font riêng, lập trình viên cần:

* Tạo thư mục assets/fonts và thêm các file .ttf hoặc .otf vào đó.
* Khai báo đường dẫn font trong file pubspec.yaml.



* Sử dụng trong TextStyle:

A screen shot of a computer

AI-generated content may be incorrect.

* Kết quả:

A white background with blue text

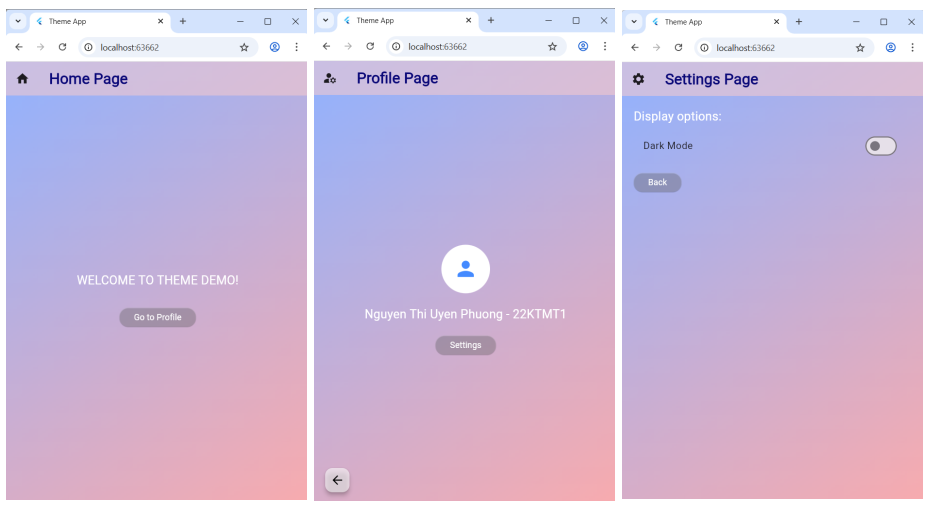
AI-generated content may be incorrect.

Ngoài ra, Flutter cho phép áp dụng ThemeData để định nghĩa phong cách chữ tổng thể cho toàn ứng dụng. Bằng cách chỉnh trong MaterialApp, bạn có thể đảm bảo mọi Text trong app tuân theo cùng một quy tắc kiểu dáng, giúp giao diện đồng bộ và dễ bảo trì.

A screenshot of a computer program

AI-generated content may be incorrect.

* Kết quả:



1. Tổng kết:

Phần Text và Styling là nền tảng quan trọng trong thiết kế giao diện Flutter. Việc hiểu rõ Text, TextStyle, RichText, TextSpan, cùng khả năng tùy chỉnh font và theme giúp lập trình viên tạo nên trải nghiệm người dùng hấp dẫn, rõ ràng và mang tính thương hiệu. Mặc dù chỉ là yếu tố hiển thị, nhưng nó góp phần lớn vào tính chuyên nghiệp và thẩm mỹ của ứng dụng.

**TÀI LIỆU THAM KHẢO**

[1]. https://docs.flutter.dev/