#5 Setup the Application Theme

Introduction

Setting up an application theme is an important step in ensuring a consistent and polished appearance throughout the app. By defining colors and typography in separate files, we can easily make changes to these elements without having to search for each instance of their use. This will save time and effort in the long run, and result in a better user experience overall.

Step-by-step guide

Here is a step-by-step guide to setting up the application theme, overall we need to create two new files in the `lib/themes` folder of our project.

Colors

Create a new file called `colors.dart` inside the `lib/themes/` folder.

This file defines all the colors we will use in the application. This is an important step, as it allows us to maintain consistency throughout the application and ensure that all colors are used uniformly.

Populate the `lib/themes/colors.dart` file:

```
import 'package:flutter/widgets.dart';

class MyColors {
  static Color primary = const Color(0xff4951f5);
  static Color black = const Color(0xff202033);
}
```

Typography

Create a new file called `typography.dart` inside the `lib/themes/` folder.

This file will contain all the typography-related information, such as font sizes, weights, and styles. By defining all of these in one place, we can easily make changes to the typography throughout the application without having to search for each instance of text individually.

Populate the `lib/themes/typography.dart` file:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

import 'colors.dart';

class MyTypography {
  static TextStyle title = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w700,
    fontSize: 24,
  );
  // titleMedium
  static TextStyle titleMedium = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w700,
    fontSize: 22,
  );
  // titleSmall
  static TextStyle titleSmall = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w700,
    fontSize: 18,
  );

  // body
  static TextStyle bodyLarge = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w500,
    fontSize: 18,
  );
  // body
  static TextStyle body = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w400,
    fontSize: 16,
  );
  // bodySmall
  static TextStyle bodySmall = GoogleFonts.nunitoSans(
    color: MyColors.black,
    fontWeight: FontWeight.w400,
    fontSize: 13,
  );
}
```

Conclusion

In conclusion, setting up an application theme is crucial for ensuring consistency and a polished appearance throughout the app. By defining colors and typography in separate files, we can easily make changes to these elements without having to search for each instance of their use. This will save time and effort in the long run, and result in a better user experience overall.