

Steps to add images , icons and fonts

1. Adding Images

Step 1: Organize Your Images

- Create a folder in your project root (next to your lib folder) called assets.
- Inside the assets folder add your image files there (e.g., welcome.png).

Step 2: Configure pubspec.yaml

Open your pubspec.yaml file and add the assets path under the flutter: section:

```
flutter:  
  assets:  
    - assets/welcome.png
```

Note: If you want to include all images in the folder, you can use:

```
flutter:  
  assets:  
    - assets/
```

Step 3: Use Images in Your Code

To display the image in your widget, use the Image.asset widget:

```
Image.asset('assets/welcome.png', height: 150),
```

This tells Flutter to load the image from the provided path inside the assets folder. Make sure the path matches exactly what you have in your pubspec.yaml.

2. Using Icons

Flutter comes with a rich set of built-in Material Icons. You do not need to add extra files for these icons.

Example:

```
Icon(Icons.login, size: 20, color: Colors.blue),
```

3. Adding Custom Fonts

Step 1: Organize Your Fonts

- Create a folder called assets/fonts in your project.
- Place your font files (e.g., Roboto-Regular.ttf, Roboto-Bold.ttf) inside this folder.

Step 2: Configure pubspec.yaml

In your pubspec.yaml, declare the fonts under the flutter: section:

flutter:

 fonts:

 - family: Roboto

 fonts:

 - asset: assets/fonts/Roboto-Regular.ttf

 - asset: assets/fonts/Roboto-Bold.ttf

 weight: 700

- family: – The name you will refer to in your code (e.g., 'Roboto').
- asset: – The path to the font file.
- weight: – (Optional) Specify different weights for bold or other variants.

Step 3: Use Custom Fonts

After configuring, use your custom font in your text styles:

Text(

 'Login',

 style: TextStyle(

 fontFamily: 'Roboto',

 fontSize: 28,

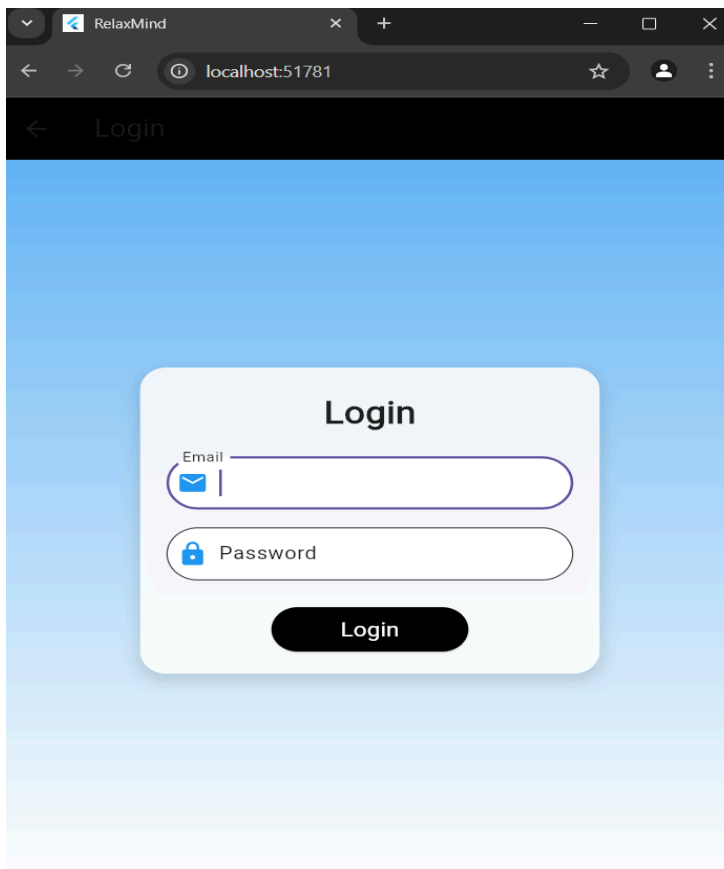
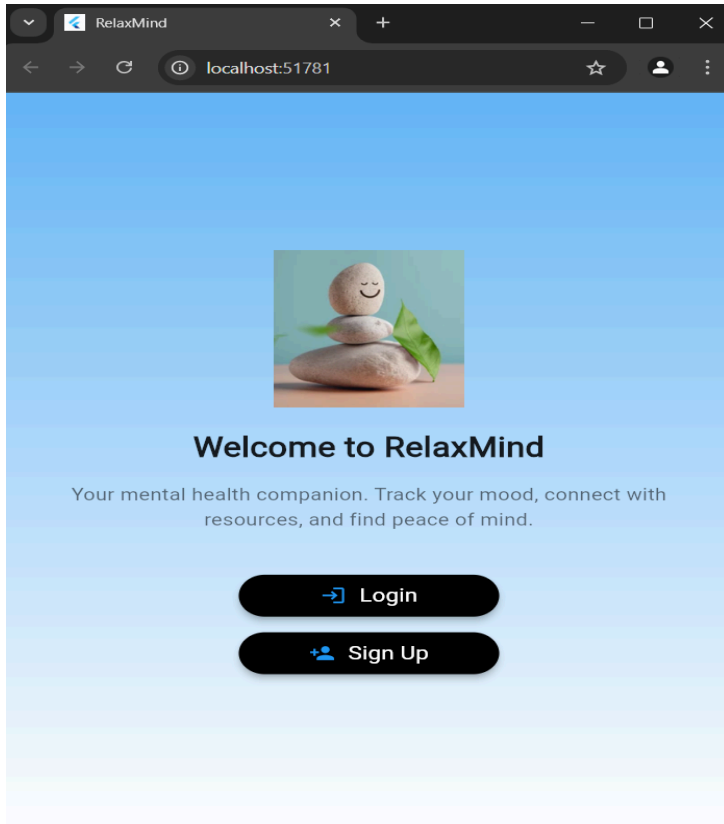
 fontWeight: FontWeight.bold,

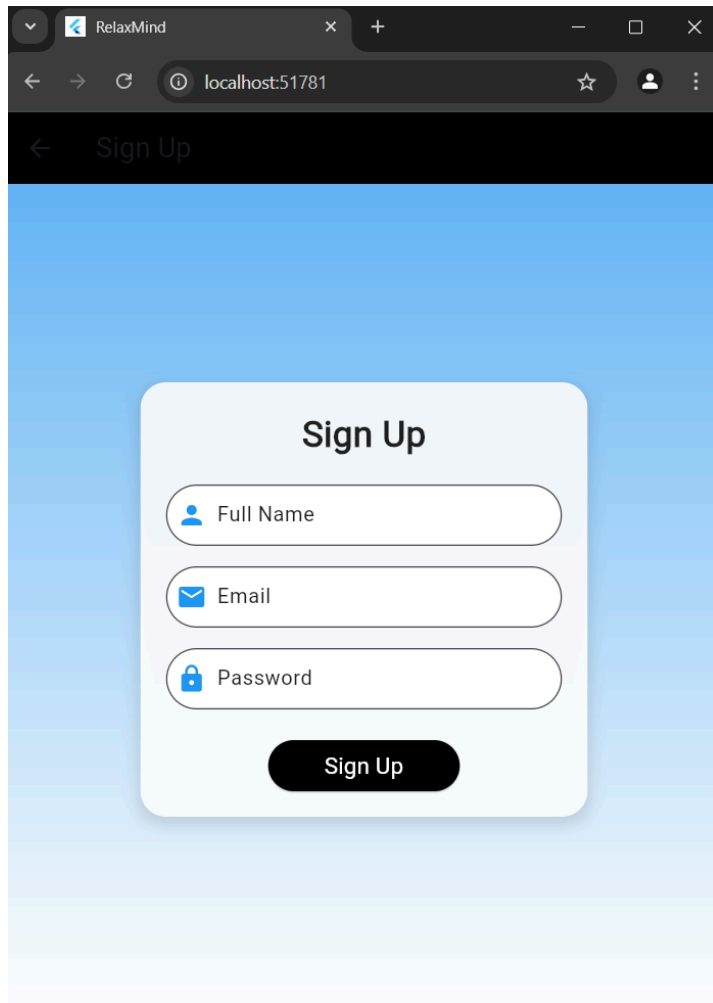
 color: Colors.black87,

),

),

This tells Flutter to use the Roboto font for the text. If you specify a fontWeight that matches one of your defined variants (like bold for weight 700), it will automatically select the correct font file.





Conclusion: Adding **images, icons, and custom fonts** in Flutter enhances the visual appeal and user experience of an app. By properly organizing assets, configuring the `pubspec.yaml` file, and using widgets like `Image.asset`, `Icon`, and `TextStyle`, developers can efficiently incorporate these elements into their UI. Images help in creating engaging layouts, icons provide intuitive navigation, and custom fonts establish a unique brand identity.