

## EXPERIMENT 3

**Aim:** To include icons, images, fonts in Flutter app

### Theory:

#### 1. Integration of Icons, Images, and Fonts in Flutter:

In Flutter, incorporating icons, images, and custom fonts into the application enhances the visual appeal and user experience. Icons from the Material Design or Cupertino icon sets are easily integrated using the `Icon` widget. The `Image` widget allows developers to include graphics and pictures, supporting various formats like JPEG, PNG, and GIF. Custom fonts, chosen for a distinctive aesthetic, can be integrated into Flutter applications to maintain a consistent typographic style. These assets are often organized within the `assets` directory of the project, and their inclusion is defined in the `pubspec.yaml` file under the `flutter` section.

#### 2. Configuring Assets in `pubspec.yaml`:

The `pubspec.yaml` file serves as a central configuration file in Flutter projects. To integrate icons, images, and fonts, the `pubspec.yaml` file includes specific entries under the `flutter` section. The `assets` entry defines the location of image files, icon sets, and custom fonts. For images and icons, developers use the `Image.asset` and `Icon` widgets, respectively, providing the path or reference to the asset in the `assets` directory. Custom fonts are specified in the `pubspec.yaml` file and referenced using the `TextStyle` class in the Flutter code.

#### 3. Application in Flutter Code:

Once assets are configured, they can be seamlessly incorporated into Flutter code. Icons are used with the `Icon` widget, images with the `Image` widget, and fonts by styling text with the desired font family. For example, in the provided code snippet, the `Icon` widget is utilized to include a search icon in the `AppBar`, the `Image.asset` widget displays book cover images in a `ListView`, and custom font styling is applied to text using the `TextStyle` class. This systematic approach to integrating icons, images,

and fonts contributes to a visually appealing and cohesive user interface in Flutter applications.

### Code:

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

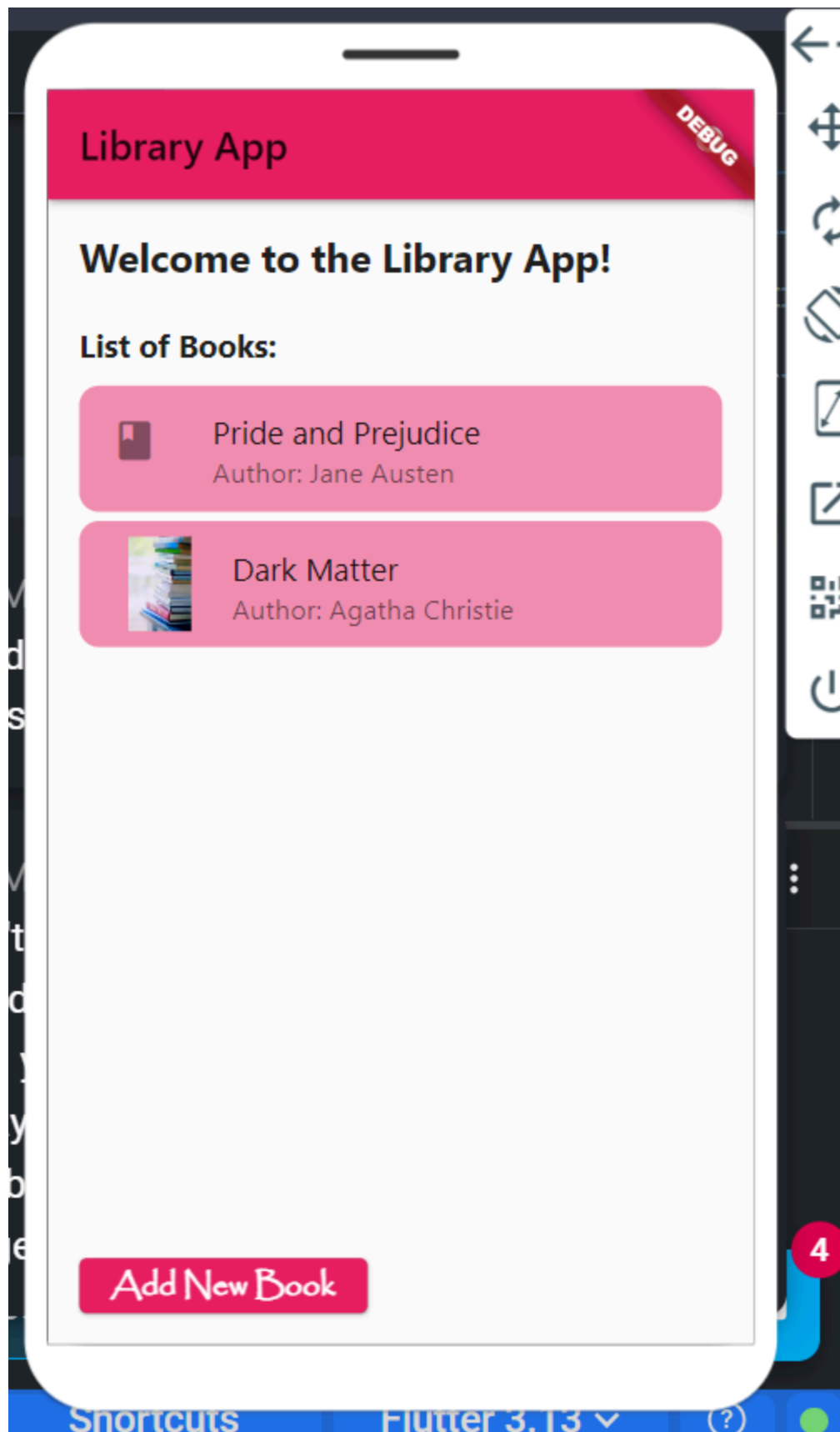
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Library App',
      theme: ThemeData(
        primarySwatch: Colors.pink,
        hintColor: Colors.black,
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            textStyle: TextStyle(
              fontFamily: 'Arial',
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ),
      home: MyLibraryPage(),
    );
  }
}
```

```
class MyLibraryPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title:
          Text('Library App', style:
Theme.of(context).textTheme.headline6),
        actions: [
          IconButton(
            icon: Icon(Icons.search),
            onPressed: () {
              // Handle search button press
            },
          ),
        ],
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              'Welcome to the Library App!',
              style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
            ),
            SizedBox(height: 20),
            Text(
              'List of Books:',
              style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold),
            ),
            SizedBox(height: 10),
          ],
        ),
      ),
    );
  }
}
```

```
Expanded(  
  child: ListView(  
    children: [  
      ListTile(  
        leading: Icon(Icons.book),  
        title: Text('Pride and Prejudice'),  
        subtitle: Text('Author: Jane Austen'),  
        tileColor: Colors.pink[200],  
        shape: RoundedRectangleBorder(  
          borderRadius: BorderRadius.circular(10),  
        ),  
        onTap: () {  
          // Handle book tap  
        },  
      ),  
      SizedBox(height: 5),  
      ListTile(  
        leading:  
          Image.asset('assets/books1.jpg', width:  
50, height: 50),  
        title: Text('Dark Matter'),  
        subtitle: Text('Author: Agatha Christie'),  
        tileColor: Colors.pink[200],  
        shape: RoundedRectangleBorder(  
          borderRadius: BorderRadius.circular(10),  
        ),  
        onTap: () {  
          // Handle book tap  
        },  
      ),  
    ],  
  ),  
),  
SizedBox(height: 20),
```

```
        ElevatedButton(  
          onPressed: () {  
            // Handle button press  
          },  
          child: Text('Add New Book'),  
        ),  
      ],  
    ),  
  ),  
);  
}
```



**Conclusion:** Thus we have successfully implemented using icons, images and customized fonts in our Flutter app.