

MPL - 3

AIM:- To add advanced Flutter UI by including widgets like Image, Fonts, Icons.

THEORY:-

Flutter provides robust mechanisms for working with images, fonts, and icons in your app's user interface. Here's a summary of their functionalities and considerations:

Images:

- **Loading and Displaying:** Use the Image widget to load and display images from various sources like assets, network URLs, or files. Adjust properties like fit, alignment, and opacity for customization.
- **Asset Management:** Store images within your app's assets directory (usually under assets/images/). Flutter automatically handles different screen resolutions and densities.
- **Network Images:** Use the Image.network constructor to directly load images from URLs. Ensure proper internet connectivity and consider caching mechanisms for efficiency.
- **Caching and Performance:** Flutter automatically caches downloaded images. For complex scenarios, explore advanced caching libraries like `cached_network_image`.

Fonts:

- **Using System Fonts:** Access system fonts available on the device using the Text widget's `fontFamily` property.
- **Custom Fonts:** Include custom fonts in your app's `pubspec.yaml` file and integrate them using the GoogleFonts package or by loading font files manually.
- **Font Styling:** Control font properties like size, weight, color, and more using the TextStyle class within the Text widget.
- **Text Layouts and Effects:** Flutter offers rich text editing and layout features. Explore properties like `textAlign`, `overflow`, and `textSpan` for advanced text formatting and effects.

Icons:

- **Material Icons:** Flutter provides built-in access to a vast collection of Material Design icons through the Icons class. Use them with the Icon widget for simple icon display.
- **Custom Icons:** You can create custom vector icons or use icon fonts. Popular packages like flutter_icons and font_awesome_flutter provide diverse icon sets.
- **Icon Styling:** Modify icons' colors, sizes, and other properties directly through the Icon widget's parameters
- **Animations and Interactions:** Integrate icon animations and interactions using gestures, animations, and state management techniques.

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

void main() {
  runApp(const WeddingPlannerApp());
}

class WeddingPlannerApp extends StatelessWidget {
  const WeddingPlannerApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Wedding Planner Dashboard',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const Dashboard(),
    );
  }
}

class Dashboard extends StatelessWidget {
  const Dashboard({Key? key}) : super(key: key);

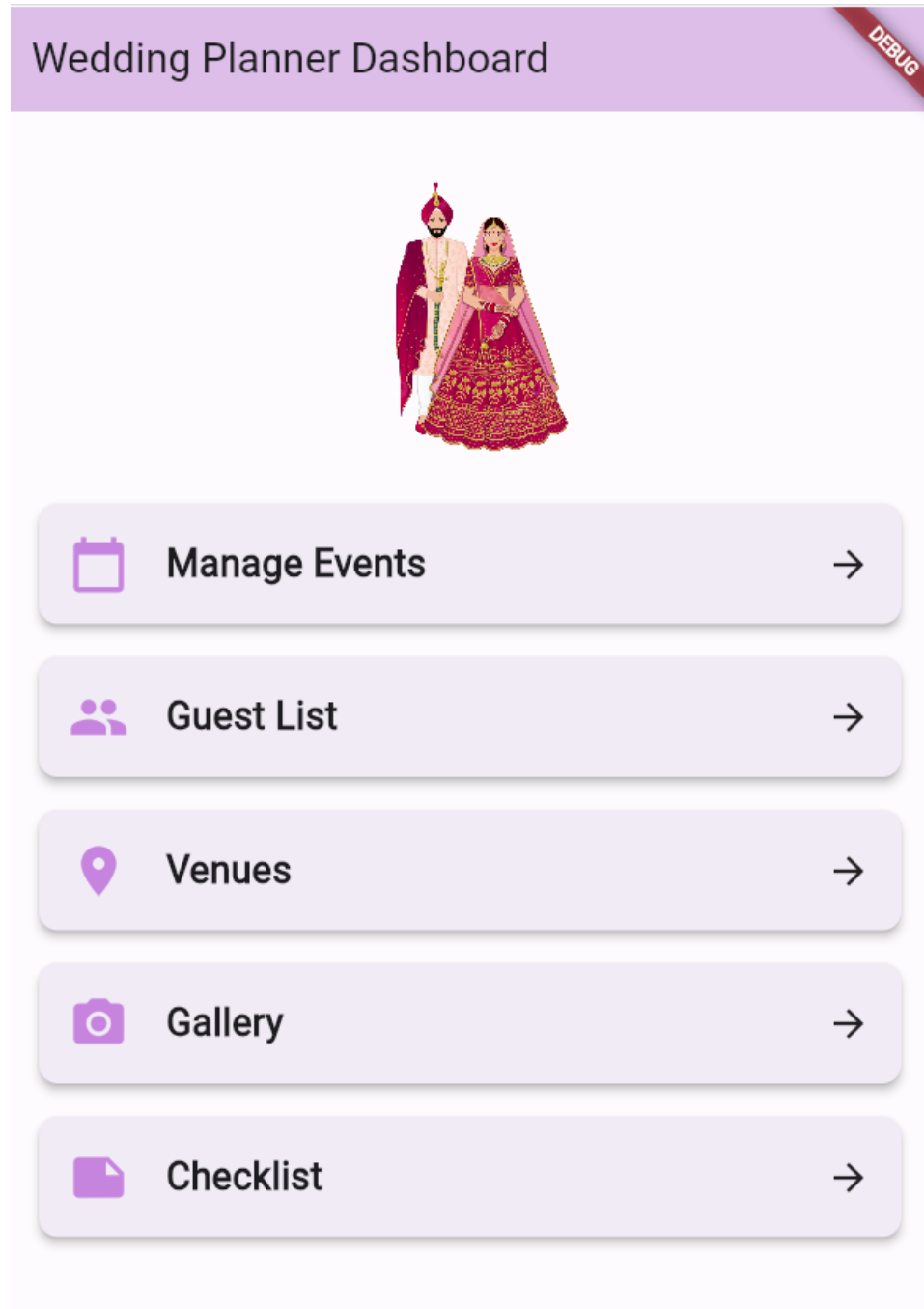
  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```
appBar: AppBar(  
  title: const Text('Wedding Planner Dashboard'),  
  backgroundColor: Colors.purple[100],  
) ,  
body: Padding(  
  padding: const EdgeInsets.all(16.0),  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.stretch,  
    children: <Widget>[  
      // Logo  
      Padding(  
        padding: const EdgeInsets.symmetric(vertical: 20.0),  
        child: Image.asset(  
          'wed_img.png',  
          height: 150,  
          width: 150,  
        ),  
      ),  
      // Cards  
      buildCard(  
        context,  
        title: 'Manage Events',  
        icon: Icons.calendar_today,  
        onTap: () {  
          // Navigate to event management page  
        },  
      ),  
      const SizedBox(height: 10),  
      buildCard(  
        context,  
        title: 'Guest List',  
        icon: Icons.people,  
        onTap: () {  
          // Navigate to guest list page  
        },  
      ),  
      const SizedBox(height: 10),  
      buildCard(  
        context,  
        title: 'Venues',
```

```
        icon: Icons.location_on,
        onTap: () {
          // Navigate to venue management page
        },
      ),
      const SizedBox(height: 10),
      buildCard(
        context,
        title: 'Gallery',
        icon: Icons.photo_camera,
        onTap: () {
          // Navigate to gallery page
        },
      ),
      const SizedBox(height: 10),
      buildCard(
        context,
        title: 'Checklist',
        icon: Icons.note,
        onTap: () {
          // Navigate to checklist page
        },
      ),
    ],
  ),
),
);
}
```

```
Widget buildCard(BuildContext context,
  {required String title,
  required IconData icon,
  required VoidCallback onTap}) {
  return InkWell(
    onTap: onTap,
    child: Card(
      elevation: 4,
      shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10)),
      child: Padding(
```

```
padding: const EdgeInsets.all(16.0),
child: Row(
  children: [
    Icon(icon,
      size: 32, color: const Color.fromARGB(255, 207, 133,
220)),
    const SizedBox(width: 20),
    Text(
      title,
      style:
        const TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
    ),
    const Spacer(),
    const Icon(Icons.arrow_forward),
  ],
),
),
);
}
```



Conclusion: We have successfully added advanced Flutter UI by including widgets like Image, Fonts, Icons.