NAME : SOLANKI DAKSHAY
ROLL NO : CE134
ID : 20CEUBG067

## ListView:

- ListView is the most commonly used scrolling widget. It displays its children one after another in the scroll direction. In the cross axis, the children are required to fill the ListView.
- If non-null, the itemExtent forces the children to have the given extent in the scroll direction. If non-null, the prototypeItem forces the children to have the same extent as the given widget in the scroll direction.
- Specifying an itemExtent or an prototypeItem is more efficient than letting the children determine their own extent because the scrolling machinery can make use of the foreknowledge of the children's extent to save work, for example when the scroll position changes drastically.
- You can't specify both itemExtent and prototypeItem, only one or none of them.

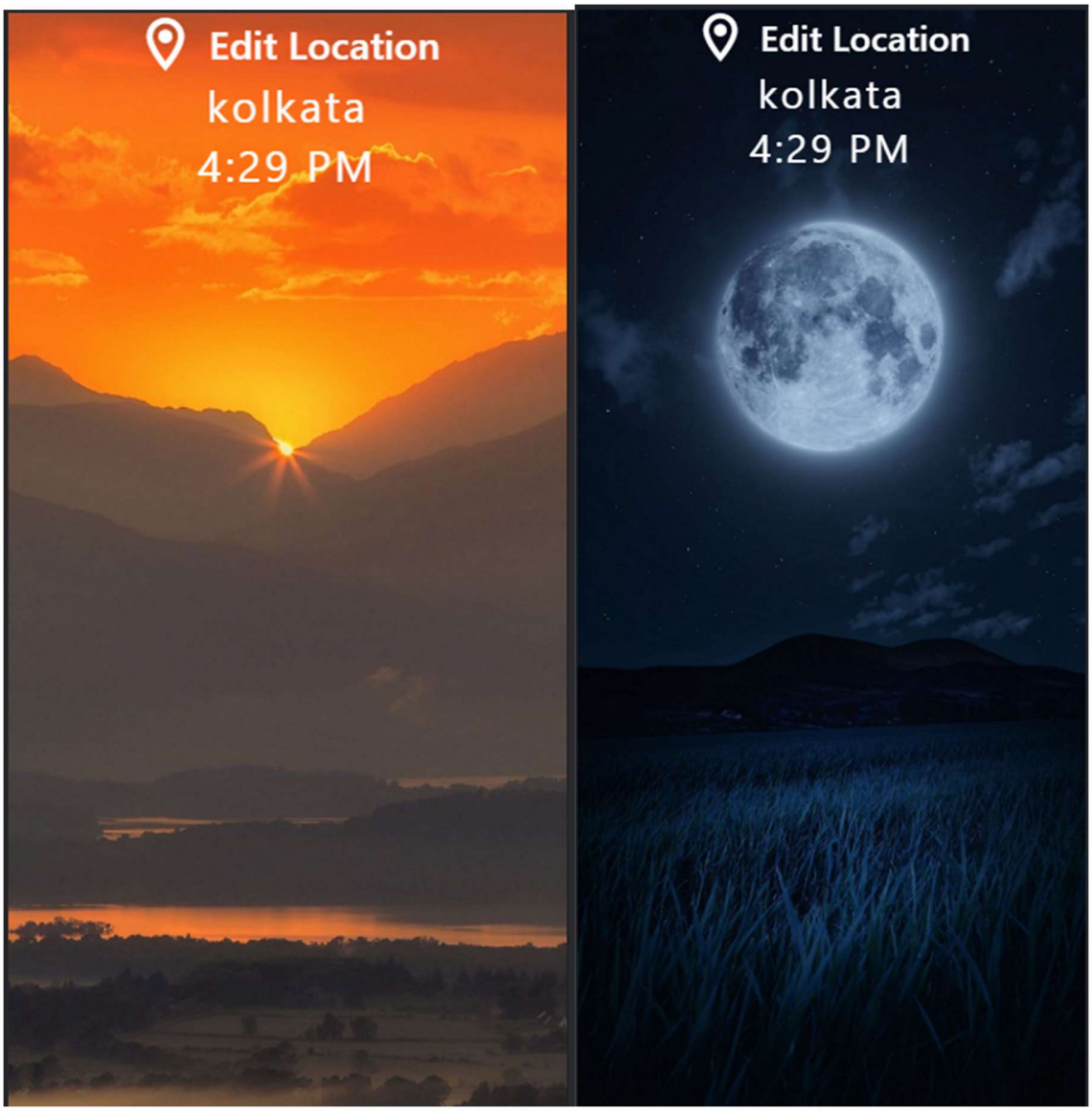
## ListTile:

- A list tile contains one to three lines of text optionally flanked by icons or other widgets, such as check boxes. The icons (or other widgets) for the tile are defined with the leading and trailing parameters. The first line of text is not optional and is specified with title. The value of subtitle, which *is* optional, will occupy the space allocated for an additional line of text, or two lines if isThreeLine is true. If dense is true then the overall height of this tile and the size of the DefaultTextStyles that wrap the title and subtitle widget are reduced.
- It is the responsibility of the caller to ensure that title does not wrap, and to ensure that subtitle doesn't wrap (if isThreeLine is false) or wraps to two lines (if it is true).
- The heights of the leading and trailing widgets are constrained according to the Material spec. An exception is made for one-line ListTiles for accessibility. Please see the example below to see how to adhere to both Material spec and accessibility requirements.

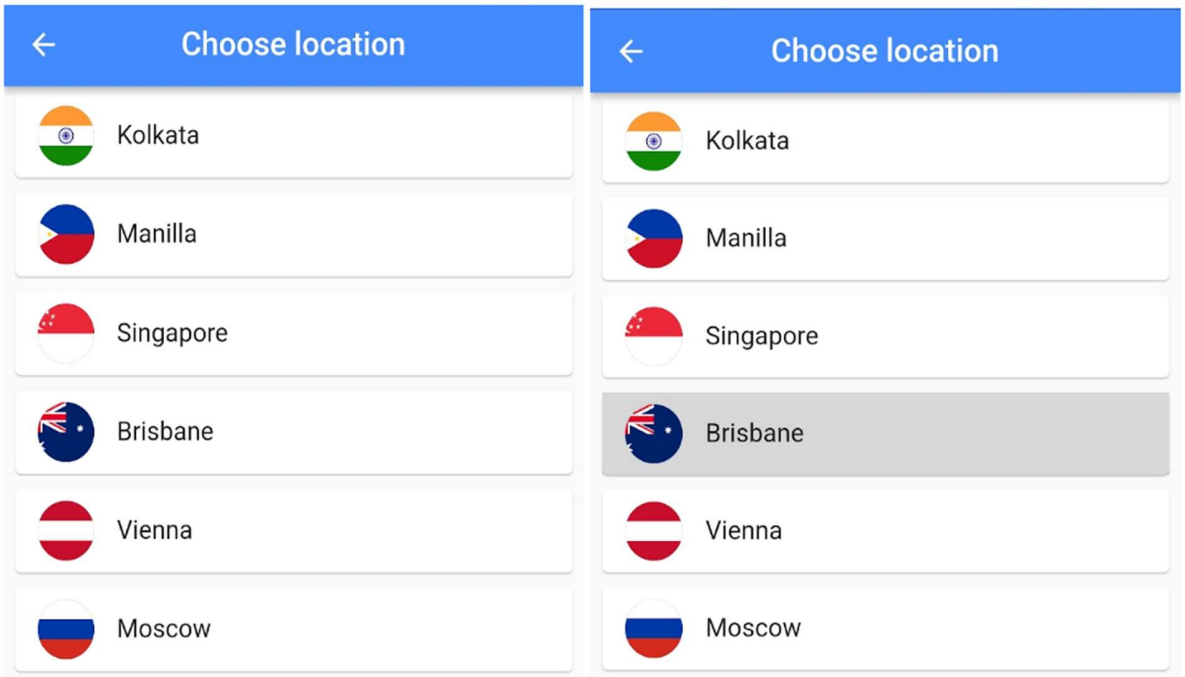
Home.dart :

```dart
class _HomeState extends State<Home> {
  Map<dynamic, dynamic>? data = {};
  @override
  Widget build(BuildContext context) {
    data = ModalRoute.of(context)?.settings.arguments as Map;
    print(data);

    String bgImg = data!['isDayTime'] ? 'day.jpg' : 'night.jpg';
    return Scaffold(
      body: SafeArea(
          child: Container(
        decoration: BoxDecoration(
            image: DecorationImage(image: AssetImage('assets/images/$bgImg'))),
        child: Center(
          child: Column(
            children: [
              TextButton.icon(
                onPressed: () {
                  Navigator.pushNamed(context, '/location');
                },
                icon: const Icon(
                  Icons.location_on_outlined,
                  size: 40,
                  color: Colors.white,
                ),
                label: const Text(
                  "Edit Location",
                  style: TextStyle(fontSize: 25, color: Colors.white),
                ),
              )
            ],
          ),
        ),
      )),
    );
  }
}
```


Edit Location
kolkata
4:29 PM
Edit Location
kolkata
4:29 PM

Code Test 3 :




Main.dart :

```
void main() => runApp(MaterialApp(

      debugShowCheckedModeBanner: false,

      initialRoute: '/',

      routes: {

        '/': (context) => const Loading(),

        '/home': (context) => const Home(),

        '/location': (context) => const ChooseLocation(),

      },

    ));
```

Choose_location.dart :

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

import 'package:flutter_lab/services/world_time.dart';

class ChooseLocation extends StatefulWidget {

  const ChooseLocation({Key? key}) : super(key: key);

  // const ChooseLocation({Key? key}) : super(key: key);

  @override

  State<ChooseLocation> createState() => _ChooseLocationState();

}



class _ChooseLocationState extends State<ChooseLocation> {

  int counter = 0;

  List<WordTime> locations = [

    WordTime(location: 'kolkata', flag: 'india.png', url: 'Asia/Kolkata'),

    WordTime(

        location: 'manilla', flag: 'phillipines.png', url: 'Asia/Manilla'),

    WordTime(

        location: 'singapore', flag: 'singapore.png', url: 'Asia/Singapore'),

    WordTime(

        location: 'brisbane', flag: 'australia.png', url: 'Australia/Brisbane'),

    WordTime(location: 'vienna', flag: 'austria.png', url: 'Europe/Vienna'),

    WordTime(location: 'moscow', flag: 'russia.png', url: 'Europe/Moscow'),

  ];



  // @override

  // void initState() {
```

```
  //   super.initState();
  // }


  void updateTime(dynamic index) async {
    WordTime instance = locations[index];
    await instance.getTime();


    Navigator.pop(context, {
      'location': instance.location,
      'flag': instance.flag,
      'time': instance.time,
      'isDayTime': instance.isDayTime,
    });
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Choose location'),
        centerTitle: true,
        backgroundColor: Colors.blueAccent,
      ),
      body: ListView.builder(
          itemCount: locations.length,
```

```
          itemBuilder: (context, index) {
            return Padding(
              padding:
                  const EdgeInsets.symmetric(vertical: 1.0, horizontal: 4.0),
              child: Card(
                child: ListTile(
                  onTap: () {
                    updateTime(index);
                  },
                  title: Text(locations[index].location.toString()),
                  leading: CircleAvatar(
                    backgroundImage:
                        AssetImage('assets/flags/${locations[index].flag}'),
                  ),
                ),
              ),
            );
          }),
    );
  }
}
```

Home.dart

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

```dart
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  Map<dynamic, dynamic>? data = {};
  @override
  Widget build(BuildContext context) {
    data = ModalRoute.of(context)?.settings.arguments as Map;
    print(data);

    String bgImg = data!['isDayTime'] ? 'day.jpg' : 'night.jpg';
    // String bgImg = 'night.jpg';
    Color? appBgColor = data!['isDayTime'] ? Colors.blue :
Colors.indigo[900];
    return Scaffold(
      backgroundColor: appBgColor,
      body: SafeArea(
          child: Container(
```

```
        decoration: BoxDecoration(
            image: DecorationImage(image:
AssetImage('assets/images/$bgImg'))),
        child: Center(
          child: Column(
            children: [
              TextButton.icon(
                onPressed: () {
                  Navigator.pushNamed(context, '/location');
                },
                icon: const Icon(
                  Icons.location_on_outlined,
                  size: 40,
                  color: Colors.white,
                ),
                label: const Text(
                  "Edit Location",
                  style: TextStyle(fontSize: 25, color:
Colors.white),
                ),
              ),
              Text(
                data!['location'],
```

```
                style: TextStyle(
                    letterSpacing: 2, fontSize: 28, color:
Colors.white),
              ),
              Text(
                data!['time'],
                style: TextStyle(
                    letterSpacing: 2, fontSize: 28, color:
Colors.white),
              ),
            ],
          ),
        ),
      )),
    );
  }
}
```

Word_time.dart :

```
// ignore_for_file: public_member_api_docs,
sort_constructors_first
import 'dart:convert';


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

```dart
import 'package:http/http.dart';

import 'package:intl/intl.dart';


class WordTime {

  String? location;

  String? time;

  String? flag;

  String? url;

  bool? isDayTime;


  WordTime({this.location, this.flag, this.url});


  Future<void> getTime() async {

    try {

      Response response =

          await
get(Uri.parse('http://worldtimeapi.org/api/timezone/$url'));

      Map timeData = jsonDecode(response.body);


      String dateTime = timeData['datetime'];

      String offset = timeData['utc_offset'];


      String offsetHours = offset.substring(1, 3);
```

```
      String offsetMinutes = offset.substring(4, 6);


      DateTime currentTime = DateTime.parse(dateTime);


      currentTime = currentTime.add(Duration(
          minutes: int.parse(offsetMinutes), hours:
int.parse(offsetHours)));


      // time = currentTime.toString();
      isDayTime =
          (currentTime.hour > 6 && currentTime.hour < 20) ? true :
false;
      time = DateFormat.jm().format(currentTime);
    } catch (e) {
      print('Error Caught $e');
      time = "Could not get time";
    }
  }
}
```

Repository link :-