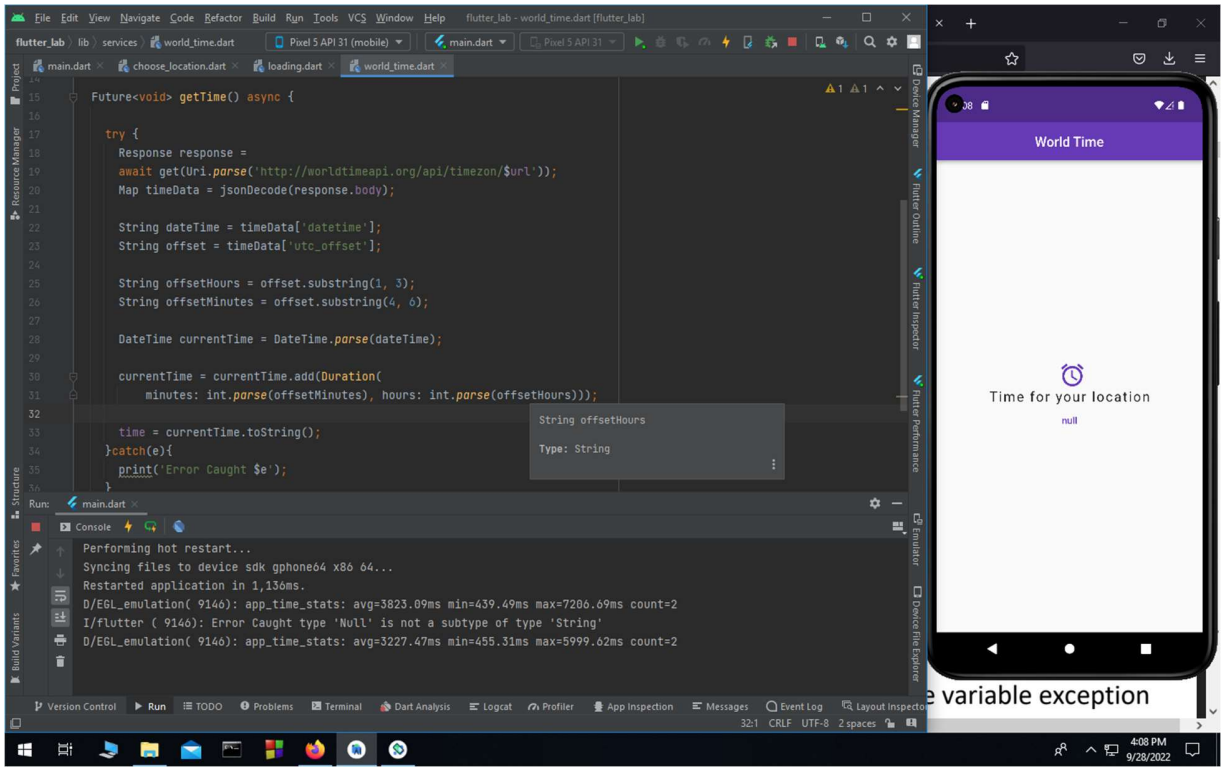
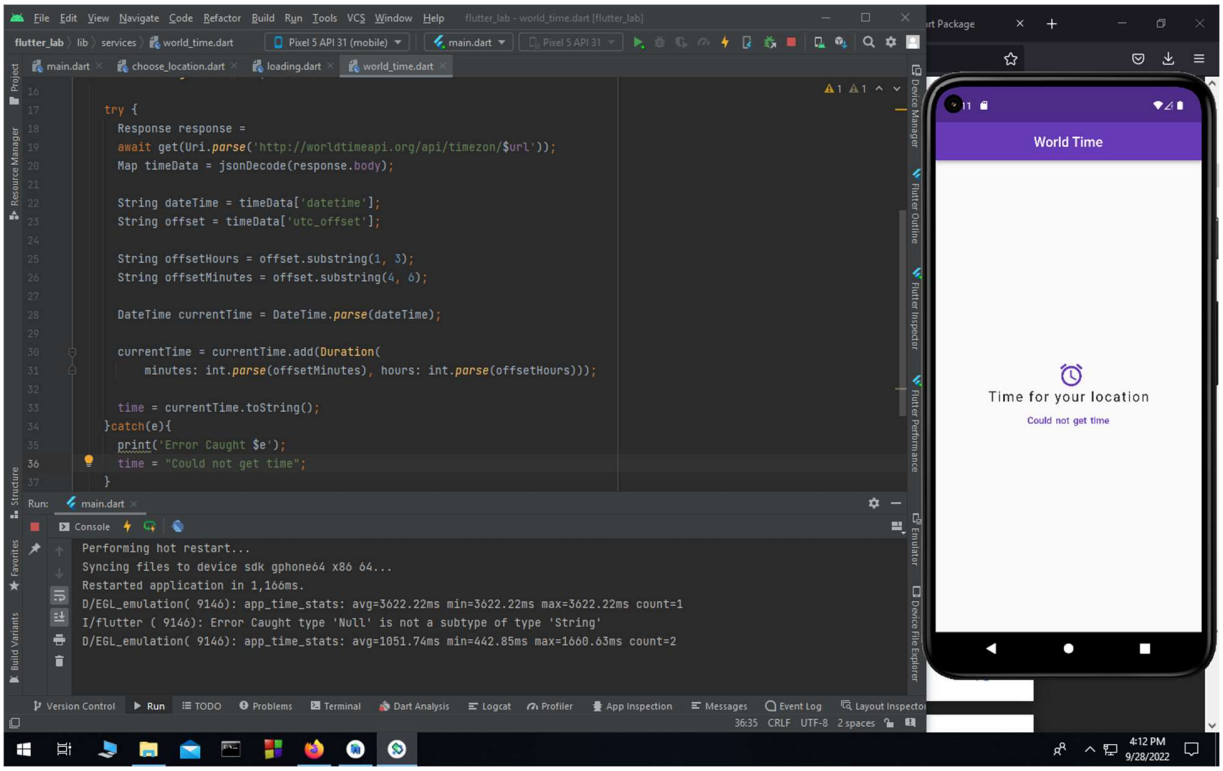
NAME : Faizan Vora
ROLL NO : CE157
ID : 20CEUOS020

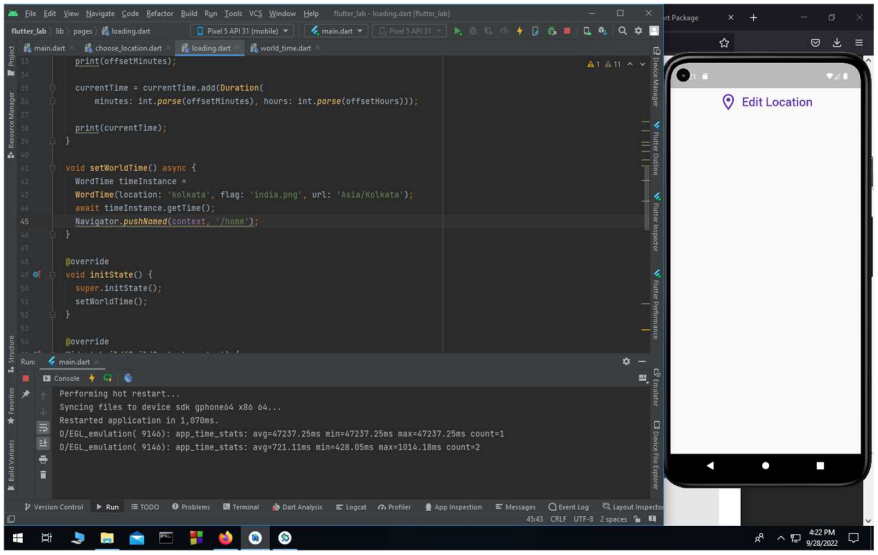
DateFormat Class:

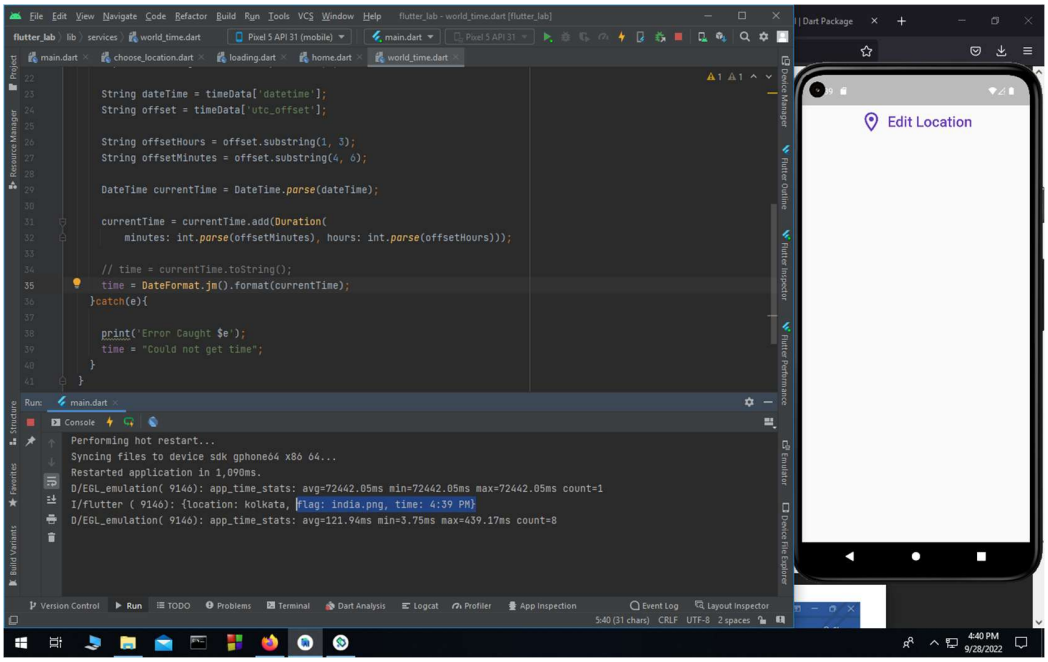
- DateFormat is for formatting and parsing dates in a locale-sensitive manner.
- It allows the user to choose from a set of standard date time formats as well as specify a customized pattern under certain locales. Date elements that vary across locales include month name, week name, field order, etc. We also allow the user to use any customized pattern to parse or format date-time strings under certain locales. Date elements that vary across locales include month name, weekname, field, order, etc.



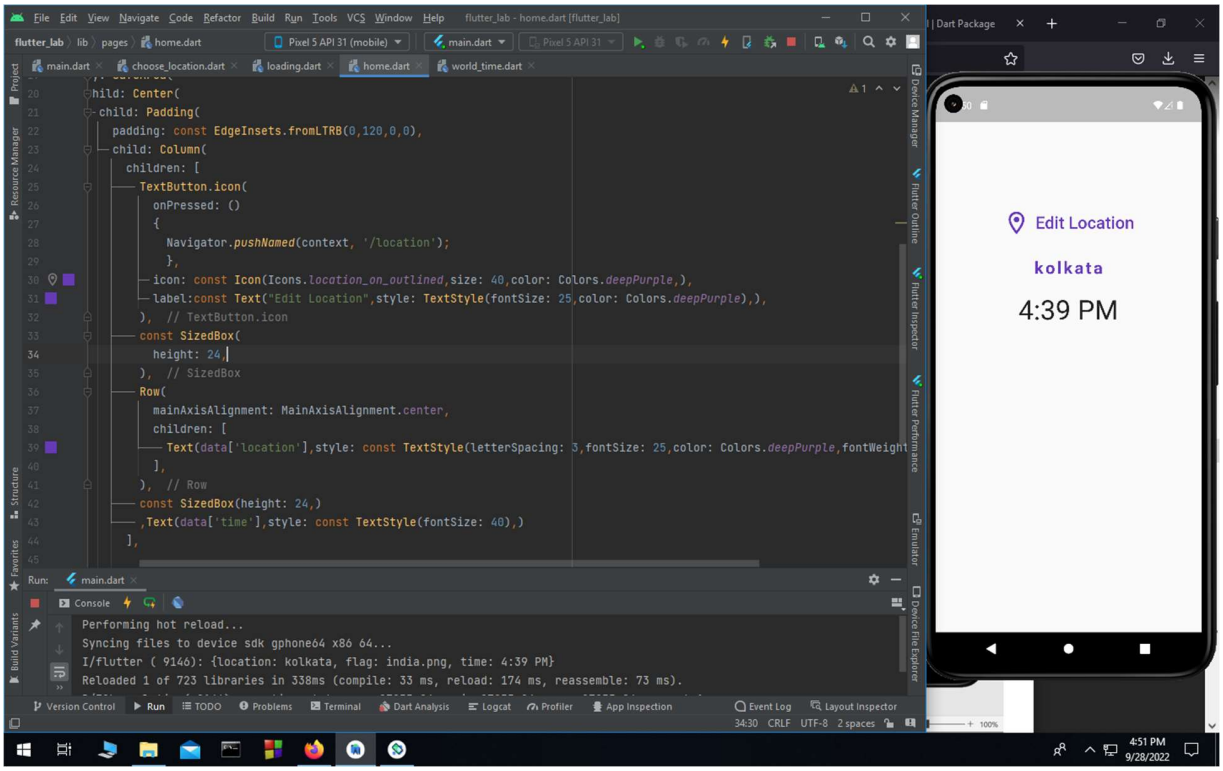


Navigate to home page after getting timeinstance :



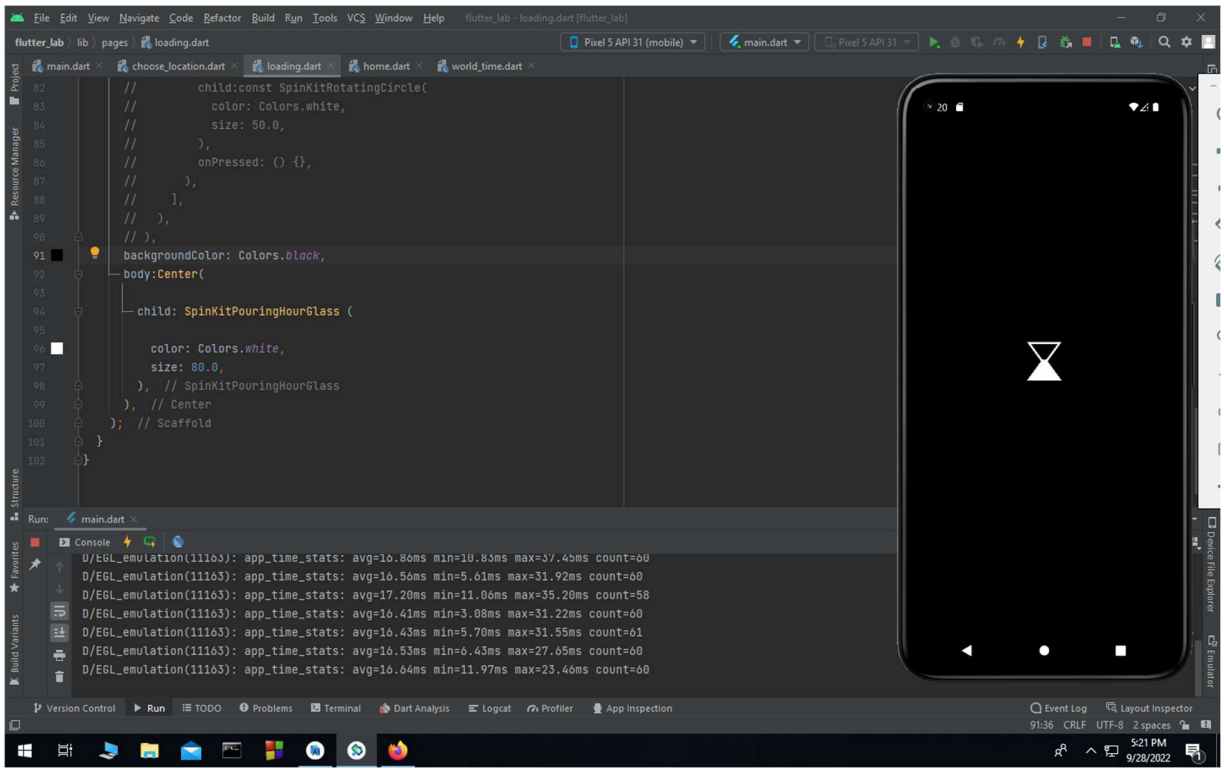

String dateTime = timeData['datetime'];
String offset = timeData['utc_offset'];
String offsetHours = offset.substring(1, 3);
String offsetMinutes = offset.substring(4, 6);
DateTime currentTime = DateTime.parse(dateTime);
currentTime = currentTime.add(Duration(
minutes: int.parse(offsetMinutes), hours: int.parse(offsetHours)));
// time = currentTime.toString();
time = DateFormat.jm().format(currentTime);
}catch(e){
print('Error Caught $e');
time = "Could not get time";
}
Performing hot restart...
Syncing files to device sdk gphone64 x86 64...
Restarted application in 1,090ms.
D/EGL_emulation(9146): app_time_stats: avg=72442.05ms min=72442.05ms max=72442.05ms count=1
I/flutter (9146): {location: kolkata, flag: india.png, time: 4:39 PM}
D/EGL_emulation(9146): app_time_stats: avg=121.94ms min=3.75ms max=439.17ms count=8
Edit Location
4:40 PM
9/28/2022

Home.dart :


child: Center(
child: Padding(
padding: const EdgeInsets.fromLTRB(0,120,0,0),
child: Column(
children: [
TextButton.icon(
onPressed: ()
{
Navigator.pushNamed(context, '/location');
},
icon: const Icon(Icons.location_on_outlined,size: 40,color: Colors.deepPurple,),
label:const Text("Edit Location",style: TextStyle(fontSize: 25,color: Colors.deepPurple),),
), // TextButton.icon
const SizedBox(
height: 24,
), // SizedBox
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(data['location'],style: const TextStyle(letterSpacing: 3,fontSize: 25,color: Colors.deepPurple,fontWeight
],
), // Row
const SizedBox(height: 24,)
,Text(data['time'],style: const TextStyle(fontSize: 40),)
],
Performing hot reload...
Syncing files to device sdk gphone64 x86 64...
I/flutter (9146): {location: kolkata, flag: india.png, time: 4:39 PM}
Reloaded 1 of 723 libraries in 338ms (compile: 33 ms, reload: 174 ms, reassemble: 73 ms).
Edit Location
kolkata
4:39 PM
4:51 PM
9/28/2022

SpinKit:

- Spinkit is a collection of loading indicators animated with flutter. It has huge animated loading indicators which are basically used when we are loading something.



Home.dart

```dart
import 'package:flutter/material.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);
    return Scaffold(
      body: SafeArea(
        child: Center(
          child: Padding(
            padding: const EdgeInsets.fromLTRB(0,120,0,0),
            child: Column(
              children: [
                TextButton.icon(
                  onPressed: ()
                  {
                    Navigator.pushNamed(context, '/location');
                    },
                  icon: const Icon(Icons.location_on_outlined,size: 40,color:
Colors.deepPurple,),
                  label:const Text("Edit Location",style: TextStyle(fontSize:
25,color: Colors.deepPurple),),
                ),
                const SizedBox(
                  height: 24,
                ),
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Text(data['location'],style: const TextStyle(letterSpacing:
3,fontSize: 25,color: Colors.deepPurple,fontWeight: FontWeight.bold),)
                  ],
```

```
                  ),
                  const SizedBox(height: 24,)
                  ,Text(data['time'],style: const TextStyle(fontSize: 40),)
                ],

              ),

            ),

          )
        ),
      );
    }
}
```

Main.dart :

```
void main() => runApp(MaterialApp(
      debugShowCheckedModeBanner: false,
      // initialRoute: '/home',
      initialRoute: '/',
      routes: {
        '/': (context) => const Loading(),
        '/home': (context) => const Home(),
        '/location': (context) => const ChooseLocation(),
      },
    ));
```

Loading.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:flutter_lab/services/world_time.dart';
import 'package:http/http.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';

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

```
  @override
  State<Loading> createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
  void getTime() async {
    final response = await get(
        Uri.parse("http://worldtimeapi.org/api/timezone/Asia/Kolkata"));
    print(response.body);

    Map timeData = jsonDecode(response.body);
    print(timeData);
    String dateTime = timeData['datetime'];
    String offset = timeData['utc_offset'];

    print(dateTime);
    print(offset);
    DateTime currentTime = DateTime.parse(dateTime);
    print(currentTime);

    String offsetHours = offset.substring(1, 3);
    print(offsetHours);
    String offsetMinutes = offset.substring(4, 6);
    print(offsetMinutes);

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

    print(currentTime);
  }

  void setWorldTime() async {
    WordTime timeInstance =
    WordTime(location: 'kolkata', flag: 'india.png', url: 'Asia/Kolkata');
    await timeInstance.getTime();

    // Navigator.pushNamed(context, '/home',arguments: {
    //   'location' : timeInstance.location,
    //   'flag' : timeInstance.flag,
    //   'time' : timeInstance.time
    // });
  }
```

```dart
@override
void initState() {
  super.initState();
  setWorldTime();
}

@override
Widget build(BuildContext context) {
  return const Scaffold(
    // appBar: AppBar(
    //
    //   backgroundColor: Colors.deepPurple,
    //   title: const Text('World Time'),
    //   centerTitle: true,
    // ),
    // body: Center(
    //   child: Column(
    //     mainAxisAlignment: MainAxisAlignment.center,
    //     children: [
    //       IconButton(
    //           onPressed: () {},
    //           icon: const Icon(
    //             Icons.alarm,
    //             color: Colors.deepPurple,
    //             size: 40,
    //           )),
    //       const Text("Time for your location",
    //           style: TextStyle(fontSize: 20, letterSpacing: 2)),
    //       TextButton(
    //         child:const SpinKitRotatingCircle(
    //           color: Colors.white,
    //           size: 50.0,
    //         ),
    //         onPressed: () {},
    //       ),
    //     ],
    //   ),
    // ),
    backgroundColor: Colors.black,
    body:Center(

      child: SpinKitPouringHourGlass (

        color: Colors.white,
```

```
          size: 80.0,
        ),
      ),
    );
  }
}
```

World_time.dart

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

import 'package:flutter/scheduler.dart';
import 'package:http/http.dart';
import 'package:intl/intl.dart';

class WordTime {
  String? location;
  String? time;
  String? flag;
  String? url;

  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();
      time = DateFormat.jm().format(currentTime);
```

```
    }catch(e){

      print('Error Caught $e');
      time = "Could not get time";
    }
  }
}
```