Practical No – 13

Title: Flutter program based on RestAPI

Q.1) Create a Flutter application to demonstrate REST API

Implementation:

pages

home.dart

```
import 'package:flutter/material.dart';
class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}
class _HomeState extends State<Home> {
  Map data = {};
  @override
  Widget build(BuildContext context) {
    data = data.isNotEmpty ? data : ModalRoute.of(context)!.settings.arguments as Map;
    //set background image
    String bgImage = data['isDaytime'] ? 'day.png' : 'night.png';
    Color? bgColor = data['isDaytime'] ? Colors.blue : Colors.indigo[700];
    return Scaffold(
      backgroundColor: bgColor,
      body: SafeArea(
        child: Container(
          decoration: BoxDecoration(
            image: DecorationImage(
              image: AssetImage('assets/$bgImage'),
              fit: BoxFit.cover,
            )
```

```
),
child: Padding(
  padding: const EdgeInsets.fromLTRB(0, 120.0, 0, 0),
  child: Column(
    children: <Widget>[
      TextButton.icon(
        onPressed: () async {
          dynamic result = await Navigator.pushNamed(context, '/location');
          if(result != null) {
            setState(() {
              data = {
                'time': result['time'],
                'location': result['location'],
                'isDaytime': result['isDaytime'],
                'flag': result['flag']
              };
            });
          }
        },
        icon: Icon(
          Icons.edit_location,
          color: Colors.grey[300],
        ),
        label: Text(
          'Select Location',
          style: TextStyle(
            color:  Colors.grey[300],
          ),
        ),
      ),
      const SizedBox(height: 20.0),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text(
            data['location'],
            style: const TextStyle(
              fontSize: 28.0,
              letterSpacing: 2.0,
              color: Colors.white,
            ),
          ),
        ],
      ),
      const SizedBox(height: 20.0),
      Text(
        data['time'],
        style: const TextStyle(
          fontSize: 66.0,
          color: Colors.white
        )),],), ), ),), ); }}
```

<u>loading.dart</u>

```
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
import 'package:rest_api/services/world_time.dart';
class Loading extends StatefulWidget {
  @override
  _LoadingState createState() => _LoadingState();
}
```

```
class _LoadingState extends State<Loading> {
  void setupWorldTime() async {
    WorldTime instance = WorldTime(location: 'Kolkata', flag: 'india.png', url: 'Asia/Kolkata');
    await instance.getTime();
    Navigator.pushReplacementNamed(context, '/home', arguments: {
      'location': instance.location,
      'flag': instance.flag,
      'time': instance.time,
      'isDaytime': instance.isDaytime
    });
  }
  @override
  void initState() {
    super.initState();
    setupWorldTime();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[900],
      body: const Center(
        child: SpinKitHourGlass(
          duration: Duration(
            milliseconds: 2000,
          ),
          color: Colors.white,
          size: 50.0,
        ) ));
  }}
```

choose_location.dart

```
import 'package:flutter/material.dart';
import 'package:rest_api/services/world_time.dart';
class ChooseLocation extends StatefulWidget {
  @override
  _ChooseLocationState createState() => _ChooseLocationState();
}
class _ChooseLocationState extends State<ChooseLocation> {
  List<WorldTime> locations = [
    WorldTime(url: 'Asia/Kolkata', location: 'Kolkata', flag: 'india.png'),
    WorldTime(url: 'Europe/Berlin', location: 'Berlin', flag: 'germany.png'),
    WorldTime(url: 'Europe/London', location: 'London', flag: 'uk.png'),
    WorldTime(url: 'Europe/Berlin', location: 'Athens', flag: 'greece.png'),
    WorldTime(url: 'Africa/Cairo', location: 'Cairo', flag: 'egypt.png'),
    WorldTime(url: 'Africa/Nairobi', location: 'Nairobi', flag: 'kenya.png'),
    WorldTime(url: 'America/Chicago', location: 'Chicago', flag: 'usa.png'),
    WorldTime(url: 'America/New_York', location: 'New York', flag: 'usa.png'),
    WorldTime(url: 'Asia/Seoul', location: 'Seoul', flag: 'south_korea.png'),
    WorldTime(url: 'Asia/Jakarta', location: 'Jakarta', flag: 'indonesia.png'),
  ];
  void updateTime(index) async {
    WorldTime instance = locations[index];
    await instance.getTime();
    Navigator.pop(context, {
      'location': instance.location,
      'time': instance.time,
      'flag': instance.flag,
      'isDaytime': instance.isDaytime,
    });
  }
  @override
  void initState() {
    super.initState();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[200],
      appBar: AppBar(
        backgroundColor: Colors.blue[900],
        title: const Text('Choose a Location'),
        centerTitle: true,
        elevation: 0,
      ),
      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),
                leading: CircleAvatar(
                  backgroundImage: AssetImage('assets/${locations[index].flag}'),
                ), ), ), );
        }),
    );}}
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:rest_api/pages/home.dart';
import 'package:rest_api/pages/loading.dart';
import 'package:rest_api/pages/choose_location.dart';
import 'choose_location.dart';
import 'home.dart';
import 'loading.dart';
void main() => runApp(MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => Loading(),
    '/home': (context) => Home(),
    '/location': (context) => ChooseLocation(),
  }
));
```

services

world_time.dart

```
import 'package:http/http.dart';
import'dart:convert';
import 'package:intl/intl.dart';
class WorldTime {
  String location;
  String time = ' ';
  String flag;
  String url;
  bool isDaytime = true;
  WorldTime({required this.location, required this.flag, required this.url});
  Future<void> getTime() async {
    try {
      Response response = await
get(Uri.parse('http://worldtimeapi.org/api/timezone/$url'));
      Map data = jsonDecode(response.body);
      //get propertise from json
      String datetime = data['datetime'];
      String offset = data['utc_offset'].substring(1,3);
      //create datetime object
      DateTime now = DateTime.parse(datetime);
      now = now.add(Duration(hours: int.parse(offset)));
      isDaytime = now.hour > 6 && now.hour < 20? true : false;
      time = DateFormat.jm().format(now);
    }
    catch (e) {
      print(e);
      time = 'could not get time';
    }
  }
}
```

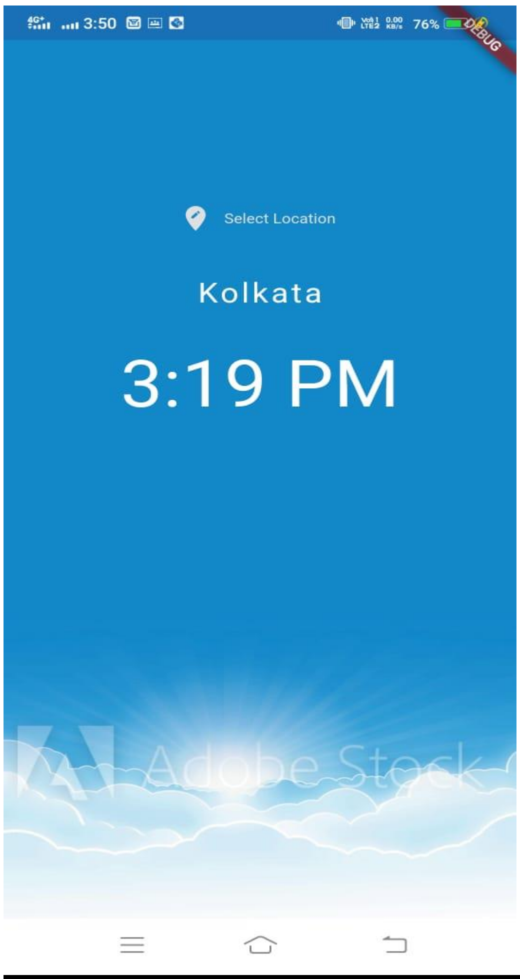
pubspec.yml

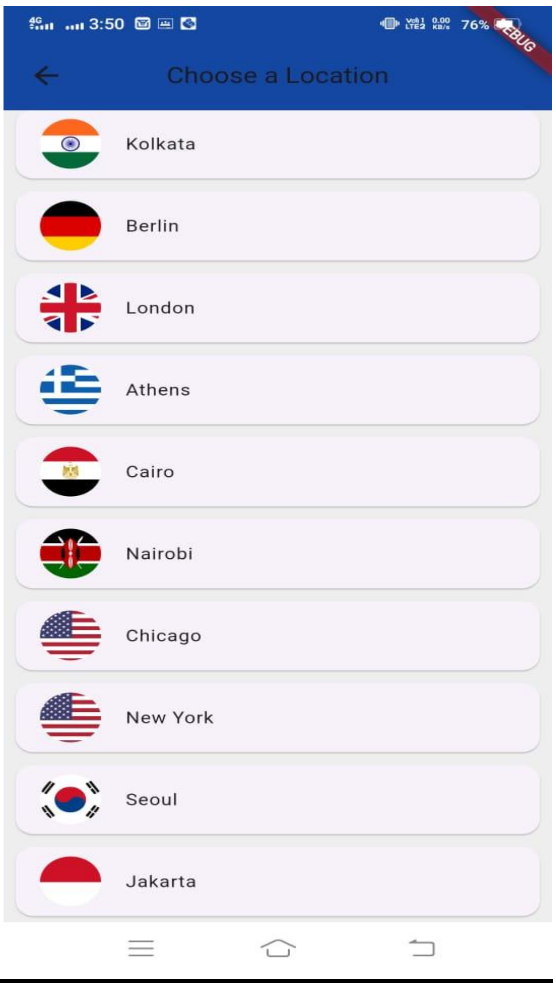
```
name: rest_api
description: "A new Flutter project."
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
```

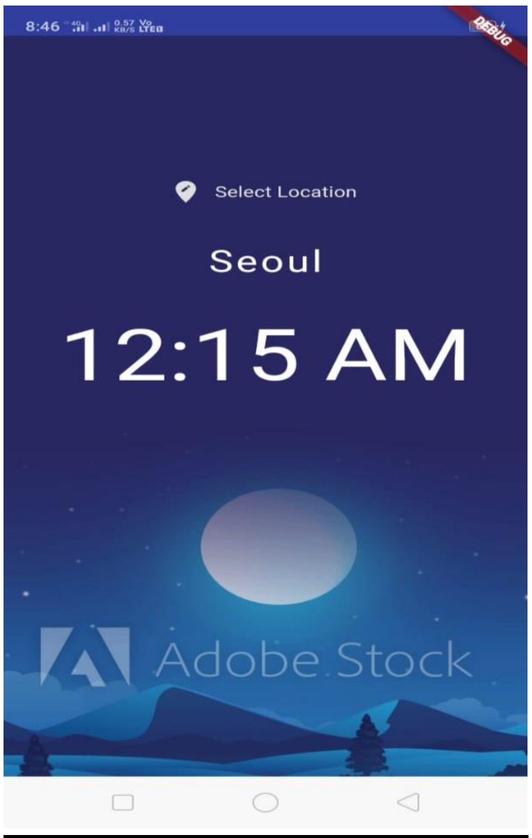
```
# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as
CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyR
eference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1
environment:
  sdk: '>=3.2.2 <4.0.0'
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  flutter_spinkit: ^5.2.0
  http: ^1.1.2
  intl: ^0.19.0
dev_dependencies:
  flutter_test:
    sdk: flutter
  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^2.0.0
# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec
# The following section is specific to Flutter packages.
flutter:
  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true
  # To add assets to your application, add an assets section, like this:
  assets:
    - assets/
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg
  # An image asset can refer to one or more resolution-specific "variants", see
  # https://flutter.dev/assets-and-images/#resolution-aware
  # For details regarding adding assets from package dependencies, see
  # https://flutter.dev/assets-and-images/#from-packages

  # To add custom fonts to your application, add a fonts section here,
  # in this "flutter" section. Each entry in this list should have a
  # "family" key with the font family name, and a "fonts" key with a
  # list giving the asset and other descriptors for the font. For
  # example:
  # fonts:
  #   - family: Schyler
  #     fonts:
  #       - asset: fonts/Schyler-Regular.ttf
  #       - asset: fonts/Schyler-Italic.ttf
  #         style: italic
  #   - family: Trajan Pro
  #     fonts:
  #       - asset: fonts/TrajanPro.ttf
  #       - asset: fonts/TrajanPro_Bold.ttf
  #         weight: 700
  #
  # For details regarding fonts from package dependencies,
  # see https://flutter.dev/custom-fonts/#from-packages
```

Output :


3:50
76%
DEBUG
Select Location
Kolkata
3:19 PM
Adobe Stock


3:50
76%
DEBUG
Choose a Location
Kolkata
Berlin
London
Athens
Cairo
Nairobi
Chicago
New York
Seoul
Jakarta


8:46
DEBUG
Select Location
Seoul
12:15 AM
Adobe Stock