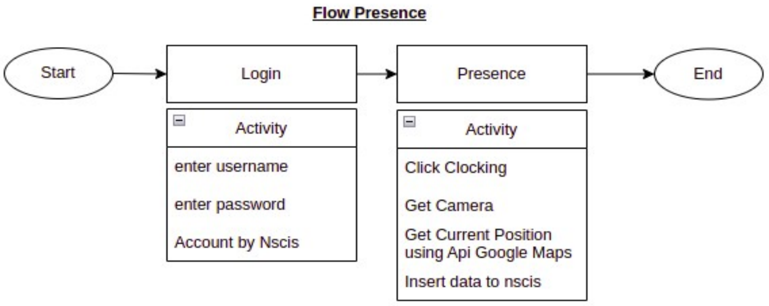
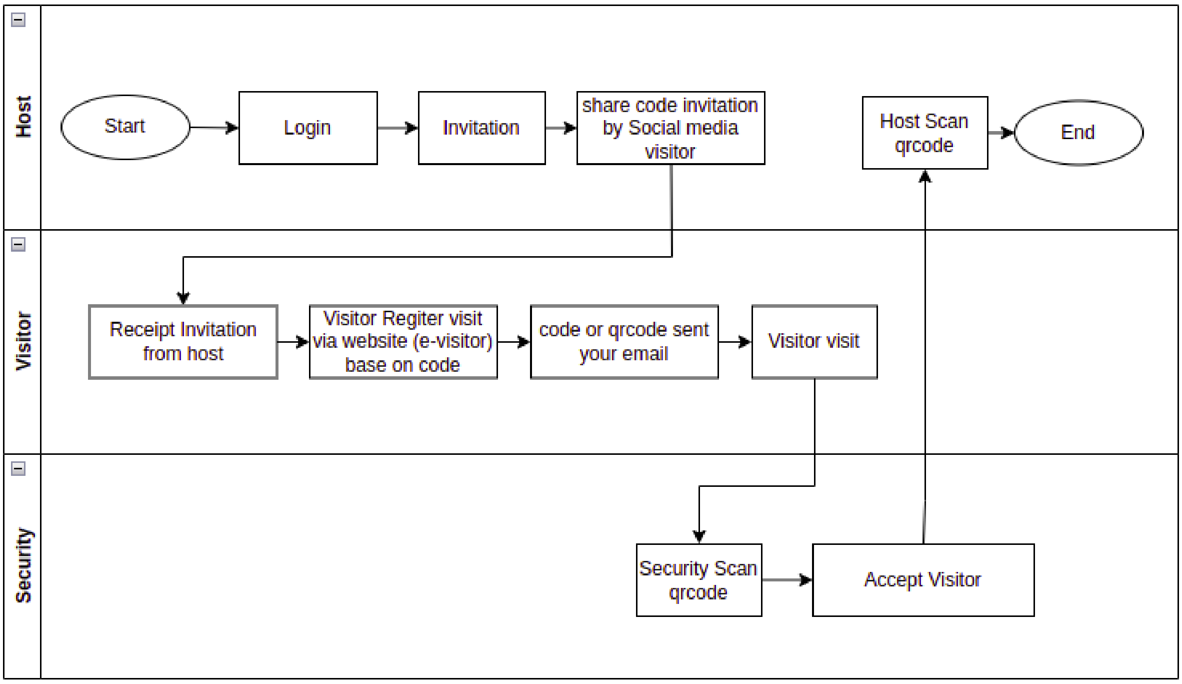
Presence Realtime (by api google maps) & E-Visitor

1. Flow Presence Realtime



2. E-Visitor



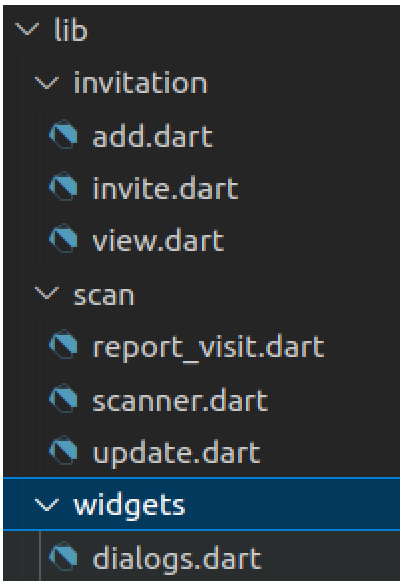
Host
Start
Login
Invitation
share code invitation by Social media visitor
Host Scan qrcode
End
Visitor
Receipt Invitation from host
Visitor Regiter visit via website (e-visitor) base on code
code or qrcode sent your email
Visitor visit
Security
Security Scan qrcode
Accept Visitor

3. List API

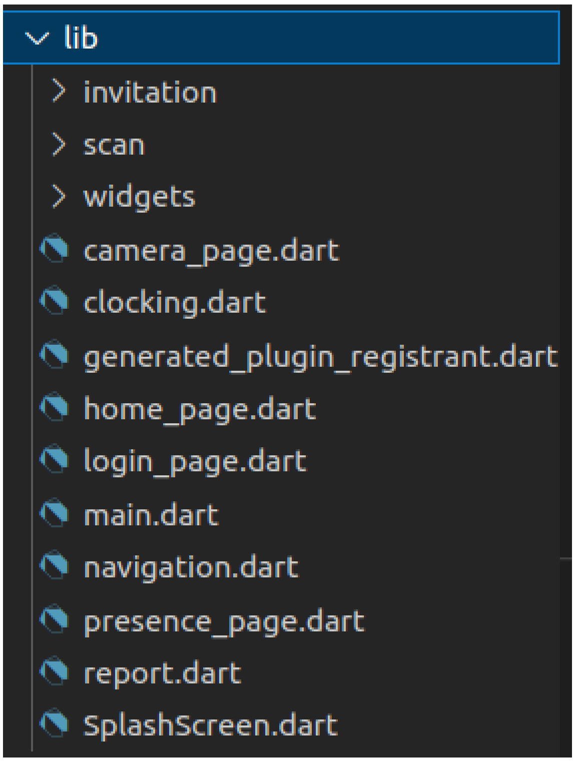
API	Parameter	Methode	Description
https://nscis.nsctechnology.com/index.php?r=auth/login	Username, Password	POST	API Login using username and password parameters
https://nscis.nsctechnology.com/index.php?r=user/view-api&id=" + username	Username	POST	Displaying logged in user sessions
https://nscis.nsctechnology.com/index.php?r=precense/user-api&id='${username}'	Username	GET	Display fullname
https://nscis.nsctechnology.com/index.php?r=precense/create-api	User_id, latitude, Longtitude, location	POST	Insert data to database presence
https://nscis.nsctechnology.com/index.php?r=precense/list-api&id='${username}'	Username	GET	Display report presence base on account

https://nscis.nsctechnology.com/index.php?r=t-invitation/invitation-api&id=" + username	Username	GET	Data Invitation Base on username
https://nscis.nsctechnology.com/index.php?r=t-invitation/create-api	user_id, schedule_date, time_schedule,description	POST	Insert data invitation to database invitation
https://nscis.nsctechnology.com/index.php?r=t-invitation/view-api&id='${widget.id}'	id_invitation	POST	View data invitation base on id invitation
https://nscis.nsctechnology.com/index.php?r=t-visitor/view-security&id=" + username	username	GET	View data otoritas to akses Security
http://nscis.nsctechnology.com/index.php?r=t-visitor/view-api&id=${widget.id}	-	GET	View data visitor
https://nscis.nsctechnology.com/index.php?r=t-visitor/update-api	id_visitor, status	POST	Update status data visitor base on otoritas
https://nscis.nsctechnology.com/index.php?r=t-visitor/visitor-api&id=" + username	username	GET	View data result visitor

4. Structure Directory Project



```
lib
  invitation
    add.dart
    invite.dart
    view.dart
  scan
    report_visit.dart
    scanner.dart
    update.dart
  widgets
    dialogs.dart
```



```
lib
  invitation
  scan
  widgets
  camera_page.dart
  clocking.dart
  generated_plugin_registrant.dart
  home_page.dart
  login_page.dart
  main.dart
  navigation.dart
  presence_page.dart
  report.dart
  SplashScreen.dart
```

5. List Package
   location: pubspec.yml

```
dependencies:
  camera: ^0.10.0+3
  cupertino_icons: ^1.0.2
  flutter:
    sdk: flutter
  flutter_easyloading: ^3.0.5
  flutter_geocoder: ^0.2.2-nullsafety
  flutter_staggered_grid_view: ^0.6.2
  google_maps_flutter: ^2.2.1
  http: ^0.13.5
  intl: ^0.17.0
  location: ^4.4.0
  path_provider: ^2.0.11
  percent_indicator: ^4.2.2
  qr_code_scanner: ^1.0.1
  share_plus: ^5.0.0
  shared_preferences: ^2.0.15
```

6. File Configuration
   Location : android/app/src/main/AndroidManifest.xml (

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.biu_project">
    <uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
   <uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
   <application
```

```
        android:label="biu_project"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">
        <meta-data android:name="com.google.android.geo.API_KEY"
              android:value="api key Maps from google"/>
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|
smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:usesCleartextTraffic="true"
            android:windowSoftInputMode="adjustResize">
            <!-- Specifies an Android theme to apply to this Activity as soon as
                 the Android process has started. This theme is visible to the user
                 while the Flutter UI initializes. After that, this theme continues
                 to determine the Window background behind the Flutter UI. -->
            <meta-data
              android:name="io.flutter.embedding.android.NormalTheme"
              android:resource="@style/NormalTheme"
              />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <!-- Don't delete the meta-data below.
             This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

Note: Just change the highlight

7. Source Code sile

SplashScreen.dart

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

class SplashScreen extends StatefulWidget {
  final Color backgroundColor = Colors.white;
  final TextStyle styleTextUnderTheLoader = TextStyle(
      fontSize: 18.0, fontWeight: FontWeight.bold, color: Colors.black);

  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {

  String _versionName = 'V1.0';
  final splashDelay = 7;
```

```
@override
void initState() {
  super.initState();

  _loadWidget();
}

_loadWidget() async {
  var _duration = Duration(seconds: splashDelay);
  return Timer(_duration, navigationPage);
}

void navigationPage() {
  Navigator.pushReplacement(context, MaterialPageRoute(builder: (BuildContext context) => PageLogin()));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[600],
    body: InkWell(
      child: Stack(
        fit: StackFit.expand,
        children: <Widget>[
          Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: <Widget>[
              Expanded(
                flex: 7,
                child: Container(
                    child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    Image.asset(
                      'assets/images/biu.png',
                      height: 150,
                      width: 150,
                    ),
                    Text("BIU - Human Resource", style: TextStyle(
                          color: Colors.black,
                          fontSize: 20,
                          fontWeight: FontWeight.bold),),
                    Padding(
                      padding: const EdgeInsets.only(top: 10.0),
                    ),
                  ],
                )),
              ),
              Expanded(
                child: Column(
                  children: <Widget>[
                    CircularProgressIndicator(),
                    Container(
                      height: 10,
                    ),
                    Row(
                        mainAxisAlignment: MainAxisAlignment.spaceAround,
                        children: <Widget>[
                          Spacer(),
                          Text(_versionName, style: TextStyle(
                          color: Colors.white,
                          fontSize: 12,
                          fontWeight: FontWeight.bold),),
                          Spacer(
                            flex: 4,
                          ),
                          Text('Bina Insani University', style: TextStyle(
                          color: Colors.white,
                          fontSize: 12,
                            fontWeight: FontWeight.bold),),
                            Spacer(),
                          ])
                    ],
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

Main.dart

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

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

void configLoading() {
  EasyLoading.instance
    ..displayDuration = const Duration(milliseconds: 2000)
    ..indicatorType = EasyLoadingIndicatorType.fadingCircle
    ..loadingStyle = EasyLoadingStyle.dark
    ..indicatorSize = 45.0
    ..radius = 10.0
    ..progressColor = Colors.yellow
    ..backgroundColor = Colors.green
    ..indicatorColor = Colors.yellow
    ..textColor = Colors.yellow
    ..maskColor = Colors.blue.withOpacity(0.5)
    ..userInteractions = true
    ..dismissOnTap = false;
    //..customAnimation = CustomAnimation();
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'BIU - HRIS',
      home: SplashScreen(),
      debugShowCheckedModeBanner: false,
      builder: EasyLoading.init(),
    );
  }
}
```

login_page.dart

```
import 'dart:convert';
import 'package:biu_project/home_page.dart';
import 'package:biu_project/widgets/dialogs.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
import 'navigation.dart';

class PageLogin extends StatefulWidget {
  const PageLogin({Key? key}) : super(key: key);
  @override
  _PageLoginState createState() => _PageLoginState();
}

class HeadClipper extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    Path path = Path();
    path.lineTo(0, size.height);
    path.quadraticBezierTo(
        size.width / 4, size.height - 40, size.width / 2, size.height - 20);
    path.quadraticBezierTo(
        3 / 4 * size.width, size.height, size.width, size.height - 30);
    path.lineTo(size.width, 0);

    return path;
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) => false;
}

class _PageLoginState extends State<PageLogin> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  var txtEditUsername = TextEditingController();
  var txtEditPwd = TextEditingController();

  Widget inputUsername() {
    return TextFormField(
        cursorColor: Colors.white,
        keyboardType: TextInputType.text,
        autofocus: false,
        // validator: (email) => email != null && !EmailValidator.validate(email)
        //     ? 'Masukkan email yang valid'
        //     : null,
        validator: (String? arg) {
        if (arg == null || arg.isEmpty) {
          return 'Username can not blank';
        } else {
          return null;
        }
      },
        controller: txtEditUsername,
        onSaved: (String? val) {
          txtEditUsername.text = val!;
        },
        decoration: InputDecoration(
          hintText: 'Username',
          hintStyle: const TextStyle(color: Colors.white),
          labelText: "Enter an Username",
          labelStyle: const TextStyle(color: Colors.white),
          prefixIcon: const Icon(
            Icons.person,
            color: Colors.white,
          ),
          fillColor: Colors.white,
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(25.0),
            borderSide: const BorderSide(
              color: Colors.white,
            ),
          ),
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(25.0),
            borderSide: const BorderSide(
              color: Colors.white,
              width: 2.0,
            ),
          ),
        ),
        style: const TextStyle(fontSize: 16.0, color: Colors.white));
  }

  Widget inputPassword() {
    return TextFormField(
      cursorColor: Colors.white,
      keyboardType: TextInputType.text,
      autofocus: false,
      obscureText: true, //make decript inputan
      validator: (String? arg) {
        if (arg == null || arg.isEmpty) {
          return 'Password can not blank';
        } else {
          return null;
        }
      },
      controller: txtEditPwd,
      onSaved: (String? val) {
        txtEditPwd.text = val!;
      },
      decoration: InputDecoration(
        hintText: 'Password',
        hintStyle: const TextStyle(color: Colors.white),
        labelText: "Enter a Password",
        labelStyle: const TextStyle(color: Colors.white),
        prefixIcon: const Icon(
          Icons.lock_outline,
          color: Colors.white,
        ),
        fillColor: Colors.white,
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(25.0),
          borderSide: const BorderSide(
            color: Colors.white,
          ),
        ),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(25.0),
          borderSide: const BorderSide(
            color: Colors.white,
            width: 2.0,
          ),
        ),
      ),
      style: const TextStyle(fontSize: 16.0, color: Colors.white),
    );
  }

  void _validateInputs() {
    if (_formKey.currentState!.validate()) {

      //If all data are correct then save data to out variables
      _formKey.currentState!.save();
      doLogin(txtEditUsername.text, txtEditPwd.text);

    }
  }

  doLogin(username, password) async {

    final GlobalKey<State> _keyLoader = GlobalKey<State>();
    Dialogs.loading(context, _keyLoader, "Proses ...");

    try {
      final response = await http.post(
          Uri.parse("https://nscis.nsctechnology.com/index.php?r=auth/login"),
          body: {
            "username": username,
            "password": password,

          }).then((value) {
            var data = jsonDecode(value.body);
            Navigator.of(_keyLoader.currentContext!, rootNavigator: false).pop();
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                  content: Text(
                data['message'],
                style: const TextStyle(fontSize: 16),
              )),
            );

            if (data['success'] == true) {
              saveSession(username);
            }
          });


    } catch (e) {
      Navigator.of(_keyLoader.currentContext!, rootNavigator: false).pop();
      Dialogs.popUp(context, '$e');
      debugPrint('$e');
    }
  }

  saveSession(String username) async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    await pref.setString("username", username);
    await pref.setBool("is_login", true);

    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => HomePage(),
      ),
      (route) => false,
    );
  }

  void ceckLogin() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin) {
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => HomePage(),
        ),
        (route) => false,
      );
    }
  }

  @override
  void initState() {
    ceckLogin();
    super.initState();
  }

  @override
  dispose() {
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
  //backgroundColor: Colors.blueAccent,
  backgroundColor: Colors.grey[600],
  body: Container(
    margin: const EdgeInsets.all(0),
    decoration: const BoxDecoration(
        gradient: LinearGradient(
      begin: Alignment.topCenter,
      end: Alignment.bottomCenter,
      colors: [Colors.grey, Color.fromARGB(255, 5, 6, 6)],
    )),
    child: Form(
      key: _formKey,
      child: ListView(
        children: [
          ClipPath(
            clipper: HeadClipper(),
            child: Container(
              margin: const EdgeInsets.all(0),
              width: double.infinity,
              height: 180,
              decoration: const BoxDecoration(
                color: Colors.white24,
                image: DecorationImage(
                  image: AssetImage('assets/images/biu.png'),
                ),
              ),
            ),
          ),
          const SizedBox(height: 20),
          Container(
            padding: const EdgeInsets.all(8.0),
            alignment: Alignment.center,
            child: const Text(
              "LOGIN APP BIU-HRIS",
              style: TextStyle(
                  color: Colors.white,
                  fontSize: 22,
                  fontWeight: FontWeight.bold),
            ),
          ),
          Container(
              padding:
                  const EdgeInsets.only(top: 35.0, left: 20.0, right: 20.0),
              child: Column(
                children: <Widget>[
                  inputUsername(),
                  const SizedBox(height: 20.0),
                  inputPassword(),
                  const SizedBox(height: 5.0),
                ],
              )),
          Container(
            padding:
                const EdgeInsets.only(top: 35.0, left: 20.0, right: 20.0),
            child: ElevatedButton.icon(
                style: ElevatedButton.styleFrom(
                    primary: Colors.amber,
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(20.0),
                      side: const BorderSide(color: Colors.white),
                    ),
                    elevation: 10,
                    minimumSize: const Size(200, 58)),
                onPressed: () => _validateInputs(),
                icon: const Icon(Icons.arrow_right_alt),
                label: const Text(
                  "LOG IN",
                  style:
                      TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                      )),
                )
              ],
            ),
          ),
        ),
      );
    }
  }
```

home_page.dart

```
import 'dart:async';
import 'dart:convert';
import 'package:biu_project/report.dart';
import 'package:biu_project/scan/report_visit.dart';
import 'package:biu_project/scan/scanner.dart';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:qr_code_scanner/src/types/barcode.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'invitation/invite.dart';
import 'login_page.dart';
import 'package:http/http.dart' as http;
import 'clocking.dart';


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

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  String photo = "";
  String username = "";
  String firstName = "";
  String lastName = "";
  List precenseUser = [];
  Timer? _timer;

  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    setState(() {
      preferences.remove("is_login");
      preferences.remove("username");
    });

    Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
void initState() {
  super.initState();
  getPref();
  _getData();
  _getPrecence();
  EasyLoading.addStatusCallback((status) {
    print('EasyLoading Status $status');
    if (status == EasyLoadingStatus.dismiss) {
      _timer?.cancel();
    }
  });
  //EasyLoading.showSuccess('Use in initState');
}

@override
dispose() {
  super.dispose();
}



Future _getData() async {
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    print(username);
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id

        "https://nscis.nsctechnology.com/index.php?r=user/view-api&id=" + username));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        photo = data["photo"];
        firstName = data["first_name"];
        lastName = data["last_name"];
      });
    }
  } catch (e) {
    print(e);
  }
}

Future _getPrecence() async {
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        "https://nscis.nsctechnology.com/index.php?r=precense/user-api&id='${username}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        precenseUser = data;
      });
    }
  } catch (e) {
    print(e);
  }
}

showImage(String image) {

        if (image.length % 4 > 0) {
          image += '=' * (4 - image .length % 4); // as suggested by Albert221
      }
  return Image.memory(
    base64Decode(image),
  );
}

final List<String> _listItem = [
  'assets/images/two.jpg',
  'assets/images/three.jpg',
  'assets/images/four.jpg',
  'assets/images/five.jpg',
  'assets/images/one.jpg',
  'assets/images/two.jpg',
  'assets/images/three.jpg',
  'assets/images/four.jpg',
  'assets/images/five.jpg',
];

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[600],
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    body:  SafeArea(
      child: Container(
        padding: EdgeInsets.all(20.0),
        child: Column(
children: <Widget>[
  Container(
    width: double.infinity,
    height: 180,
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        image: DecorationImage(
            image: AssetImage('assets/images/background1.jpg'),
            fit: BoxFit.cover)),
    child: Container(
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
          gradient:
              LinearGradient(begin: Alignment.bottomRight, colors: [
            Colors.black.withOpacity(.4),
            Colors.black.withOpacity(.2),
          ])),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: <Widget>[
          Text(
            "Hi, ${firstName} ${lastName}",
            style: TextStyle(
                color: Colors.white,
                fontSize: 18,
                fontWeight: FontWeight.bold),
          ),
          Text(
            "Welcome to N-HR Mobile",
            style: TextStyle(color: Colors.white, fontSize: 14),
          ),
          SizedBox(
            height: 10,
          ),
          Container(
            height: 50,
            margin: EdgeInsets.symmetric(horizontal: 40),
            child: Center(
              child: ElevatedButton.icon(
                  style: ElevatedButton.styleFrom(
                      primary: Colors.amber,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(20.0),
                        side: const BorderSide(color: Colors.white),
                      ),
                      elevation: 10,
                      minimumSize: const Size(200, 58)),

                  onPressed: () async {
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (_) => Clocking()));
                    setState(() {});
                  },
                  icon: const Icon(Icons.arrow_right_alt),
                  label: const Text(
                    "Clocking",
                    style: TextStyle(
                        fontSize: 18, fontWeight: FontWeight.bold),
                  )),

            ),
          ),
          SizedBox(
            height: 30,
          ),
        ],
      ),
  ),
),
SizedBox(
  height: 10,
),
 SizedBox(
  height: 30,
  child: Container(
    child: Column(
      children: [
        Text("MENU", textAlign: TextAlign.right,style: TextStyle(
                        fontSize: 18, fontWeight: FontWeight.bold, color: Colors.white))
      ],
    ),
  ),
 ),
Expanded(

  child: Container(
    child: GridView(
      shrinkWrap: true,
       gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 3,
        crossAxisSpacing: 10,
        mainAxisSpacing: 10),
        children: [
          GestureDetector(
            onTap: () async {
            Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (_) => MainInvite()));
                  setState(() {});
          },
            child: Container(
              decoration:BoxDecoration(
                borderRadius: BorderRadius.circular(10),
                color: Colors.grey,
                ),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,

                children: [
                  Icon(Icons.insert_invitation, size: 50, color: Colors.amber,),
                  SizedBox(
                    height: 3,
                  ),
                  Text("Invitation", style: TextStyle(
                        fontSize: 12),),
                ],
              ),
              ),
          ),
            GestureDetector(
              onTap: () async {
                Navigator.push(
                          context,
                          MaterialPageRoute(
                              builder: (_) => const QRViewExample()));
                      setState(() {});
              },

              child: Container(
              decoration:BoxDecoration(
                borderRadius: BorderRadius.circular(10),
                color: Colors.grey,
                ),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
          children: [
           Icon(Icons.qr_code_scanner, size: 50, color: Colors.amber,),
           SizedBox(
            height: 3,
           ),
           Text("Scanner", style: TextStyle(
              fontSize: 12),),
          ],
         ),
         ),
        ),
       GestureDetector(
        onTap: () async {
          Navigator.push(
               context,
               MaterialPageRoute(
                 builder: (_) => ReportVisit()));
             setState(() {});
         },
         child: Container(
         decoration:BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Colors.grey,
          ),
         child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
           Icon(Icons.file_copy_outlined, size: 50, color: Colors.amber,),
           SizedBox(
            height: 3,
           ),
           Text("Data Visit", style: TextStyle(
              fontSize: 12),),
          ],
         ),
         ),
        ),
        GestureDetector(
         onTap: () async {
          Navigator.push(
               context,
               MaterialPageRoute(
                 builder: (_) => Report()));
             setState(() {});
         },
         child: Container(
         decoration:BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Colors.grey,
          ),
         child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
           Icon(Icons.manage_history, size: 50, color: Colors.amber,),
           SizedBox(
            height: 3,
           ),
           Text("Attendance", style: TextStyle(
              fontSize: 12),),
          ],
         ),
         ),
        ),
      ],
    ),
   ),
  )
 ],
),
        ),
      ),
    );
  }
}

class CustomCard extends StatelessWidget {
  //ini adalah konstruktor, saat class dipanggil parameter konstruktor wajib diisi
  //parameter ini akan mengisi title dan gambar pada setiap card
  CustomCard({required this.title, required this.image});

  String title;
  String image;

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.only(top: 5),
      child: Card(
        //menambahkan bayangan
        elevation: 4,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              decoration: BoxDecoration(
                  image: DecorationImage(
                      image: AssetImage(
                        image,
                      ),
                      fit: BoxFit.fill)),
            ),
            Padding(
              padding: const EdgeInsets.all(5),
              child: Center(
                  child: Text(
                title,
                style: TextStyle(
                    fontWeight: FontWeight.bold, height: 2, fontSize: 14),
              )),
            )
          ],
        ),
      ),
    );
  }
}
```

Navigation.dart

```
import 'package:flutter/material.dart';
import 'clocking.dart';
import 'report.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'login_page.dart';
import 'home_page.dart';
import 'package:biu_project/invitation/invite.dart';

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

  @override
  _NavigationState createState() => _NavigationState();
}

class _NavigationState extends State<Navigation> {
  String username = "";
  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
  var islogin = pref.getBool("is_login");
  if (islogin != null && islogin == true) {
    setState(() {
      username = pref.getString("username")!;
    });
  } else {
    Navigator.of(context, rootNavigator: true).pop();
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => const PageLogin(),
      ),
      (route) => false,
    );
  }
}

logOut() async {
  SharedPreferences preferences = await SharedPreferences.getInstance();
  setState(() {
    preferences.remove("is_login");
    preferences.remove("username");
  });

  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
void initState() {
  getPref();
  super.initState();
}

@override
dispose() {
  super.dispose();
}

int _selectedIndex = 0;

static const List<Widget> _pages = <Widget>[
  HomePage(),
  Clocking(),
  MainInvite(),
  Report(),
];

void _onItemTapped(int index) {
  setState(() {
    _selectedIndex = index;
  });
}

@override
Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[600],
      body: Center(
        child: _pages.elementAt(_selectedIndex),
      ),
      bottomNavigationBar: BottomNavigationBar(
        backgroundColor: Colors.white10,
        elevation: 0,
        selectedFontSize: 14,
        selectedIconTheme: IconThemeData(color: Colors.amberAccent, size: 40),
        selectedItemColor: Colors.amberAccent,
        selectedLabelStyle: TextStyle(fontWeight: FontWeight.bold),
        unselectedIconTheme: IconThemeData(
          color: Colors.white,
        ),
        unselectedItemColor: Colors.white,
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: 'Home',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.punch_clock),
            label: 'Clocking',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.qr_code_scanner),
            label: 'Scan',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.insert_invitation),
            label: 'Invitation',
          ),
        ],
        currentIndex: _selectedIndex,
        onTap: _onItemTapped,
      ),
    );
  }
}
```

Clocking.dart

```
import 'package:flutter/material.dart';
import 'package:location/location.dart';
import 'dart:io';
import 'camera_page.dart';
import 'package:flutter_geocoder/geocoder.dart';
import 'package:intl/intl.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'login_page.dart';
import 'dart:convert';
import 'package:http/http.dart' as http;

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

  @override
  _ClockingState createState() => _ClockingState();
}

class _ClockingState extends State<Clocking> {
  File? imageFile;
  LatLng _initialcameraposition = LatLng(-6.2595697, 106.994477);
  GoogleMapController? _controller;
  Location _location = Location();
  Set<Marker> _markers = {};
  LocationData? _currentPosition;
```

```
String? _address, _dateTime;
String photo = "";
String id = "";
String username = "";
List precenseUser = [];
getPref() async {
  SharedPreferences pref = await SharedPreferences.getInstance();
  var islogin = pref.getBool("is_login");
  if (islogin != null && islogin == true) {
    setState(() {
      username = pref.getString("username")!;
    });
  } else {
    Navigator.of(context, rootNavigator: true).pop();
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => const PageLogin(),
      ),
      (route) => false,
    );
  }
}

logOut() async {
  SharedPreferences preferences = await SharedPreferences.getInstance();
  setState(() {
    preferences.remove("is_login");
    preferences.remove("username");
  });

  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
dispose() {
  super.dispose();
}

@override
void initState() {
  // TODO: implement initState
  super.initState();
  getLoc();
  getPref();
  _getData();
  _getPrecence();
}

Future _getData() async {
  print(username);
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id

        "https://nscis.nsctechnology.com/index.php?r=user/view-api&id='${username}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        photo = data["photo"];
        id = data["id"];
        //print(photo);
      });
    }
  } catch (e) {
    print(e);
  }
}

Future _getPrecence() async {
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        // "http://nscis.nsctechnology.com/index.php?r=precense/list"
        "https://nscis.nsctechnology.com/index.php?r=precense/user-api&id='${username}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        precenseUser = data;
      });
    }
  } catch (e) {
    print(e);
  }
}

showImage(String image) {

        if (image.length % 4 > 0) {
          image += '=' * (4 - image .length % 4); // as suggested by Albert221
      }
  return Image.memory(
    base64Decode(image),
  );
}

void _onMapCreated(GoogleMapController _Cntlr) {
  _controller = _Cntlr;
  _location.onLocationChanged.listen((l) {
    _controller!.animateCamera(
      CameraUpdate.newCameraPosition(
        CameraPosition(target: LatLng(l.latitude!, l.longitude!), zoom: 15),
      ),
    );
  print(l.latitude);
  print(l.longitude);
  _initialcameraposition = LatLng(l.latitude!, l.latitude!);
  setState(() {
   _markers.add(Marker(
     markerId: MarkerId('Home'),
     position: LatLng(l.latitude ?? 0.0, l.longitude ?? 0.0)));
  });
 });
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  backgroundColor: Colors.grey[600],
  appBar: AppBar(
   backgroundColor: Colors.transparent,
   elevation: 0,
   //leading: Icon(Icons.menu),
   title: Text("BIU"),
   actions: <Widget>[
    IconButton(
     icon: Icon(Icons.logout),
     onPressed: () {
      logOut();
     },
    ),
   ],
  ),
  body: SafeArea(
   child: Container(
    padding: EdgeInsets.all(5.0),
    child: Column(
     children: [
      Container(
       child: Container(
        margin: EdgeInsets.all(5),
        height: 300,
        color: Colors.grey[200],
        child: Stack(
         children: [
          GoogleMap(
           initialCameraPosition:
             CameraPosition(target: _initialcameraposition),
           mapType: MapType.normal,
           onMapCreated: _onMapCreated,
           myLocationEnabled: true,
           markers: _markers,
          ),
         ],
        ),
       ),
      ),
      SizedBox(
       height: 180,
       child: Container(
        margin: EdgeInsets.all(5),
        color: Colors.grey[600],
       child: precenseUser.isEmpty ? Center(child: Text('No data available for today, feel free to clock')) : Expanded(
         child:ListView.builder(
     itemCount: precenseUser.length,
     shrinkWrap: true,
     itemBuilder: (context, index) {
      return Container(
       width: MediaQuery.of(context).size.width,
       padding: EdgeInsets.symmetric(horizontal: 5.0, vertical: 5.0),
       child: Card(
        elevation: 5.0,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(2.0)),
        child: Container(
         width: MediaQuery.of(context).size.width,
         padding:
           EdgeInsets.symmetric(horizontal: 10.0, vertical: 10.0),
         child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
           Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
             Container(
              width: 55.0,
              height: 55.0,
              child: CircleAvatar(
               backgroundColor: Colors.green,
               backgroundImage: NetworkImage(
                 'https://nscis.nsctechnology.com/' + precenseUser[index]['photo']),
              ),
             ),
             SizedBox(
              width: 5.0,
             ),

            ],
           ),
           Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
             Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
               Text('Clocking'),
               Text(precenseUser[index]['time'],style: TextStyle(
                   color: Colors.black,
                   fontSize: 18.0,
                   fontWeight: FontWeight.bold)),

              ],
             ),
            ],
           ),
           Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
             Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
               Text('Date'),
               Text(precenseUser[index]['date'],style: TextStyle(
                   color: Colors.black,
                   fontSize: 18.0,
                   fontWeight: FontWeight.bold)),

              ],
             ),
            ],
           ),
           SizedBox(
              width: 5.0,
             ),
          ],
         ),
        ),
       ),
                   );
                 })),
                   ),
                     ),
                   Container(
                     padding: EdgeInsets.all(10.0),
                     child: Center(
                       child: ElevatedButton.icon(
                           style: ElevatedButton.styleFrom(
                               primary: Colors.amber,
                               shape: RoundedRectangleBorder(
                                 borderRadius: BorderRadius.circular(20.0),
                                 side: const BorderSide(color: Colors.white),
                               ),
                               elevation: 10,
                               minimumSize: const Size(200, 58)),
                           onPressed: () async {
                             imageFile = await Navigator.push<File>(
                                 context,
                                 MaterialPageRoute(
                                     builder: (_) => CameraPage(
                                           iniPosition: _initialcameraposition,
                                           address: _address,
                                           id_user: id,
                                         )));
                             setState(() {});
                           },
                           icon: const Icon(Icons.arrow_right_alt),
                           label: const Text(
                             "Clocking",
                             style: TextStyle(
                                 fontSize: 18, fontWeight: FontWeight.bold),
                           )),
                     ),
                   ),
                   Container(),
                 ],
               ),
             ),
           ),
         );
       }

       getLoc() async {
         bool _serviceEnabled;
         PermissionStatus _permissionGranted;
         _serviceEnabled = await _location.serviceEnabled();
         if (!_serviceEnabled) {
           _serviceEnabled = await _location.requestService();
           if (!_serviceEnabled) {
             return;
           }
         }
         _permissionGranted = await _location.hasPermission();
         if (_permissionGranted == PermissionStatus.denied) {
           _permissionGranted = await _location.requestPermission();
           if (_permissionGranted != PermissionStatus.granted) {
             return;
           }
         }
         _currentPosition = await _location.getLocation();
         _initialcameraposition = LatLng(
             _currentPosition!.latitude ?? 0.0, _currentPosition!.longitude ?? 0.0);
         _location.onLocationChanged.listen((LocationData currentLocation) {
           print("${currentLocation.longitude} : ${currentLocation.longitude}");
           setState(() {
             _currentPosition = currentLocation;
             _initialcameraposition = LatLng(_currentPosition!.latitude ?? 0.0,
                 _currentPosition!.longitude ?? 0.0);
               DateTime now = DateTime.now();
               _dateTime = DateFormat('EEE d MMM kk:mm:ss ').format(now);
               _getAddress(_currentPosition!.latitude ?? 0.0,
                       _currentPosition!.longitude ?? 0.0)
                   .then((value) {
                 setState(() {
                   _address = "${value.first.addressLine}";
                 });
               });
             });
           });
         }

         Future<List<Address>> _getAddress(double lat, double lang) async {
           final coordinates = Coordinates(lat, lang);
           List<Address> add =
               await Geocoder.local.findAddressesFromCoordinates(coordinates);
           return add;
         }
       }
```

camera_page.dart

```
import 'dart:ffi';
import 'dart:io';
import 'package:camera/camera.dart';
import 'package:flutter/material.dart';
import 'package:path_provider/path_provider.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'presence_page.dart';

class CameraPage extends StatefulWidget {
  LatLng? iniPosition;
  String? address;
  String? id_user;
  CameraPage({Key? key, @required this.iniPosition, this.address, this.id_user})
      : super(key: key);
  @override
  State<CameraPage> createState() =>
      _CameraPageState(initPosition: iniPosition, address: address, id_user: id_user);
}

class _CameraPageState extends State<CameraPage> {
  CameraController? controller;
  LatLng? initPosition;
  String? address;
  String? id_user;
  _CameraPageState({this.initPosition, this.address, this.id_user});
  Future<void> initializeCamera() async {
    var cameras = await availableCameras();
    controller = CameraController(cameras[1], ResolutionPreset.medium);
    await controller!.initialize();
  }

  @override
  Void? dispose() {
    controller!.dispose();
    super.dispose();
  }

  Future<File?> takePicture() async {
    Directory root = await getTemporaryDirectory();
    String directoryPath = '${root.path}/Guided_Camera';
    await Directory(directoryPath).create(recursive: true);
    String filePath = '$directoryPath/${DateTime.now()}.jpg';
    try {
      XFile picture = await controller!.takePicture();
      picture.saveTo(filePath);
    } catch (e) {
```

```
  return null;
 }
 return File(filePath);
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  resizeToAvoidBottomInset: false,
  backgroundColor: Colors.grey,
  body: SingleChildScrollView(
   child: FutureBuilder(
     future: initializeCamera(),
     builder: (_, snapshot) => (snapshot.connectionState ==
         ConnectionState.done)
       ? Stack(
         children: [
          Column(
           children: [
            SizedBox(
             width: MediaQuery.of(context).size.width,
             height: MediaQuery.of(context).size.width *
               controller!.value.aspectRatio,
             child: CameraPreview(controller!),
            ),
            Container(
             width: 70,
             height: 80,
             margin: EdgeInsets.only(top: 10),
             child: SingleChildScrollView(
              child: RaisedButton(
               onPressed: () async {
                if (!controller!.value.isTakingPicture) {
                 File? result = await takePicture();
                 Navigator.push(
                   context,
                   new MaterialPageRoute(
                    builder: ((context) => new PresencePage(
                       imagefile: result,
                       initPosition: initPosition,
                       address: address,
                       id_user: id_user,

                      )),
                   ));
                }
               },
               shape: CircleBorder(),
               color: Colors.blue,
              ),
             ),
            ),
           ],
          ),
          SizedBox(
           width: MediaQuery.of(context).size.width,
           height: MediaQuery.of(context).size.width *
             controller!.value.aspectRatio,
          ),
         ],
        )
       : Center(
         child: SizedBox(
          height: 20,
          width: 20,
          child: CircularProgressIndicator(),
         ),
        )),
  ),
   );
  }
}
```

presence_page.dart

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'dart:convert';
import 'dart:typed_data';
import 'package:http/http.dart' as http;
import 'clocking.dart';

class PresencePage extends StatefulWidget {
  File? imagefile;
  LatLng? initPosition;
  String? address;
  String? id_user;
  PresencePage(
      {Key? key,
      @required this.imagefile,
      this.initPosition,
      this.address,
      this.id_user})
      : super(key: key);
  @override
  State<PresencePage> createState() => _PresencePageState(
      imageFile: imagefile,
      initposition: initPosition,
      addr: address,
      ID: id_user);
}

class _PresencePageState extends State<PresencePage> {
  final _formKey = GlobalKey<FormState>();
  File? imageFile;
  LatLng? initposition;
  String? addr;
  String? ID;
  String? base64Image;
  _PresencePageState({this.imageFile, this.initposition, this.addr, this.ID});

  var IDUser = TextEditingController();
  var Latitude = TextEditingController();
  var Longtitude = TextEditingController();
  var Location = TextEditingController();
  var Photo = TextEditingController();

  @override
  void initState() {
    super.initState();
    //in first time, this method will be executed
    _getData();
  }


  Future _getData() async {
        try {



              Uint8List imagebytes = await imageFile!.readAsBytes(); //convert to bytes
              String base64Image = base64.encode(imagebytes); //convert bytes to base64 string

              String fileName = imageFile!.path.split('/').last;
              print(fileName);

            setState(() {
```

```
      IDUser = TextEditingController(text: '${ID}');
      Latitude = TextEditingController(text: '${initposition!.latitude}');
      Longtitude = TextEditingController(text: '${initposition!.longitude}');
      Location = TextEditingController(text: '${addr}');
      Photo = TextEditingController(text: '${base64Image}');
     });

   } catch (e) {
    print(e);
   }
  }

  Future _onUpdate(context) async {
   try {
    var postUri = Uri.parse("https://nscis.nsctechnology.com/index.php?r=precense/create-api");

    http.MultipartRequest request = new http.MultipartRequest("POST", postUri);

    http.MultipartFile multipartFile = await http.MultipartFile.fromPath(
      'photo_precense', imageFile!.path);

    request.fields['user_id'] = IDUser.text;
    request.fields['latitude'] = Latitude.text;
    request.fields['longitude'] = Longtitude.text;
    request.fields['location'] = Location.text;
    request.files.add(multipartFile);

    EasyLoading.show(status: 'Uploading Image...');

    var response = await request.send();
    if (response.statusCode == 200) {
     EasyLoading.showSuccess('Image Upload Success!');
     print("Upload done");
     // Navigator.of(context)
     //   .pushNamedAndRemoveUntil('/', (Route<dynamic> route) => false);

     Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) =>
     Clocking()), (Route<dynamic> route) => false);

     EasyLoading.dismiss();
    }
    setState(() {

    });
   } catch (e) {
    print(e);
   }
  }

@override
Widget build(BuildContext context) {
 return Scaffold(
  backgroundColor: Colors.grey[600],
  appBar: AppBar(
   backgroundColor: Colors.transparent,
   elevation: 0,
   //leading: Icon(Icons.menu),
   title: Text("BIU"),
  ),
  body: Center(
   child: Column(
    children: [
     Container(
      margin: EdgeInsets.only(top: 20, bottom: 10),
      width: 300,
      height: 400,
      color: Colors.grey[200],
      child: (imageFile != null) ? Image.file(imageFile!) : SizedBox(),
     ),
//Text('${imageFile}'),
Container(
child:  Form(
key: _formKey,
child: Container(
 padding: EdgeInsets.all(20.0),
 child: SingleChildScrollView(
  child: Column(
   crossAxisAlignment: CrossAxisAlignment.start,
   children: [
    SizedBox(height: 5),
    Visibility(
     child: TextFormField(
       readOnly: true,
       enabled: true,
       controller: IDUser,
       decoration: InputDecoration(
         hintText: "Type Note Name",
         border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
         ),
         fillColor: Colors.white,
         filled: true),
       style: const TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 16,
       ),
       validator: (value) {
        if (value!.isEmpty) {
         return 'Note Title is Required!';
        }
        return null;
       },
      ),
      visible: false,
    ),

    SizedBox(height: 5),
    Visibility(
     child: TextFormField(
      controller: Latitude,
      decoration: InputDecoration(
        hintText: "Type Note Name",
        border: OutlineInputBorder(
         borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true),
      style: const TextStyle(
       fontWeight: FontWeight.bold,
       fontSize: 16,
      ),
      validator: (value) {
       if (value!.isEmpty) {
        return 'Note Title is Required!';
       }
       return null;
      },
     ),
     visible: false,
    ),
    SizedBox(height: 5),
    Visibility(
     child: TextFormField(
      controller: Longtitude,
      decoration: InputDecoration(
        hintText: "Type Note Name",
        border: OutlineInputBorder(
         borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true),
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 16,
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Note Title is Required!';
      }
      return null;
    },
  ),
  visible: false,
),
Visibility(
  child: TextFormField(
    controller: Location,
    keyboardType: TextInputType.multiline,
    minLines: 5,
    maxLines: null,
    decoration: InputDecoration(
        hintText: 'Type Note Address',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true),
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 16,
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Note Content is Required!';
      }
      return null;
    },
  ),
  visible: false,
),
Visibility(
  child: TextFormField(
    controller: Photo,
    keyboardType: TextInputType.multiline,
    minLines: 5,
    maxLines: null,
    decoration: InputDecoration(
        hintText: 'Type Note Address',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true),
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 16,
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Note Content is Required!';
      }
      return null;
    },
  ),
  visible: false,
),
SizedBox(height: 10),
                    ElevatedButton.icon(
                    style: ElevatedButton.styleFrom(
                        primary: Colors.amber,
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(20.0),
                          side: const BorderSide(color: Colors.white),
                        ),
                        elevation: 10,
                        minimumSize: const Size(200, 58)),
                     onPressed: () {
                        //validate
                        if (_formKey.currentState!.validate()) {
                          //send data to database with this method
                          _onUpdate(context);
                         }
                       },
                    icon: const Icon(Icons.save),
                    label: const Text(
                      "Use Picture",
                      style:
                          TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                    )),
                  ],
                ),
              ),
            ),
          ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Report.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'login_page.dart';
import 'package:http/http.dart' as http;

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

  @override
  _ReportState createState() => _ReportState();
}

class _ReportState extends State<Report> {
  String photo = "";
  String username = "";
  List precenseUser = [];

  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
      (route) => false,
    );
  }
}

logOut() async {
  SharedPreferences preferences = await SharedPreferences.getInstance();
  setState(() {
    preferences.remove("is_login");
    preferences.remove("username");
  });

  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  _getData();
  getPref();
  _getPrecence();
}

Future _getData() async {
  print(username);
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id

        "https://nscis.nsctechnology.com/index.php?r=user/view-api&id='${username}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        photo = data["photo"];
        print(photo);
      });
    }
  } catch (e) {
    print(e);
  }
}

Future _getPrecence() async {
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        "https://nscis.nsctechnology.com/index.php?r=precense/list-api&id='${username}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        precenseUser = data;
      });
    }
  } catch (e) {
    print(e);
  }
}

showImage(String image) {
  if (image.length % 4 > 0) {
    image += '=' * (4 - image.length % 4); // as suggested by Albert221
  }
  return Image.memory(
    base64Decode(image),
    // width: 100,
    // height: 100,
    fit: BoxFit.fill,
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[600],
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    body: Container(
      color: Colors.grey[600],
      child: ListView.builder(
          itemCount: precenseUser.length,
          shrinkWrap: true,
          itemBuilder: (context, index) {
            return Container(
              width: MediaQuery.of(context).size.width,
              padding: EdgeInsets.symmetric(horizontal: 5.0, vertical: 5.0),
              child: Card(
                elevation: 5.0,
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(2.0)),
                child: Container(
                  width: MediaQuery.of(context).size.width,
              padding:
                  EdgeInsets.symmetric(horizontal: 10.0, vertical: 10.0),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                  Row(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.start,
                    children: [
                      Container(
                        width: 55.0,
                        height: 55.0,
                        child: CircleAvatar(
                          backgroundColor: Colors.green,
                          backgroundImage: NetworkImage(
                              'https://nscis.nsctechnology.com/' + precenseUser[index]['photo']),
                        ),
                      ),
                      SizedBox(
                        width: 5.0,
                      ),

                    ],
                  ),
                  Row(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.start,
                    children: [
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text('Clocking'),
                          Text(precenseUser[index]['time'],style: TextStyle(
                                  color: Colors.black,
                                  fontSize: 18.0,
                                  fontWeight: FontWeight.bold)),

                        ],
                      ),
                    ],
                  ),
                  Row(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.start,
                    children: [
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text('Date'),
                          Text(precenseUser[index]['date'],style: TextStyle(
                                  color: Colors.black,
                                  fontSize: 18.0,
                                  fontWeight: FontWeight.bold)),

                        ],
                      ),
                    ],
                  ),
                  SizedBox(
                        width: 5.0,
                      ),
                ],
              ),
            ),
          ),
        );
      }),
),
    );
  }
}

class CustomCard extends StatelessWidget {
  //ini adalah konstruktor, saat class dipanggil parameter konstruktor wajib diisi
  //parameter ini akan mengisi title dan gambar pada setiap card
  CustomCard({required this.title, required this.image});

  String title;
  String image;

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.only(top: 5),
      child: Card(
        //menambahkan bayangan
        elevation: 4,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              height: 110,
              decoration: BoxDecoration(
                  image: DecorationImage(
                      image: AssetImage(
                        image,
                      ),
                      fit: BoxFit.cover)),
            ),
            Padding(
              padding: const EdgeInsets.all(5),
              child: Center(
                  child: Text(
                title,
                style: TextStyle(
                    fontWeight: FontWeight.bold, height: 2, fontSize: 14),
              )),
            )
          ],
        ),
      ),
    );
  }
}
```

widget/dialogs.dart

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

class Dialogs {
  static Future<void> loading(
      BuildContext context, GlobalKey key, String text) async {
    return showDialog<void>(
        context: context,
        barrierDismissible: false,
        builder: (BuildContext context) {
          return WillPopScope(
              onWillPop: () async => false,
              child: SimpleDialog(
                  key: key,
                  backgroundColor: Colors.black54,
                  children: <Widget>[
                    Center(
                      child: Column(children: [
                        const CircularProgressIndicator(
                          color: Colors.white,
                        ),
                        const SizedBox(
                          height: 10,
                        ),
                        Text(
                          text,
                          style: const TextStyle(color: Colors.white),
                        )
                      ]),
                    )
                  ]));
        });
  }

  static Future<void> popUp(BuildContext context, String text) async {
    return showDialog<void>(
      context: context, barrierDismissible: false, // user must tap button!
      builder: (BuildContext context) {
        return AlertDialog(
          title: const Text('Informasi'),
          content: SingleChildScrollView(
            child: ListBody(
              children: [
                Text(text),
              ],
            ),
          ),
          actions: [
            TextButton(
              child: const Text('OK'),
              onPressed: () {
                Navigator.of(context).pop();
              },
            ),
          ],
        );
      },
    );
  }
}
```

invitation/invitation.dart

```
import 'dart:convert';
import 'dart:typed_data';
import 'package:flutter/material.dart';
import 'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:http/http.dart' as http;
import 'package:biu_project/invitation/view.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../login_page.dart';
import 'add.dart';

class MainInvite extends StatefulWidget {
  //const MainInvite({Key? key}) : super(key: key);
  const MainInvite(
      {Key? key})
      : super(key: key);
  @override
  MainInviteState createState() => MainInviteState();
}

class MainInviteState extends State<MainInvite> {
  //make list variable to accomodate all data from database
  String id = "";
  String username = "";
  List _get = [];
  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    setState(() {
      preferences.remove("is_login");
      preferences.remove("username");
    });

    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => const PageLogin(),
      ),
      (route) => false,
    );

    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
          content: Text(
        "Berhasil logout",
        style: TextStyle(fontSize: 16),
      )),
    );
  }

  //make different color to different card
  final _lightColors = [
    Colors.amber.shade300,
    Colors.lightGreen.shade300,
    Colors.lightBlue.shade300,
    Colors.orange.shade300,
    Colors.pinkAccent.shade100,
    Colors.tealAccent.shade100
  ];

  @override
  dispose() {
    super.dispose();
  }

  @override
  void initState() {
    super.initState();
    //in first time, this method will be executed
    getPref();
    _getData();
  }

  Future _getData() async {
    print(username);
    try {
     SharedPreferences pref = await SharedPreferences.getInstance();
      var islogin = pref.getBool("is_login");
      setState(() {
        username = pref.getString("username")!;
      });
      print(username);
      final response = await http.get(Uri.parse(
          //you have to take the ip address of your computer.
          //because using localhost will cause an error
          "https://nscis.nsctechnology.com/index.php?r=t-invitation/invitation-api&id=" + username));

      // if response successful
      if (response.statusCode == 200) {
        final data = jsonDecode(response.body);

        // entry data to variabel list _get
        setState(() {
          _get = data;
        });
      }
    } catch (e) {
      print(e);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[600],
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0,
        //leading: Icon(Icons.menu),
        title: Text("BIU"),
        actions: <Widget>[
          IconButton(
            icon: Icon(Icons.logout),
            onPressed: () {
              logOut();
            },
          ),
        ],
      ),
      //if not equal to 0 show data
      //else show text "no data available"
      body: _get.length != 0
          //we use masonry grid to make masonry card style
          ? MasonryGridView.count(
              crossAxisCount: 2,
              itemCount: _get.length,
              itemBuilder: (context, index) {
                return GestureDetector(
                  onTap: () {
                    Navigator.push(
                        context,
                        //routing into edit page
                        //we pass the id note
                        MaterialPageRoute(
                            builder: (context) => View(
                                  id: _get[index]['id_invitation'],
                                )));
                  },
                  child: Card(
                    //make random color to eveery card
                    color: _lightColors[index % _lightColors.length],
                    child: Container(
                      //make 2 different height
                      // constraints:
                      //     BoxConstraints(minHeight: (index % 2 + 1) * 85),
                      padding: EdgeInsets.all(15),
                      child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Text(
                          'Date: ${_get[index]['schedule_date']}',
                          style: TextStyle(color: Colors.black),
                        ),
                        Text(
                          'Time: ${_get[index]['time_schedule']}',
                          style: TextStyle(color: Colors.black),
                        ),
                        SizedBox(height: 8),
                        Text(
                          'Invitation Code: ${_get[index]['invitation_code']}',
                          style: TextStyle(
                            color: Colors.black,
                            fontSize: 14,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        Text(
                          'Status: ${_get[index]['status']}',
                          style: TextStyle(
                            color: Colors.black,
                            fontSize: 14,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        Text(
                          'Desc: ${_get[index]['description']}',
                          style: TextStyle(
                            color: Colors.black,
                            fontSize: 14,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                      ],
                    ),
                  ),
                ),
              );
            },
          )
        : Center(
            child: Text(
              "No Data Available",
              style: TextStyle(
                // color: Colors.white,
                fontSize: 20,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
    floatingActionButton: Padding(
      padding: const EdgeInsets.only(bottom: 50.0),
      child: FloatingActionButton.extended(
        backgroundColor: Colors.black,
        elevation: 4.0,
        icon: const Icon(Icons.add_rounded),
        label: const Text('Add Guest'),
        // onPressed: () => null,
        onPressed: () {
            Navigator.push(
                context,
                //routing into add page
                MaterialPageRoute(builder: (context) => Add()));
          },
      ),
    ),

  );
```

```
  }
}
```

invitation/add.dart

```
import 'dart:convert';
import 'package:biu_project/home_page.dart';
import 'package:biu_project/invitation/invite.dart';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';
import '../login_page.dart';

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

  @override
  State<Add> createState() => _AddState();
}

class _AddState extends State<Add> {
  final _formKey = GlobalKey<FormState>();
  String id = "";
  String username = "";
  //inisialize field
  var desc = TextEditingController();
  var IDUser = TextEditingController();
  final _dateC = TextEditingController();
  final _timeC = TextEditingController();

  ///Date
  DateTime selected = DateTime.now();
  DateTime initial = DateTime(2000);
  DateTime last = DateTime(2025);

  ///Time
  TimeOfDay timeOfDay = TimeOfDay.now();

  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    setState(() {
      preferences.remove("is_login");
      preferences.remove("username");
    });

    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
dispose() {
  super.dispose();
}

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  getPref();
  _getData();
}

Future _getData() async {
  print(username);
  try {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id

        "https://nscis.nsctechnology.com/index.php?r=user/view-api&id=" + username));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
         id = data["id"];
        //  print(id);
         //IDUser = TextEditingController(text: id );

      });
        // print(id);
    }
  } catch (e) {
    print(e);
  }
}

Future _onSubmit() async {

  try {
    EasyLoading.show(status: 'Saving...');
    return await http.post(
      Uri.parse(
          "https://nscis.nsctechnology.com/index.php?r=t-invitation/create-api"),
      body: {
        "user_id": id,
        "schedule_date": _dateC.text,
        "time_schedule": _timeC.text,
        "description": desc.text,
      },

    ).then((value) {
      //print message after insert to database
      //you can improve this message with alert dialog

      EasyLoading.showSuccess('data successfully saved');
      var data = jsonDecode(value.body);
      print(data["message"]);

      Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) =>
          MainInvite()), (Route<dynamic> route) => false);

          EasyLoading.dismiss();
    });
  } catch (e) {
    print(e);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: false,
    backgroundColor: Colors.grey[600],
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      //leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    body: Form(
      key: _formKey,
      child: Container(
        color: Colors.white,
        padding: EdgeInsets.all(20.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            TextFormField(
              controller: _dateC,
              decoration: const InputDecoration(
                  labelText: 'Schedule Date',
                  border: OutlineInputBorder(),
                  // fillColor: Colors.white,
                  // filled: true
                  ),
            ),
            ElevatedButton(
                onPressed: () => displayDatePicker(context),
                child: const Text("Pick Date")),
            const SizedBox(height: 10),
            TextFormField(
              controller: _timeC,
              decoration: const InputDecoration(
                  labelText: 'Time',
                  border: OutlineInputBorder(),
                  // fillColor: Colors.white,
                  // filled: true
                  ),
            ),
            ElevatedButton(
                onPressed: () => displayTimePicker(context),
                child: const Text("Pick Time")),
            SizedBox(height: 10),
            TextFormField(
              controller: desc,
              keyboardType: TextInputType.multiline,
              minLines: 5,
              maxLines: null,
              decoration: InputDecoration(
                  labelText: 'Description',
                  border: OutlineInputBorder(),
                  //hintText: ' Entry your Address',
                  // border: OutlineInputBorder(
                  //   borderRadius: BorderRadius.circular(15.0),
                  // ),
                  // fillColor: Colors.white,
                  // filled: true
                  ),
              // style: const TextStyle(
              //   fontWeight: FontWeight.bold,
              //   fontSize: 16,
              // ),
              validator: (value) {
                if (value!.isEmpty) {
                  return 'Note Address is Required!';
                }
                return null;
              },
            ),
            SizedBox(height: 15),
            ElevatedButton(
              style: ElevatedButton.styleFrom(
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(15),
                  side: const BorderSide(color: Colors.white),
                ),
                elevation: 10,
                minimumSize: const Size(200, 45),
              ),

              child: Text(
                "Submit",
                style: TextStyle(color: Colors.white),

              ),

              onPressed: () {
                //validate
                if (_formKey.currentState!.validate()) {
                  //send data to database with this method
                  _onSubmit();
                }
              },
            )
          ],
        ),
      ),
    ),
  );
}

Future displayDatePicker(BuildContext context) async {
  var date = await showDatePicker(
    context: context,
    initialDate: selected,
    firstDate: initial,
    lastDate: last,
    );

    if (date != null) {
      setState(() {
        _dateC.text = date.toLocal().toString().split(" ")[0];
      });
    }
  }

  Future displayTimePicker(BuildContext context) async {
    var time = await showTimePicker(context: context, initialTime: timeOfDay);

    if (time != null) {
      setState(() {
        _timeC.text = "${time.hour}:${time.minute}";
      });
    }
  }
}
```

View.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:share_plus/share_plus.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../login_page.dart';

class View extends StatefulWidget {
  View({required this.id});

  String id;

  @override
  State<View> createState() => _ViewState();
}

class _ViewState extends State<View> {
  final _formKey = GlobalKey<FormState>();

  //inisialize field
  String username = "";
  var code = TextEditingController();
  var address = TextEditingController();
  String? schedule_date, invitation_code, time_schedule, status, description;

  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    setState(() {
```

```
    preferences.remove("is_login");
    preferences.remove("username");
  });

  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

@override
dispose() {
  super.dispose();
}

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  getPref();
  _getData();
}

//Http to get detail data
Future _getData() async {
  try {
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id
        "https://nscis.nsctechnology.com/index.php?r=t-invitation/view-api&id='${widget.id}'"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        //code = TextEditingController(text: data['invitation_code']);
        invitation_code = data['invitation_code'] !;
        schedule_date = data['schedule_date'];
        time_schedule = data['time_schedule'];
        status = data['status'];
        description = data['description'];
      });
    }
  } catch (e) {
    print(e);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Color.fromARGB(255, 177, 172, 172),
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      //leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    body: Form(
      key: _formKey,
      child: Card(
        child: Container(
          height: 200,
          padding: EdgeInsets.all(10.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Invitation Code: ${invitation_code}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Date: ${schedule_date}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Time: ${time_schedule}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Status: ${status}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Description: ${description}',
                style: TextStyle(color: Colors.black),
              ),
              SizedBox(
                height: 10,
              ),
              if (status == 'Created')
                 Center(
                child: ElevatedButton.icon(
                    style: ElevatedButton.styleFrom(
                        primary: Colors.blue,
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(10.0),
                          side: const BorderSide(color: Colors.white),
                        ),
                        elevation: 10,
                        minimumSize: const Size(150, 40)),
                    onPressed: () async {
                      final visit =
                          'Detail information, please visit the site';
                      final url =
                          'https://nscis.nsctechnology.com/index.php?r=site%2Fvisit';
                      final info =
                          'Select the "Visit" menu and insert these codes:';
                      await Share.share(
                          '${visit} ${url} ${info} ${invitation_code}');
                    },
                    icon: const Icon(Icons.share),
                    label: const Text(
                      "Share",
                      style: TextStyle(
                          fontSize: 18, fontWeight: FontWeight.bold),
                    )),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

scan/scanner.dart

```
import 'dart:developer';
import 'dart:io';
import 'package:biu_project/scan/update.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:qr_code_scanner/qr_code_scanner.dart';
import '../home_page.dart';

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

  @override
  State<StatefulWidget> createState() => _QRViewExampleState();
}

class _QRViewExampleState extends State<QRViewExample> {
  Barcode? result;
  String id = "";
  QRViewController? controller;
  final GlobalKey qrKey = GlobalKey(debugLabel: 'QR');

  // In order to get hot reload to work we need to pause the camera if the platform
  // is android, or resume the camera if the platform is iOS.
  @override
  void reassemble() {
    super.reassemble();
    if (Platform.isAndroid) {
      controller!.pauseCamera();
    }
    controller!.resumeCamera();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        children: <Widget>[
          Expanded(flex: 4, child: _buildQrView(context)),
          Expanded(
            flex: 1,
            child: FittedBox(
              fit: BoxFit.contain,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  // if (result != null)

                  //   Text(
                  //       'Barcode Type: ${describeEnum(result!.format)}   Data: ${result!.code}')
                  // else
                  //   const Text('Scan a code'),

                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    crossAxisAlignment: CrossAxisAlignment.center,
                    children: <Widget>[
                    Container(
                      margin: const EdgeInsets.all(8),
                      child: ElevatedButton(
                        onPressed: () async {
                          await controller?.pauseCamera();
                        },
                        child: const Text('pause',
                            style: TextStyle(fontSize: 20)),
                      ),
                    ),
                    Container(
                      margin: const EdgeInsets.all(8),
                      child: ElevatedButton(
                        onPressed: () async {
                          await controller?.resumeCamera();
                        },
                        child: const Text('resume',
                            style: TextStyle(fontSize: 20)),
                      ),
                    )
                  ],
                ),
              ],
            ),
          ),
        )
      ],
    ),
  );
}

Widget _buildQrView(BuildContext context) {
  // For this example we check how width or tall the device is and change the scanArea and overlay accordingly.
  var scanArea = (MediaQuery.of(context).size.width < 400 ||
          MediaQuery.of(context).size.height < 400)
      ? 150.0
      : 300.0;
  // To ensure the Scanner view is properly sizes after rotation
  // we need to listen for Flutter SizeChanged notification and update controller
  return QRView(
    key: qrKey,
    onQRViewCreated: _onQRViewCreated,
    overlay: QrScannerOverlayShape(
        borderColor: Colors.red,
        borderRadius: 10,
        borderLength: 30,
        borderWidth: 10,
        cutOutSize: scanArea),
    onPermissionSet: (ctrl, p) => _onPermissionSet(context, ctrl, p),
  );
}

void _onQRViewCreated(QRViewController controller) {
  setState(() {
    this.controller = controller;
  });
  controller.scannedDataStream.listen((scanData) {
    setState(() {
      result = scanData;
      Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) =>
          Update(id: result!.code)), (Route<dynamic> route) => false);
      print(result!.code);

    });
  });
}

void _onPermissionSet(BuildContext context, QRViewController ctrl, bool p) {
  log('${DateTime.now().toIso8601String()}_onPermissionSet $p');
```

```
    if (!p) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('no Permission')),
      );
    }
  }

  @override
  void dispose() {
    controller?.dispose();
    super.dispose();
  }
}
```

scan/update.dart

```
import 'dart:convert';
import 'package:biu_project/home_page.dart';
import 'package:biu_project/report.dart';
import 'package:biu_project/scan/report_visit.dart';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:http/http.dart' as http;
import 'package:share_plus/share_plus.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../login_page.dart';

class Update extends StatefulWidget {
  Update({required this.id});

  String? id;

  @override
  State<Update> createState() => _UpdateState();
}

class _UpdateState extends State<Update> {
  final _formKey = GlobalKey<FormState>();

  //inisialize field
  String username = "";
  var status_visit = TextEditingController();
  var status_done = TextEditingController();


  String? id_visitor, schedule_date,host, time_schedule, invitation_code, email, no_phone, full_name,company, address, status,
visitor_code, security;

  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
  setState(() {
    preferences.remove("is_login");
    preferences.remove("username");
  });

  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(
      builder: (BuildContext context) => const PageLogin(),
    ),
    (route) => false,
  );

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
      "Berhasil logout",
      style: TextStyle(fontSize: 16),
    )),
  );
}

 _getDataSecurity() async {
  try {
     SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    print(username);
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        "https://nscis.nsctechnology.com/index.php?r=t-visitor/view-security&id=" + username));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        security = data['item_name'];

      });
    }


  } catch (e) {
    print(e);
  }
}

//Http to get detail data
_getData() async {
  try {
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        //get detail data with id
        "http://nscis.nsctechnology.com/index.php?r=t-visitor/view-api&id=${widget.id}"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        //code = TextEditingController(text: data['invitation_code']);
        host = data['host'];
        //invitation_code = data['invitation_code'] !;
```

```
          schedule_date = data['schedule_date'];
          time_schedule = data['time_schedule'];
          status = data['status'];
          full_name = data['full_name'];
          email = data['email'];
          no_phone = data['no_phone'];
          full_name = data['full_name'];
          company = data['company'];
          address = data['address'];
          status_visit = TextEditingController(text: 'Validasi');
          status_done = TextEditingController(text: 'Finish');
          id_visitor = data['id_visitor'];

        });


    }
  } catch (e) {
    print(e);
  }
}

@override
dispose() {
  super.dispose();
}

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  getPref();
  _getData();
  _getDataSecurity();
}



Future _onUpdateSecurity(context) async {
      EasyLoading.show(status: 'Saving...');
      try {
        return await http.post(
          Uri.parse("https://nscis.nsctechnology.com/index.php?r=t-visitor/update-api"),
          body: {
            "id_visitor": id_visitor,
            "status": "Validasi",
          },
        ).then((value) {
          //print message after insert to database
          //you can improve this message with alert dialog
          EasyLoading.showSuccess('data successfully saved');
          var data = jsonDecode(value.body);
          print(data["message"]);

          Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) =>
          HomePage()), (Route<dynamic> route) => false);

          EasyLoading.dismiss();

        });
      } catch (e) {
        print(e);
      }
    }

Future _onUpdateHost(context) async {
      EasyLoading.show(status: 'Saving...');
      try {
        return await http.post(
          Uri.parse("https://nscis.nsctechnology.com/index.php?r=t-visitor/update-api"),
          body: {
            "id_visitor": id_visitor,
            "status": "Finish",
          },
        ).then((value) {
          //print message after insert to database
          //you can improve this message with alert dialog
          EasyLoading.showSuccess('data successfully saved');
          var data = jsonDecode(value.body);
          print(data["message"]);

          Navigator.of(context).pushAndRemoveUntil(MaterialPageRoute(builder: (context) =>
          ReportVisit()), (Route<dynamic> route) => false);

          EasyLoading.dismiss();

        });
      } catch (e) {
        print(e);
      }
    }

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Color.fromARGB(255, 177, 172, 172),
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      //leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    body: Form(
      key: _formKey,
      child: Card(
        child: Container(
          height: 200,
          padding: EdgeInsets.all(10.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Date: ${schedule_date}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Time: ${time_schedule}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Name: ${full_name}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Company: ${company}',
                style: TextStyle(color: Colors.black),
              ),
              Text(
                'Email: ${email}',
                style: TextStyle(color: Colors.black),
                ),
                Text(
                  'Phone Number: ${no_phone}',
                  style: TextStyle(color: Colors.black),
                ),
                Text(
                  'Status: ${status}',
                  style: TextStyle(color: Colors.black),
                ),
                Text(
                  'Address: ${address}',
                  style: TextStyle(color: Colors.black),
                ),
                SizedBox(
                  height: 10,
                ),
                if (status == 'Created' && security == 'security')
                  Center(
                    child: ElevatedButton(
                      style: ElevatedButton.styleFrom(
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(15),
                        ),
                      ),
                      child: Text(
                        "Accept",
                        style: TextStyle(color: Colors.white),
                      ),
                      onPressed: () {
                        //validate
                        if (_formKey.currentState!.validate()) {
                          //send data to database with this method
                          _onUpdateSecurity(context);
                        }
                      },
                    ),
                  ),
                if (status == 'Validasi' && username == host)
                Center(
                    child: ElevatedButton(
                      style: ElevatedButton.styleFrom(
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(15),
                        ),
                      ),
                      child: Text(
                        "Accept",
                        style: TextStyle(color: Colors.white),
                      ),
                      onPressed: () {
                        //validate
                        if (_formKey.currentState!.validate()) {
                          //send data to database with this method
                          _onUpdateHost(context);
                        }
                      },
                    ),
                  ),

              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

scan/report_visit.dart

```
import 'dart:convert';
import 'dart:typed_data';
import 'package:flutter/material.dart';
import 'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:http/http.dart' as http;
import 'package:biu_project/invitation/view.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../login_page.dart';

class ReportVisit extends StatefulWidget {
  const ReportVisit(
      {Key? key})
      : super(key: key);
  @override
  ReportVisitState createState() => ReportVisitState();
}

class ReportVisitState extends State<ReportVisit> {
  String id = "";
  String username = "";
  List _get = [];
  getPref() async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    if (islogin != null && islogin == true) {
      setState(() {
        username = pref.getString("username")!;
      });
    } else {
      Navigator.of(context, rootNavigator: true).pop();
      Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
          builder: (BuildContext context) => const PageLogin(),
        ),
        (route) => false,
      );
    }
  }

  logOut() async {
    SharedPreferences preferences = await SharedPreferences.getInstance();
    setState(() {
      preferences.remove("is_login");
      preferences.remove("username");
    });

    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => const PageLogin(),
      ),
      (route) => false,
    );

    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
          content: Text(
        "Berhasil logout",
        style: TextStyle(fontSize: 16),
      )),
    );
  }

  _getData() async {
    print(username);
    try {
     SharedPreferences pref = await SharedPreferences.getInstance();
    var islogin = pref.getBool("is_login");
    setState(() {
      username = pref.getString("username")!;
    });
    print(username);
    final response = await http.get(Uri.parse(
        //you have to take the ip address of your computer.
        //because using localhost will cause an error
        "https://nscis.nsctechnology.com/index.php?r=t-visitor/visitor-api&id=" + username));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        _get = data;
      });
    }
  } catch (e) {
    print(e);
  }
}

//make different color to different card
final _lightColors = [
  Colors.amber.shade300,
  Colors.lightGreen.shade300,
  Colors.lightBlue.shade300,
  Colors.orange.shade300,
  Colors.pinkAccent.shade100,
  Colors.tealAccent.shade100
];

@override
dispose() {
  super.dispose();
}

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  getPref();
  _getData();
}



@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[600],
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      //leading: Icon(Icons.menu),
      title: Text("BIU"),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: () {
            logOut();
          },
        ),
      ],
    ),
    //if not equal to 0 show data
```

```
      //else show text "no data available"
      body: _get.length != 0
          //we use masonry grid to make masonry card style
          ? MasonryGridView.count(
              crossAxisCount: 2,
              itemCount: _get.length,
              itemBuilder: (context, index) {
                return GestureDetector(

                  child: Card(
                    //make random color to eveery card
                    color: _lightColors[index % _lightColors.length],
                    child: Container(
                      //make 2 different height
                      // constraints:
                      //     BoxConstraints(minHeight: (index % 2 + 1) * 85),
                      padding: EdgeInsets.all(15),
                      child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text(
                            'Name: ${_get[index]['full_name']}',
                            style: TextStyle(color: Colors.black),
                          ),
                          Text(
                            'Company: ${_get[index]['company']}',
                            style: TextStyle(color: Colors.black),
                          ),
                          SizedBox(height: 8),
                          Text(
                            'Status: ${_get[index]['status']}',
                            style: TextStyle(
                              color: Colors.black,
                              fontSize: 14,
                              fontWeight: FontWeight.bold,
                            ),
                          ),
                        ],
                      ),
                    ),
                  ),
                );
              },
            )
          : Center(
              child: Text(
                "No Data Available",
                style: TextStyle(
                  // color: Colors.white,
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
    );
  }
}
```