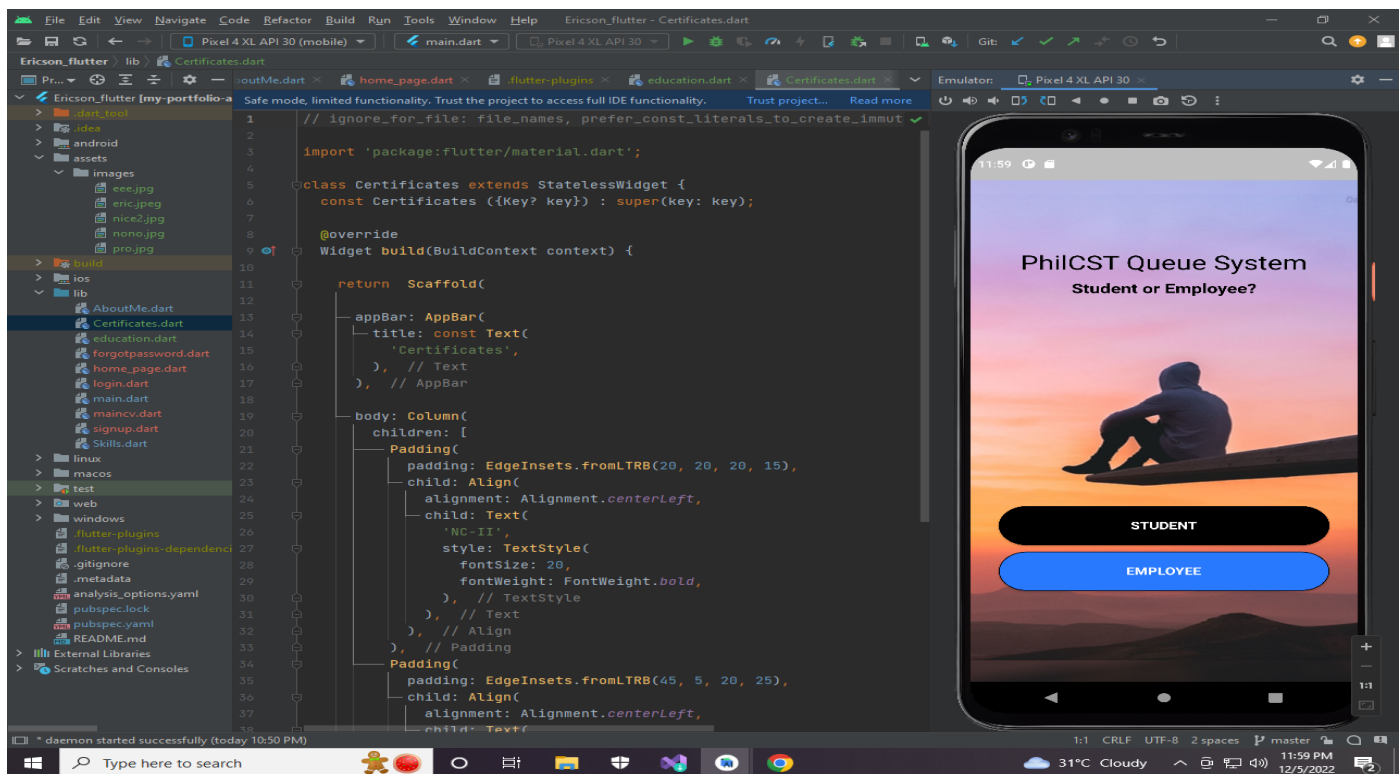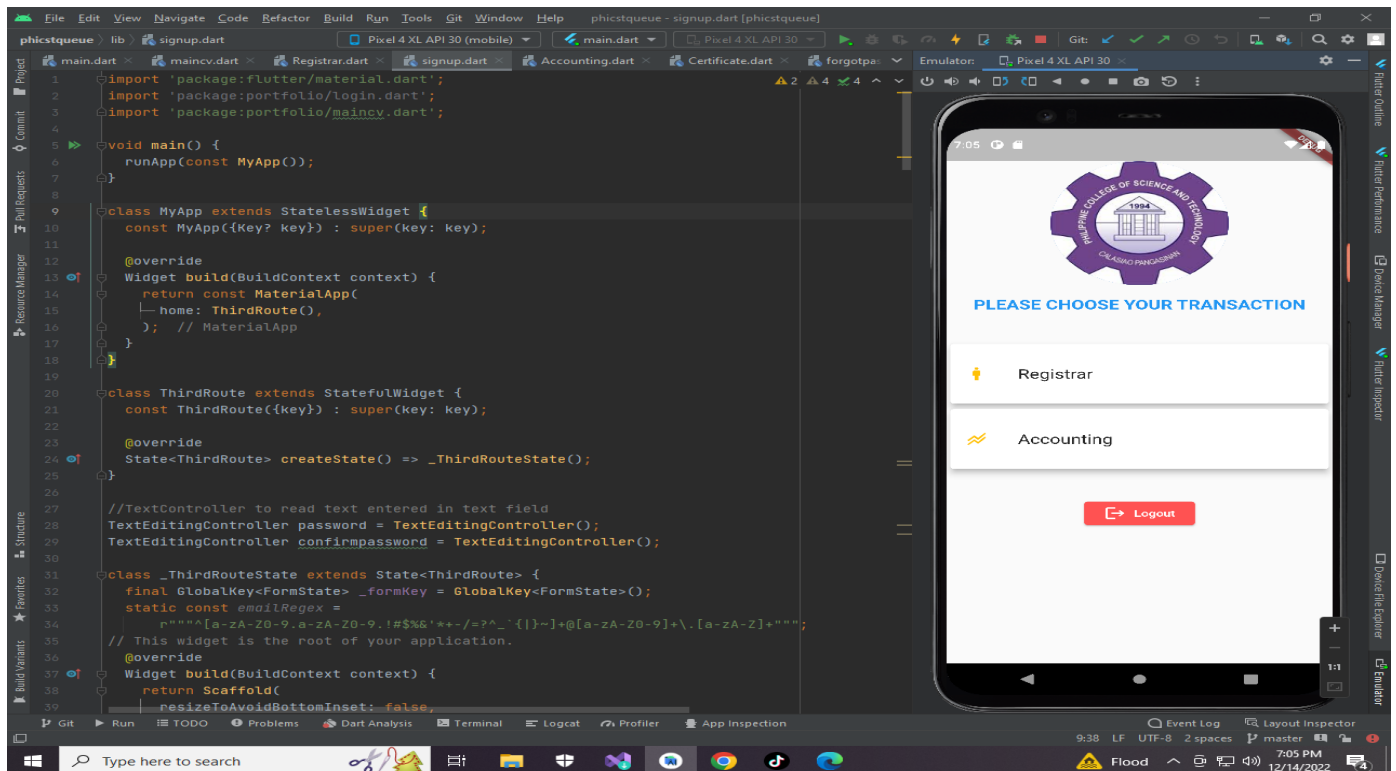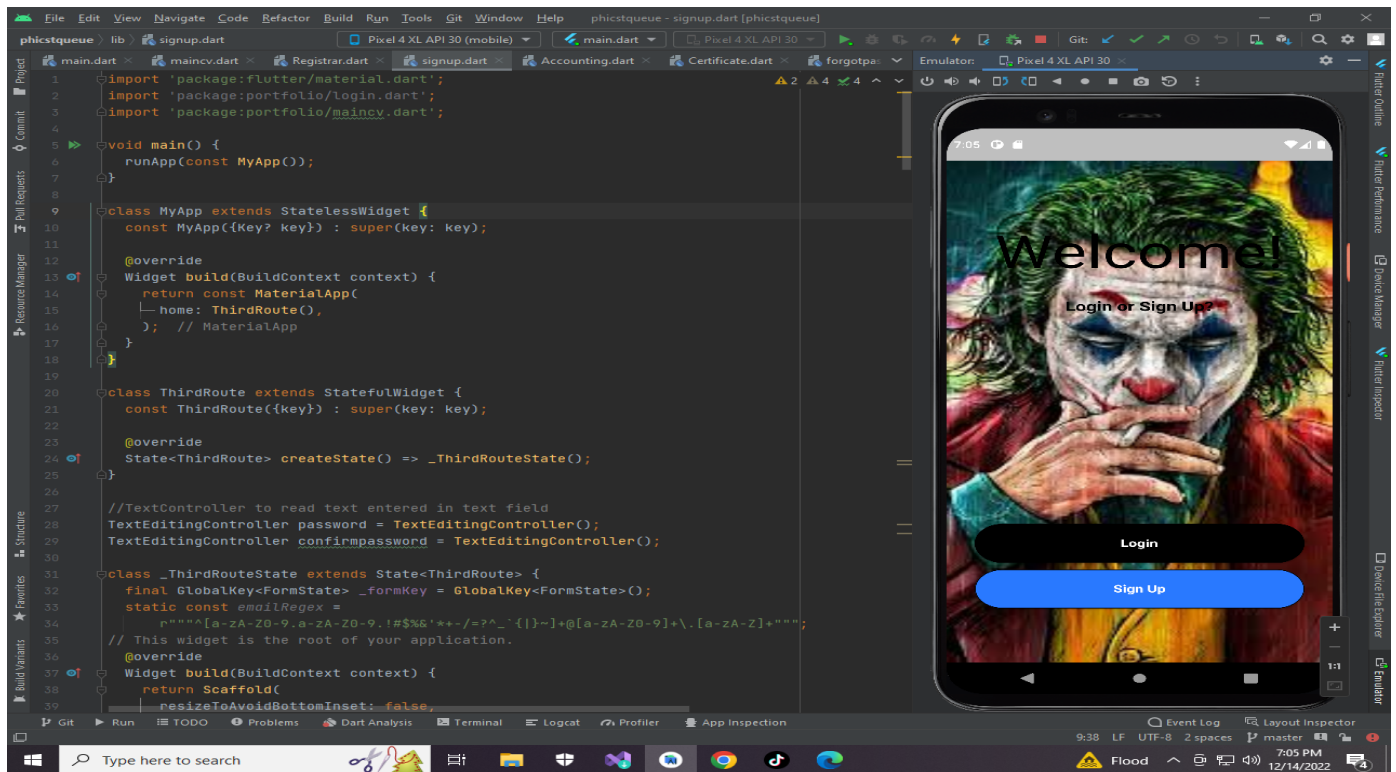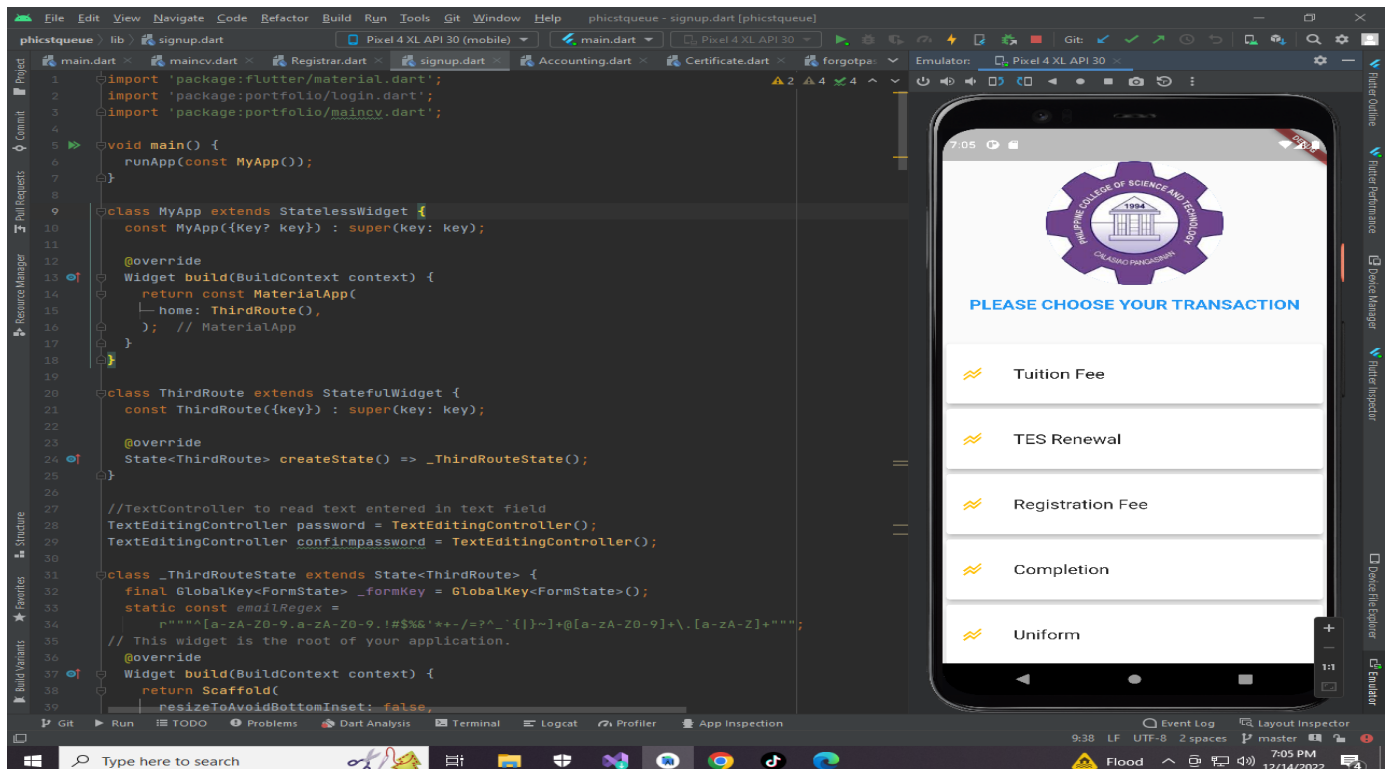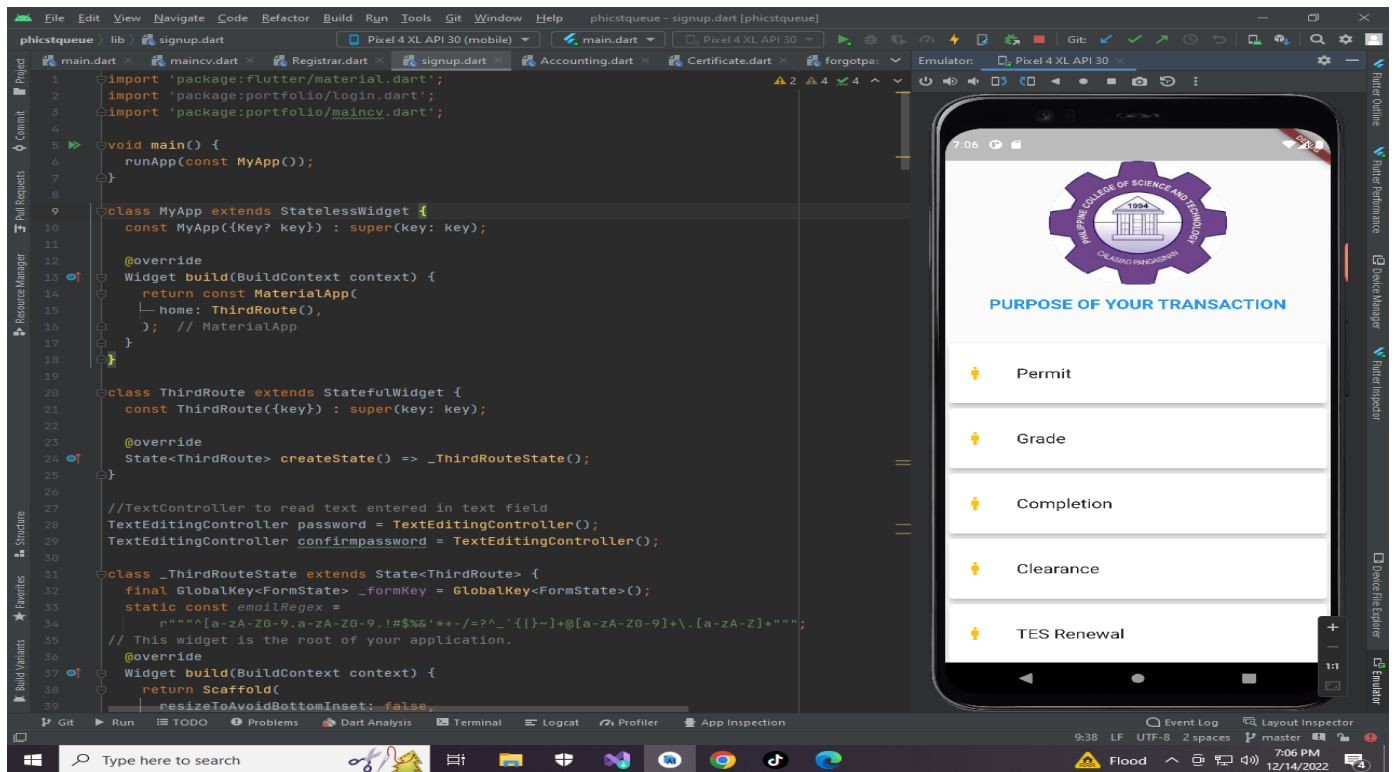**Ericson A. Alegria**

BSIT-4 Block –1

MY COMPILATION LABORATORY

PHILCST QUEUE SYSTEM

Void main () is the entry point for execution in C program. The void is a keyword that represents function will not return anything but a void value. Main is the name of the function and () represents parameter list that can be passed to function in this case nothing is passed.

File Edit View Navigate Code Refactor Build Run Tools Git Window Help

phicstqueue ` lib ` signup.dart

main.dart   maincv.dart   Registrar.dart   signup.dart   Accounting.dart   Certificate.dart   forgotpa...

Emulator: Pixel 4 XL API 30

```
import 'package:flutter/material.dart';
import 'package:portfolio/login.dart';
import 'package:portfolio/maincv.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: ThirdRoute(),
    ); // MaterialApp
  }
}

class ThirdRoute extends StatefulWidget {
  const ThirdRoute({key}) : super(key: key);

  @override
  State<ThirdRoute> createState() => _ThirdRouteState();
}

//TextController to read text entered in text field
TextEditingController password = TextEditingController();
TextEditingController confirmpassword = TextEditingController();

class _ThirdRouteState extends State<ThirdRoute> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  static const emailRegex =
      r"""^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+-/=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+""";
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
```

Git   Run   TODO   Problems   Dart Analysis   Terminal   Logcat   Profiler   App Inspection

Emulator screen 1:

Welcome!

Login or Sign Up?

Login

Sign Up

Emulator screen 2:

PLEASE CHOOSE YOUR TRANSACTION

Registrar

Accounting

Logout

CV_RESUME_ERICSON

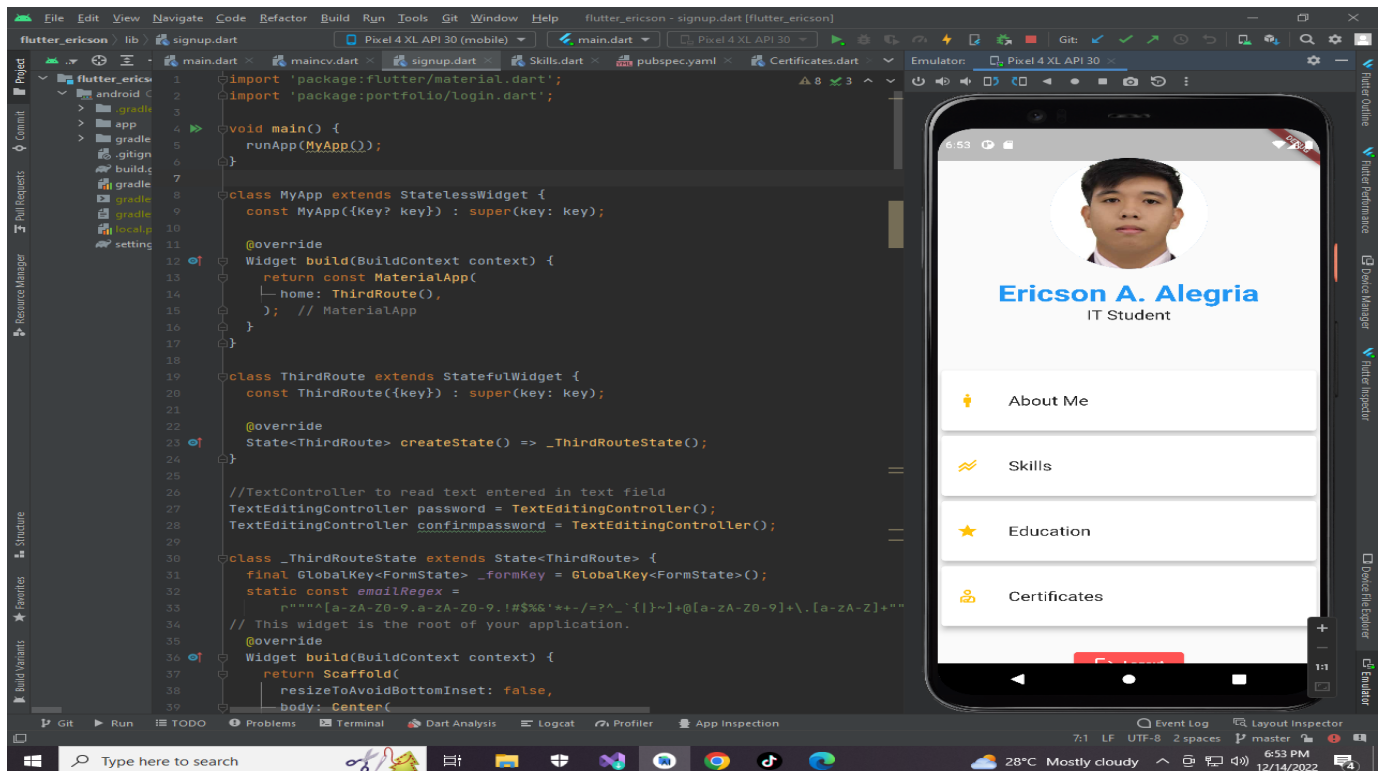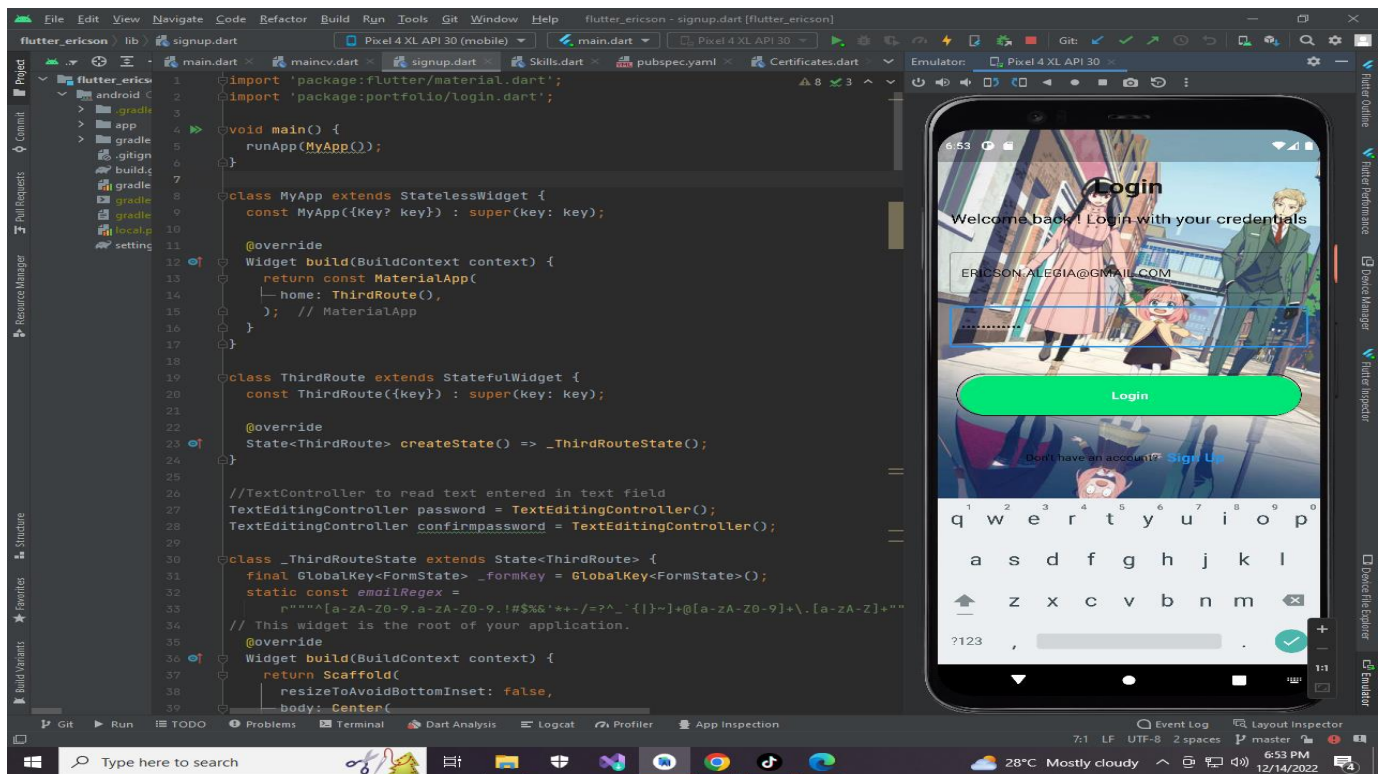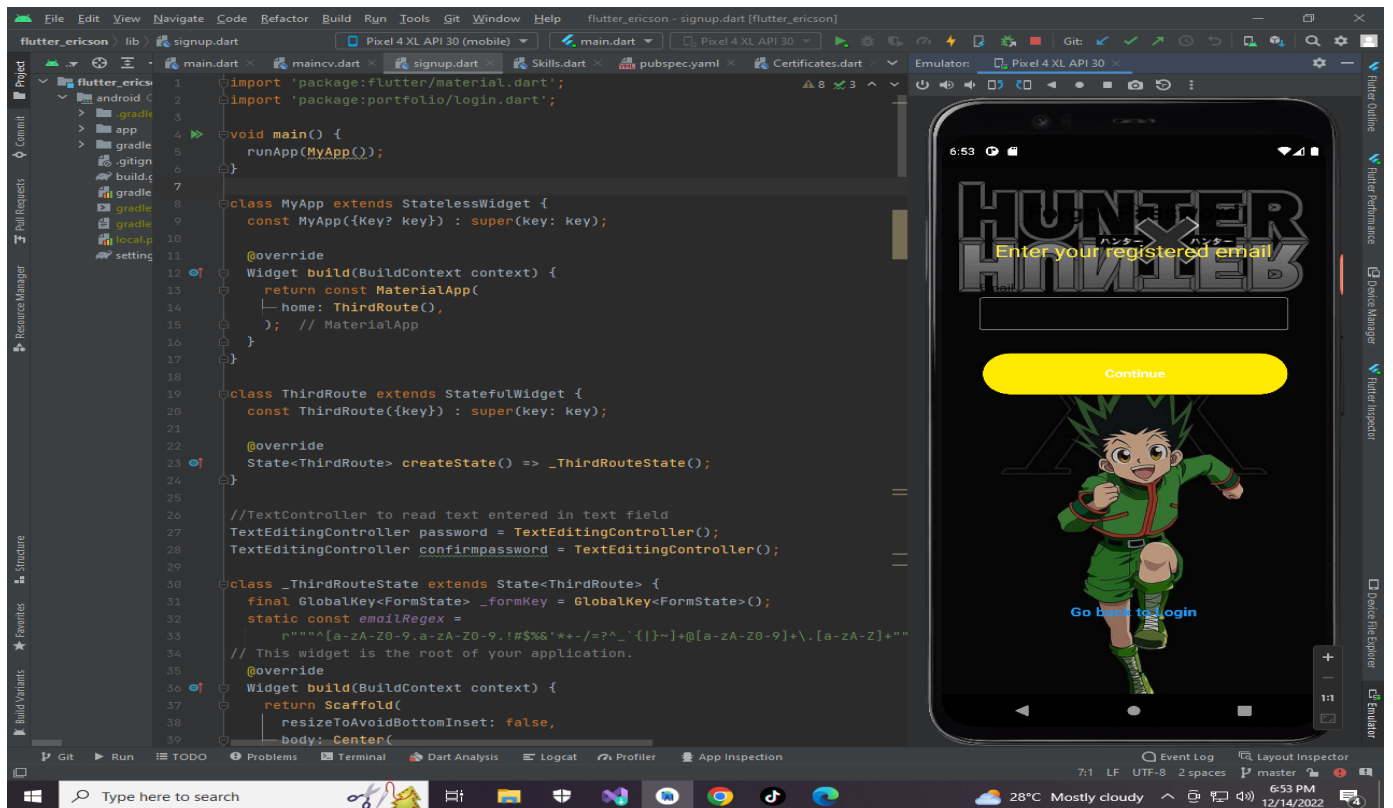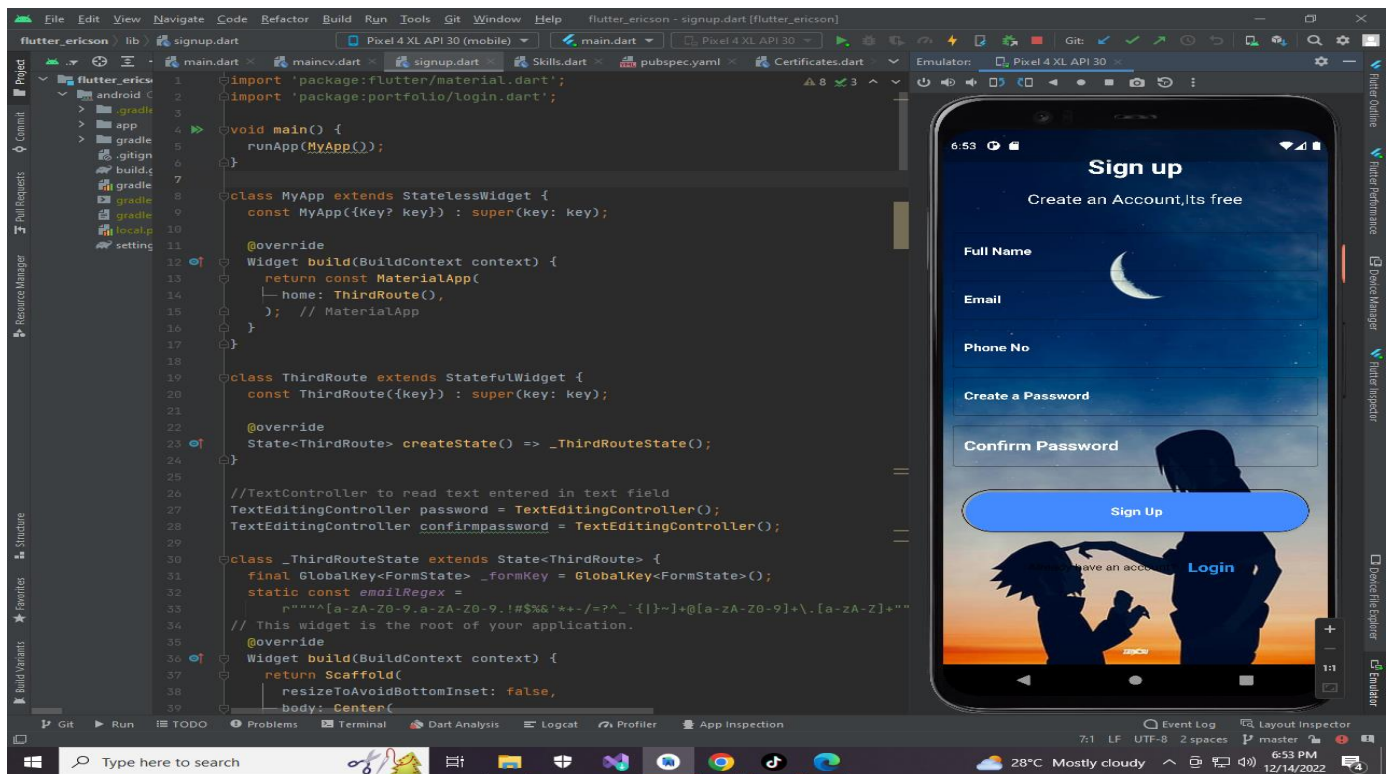Stateless widget are useful when the part of the user interface you are describing does not depend on anything other than the configuration information in the object itself and the BuildContext in which the widget is inflated. For compositions that can change dynamically, e.g. due to having an internal clock-driven state, or depending on some system state, consider using StatefulWidget
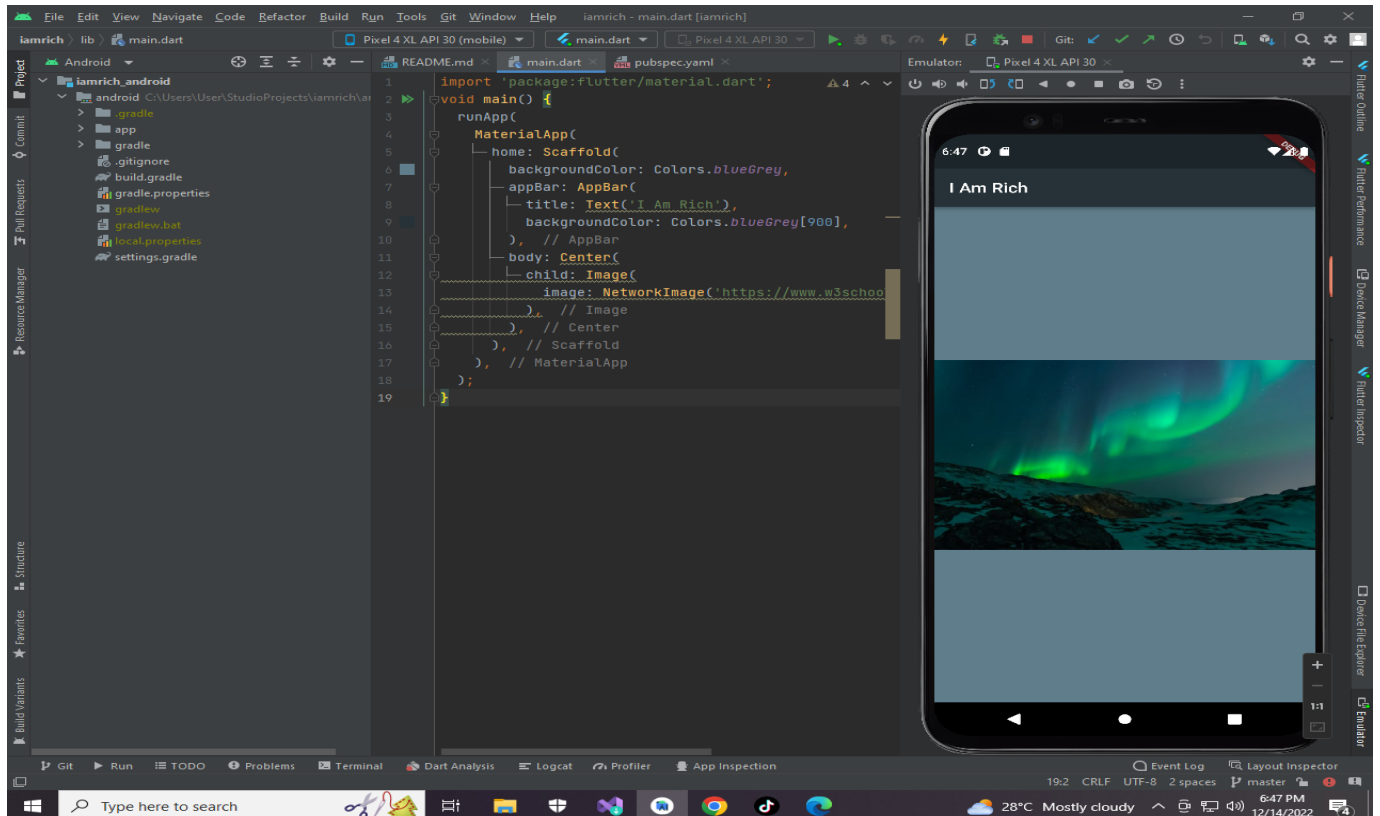
Screenshot 1:

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

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: ThirdRoute(),
    ); // MaterialApp
  }
}

class ThirdRoute extends StatefulWidget {
  const ThirdRoute({key}) : super(key: key);

  @override
  State<ThirdRoute> createState() => _ThirdRouteState();
}

//TextController to read text entered in text field
TextEditingController password = TextEditingController();
TextEditingController confirmpassword = TextEditingController();

class _ThirdRouteState extends State<ThirdRoute> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  static const emailRegex =
      r"""^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+-/=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z-Z]+""";
// This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      body: Center(
```

Sign up
Create an Account,Its free
Full Name
Email
Phone No
Create a Password
Confirm Password
Sign Up
have an account   Login

Screenshot 2:

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

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: ThirdRoute(),
    ); // MaterialApp
  }
}

class ThirdRoute extends StatefulWidget {
  const ThirdRoute({key}) : super(key: key);

  @override
  State<ThirdRoute> createState() => _ThirdRouteState();
}

//TextController to read text entered in text field
TextEditingController password = TextEditingController();
TextEditingController confirmpassword = TextEditingController();

class _ThirdRouteState extends State<ThirdRoute> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  static const emailRegex =
      r"""^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+-/=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z-Z]+""";
// This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      body: Center(
```

HUNTER X HUNTER
Enter your registered email
Continue
Go back to login

I AM RICH

A package is a namespace that contains a group of similar types of classes, interfaces**,** and sub-packages. We can think of packages as similar to different folders on our computers where we might keep movies in one folder, images in another folder, software in another folder, etc.

I AM POOR

An app bar consists of a toolbar and potentially other widgets, such as a TabBar and a FlexibleSpaceBar. App bars typically expose one or more common actions with IconButtons which are optionally followed by a PopupMenuButton for less common operations (sometimes called the "overflow menu").