Researching Project IoT Platform using Laravel and Mobile App

‘ Week8 Report ’

Lecturer: Mr. HEL Chanthan

Prepared by Mr. VANNAK Sovannroth

2019-2020

Outline

- ❖ Planning for First Month
- ❖ Planning for Second Month
- ❖ Main Coding and Emulator
- ❖ Adding Library of Charts in Flutter
- ❖ Coding
- ❖ Sample Charts Code
- ❖ View on Mobile Application

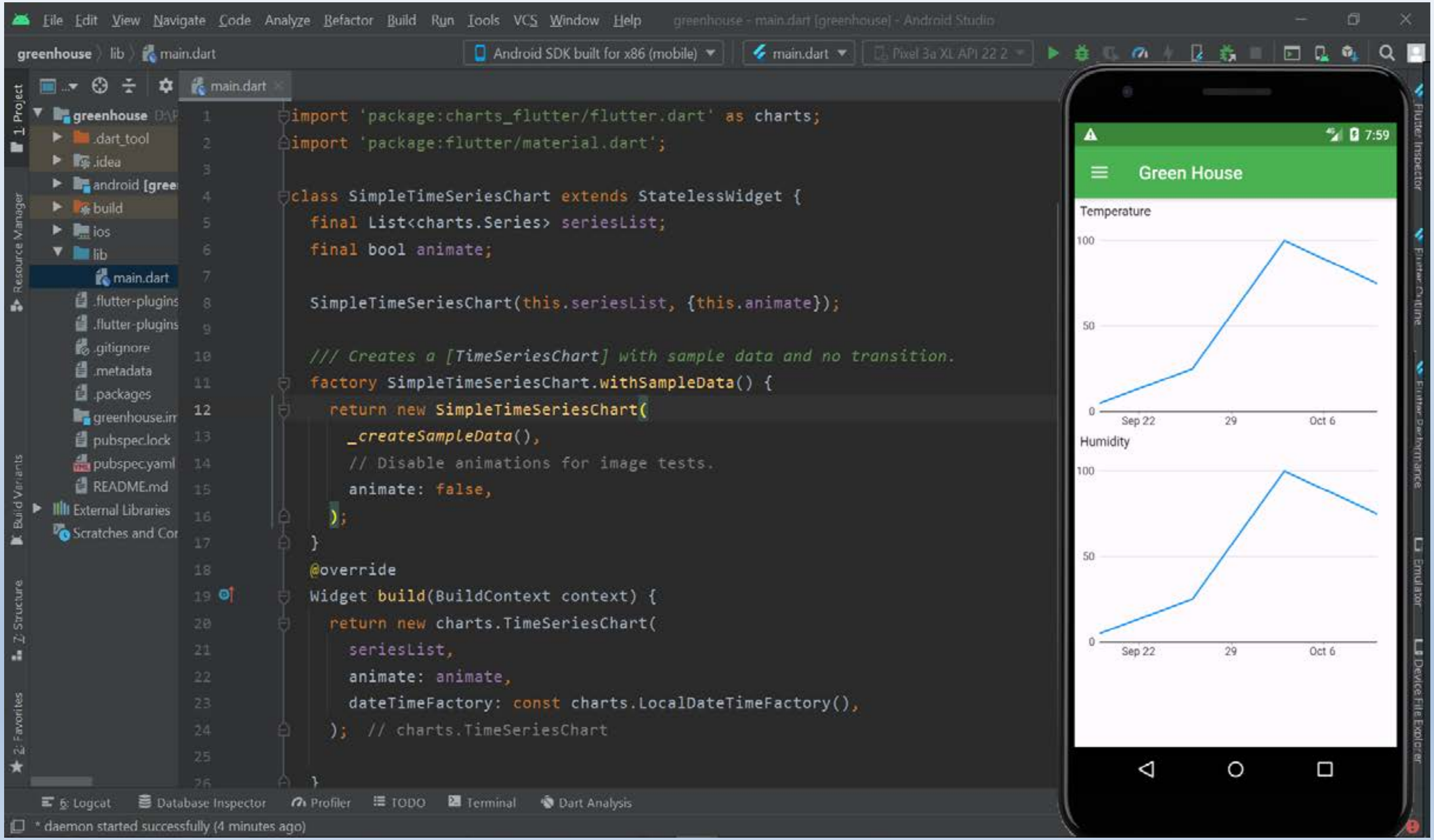
❖ Planning for First Month

Activity (Start on 06.08.2020)	Thursday (Presentation day)	Friday	Monday	Tuesday	Wednesday (Meeting day)
Week 1	Installation MongoDB (Shell and Compass) and connect localhost between Shell and Compass.(Locally)	Performing CRUD (Create, Read, Update, Delete) on Mongo Shell and MongoDB Compass.	Creating MongoDB Atlas account and connect it to Mongo Shell or MongoDB Compass, and Driver (Node.js).(Cloud)	Installation Visual Studio Code and connect driver to MongoDB Atlas.(Cloud)	Connecting sample Backend to MongoDB Cluster and storing Products in the Database.
Week 2	Connecting driver to Mongo Shell.(Locally)	Starting to build sample backend connect with or without MongoDB Atlas.	Build creating, editing and deleting Products on server.	Transmitting and fetching data to/from the Database.	Creating Login and Signup. (Email and password)
Week 3	Learning template Bootstrap 4	Install wampserver and Laravel	Run Laravel server with MySQL(Database)	Choose free template Bootstrap for Front-End & Backend	Installation Laravel Module for Backend and Front-End
Week 4	Starting to build backend for project. & Creating Login and Signup.	User management: • Create new user • Reset user password • Delete and edit user	Manage Info: • Logo • Name • Contact info • Map • Social	Manage new category: • Create new category • Edit and delete exiting categories.	Manage news: • Post news • Edit news • Delete news

❖ Planning for Second Month

Activity (Start on 06.08.2020)	Thursday (Presentation day)	Friday	Monday	Tuesday	Wednesday (Meeting day)
Week 5	Create Dashboard	Create Dashboard	Manage Calculate: • Addition • Subtraction • Multiplication • Division	Write API & Read API	Write API & Read API
Week 6	Connect to ESP8266 or NodeMcu & fetch data from Arduino	Get data from Sensors and Calculate.	Show data on Dashboard	Connect Backend to Front-End	Front-End fetch data from Backend
Week 7	Choose Flutter or React Native for build Mobile App.	Learning Flutter or React native	Start build sample project	Create Horizontal list view & build recent products grid view.	Manage Info: • Product details • Home page Ui modification • Shopping cart & list page • User login & Register • Contact info
Week 8	Creating Users, brand and category on database & Adding products to the database.	Auto complete search for categories and brands & Uploading Images to firebase.	Login and signup screen redesign & Google sign in for IOS and Android .	Firebase Auth using provider package & Dashboard UI/UX and Database product modeling.	Load products from firebase part & loading products from database.

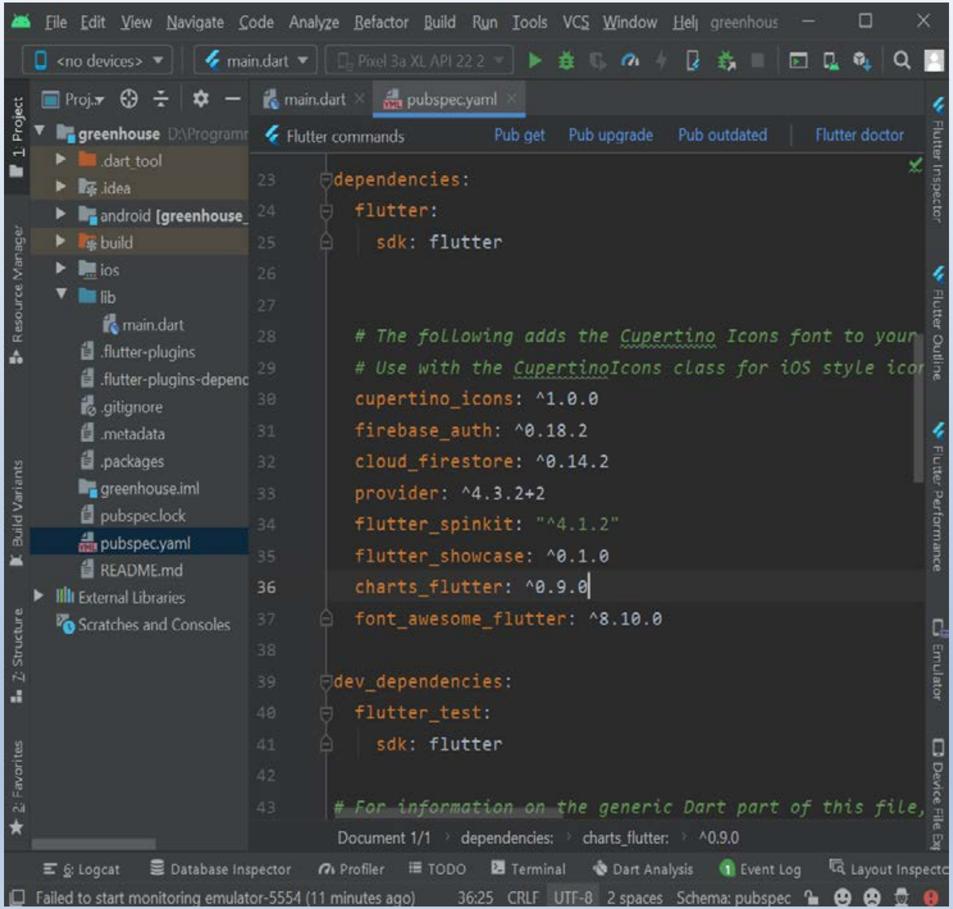
<<< Main Coding and Emulator >>>>



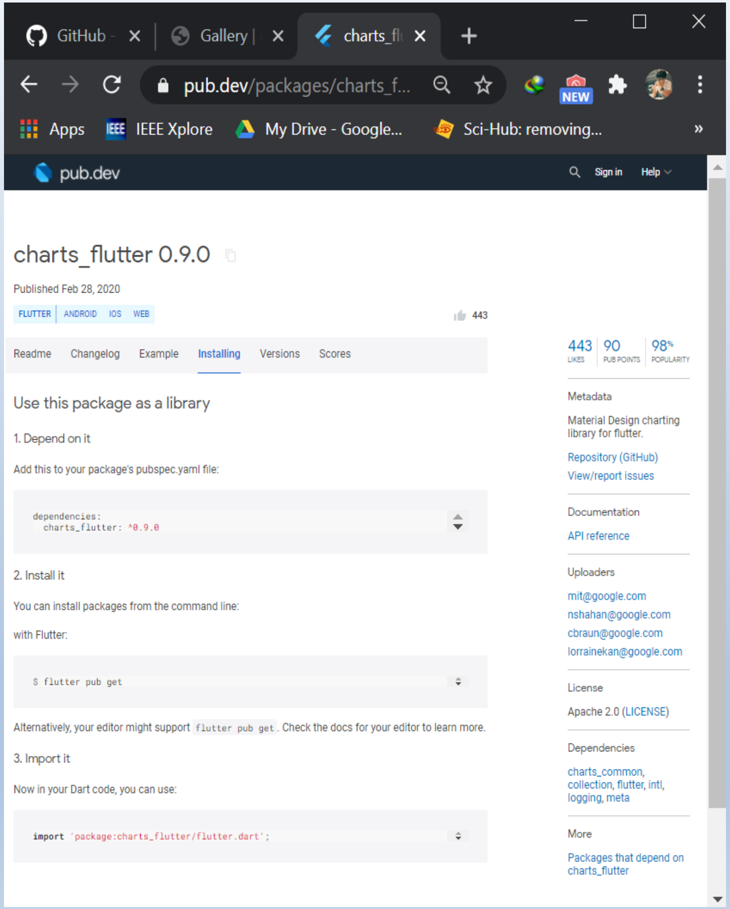
```
import 'package:charts_flutter/flutter.dart' as charts;
import 'package:flutter/material.dart';

class SimpleTimeSeriesChart extends StatelessWidget {
  final List<charts.Series> seriesList;
  final bool animate;

  SimpleTimeSeriesChart(this.seriesList, {this.animate});

  /// Creates a [TimeSeriesChart] with sample data and no transition.
  factory SimpleTimeSeriesChart.withSampleData() {
    return new SimpleTimeSeriesChart(
      _createSampleData(),
      // Disable animations for image tests.
      animate: false,
    );
  }
  @override
  Widget build(BuildContext context) {
    return new charts.TimeSeriesChart(
      seriesList,
      animate: animate,
      dateTimeFactory: const charts.LocalDateTimeFactory(),
    ); // charts.TimeSeriesChart

  }
```

```
* daemon started successfully (4 minutes ago)
```

<<< Adding Library of Charts in Flutter>>>>



```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your
  # Use with the CupertinoIcons class for iOS style icon
  cupertino_icons: ^1.0.0
  firebase_auth: ^0.18.2
  cloud_firestore: ^0.14.2
  provider: ^4.3.2+2
  flutter_spinkit: "^4.1.2"
  flutter_showcase: ^0.1.0
  charts_flutter: ^0.9.0
  font_awesome_flutter: ^8.10.0

dev_dependencies:
  flutter_test:
    sdk: flutter

# For information on the generic Dart part of this file,
```



charts_flutter 0.9.0

Published Feb 28, 2020

FLUTTER ANDROID IOS WEB

Use this package as a library

1. Depend on it

Add this to your package's pubspec.yaml file:

```
dependencies:
  charts_flutter: ^0.9.0
```

2. Install it

You can install packages from the command line:

with Flutter:

```
$ flutter pub get
```

Alternatively, your editor might support `flutter pub get`. Check the docs for your editor to learn more.

3. Import it

Now in your Dart code, you can use:

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

Metadata

Material Design charting library for flutter.

License

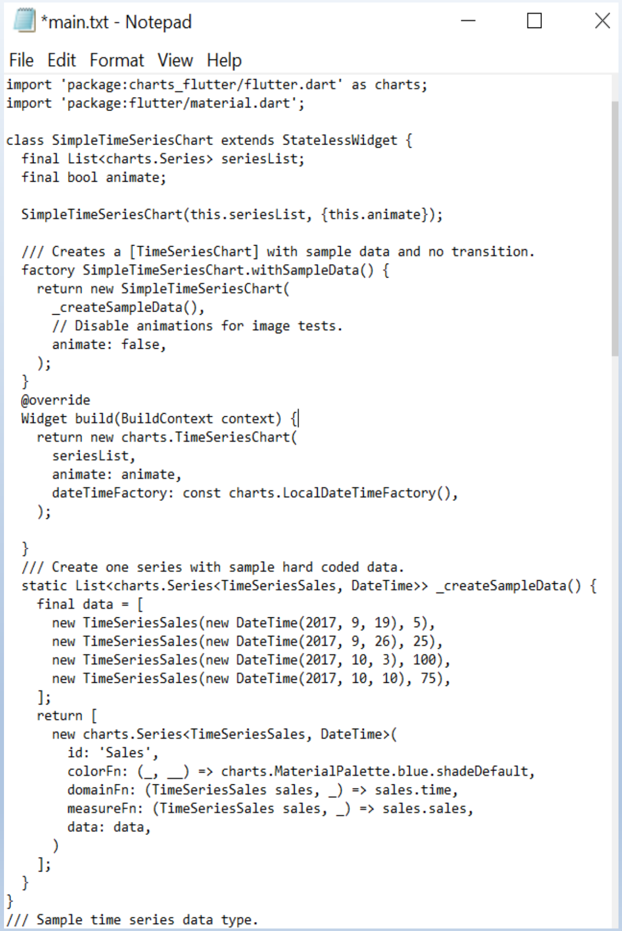
Apache 2.0 (LICENSE)

Dependencies

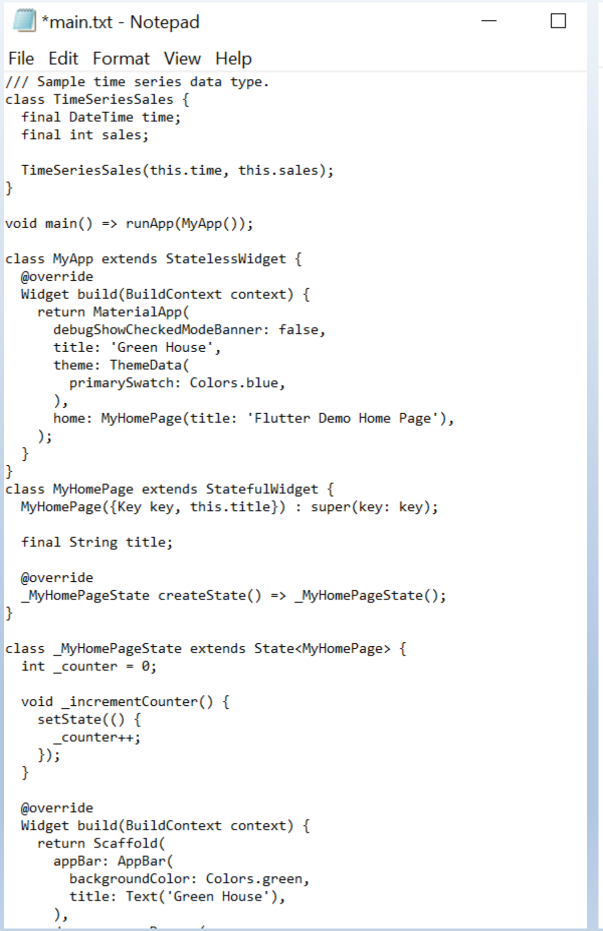
charts_common, collection, flutter, intl, logging, meta

https://pub.dev/packages/charts_flutter/install

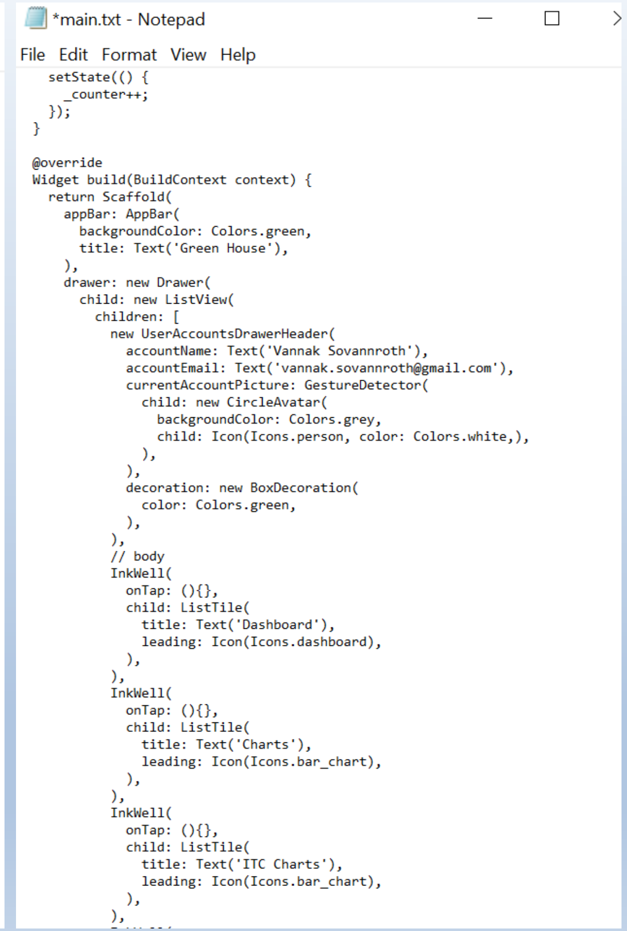
<<< Coding>>>>



```
import 'package:charts_flutter/flutter.dart' as charts;
import 'package:flutter/material.dart';

class SimpleTimeSeriesChart extends StatelessWidget {
  final List<charts.Series> seriesList;
  final bool animate;

  SimpleTimeSeriesChart(this.seriesList, {this.animate});

  /// Creates a [TimeSeriesChart] with sample data and no transition.
  factory SimpleTimeSeriesChart.withSampleData() {
    return new SimpleTimeSeriesChart(
      _createSampleData(),
      // Disable animations for image tests.
      animate: false,
    );
  }
  @override
  Widget build(BuildContext context) {
    return new charts.TimeSeriesChart(
      seriesList,
      animate: animate,
      dateTimeFactory: const charts.LocalDateTimeFactory(),
    );

  }
  /// Create one series with sample hard coded data.
  static List<charts.Series<TimeSeriesSales, DateTime>> _createSampleData() {
    final data = [
      new TimeSeriesSales(new DateTime(2017, 9, 19), 5),
      new TimeSeriesSales(new DateTime(2017, 9, 26), 25),
      new TimeSeriesSales(new DateTime(2017, 10, 3), 100),
      new TimeSeriesSales(new DateTime(2017, 10, 10), 75),
    ];
    return [
      new charts.Series<TimeSeriesSales, DateTime>(
        id: 'Sales',
        colorFn: (_, __) => charts.MaterialPalette.blue.shadeDefault,
        domainFn: (TimeSeriesSales sales, _) => sales.time,
        measureFn: (TimeSeriesSales sales, _) => sales.sales,
        data: data,
      )
    ];
  }
}
/// Sample time series data type.
```

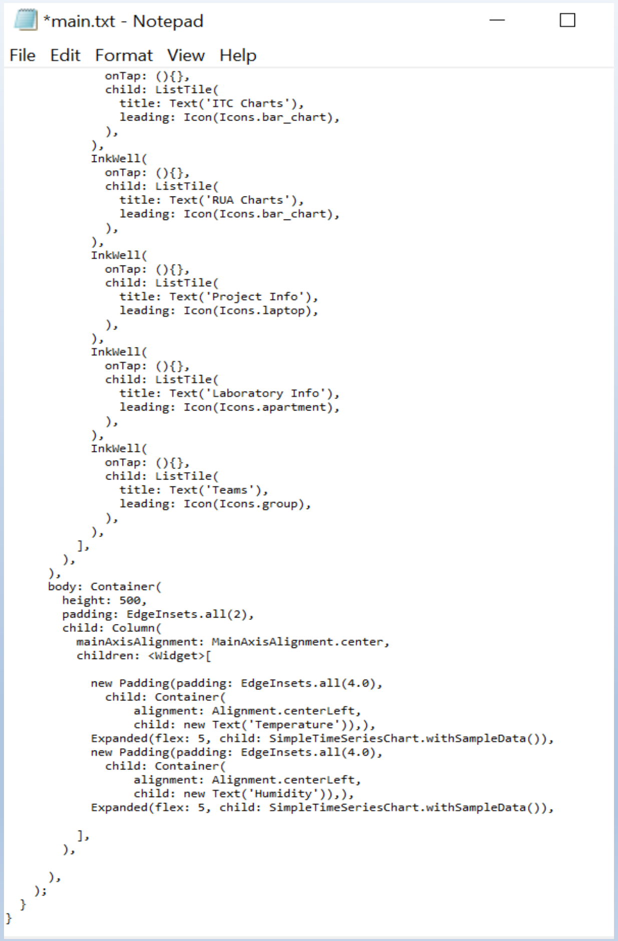


```
/// Sample time series data type.
class TimeSeriesSales {
  final DateTime time;
  final int sales;

  TimeSeriesSales(this.time, this.sales);
}

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Green House',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}
class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  final String title;

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

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.green,
        title: Text('Green House'),
      ),
```

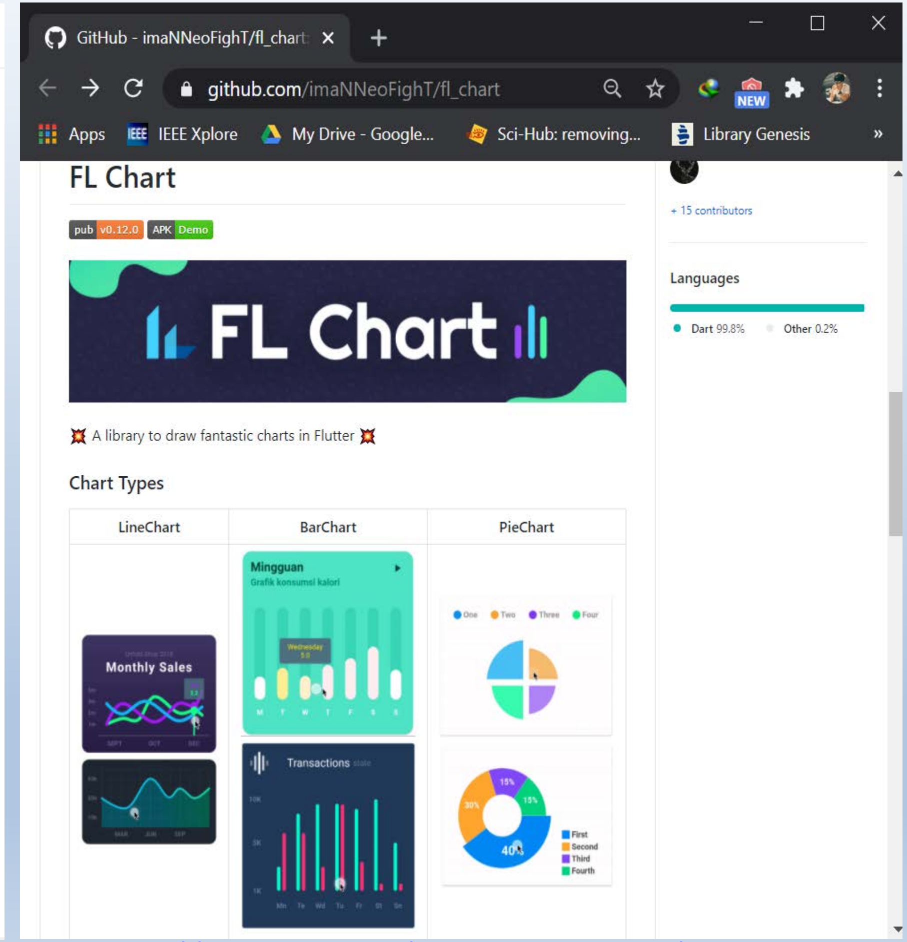


```
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.green,
        title: Text('Green House'),
      ),
      drawer: new Drawer(
        child: new ListView(
          children: [
            new UserAccountsDrawerHeader(
              accountName: Text('Vannak Sovannroth'),
              accountEmail: Text('vannak.sovannroth@gmail.com'),
              currentAccountPicture: GestureDetector(
                child: new CircleAvatar(
                  backgroundColor: Colors.grey,
                  child: Icon(Icons.person, color: Colors.white,),
                ),
              ),
              decoration: new BoxDecoration(
                color: Colors.green,
              ),
            ),
            // body
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('Dashboard'),
                leading: Icon(Icons.dashboard),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('Charts'),
                leading: Icon(Icons.bar_chart),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('ITC Charts'),
                leading: Icon(Icons.bar_chart),
              ),
            ),
```

<<< Coding>>>>



```
              onTap: (){},
              child: ListTile(
                title: Text('ITC Charts'),
                leading: Icon(Icons.bar_chart),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('RUA Charts'),
                leading: Icon(Icons.bar_chart),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('Project Info'),
                leading: Icon(Icons.laptop),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('Laboratory Info'),
                leading: Icon(Icons.apartment),
              ),
            ),
            InkWell(
              onTap: (){},
              child: ListTile(
                title: Text('Teams'),
                leading: Icon(Icons.group),
              ),
            ),
          ],
        ),
      ),
      body: Container(
        height: 500,
        padding: EdgeInsets.all(2),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[

            new Padding(padding: EdgeInsets.all(4.0),
              child: Container(
                  alignment: Alignment.centerLeft,
                  child: new Text('Temperature')),),
            Expanded(flex: 5, child: SimpleTimeSeriesChart.withSampleData()),
            new Padding(padding: EdgeInsets.all(4.0),
              child: Container(
                  alignment: Alignment.centerLeft,
                  child: new Text('Humidity')),),
            Expanded(flex: 5, child: SimpleTimeSeriesChart.withSampleData()),

          ],
        ),

      ),
    );
  }
}
```



FL Chart

pub v0.12.0 APK Demo

FL Chart

💥 A library to draw fantastic charts in Flutter 💥

Chart Types

LineChart	BarChart	PieChart

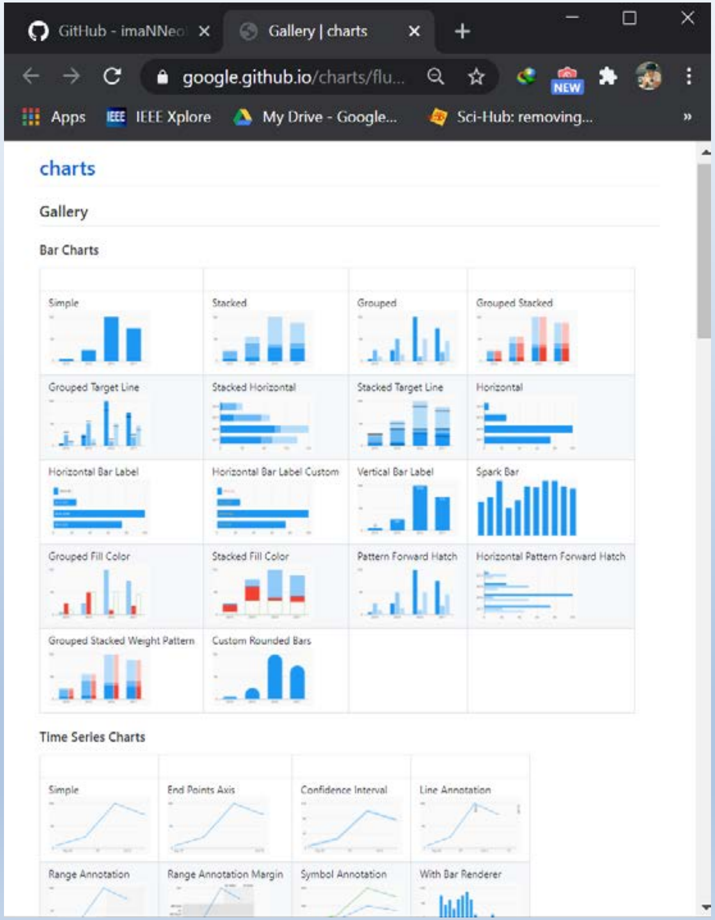
+ 15 contributors

Languages

● Dart 99.8% ● Other 0.2%

https://github.com/imaNNeoFighT/fl_chart

<<< Sample Charts Code >>>>



charts

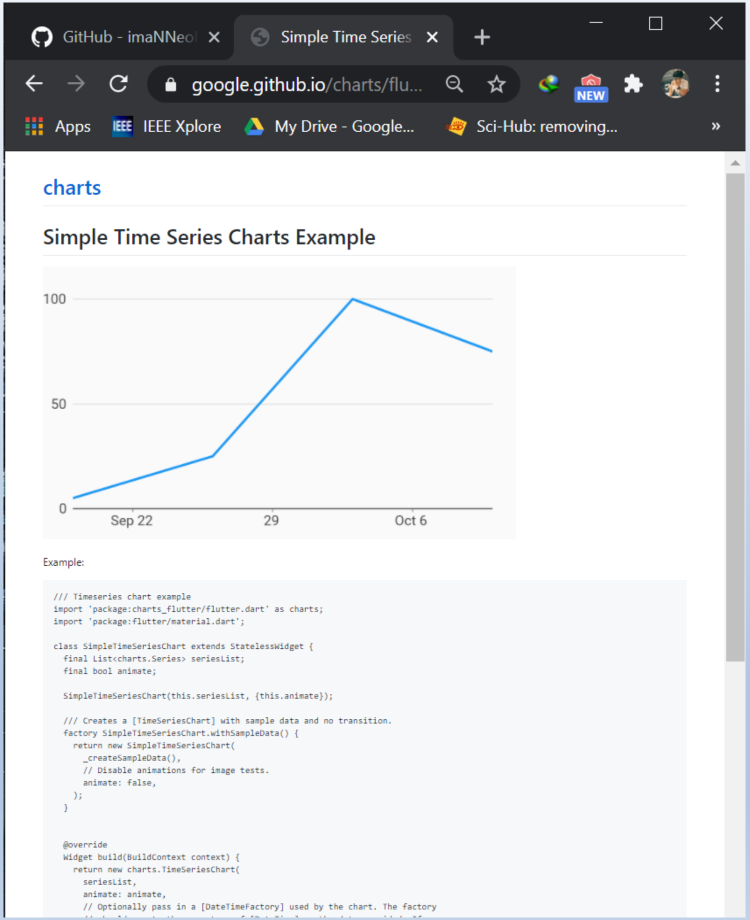
Gallery

Bar Charts

Simple	Stacked	Grouped	Grouped Stacked
Grouped Target Line	Stacked Horizontal	Stacked Target Line	Horizontal
Horizontal Bar Label	Horizontal Bar Label Custom	Vertical Bar Label	Spark Bar
Grouped Fill Color	Stacked Fill Color	Pattern Forward Hatch	Horizontal Pattern Forward Hatch
Grouped Stacked Weight Pattern	Custom Rounded Bars		

Time Series Charts

Simple	End Points Axis	Confidence Interval	Line Annotation
Range Annotation	Range Annotation Margin	Symbol Annotation	With Bar Renderer



charts

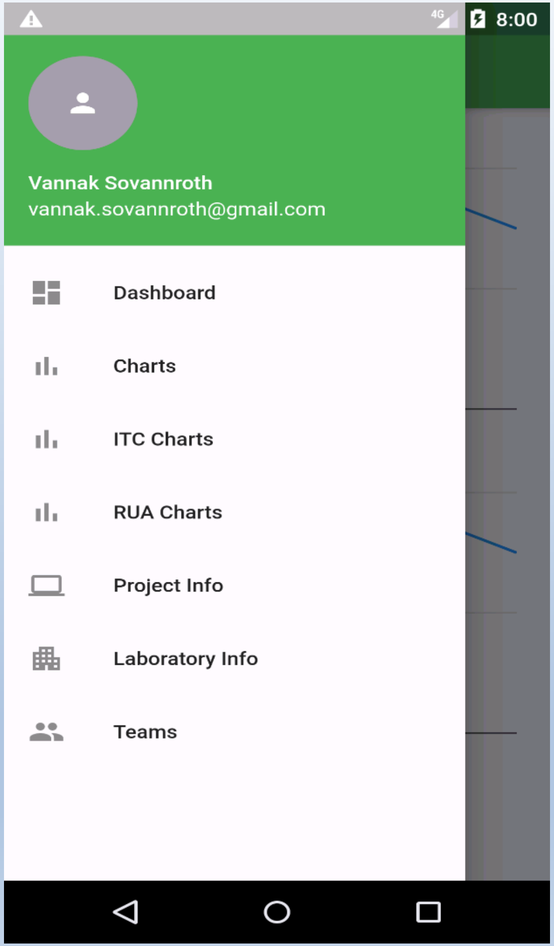
Simple Time Series Charts Example

Example:

```
/// Timeseries chart example
import 'package:charts_flutter/flutter.dart' as charts;
import 'package:flutter/material.dart';

class SimpleTimeSeriesChart extends StatelessWidget {
  final List<charts.Series> seriesList;
  final bool animate;

  SimpleTimeSeriesChart(this.seriesList, {this.animate});

  /// Creates a [TimeSeriesChart] with sample data and no transition.
  factory SimpleTimeSeriesChart.withSampleData() {
    return new SimpleTimeSeriesChart(
      _createSampleData(),
      // Disable animations for image tests.
      animate: false,
    );
  }


  @override
  Widget build(BuildContext context) {
    return new charts.TimeSeriesChart(
      seriesList,
      animate: animate,
      // Optionally pass in a [DateTimeFactory] used by the chart. The factory
```

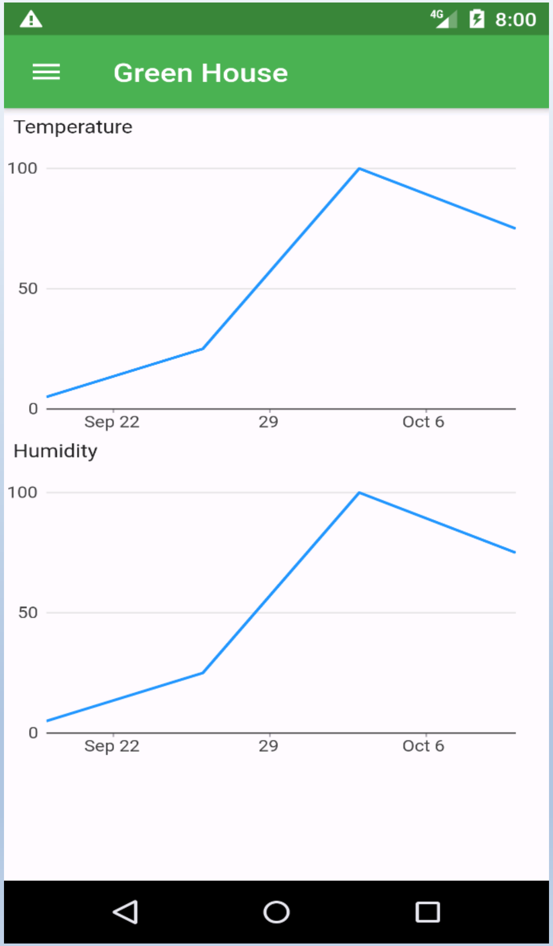
https://google.github.io/charts/flutter/gallery.htm

<<< View on Mobile Application >>>>



Vannak Sovannroth
vannak.sovannroth@gmail.com

- Dashboard
- Charts
- ITC Charts
- RUA Charts
- Project Info
- Laboratory Info
- Teams



Thank You!