INVESTMENT TRACKER

YOUR APPLICATION ONE

A guide to build your first applicaton

By
Santhujan Vijeyakumar

Contents

This book is dedicated to my family and professors. Cover page is from using canvas template.

Foreword

When it comes to investing there's various options available. Investor must analyze them and find what's the best method based on his local market and other varying factors. Most of the current apps in the market can help in tracking the income and expense of a person. investing means allocating money expecting profit in return. In most banks they provide high interest rate for the elders, in order to use this opportunity young people who are not eligible for a high percent interest will move their fixed deposits to their parents or grandparents' name. This also comes under the investment logic. This kind of investment cannot be tracked and analyzed like investing on real-estate or equity. The goal of this study is to identify key issues in investing and develop a cloud application which can track the various investment. This management platform will be built using core principles of investment so it can be used to track official and unofficial investment. As the result of this research from ordinary people to professional investors will be able to have their own personal financial consultant with total privacy. Data breach has been the end of many some of the largest companies. It's essential for the investing company or private investors to protect their financial data at all costs. The risk of financial data breach from third party sources will be reduced.

Preface

When I started my undergraduate, I always wonder how these coding theories are going to solve real life issues. over time I worked on various projects (mostly copied from github). I just never had a problem of my own to solve. Once I started working and I decided to invest my money for the future I realized there's application for tracking stocks market investments, real-estate and personal wallet. Most of the investments these days still follow the core investment principles. By building an application which uses the core investment tracking principles will allow it to track any type of investments.
Out of all the programming languages and technologies available I decided to choose dart, flutter because of its modern feature. When we build an application in dart we build once deploy it for ios, windows and android using an environment wrapper. This saves s ton of building time.
This will be a could solid application with a user-friendly build process.

Who is this book for?

If you're new to development this book can help you get a basic understanding of why certain feature are used. This project can help you set up your initial project. Each tool used in the project are explained. Some people have a theory understanding of building an application with no0 practical knowledge. By following this book, you can get your practical checked and have yourself a cool application.

Typographic conventions

This books consists of a lot of code snippet. Its can be identified using the bigger font and dark background.

```
1 sample code 1
```

```
2 sample code 2
```

```
3 sample code 3
```

“foldername”
The above text represents a folder name. The folders inside the root folder are taken as default the “/” is used to represent a sub folder.

Please help improve this book!

If this book helps you in understanding the core concepts of building an application kindly shar it with your peers and social media.

About the Author

Santhujan Vijeyakumar, developer who has worked in programming-based engineering companies. Santhujan is from Sri Lanka currently doing his master's in information technology at Sri Lanka Institute of Information Technology.

Introduction

Working title: Development of an Investment Management platform – itracker

Background

When it comes to investing there's various options available. Investor must analyze them and find what's the best method based on his local market and other varying factors. Most of the current apps in the market can help in tracking the income and expense of a person. investing means allocating money expecting profit in return. In most banks they provide high interest rate for the elders, in order to use this opportunity young people who are not eligible for a high percent interest will move their fixed deposits to their parents or grandparents' name. This also comes under the investment logic. This kind of investment cannot be tracked and analyzed like investing on real-estate or equity. The goal of this study is to identify key issues in investing and develop a cloud application which can track the various investment. This management platform will be built using core principles of investment so it can be used to track official and unofficial investment. As the result of this research from ordinary people to professional investors will be able to have their own personal financial consultant with total privacy. Data breach has been the end of many some of the largest companies. It's essential for the investing company or private investors to protect their financial data at all costs. The risk of financial data breach from third party sources will be reduced.

Significance of the Application

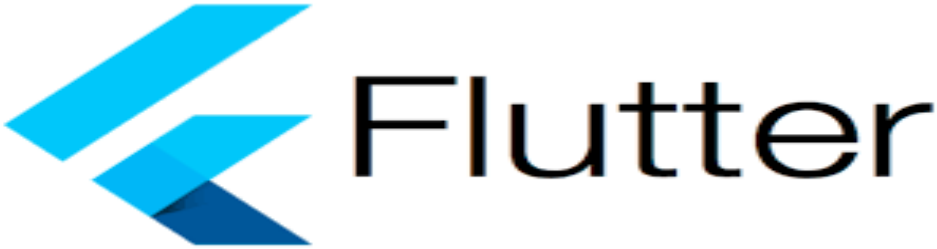
This study can help investors to manage their investment management anonymously without having to rely on a third-party financial service. This management system will be using the core principles of the investment and return technique. So, this system can be used to track business of any nature. This is not limited to the professional investors only. An ordinary person wants to make better financial decisions, for example comparing the groceries bought from various places over time then see which pattern had the most value. A crypto trader can use it track the variance of ROI (Return of Investment) from different crypto coins.

Technologies Used

Version control: GitHub

GitHub is one of widely used version controlling system. In this application we use GitHub to keep track of new features. Roll back if any of them don't work or not wanted anymore.

UI: Flutter



There is a lot of similar tools but what's special about flutter is it has the shortest learning curve. A person with vague programming knowledge can build and get it running in a very short time compared to other products available in the market.

Programming language: Dart

Dart is not the best programming language out there, programming languages like python and JS are much easier to learn. Only Dart is supported by the Flutter sdk.

Data base: PostgreSQL

PostgreSQL is one of the widely used database technologies and it provides high data protection.

Authentication: Google Firebase Authentication

Google fire base is relatively new when compared to his it competitors but firebase has a lot of out of the box features which makes it's a good option for beginners.

Deploying: Fire base hosting service

		SPARK Free Generous limits for hobbyists	FLAME $25/month Predictable pricing for growing apps	BLAZE Pay as you go Commodity pricing for apps at scale
Included Free Analytics, App Indexing, Authentication, Dynamic Links, FCM, Invites, Notifications, Crash Reporting, & Remote Config		✓	✓	✓
Realtime Database	Simultaneous connections	100	Unlimited [1]	Unlimited [1]
	GB stored	1 GB	2.5 GB	$5/GB
	GB downloaded	10 GB	20 GB	$1/GB
	Daily private backups	✗	✓	✓
File Storage	GB stored	5 GB	50 GB	$0.026/GB [2]
	GB downloaded	30 GB	50 GB	about $0.12/GB [2]
	Uploads & downloads	50,000 each	100,000 each	about $0.01/K [2]
Hosting	GB stored	1 GB	10 GB	$0.026/GB
	GB downloaded	10 GB	50 GB	$0.15/GB
	Custom domain hosting & SSL	✓	✓	✓
Test Lab	Device hours	✗	✗	$5/device hr
Google Cloud Platform	Use BigQuery & other IaaS[3]	✗	✗	✓

Since we are already using the google fire base service, we can use its hosting service as well.

Project Design

Planning

Let's see what's the goal of the project to list down the functional requirements we need.

- what if a person wants to track his investments in anonymously without the involvement of any third-party financial companies, consultancy?

- What if an investor wants to track multiple returns from different investments?

- How to prevent the financial data breaches?

- How can every person have their own personal financial consultancy?

  The application requirements are as follows

- The data should be stored in a cloud storage.

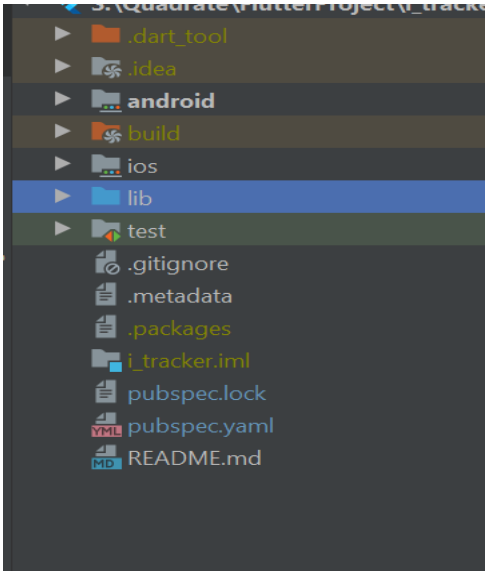
- The application should provide complete high-end security.

- Need a user-friendly dashboard.

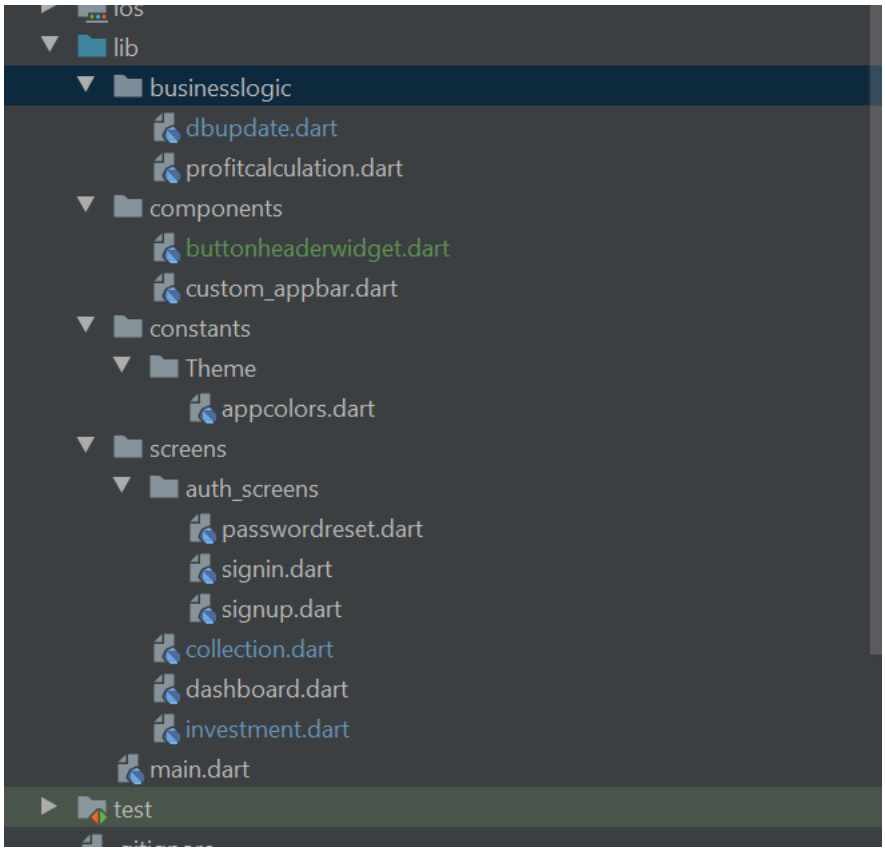
- The business logic behind this application should use the core principle of Investment management to make to appropriate for all kind of investments.

Project Files structure

This is the structure used in this project. All the files which serve similar purpose are grouped together with appropriate names.



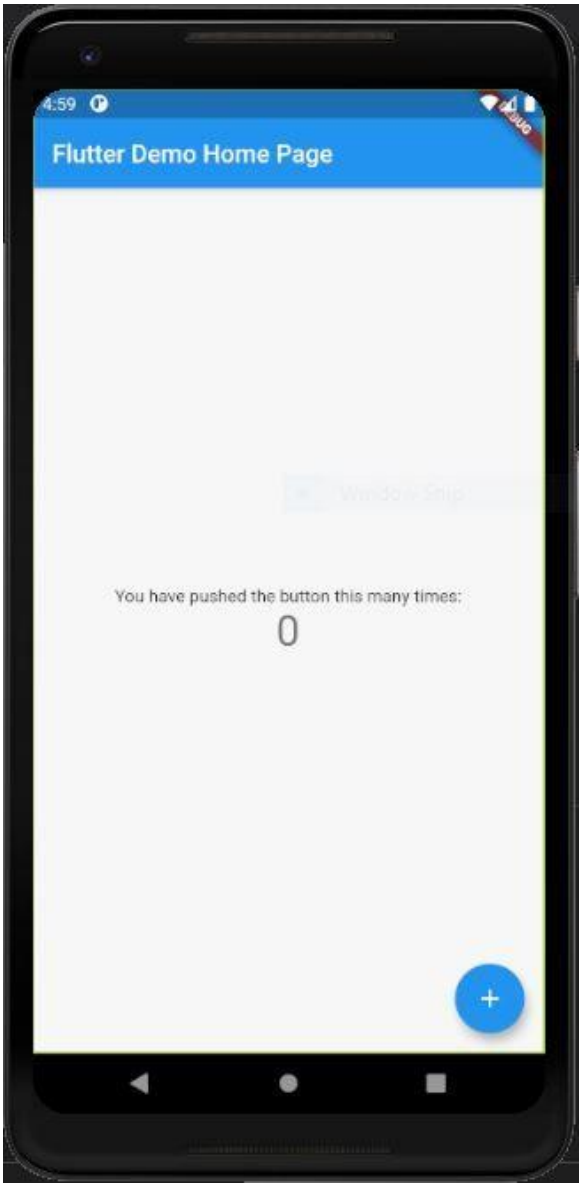
This is the default project structure came with the template. In this we split the files into four categories in four folders named businesslogic, components, constants, screens. Screens is future divided into authentication, which will contain the authentication screens like sign in, sign up and forgot password.


ios
lib
businesslogic
dbupdate.dart
profitcalculation.dart
components
buttonheaderwidget.dart
custom_appbar.dart
constants
Theme
appcolors.dart
screens
auth_screens
passwordreset.dart
signin.dart
signup.dart
collection.dart
dashboard.dart
investment.dart
main.dart
test

Step By Step Procedure

Standard Theme

We start the build by creating a GitHub repository named it “Itracker” and leave the description empty for now. Then open the android studio clone the GitHub repository. Open the AVD manager and select a device model with similar specs to devices in 2021. The android studio should provide a base demo template code. As shown in the figure bellow



Open the lib folder and create a folder name “constant”. In this folder we till add the things like theme color, Font style. Later during the projects when we are designing something new, we can

refer this file to have a constant design theme throughout the project. Create a dart file named “appcolors” and add the following code.

Appcolors

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

class ThemeColor {
  static const background = Colors.white;
  static const button = Colors.blueAccent;
  static const text = Colors.black;
  static const warning = Colors.red;
  static const grey = Colors.grey;
}
class StockInfo {
  static const gain = Colors.green;
  static const loss = Colors.redAccent;
  static const grey = Colors.grey;
}
```

These color themes can be accessed during the build by using “color: ThemeColor.button” or “color: ThemeColor.background”.

Standard Components

These days the requirements quality UI is a big deal. If the device screen size is same with a relative theme color, its necessary to have relative components. To achieve this, we can create a separate folder and create a dart file and customize each widget. In this application we create customs components style for appbar and button header.

MyAppBar

```
import 'package:i_tracker/constants/Theme/appcolors.dart';

import 'package:flutter/material.dart';

class MyAppBar extends AppBar {
  MyAppBar({Key key, Widget title})
      : super(
            key: key,
            title: title,
            backgroundColor: ThemeColor.button,
```

```
            actions: <Widget>[
              new IconButton(
                icon: new Icon(Icons.notifications),
              )
            ]);
}
```

ButtonHeaderWidget

```
import 'package:flutter/material.dart';
import 'package:i_tracker/constants/Theme/appcolors.dart';

class ButtonHeaderWidget extends StatelessWidget {
  final String title;
  final String text;
  final VoidCallback onClicked;

  const ButtonHeaderWidget({
    Key key,
    @required this.title,
    @required this.text,
    @required this.onClicked,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) => HeaderWidget(
        title: title,
        child: ButtonWidget(
          text: text,
          onClicked: onClicked,
        ),
      );
}

class ButtonWidget extends StatelessWidget {
  final String text;
  final VoidCallback onClicked;

  const ButtonWidget({
    Key key,
```

```
    @required this.text,
    @required this.onClicked,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) => ElevatedButton(
        style: ElevatedButton.styleFrom(
          alignment: Alignment.centerLeft,
          minimumSize: Size.fromHeight(40),
          primary: Colors.white,
        ),
        child: FittedBox(
          child: Text(
            text,
            style: TextStyle(fontSize: 15, color:
Colors.grey[600]),
          ),
        ),
        onPressed: onClicked,
      );
}

class HeaderWidget extends StatelessWidget {
  final String title;
  final Widget child;

  const HeaderWidget({
    Key key,
    @required this.title,
    @required this.child,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) => Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            title,
            style: TextStyle(
```

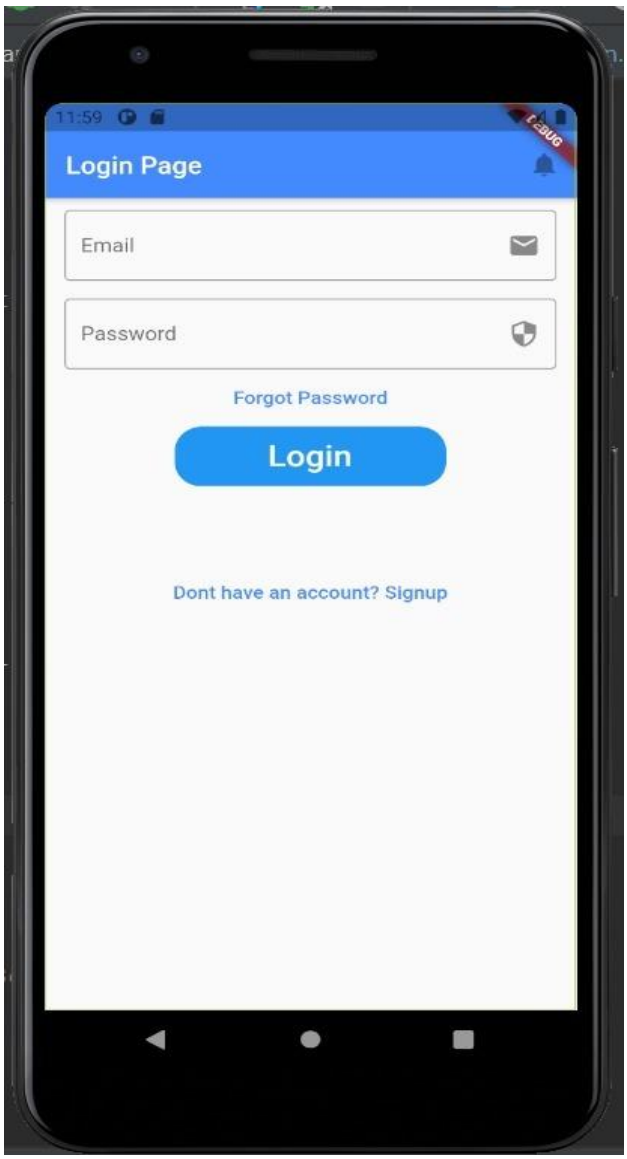
```
              color: Colors.red,
              fontSize: 20,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 8),
          child,
        ],
      );
}
```

Authentication Screens

Authentication covers the login pages, Email and password authentication, OTP. We start by creating a folder named “screens”. Create a sub folder named “auth_screens” to store all the dart files.

Sign in screen

When the user opens the application, credentials need to be authenticated. In the sign in Page there will be a two text boxes to enter the email and password, submit, forgot password buttons. Figure shows the login screen.



If the user’s password or email is incorrect an appropriate error message will be displayed.