

App Dev-Track

Build your own first mobile app in Flutter



Google Developer Student Clubs

Indian Institute of Technology Kanpur

Overview of the Project:

This project is designed to guide participants through the exploration and hands-on learning of Flutter and Dart, the powerful frameworks for building cross-platform mobile applications. Participants will start with the installation of the necessary tools, explore the basics of Dart and Flutter, build their first Flutter app, and delve into advanced topics such as Material Design, networking, state management, using plugins, and testing.

Tech Stacks:

- **Languages:** Dart
- **Frameworks:** Flutter

Project Timeline:

- **Week 1:**

Part 1: Installation

1. Try playing with DartPad, the online editor for Flutter and Dart.
2. Install the necessary tools using installation guides to set up the development environment.

Part 2: Explore Flutter

3. Get familiar with the Flutter documentation.
4. Visit the samples gallery.
5. Explore Dart code and understand the basics.
6. Explore Flutter conversion guides if coming from another SDK or language.
7. Learn what makes Flutter different, including using hot reload.

Part 3: Explore Dart & Flutter

8. Learn about Dart by visiting Dart.dev, the official site for Dart.

9. Understand why Flutter uses Dart and complete the Dart cheatsheet codelab.
10. Explore the concept that "Everything's a widget" by reading the Introduction to Widgets guide.

Part 4: Building your first Flutter App

11. Build your first Flutter app, part 1.
12. Continue building your first Flutter app, part 2.

- **Week 2:**

Part 5: Adding Material Design to your apps

13. Check out Material components.
14. Perform Material codelabs to enhance your understanding.
15. Explore additional learning resources, including videos and community-maintained indexes.

Part 6: Learn about Networking

16. Explore networking concepts in Flutter using the Flutter Cookbook.
17. Learn about the http package and how to deserialize JSON.
18. Explore a networking sample.

Part 7: State Management in Flutter

19. Watch an I/O talk on what "state management" means for Flutter.
20. Experiment with StatefulWidget and consider trying Provider with ChangeNotifier.
21. Watch relevant Boring Show episodes and explore state management samples.

Part 8: Using plugins to access native APIs

22. Read the guide on using packages and plugins.
23. Watch demonstrations of using plugins in the Boring Show.
24. Explore pub.dev, the package ecosystem for Dart and Flutter.
25. Learn about package scores and Flutter Favorites.

Part 9: Testing your code!

26. Read the guide to testing in Flutter.
27. Go through the testing sample, which demonstrates multiple types of tests.
28. Practice your skills with the Testing codelab.

- **Week 3 and 4:**

Project Building: Developing a useful application in Flutter

All Resources to be Used and Delivered:

1. **Learning Materials:** Curated tutorials, articles, and documentation.
2. **Hands-On Exercises:** Practical coding assignments to reinforce learning.
3. **Projects:** Real-world projects to apply acquired skills.
4. **Interactive Sessions:** Live sessions for doubt resolution and discussions.
5. **Feedback Mechanism:** Regular feedback and assessment to track progress.
6. **Final Project:** A capstone project to demonstrate proficiency in app dev concepts.

This structured project plan ensures a step-by-step exploration of Flutter and Dart, covering installation, basic concepts, app building, advanced topics, and practical application through real-world projects. Feel free to customize this template based on your specific project details and requirements.

Prerequisite:

Enthusiasm to learn and complete the project

Project Mentors:

Sahil Singh

[Linkedin](#)

[Github](#)

Video Playlist

https://www.youtube.com/playlist?list=PL_fkUfeFmd2u_VGZLSXjbfr--0xMhDCd5

Roadmap

<https://roadmap.sh/flutter>

Attendance Sheet

https://docs.google.com/spreadsheets/d/1k75PJnDP5mtk_gonA7ZDn7M6PVmbXXry0S0ClaSoq5s/edit#gid=0

Session Infos:

Session 1: 13/12/2023

<https://youtu.be/jA631509Gmw>

Agenda:

1. Introduction to Flutter:
<https://docs.google.com/presentation/d/17mmmCXEmjUdazJEdnM2jp9LdjNHLa0JeNg76aehj5ko/edit?usp=sharing>
2. Try playing with DartPad, the online editor for Flutter and Dart
3. Install Flutter : <https://flutter.dev/docs/get-started/install>

ASSIGNMENT 1:

1. Watch all the videos in this playlist:
<https://www.youtube.com/playlist?list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG>
2. Watch this Dart tutorial (1.5X recommended)
https://www.youtube.com/watch?v=Ej_Pcr4uC2Q
3. Read the Language, Effective Dart and Core Libraries from this documentation
4. <https://dart.dev/guides/libraries/library-tour>
5. Watch this video:
<https://www.youtube.com/watch?v=I-YO9CmaSUM>
6. Read about Hot Reload:
<https://docs.flutter.dev/tools/hot-reload>
7. Read about why Flutter uses Dart: [OPTIONAL]
<https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf>
8. Read what are Widgets in Flutter:
<https://docs.flutter.dev/ui>
9. Programming Assignment on DART
<https://docs.google.com/document/d/1vxCJEAroiR69n0q7R-JXPnm2HrdFfYefnV948aYhOKg/edit?usp=sharing>
(Don't use ChatGPT at the beginning of your journey, at least have the basic knowledge right)

DEADLINE of all the above tasks is **18th December EOD**

Session 2: 16/12/2023

<https://youtu.be/3UeleAwLyK4>

Agenda:

1. Building your First Mobile Application:
https://docs.google.com/presentation/d/1hkl5NruLYkrHky5nC1AU_544Pcqvl9tvd4c0FetTU6U/edit?usp=sharing
2. CodeLab: <https://codelabs.developers.google.com/codelabs/first-flutter-app-pt1/#0>

ASSIGNMENT 2:

1. Build this application:
<https://drive.google.com/file/d/1VrxrQbzMuq-tGfsFY3uoMWZuUxN4H9rA/view?usp=sharing>

CodeLab Reference:

<https://codelabs.developers.google.com/codelabs/flutter-codelab-first#0>

2. Complete this codelab and build this application: [OPTIONAL]
<https://codelabs.developers.google.com/codelabs/flutter-boring-to-beautiful?hl=en#0>

Submission Link: <https://forms.gle/2qJFAB5EmtndwWir7>

Deadline: 22 December 2023 EOD

Session 3: 21/12/2023

<https://youtu.be/Q7ZomzYOLqI>

Agenda:

1. Material Design and Navigation:
<https://docs.google.com/presentation/d/1zXPz3NJeV-zxYHa31sLvoh6nzCQ4P1bdJKAYkIc4yZk/edit?usp=sharing>
2. Hands on Demo on adding Material Design and using Navigation in your application

ASSIGNMENT 3:

1. Create an application in Flutter with the following functionalities:
 - The app will have 2 pages (screens to navigate)
 - You can use either page based or route based navigation
 - The default page should have 4 input fields (use TextField): Name, Email, Roll No and Phone Number, and one button which should say 'Show Details'
 - When this button is clicked you should navigate to a different page, where you should display the 4 details of the user on this second page. Pass this data through Navigation from the first page. When the user goes back to the previous page, then there should be a popup on the screen saying that 'User {username} Logged out'.
 - There is no fixed UI scheme that you have to follow. Just stick to basic Material Design and complete the assignment.
 - Make sure you upload the assignment video by entering your details in the application.
2. Create the application of which the demo was given in Session 4. Include these features:
 - Shows all the list of albums from get request
 - Add a new album from post request
 - Delete any album. The delete icon should be present to the right end of the list tile.
 - Change the UI and explore a bit and make your UI a bit creative and unique

Submission Link: <https://forms.gle/jUhxtSNbK1D11GE6>

Deadline: 30 December 2023

Session 4: 25/12/2023

https://youtu.be/ni_bd-OIHBs

Agenda:

1. Networking:
<https://docs.google.com/presentation/d/1duwrYGuGpDx82DEETpncE7PwxumM4YFU4d78Vtt3-R0/edit?usp=sharing>
2. Hands on Demo on adding Networking in your application

ASSIGNMENT 4:

None: Merry Christmas and Enjoy your new year.

Session 5: 05/01/2024

<https://youtu.be/u5jqkAFXJdY>

Agenda:

3. State Management using Providers:
https://docs.google.com/presentation/d/1V8QCKTSSdYuUXd-21i_tPxkLxTRCB1UDSGArVK2L0OU/edit?usp=sharing
4. Hands on Demo on adding Providers in your application

ASSIGNMENT 5:

1. Watch all the videos mentioned in the slides:
 - a. <https://www.youtube.com/watch?v=vU9xDLdEZtU>
 - b. https://www.youtube.com/watch?v=d_m5csmrf7I
 - c. <https://www.youtube.com/watch?v=m2hWRdTBLQ8>
 - d. <https://www.youtube.com/watch?v=MkFjtCov62g>
 - e. https://www.youtube.com/watch?v=zYdl_Lb-rj0
2. Watch this playlist from Video 1 - Video 10:
<https://www.youtube.com/watch?v=IU4p8yhMghc&list=PLFyjjoCMAptzn7tFLRV3eny7G74LnIMRt>
3. Create a part of a e-commerce application using Provider.
 - a. The application will contain two screens (use Navigation to move between the two screens)
 - b. The launch screen will be a listing product page, which will have a list of products (can be anything for example mobile phones). You will have to hardcode this data. There should be atleast 10 entries here (maximum 20). The product row should look like this:
 - i. Product Image (take any image from internet, use asset image: download and store it in the assets folder and then use in app: google it)

- ii. Product Name
- iii. Product Price
- iv. Add to Cart icon button: if added to cart, a tick icon and a minus(remove) icon button should be shown
- c. The launch screen will also have a cart icon at the right side of the app bar.
- d. When this app bar is clicked, it should navigate to the next screen, which will display all the products added to the app with their image, name and price. There should also be a total price text and a BUY button at the bottom of the screen. When this BUY button is clicked, the cart should get empty.
- e. You should use Provider to implement this.
- f. Create a data model of the Product (image, name, price)
- g. Create a list of the data model of the Product and then use it in the Provider to add to cart or remove.
- h. Refer to the session 5 video at the end, assignment is discussed there
- i. No design constraints, it is up to your creativity, just follow the basic skeleton
- 4. Watch this video about Unit Testing: <https://www.youtube.com/watch?v=Cdl0z-xhZks>
- 5. Complete this short codelab on Testing in Flutter:
<https://codelabs.developers.google.com/codelabs/flutter-app-testing?hl=en#0>

Submission Link: <https://forms.gle/W9nbW3An15TLkK6NA>

Deadline: 14th January 2024 (long assignment, please start as soon as possible)

Submit Doubts here: <https://forms.gle/Ds4n3QJXSfuA3PSq8> (to be taken in next doubt session)

Session 6: 07/01/2024

<https://youtu.be/nB6RJDL5MJE>

Agenda:

1. Doubt Clearing and Final Remarks

Quizzes

Quiz 1:

<https://forms.gle/ySJVedEBTtd6oF3R8>

Quiz 2:

<https://forms.gle/Jps2fAG7b2ERCuAZ8>

Final Project: OPTIONAL

- You can do this project in a group too
- This is an optional assignment
- You are encouraged to do this project
- This project will require a bit of learning and a lots of implementation

Build a Chat Application with Firebase Cloud Firestore. Login the user using Firebase Authentication. UI can be up to your creativity.

Submission Link: <https://forms.gle/cKnnVgzKxenZz1xz9>

If you have decided to complete this final project, please try to complete it before January 30th, 2024

References:

- Firebase Tutorial: <https://www.youtube.com/watch?v=DIN07bJ0pFM>
- Firebase Flutter:
https://www.youtube.com/watch?v=kwqb-6QyYt8&list=PLjVLYmrlmjGeA6_i1WOallrMbTzZtBcp8
- Firebase with Flutter Demo: <https://www.youtube.com/watch?v=u8H652UY-L8>
- Chat App with Firebase:
<https://www.youtube.com/watch?v=jQ5S4yg6pME&list=PLr7P7IMIUTusNB85U2dW9na-aB7zFkcmI>
- Cloud Group Chat Github Repo (written by me):
<https://github.com/Sahilsingh0808/Cloud-Group-Chat>
- <https://blog.logrocket.com/how-to-build-chat-application-flutter-firebase/>
- <https://medium.com/flutter-community/building-a-chat-app-with-flutter-and-firebase-from-scratch-9eaa7f41782e>

Additional Resources:

Part 1 : Install

- 1) Try playing with [DartPad](#), the online editor for Flutter and Dart
- 2) Install Use the [installation guides](#) to help get your environment ready

Part 2 : Explore about Flutter

- 3) [Get to know the Flutter docs](#)
- 4) Visit [the samples gallery](#)
- 5) [Get a feel for Dart code](#)
- 6) Coming from another SDK or language?
 - [Flutter conversion guides](#)
 - [What Makes Flutter Different](#)
 - [Using hot reload](#)

Part 3 : Explore Dart & Flutter

- 7) Learn about Dart
 - Visit [Dart.dev](#) the official site for Dart
 - Learn [why Flutter uses Dart](#)
 - Complete the [Dart cheatsheet](#) codelab
- 8) Everything's a widget!
 - Read the [Introduction to Widgets](#)
 - [This guide](#) explains how to compose [layouts](#) with widgets

Part 1 : Building your first Flutter App

- [Building your first Flutter app, part 1](#)
- [Building your first Flutter app, part 2](#)
- <https://codelabs.developers.google.com/?cat=Flutter&text=flutter>

Part 2 : Adding [Material Design](#) to your apps

- Check out [Material components](#)
- Perform these [Material codelabs](#)

Additional Learning

- Try [Johannes Milke's videos](#)
- Or [Super Declarative's widget workshops](#)
- Or a community-maintained index like [Awesome Flutter](#)

Part 3 : Learn about Networking

- Have you seen the Flutter Cookbook? It has [a set of articles on networking](#)
- Many apps use the [http package](#)
- Learn about [how to deserialize JSON](#)
 - There's also a [sample](#)!

State Management in Flutter

- Watch [an I/O talk on what "state management" means](#) for Flutter
- You'd be surprised how far you can get with just a [StatefulWidget](#)
- A good next step is to try [Provider](#) with [ChangeNotifier](#)
- Watch this [Boring Show episode](#)
- Or these samples: [provider_counter](#), [provider_shopper](#)

Part 1 : Using plugins to access native APIs

- Read [the guide on using packages and plugins](#)
- Watch [Matt and Andrew use the barometer](#) on the Boring Show
- Check out [pub.dev](#), the package ecosystem for Dart and Flutter
 - [Learn about package scores](#)
 - ...and the [Flutter Favorites](#)

Part 2 : Testing your code!

- Read the [guide to testing in Flutter](#)
- Read through [the testing sample](#), which shows multiple types of tests
- Practise your skills with the [Testing codelab](#)