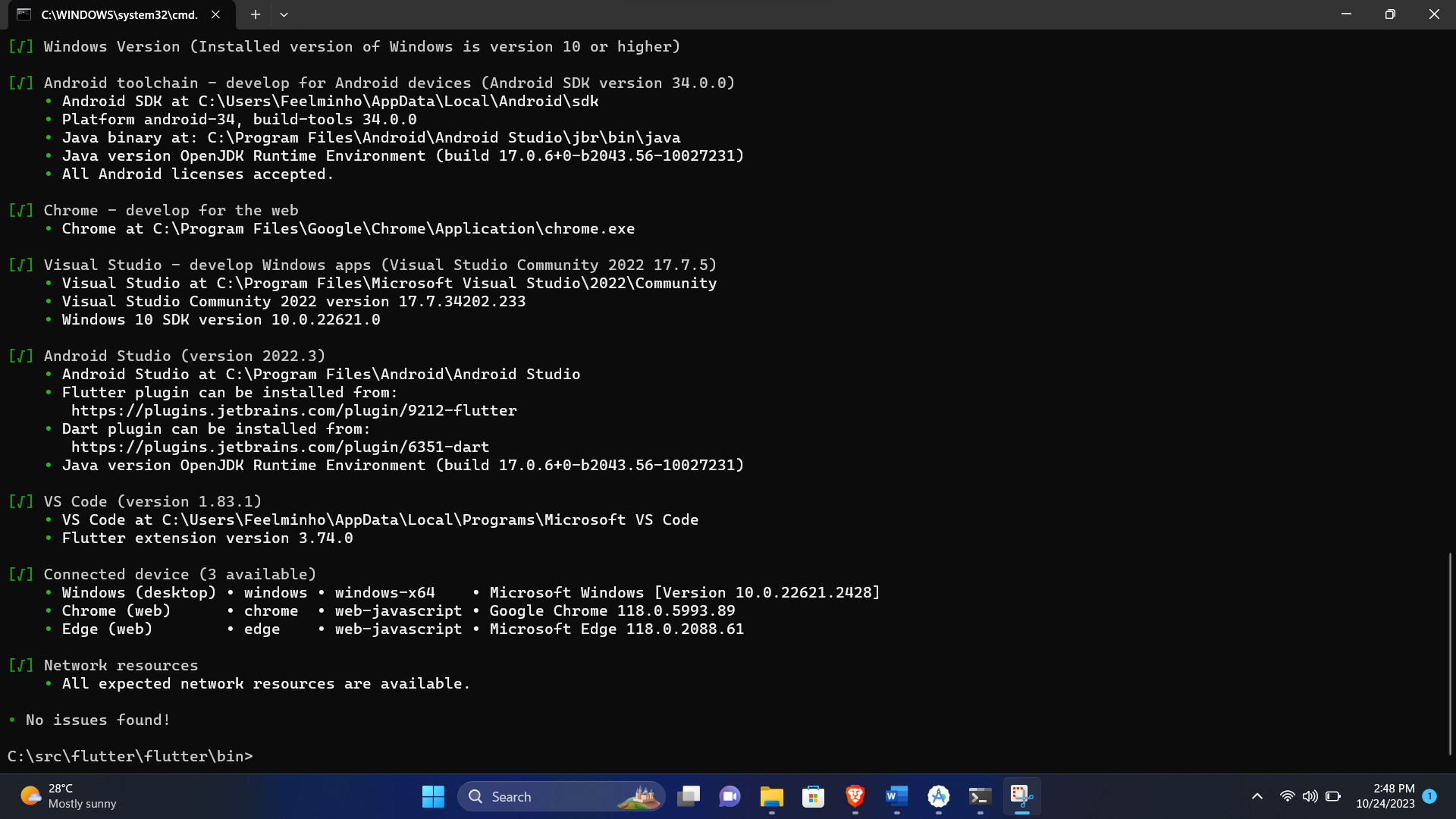
# **TP-Session-1**

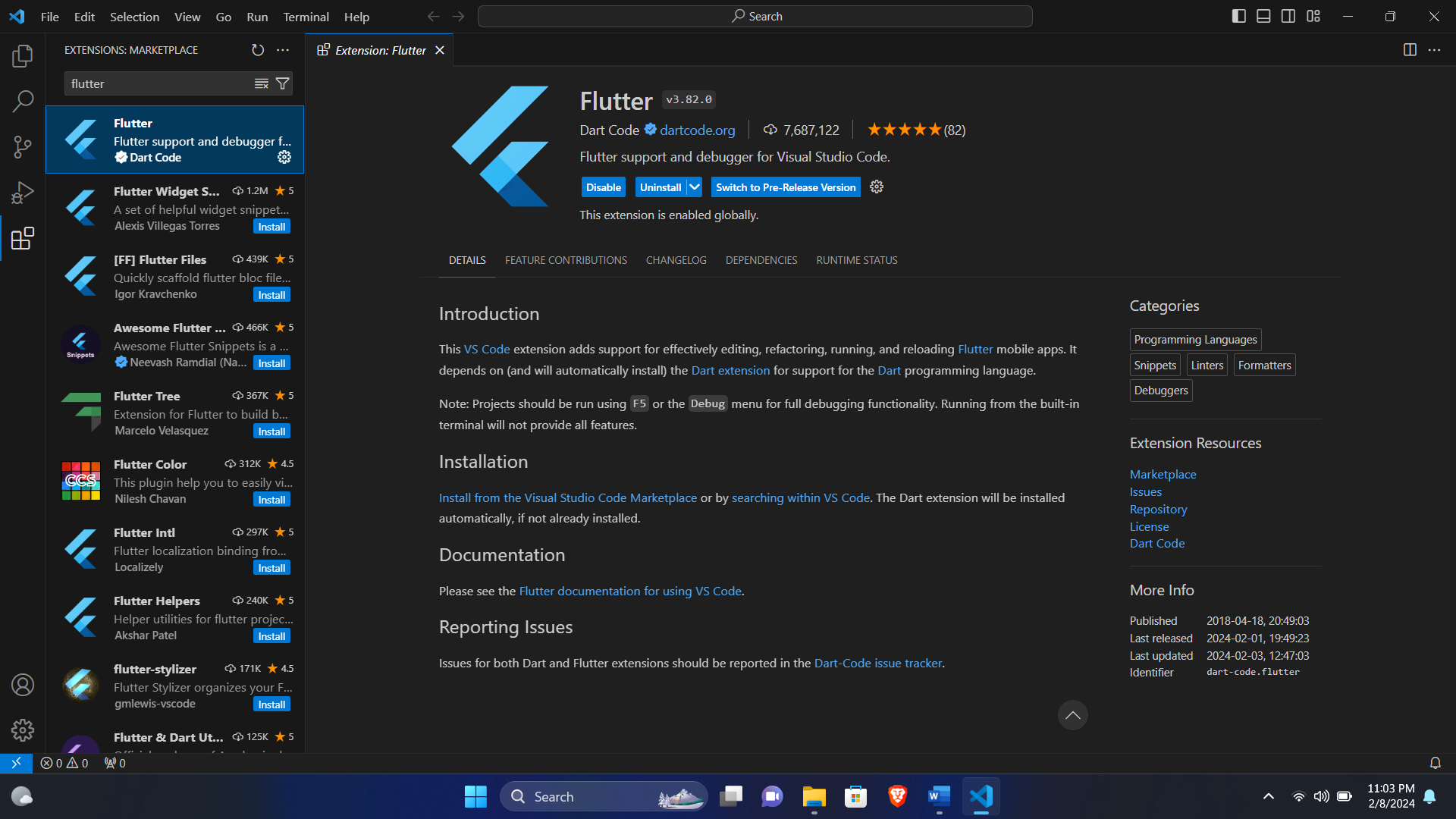
**Lab session 1: Flutter Installation and UI Design using Flutter**

**Task-1**

1. **Vs code**

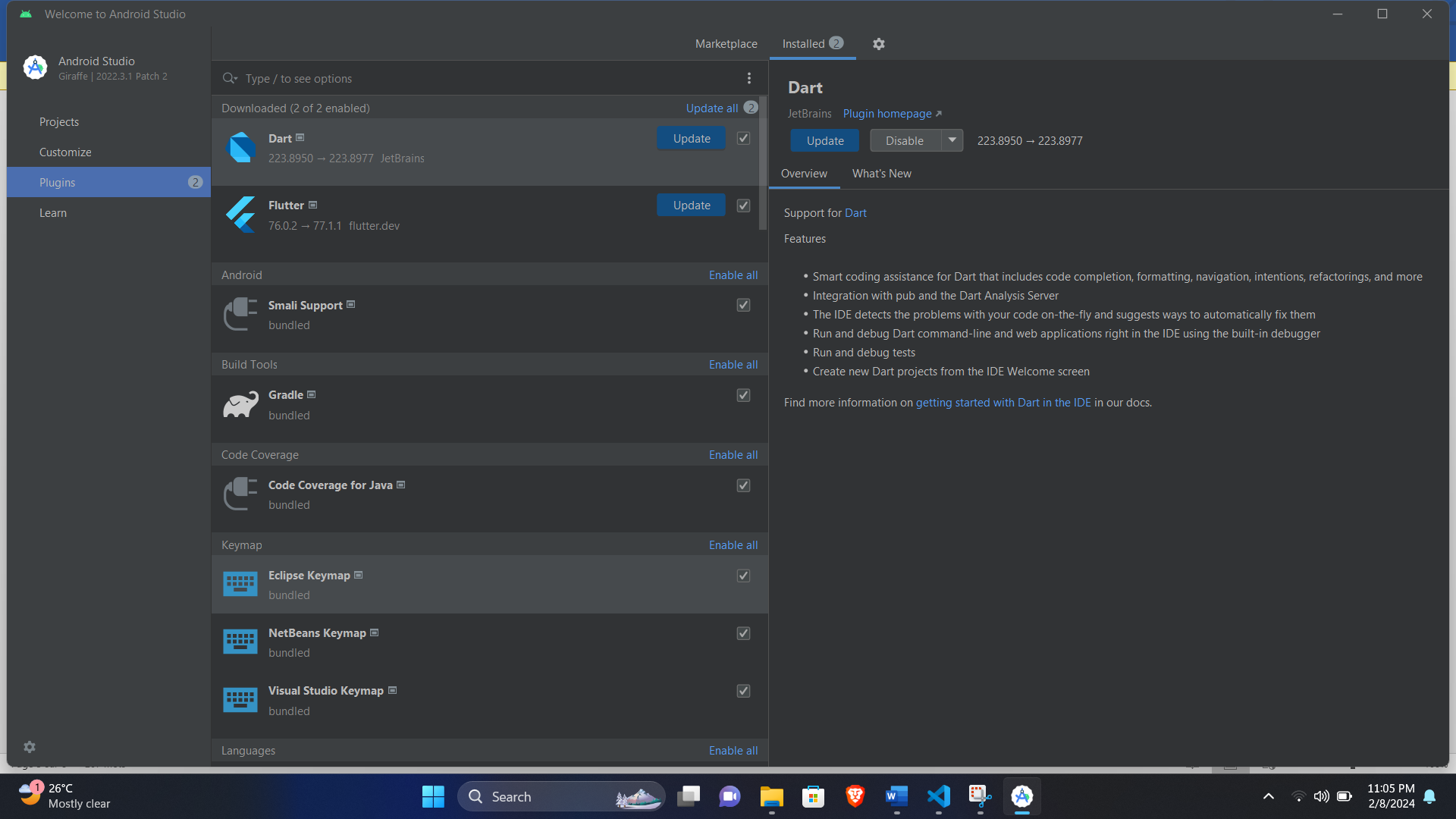
**C:\src\flutter\bin\Flutter doctor -v**





* 1. **Description for the installation**
* Install Visual Studio Code
* Flutter relies on the Dart programming language, so you need to install the Dart SDK. Follow the instructions on the Dart SDK official website: Dart SDK.  
    
  Add the Flutter binary directory to your system's PATH environment variable. This allows you to run Flutter commands from any terminal window.
* Install Flutter Extension in VS Code:  
  Go to the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of the window or press **Ctrl+Shift+X**.  
  Search for "Flutter" in the Extensions view search box.

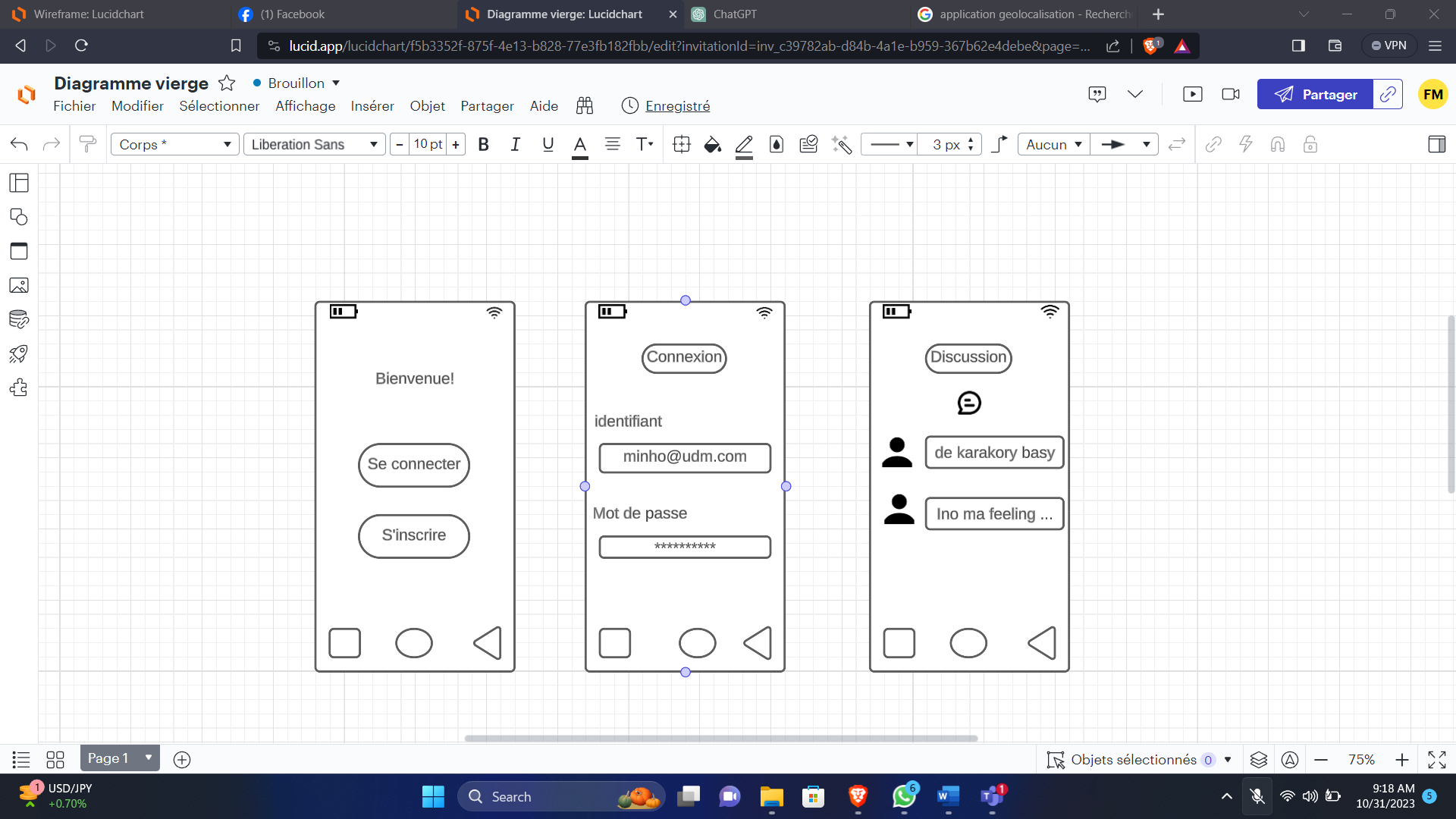
1. **Android Studio**



**2.2- Description for the installation**

* Download and install Android Studio
* **Install Flutter plugin in Android Studio:** Open Android Studio, go to "File" > "Settings," select "Plugins," search for "Flutter," install it, and restart Android Studio.
* **Download and extract Flutter SDK:** Get it from Flutter SDK and extract the downloaded archive to a location of your choice.
* **Set up environment variables:** Add the Flutter binary directory to the system's PATH
* **Install Flutter dependencies:** Open a terminal in Android Studio, navigate to your Flutter project directory, and run **flutter pub get** to install project dependencies.

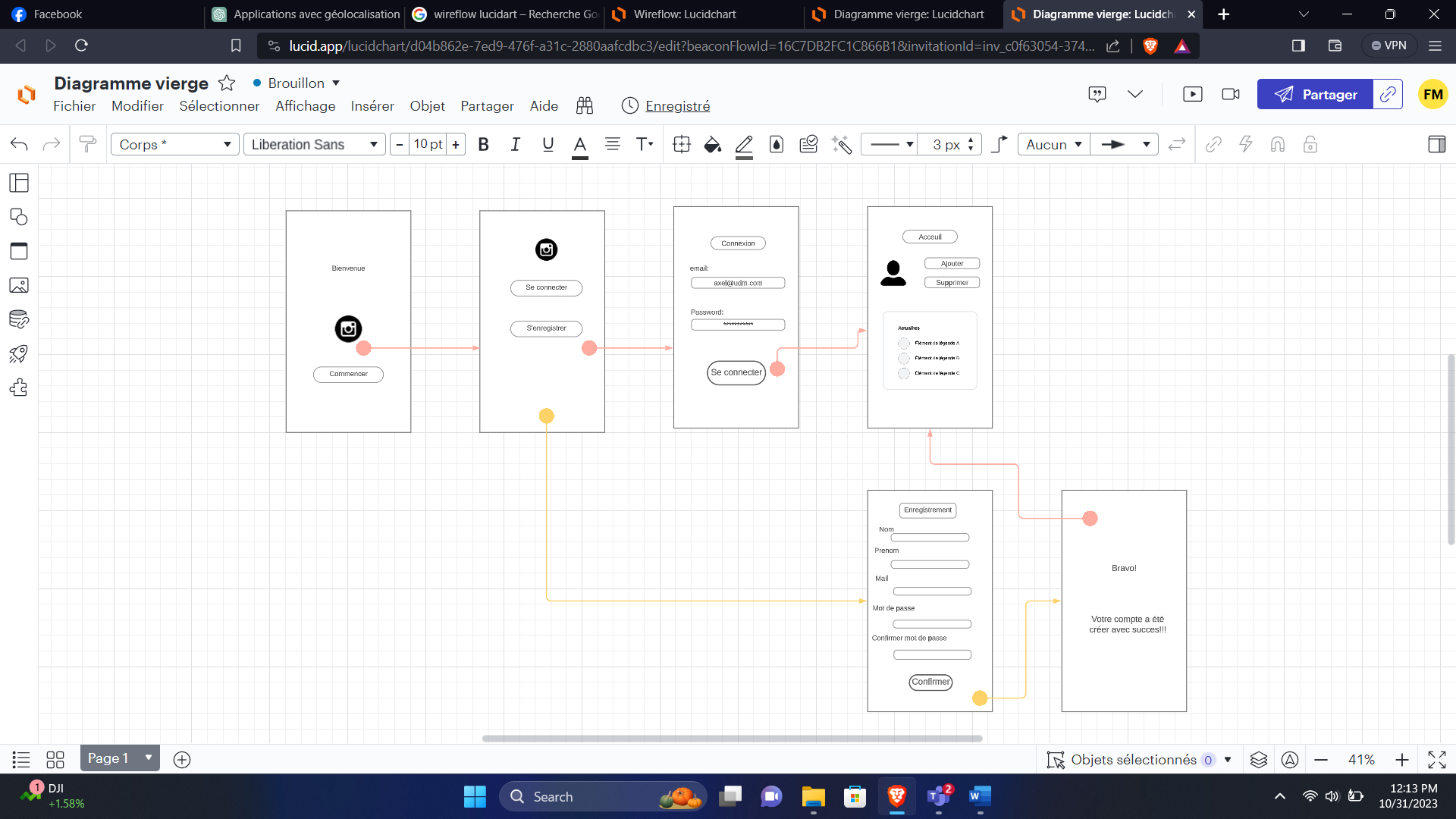
**Task-2: Using a Lucidchart to create a wireframe for a Flutter App**

****

**A short description and screenshots for all the steps of the lab:**

* Sign in to Lucidchart
* Once signed in, click on the "Create" button to start a new document. Select "Blank Document" to create a wireframe from scratch.
* Use basic shapes to represent buttons, text fields, images, and other UI components. Customize each element according to your app's design.

**Task-3: Using an appropriate online tool create a wireflow for a Flutter App**

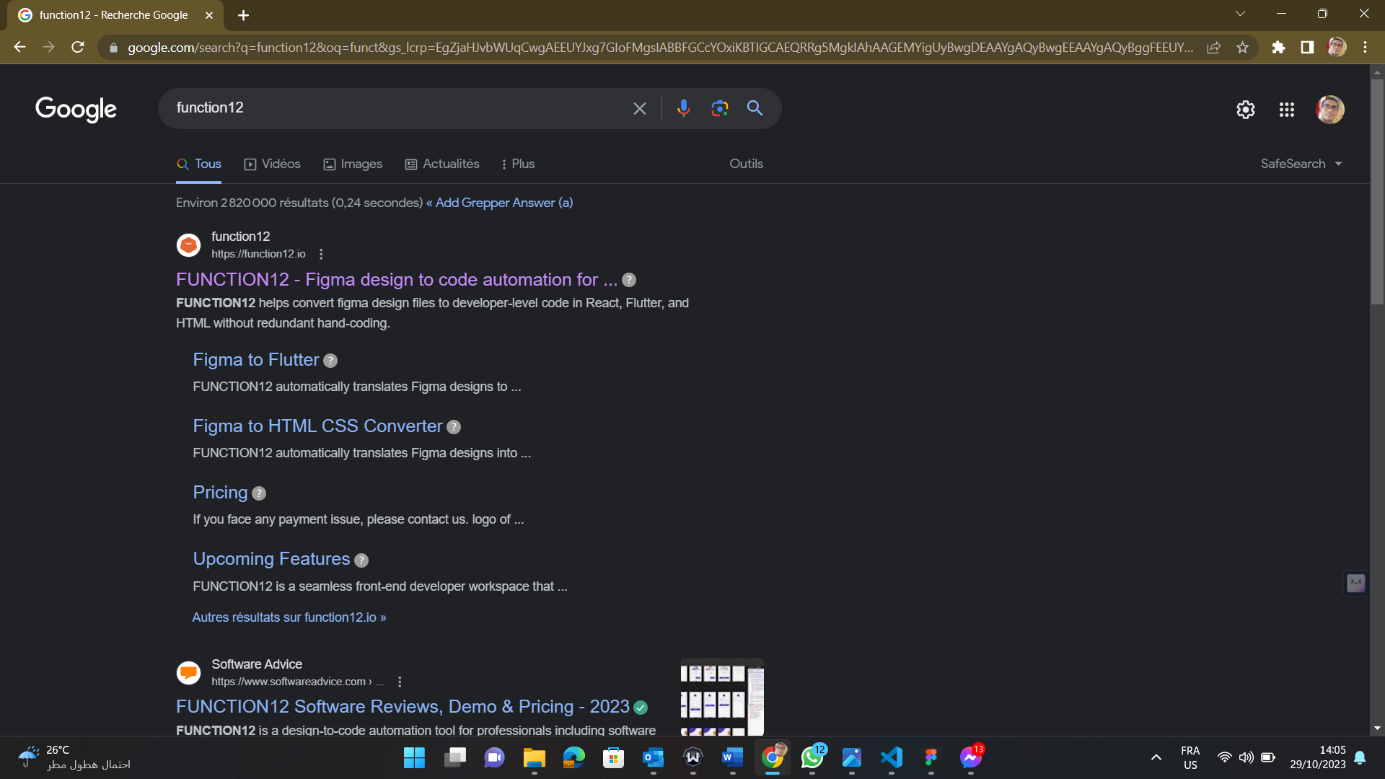
****

**A short description and screenshots for all the steps of the lab:**

* Sign in to Lucidchart
* Once signed in, click on the "Create" button to start a new document. Select "Blank Document" to create a wireframe from scratch.
* Use basic shapes to represent buttons, text fields, images, and other UI components. Customize each element according to your app's design.
* Use lines and arrows to connect elements and represent the flow of your app. Create a logical sequence of screens and interactions.
* Include annotations and descriptions to provide additional information about specific elements, interactions, or functionalities.

**Task-4: Using an appropriate online tool create a prototype for a Flutter App**

* Visiter le site “Function12”

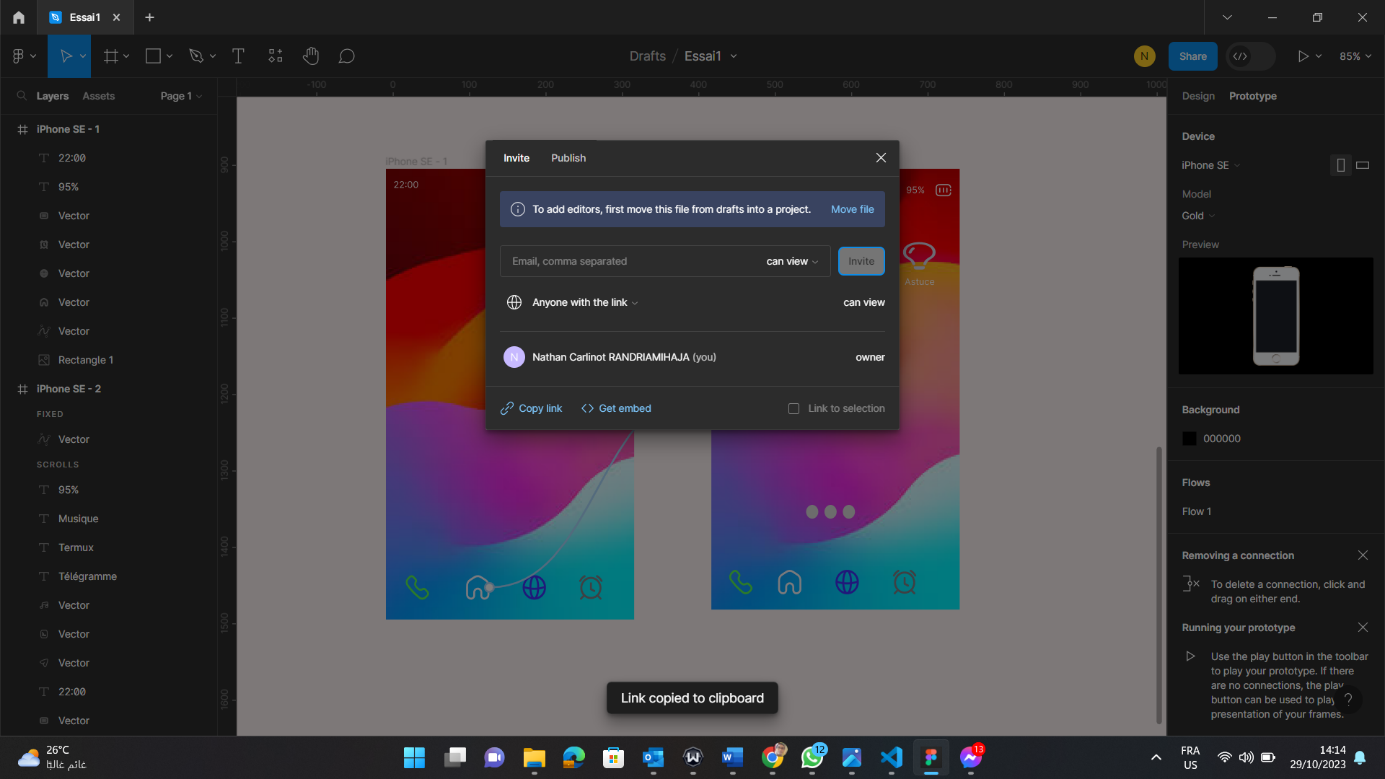


* Traduire le code à l’aide de l’URL sur Figna

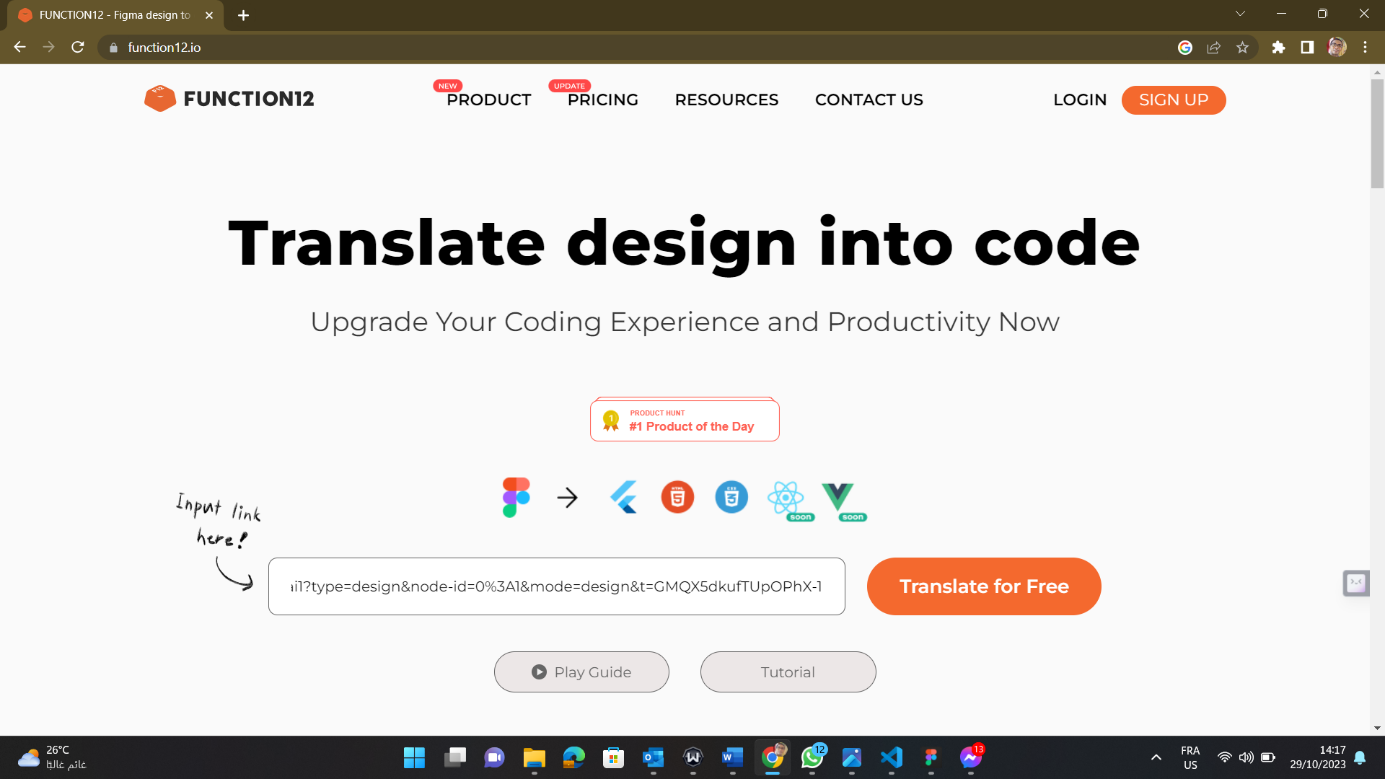
A screenshot of a computer

Description générée automatiquement

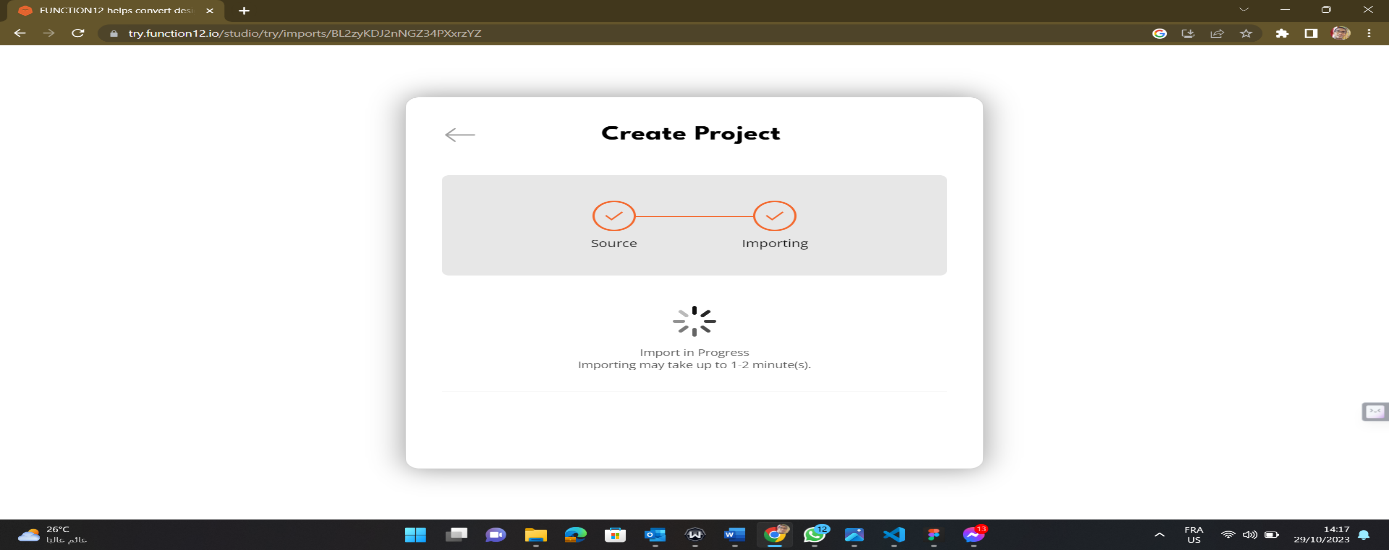
* Prendre le lien sur Figma, clique sur le bouton « Share » puis « copy link »

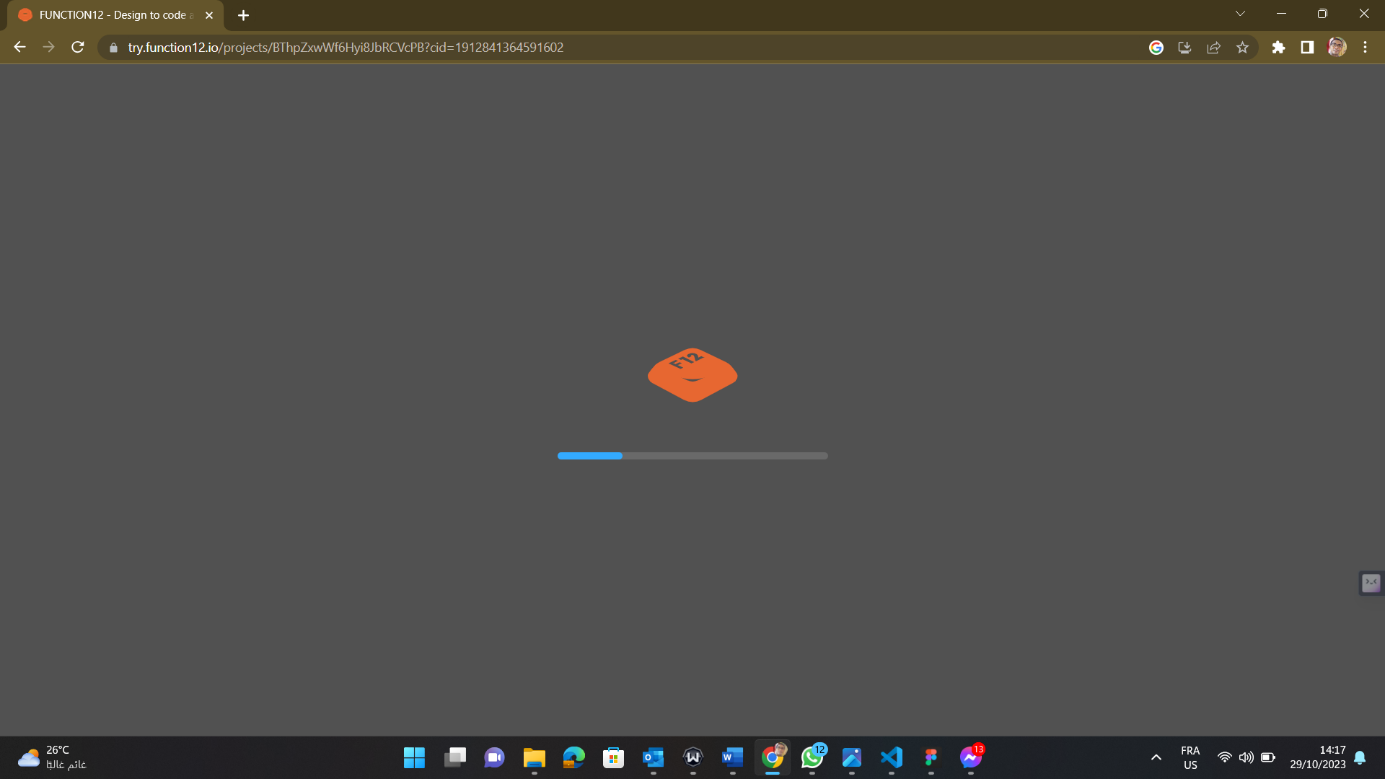


* Une fois copié l’URL sur figma, on le met sur « function12 » puis, on clique sur « Translate for free »



* Les lancements



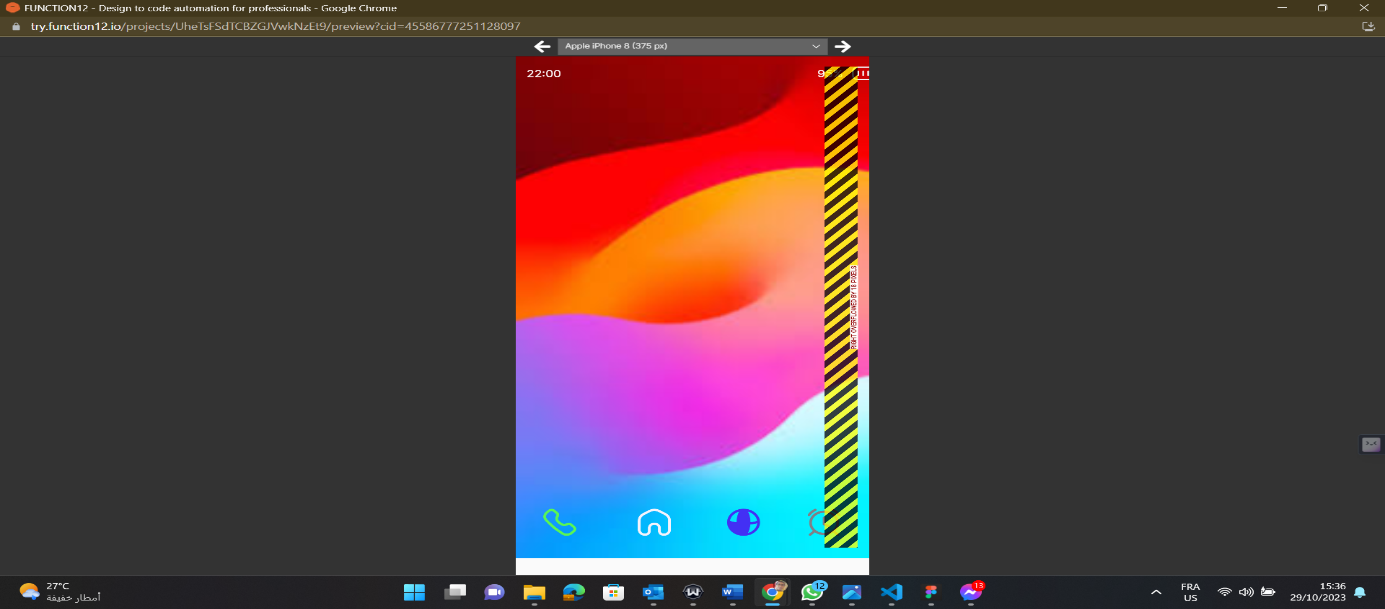


* Cocher sur le logo flutter puis lancer

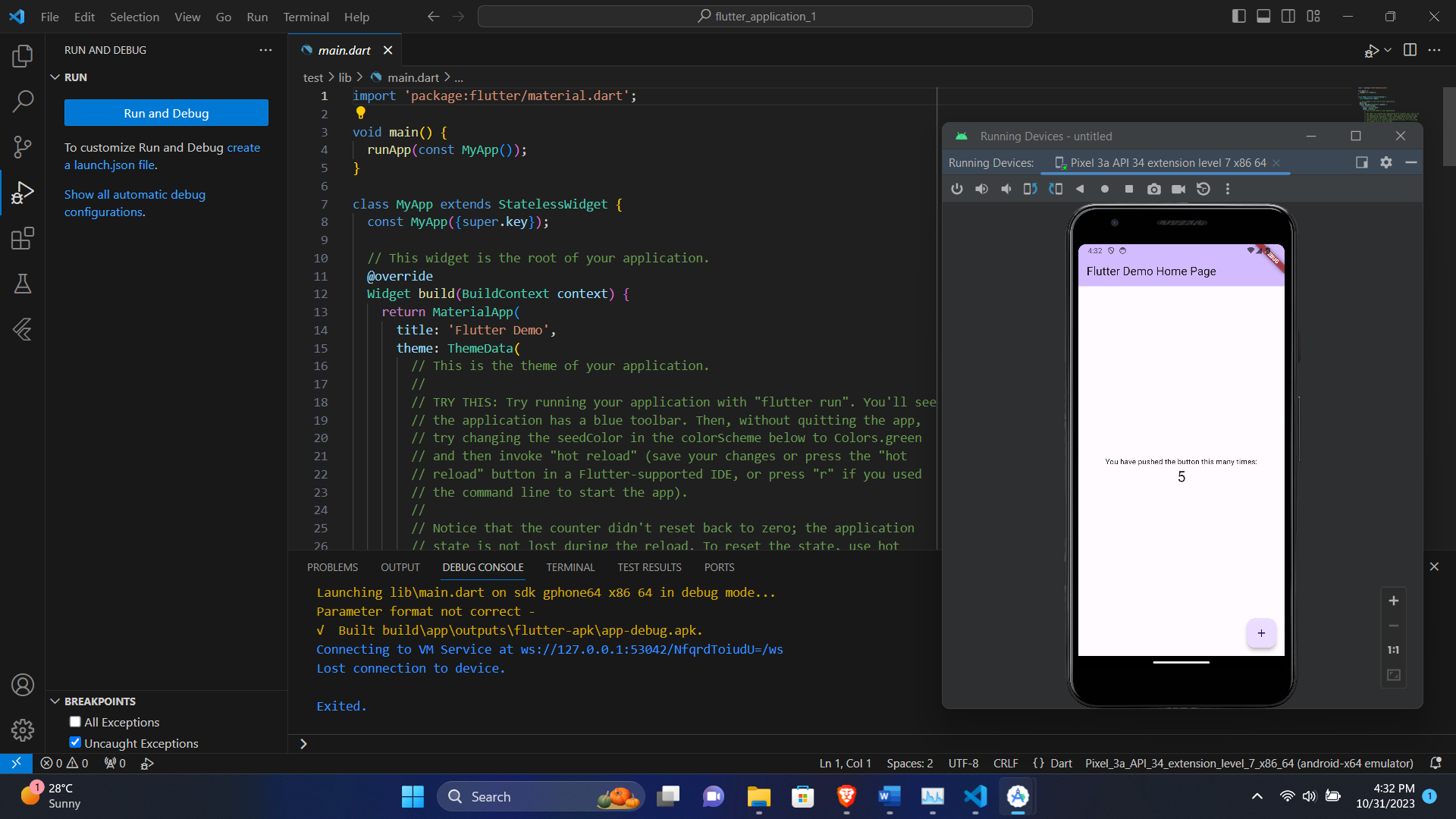
A screenshot of a computer

Description générée automatiquement

* Réalisation finale



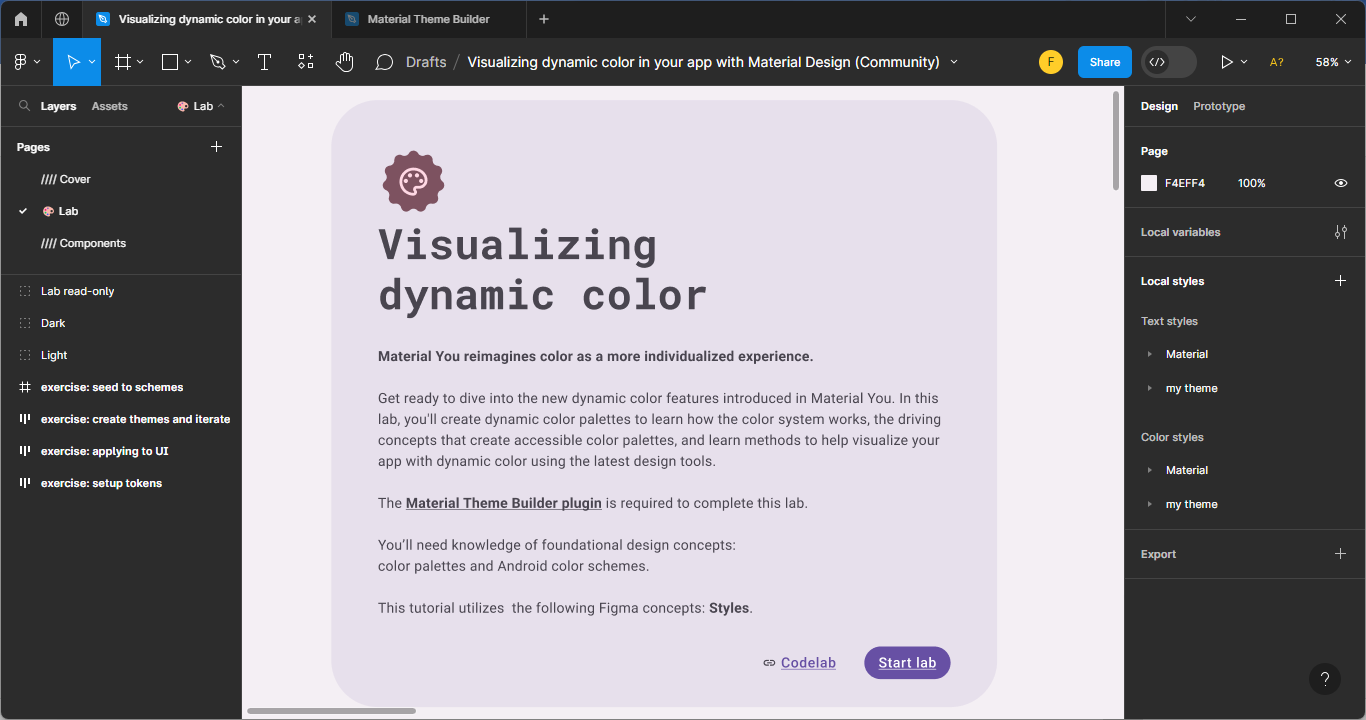
**Task-5: Create your first Flutter App in VSCode**

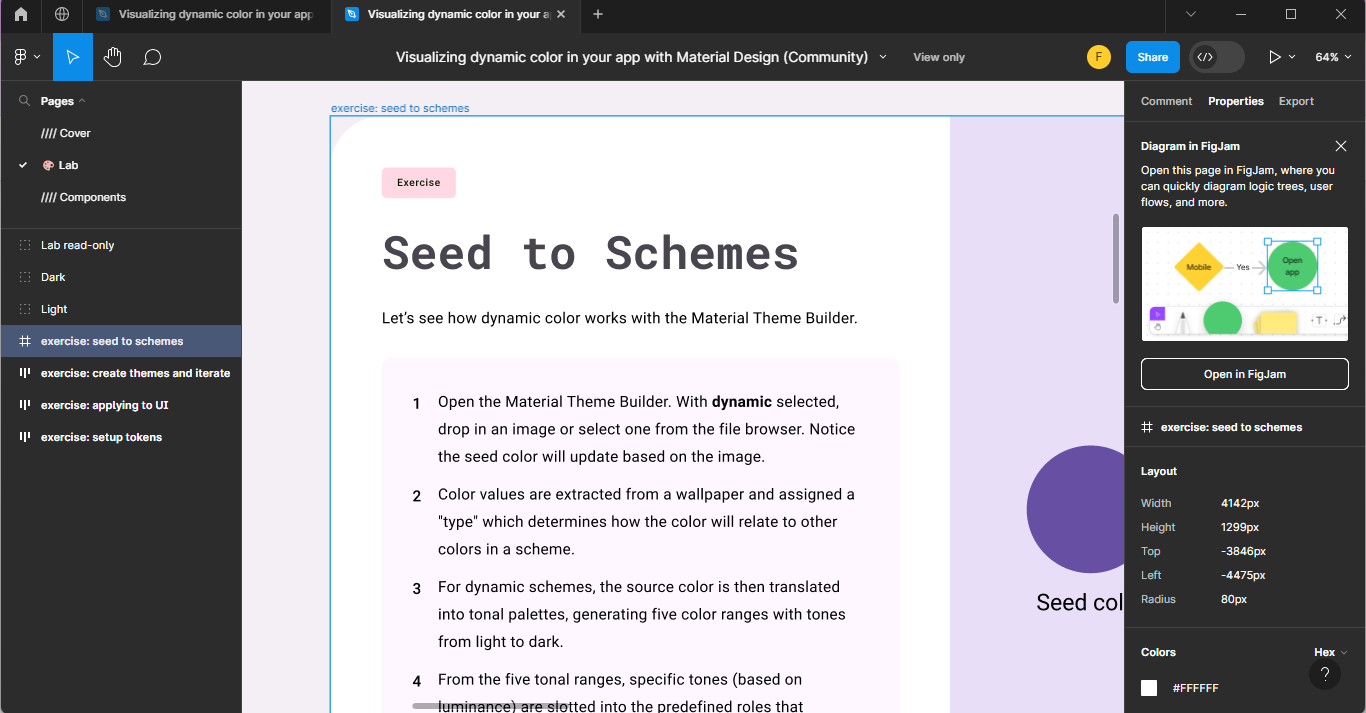
****

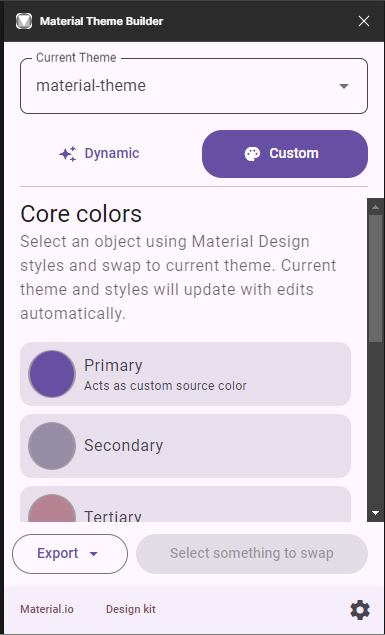
**A short description and screenshots for all the steps of the lab.**

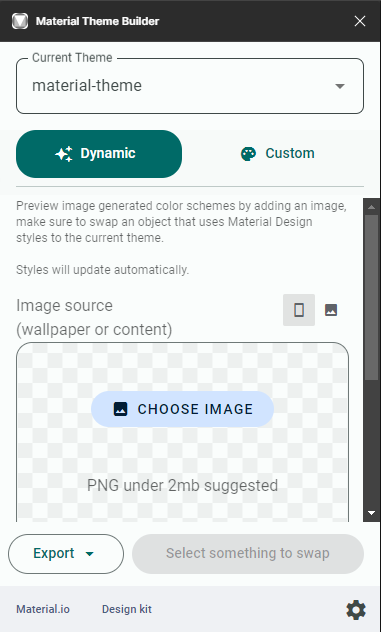
* Download and install Visual Studio Code
* Install the Flutter and Dart extensions for VSCode. Open VSCode, go to the Extensions view (Ctrl+Shift+X or Cmd+Shift+X), and search for "Flutter" and "Dart." Install both extensions.
* Open the Command Palette (Ctrl+Shift+P or Cmd+Shift+P) and run the command Flutter: New Project.
* Familiarize yourself with the project structure. Key folders include lib (for Dart code) and test (for unit tests). The main.dart file in the lib folder is the entry point for your app.
* Open the terminal in VSCode (Ctrl+`` or View > Terminal) and run the command flutter run`. This will build and launch your app on an emulator or connected device.

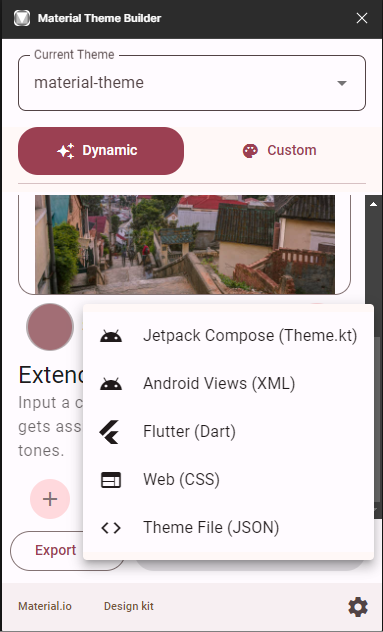
**Task 6- Visualizing dynamic color in your app – Codelab**

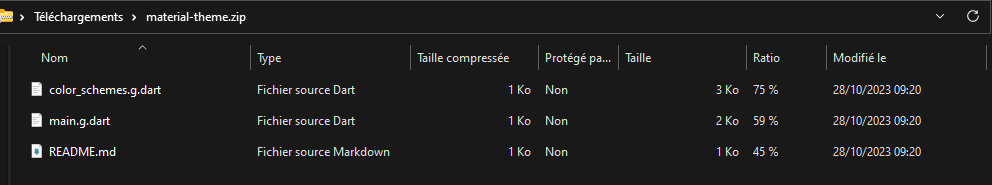












During the codlab session, we explored the new dynamic color features introduced with Material You. The focus was on creating dynamic color palettes and understanding the underlying concepts that contribute to accessible color schemes. Additionally, we learned how to apply user-generated colors to our app and utilized the latest design tools to visualize our app with dynamic colors.

In this session, we covered the following topics:

* Understanding the updates in Material Design color
* Applying user-generated colors to our app
* Utilizing tools such as Figma and the Material Theme Builder plugin
* Building upon foundational design concepts, including color palettes and Android color schemes

To participate in the codlab, we needed the following:

* A Figma Account
* The Figma Dynamic color Designlab file

The Figma plugin Material Theme Builder

Through the examination of this step, we acquired a deeper understanding of how vibrant colors can be extracted from images and organized to form cohesive color schemes within the Material Theme Builder. Additionally, we found that the Material Theme Builder provides the capability to instantly download the generated color scheme in the form of a Dart file. This functionality allows us to effortlessly acquire a pre-configured theme for our apps. By incorporating the downloaded color scheme as a Dart file, integration into our Flutter projects becomes seamless, thereby saving time and effort that would otherwise be spent on manual theme color configuration. This feature significantly enhances the efficiency and convenience of implementing consistent and visually appealing themes across our applications.

**Task-7: Create your first App using MIT AppInventor**

This documentation guides us through the steps of developing our inaugural app using MIT App Inventor, a visual development platform designed for crafting mobile applications for Android devices. What sets it apart is its avoidance of traditional programming languages, making app creation accessible to a broader audience. Throughout this walkthrough, we'll delve into the different stages of app development, emphasizing the essential concepts and features integral to MIT App Inventor.

