**Flutter Project Setup Instruction and Troubleshooting on**

**macOS**

**For**

**Plant Diagnosis System Merged**

1. **System Requirements**

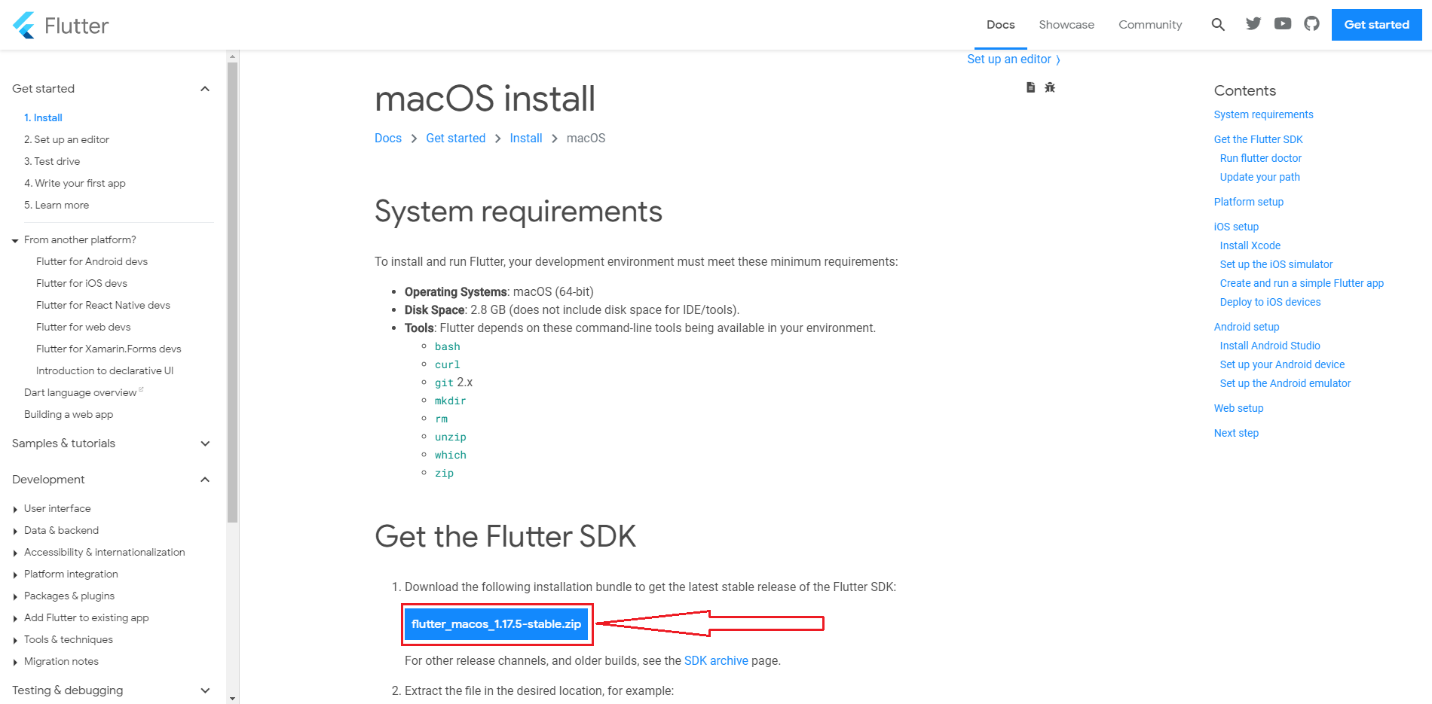
To install and run Flutter, your development environment must meet these minimum requirements:

* **Operating Systems**: macOS (64-bit)
* **Disk Space**: 2.8 GB (does not include disk space for IDE/tools).
* **Tools**: Flutter depends on these command-line tools being available in your environment.
  + bash
  + curl
  + git 2.x
  + mkdir
  + rm
  + unzip
  + which
  + zip

**Download Flutter SDK**

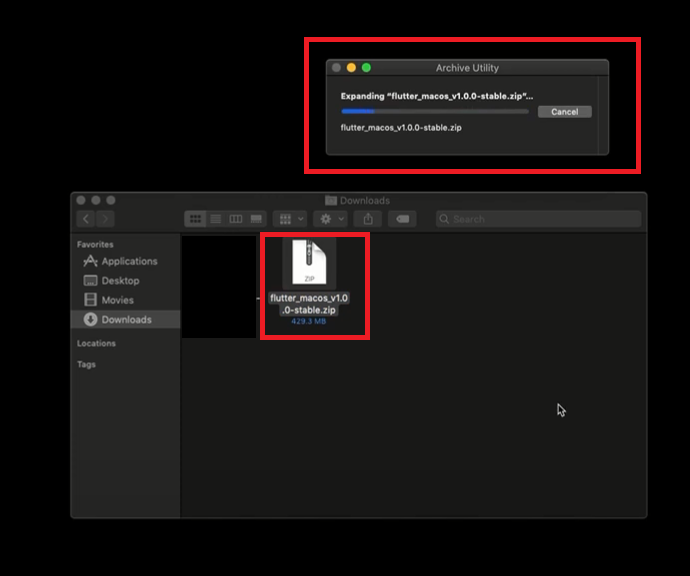
Download the following installation bundle to get the latest stable release of the Flutter SDK:

<https://flutter.dev/docs/get-started/install/macos>



Extract the zip file

After download the SDK you will find it inside download directory, just double click on it

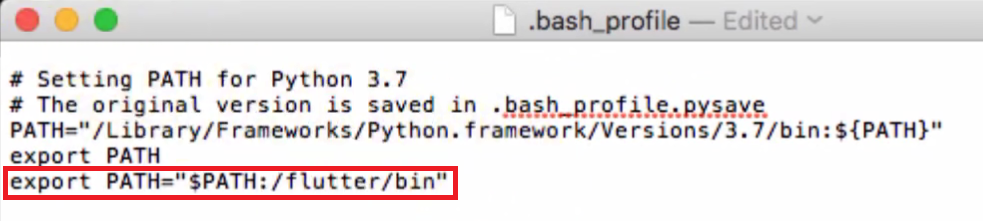


# **Update Path of Environment Variable**

1. Open (or create) the rc file for your shell. Typing “echo $SHELL” in your Terminal tells you which shell you’re using. If you’re using “Bash”, edit “$HOME/.bash\_profile” or “$HOME/.bashrc”. If you’re using “Z shell”, edit “$HOME/.zshrc”. If you’re using a different shell, the file path and filename will be different on your machine.
2. Open terminal and type “open –e .bash\_profile” and press enter

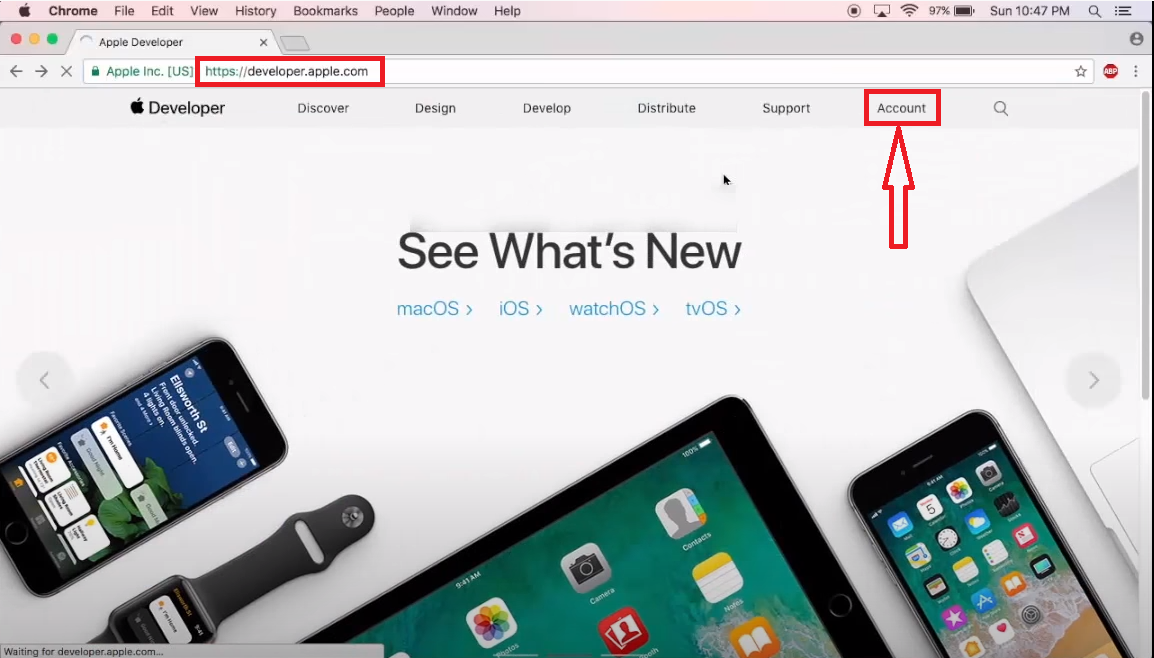


1. Now type “export PATH=$PATH:[PATH\_TO\_FLUTTER\_SDK]/flutter/bin” and press enter

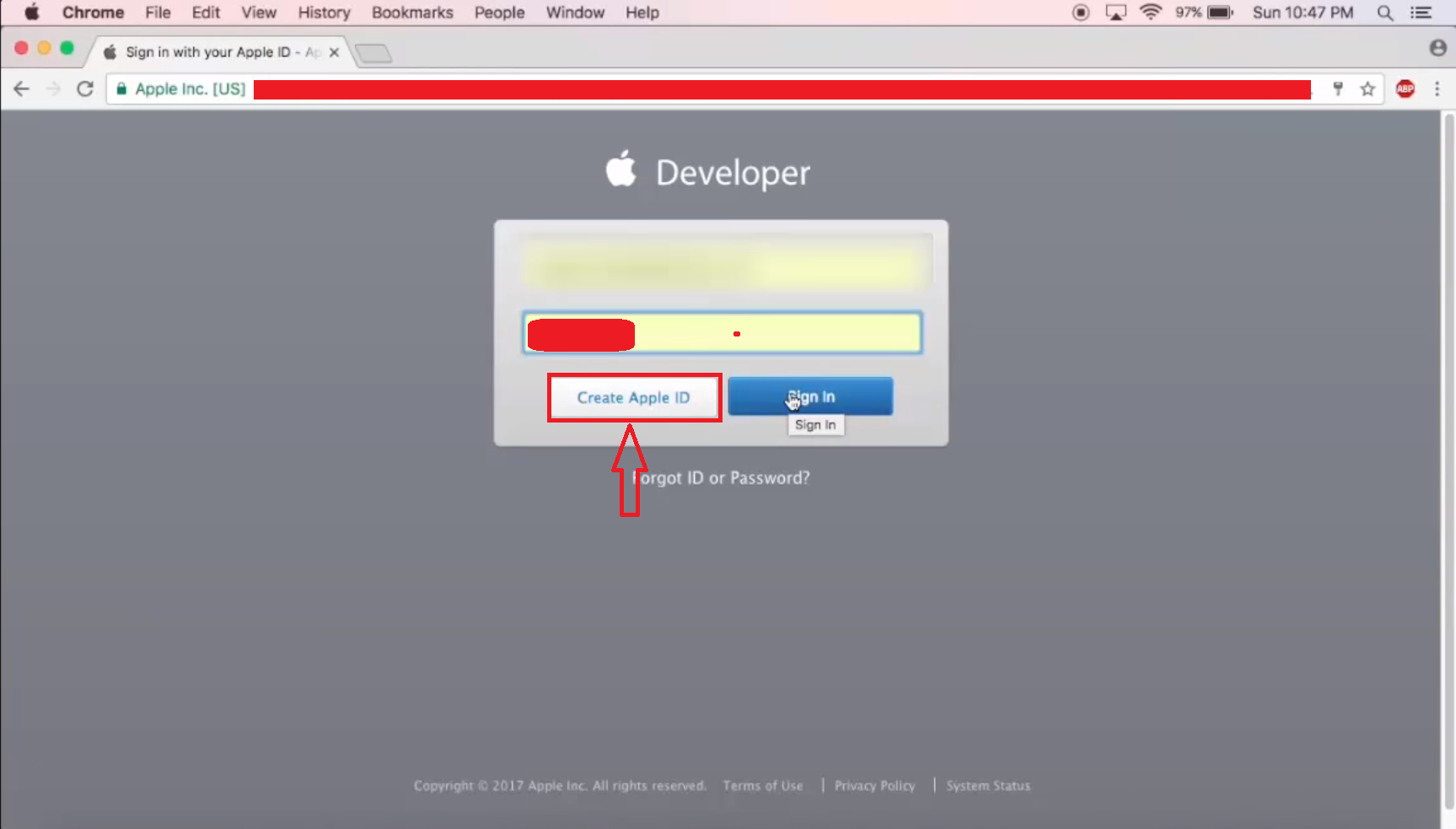


# **Install Xcode (Xcode download & setup from apple appstore)**

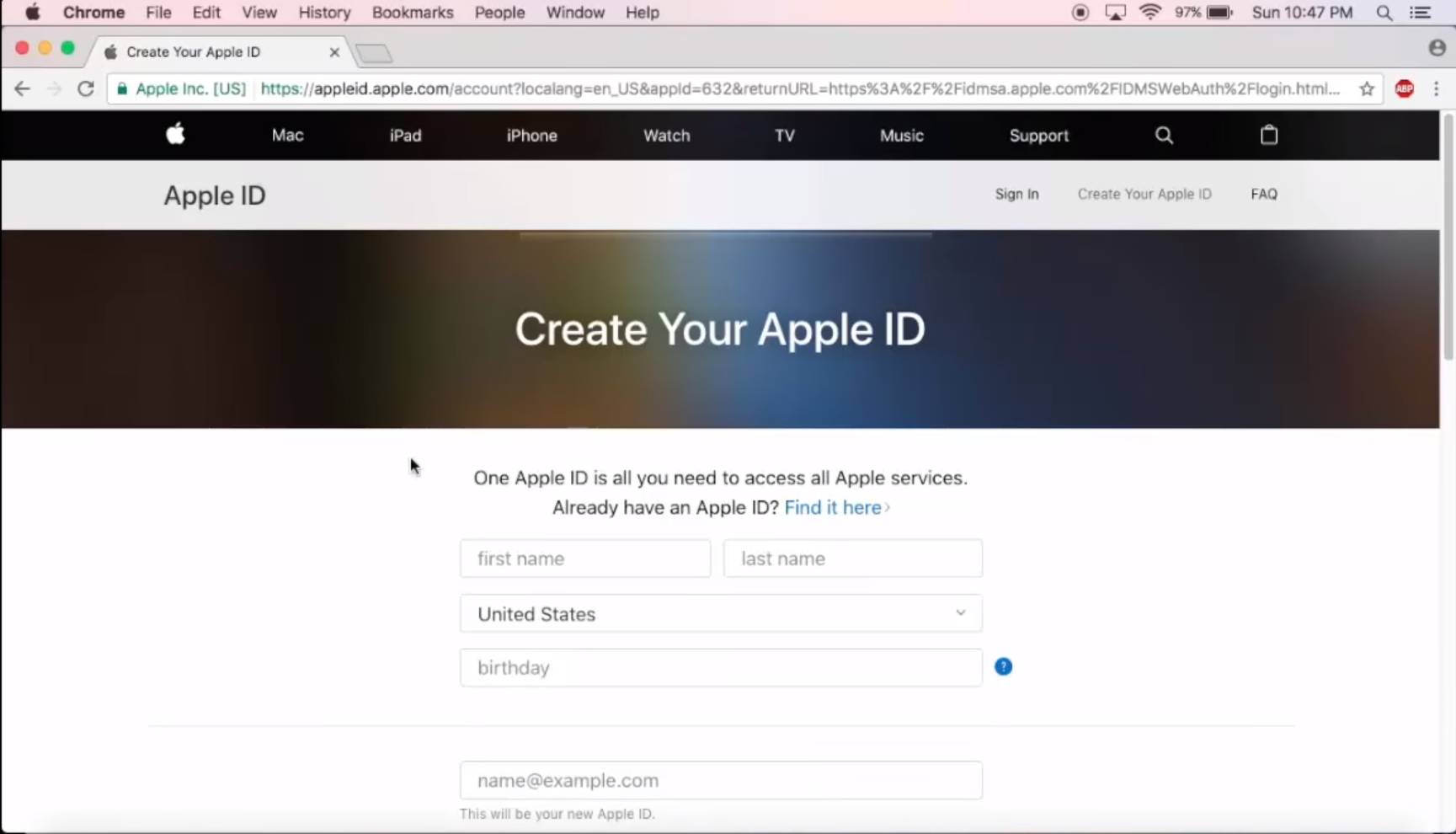
1. If you already have an Apple ID then skip this point and start from point two(ii), and if you don’t have any Apple ID then go to “https://developer.apple.com” this link and click Account



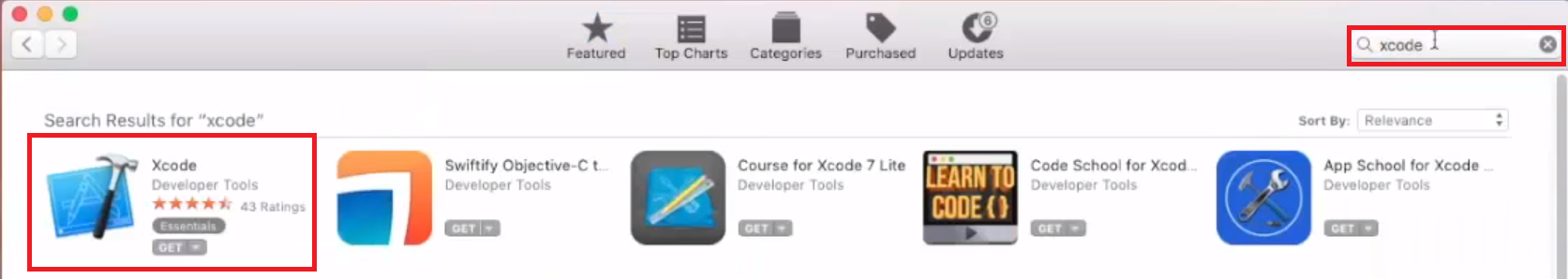
and then click “Create Apple ID”



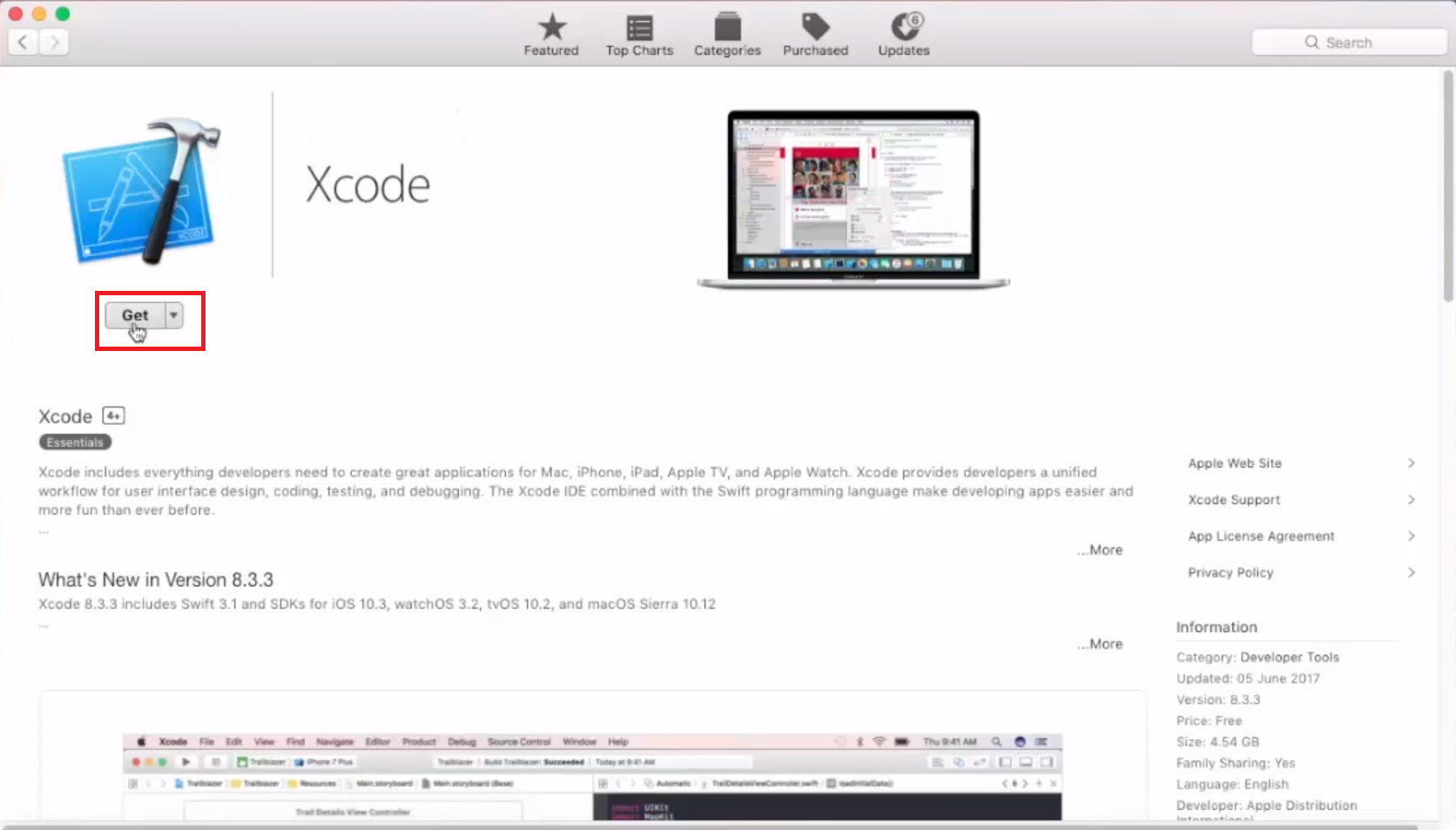
Now fill up the form and submit



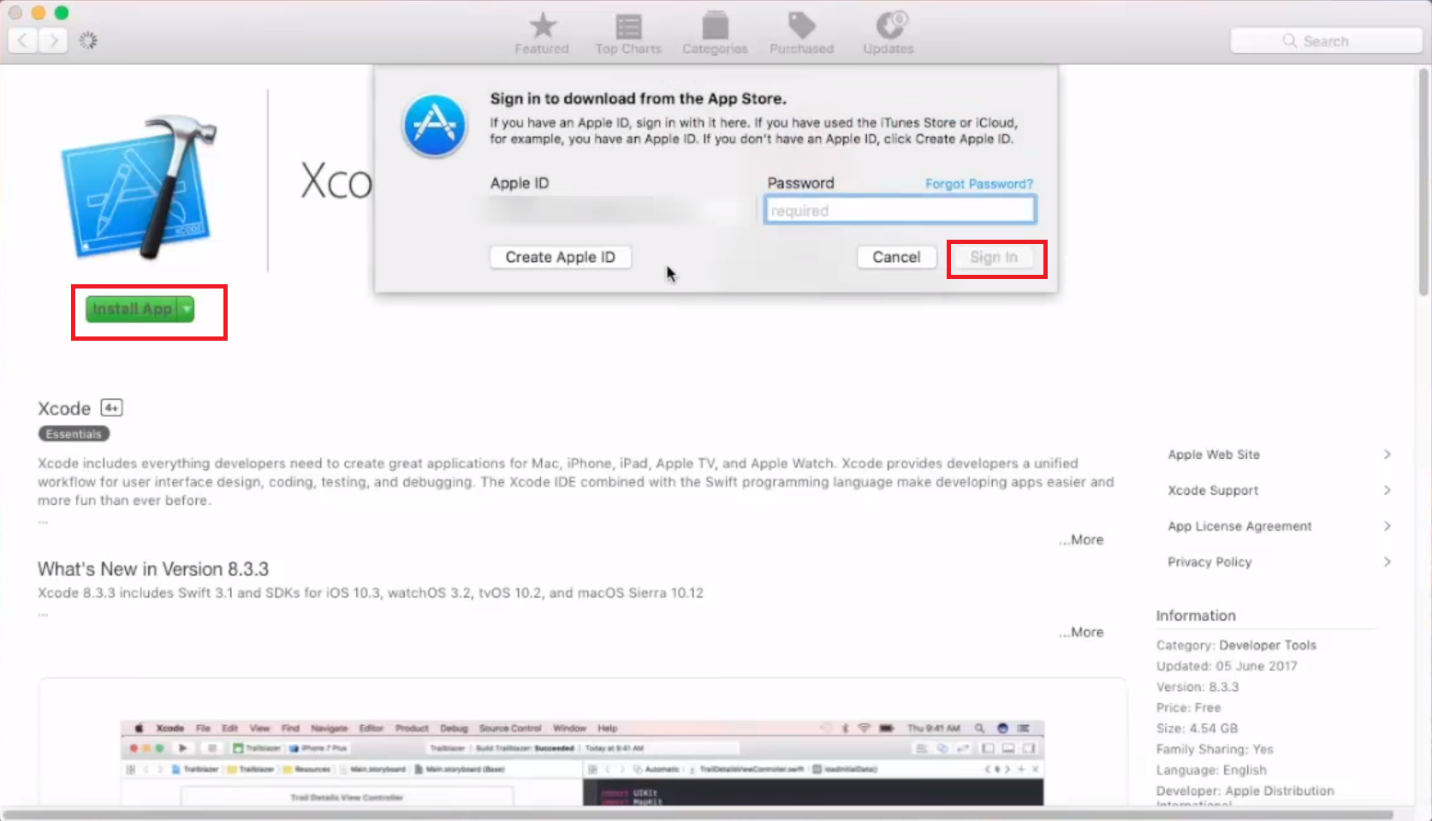
1. Now open Apple App Store and search for “xcode”



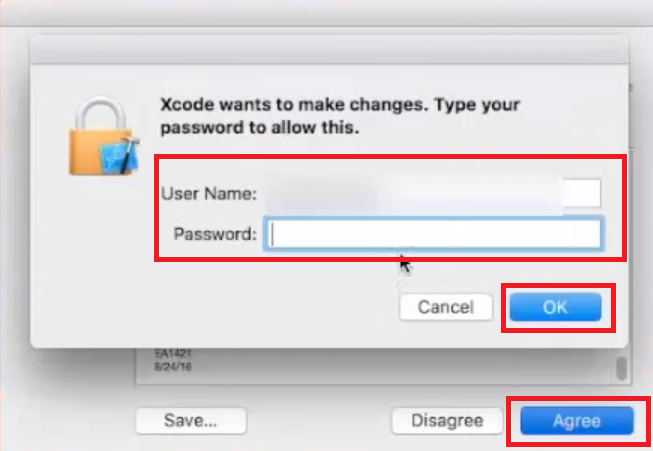
1. Click on Xcode and click on “Get”



1. Click on “Install App” and Authenticate with your Apple ID and click on Sign In



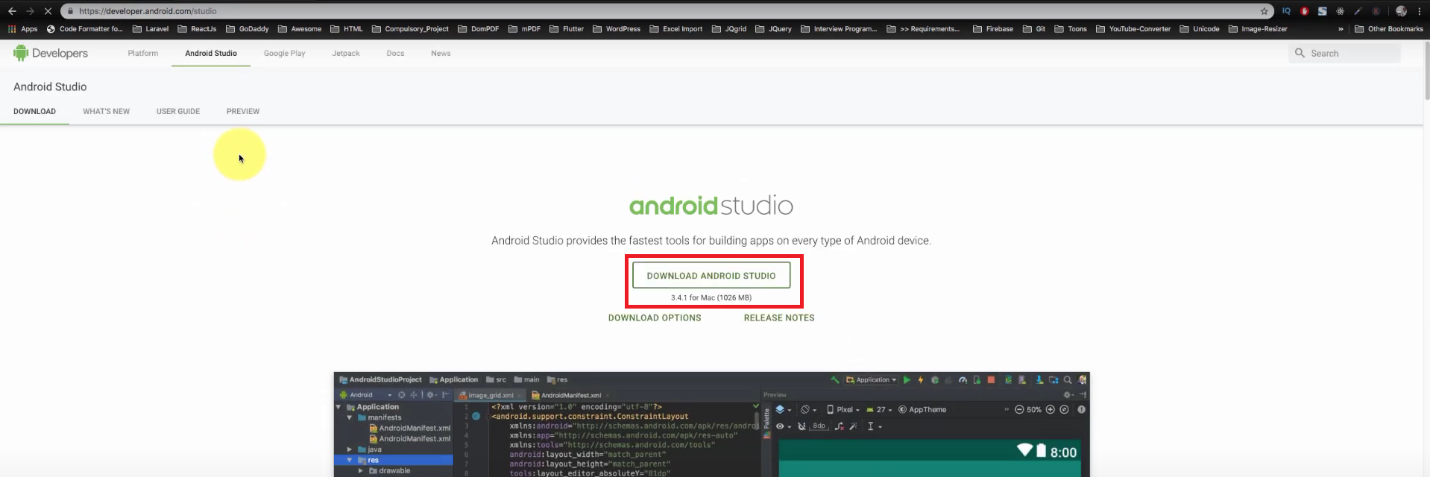
1. After completing download find xcode from your machine LaunchPad and double click on xcode to run
2. Now click on Agree and authenticate with your “mac username and password” and click on “Ok”



1. And Xcode will be successfully installed in your machine

# **Install Android Studio**

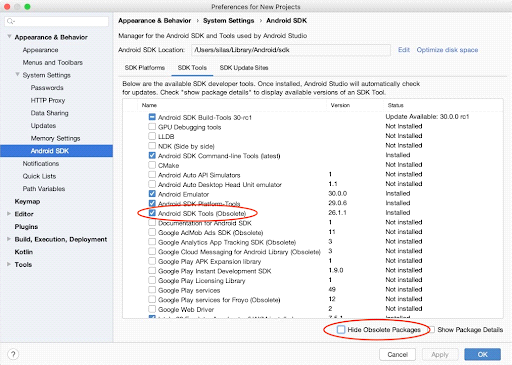
1. Download and install from: <https://developer.android.com/studio>

****

**Warning:** In Android Studio 3.6 or later, it needs to manually add the old version of the Android SDK Tools for Flutter to work. To do this:

* Open the **Android Studio SDK Manager**
* In the Android SDK tab, uncheck **Hide Obsolete Packages**
* Check **Android SDK Tools (Obsolete)**

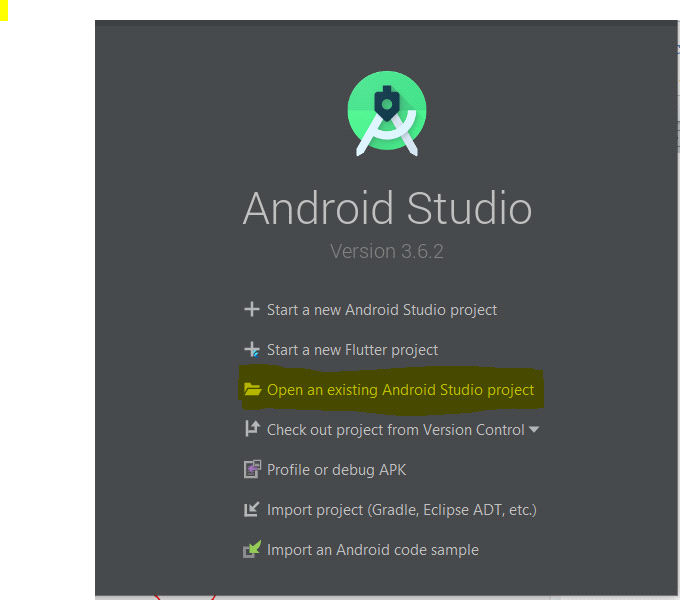
The image below shows the appropriate settings:



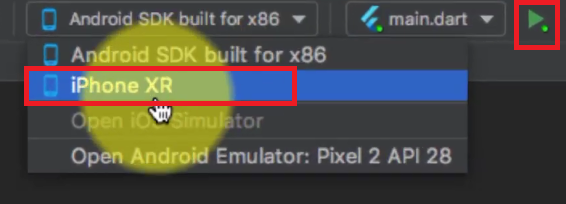
This is a [**known issue**](https://github.com/flutter/flutter/issues/51712) that will be addressed in an upcoming version of Flutter.

# **Running Project in the Simulator:**

1. Check out the project from Git.
2. Open the project in android studio marked menu.



1. Select the project folder which is downloaded from Git Repository.
2. Download all dependencies by clicking **Get dependencies**
3. Search for Simulator in the Spotlight Search by type “Simulator” and press enter, then there will be open simulator
4. Select the simulator and run the project by clicking the marked run button

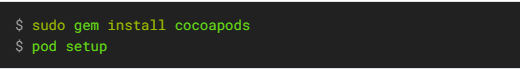


After that PDS-Merged App will be opened in the simulator.

1. **Deploy to iOS devices**

To deploy your Flutter app to a physical iOS device you need the third-party CocoaPods dependency manager and an Apple Developer account. You’ll also need to set up physical device deployment in Xcode.

1. Install and set up CocoaPods by running the following commands:

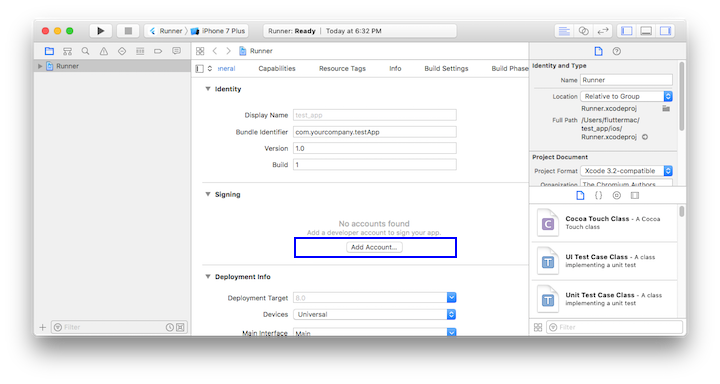


1. Follow the Xcode signing flow to provision your project:
2. Open the default Xcode workspace in your project by running open ios/Runner.xcworkspace in a terminal window from your Flutter project directory.
3. Select the device you intend to deploy to in the device drop-down menu next to the run button.
4. Select the Runner project in the left navigation panel.
5. In the Runner target settings page, make sure your Development Team is selected. The UI varies depending on your version of Xcode.

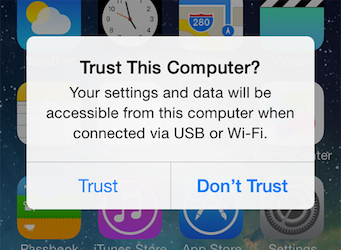
* For Xcode 10, look under **General > Signing > Team**.
* For Xcode 11 and newer, look under **Signing & Capabilities > Team**.

When you select a team, Xcode creates and downloads a Development Certificate, registers your device with your account, and creates and downloads a provisioning profile (if needed).

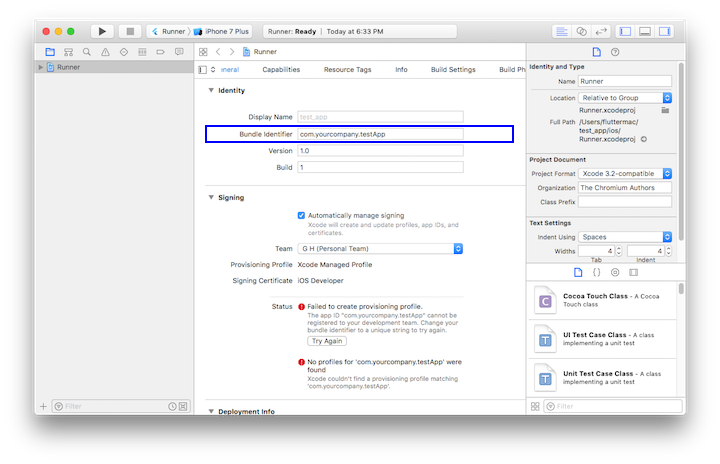
* To start your first iOS development project, you might need to sign into Xcode with your Apple ID.



* The first time you use an attached physical device for iOS development, you need to trust both your Mac and the Development Certificate on that device. Select Trust in the dialog prompt when first connecting the iOS device to your Mac.



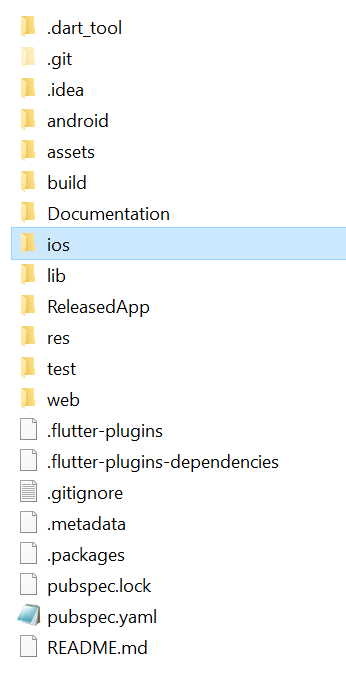
* Then, go to the Settings app on the iOS device, select **General > Device Management** and trust your Certificate.
* If automatic signing fails in Xcode, verify that the project’s **General > Identity > Bundle Identifier** value is unique.



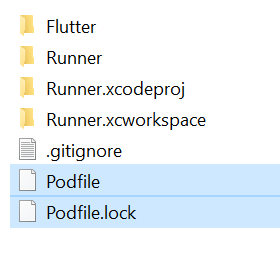
1. Start your app by running flutter run.

**Deploy to iOS devices(Troubleshooting)**

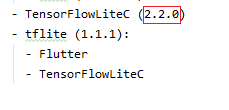
1. First of all, open global terminal and
2. Uninstall the cocoapods by the command ‘$sudo gem install cocoapods’.
3. Then reinstall the cocoapods by the command ‘$sudo gem install cocoapods’.
4. Go to the root directory of the project.
5. Find and go to the ‘ios’ directory.



1. Remove the ‘Podfile’ and ‘Podfile.lock’.



1. Open terminal in the project root directory and Execute those commands synchronously
2. $flutter clean
3. $flutter pub get
4. $flutter upgrade
5. Open terminal in the ‘ios’ directory and execute the command ‘$pod install’.
6. From the ‘ios’ directory open the ‘Podfile.lock’ in editor.
7. Edit the TensorFlowLiteC (xyz.xyz.xyz), whatever the version just changes it to (2.2.0).

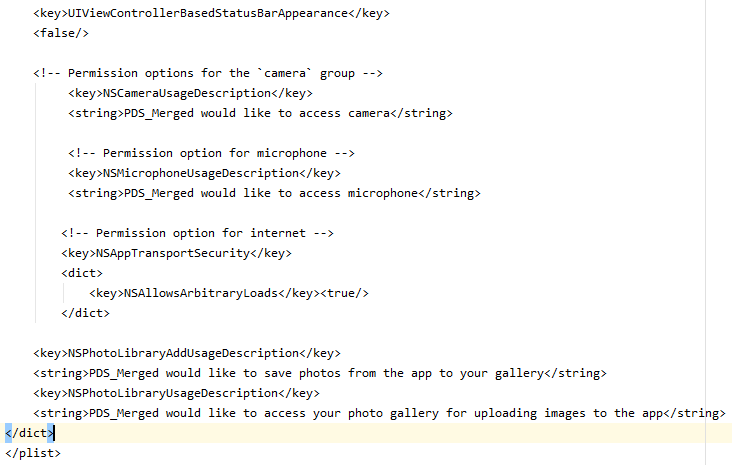


1. If got “**'vector' file not found**" issue then follow those steps bellow:
2. Open ios/Runner.xcworkspace in Xcode, click Runner > Tagets > Runner > Build Settings, search Compile Sources As, change the value to Objective-C++
3. Go to iOS>flutter>Release.xcconfig, add those and save

“#include "Pods/Target Support Files/Pods-Runner/Pods-Runner.release.xcconfig"

#include "Pods/Target Support Files/Pods-Runner/Pods-Runner.profile.xcconfig"”

1. Go to Runner directory from the ios directory and open Info.plist file in editor.
2. Inside the ‘<dict></dict>’ tag, in the bottom (after the <false/>) just before the closing tag(</dict>) add those lines by copy and paste.



The lines are.

<!-- Permission options for the `camera` group -->

<key>NSCameraUsageDescription</key>

<string>PDS\_Merged would like to access camera</string>

<!-- Permission option for microphone -->

<key>NSMicrophoneUsageDescription</key>

<string>PDS\_Merged would like to access microphone</string>

<!-- Permission option for internet -->

<key>NSAppTransportSecurity</key>

<dict>

<key>NSAllowsArbitraryLoads</key><true/>

</dict>

<key>NSPhotoLibraryAddUsageDescription</key>

<string>PDS\_Merged would like to save photos from the app to your gallery</string>

<key>NSPhotoLibraryUsageDescription</key>

<string>PDS\_Merged would like to access your photo gallery for uploading images to the app</string>

1. Start Simulator and run the Project.