



Flutter dokumentacija i upustvo

Sadržaj:

1.O Flutter-u	2
2.Instalacija Flutter-a na Windows operativnom sistemu	2
3.Instalacija Flutter SDK-a	4
4.Nadogradnja PATH varijable	5
5.Instalacija Android Studio-a	7
5.1.Instalacija na Android telefonu	7
5.2.Postavljanje Android emulatora	8
6.Web Setup	8
7.Instalacije Visual Studio Code-a	9
8.Prednosti Flutter-a i razlike sa React Native-om	10

1.O Flutter-u

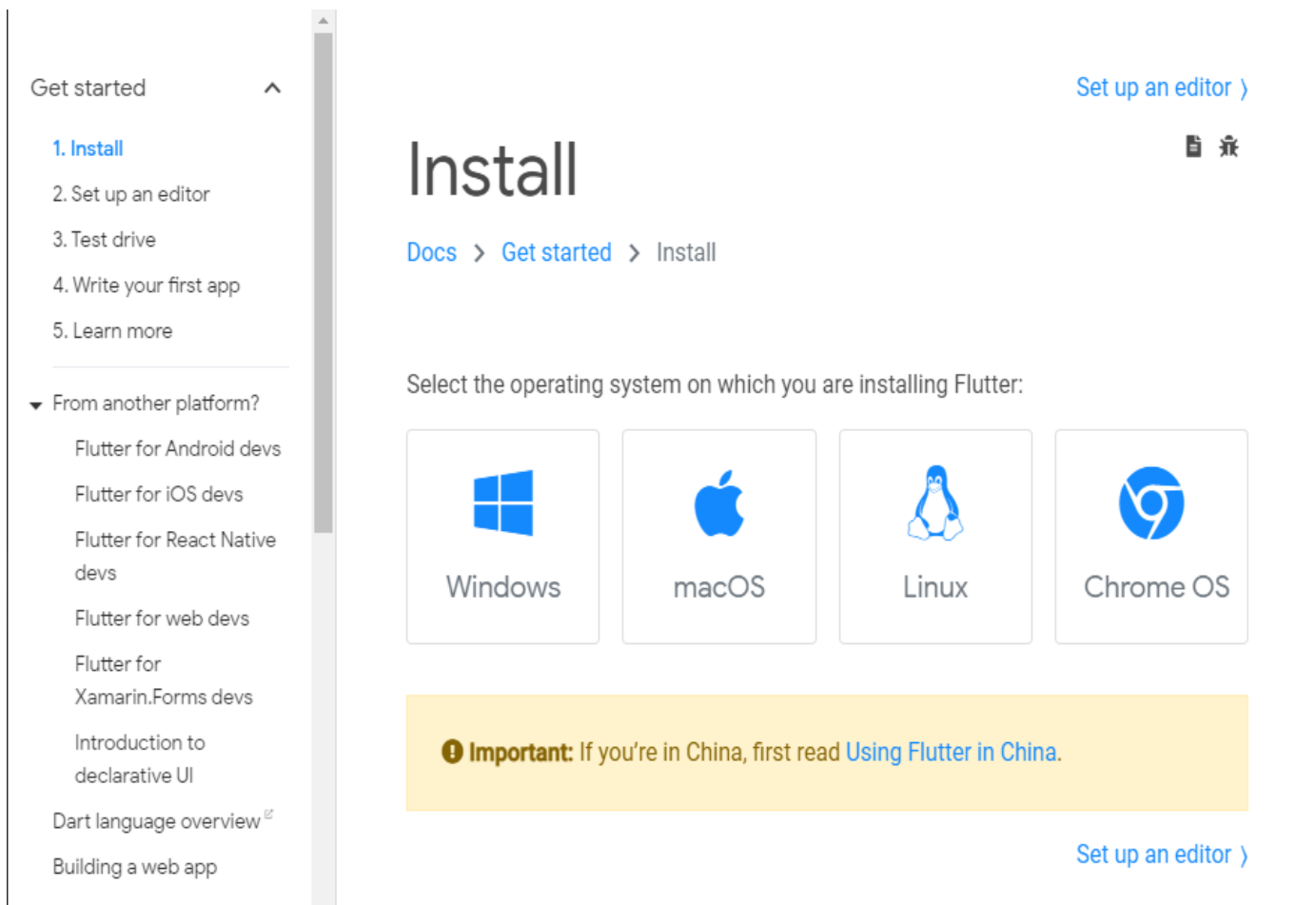
Flutter je "open-source" UI toolkit napravljen od strane kompanije Google za kreiranje mobilnih, web i desktop aplikacija iz jedinstvene kodne baze. To omogućava developerima da pomocu jednog programskog jezika i jedne kodne baze razvijaju aplikacije za razlicite operative sisteme(Android, iOS, Mac, Windows, Chrome OS, web...).

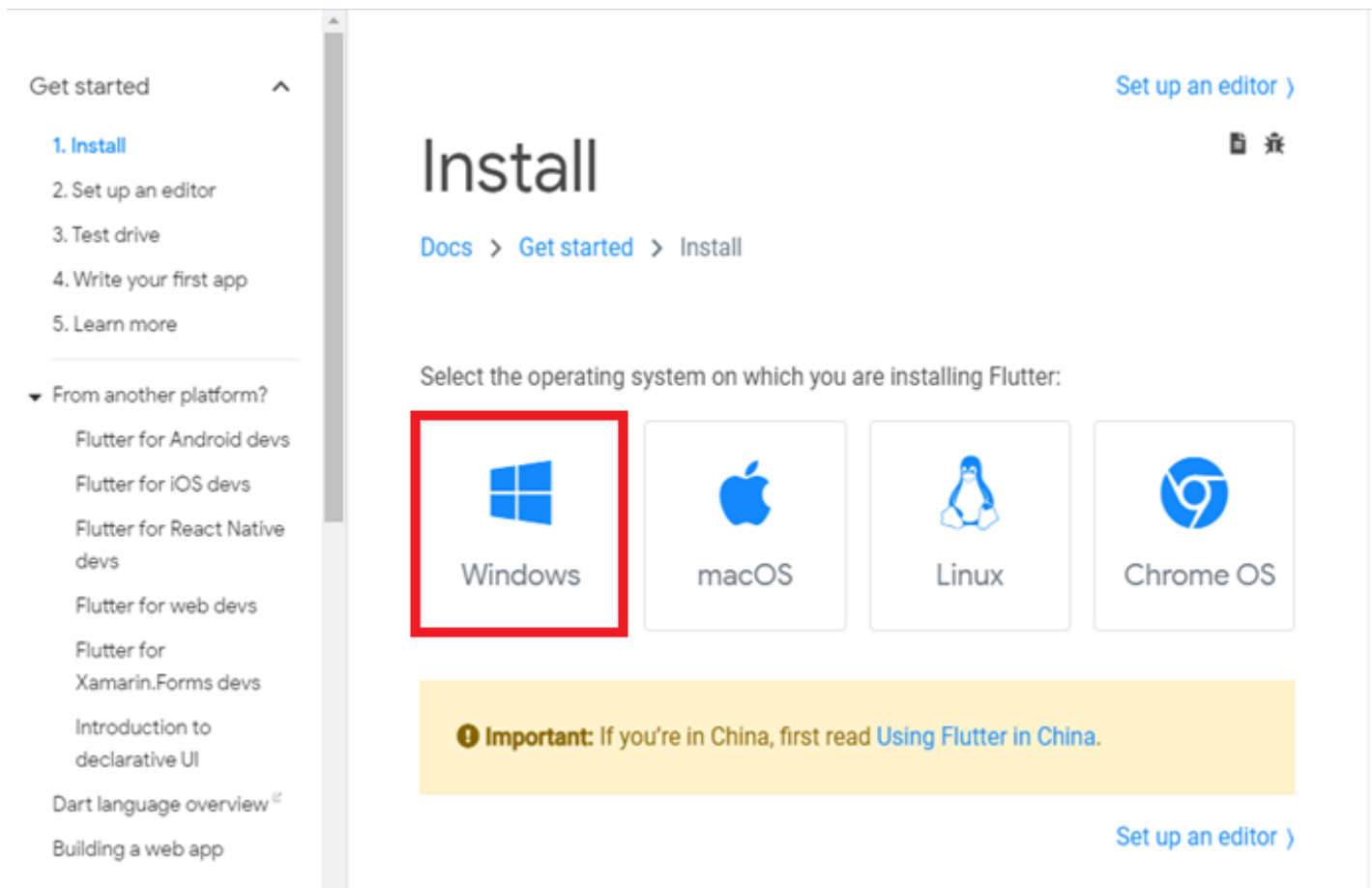
-Flutter koristi Dart programski jezik ciji kompajler omogućava "gradnju" aplikacije direktno u masinski kod cime se eliminisu bilo kakvi "bagovi" prilikom procesa interpretacije koda.

2.Instalacija Flutter-a na Windows operativnom sistemu

-Potrebno je posetiti stranicu na kojoj mozete naci odgovarajucu verziju vase *Flutter* instalacije:

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





-Potrebno je izabrati odgovarajući operativni sistem, kao što se vidi sa slike u prilogu *Flutter* je dostupan za *Windows*, *macOS*, *Linux* i *Chrome OS* operativne sisteme.

-U našem slučaju to će biti **Windows** operativni sistem.

Kao što se vidi postoji izuzetak ako ste iz Kine, o tome detaljnije možete pročitati klikom na *Using Flutter in China*.

-Da biste instalirali *Flutter* potrebno je da imate:

- Operativni sistem Windows 7 SP1 ili noviji (64-bit)

- Memorije na disku : 1.32GB (ne uključujući prostor potreban za *IDE*)

- Alatkje od kojih *Flutter* zavisi:

 - Windows PowerShell 5.0 ili novije (dolazi instalirano uz Windows 10)

 - Git za Windows 2.x, sa opcijom Use Git from the Windows Command Prompt koju treba izabrati tokom instalacije istog.

3.Instalacija Flutter SDK-a

Get the Flutter SDK

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

`flutter_windows_2.0.1-stable.zip`

For other release channels, and older builds, see the [SDK releases](#) page.

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

⚠ Warning: Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

-Klikom na polje `flutter_windows_2.0.1-stable.zip` mozete skinuti najnoviju stabilnu verziju *Flutter SDK*.

-Potrebno je raspakovati zip fajl na lokaciji kao sto je `C:\src\flutter`, preporuceno je izbegavati instaliranje na lokacijama poput "`C:\Program Files\`" jer zahtevaju dodatne administratorske privilegije koje vam mogu napraviti probleme pri radu.

-Ako kojim sucajem ne zelite da instalirate fiksne verzije Flutter SDK, mozete preskociti gore navedene korake tako sto cete preko Windows Command Prompt-a(cmd.exe) pozvati komandu bez navodnika:

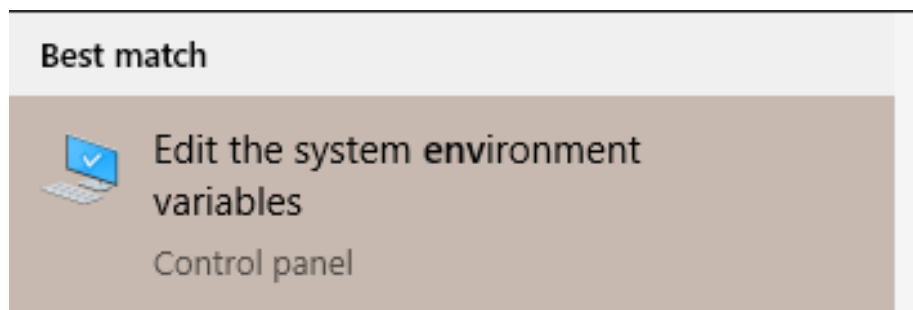
- `"C:>git clone https://github.com/flutter/flutter.git -b stable"`

-Ovom komandom cete uvek skinuti najnoviju verziju Flutter-a koja ce se klonirati u folder "`C:\src\flutter`".

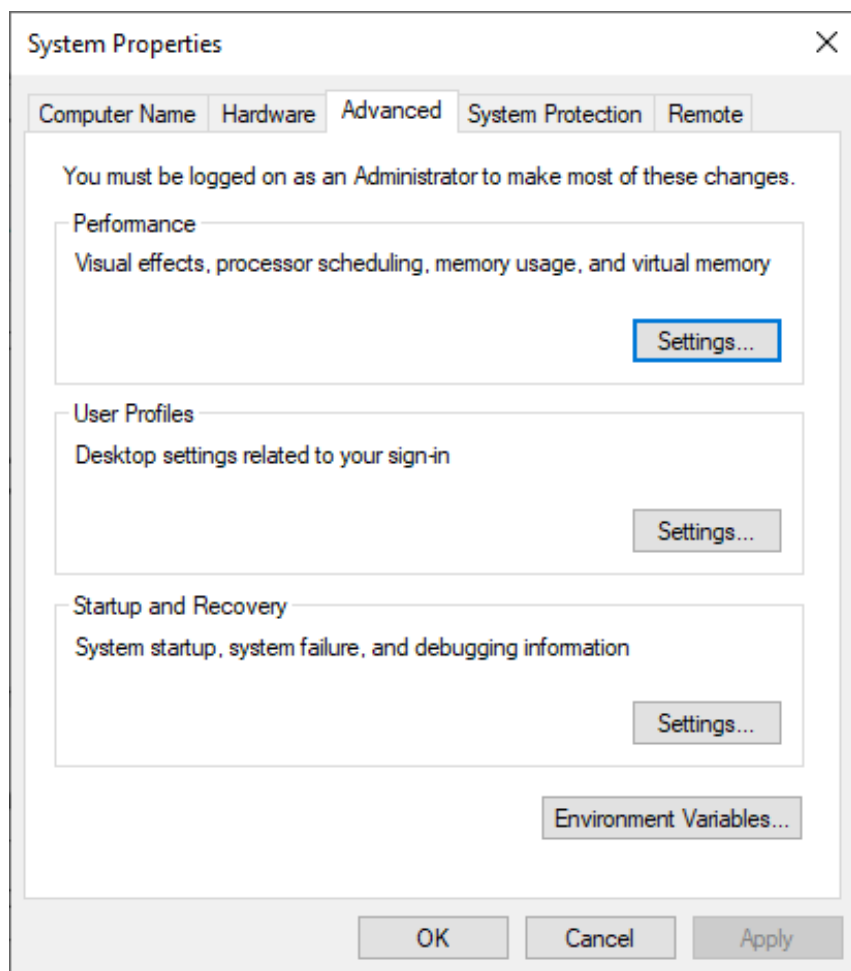
4.Nadogradnja PATH variable

-Ukoliko zelite da *Flutter* komande rade direktno u Windows Command Prompt-u(cmd.exe) potrebno je da dodate *Flutter* u PATH varijablu vaseg sistem.

-Prvo unutar Windows Start pretrazivaca pronadjite "env" i izaberite "Edit environment variables for your account".

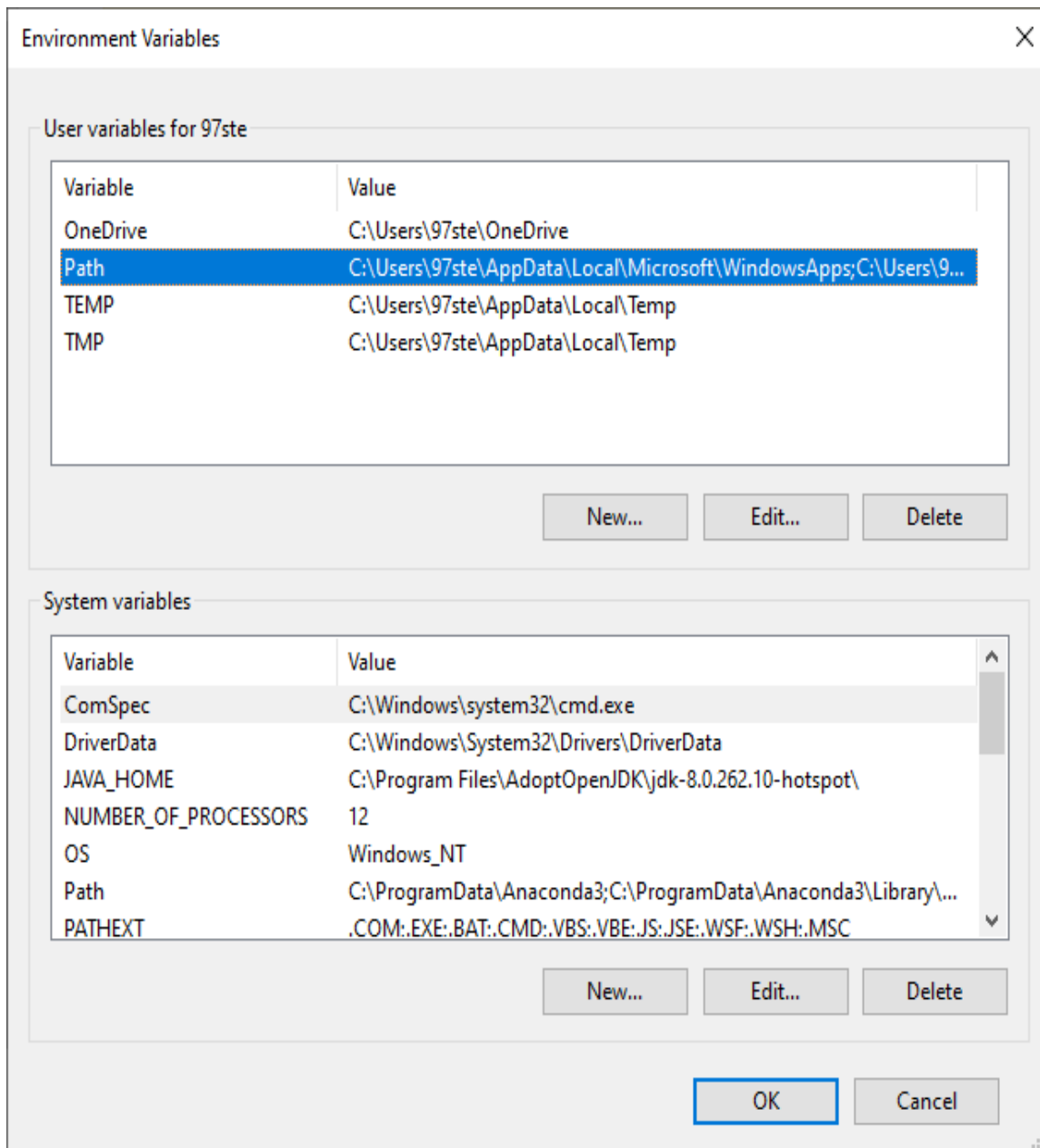


-Nakon toga izaberite "Environment Variables..." na dnu Advanced tab-a.



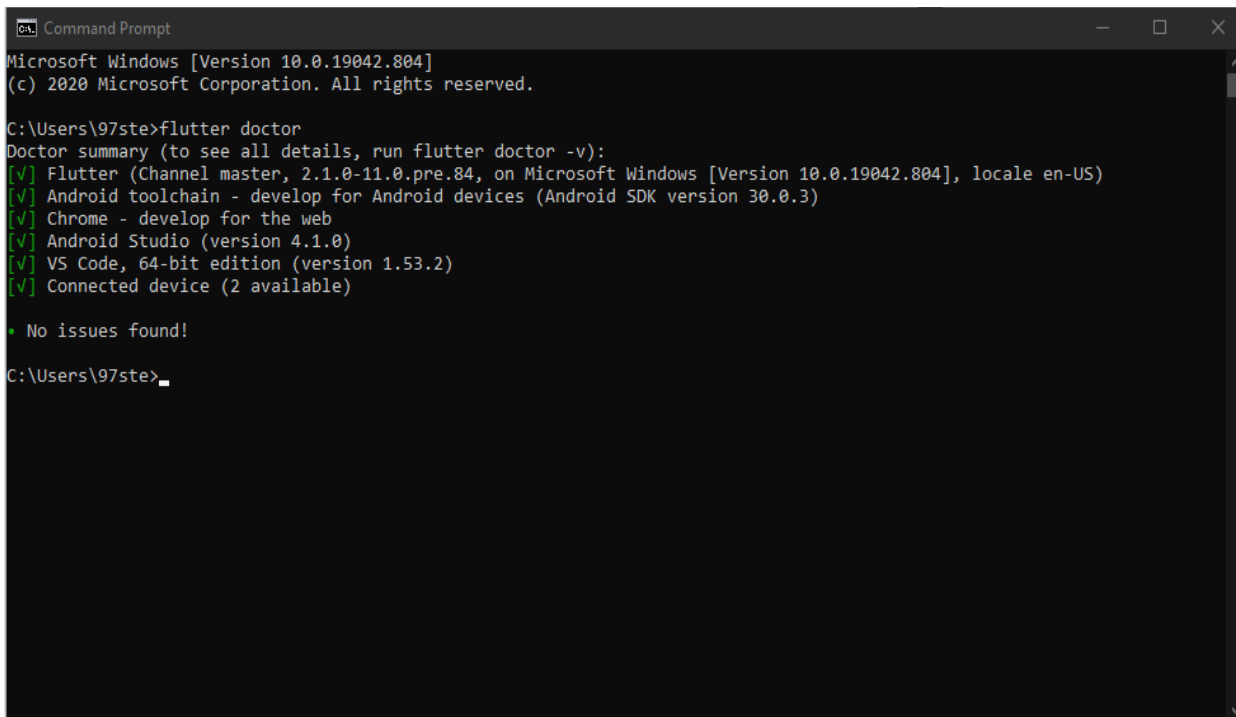
-Konacno kada pristupite varijablama ukoliko PATH postoji na kraj putanje klikom na dugme Edit dodati *flutter* koristeći ; kao separator izmedju vec postojećih vrednosti.

-Ukoliko PATH ne postoji, kreirati novu varijablu po nazivom Path i dodati punu putanju do *flutter*-a. Npr. C:\flutter\bin



-Konacno potrebno je zatvoriti i ponovo otvoriti sve Windows konzole da bi ove promene imale efekta.

-Pokrenuti komandu `flutter doctor` unutar Command Prompt--a kojom mozete proveriti da li je *Flutter* uspesno postavljen na vasem racunaru.



```
Command Prompt
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\97ste>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel master, 2.1.0-11.0.pre.84, on Microsoft Windows [Version 10.0.19042.804], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] VS Code, 64-bit edition (version 1.53.2)
[✓] Connected device (2 available)

• No issues found!

C:\Users\97ste>
```

-Tokom ovog koraka mozete dobiti poruku da *Flutter plugin* i *Dart plugin* za *Android Studio* nisu instalirani ,te korake cemo pokriti kasnije tokom instalacije odgovarajucih editor-a.

5.Instalacija Android Studio-a

1.Skinuti I instalirati Android studio klikom na “DOWNLOAD ANDROID STUDIO”.

<https://developer.android.com/studio>

2.Startovati Android Studio I proci kroz “Android Studio Setup Wizard”. Ovo ce instalirati Najnoviji *Android SDK-a*,*Android SDK Command-line* alatke i *Android SDK Build* alatke, koje su potrebne za *Flutter*.

5.1 Instalacija na Android telefonu

-Da bi pokrenuli i testirali Flutter aplikaciju na Android uredjaju, potreban vam je Android 4.1 sistem ili noviji.

1.Ukljuciti *Developer opcije* i *USB debugging* na vasem uredjaju. Detaljnije na:

<https://developer.android.com/studio/debug/dev-options>

2.Jedino na Windows-u potrebno je instalirati Google USB Driver:
<https://developer.android.com/studio/run/win-usb>

3.Pomocu USB kabla povezati telefon sa racunarom i dozvoliti potrebne autorizacije

4.Unutar *Windows Command Prompt*-a pokrenuti komandu `flutter devices` kako biste proverili da li je *Flutter* prepoznao vas uredjaj.

5.2 Postavljanje Android emulatora

-Da biste pokrenuli *Flutter* aplikaciju preko Android Emulator-a pratite sledece korake:

1. Omoguciti *VM acceleration* na vashoj masini:
<https://docs.microsoft.com/en-us/xamarin/android/get-started/installation/android-emulator/hardware-acceleration?pivots=windows>
2. Pokrenuti Android Studio, kliknuti na AVD Manager i izabrati Create Virtual Device...
 - Na starijim verzijama Android Studio-a, potrebno je pokrenuti Android Studio>Tools>Android>AVD Manager i izabrati Create Virtual Device...
 - Ako nemate otvoren projekat, mozete izabrati Configure>AVD manager i zatim izabrati Create Virtual Device...
3. Izaberite definiciju vashog uredjaja i kliknite Next.
4. Izaberite jednu ili vise sistemskih slika za Android verzije koje zelite da emulirate, nakon toga izaberite Next. (*Preporuka je x86 ili x86_64 slika*)
5. Ispod Emulated Performance, izaberite Hardware-GLES 2.0 i ukljucite hardware acceleration.
6. Potvrdite da li je AVD konfiguracija postavljena kako treba i izaberite Finish.
7. Unutar Android Virtual Device Manager-a, izaberite Run u toolbar-u. Emulator ce se pokrenuti i pokazati default platno za uredjaj i verziju operativnog sistema kojeg ste izabrali.

6.Web setup

-Flutter poseduje podrsku za pravljenje web aplikacija unutar stable kanala. Radi testiranja i debugovanja vase aplikacije potrebno je instalirati Google Chrome pretrazivac.

-Svaka aplikacija kreirana unutar Flutter 2 ce automatski biti pravljen i za web verziju vase aplikacije.

7.Instalacija Visual Studio Code-a

-Visual Studio Code je besplatan editor napravljen od strane Microsoft-a za Windows,Linux i macOS sisteme.

-Posедује podršku za *debugovanje,sintaksno obeležavanje,inteligentno popunjavanje koda,snippet-e,code refactoring* i "embedovan" GIT.

-Link za skidanje Visual Studio Code-a: <https://code.visualstudio.com/download>

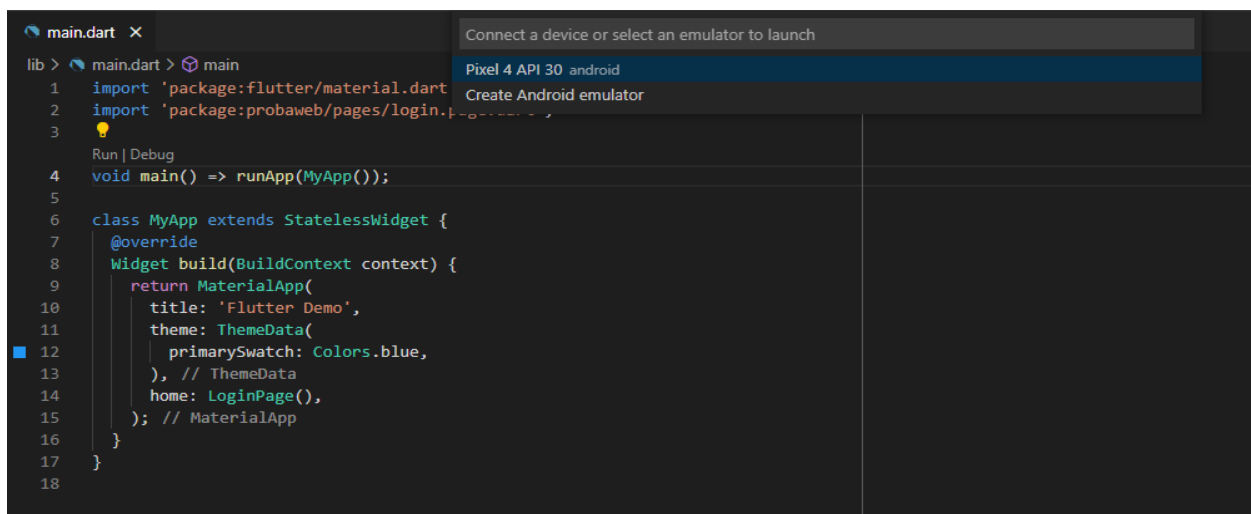
-Potrebno je instalirati Flutter i Dart plugin-ove kako bismo koristili Flutter unutar editora:

1. Pokrenite Visual Studio Code.
2. Pokrenite View>Command Palette (CTRL+SHIFT+P).
3. Ukucajte Install Extension i izaberite polje istog imena.
4. Ukucajte flutter unutar pretrage i zatim izaberite install.
5. Ponovite korak iznad samo umesto flutter ukucajte dart.

-Proverite da li je sve uspesno instalirano:

1. Pokrenite View>Command Palette (CTRL+SHIFT+P).
2. Ukucajte doctor i izaberite Flutter:Run Flutter Doctor.
3. Ovom komandom ce se unutar output panela sa donje strane editora biti prikazani izlazi provere kako biste se uverili da nije doslo do nikakvih problema.

-Takodje pomocu View>Command Palette (CTRL+SHIFT+P) mozete pokrenuti vas prethodno napravljen emulator unutar Android Studio-a tako sto cete ukucati u komandnu paletu komandu Flutter: Launch Emulator i izabrati uredjaj koji ste prethodno napravili.



```
lib > main.dart > main
1 import 'package:flutter/material.dart';
2 import 'package:probaweb/pages/login.dart';
3
4 void main() => runApp(MyApp());
5
6 class MyApp extends StatelessWidget {
7   @override
8   Widget build(BuildContext context) {
9     return MaterialApp(
10       title: 'Flutter Demo',
11       theme: ThemeData(
12         primarySwatch: Colors.blue,
13       ), // ThemeData
14       home: LoginPage(),
15     ); // MaterialApp
16   }
17 }
18
```

-U prilogu uz dokumentaciju doćicete projekat testne aplikacije koji mozete preuzeti preko gitlab-a unutar sandbox foldera pod mojim imenom.

8.Prednosti Flutter-a i razlike sa React Native-om

-React Native je najblizi konkurent Flutter-u, samim time uporedicemo pomenuta dva:

1. Flutter ce vecini ljudi biti tezi za učenje nego React Native. Glavni razlog jeste taj da Native koristi JavaScript koji je vec oformljen jezik za razliku od Dart-a cime ce proces učenja novog jezika biti dosta duzi.
2. Velika razlika izmedju Flutter-a i React-a lezi u komponentama ovih framework-ova.U React-u mali broj komponenti se adaptira platformi i pri koriscenju razlicitih sistema te komponente se moraju prilagoditi datom sistemu od strane programera. Flutter je po tom pitanju dosta napredniji jer mu je svaka komponenta(widget) adaptivna i moze se koristiti npr. i na Androidu i na iOS-u.
3. Kada su u pitanju performance Flutter prednjaci jer se kompajluje u ARM ili x86 native biblioteke,sto ga cini veoma brzim. React se ne kompajlira u native kod i poseduje JavaScript layer sto mu smanjuje performanse.
4. React Native je u sustini wrapper preko native metoda, sa tim mu je potreban "most" preko kojih ce pozive metoda proslediti API-u.Ovo postaje "bottleneck" kada se broj poziva poveca.Unutar Flutter-a ovi pozivi se izvrsavaju direktno.
5. Flutter se lakse debuguje. Ovo se dokazalo u dosadasnjem iskustvu pri radu developera sa oba framework-a.