Name:Dhruv Maurya

Div:D15B

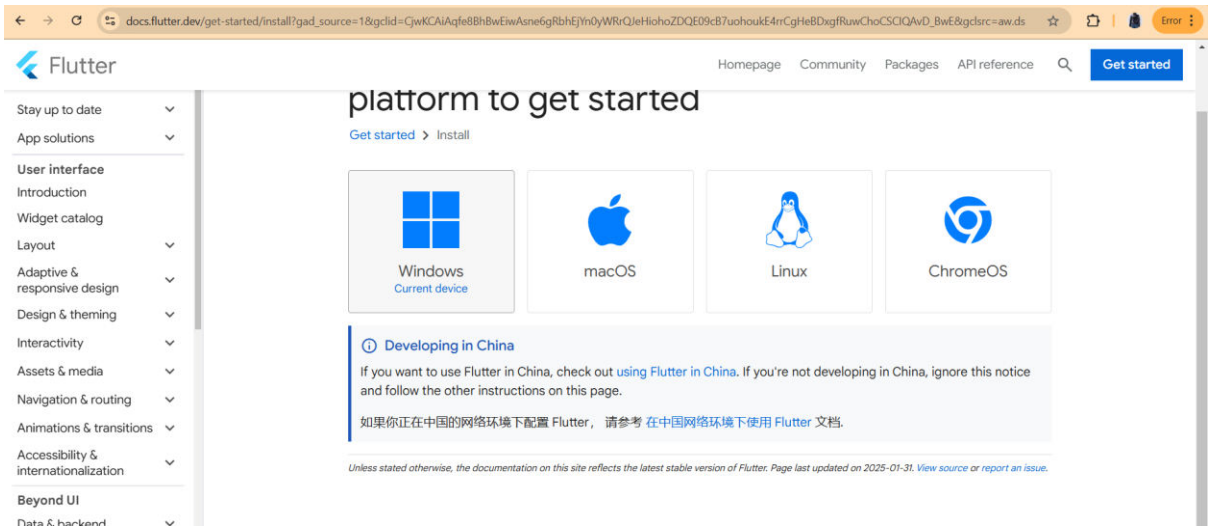
Roll No:31

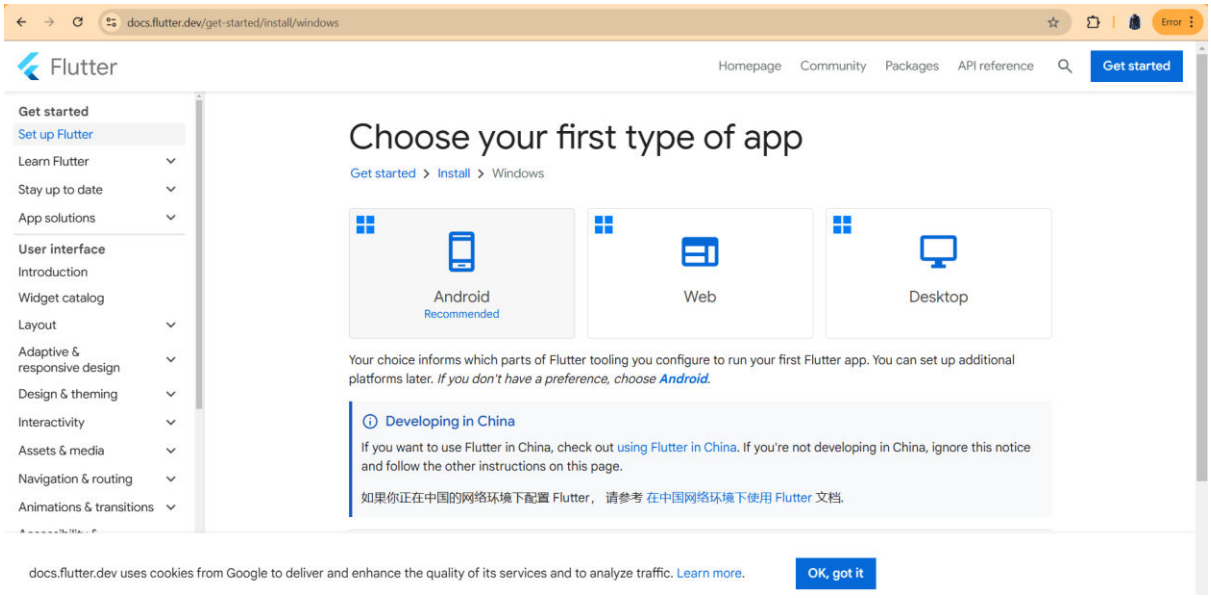
Exp No.1

Step 1: Download the installation bundle of the Flutter Software Development Kit for windows.

To download Flutter SDK, Go to its official website https://docs.flutter.dev/get-started/install ,

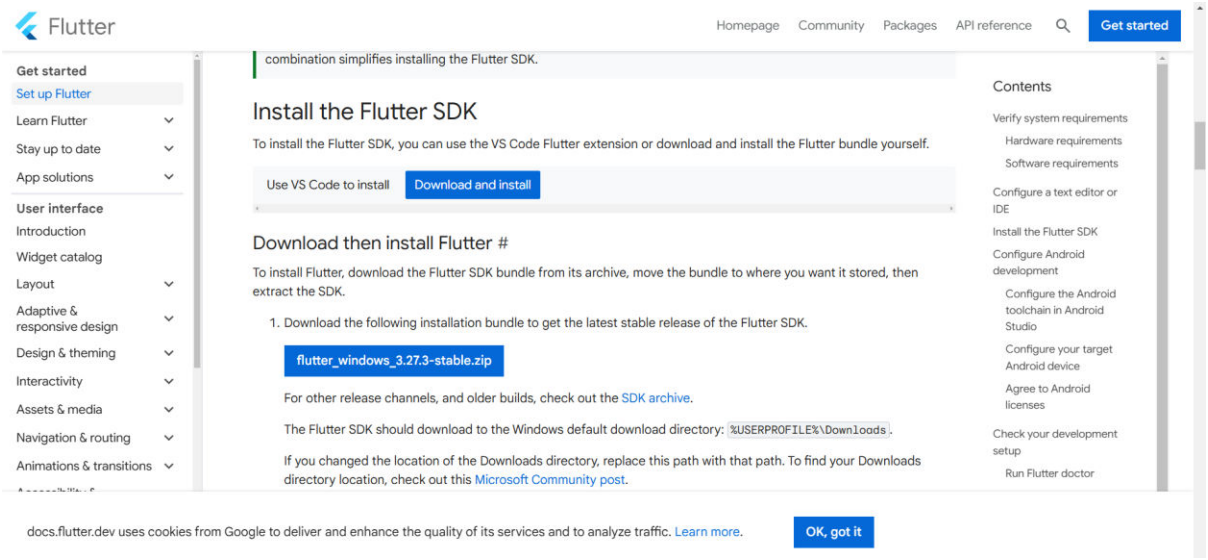
you will get the following screen.





Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

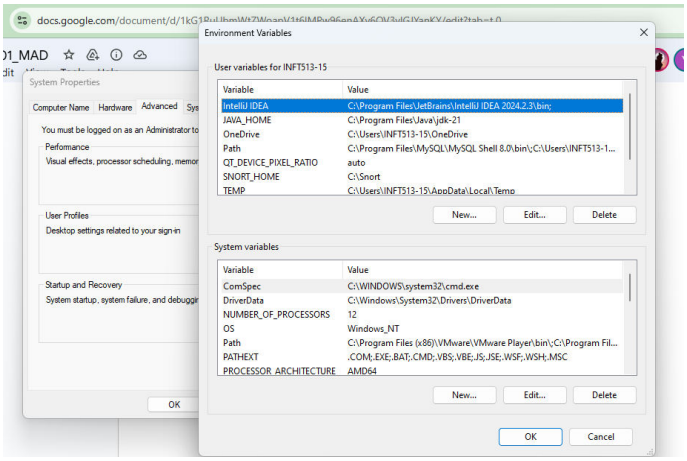
Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter.



Step 4: To run the Flutter command in regular windows console, you need to update the system

path to include the flutter bin directory. The following steps are required to do this:

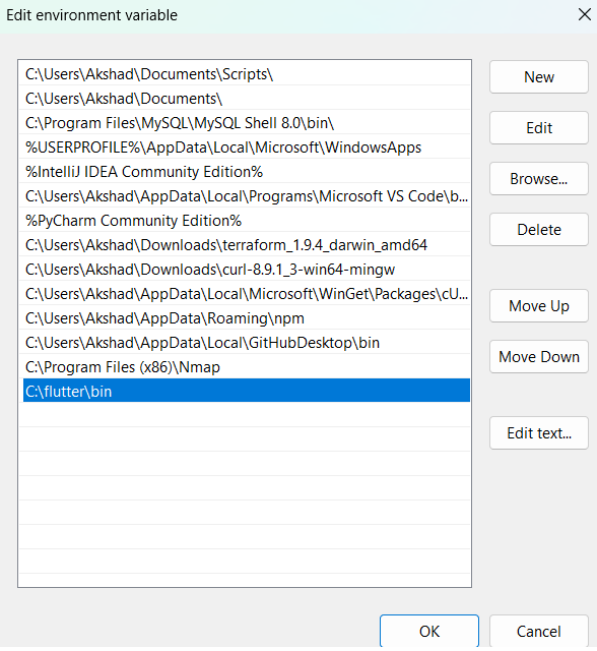
Step 4.1: Go to MyComputer properties -> advanced tab -> environment variables. You will get the following screen.



Step 4.2: Now, select path -> click on edit. The following screen appears

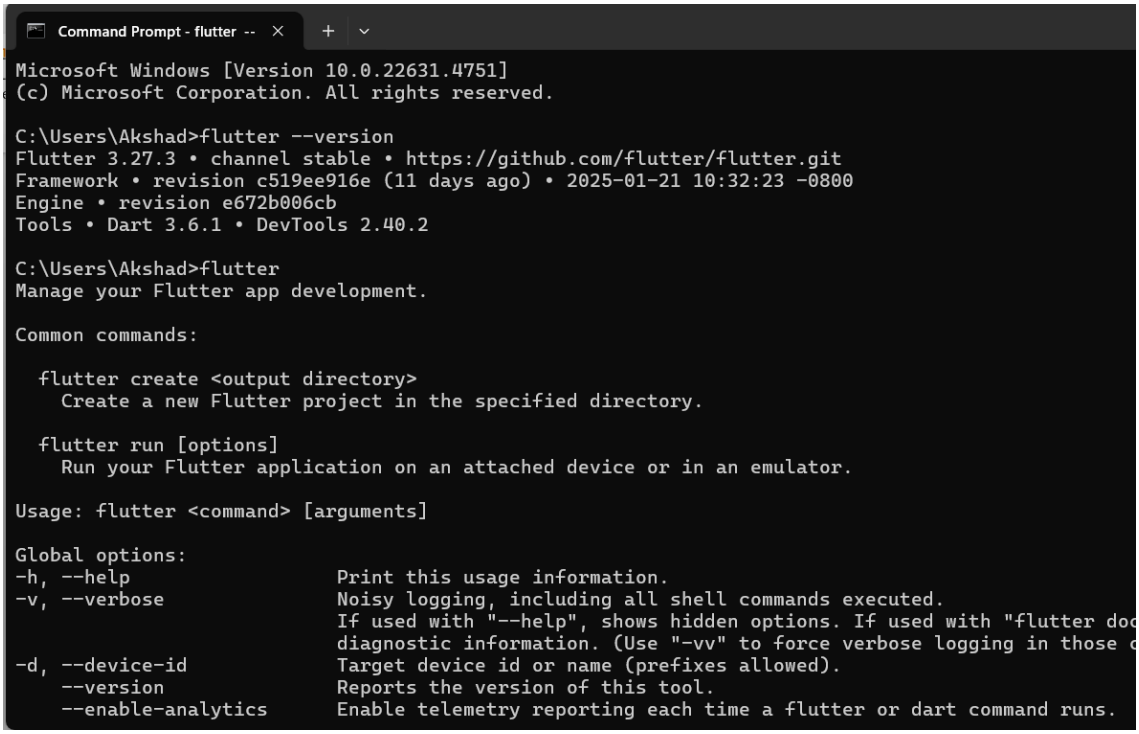
Step 4.3: In the above window, click on New->write path of Flutter bin folder in variable value -

> ok -> ok -> ok



Step 5: Now, run the $ flutter command in command prompt.

Now, run the $ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.



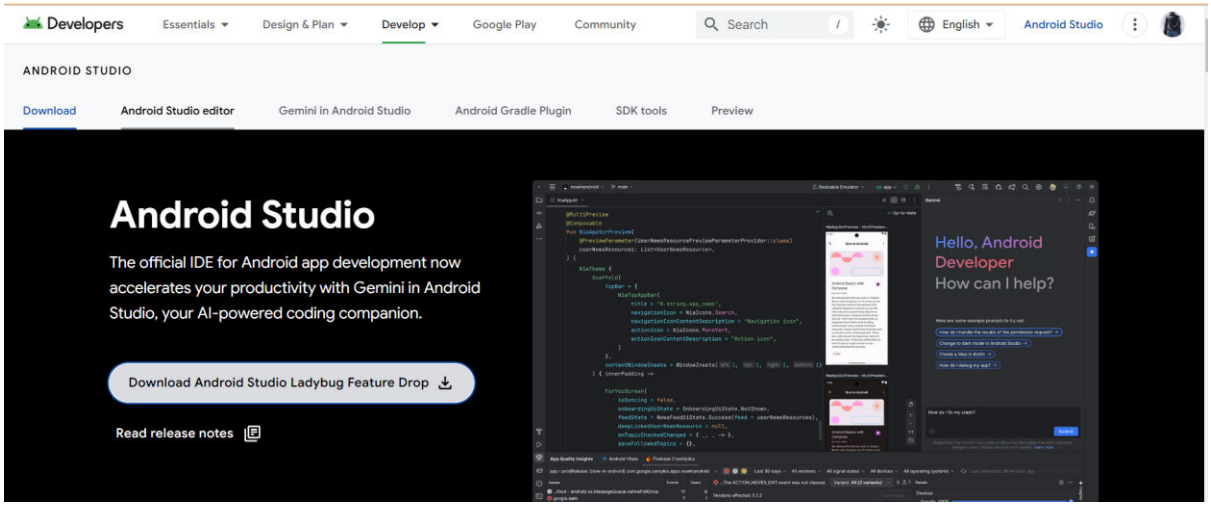
Step 6: When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which required to

run Flutter as well as the development tools that are available but not connected with the device.

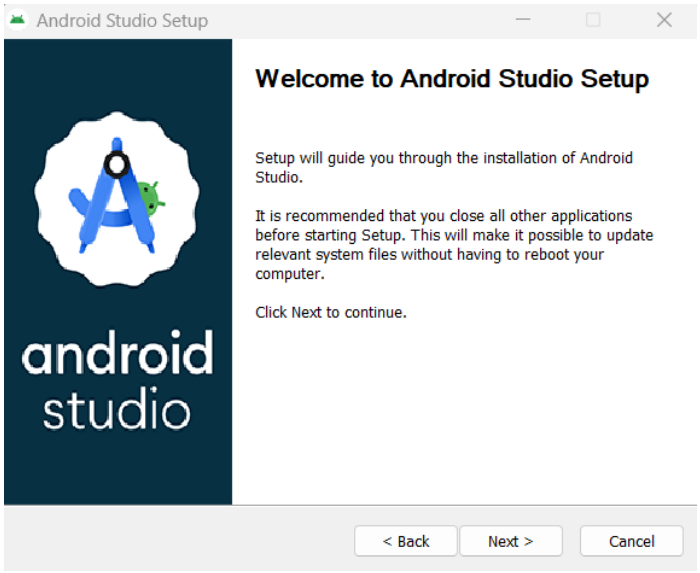
```
C:\Users\Akshad>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.226
[✗] Windows Version (the doctor check crashed)
    X Due to an error, the doctor check did not complete. If the error mess
      https://github.com/flutter/flutter/issues.
    X ProcessException: Failed to find "powershell" in the search path.
        Command: powershell
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/ind
      On first launch it will assist you in installing the Android SDK comp
      (or visit https://flutter.dev/to/windows-android-setup for detailed i
      If the Android SDK has been installed to a custom location, please us
      `flutter config --android-sdk` to update to that location.

[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows app
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including
[!] Android Studio (not installed)
[✓] VS Code (version 1.96.4)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 4 categories.

C:\Users\Akshad>
```

Step 7: Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

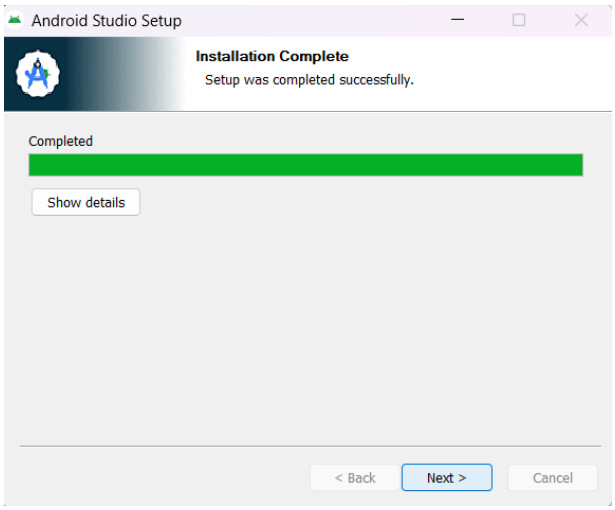
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box



Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to

choose the 'Don't import Settings option' and click OK. It will start the Android Studio.

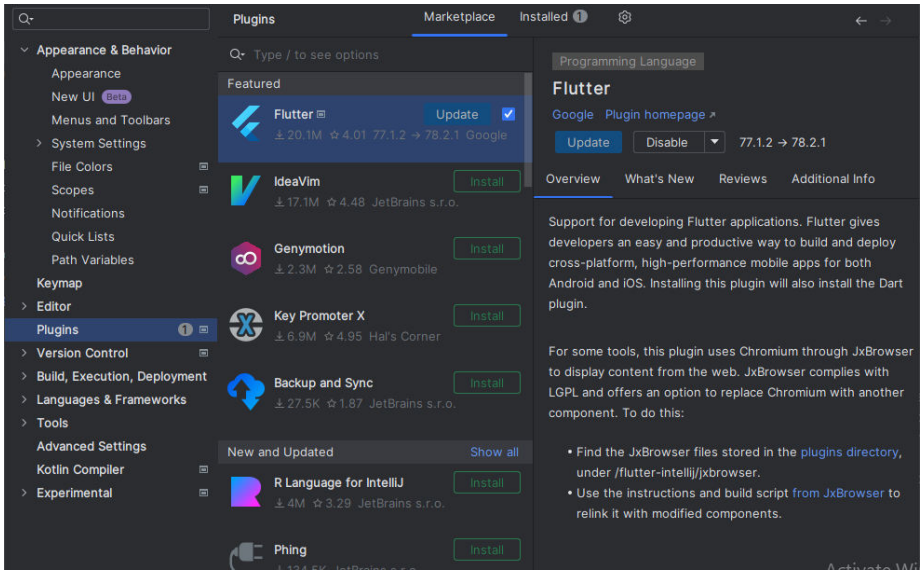
Step 8: Next, you need to set up an Android emulator. It is responsible for running and testing

the Flutter application.

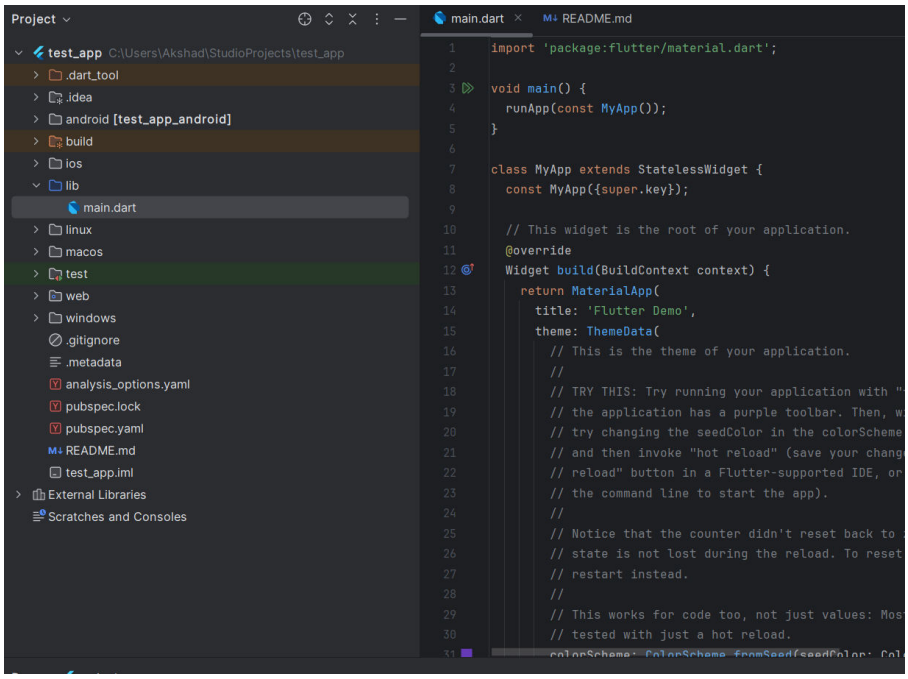
Step 8.1: To set an Android emulator, go to Android Studio > Tools > Android > AVD Manager

and select Create Virtual Device. Or, go to Help->Find Action->Type Emulator in the search

box. You will get the following screen.

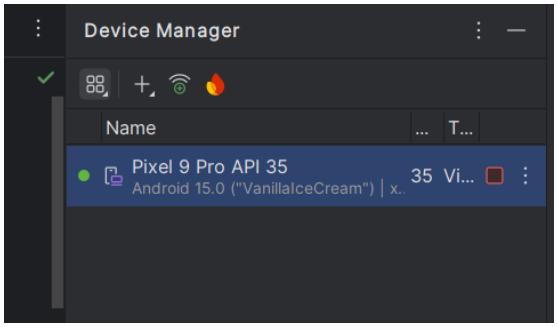


Step 8.2: Choose your device definition and click on Next.

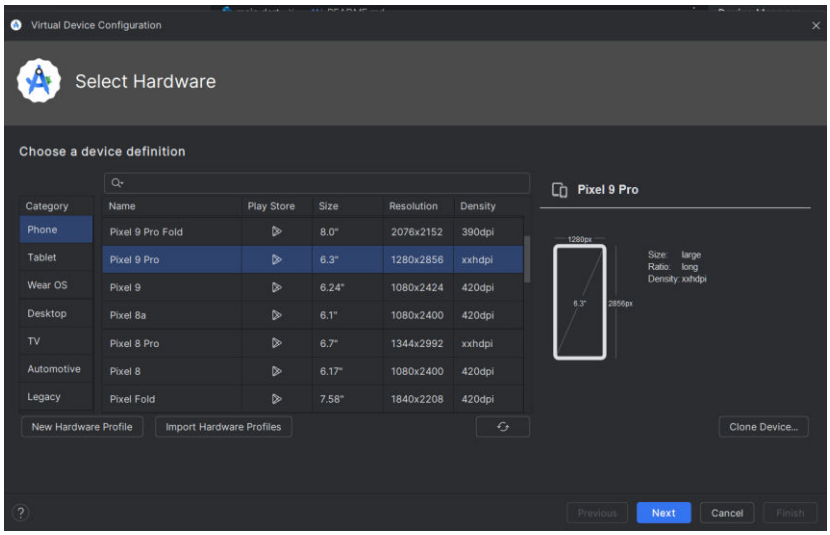


Step 8.3: Select the system image for the latest Android version and click on Next.

Step 8.4: Now, verify the all AVD configuration. If it is correct, click on Finish. The following screen appears.



Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator

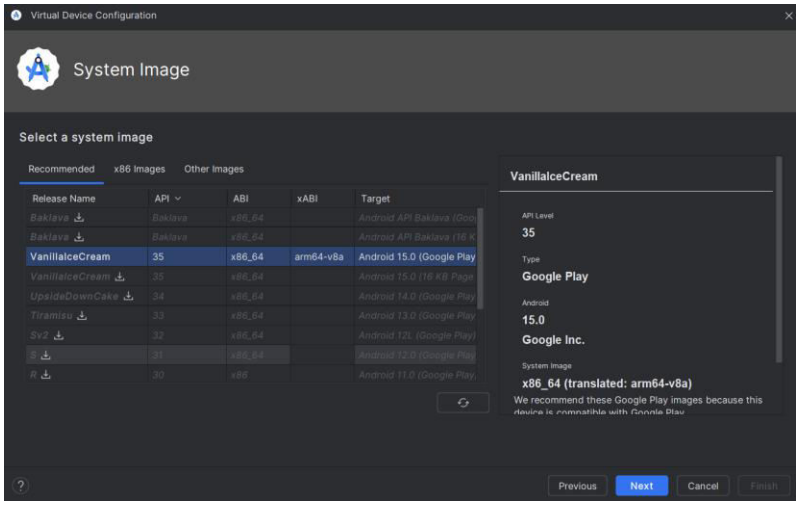


Step 9: Now, install Flutter and Dart plugin for building Flutter application in Android Studio.

These plugins provide a template to create a Flutter application, give an option to run and debug

Flutter application in the Android Studio itself. Do the following steps to install these plugins.

Step 9.1: Open the Android Studio and then go to File->Settings->Plugins.



Step 9.2: Now, search the Flutter plugin. If found, select Flutter plugin and click install. When

you click on install, it will ask you to install Dart plugin as below screen. Click yes to proceed.

Step 9.3: Restart the Android Studio.

NAME - Dhruv Maurya
DIV - D15B
ROLL NO - 31
MPL EXP 2

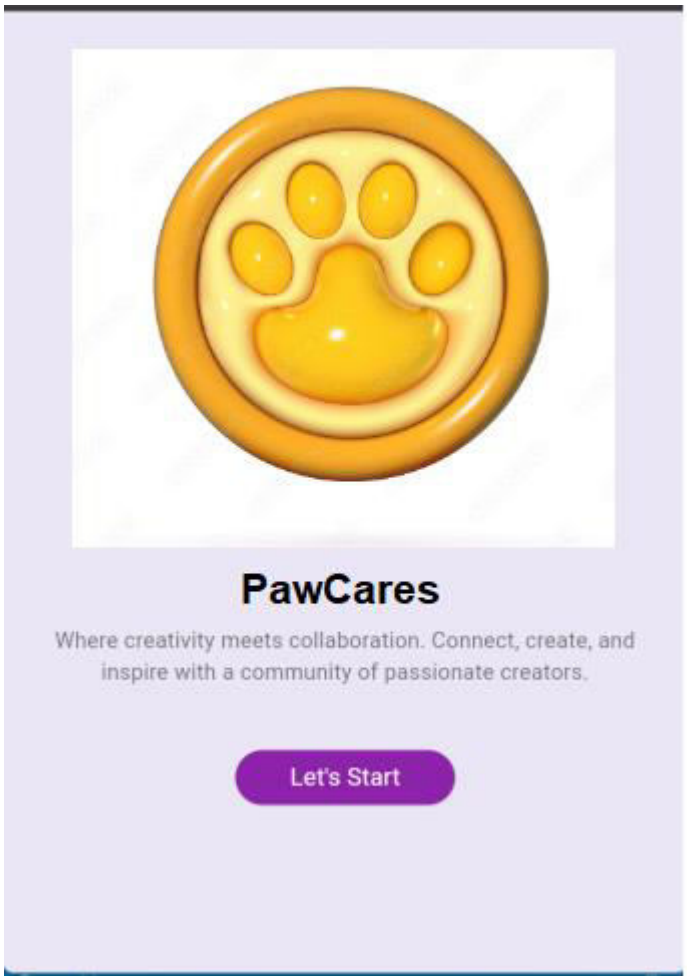
```
import 'package:flutter/material.dart';
import 'auth_service.dart';
import 'home_screen.dart';
import 'register_screen.dart';

class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final AuthService _auth = AuthService();

  void login() async {
    var user =
        await _auth.signIn(emailController.text, passwordController.text);
    if (user != null) {
      Navigator.pushReplacement(
          context, MaterialPageRoute(builder: (context) => HomeScreen()));
    } else {
      ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text('Login Failed')));
    }
  }
```

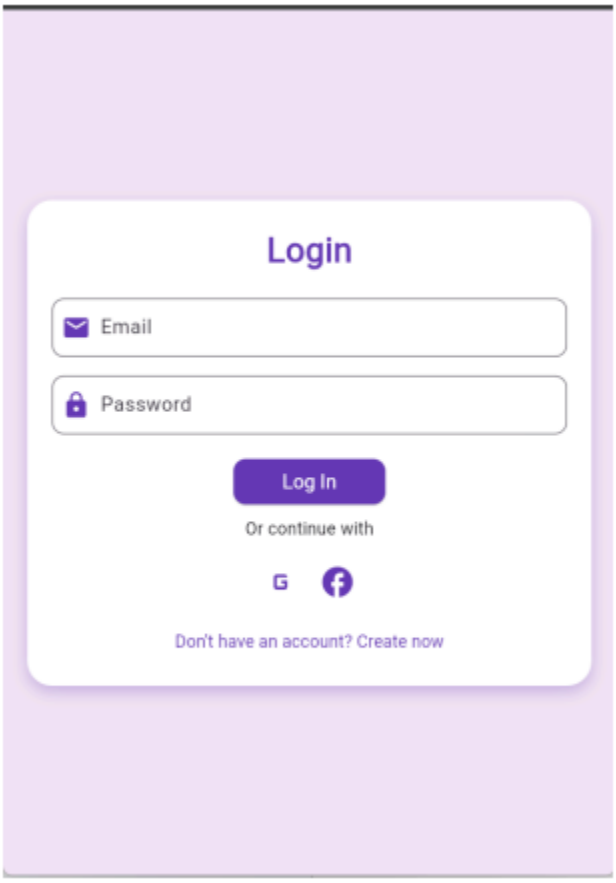
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Color(0xFFF3E5F5),
    body: Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Container(
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(20),
            boxShadow: [
              BoxShadow(
                color: Colors.deepPurple.withOpacity(0.3),
                blurRadius: 10,
                offset: Offset(0, 5),
              ),
            ],
          ),
          padding: EdgeInsets.all(20),
          child: Column(
            mainAxisSize: MainAxisSize.min,
            children: [
              Text(
                "Login",
                style: TextStyle(
                  fontSize: 28,
                  fontWeight: FontWeight.bold,
                  color: Colors.deepPurple,
                ),
              ),
              SizedBox(height: 20),
```

```
TextField(
  controller: emailController,
  decoration: InputDecoration(
    labelText: "Email",
    prefixIcon: Icon(Icons.email, color: Colors.deepPurple),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
),
SizedBox(height: 15),
TextField(
  controller: passwordController,
  decoration: InputDecoration(
    labelText: "Password",
    prefixIcon: Icon(Icons.lock, color: Colors.deepPurple),
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
  obscureText: true,
),
SizedBox(height: 20),
ElevatedButton(
  onPressed: login,
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.deepPurple,
    padding: EdgeInsets.symmetric(horizontal: 40, vertical: 15),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  ),
```

```
  child: Text("Log In",
      style: TextStyle(fontSize: 16, color: Colors.white)),
),
SizedBox(height: 10),
Text("Or continue with"),
SizedBox(height: 10),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    IconButton(
      icon: Icon(Icons.g_mobiledata,
          size: 30, color: Colors.deepPurple),
      onPressed: () {},
    ),
    IconButton(
      icon: Icon(Icons.facebook,
          size: 30, color: Colors.deepPurple),
      onPressed: () {},
    ),
  ],
),
SizedBox(height: 10),
TextButton(
  onPressed: () => Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => RegisterScreen()),
  ),
  child: Text(
    "Don't have an account? Create now",
    style: TextStyle(color: Colors.deepPurple),
  ),
),
```

```
            ],
          ),
        ),
      ),
    ),
  );
 }
}
```


PawCares
Where creativity meets collaboration. Connect, create, and inspire with a community of passionate creators.
Let's Start


Login
Email
Password
Log In
Or continue with
Don't have an account? Create now

11:06

PawCare

Find Your New Best Friend

7 pets available for adoption

Search pets by name, breed...

All | Dogs | Cats | Others

Rocky ♂

Rabbit

1 year

Daisy ♀

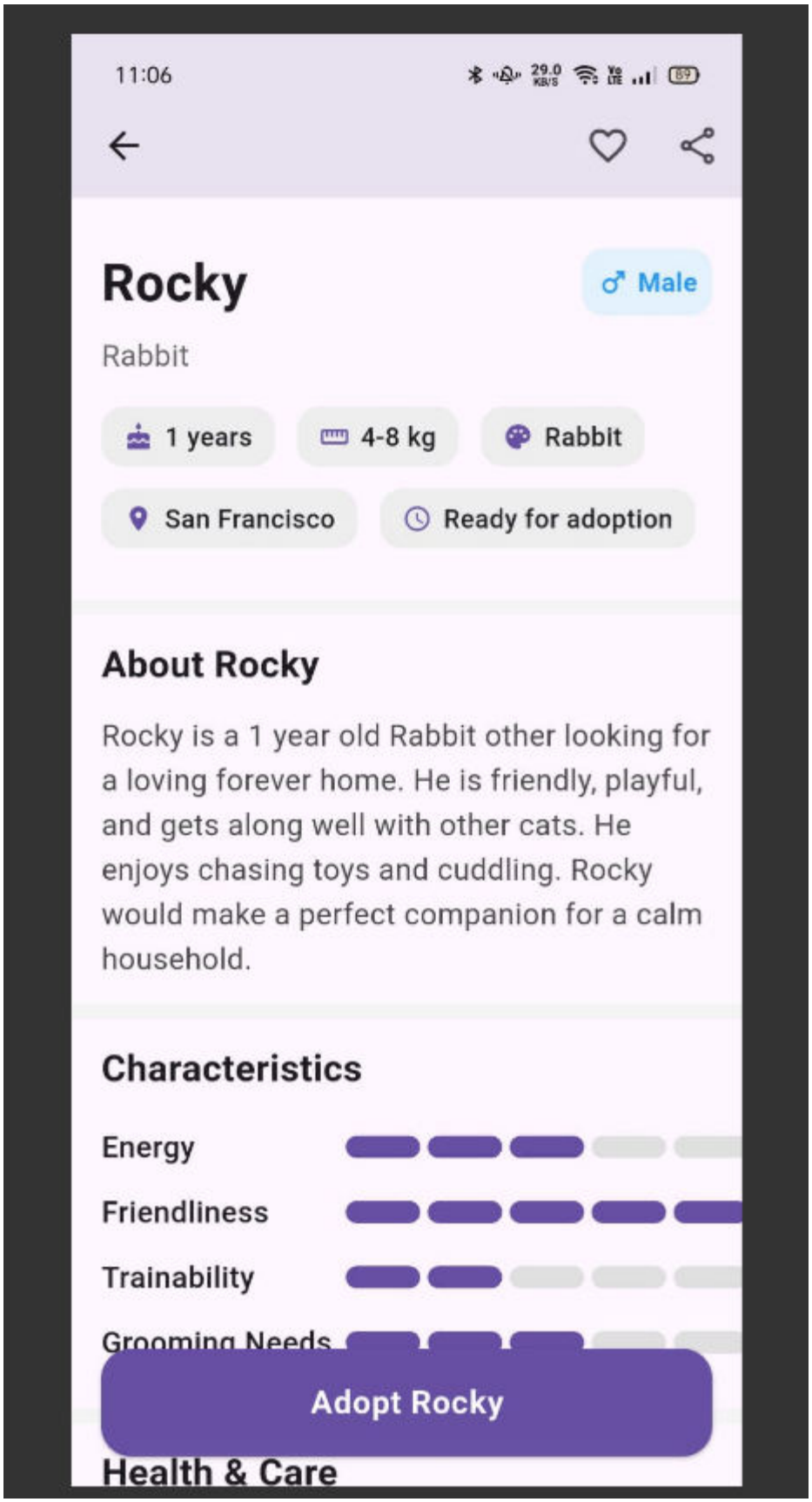
Poodle

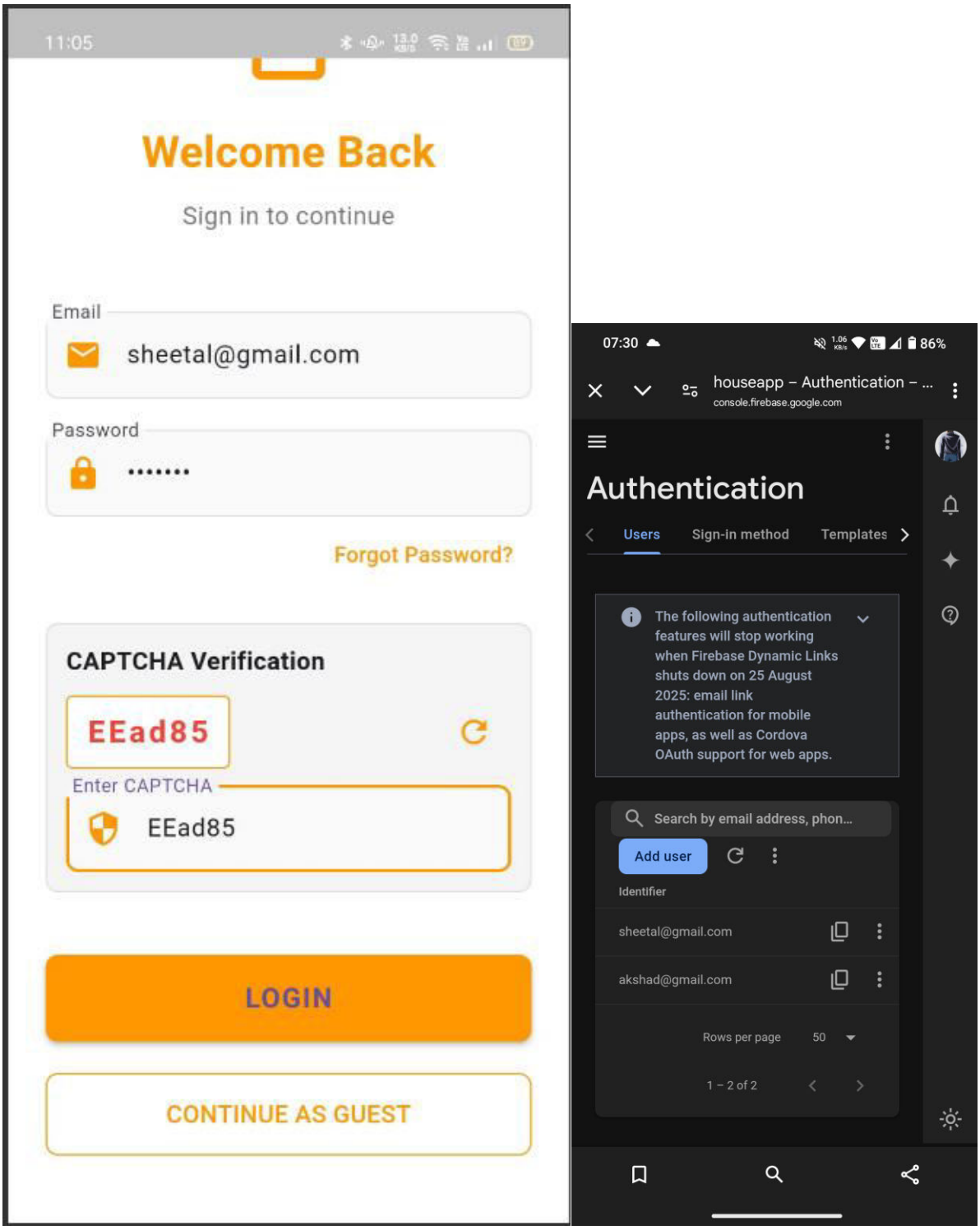
1 year

Luna ♀

Charlie ♂

My Pets | Shop | Cart | Profile


11:06
Rocky
Male
Rabbit
1 years
4-8 kg
Rabbit
San Francisco
Ready for adoption
About Rocky
Rocky is a 1 year old Rabbit other looking for a loving forever home. He is friendly, playful, and gets along well with other cats. He enjoys chasing toys and cuddling. Rocky would make a perfect companion for a calm household.
Characteristics
Energy
Friendliness
Trainability
Grooming Needs
Adopt Rocky
Health & Care


11:05
Welcome Back
Sign in to continue
Email
sheetal@gmail.com
Password
.....
Forgot Password?
CAPTCHA Verification
EEad85
Enter CAPTCHA
EEad85
LOGIN
CONTINUE AS GUEST
07:30
86%
houseapp – Authentication – ...
console.firebase.google.com
Authentication
Users
Sign-in method
Templates
The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.
Search by email address, phon...
Add user
Identifier
sheetal@gmail.com
akshad@gmail.com
Rows per page
50
1 – 2 of 2

PawCares

Pet Services

- Grooming
- Vet Checkup
- Adoption

Book an Appointment

Your Name

Pet Type

Book Appointment

Home Store Profile

PawCares - Buy Pets

$800 $600

German Shepherd

$750

Siberian Husky

$900

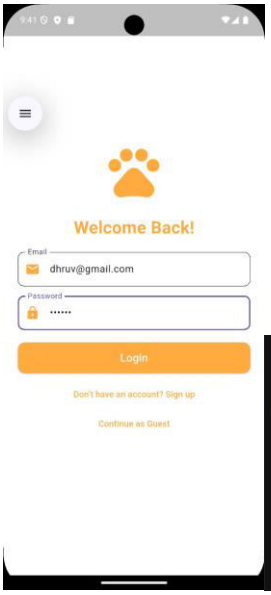
British Shorthair

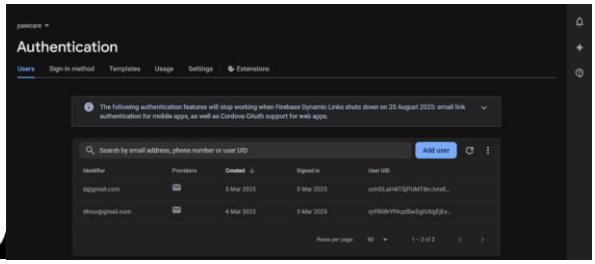
$650

Labrador Re

$850

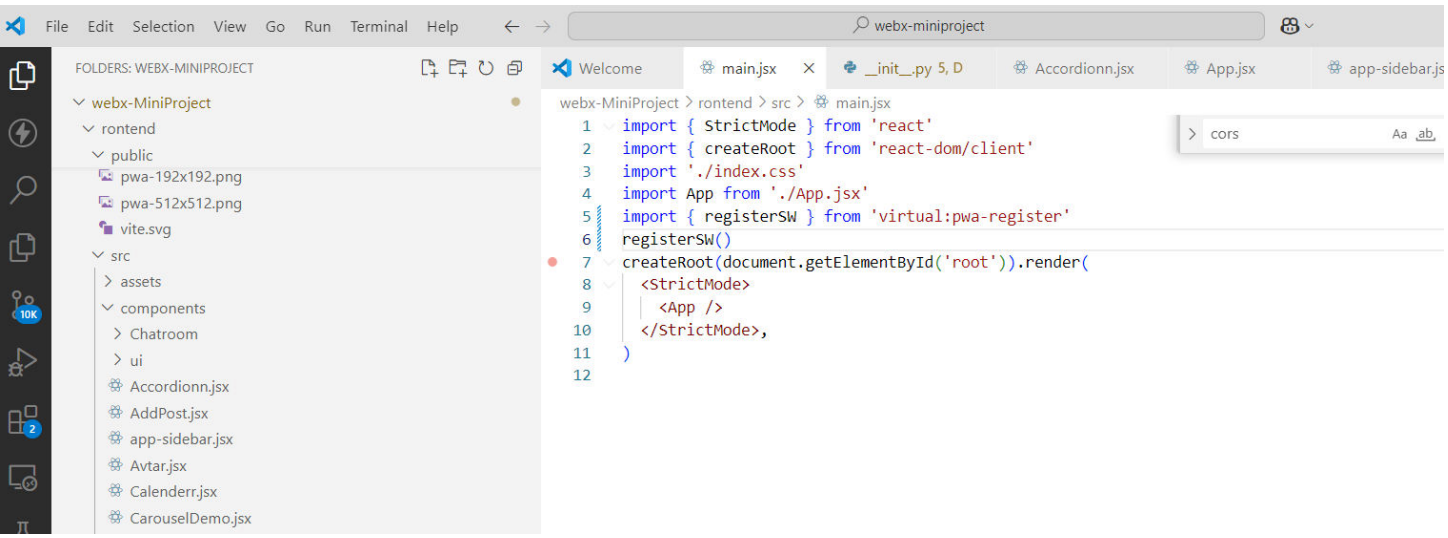
+

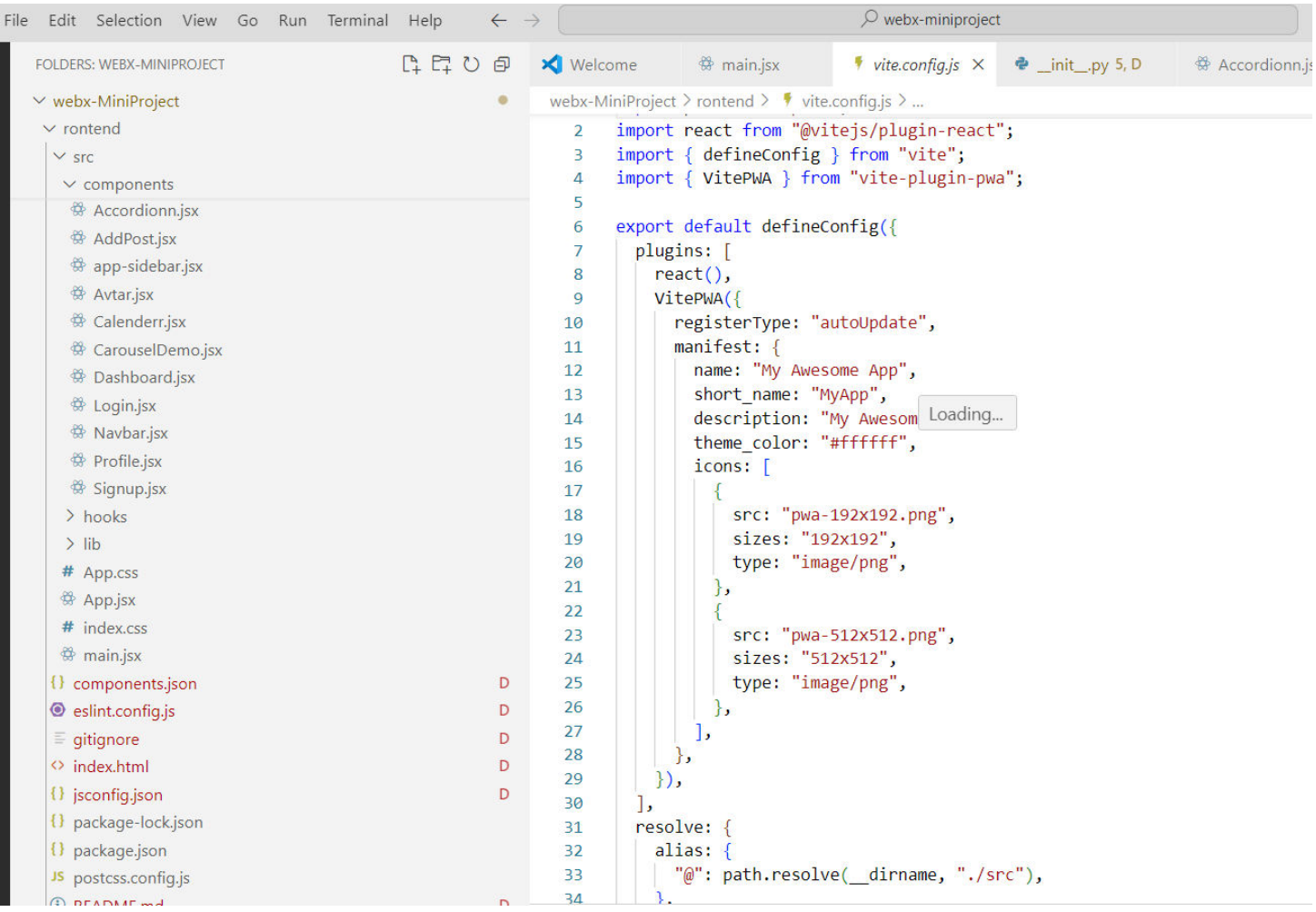

Welcome Back!
Email
dhruv@gmail.com
Password
Login
Don't have an account? Sign up
Continue as Guest

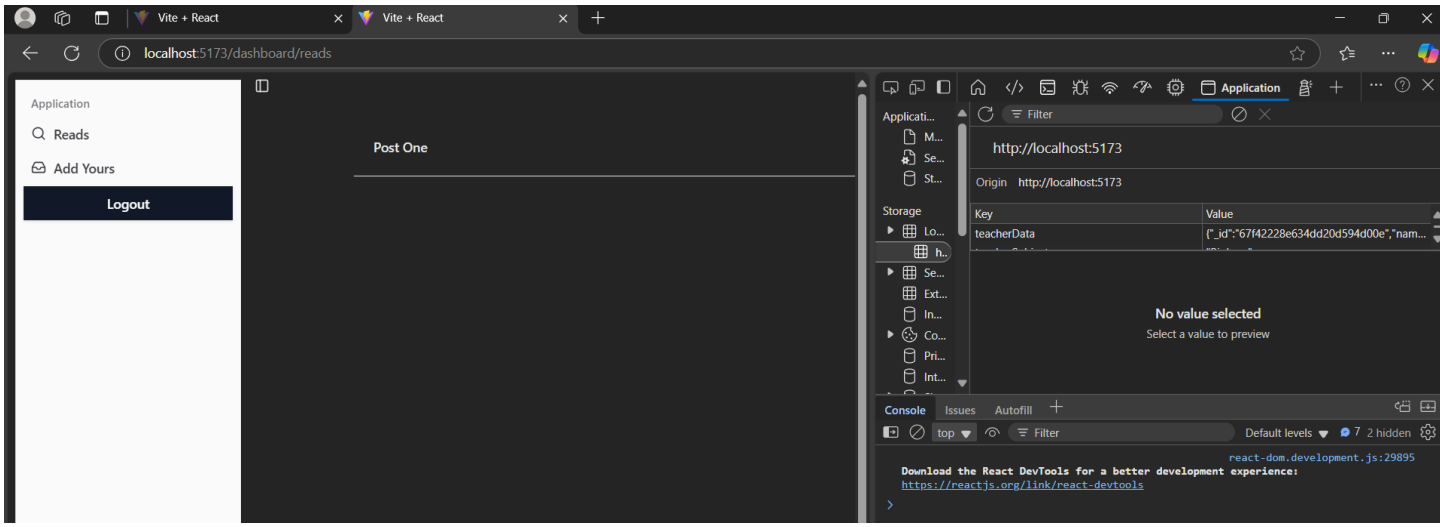

Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
Add user

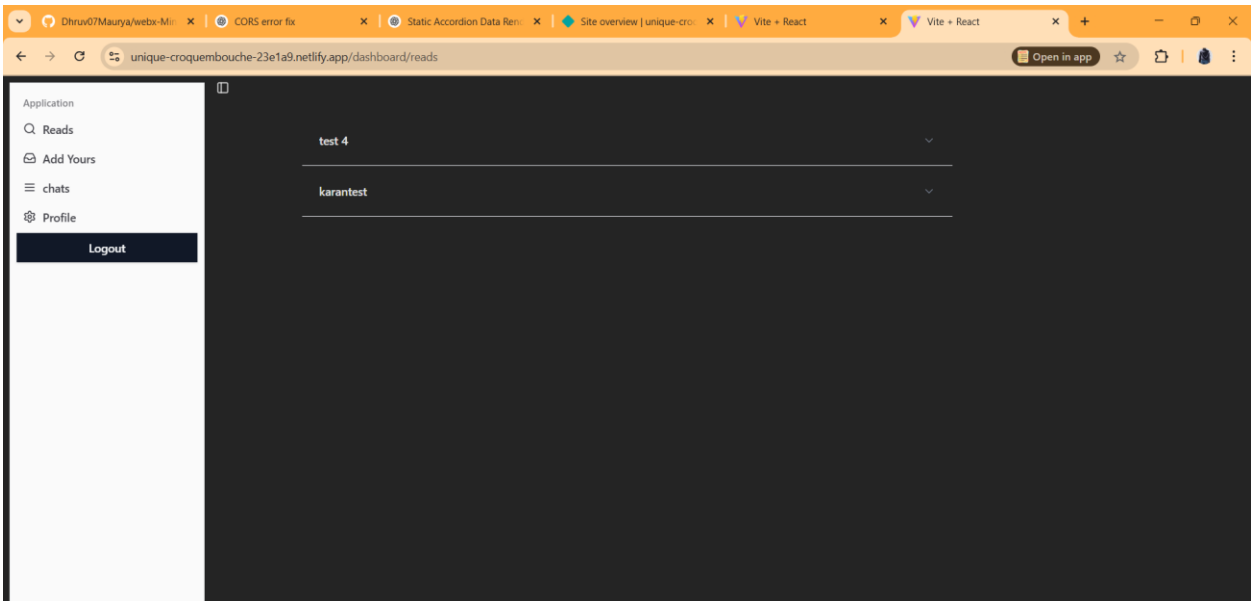
Name:Dhruv Maurya
Div:D15B
Roll No:31

PWA Practical 7



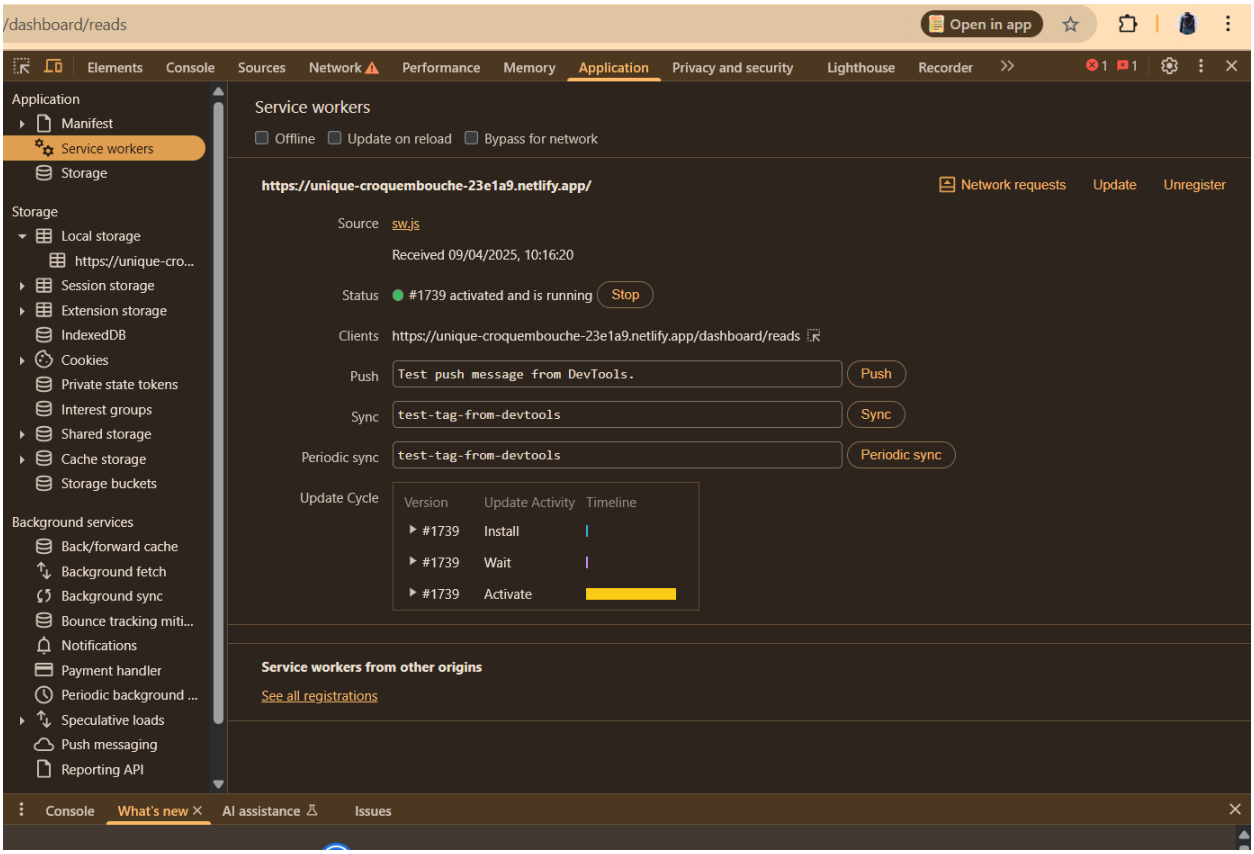



Vite + React
Vite + React
localhost:5173/dashboard/reads
Application
Reads
Add Yours
Logout
Post One
Application
Filter
http://localhost:5173
Origin http://localhost:5173
Key
Value
teacherData
{"_id":"67f42228e634dd20d594d00e","nam...
Storage
No value selected
Select a value to preview
Console
Issues
Autofill
top
Filter
Default levels
2 hidden
react-dom.development.js:29895
Download the React DevTools for a better development experience:
https://reactjs.org/link/react-devtools


CORS error fix
Vite + React
unique-croquembouche-23e1a9.netlify.app/dashboard/reads
Open in app
Application
Reads
Add Yours
chats
Profile
Logout
test 4
karantest

Name:dhruv maurya
Div:D15B
Roll No.:31

MPL Experiment No.8

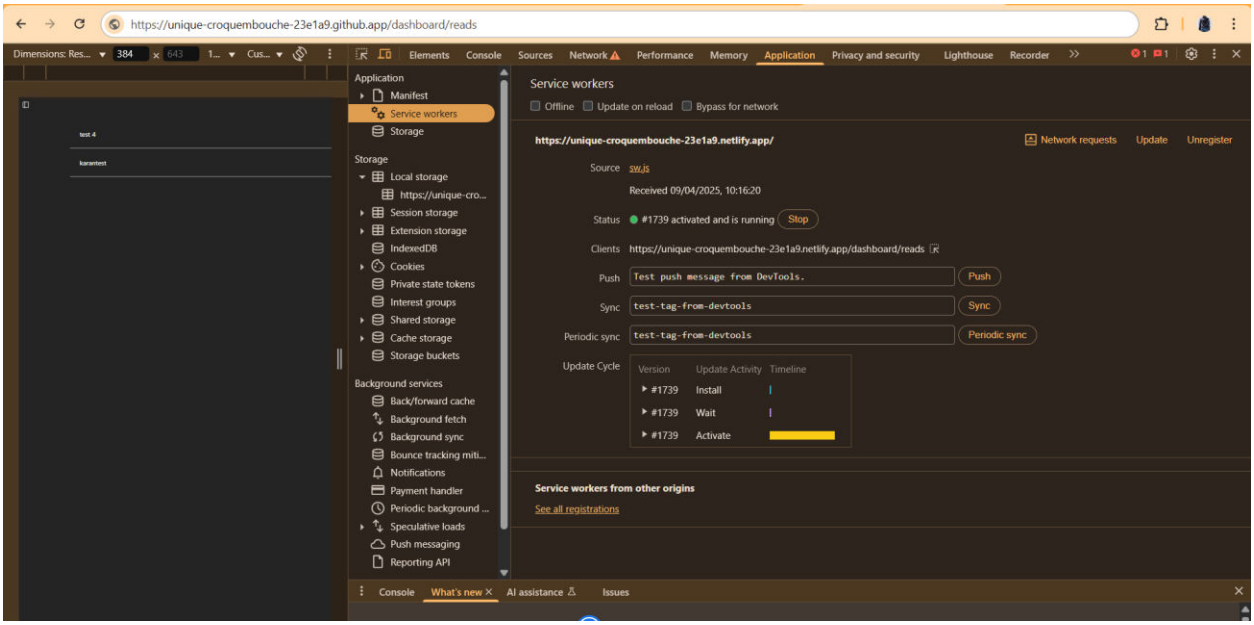


Name:Dhruv Maurya

Div:D15B

Roll No.:31

MPL Experiment No.9

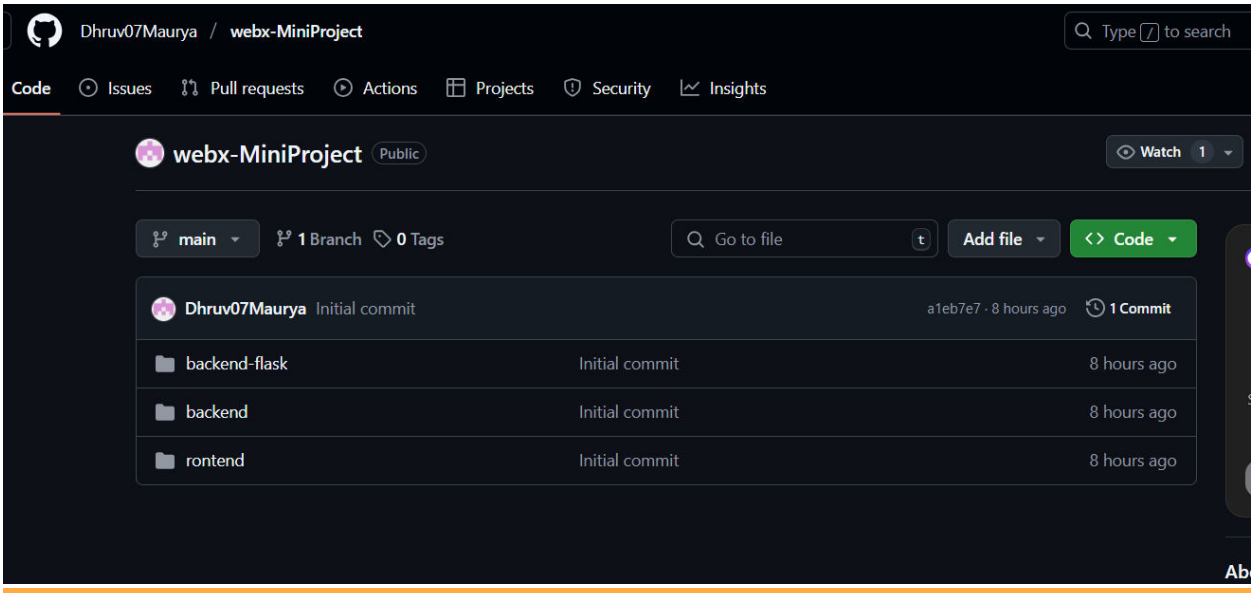


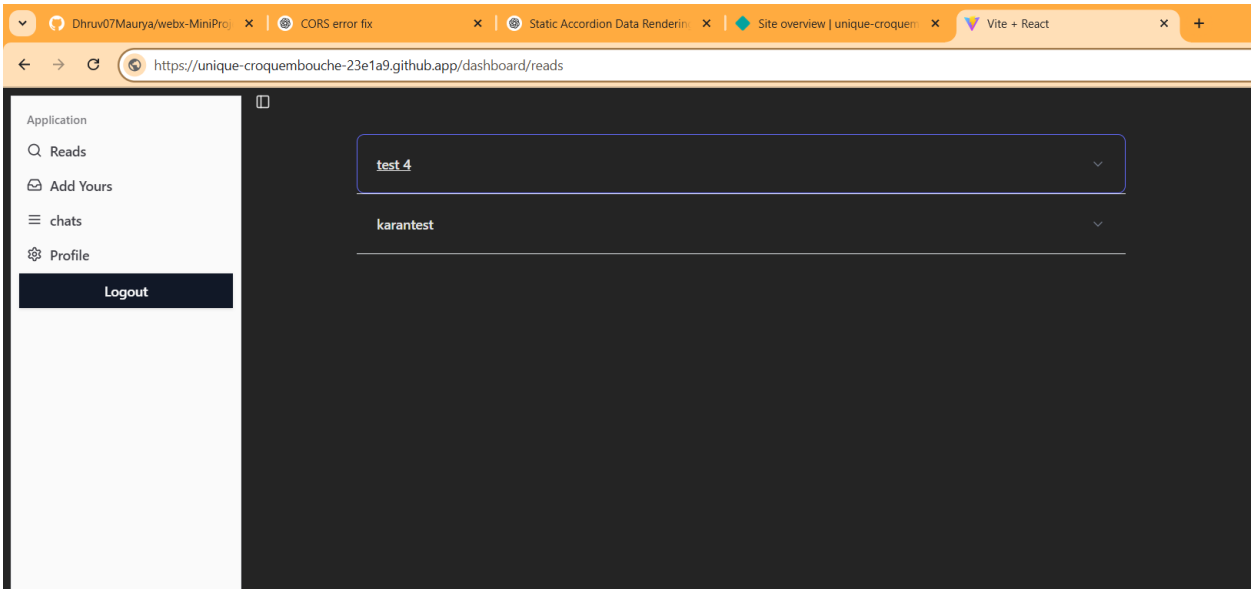
Name:Dhruv Maurya

Div:D15B

Roll No.:31

MPL Experiment No.10



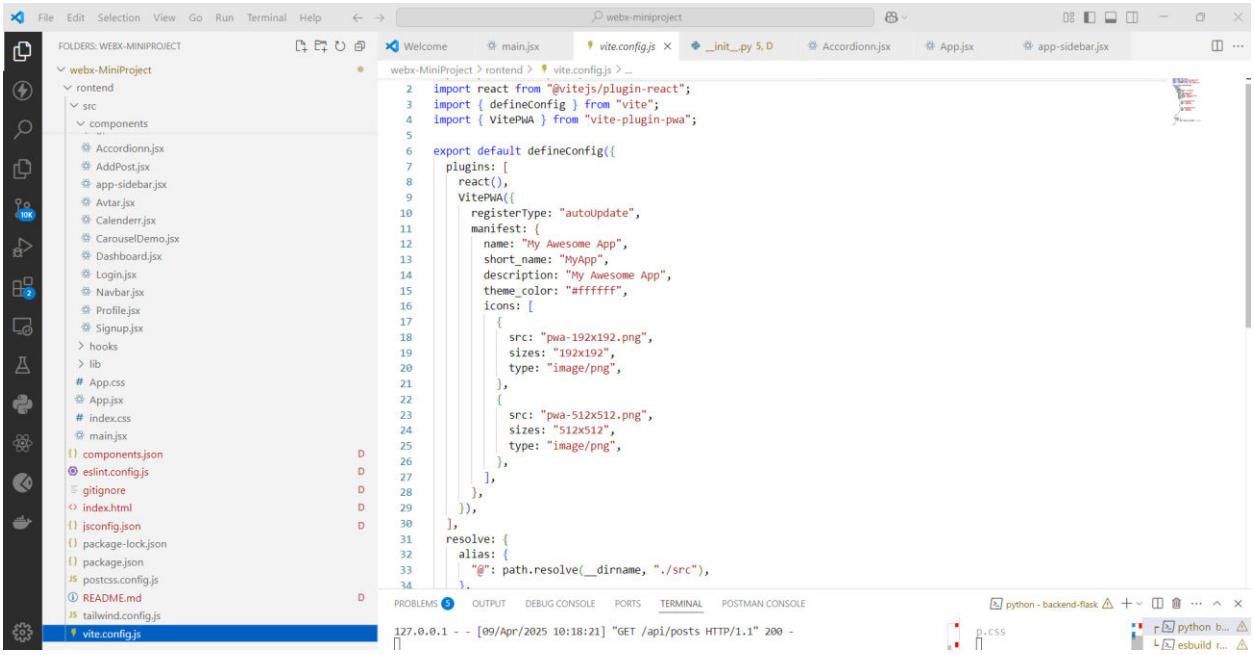


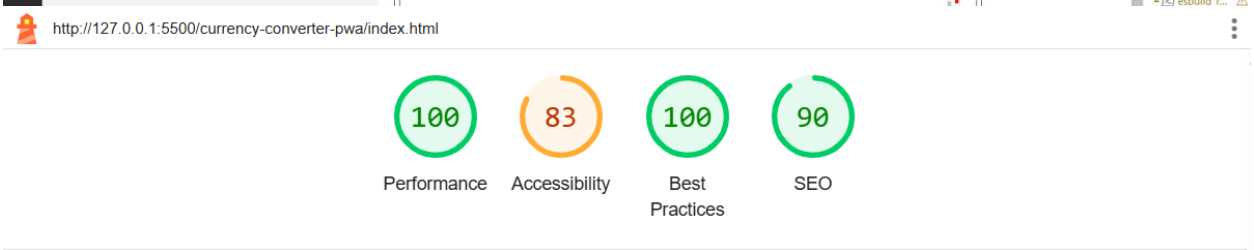
Name:Dhruv Maurya

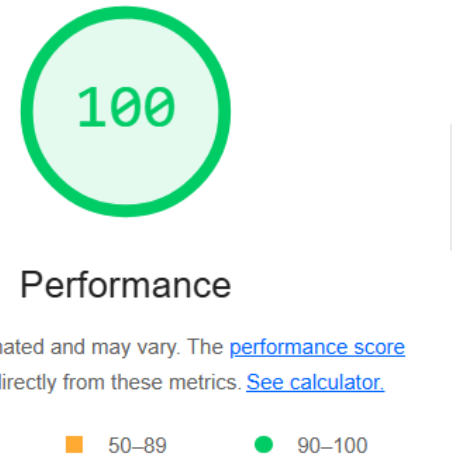
Div:D15B

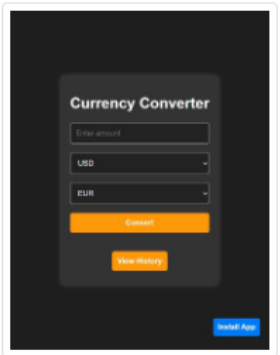
Roll No.:31

MPL Experiment No.11









METRICS Expand view

- First Contentful Paint: 0.3 s
- Largest Contentful Paint: 0.3 s
- Total Blocking Time: 0 ms
- Cumulative Layout Shift: 0
- Speed Index: 0.3 s

DIAGNOSTICS

- Eliminate render-blocking resources — Potential savings of 70 ms
- Page prevented back/forward cache restoration — 1 failure reason
- Enable text compression — Potential savings of 4 KiB
- Avoid chaining critical requests — 2 chains found
- Minimize third-party usage — Third-party code blocked the main thread for 0 ms
- Largest Contentful Paint element — 250 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

Dhruv Maurya D15B/31

MAD Assignment No. 1 A1 DATE: 04

Explain key features and advantages of using flutter for mobile ~~web~~ app development/

① Single code base for multiple platforms
② Hot Reload. Instantly see changes in the app without restarting making developement faster
③ Fast performance: uses Dart language and a compiled approach for smooth high performance app
④ Open source and Strong community Support Backend by google and a large developer community ensuring continous improvements & resources
⑤ Reduced performance issues: The app runs natively without relying on intermediate bridges like in react native reducing lag

Discuss how the flutter framework differ from traditional approaches & why it has gained popularity in developers community.

① Single codebase vs Separate codebase
Traditional approach relies on platform native component
uses a single dart based code-
platform reducing developm

@ Rendering Engine vs UI Compone

Traditional approach: Developers re
rely UI components which can lead
inconsistencies and performance issu
flutter: uses the Skia rendering engi
draws everything from scratch e
a consistent UI across devices

Why Flutter has gained popularity.

1. Faster development with hot reload.
Can instantly see UI changes without
restarting the app making the itera
process much quicker
2. Cross platform inconsistency
Business save time and resource
maintaining a single codebase for m
platforms
3. Consistent UI across devices: Since flutt
does not rely on native component
the UI looks and behaves the same
different versions
4. Improved performance:
AOT compilation and direct acces
to GPU rendering ensuring smooth
animations and high performance

DATE:

Describe the concept of the Widget tree in flutter Explain how widget composition is used to build complex user interfaces.

Widget tree in flutter:
In flutter, the widget tree is the fundamental structure that represents the UI of an application. It is a hierarchical arrangement of widgets where each widget defines a part of the user Interface.
Flutter UI is entirely built using widgets which can be stateless or stateful
The widget tree determines how the UI is rendered and updated when changes occur

Benifits of widgets Composition.

1. Reusability
2. Maintainability
3. Performance

b. Provides Example of Commonly used and their roles in creating a wi tree

⇒

1. Structural widget:
This widget act ast the foundation building the UI
Material App: The root widgets of th App that provides the essential con Scaffold provides a basic layout structur including an app bar, body, floatin action button etc
Container: A versatile widget used for p padding, margin & background custo

Ex: MaterialApp (
home: Scaffold (
appBar: AppBar (title: Text (
body: Container (
padding: EdgeInsets.all
child: Text ("Hello, Flut
));

DATE:

Inputs & Interaction Widgets:

Textfeild - Accepts text input from users
Elevation Button - A button with elevation
Gesture Detector - Detects like taps. Swipes and long press

Ex:
```
Column (
  children [
    Textfield ( decoration : Input Decor
  Elevate Button (
    onPressed : () {
      print ("Button pressed") }
      child Text ( Button Subm
  ) ; } ] ) ;
```

Display & Styling widgets.
Text - display text on the screen
Image - shows images from assets network or memory
Icon : Display Icons
Card - A material design card with rounde corners and elevation.

```
column (
  children [
    Text ("Welcome") ; )
    Image. network ("url") ] ) ;
```

Q.3)

a. Discuss the important State man
in flutter applications

→

In flutter State refers to data th
change during the lifetime of
application. This includes:

User Input
UI changes
Network changes
Animation States

There are two type of States
· Ephermal state : Small UI specific state
does not affect the whole App.
· App wide states: Data shared accross
wedgets.

Importance of state management:
1) Efficient UI updates
2) Code maintainability.
3) Data consistancy
4) Synchronizat

b. Compare & contrast the different state manage
Approaches avaliable in flutter such as set
State, provider & riverpod. Provide scenario
→ where each approach is evitable.
Set state:

DATE:

Senerios for Each approach

- use StState when managing Simple UI Elements within a Single Widget
- use provider wne changing State across multiple Widget

Explain the process of integrating firebase with flutter application discuss the benefits of using firebase a a backend Solution

1. Create a firebase project from Console
2. Register the flutter app with firebase Enter the android package name and download Jsons
3. Install Firebase dependinces
4. configure firebase for android & Ios
5. Initialize firebase in flutter

Benifits of using firebase:

(1) Easy to setup & scale

(2) Authentication
Provides email / password, etc A

(3) Cloud storage : Stores Assets on cl

(4) Push Notification : Sends real t
Notification

b. Highlight the firebase services commo
used in flutter development & provide a
overview of how data synchronization is
achieved

⇒ (1) Firebase Authentication
(2) cloud Firebase
(3) Realtime Database
(4) Firebase Cloud Messaging
(5) Firebase Analysis
(6) Firebase Hosting

Data Synchronization in firebase
Firebase ensures real time data synchroni
across multiple devices and platforms usi
firebase & real time Database

DATE:

Cloud Firestore Sync Mechanism:
Uses real time listners to update UI Instantly when data changes
Ex: Firebase Firestore.instance.collection("users").
Snapshots().listen((Snapshot>{
for(var doc in Snapshot.does){
print(doc['name']);}});

Real time database Sync Mechanism
Uses Persistent websockets connections for live updates.
Ex Database reference ref = Firebase Database.instance.
ref("messages");
ref.onValue.listen((event){
print(event.Snapshot.Value);

Offline Data Sync

Firestore caches data locally & Syncs changes when the device is online

Cloud functions for Automated updates

Automates backend logic like triggers when data changes

Dhruv Maurya D15B/31

MPL Assignment - 2 DATE: 02/05/24

1. Define progressive web App (PWA) and Explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps.
- Significance of PWA in Modern web development
- Cross platform compatibility
- Offline Support
- Fast performance
- No app store required
- Lower development cost

Key difference between PWA & traditional mobile Apps

Feature	PWA	Traditional Mobile App
Installation	Direct from brower	Download from app store
Internet required	Work offline with Caching	Usually requires Internet
Performance	Fast with service workers	Faster but needs installation
Updates	Automatic no app store approval	Manual updates needed
Development Cost	Lower	Higher

Q.2) Define responsive web design and explain in the context of progressive web apps. Contrast responsive fluid and adaptive web design approaches

→ Responsive web design (RWD) is technique that progress adjust automatically to different screen and devices

- But Importance of responsive design is PWA
- Better User experience: PWA work smoothly device
- SEO benefits: google ranks responsive like's

Comparison of web design approaches

Approach	How it works	Pros
Responsive	Use flexible grids and CSS media queries to adjust layout.	Work on all devices (complex design)
Fluid	Use percent based width instead of fixed pixels to elements resize smoothly	works well on different screen sizes easy to implement
Adaptive	Uses fixed layouts that change at specific breakpoints	Optimized for known screen sizes required design for screen size

DATE:

Key difference
Responsive adapts dynamically to all screen

3) Describe the life cycle of service worker including to registration installation and activation phase.
Lifecycle of service workers
A service worker is a script that runs in the background and helps a web app work offline, load faster and send push notifications.
1. Registration phase
The browser registers the service worker using JavaScript code Eg :-

```
If ('service worker in navigator') {
    Navigator.serviceWorker.register ('/sw.js')
    then (() => console.log ('service worker registered))
    Catch (error => Console.log (Registration failed error))
```

2) Installation phase
The service worker downloads necessary (HTML, CSS, JS) and stores them in cache.
Code Ex.

```
self.addEventListener ('install', event => {
    event.waitUntil (
        caches.open (app.cache).then (cache =>
        return cache.addAll ([ '/', 'index.html',
        'style.css' ]);
    )
})
```

This ensures the app loads even without the inte

3. Activation phase
- The old Source worker is replaced with the
- Unused cache files from the previous versio

decided final step fetch x Sync

Q.4) Explain the use of Indexed DB in the Syste

→ Worker of data Storage.

Use of Indexed DB in Service worker for dat
Storage indexed DB is browser database th
stores large amounts of structed data lik
retrieve data efficiently.

Why use Indexed DB in Service workers?

1. Offline Support - Stores data when offline and s
it later.
2. Efficient storage - Stores data in structu
like user setting, cart items or form inputs

Opening the database

let ds

let request = indexed DB. Open (My database)
request. onsuccess = function (event) {
ds = event. target. result.
}

Creating a store & adding data

request. onupgradeneeded = function (event) {
let db = event target result
let store = db. create object store (users,
{key path, id})
store add ({id: 1, name: John ax. age. 25})