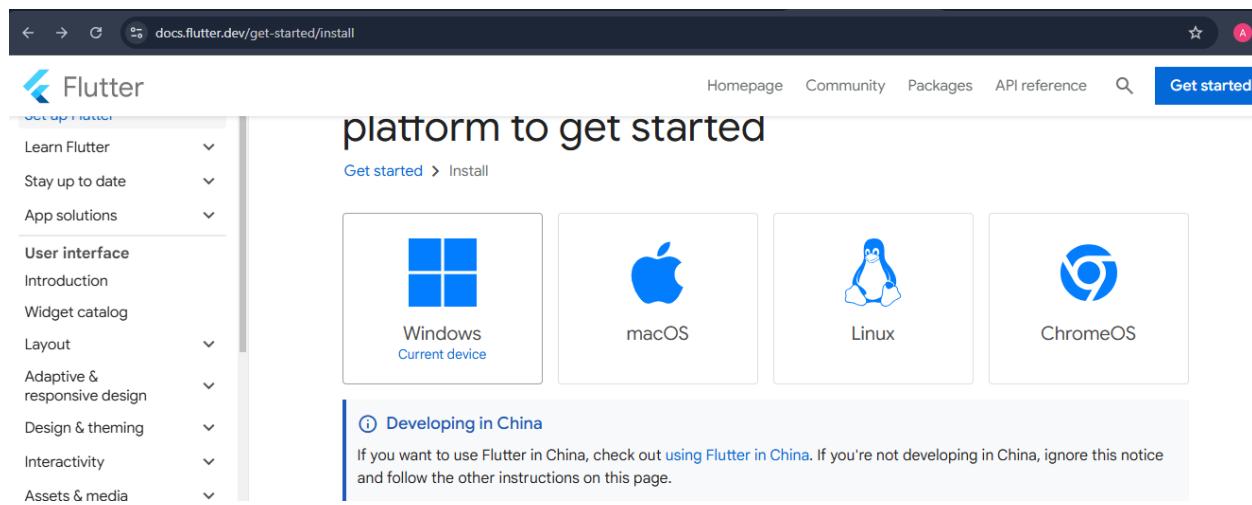


Name: Prathamesh Shetty
Div: D15B
Roll No: 53

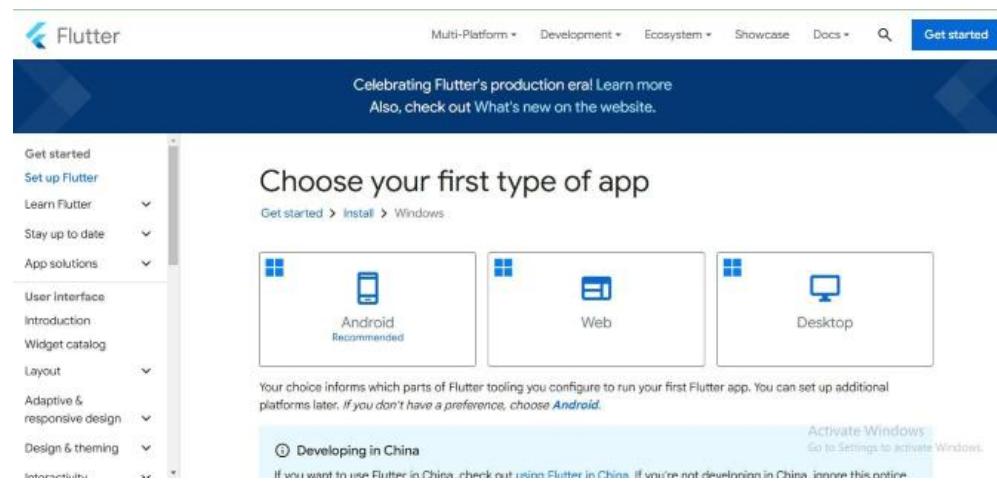
Experiment No:1

Aim:

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.



The screenshot shows the 'Get Started' page for Flutter. On the left, there's a sidebar with navigation links like 'Get started', 'Set up Flutter', 'Learn Flutter', 'Stay up to date', 'App solutions', 'User interface', 'Introduction', 'Widget catalog', 'Layout', 'Adaptive & responsive design', 'Design & theming', 'Interactivity', and 'Assets & media'. The main content area has a heading 'platform to get started' and a breadcrumb 'Get started > Install'. It features four cards representing different platforms: Windows (Current device), macOS, Linux, and ChromeOS. Below these cards is a notice about developing in China: 'If you want to use Flutter in China, check out [using Flutter in China](#). If you're not developing in China, ignore this notice and follow the other instructions on this page.'



The screenshot shows the 'Choose your first type of app' section. The sidebar on the left remains the same as the previous screenshot. The main content has a heading 'Choose your first type of app' and a breadcrumb 'Get started > Install > Windows'. It displays three cards: 'Android Recommended' (with a smartphone icon), 'Web' (with a laptop icon), and 'Desktop' (with a monitor icon). A note below the cards says: 'Your choice informs which parts of Flutter tooling you configure to run your first Flutter app. You can set up additional platforms later. If you don't have a preference, choose [Android](#)'. At the bottom, there's a 'Developing in China' notice and a link to activate Windows.

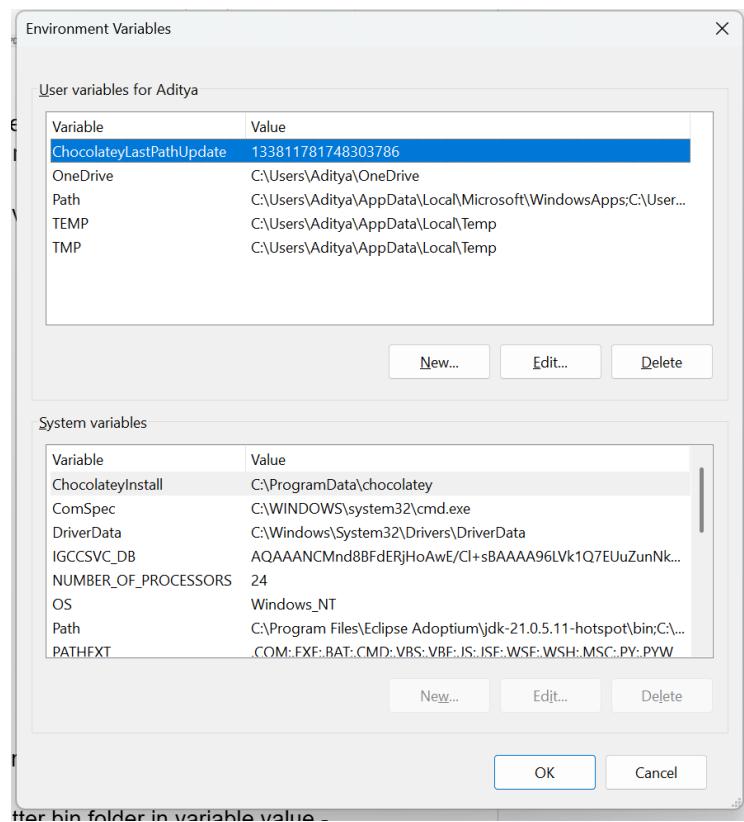
Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

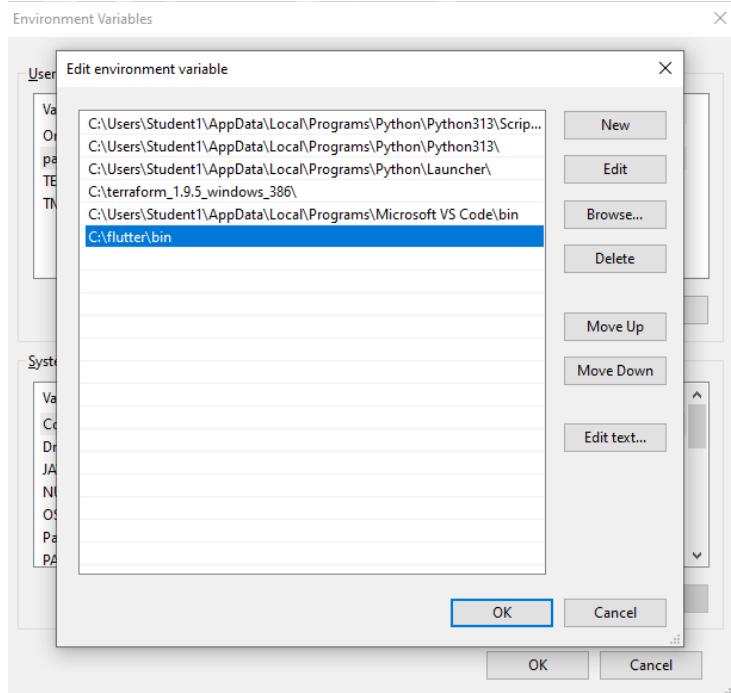
Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter.

The screenshot shows the Flutter documentation website at docs.flutter.dev/get-started/install/windows/mobile. The left sidebar has a 'Get started' menu with 'Set up Flutter' selected. The main content area has a 'Recommended' section suggesting Visual Studio Code 1.86 or later and the Flutter extension for VS Code. Below it is the 'Install the Flutter SDK' section, which includes a 'Download and install' button. To the right is a 'Contents' sidebar with links to various setup and configuration topics.

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.

```
Microsoft Windows [Version 10.0.22631.4830]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>flutter --version
Flutter 3.27.4 • channel stable • https://github.com/flutter/flutter.git
Framework • revision d8a9f9a52e (10 days ago) • 2025-01-31 16:07:18 -0500
Engine • revision 82bd5b7209
Tools • Dart 3.6.2 • DevTools 2.40.3

C:\Users\user>
```

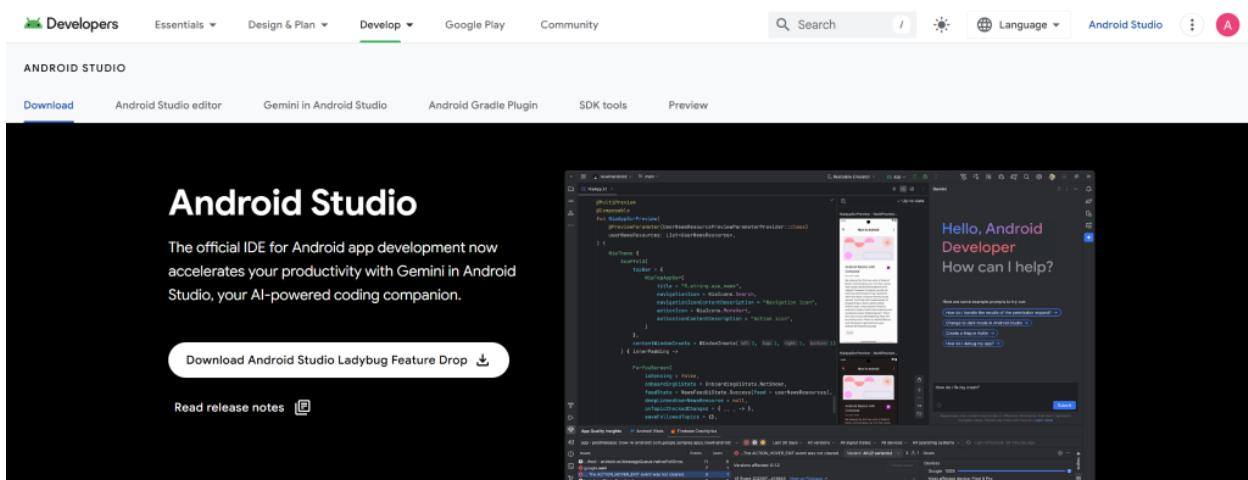
Now, run the \$ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

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.

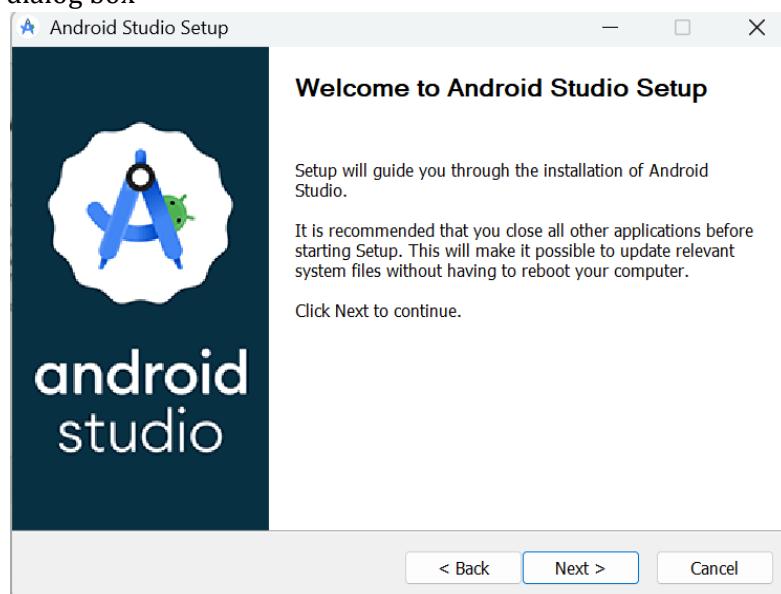
```
C:\Users\user>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.4, on Microsoft Windows [Version 10.0.22631.4830], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.31)
[✓] Android Studio (version 2024.2)
[✓] VS Code (version 1.97.0)
[✓] Connected device (3 available)
[✓] Network resources
```

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.

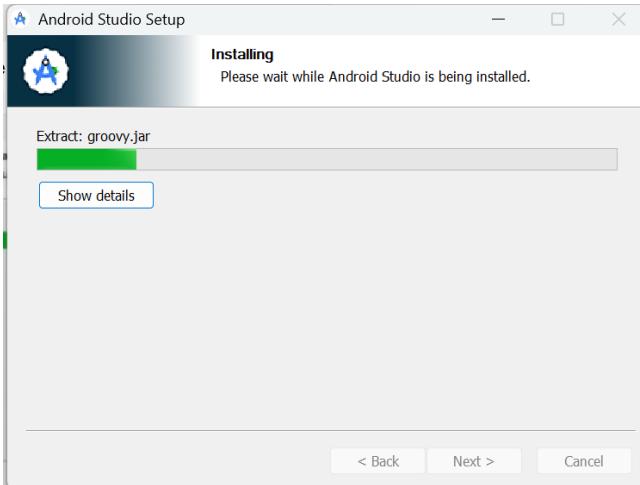
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



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



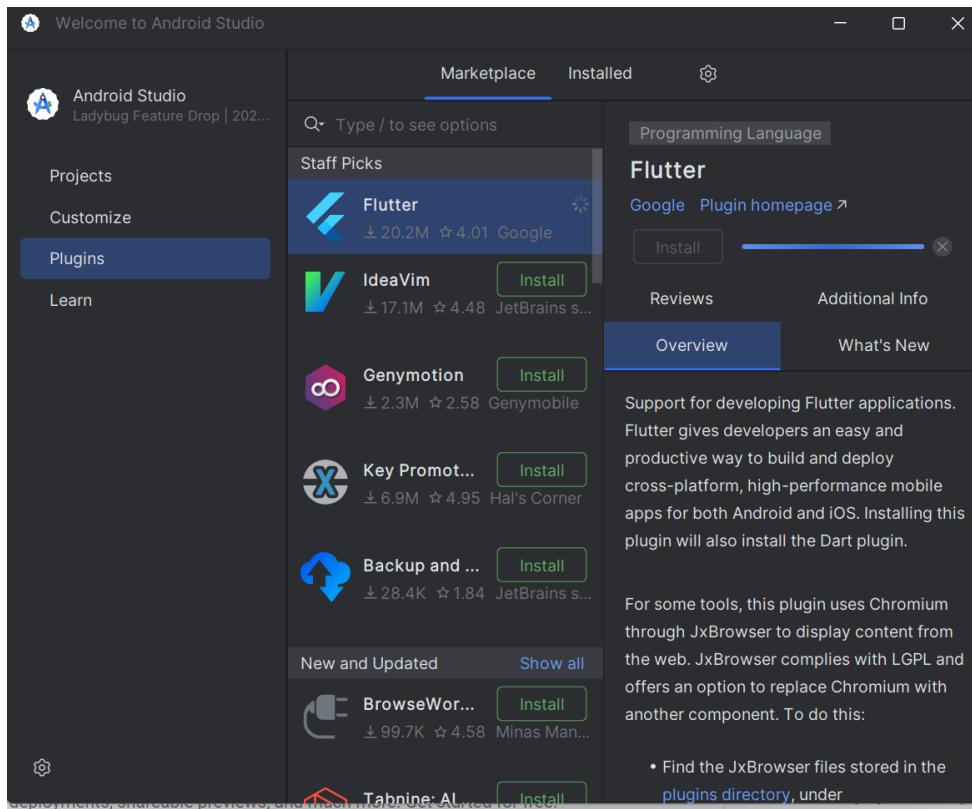
Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



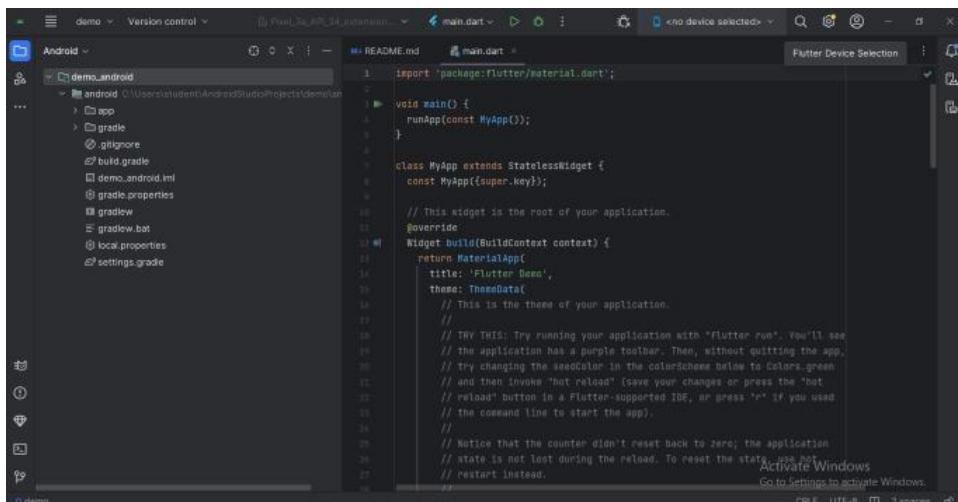
Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the 'Don't import Settings option' and click OK. It will start the Android Studio.

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.



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor contains the main.dart file for a Flutter application. At the top of the screen, there is a 'Flutter Device Selection' bar with a dropdown menu set to 'no device selected'. Below the editor, there is a message: 'Activate Windows. Go to Settings to activate Windows.' The bottom of the screen shows standard OS X keyboard shortcut keys.

```
import 'package:flutter/material.dart';

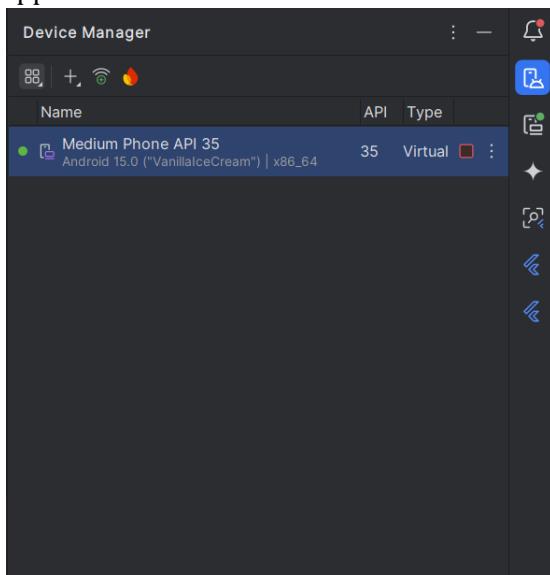
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

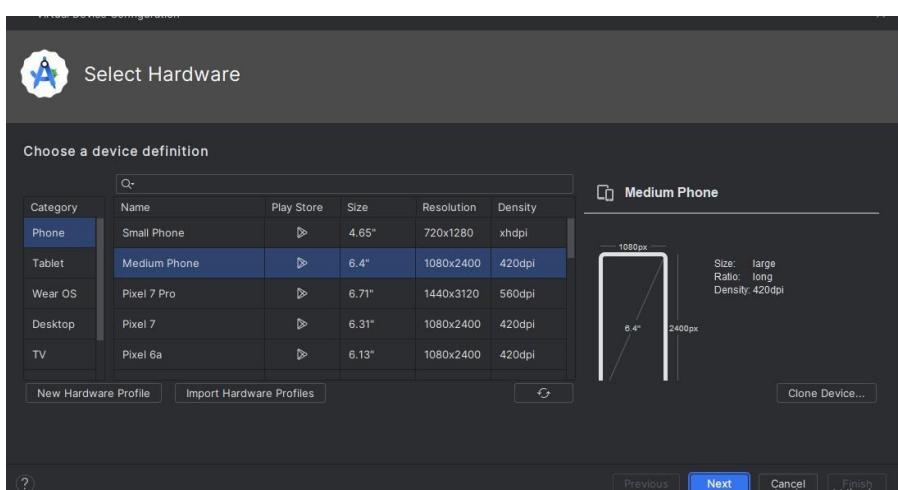
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
      ),
    );
  }
}
```

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.

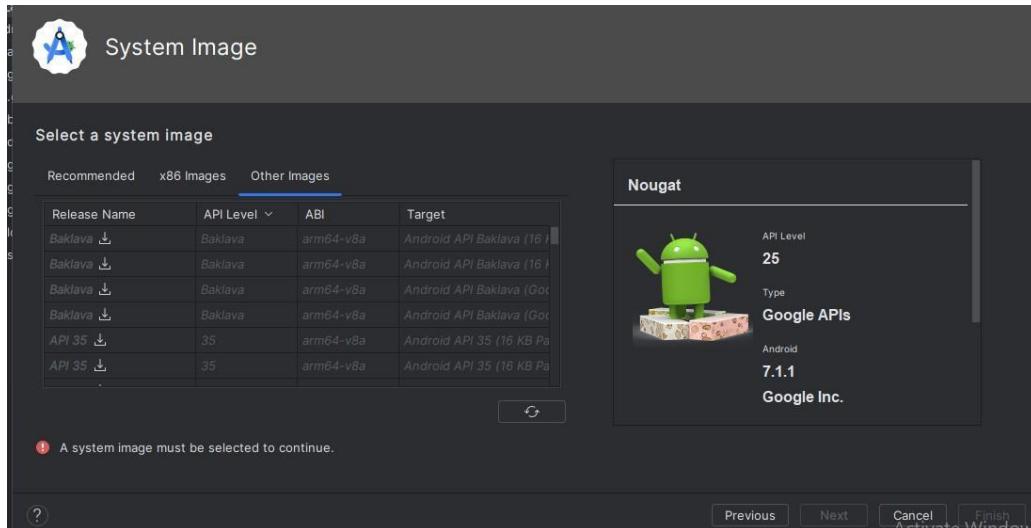


Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator



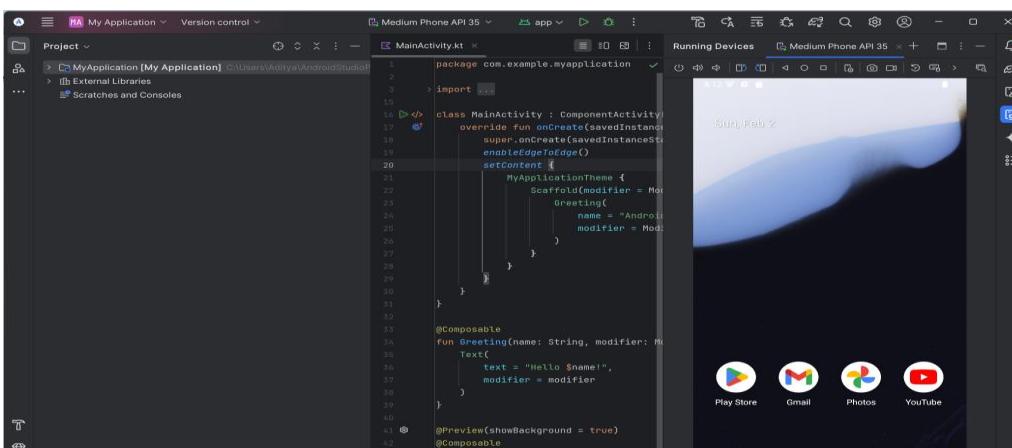
Step 9: Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself. Do the following steps to install these plugins.

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.



Conclusion:

In conclusion, the experiment was successfully completed by installing **Android Studio** and **Flutter** on the device. This setup is essential for developing and running Flutter applications. The process involved downloading and configuring the necessary tools, ensuring that both the **Android Studio IDE** and the **Flutter SDK** were properly installed and configured with the required dependencies. After completing the setup, the environment was successfully verified using the flutter doctor command, confirming that the system was ready for Flutter development. This foundational step paves the way for developing mobile applications using Flutter in the future.

Name: Prathamesh Shetty

Class: D15B

Roll No: 53

MPL Experiment 2

In **Flutter**, the entire user interface (UI) is built using **widgets**. A widget is essentially a **descriptive representation** of part of the user interface, whether it's a simple button, an image, a list, or even the layout structure of the screen itself. Unlike some other frameworks where UI elements are pre-built components, in Flutter, **everything is a widget**.

Widgets in Flutter are **immutable** (cannot be changed directly), and they are often used to **build and describe the UI** in terms of properties, layout, and interaction. A widget is **just a blueprint**, but when combined with a **StatefulWidget**, it allows the UI to **react to state changes** like user input or asynchronous operations.

1 Stateless vs Stateful Widgets

- ◆ **Stateless Widgets**

- These widgets **do not change** once built.
- Used for static UI elements like text, icons, and images.

- ◆ **Stateful Widgets**

- These widgets **can change dynamically** during runtime.
- Used for interactive UI components like buttons, text fields, and animations.

2 Visible (Structural) Widgets

These widgets define the **visual layout** of the UI.

- ◆ **Container Widgets** – Used for layout styling, including padding, margin, and background color.
- ◆ **Text Widgets** – Displays text in different styles.
- ◆ **Image Widgets** – Displays images from assets, network, or memory.
- ◆ **Button Widgets** – Includes ElevatedButton, TextButton, and IconButton for user interaction.
- ◆ **List & Grid Widgets** – Used for displaying scrollable lists (ListView) or grids (GridView).

3 Layout (Structural) Widgets

These widgets help in arranging other widgets on the screen.

- ◆ **Row & Column** – Arranges widgets horizontally or vertically.
- ◆ **Stack** – Places widgets on top of each other.
- ◆ **Expanded & Flexible** – Adjusts widget size dynamically based on available space.
- ◆ **SizedBox** – Adds spacing between widgets.

4 Interactive Widgets

Widgets that handle user interactions.

- ◆ **GestureDetector** – Detects gestures like taps, swipes, and long presses.
- ◆ **TextField** – Used for user input.
- ◆ **Checkbox, Radio, and Switch** – Used for selections and toggles.
- ◆ **Slider** – Allows selecting values within a range.

5 Animation & Effects Widgets

These widgets help in adding animations and special effects.

- ◆ **AnimatedContainer** – Allows smooth transitions when properties change.
- ◆ **Hero** – Provides shared element transitions between screens.
- ◆ **Opacity** – Controls widget transparency.
- ◆ **Transform** – Allows rotation, scaling, and translation of widgets.

6 Navigation & Routing Widgets

These widgets help in navigating between different screens.

- ◆ **Navigator** – Manages the navigation stack.
- ◆ **PageRoute** – Defines different types of screen transitions.
- ◆ **Drawer** – A slide-out navigation menu.
- ◆ **BottomNavigationBar** – A tabbed navigation bar at the bottom.

7 State Management Widgets

These widgets help in managing the state of an application.

- ◆ **InheritedWidget** – The base class for propagating state down the widget tree.
- ◆ **Provider** – A widely used state management solution.
- ◆ **Bloc & Riverpod** – Advanced state management approaches for complex apps.

Code:

main.dart

```
import 'package:flutter/material.dart';
import './screens/home_page.dart'; // Import the HomePage

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'ShareIt',
      debugShowCheckedModeBanner: false, // Removes the debug banner
      theme: ThemeData(
        primarySwatch: Colors.lightBlue,
        scaffoldBackgroundColor: Colors.white,
      ),
      home: const HomePage(), // Use the HomePage as the home screen
    );
  }
}
```

home_page.dart

```
import 'package:flutter/material.dart';

class HomePage extends StatelessWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'ShareIt',
          style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: Colors.white,
          ),
        ),
        backgroundColor: Colors.lightBlue,
        elevation: 0,
      ),
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(

```

```
begin: Alignment.topCenter,
end: Alignment.bottomCenter,
colors: [
    Colors.lightBlue.shade300,
    Colors.white,
],
),
),
),
child: SingleChildScrollView(
    child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
                const Text(
                    'Welcome to ShareIt!',
                    style: TextStyle(
                        fontSize: 32,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                        shadows: [
                            Shadow(
                                offset: Offset(2, 2),
                                blurRadius: 3.0,
                                color: Color.fromARGB(255, 0, 0, 0),
                            ),
                        ],
                    ),
                ),
                const SizedBox(height: 20),
                Card(
                    elevation: 5,
                    shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(15),
                    ),
                    child: Padding(
                        padding: const EdgeInsets.all(20.0),
                        child: Column(
                            mainAxisAlignment: MainAxisAlignment.start,
                            children: [
                                _buildFeatureSection(
                                    'Secure File Sharing Made Simple',
                                    'Share your files securely with password protection. Generate unique links for your files and share them with confidence.',
                                ),
                                const SizedBox(height: 30),
                                _buildFeatureSection(
                                    'Custom Aliases',
                                    'Create memorable links by using custom aliases for your shared files. Make your links personal and easy to remember.',
                                ),
                                const SizedBox(height: 30),
                                _buildFeatureSection(
                                    'Password Protection',
                                    'Keep your shared files private with password protection. Only users with the correct password can access your files.',
                                ),
                                const SizedBox(height: 30),
                                _buildFeatureSection(

```

```

        'Easy Download',
        'Recipients can easily download your files by simply entering
the password. No account required!',
        ),],),),),
const SizedBox(height: 30),
Card(
    elevation: 5,
    shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15),
    ),
    child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: const [
                Text(
                    'How It Works',
                    style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        color: Colors.lightBlue,
                    ),
                ),
            ],
            SizedBox(height: 15),
            Text(
                '1. Select any file you want to share\n'
                '2. Choose a custom alias for your link\n'
                '3. Set a secure password\n'
                '4. Share the generated link with others\n'
                '5. Recipients enter the password to download',
                style: TextStyle(
                    fontSize: 16,
                    height: 1.8,
                    color: Colors.black87,
                ),
            ),
        ],
        const SizedBox(height: 40),
    ],
),
),
);
}

Widget _buildFeatureSection(String title, String description,) {
return Row(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
        const SizedBox(width: 15),
        Expanded(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    Text(
                        title,
                        style: const TextStyle(
                            fontSize: 18,
                            fontWeight: FontWeight.bold,
                            color: Colors.lightBlue,
                        ),
                    ),
                ],
            ),
        ),
    ],
),
);
}

```

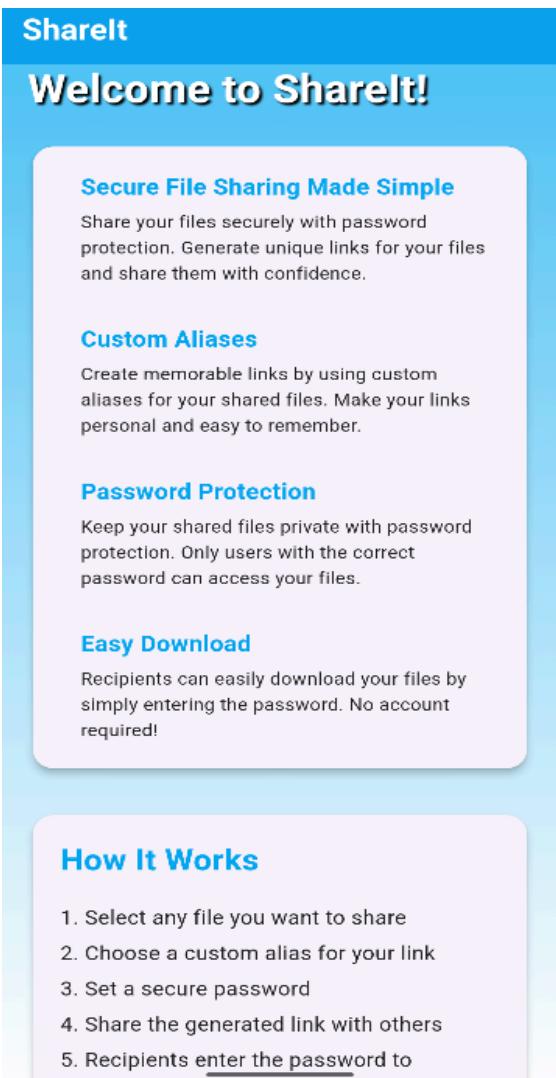
```

Widget _buildFeatureSection(String title, String description,) {
return Row(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
        const SizedBox(width: 15),
        Expanded(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    Text(
                        title,
                        style: const TextStyle(
                            fontSize: 18,
                            fontWeight: FontWeight.bold,
                            color: Colors.lightBlue,
                        ),
                    ),
                ],
            ),
        ),
    ],
),
);
}

```

```
const SizedBox(height: 5),  
Text(  
  description,  
  style: const TextStyle(  
    fontSize: 14,  
    color: Colors.black87,  
    height: 1.5,  
)},],),],);}}
```

Output:



Conclusion:

In conclusion, the experiment was successfully completed by creating a simple homepage in Flutter, which allowed for hands-on exploration of various widgets. The process involved utilizing basic widgets like Text, Container, Column to design a functional UI. This exercise provided valuable insights into how widgets are structured and combined to build layouts in Flutter. By experimenting with different widgets, the understanding of Flutter's declarative UI framework was enhanced, laying the foundation for more complex app development in the future.

Name: Prathamesh Shetty
Class: D15B
Roll No: 53

MPL Experiment 3

Code:

Icons:

```
children: [
  _buildFeatureSection(
    'Secure File Sharing Made Simple',
    'Share your files securely with password protection. Generate unique links for
    your files and share them with confidence.',
    Icons.security,
  ),
  const SizedBox(height: 30),
  _buildFeatureSection(
    'Custom Aliases',
    'Create memorable links by using custom aliases for your shared files. Make
    your links personal and easy to remember.',
    Icons.link,
  ),
  const SizedBox(height: 30),
  _buildFeatureSection(
    'Password Protection',
    'Keep your shared files private with password protection. Only users with the
    correct password can access your files.',
    Icons.lock,
  ),
  const SizedBox(height: 30),
  _buildFeatureSection(
    'Easy Download',
    'Recipients can easily download your files by simply entering the password. No
    account required!',
    Icons.download,
  ),
],
```



Secure File Sharing Made Simple

Share your files securely with password protection. Generate unique links for your files and share them with confidence.



Custom Aliases

Create memorable links by using custom aliases for your shared files. Make your links personal and easy to remember.



Password Protection

Keep your shared files private with password protection. Only users with the correct password can access your files.



Easy Download

Recipients can easily download your files by simply entering the password. No account required!

Images:

title: Row(

children: [

Image.asset(

'lib/assets/img.png', // Path to your image

height: 32, // Adjust height as needed

width: 32, // Adjust width as needed

),

const SizedBox(width: 10), // Add spacing between image and text

const Text(

'ShareIt',

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

],

),

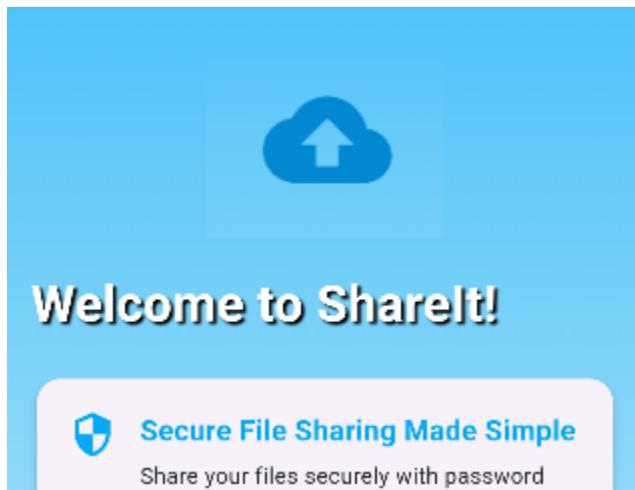
9:43 9:43



ShareIt

Center(

```
    child: Image.asset(  
        'lib/assets/UploadImage.png', // Path to your image  
        height: 150, // Adjust height as needed  
        width: 150, // Adjust width as needed  
    ),  
,
```



Fonts:

1. Add google fonts to dependencies in pubspec.yaml file and run the flutter pub get command in terminal

dependencies:

flutter:

 sdk: flutter

 google_fonts: ^6.1.0

Terminal: flutter pub get

2. Add import statement in the file to use the fonts

```
import 'package:google_fonts/google_fonts.dart';
```

3. Use the fonts in the textTheme field

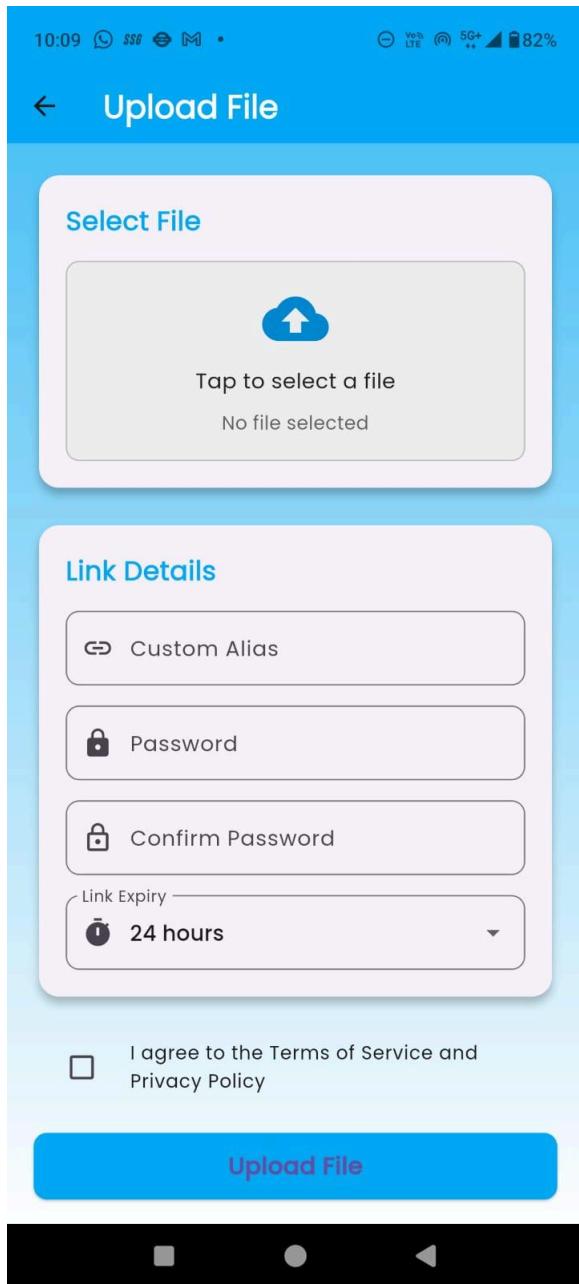
theme: ThemeData(

```
  primarySwatch: Colors.lightBlue,  
  scaffoldBackgroundColor: Colors.white,  
  textTheme: GoogleFonts.poppinsTextTheme(  
    Theme.of(context).textTheme,  
,
```



Conclusion:

In this experiment, we successfully implemented **icons, images, and fonts** in a Flutter application. We integrated icons using the `Icons` class, added images using the `Image.asset` widgets, and customized text appearance with different fonts using the `TextStyle` class. This experiment helped in understanding how to enhance the UI by incorporating visual elements and typography, improving the overall aesthetics and user experience of the app.



Conclusion: In this experiment, we successfully created an interactive form in Flutter using various widgets. By leveraging TextFields, Dropdown Buttons, Checkboxes, RadioButtons, and Buttons, we provided a dynamic and user-friendly experience for data input. Additionally, we implemented form validation to ensure data accuracy and enhance usability.

Navigation:

```
ListTile(  
  leading: const Icon(Icons.upload_file),  
  title: const Text('My Uploads'),  
  onTap: () {  
    Navigator.pop(context);  
    Navigator.push(  
      context,  
      MaterialPageRoute(builder: (context) => const UploadPage()),  
    );  
  },  
,  
  
onTap: () {  
  Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => const UploadPage()),  
  );  
},
```

Routing:

```
void main() => runApp(MaterialApp(  
  initialRoute: '/',  
  routes: {  
    '/': (context) => FirstPage(),  
    '/second': (context) => SecondPage(),  
  },  
));
```

Gestures:

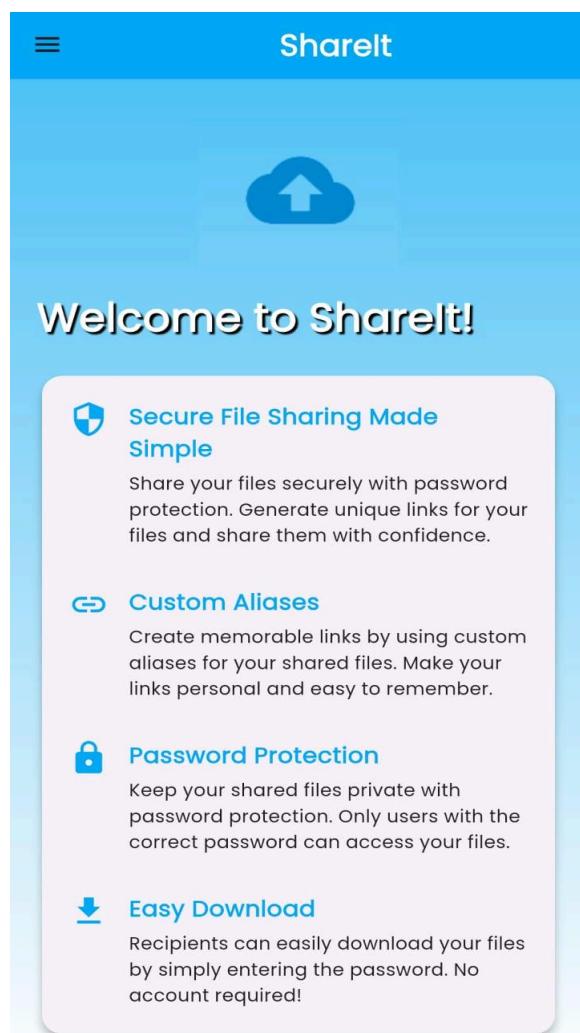
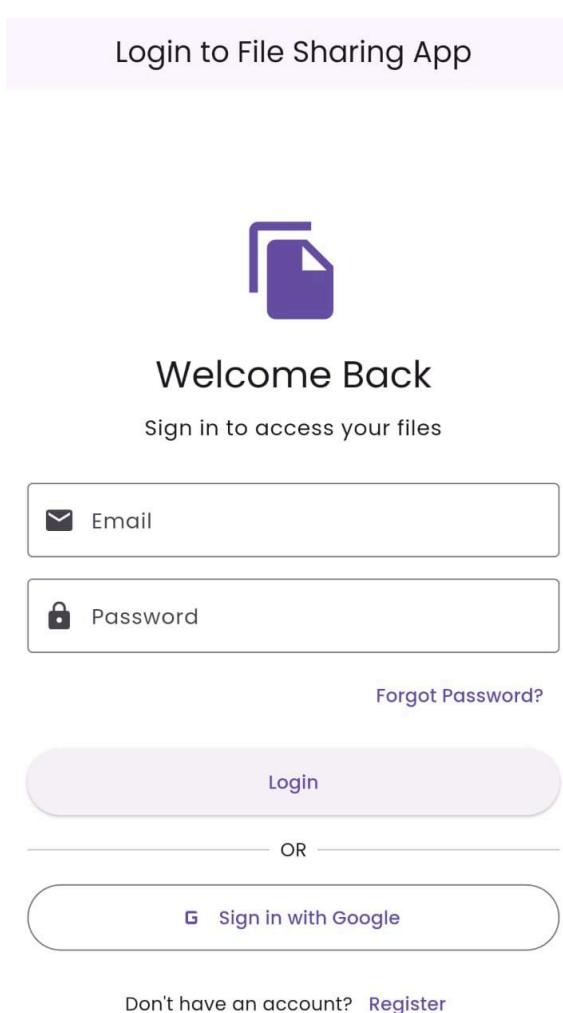
```
GestureDetector(  
  onTap: () {  
    print("Tapped!");  
  },
```

```

onDoubleTap: () {
    print("Double Tapped!");
},
onLongPress: () {
    print("Long Pressed!");
},
child: Container(
    color: Colors.blue,
    padding: EdgeInsets.all(30),
    child: Text("Tap Me", style: TextStyle(color: Colors.white)),
),
)
)

```

Implementation:



The screenshot shows a mobile application interface. At the top left is a user profile section with a blue circular icon containing a white letter 'N', the name 'NT6 Prathamesh Shetty', and the email 'nt6prathameshshetty@gmail.com'. Below this is a navigation menu with the following items: Home, My Uploads, Settings, Help & Support, and Sign Out. The main content area is titled 'Upload File' with a back arrow. It contains a 'Select File' section with a cloud icon and the text 'Tap to select a file' and 'No file selected'. Below this is a 'Link Details' section with fields for 'Custom Alias' (with a link icon), 'Password' (with a lock icon), 'Confirm Password' (with a lock icon), and 'Link Expiry' (set to '24 hours'). At the bottom is a checkbox labeled 'I agree to the Terms of Service and Privacy Policy' and a large blue 'Upload File' button.

NT6 Prathamesh Shetty
nt6prathameshshetty@gmail.com

Home
My Uploads
Settings
Help & Support
Sign Out

Upload File

Select File

Tap to select a file
No file selected

Link Details

Custom Alias

Password

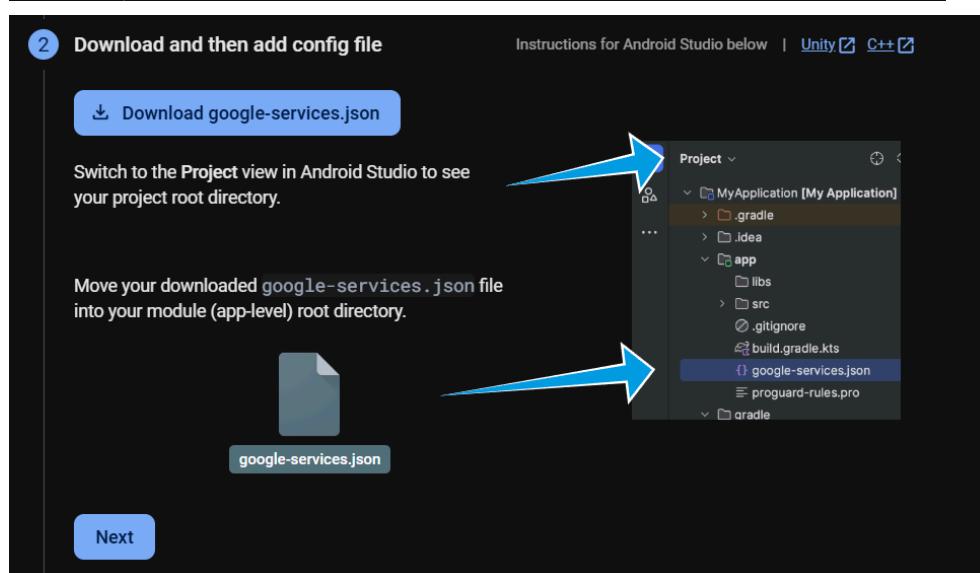
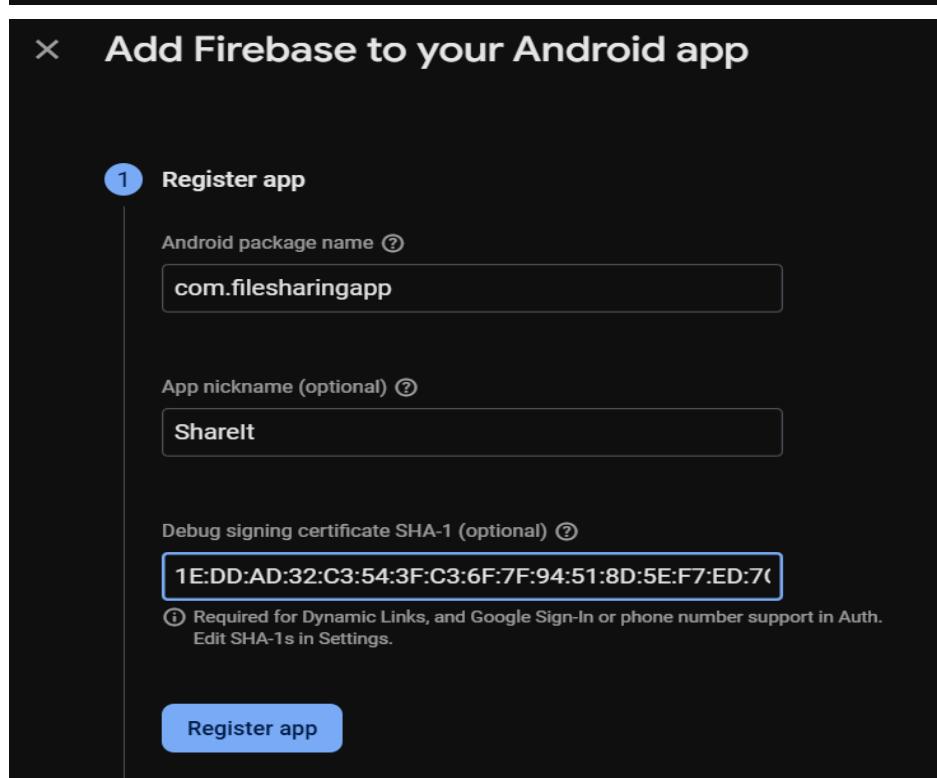
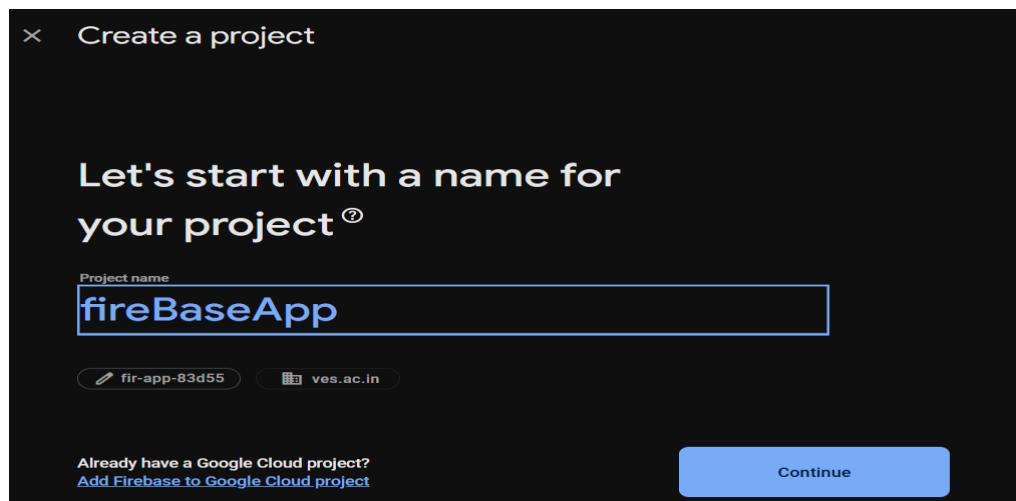
Confirm Password

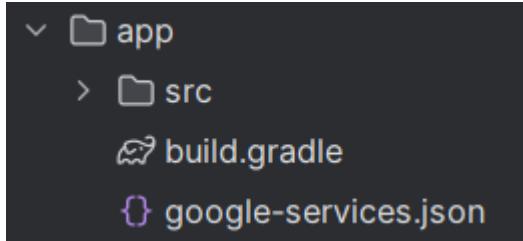
Link Expiry
24 hours

I agree to the Terms of Service and Privacy Policy

Upload File

Installation:





3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

★ Are you still using the `buildscript` syntax to manage plugins? Learn how to [add Firebase plugins](#) using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

Kotlin DSL (`build.gradle.kts`) Groovy (`build.gradle`)

Add the plugin as a dependency to your **project-level** `build.gradle.kts` file:

Root-level (project-level) Gradle file (`<project>/build.gradle.kts`):

```
plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.4.2" apply false
}
```

2. Then, in your **module (app-level)** `build.gradle.kts` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (`<project>/<app-module>/build.gradle.kts`):

```
plugins {
    id("com.android.application")
    // Add the Google services Gradle plugin
    id("com.google.gms.google-services")
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:33.12.0")) // Import the BoM

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation("com.google.firebase:firebase-analytics") // Add the desired Firebase products

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

[Previous](#)

[Next](#)

```
plugins {
    id 'com.google.gms.google-services' version '4.4.2' apply false
}

dependencies{
    implementation(platform("com.google.firebase:firebase-bom:33.10.0"))
    implementation("com.google.firebase:firebase-analytics")
}

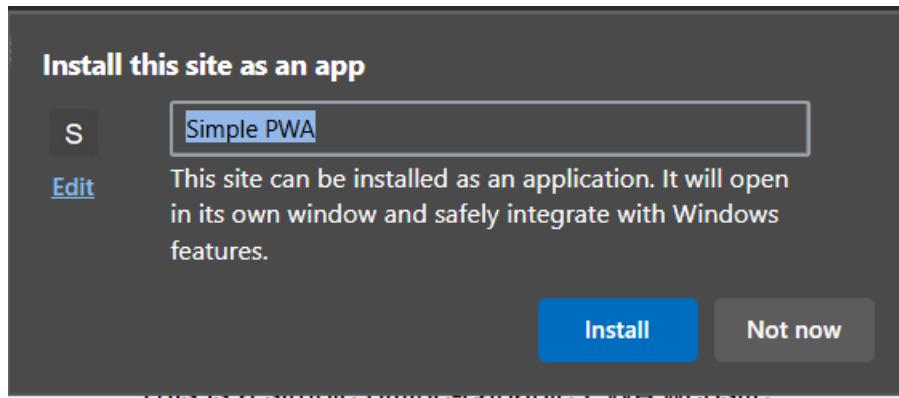
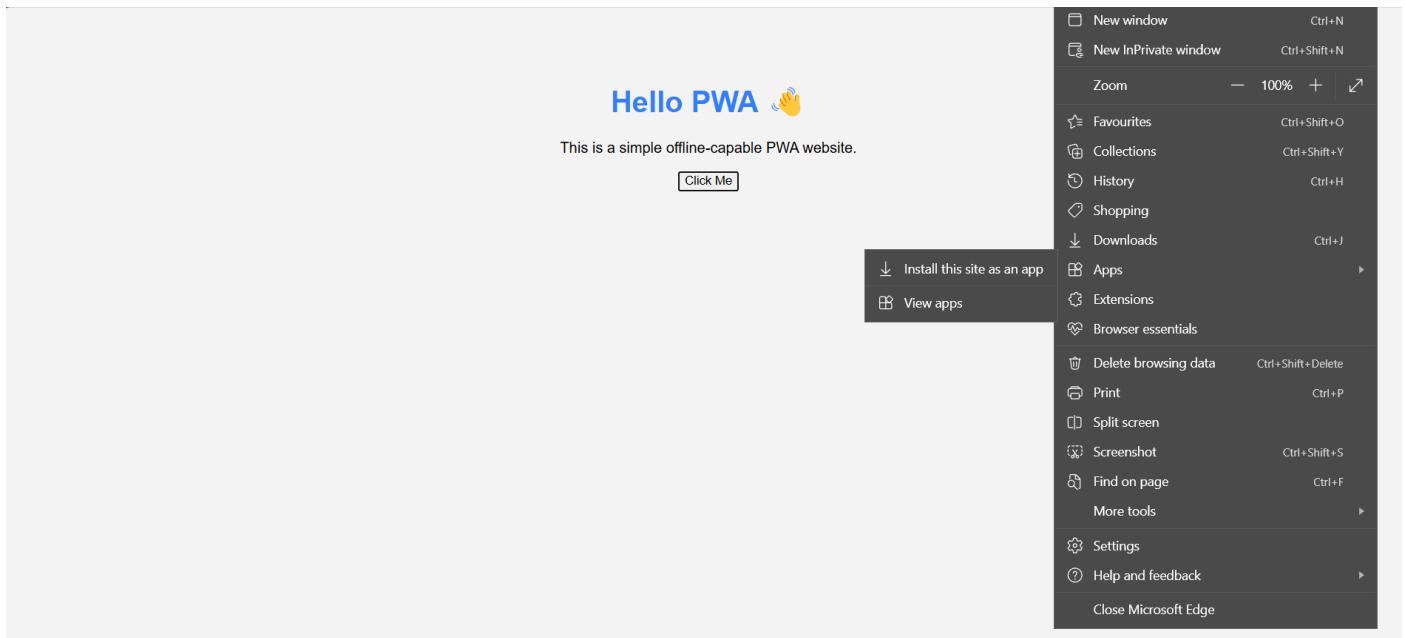
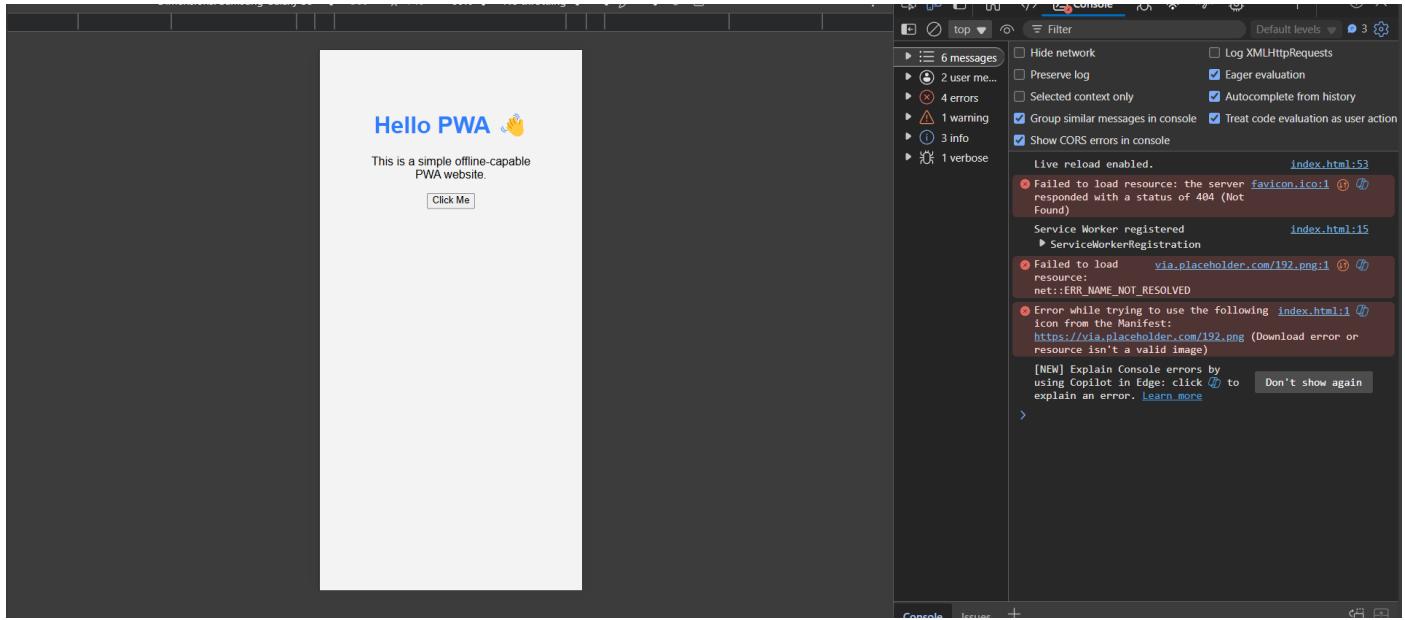
plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin
    id "dev.flutter.flutter-gradle-plugin"
    id 'com.google.gms.google-services'

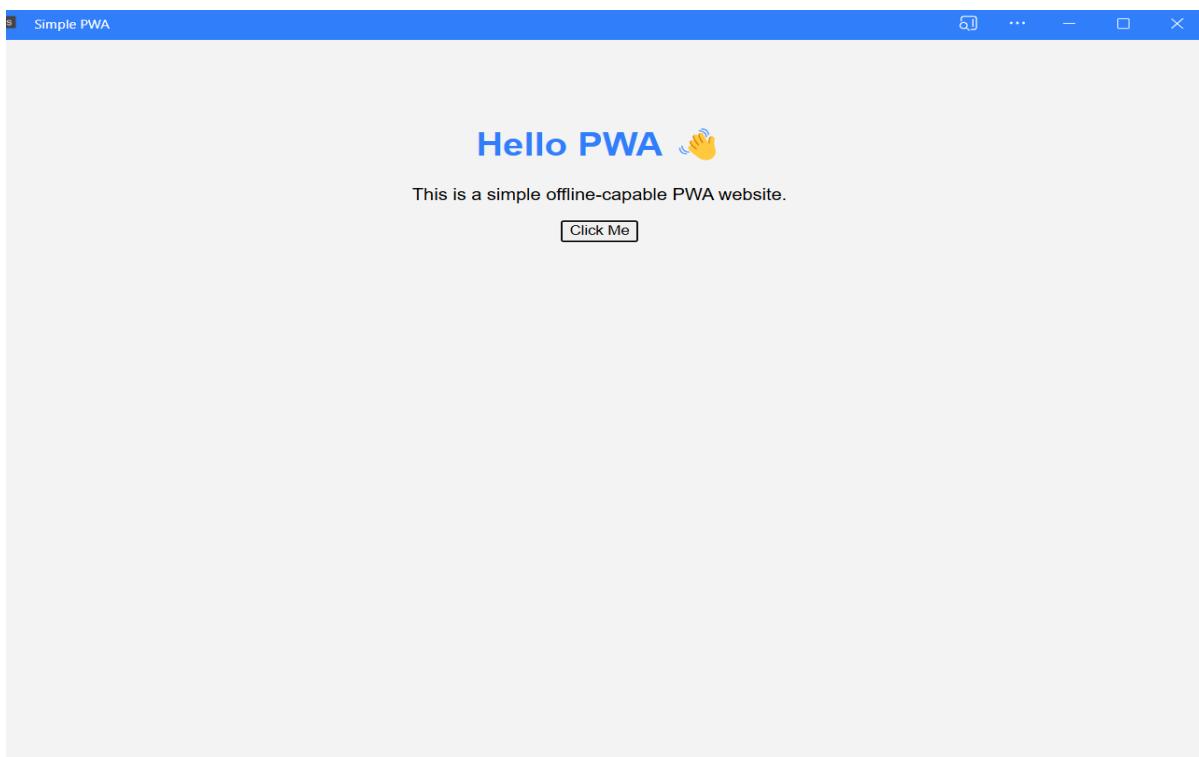
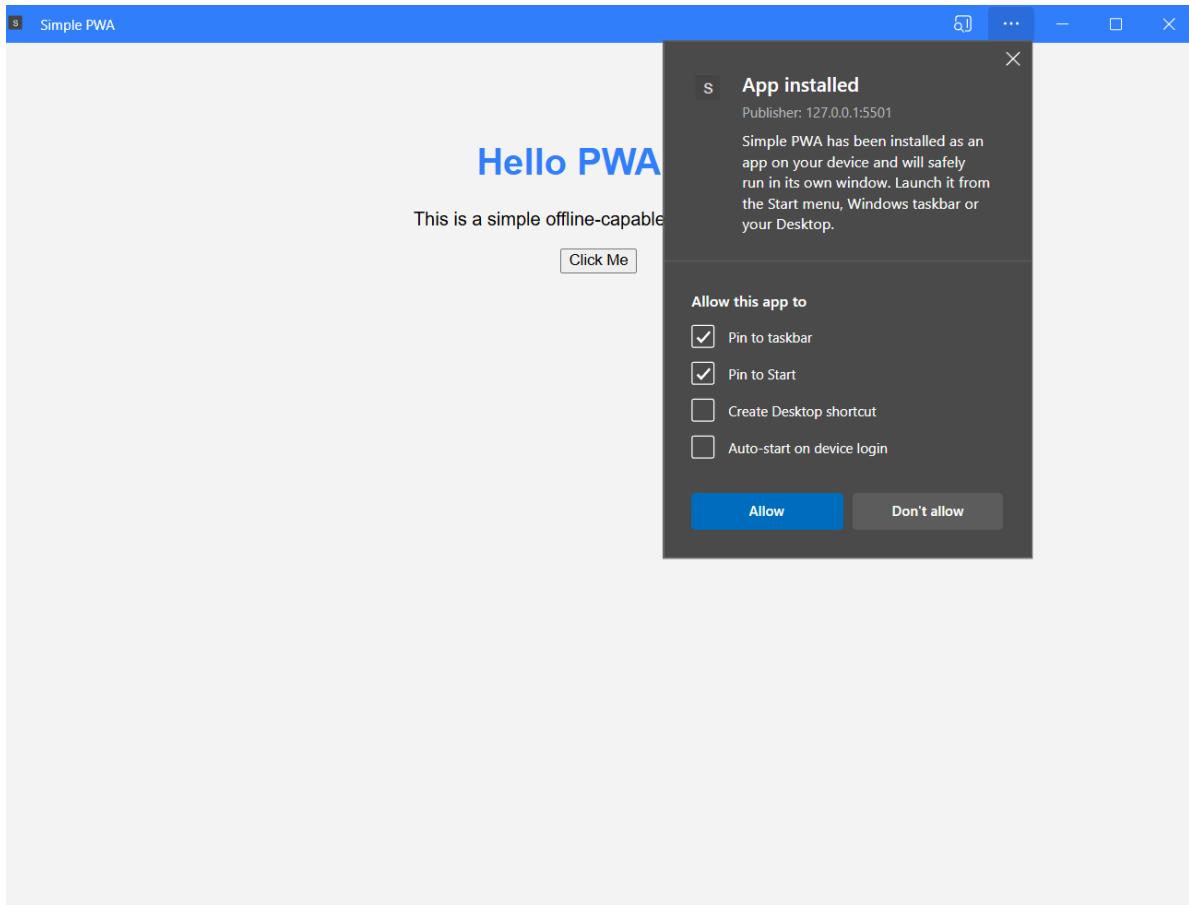
}
```

Conclusion:

By following the above steps, Firebase setup for both Android and iOS platforms in a Flutter project is completed. This allows the app to use powerful backend services like authentication, real-time database, cloud messaging, and more, with minimal configuration. Firebase simplifies backend development and helps in building scalable and secure applications efficiently.

Experiment No 7





Experiment N0 8

Code:

```
// PWA functionality
let deferredPrompt;
const installBanner = document.getElementById('install-banner');
const installButton = document.getElementById('install-button');
const dismissButton = document.getElementById('dismiss-button');

// Check if service worker is supported
if ('serviceWorker' in navigator) {
    window.addEventListener('load', () => {
        navigator.serviceWorker.register('service-worker.js')
            .then(registration => {
                console.log('Service Worker registered with scope:', registration.scope);
            })
            .catch(error => {
                console.error('Service Worker registration failed:', error);
            });
    });
}

// Listen for the beforeinstallprompt event
window.addEventListener('beforeinstallprompt', (e) => {
    // Prevent Chrome 67 and earlier from automatically showing the prompt
    e.preventDefault();

    // Stash the event so it can be triggered later
    deferredPrompt = e;

    // Show the install banner
    installBanner.classList.remove('hidden');
});

// Install button click handler
installButton.addEventListener('click', () => {
    // Hide the install banner
    installBanner.classList.add('hidden');

    // Show the install prompt
    if (deferredPrompt) {
        deferredPrompt.prompt();
    }

    // Wait for the user to respond to the prompt
    deferredPrompt.userChoice.then((choiceResult) => {
        if (choiceResult.outcome === 'accepted') {
```

```
        console.log('User accepted the install prompt');
    } else {
        console.log('User dismissed the install prompt');
    }

    // Clear the deferredPrompt variable
    deferredPrompt = null;
});
}
});

// Dismiss button click handler
dismissButton.addEventListener('click', () => {
    installBanner.classList.add('hidden');
});

// E-commerce functionality
const cartCount = document.querySelector('.cart-count');
const addToCartButtons = document.querySelectorAll('.add-to-cart');

let cartItems = 0;

// Add to cart functionality
addToCartButtons.forEach(button => {
    button.addEventListener('click', () => {
        cartItems++;
        cartCount.textContent = cartItems;

        // Animate button
        button.textContent = 'Added!';
        button.style.backgroundColor = '#4caf50';

        setTimeout(() => {
            button.textContent = 'Add to Cart';
            button.style.backgroundColor = '#4285f4';
        }, 1000);
    });
});

// Service Worker installation and activation
self.addEventListener('install', (event) => {
    console.log('Service Worker installing.');
    // Perform install steps
    event.waitUntil(
        caches.open('ecommerce-cache-v1')
        .then((cache) => {
```

```
        console.log('Opened cache');
        return cache.addAll([
            '/',
            '/index.html',
            '/styles.css',
            '/script.js',
            '/images/logo.png',
            '/images/shop_icon.png',
            // Add other assets to cache
        ]);
    })
);
};

self.addEventListener('activate', (event) => {
    console.log('Service Worker activating.');
    const cacheWhitelist = ['ecommerce-cache-v1'];
    event.waitUntil(
        caches.keys().then((cacheNames) => {
            return Promise.all(
                cacheNames.map((cacheName) => {
                    if (cacheWhitelist.indexOf(cacheName) === -1) {
                        return caches.delete(cacheName);
                    }
                })
            );
        })
    );
});

self.addEventListener('fetch', (event) => {
    event.respondWith(
        caches.match(event.request)
            .then((response) => {
                // Cache hit - return response
                if (response) {
                    return response;
                }
                return fetch(event.request);
            })
    );
});
});
```

Service workers

Offline Update on reload Bypass for network

<http://127.0.0.1:5500/PWA/>

Source [service-worker.js](#)

Received 3/20/2025, 7:41:14 PM

Status ● #187 activated and is running [Stop](#)

Push Test push message from DevTools. [Push](#)

Sync test-tag-from-devtools [Sync](#)

Periodic sync test-tag-from-devtools [Periodic sync](#)

Update Cycle Version Update Activity Timeline

▶ #187 Install

▶ #187 Wait

▶ #187 Activate

Featured Products



Wireless Headphones

\$79.99

Add to Cart

Fetching: <chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/js.js>

⚠ Service Worker was updated because "Update on reload" was checked in the DevTools Ap

✓ Service Worker Installed

✓ Caching assets

✓ Service Worker Activated

Fetching: <http://127.0.0.1:5500/PWA/index.html>

Fetching: <http://127.0.0.1:5500/PWA/styles.css>

Fetching: <https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-4.0.0>

Fetching: <https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.0>

Fetching: <https://images.unsplash.com/photo-1579586337278-3befd40fd17a?ixlib=rb-4.0.0>

Fetching: <https://images.unsplash.com/photo-1608043152269-423dbba4e7e1?ixlib=rb-4.0.0>

Fetching: <http://127.0.0.1:5500/PWA/app.js>

Navigated to <http://127.0.0.1:5500/PWA/index.html>

Fetching: <http://127.0.0.1:5500/PWA/manifest.json>

✓ Service Worker registered successfully: ▶ ServiceWorkerRegistration {installing:

Fetching: http://127.0.0.1:5500/PWA/images/shop_icon144.png

Fetching: <http://127.0.0.1:5500/PWA/images/hero-bg.jpg>

Fetching: <chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/js.js>

Fetching: <chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/js/dom.js>

```
✓ Service Worker Installed
✓ Caching assets
Fetching: chrome-extension://gppongmhjkpfnbhagpmjfkanfbllamg/js/js.js
✓ Service Worker Activated
Fetching: http://127.0.0.1:5500/PWA/index.html
Fetching: http://127.0.0.1:5500/PWA/styles.css
Fetching: https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-4.0.
Fetching: https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.3&a
Navigated to http://127.0.0.1:5500/PWA/index.html
Fetching: https://images.unsplash.com/photo-1579586337278-3befd40fd17a?ixlib=rb-4.0.
Fetching: https://images.unsplash.com/photo-1608043152269-423dbba4e7e1?ixlib=rb-4.0.
Fetching: http://127.0.0.1:5500/PWA/app.js
Fetching: http://127.0.0.1:5500/PWA/manifest.json
```

X	Headers	Payload	Preview	Response	Initiator	Timing	Adblock
▼ General							
Request URL:	https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.3&auto=format&q=80						
Request Method:	GET						
Status Code:	200 OK (from service worker)						
Referrer Policy:	strict-origin-when-cross-origin						
▼ Response Headers							
Accept-Ranges:	bytes						
Access-Control-Allow-Origin:	*						
Age:	2196289						
Cache-Control:	public, max-age=31536000						
Content-Length:	22112						
Content-Type:	image/avif						
Cross-Origin-Resource-Policy:	cross-origin						
Date:	Thu, 20 Mar 2025 13:13:33 GMT						
Last-Modified:	Sun, 23 Feb 2025 03:08:44 GMT						
Server:	imgix						
Timing-Allow-Origin:	*						
Vary:	Accept, User-Agent						
X-Cache:	HIT, HIT, HIT						
X-Content-Type-Options:	nosniff						
X-Imgix-Id:	18f9bb0d88bd11511c6c61c3e37b1008b5606d4e						
X-Served-By:	cache-chi-kigq8000041-CHI, cache-tyo11976-TYO, cache-bom4739-B						

C X Y Filter by path

http://127.0.0.1:5500	
Origin	http://127.0.0.1:5500
Bucket name	default
Is persistent	No
Durability	relaxed
Quota	0 B
Expiration	None
#	Name
0	/PWA/
1	/PWA/app.js
2	/PWA/index.html
3	/PWA/styles.css

Experiment No 9

Code:

```
const CACHE_NAME = "pwa-cache-v1";
const ASSETS_TO_CACHE = [
    "/PWA/",
    "/PWA/index.html",
    "/PWA/styles.css",
    "/PWA/app.js",
    // Add other assets to cache
];

// Install event - Cache important assets
self.addEventListener("install", (event) => {
    console.log("✓ Service Worker Installed");
    event.waitUntil(
        caches.open(CACHE_NAME).then((cache) => {
            console.log("✓ Caching assets");
            return cache.addAll(ASSETS_TO_CACHE);
        })
    );
});

// Activate event - Cleanup old caches
self.addEventListener("activate", (event) => {
    console.log("✓ Service Worker Activated");
    event.waitUntil(
        caches.keys().then((cacheNames) =>
            Promise.all(
                cacheNames.map((cache) => {
                    if (cache !== CACHE_NAME) {
                        console.log("🗑 Deleting old cache:", cache);
                        return caches.delete(cache);
                    }
                })
            )
        )
    );
});

// Fetch event - Serve cached files when offline
self.addEventListener("fetch", (event) => {
    console.log("Fetching:", event.request.url);
    event.respondWith(
        caches
```

```
.match(event.request)
    .then((response) => response || fetch(event.request))
    .catch(() => caches.match("/PWA/index.html")) // Fallback to
index.html if offline
);
});

// Sync event - Background sync for failed requests
self.addEventListener("sync", (event) => {
  if (event.tag === "sync-cart") {
    event.waitUntil(syncCart());
  }
});

async function syncCart() {
  // Logic to sync cart data with the server
  console.log("⌚ Syncing cart data with the server");
  // Example: Fetch unsynced cart data from IndexedDB and send to
server
}

// Push event - Handle push notifications
self.addEventListener("push", (event) => {
  const data = event.data.json();
  console.log("🔔 Push notification received:", data);

  const options = {
    body: data.body,
    icon: "/PWA/icons/icon-192x192.png",
    badge: "/PWA/icons/icon-192x192.png",
  };

  event.waitUntil(
    self.registration.showNotification(data.title, options)
  );
});
```

<input checked="" type="checkbox"/>	Service Worker Installed
<input checked="" type="checkbox"/>	Caching assets
<input checked="" type="checkbox"/>	Service Worker Activated
<input checked="" type="checkbox"/>	Fetching: http://127.0.0.1:5500/PWA/index.html
<input checked="" type="checkbox"/>	Fetching: http://127.0.0.1:5500/PWA/styles.css

Featured Products



Wireless Headphones
\$79.99

Add to Cart

Fetchings:

- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/js.js>
- ⚠️ Service Worker was updated because "Update on reload" was checked in the DevTools Application panel.
- ✓ Service Worker Installed
- ✓ Caching assets
- ✓ Service Worker Activated
- Fetching: <http://127.0.0.1:5500/PWA/index.html>
- Fetching: <http://127.0.0.1:5500/PWA/styles.css>
- Fetching: <https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80>
- Fetching: <https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80>
- Fetching: <https://images.unsplash.com/photo-1579586337278-3befd40fd17a?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80>
- Fetching: <https://images.unsplash.com/photo-1608043152269-423dbba4e7e1?ixlib=rb-4.0.3&auto=format&fit=crop&w=500&q=80>
- Fetching: <http://127.0.0.1:5500/PWA/app.js>

Navigated to <http://127.0.0.1:5500/PWA/index.html>

Fetchings:

- Fetching: http://127.0.0.1:5500/PWA/images/shop_icon144.png
- Fetching: <http://127.0.0.1:5500/PWA/images/hero-bg.jpg>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/js.js>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/dom.js>

Featured Products



Fetchings:

- Group similar messages in console
- Show CORS errors in console

⚠️ NOTIFICATION PERMISSION DENIED. PLEASE ENABLE NOTIFICATIONS IN YOUR BROWSER SETTINGS.

Fetchings:

- Fetching: http://127.0.0.1:5500/PWA/images/shop_icon144.png
- Fetching: <http://127.0.0.1:5500/PWA/images/hero-bg.jpg>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/js.js>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/dom.js>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/js.js>
- Fetching: <chrome-extension://gppongmhjkpfnbhagomjfkannfbllamg/js/dom.js>

⚠️ Push notification received: {method: 'pushMessage', message: 'Hello!'}

<http://127.0.0.1:5500/PWA/>

Source [service-worker.js](#)  12

Received 3/20/2025, 8:09:35 PM

Status  #211 activated and is running [Stop](#)

Push `{ "method": "pushMessage", "message": "Hello!" }` [Push](#)

Sync `test-tag-from-devtools` [Sync](#)

Periodic sync `test-tag-from-devtools` [Periodic sync](#)

 Push notification received: ➔ `{method: 'pushMessage', message: 'Hello!'}`

Experiment N0 10

The screenshot shows two main sections of a GitHub interface.

Repository Overview: At the top, there's a header with a user icon, the repository name "pwa" (marked as public), and buttons for "Pin" and "Unwatch". Below this, it shows "main" branch, 1 branch, 0 tags, a search bar ("Go to file"), and buttons for "Add file" and "Code".

File List: A table lists the repository's files:

File	Description	Last Commit
images	Initial commit	8 hours ago
README.md	Initial commit	8 hours ago
app.js	Initial commit	8 hours ago
index.html	Initial commit	8 hours ago
manifest.json	Initial commit	8 hours ago
service-worker.js	Initial commit	8 hours ago
styles.css	Initial commit	8 hours ago

GitHub Pages Settings: This section is under the "Settings" tab. It includes:

- General:** Options for Access, Collaborators, and Moderation options.
- Build and deployment:** Set to "Deploy from a branch".
 - Branch:** Set to "main".
 - Source:** Set to "/ (root)".
- Custom domain:** Allows serving from a custom domain.
- Security:** Includes "Code Security" and "Deploy keys".

<https://shetty1301.github.io/pwa/>

https://shetty1301.github.io/pwa/

Google Lens

ShopEasy

Home Products Categories Deals Contact

Shop Smart, Shop Easy

Discover amazing products at incredible prices

Shop Now

Featured Products



Experiment No 11

PageSpeed Insights [Docs](#)

Enter a web page URL [Analyze](#)

Make your web pages fast on all devices

CHECK OUT
What's new
Documentation
Learn about Web Performance



This site uses cookies from Google to deliver its services and to analyze traffic. [Learn More.](#) [Ok, Got it.](#)

Report from Apr 11, 2025, 9:48:34 AM

<https://shetty1301.github.io/pwa/> [Analyze](#)

[!\[\]\(0ae8361765f665634e41f95a70e8b74b_img.jpg\) Mobile](#) [!\[\]\(31da6726007de2d76ed0eaa1ac7eb898_img.jpg\) Desktop](#)

[!\[\]\(1cbc4415309750e22733af676d9c3f09_img.jpg\) Discover what your real users are experiencing](#) [!\[\]\(6b8994a45ef6ddc913862c771c6dc730_img.jpg\) No Data](#)

Diagnose performance issues

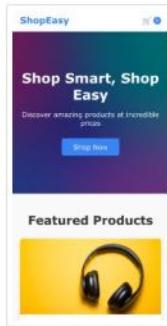
 97	 93	 96	 100
Performance	Accessibility	Best Practices	SEO



Performance

Values are estimated and may vary. The [performance score](#) is calculated directly from these metrics. See calculator.

 0–49  50–89  90–100



METRICS [Expand view](#)

100
Performance

93
Accessibility

96
Best Practices

100
SEO

100
Performance

Values are estimated and may vary. The [performance score](#) is calculated directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100

METRICS

- First Contentful Paint
0.2 s
- Total Blocking Time
0 ms
- Speed Index
0.7 s

Captured at Apr 11, 2025, 9:48 AM GMT+5:30

Initial page load

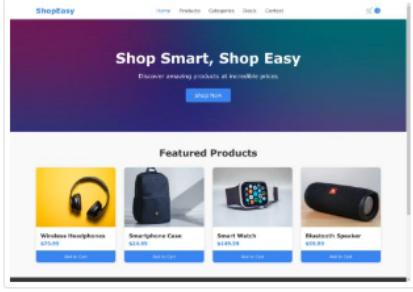
Emulated Desktop with Lighthouse 12.4.0

Custom throttling

Single page session

Using HeadlessChromium 134.0.6998.165 with Ir

Expand view



DIAGNOSTICS

- ▲ Minimize main-thread work — 3.3 s
- ▲ Serve images in next-gen formats — Potential savings of 80 KiB
- ▲ Largest Contentful Paint element — 2,740 ms
- Preload Largest Contentful Paint image
- Minify JavaScript — Potential savings of 4 KiB
- Serve static assets with an efficient cache policy — 3 resources found
- Properly size images — Potential savings of 8 KiB
- Reduce unused JavaScript — Potential savings of 97 KiB
- Avoid long main-thread tasks — 7 long tasks found
- Avoids an excessive DOM size — 74 elements
- JavaScript execution time — 0.6 s
- User Timing marks and measures — 1 user timing
- Initial server response time was short — Root document took 40 ms
- Avoids enormous network payloads — Total size was 245 KiB

96

Best Practices

TRUST AND SAFETY

- ▲ Requests the notification permission on page load
- Ensure CSP is effective against XSS attacks
- Use a strong HSTS policy
- Ensure proper origin isolation with COOP

PASSED AUDITS (14)

Show

NOT APPLICABLE (2)

Show

93

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

CONTRAST

- ▲ Background and foreground colors do not have a sufficient contrast ratio. ▾

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (13)

Show

NOT APPLICABLE (43)

Show

100

SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

- Structured data is valid ▾

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8)

Show

NOT APPLICABLE (2)

Show

📅 Captured at Mar 20, 2025, 8:40

PM GMT+5:30

⌚ Initial page load

💻 Emulated Moto G Power with

Lighthouse 12.3.0

⚡ Slow 4G throttling

👤 Single page session

🌐 Using Chromium 134.0.0.0 with devtools