

Name:Tarunkumar Sharma
Div:D15B
Roll no:52

EXP 1: Installation and Configuration of Flutter Environment.

Aim: Installation and Configuration of Flutter Environment.

Theory: Flutter is an open-source framework developed by Google that is primarily used for building natively compiled applications for mobile, web, and desktop from a single codebase. It is particularly popular for creating mobile apps for iOS and Android, but it can also be used to develop apps for macOS, Windows, Linux, and the web.

Key Features of Flutter:

1. **Cross-Platform Development:** With Flutter, developers can write one codebase and deploy it across multiple platforms (mobile, web, desktop). This greatly reduces the development effort and cost compared to maintaining separate codebases for different platforms.
2. **Hot Reload:** Flutter's "hot reload" feature allows developers to see code changes immediately reflected in the app without restarting it. This makes development faster and smoother.
3. **Widgets:** Flutter is built around a rich set of customizable widgets. These widgets describe how the app should look and behave, and they can be combined and customized to create complex UIs.
4. **Dart Programming Language:** Flutter uses Dart, a language developed by Google, which is object-oriented and compiled to native code. Dart is easy to learn and has strong support for asynchronous programming, which is crucial for building modern apps.

5. Native Performance: Since Flutter is compiled directly into native ARM machine code, it can provide near-native performance on both iOS and Android. This is one of the main advantages over other cross-platform frameworks like React Native.
6. Customizable UI: With Flutter, developers have complete control over the design and the user interface (UI). It doesn't rely on the platform's native components; instead, it provides its own set of widgets, which means you can create custom UIs that maintain consistency across platforms.

Install the Flutter SDK

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.

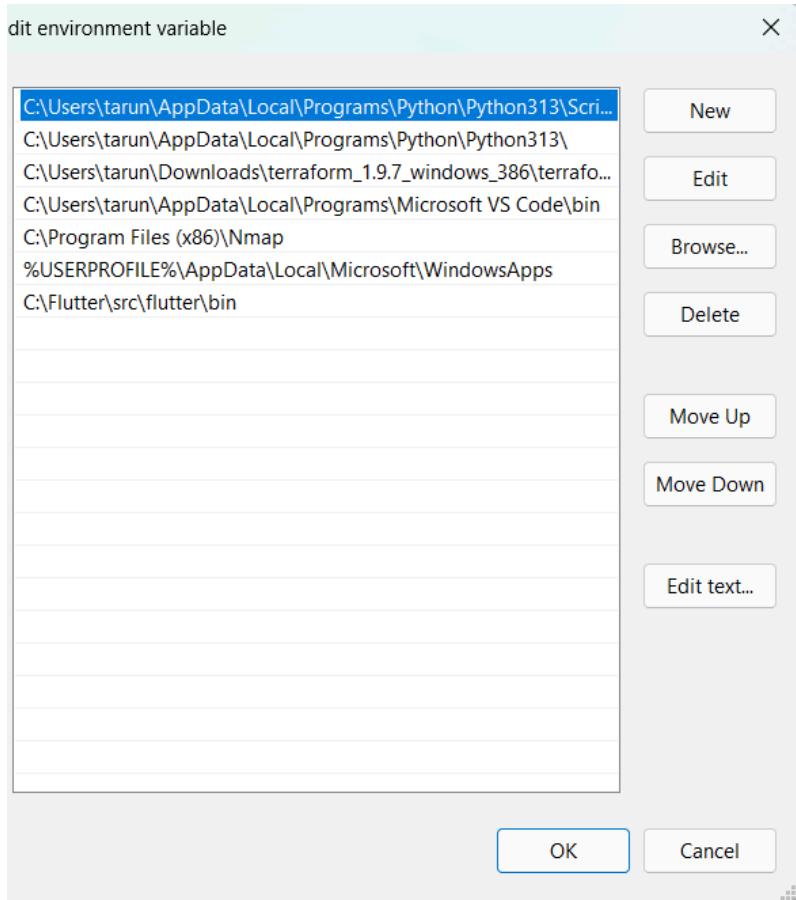
The screenshot shows the Flutter Docs website with the URL <https://docs.flutter.dev/get-started/install>. The page has a navigation bar with links for Homepage, Community, Packages, API reference, a search icon, and a 'Get started' button. On the left, there's a sidebar with sections like 'Get started' (which is active), 'Set up Flutter' (highlighted in blue), 'Learn Flutter', 'Stay up to date', 'App solutions', 'User interface' (with sub-links for Introduction, Widget catalog, Layout, Adaptive & responsive design, Design & theming, Interactivity, and Assets & media), and a note about 'Developing in China'. The main content area features the heading 'Choose your development platform to get started' with four options: Windows (Current device), macOS, Linux, and ChromeOS, each with its respective logo.

Step 2 : Download Flutter SDK: Go to the Flutter website and click on the Windows icon to download the latest SDK.

Step 3 : Extract the Zip: Once the download is complete, extract the zip file and place it in a location, like `C:\Flutter`.

Step 4 : Update System Path:

- Right-click **This PC** → **Properties** → **Advanced system settings** → **Environment Variables**.
- In the **System Variables** section, find the **Path** variable, click **Edit**, and add **C:\flutter\bin** to the list.



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.

```
Command Prompt - flutter doctor + ▾
Microsoft Windows [Version 10.0.26100.3037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tarun>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.4, on Microsoft Windows [Version 10.0.26100.3037], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✗] Android toolchain - develop for Android devices
    ✗ Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
        'flutter config --android-sdk' to update to that location.

[✓] Chrome - develop for the web
[✗] Visual Studio - develop Windows apps
    ✗ Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.2)
[✓] VS Code (version 1.97.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.

C:\Users\tarun>flutter
Manage your Flutter app development.
```

Step 6: Now, run the \$ flutter command in command

```
Command Prompt - flutter dr + ▾
C:\Users\tarun>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

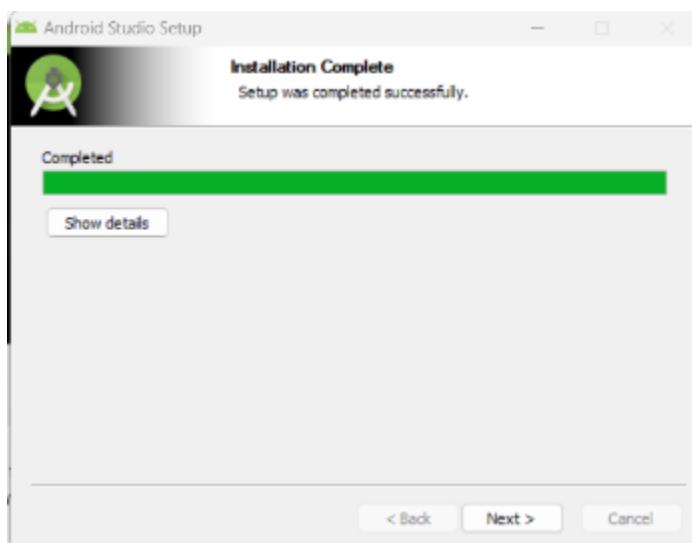
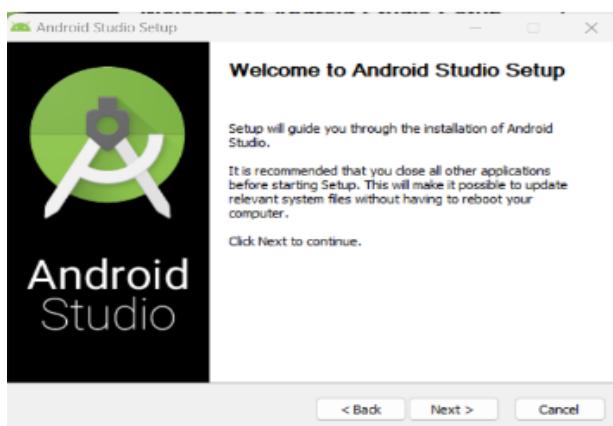
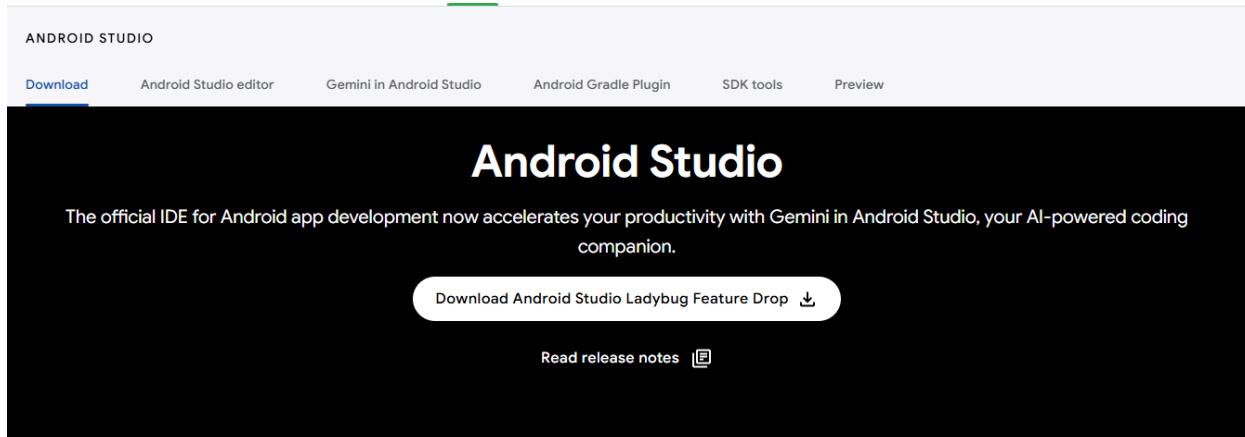
Usage: flutter <command> [arguments]

Global options:
  -h, --help                  Print this usage information.
  -v, --verbose                Noisy logging, including all shell commands executed.
                                If used with "--help", shows hidden options. If used with "flutter doctor", shows additional
                                diagnostic information. (Use "-vv" to force verbose logging in those cases.)
  -d, --device-id              Target device id or name (prefixes allowed).
  --version                   Reports the version of this tool.
  --enable-analytics          Enable telemetry reporting each time a flutter or dart command runs.
  --disable-analytics         Disable telemetry reporting each time a flutter or dart command runs, until it is
                                re-enabled.
  --suppress-analytics        Suppress analytics reporting for the current CLI invocation.

Available commands:

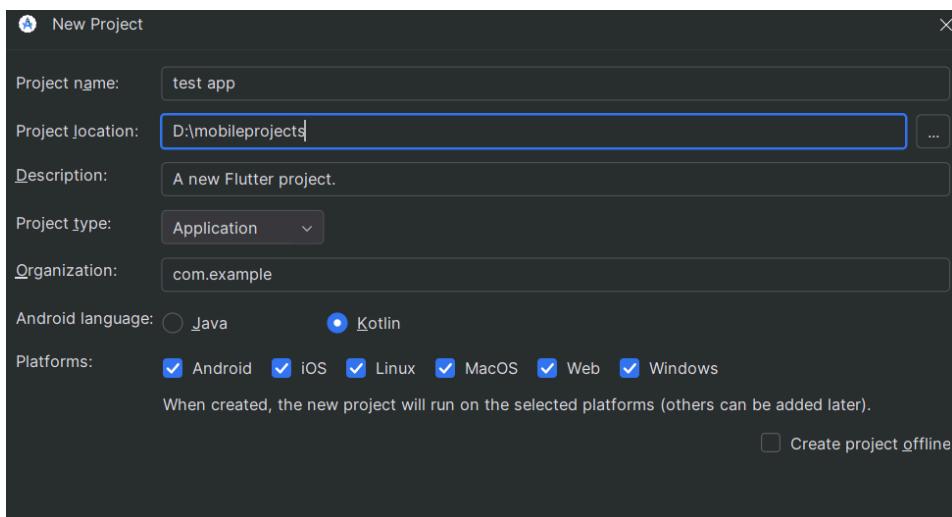
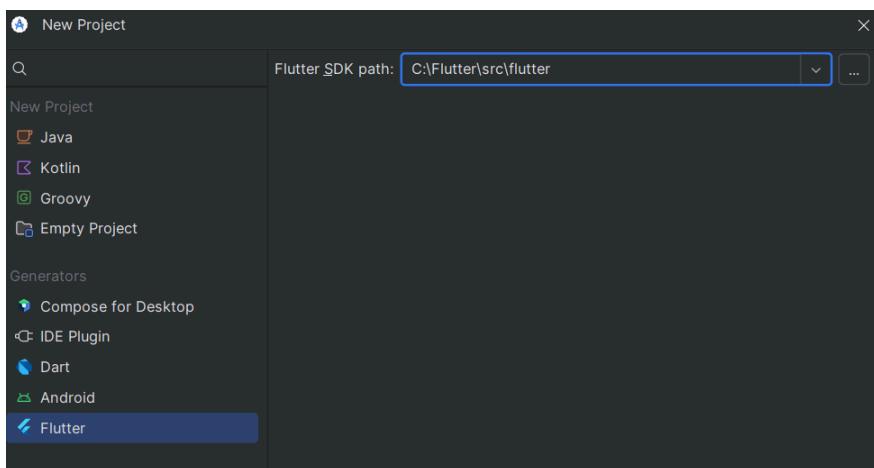
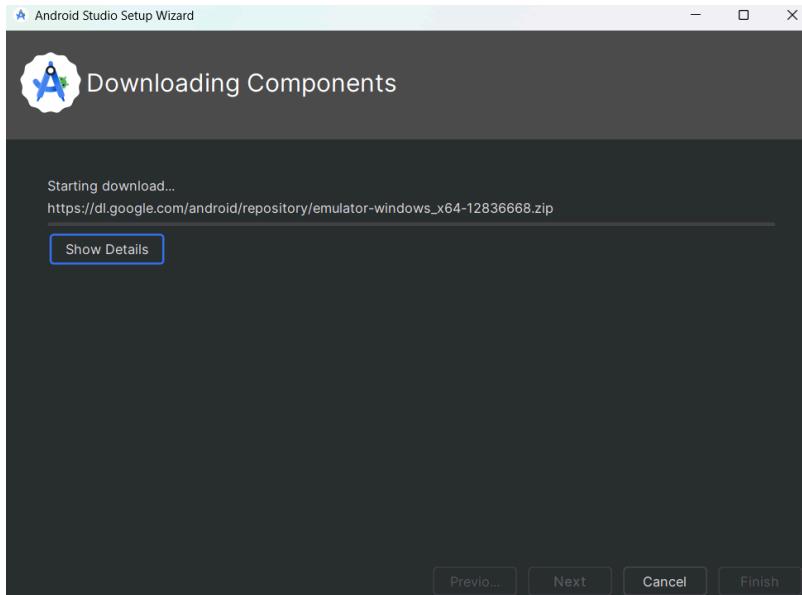
Flutter SDK
  bash-completion  Output command line shell completion setup scripts.
```

Step 7: Download and install android studio from the website



Step 8: Start Android Studio with default configurations and Download the components

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 'Virtual Device Configuration' screen in Android Studio. At the top, there's a search bar and a 'Select Hardware' button. Below it, a table lists various device categories and their details. A specific row for 'Pixel 9' is selected, showing its dimensions (6.24" screen, 1080x2424 resolution), density (420dpi), and aspect ratio (large). Below the table are buttons for 'New Hardware Profile', 'Import Hardware Profiles', and 'Clone Device...'. The second part of the screenshot shows the 'Select a system image' screen, which lists several Android versions (Baklava, VanillalceCream, UpsideDownCake, Tiramisu, Sv2, S, R) with their API levels, ABIs, xABIs, and targets. The 'Recommended' tab is selected. The bottom part of the screenshot shows the code editor with a Flutter application and a preview window showing a smartphone displaying the Flutter logo.

| Category | Name | Play Store | Size | Resolution | Density |
|------------|------------------|------------|-------|------------|---------|
| Phone | Pixel 9 Pro Fold | | 8.0" | 2076x2152 | 390dpi |
| Tablet | Pixel 9 Pro | | 6.3" | 1280x2856 | xxhdpi |
| Wear OS | Pixel 9 | | 6.24" | 1080x2424 | 420dpi |
| Desktop | Pixel 8a | | 6.1" | 1080x2400 | 420dpi |
| TV | Pixel 8 Pro | | 6.7" | 1344x2992 | xxhdpi |
| Automotive | Pixel 8 | | 6.17" | 1080x2400 | 420dpi |
| Legacy | Pixel Fold | | 7.58" | 1840x2208 | 420dpi |

| Release Name | API | ABI | xABI | Target |
|-------------------|---------|--------|-----------|-----------------------------------|
| Baklava ↴ | Baklava | x86_64 | | Android API Baklava (Google Play) |
| Baklava ↴ | Baklava | x86_64 | | Android API Baklava (16 KB Page) |
| VanillalceCream | 35 | x86_64 | arm64-v8a | Android 15.0 (Google Play) |
| VanillalceCream ↴ | 35 | x86_64 | | Android 15.0 (16 KB Page) |
| UpsideDownCake ↴ | 34 | x86_64 | | Android 14.0 (Google Play) |
| Tiramisu ↴ | 33 | x86_64 | | Android 13.0 (Google Play) |
| Sv2 ↴ | 32 | x86_64 | | Android 12L (Google Play) |
| S ↴ | 31 | x86_64 | | Android 12.0 (Google Play) |
| R ↴ | 30 | x86 | | Android 11.0 (Google Play) |


```

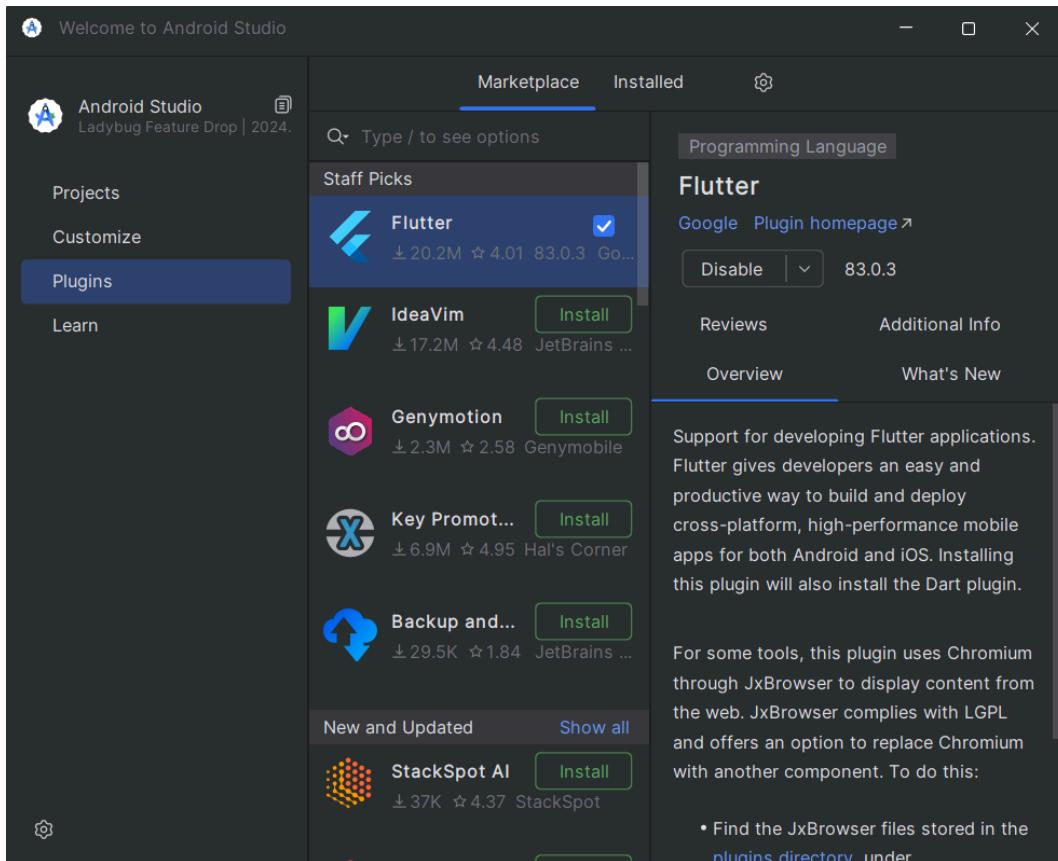
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // TRY THIS: Try running your application with "flutter run"
19       // the application has a purple toolbar. Then, without quitting
20       // try changing the seedColor in the colorScheme below to C

```

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, you've successfully installed Flutter and Android Studio, set up the Android Emulator, and are now ready to begin developing cross-platform mobile applications. With these tools in place, you can efficiently build, test, and debug your apps. Start creating your Flutter projects, experiment with different widgets, and enjoy the development process!

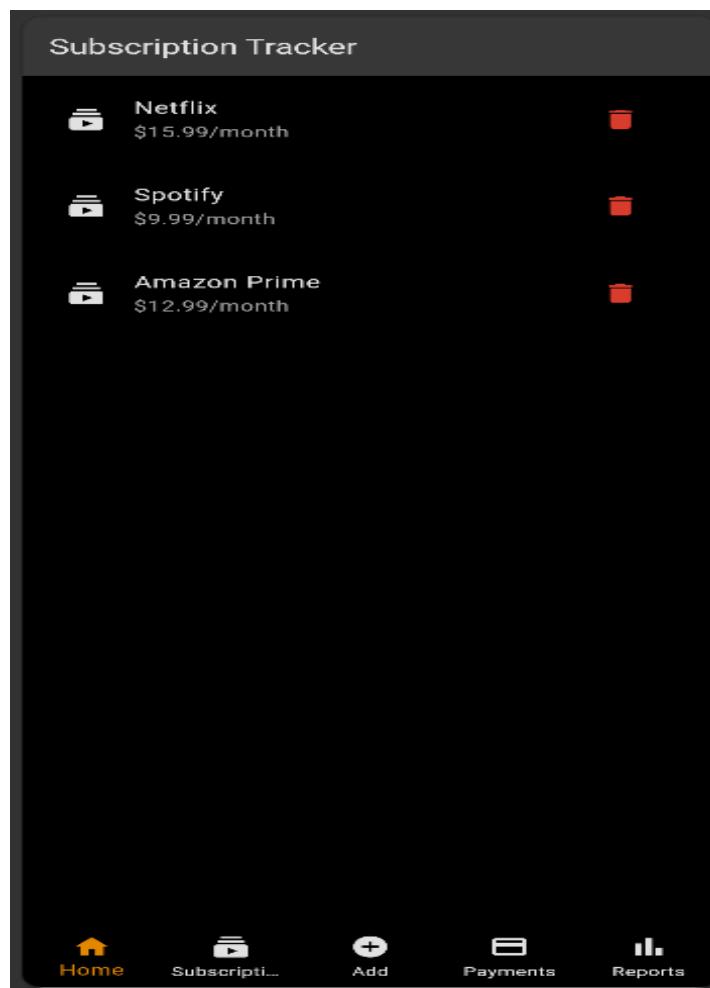
Name:Tarunkumar sharma

Div:D15B

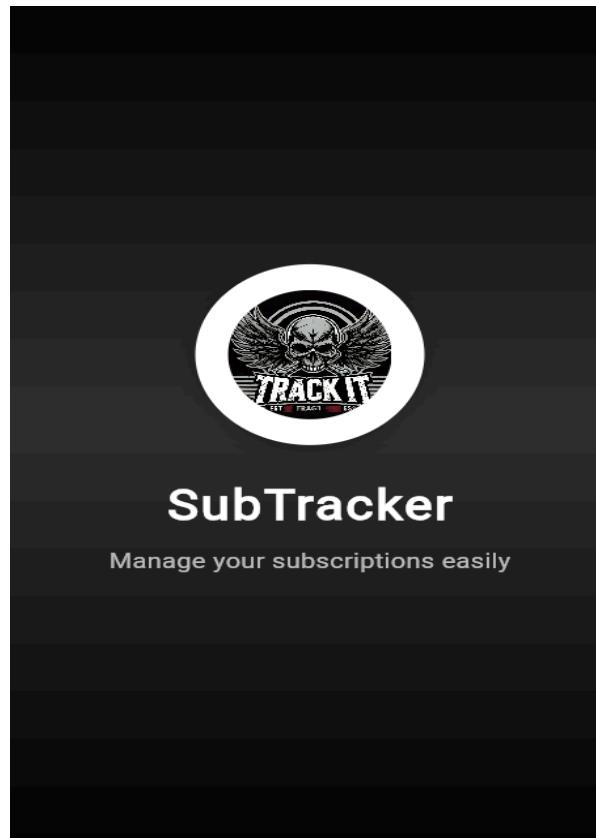
Roll no:52

Experiment 02 : To design Flutter UI by including common widgets.

1) Use of scaffold Widget



USE OF SPLASH SCREEN WIDGET



Name:Tarunkumar Sharma
Div:D15B
Roll no:52

Experiment 03 : To include icons, images, fonts in Flutter app

Icons in Flutter

Icons in Flutter help enhance UI by providing visual cues for actions, inputs, and navigation. Flutter provides **Material Icons** by default, but you can also use **custom icons** via **.ttf** font files.

1. Material Icons (Built-in)

Flutter has a built-in Icons class for Material Design icons. These are included in the Flutter SDK and do not require additional setup.

Usage in Code:

```
Icon(Icons.email, color: Colors.orange);
```

```
Icon(Icons.lock, color: Colors.orange);
```

2. Custom Icons (Using .ttf Font Files)

To use custom icons, you need a **.ttf font file** (e.g., FontAwesome or custom-designed icons).

Steps to Add Custom Icons:

Declare in pubspec.yaml:

```
flutter:
```

```
  Fonts
```

```
    - family: CustomIcons
```

```
      fonts:
```

```
- asset: assets/fonts/CustomIcons.ttf
```

Images in Flutter

Images are an essential part of a Flutter app's UI, used for branding (logos), illustrations, and user interaction. Flutter supports different types of images:

1. Types of Images in Flutter

- **Asset Images:** Stored in the project directory (`assets/` folder).
- **Network Images:** Loaded from an online URL.
- **File Images:** Loaded from the local device storage.

Since you are using an **asset image (logo.jpg)** in your code, we will focus on that.

2. Using Asset Images in Flutter

Declare Image in pubspec.yaml

Open `pubspec.yaml` and add the image under `flutter > assets`:

```
flutter:
```

```
  assets:
```

```
    - assets/images/logo.jpg
```

Fonts in Flutter

Fonts play a crucial role in designing a visually appealing UI. By default, Flutter uses the **Roboto** font, but you can add custom fonts to give your app a unique look.

1. Adding Custom Fonts in Flutter

Step 1: Place the Font File in the assets/fonts/ Folder

Save your font file (e.g., **Poppins-Regular.ttf**) inside the assets/fonts/ folder:

Declare the Font in `pubspec.yaml`

Open `pubspec.yaml` and **register the font** under `flutter > fonts`:

```
flutter:  
  fonts:  
    - family: Poppins  
      fonts:  
        - asset: assets/fonts/Poppins-Regular.ttf  
        - asset: assets/fonts/Poppins-Bold.ttf  
          weight: 700
```



Track It

Create your account

Email

Password



Sign Up

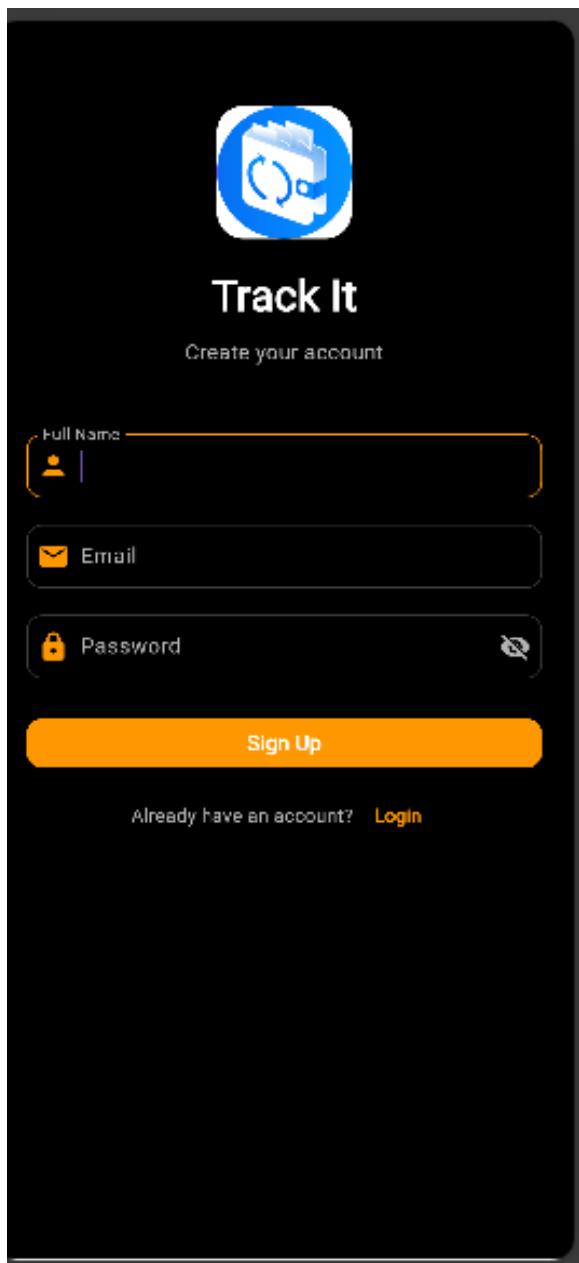
Already have an account? [Login](#)

Name:Tarunkumar Shaarma

Div:D15B

Roll no:52

Exp 4 To create an interactive Form using form widget



Experiment no:5

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:sub/pages/signup_screen.dart';

// Import screens with correct paths
import 'pages/login_screen.dart';
import 'pages/signup_screen.dart';
import 'pages/home_screen.dart';
import 'pages/splash_screen.dart';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Track It',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.orange,
        scaffoldBackgroundColor: Colors.black,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: SplashScreen(),
    );
  }
}
```

```
routes: {
    '/login': (context) => LoginScreen(),
    '/signup': (context) => SignupScreen(),
    '/home': (context) => HomePage(),
    '/splash': (context) => SplashScreen(),
},
);
}
}

// Auth wrapper to check if user is logged in
class AuthWrapper extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return StreamBuilder<User?>(
            stream: FirebaseAuth.instance.authStateChanges(),
            builder: (context, snapshot) {
                if (snapshot.connectionState == ConnectionState.active) {
                    User? user = snapshot.data;
                    return user == null ? LoginScreen() : HomePage();
                }
                return Scaffold(
                    backgroundColor: Colors.black,
                    body: Center(
                        child: CircularProgressIndicator(color: Colors.orange),
                    ),
                );
            },
        );
    }
}
```

Track It

AVG. SPENDING

| | | | | | |
|-------|-----------------|---------|----------------|------|---------------|
| | EXPENSES | | BALANCE | | INCOME |
| ₹0.00 | | ₹200.00 | | ₹200 | |

THIS MONTH'S SPEND

MAR

● Expenses ● Income

₹0 ₹300

No subscriptions yet

Tap the + button to add your first subscription

Home Subscript... + Payments Reports

Track It

Add New Subscription

Select a subscription service

Price

Renewal Period

Monthly

Next Billing Date

3/3/2025

Update Monthly Income

Cancel Add Subscription

Home Subscript... + Payments Reports

```
// File generated by FlutterFire CLI.  
// ignore_for_file: type=lint  
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;  
import 'package:flutter/foundation.dart'  
    show defaultTargetPlatform, kIsWeb, TargetPlatform;  
  
/// Default [FirebaseOptions] for use with your Firebase apps.  
///  
/// Example:  
/// ``dart  
/// import 'firebase_options.dart';  
/// // ...  
/// await Firebase.initializeApp(  
///   options: DefaultFirebaseOptions.currentPlatform,  
/// );  
/// ``  
class DefaultFirebaseOptions {  
  static FirebaseOptions get currentPlatform {  
    if (kIsWeb) {  
      return web;  
    }  
    switch (defaultTargetPlatform) {  
      case TargetPlatform.android:  
        return android;  
      case TargetPlatform.iOS:  
        return ios;  
      case TargetPlatform.macOS:  
        return macos;  
      case TargetPlatform.windows:  
        return windows;  
      case TargetPlatform.linux:  
        throw UnsupportedError(  
          'DefaultFirebaseOptions have not been configured for linux - '  
          'you can reconfigure this by running the FlutterFire CLI again.',  
        );  
      default:  
        throw UnsupportedError(  
          'DefaultFirebaseOptions are not supported for this platform.',  
        );  
    }  
  }  
  
  static const FirebaseOptions web = FirebaseOptions(  
    apiKey: 'AlzaSyA60WlsPf1Xxmi8oV3eTG4RQDLnvTM42E0',
```

```
appId: '1:122488913319:web:3be64022079add72a48a82',
messagingSenderId: '122488913319',
projectId: 'trackit-20',
authDomain: 'trackit-20.firebaseio.com',
storageBucket: 'trackit-20.firebaseiostorage.app',
measurementId: 'G-P7CFY0RGZJ',
);

static const FirebaseOptions android = FirebaseOptions(
  apiKey: 'AlzaSyAH1wegEW-CSipF4jTOZyv1ZreOcvWBWF4',
  appId: '1:122488913319:android:c29509bcd792f52a48a82',
  messagingSenderId: '122488913319',
  projectId: 'trackit-20',
  storageBucket: 'trackit-20.firebaseiostorage.app',
);

static const FirebaseOptions ios = FirebaseOptions(
  apiKey: 'AlzaSyDI37LBet0qeiMUGyN7Dqqb7fZwKrBRGOE',
  appId: '1:122488913319:ios:d5a35a665624dd0fa48a82',
  messagingSenderId: '122488913319',
  projectId: 'trackit-20',
  storageBucket: 'trackit-20.firebaseiostorage.app',
  iosBundleId: 'com.example.sub',
);

static const FirebaseOptions macos = FirebaseOptions(
  apiKey: 'AlzaSyDI37LBet0qeiMUGyN7Dqqb7fZwKrBRGOE',
  appId: '1:122488913319:ios:d5a35a665624dd0fa48a82',
  messagingSenderId: '122488913319',
  projectId: 'trackit-20',
  storageBucket: 'trackit-20.firebaseiostorage.app',
  iosBundleId: 'com.example.sub',
);

static const FirebaseOptions windows = FirebaseOptions(
  apiKey: 'AlzaSyA60WlsPf1Xxmi8oV3eTG4RQDLnvTM42E0',
  appId: '1:122488913319:web:f43c971a2e7f322fa48a82',
  messagingSenderId: '122488913319',
  projectId: 'trackit-20',
  authDomain: 'trackit-20.firebaseio.com',
  storageBucket: 'trackit-20.firebaseiostorage.app',
  measurementId: 'G-XE9CQSDY7G',
);
```

Trackit ▾

Authentication

Users Sign-in method Templates Usage Settings Extensions

The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

| Identifier | Providers | Created | Signed In | User UID |
|------------------------|-----------|--------------|--------------|-----------------------------|
| swayamraut8@gmail.c... | ✉️ | Mar 18, 2025 | Mar 18, 2025 | M2ljhPJs5tZaRwp9MDbERp69... |
| tarun12@gmail.com | ✉️ | Mar 18, 2025 | Mar 24, 2025 | WhKpBhi0HpT9Mfi2X24unWY... |
| death12@gmail.com | ✉️ | Mar 4, 2025 | Mar 18, 2025 | wdS30GeD5SZSPs1AmIX9Ao... |
| death@gmail.com | ✉️ | Mar 3, 2025 | Mar 3, 2025 | Tiuk9Y07bsRhw0BsLsRqJ6FJ... |
| tarun20@gmail.com | ✉️ | Mar 3, 2025 | Mar 3, 2025 | QsSUkSiStJOWx1GqNfZOPKO... |

Search by email address, phone number, or user UID Add user ⌂ ⌂

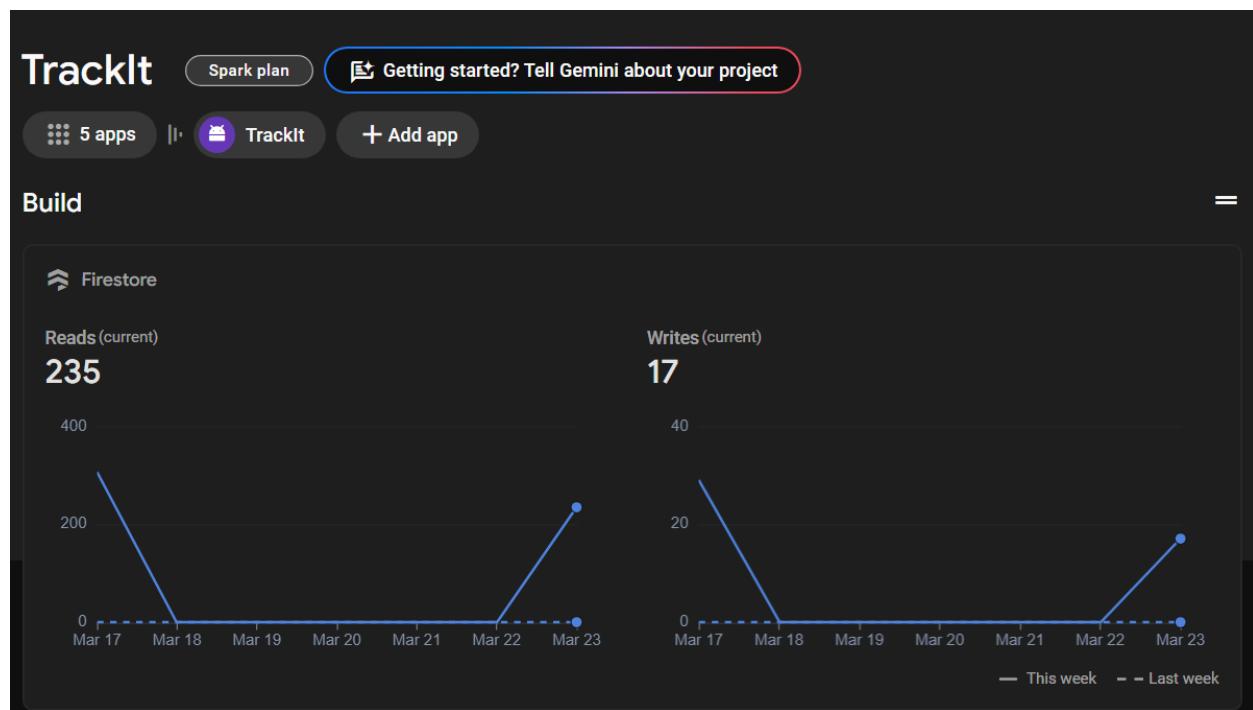
users > users > WhKpBhi0HpT9... > subscriptions More in Google Cloud ▾

| users | WhKpBhi0HpT9Mfi2X24unWYFZF23 | subscriptions |
|------------------------------|--|----------------------|
| + Add document | + Start collection | + Add document |
| M2ljhPJs5tZaRwp9MDbERp69Qmv2 | expenses | 4jDEAT05xwH7JmQAMFLZ |
| WhKpBhi0HpT9Mfi2X24unWYFZF23 | subscriptions > | bskLVe8CfDwICY0x4nDn |
| wdS30GeD5SZSPs1AmIX9Ao5jdL2 | + Add field | i0kENF1vlpHctE08T7fR |
| | createdAt: March 18, 2025 at 9:21:01 AM UTC+5:30 | rY1Qu5dh0mfUw4QXbAGM |
| | email: "tarun12@gmail.com" | yMPFhx0zqQ0jjYWt357A |
| | monthlyIncome: 1000 | |
| | name: "tarun" | |
| | updatedAt: March 24, 2025 at 4:19:49 PM UTC+5:30 | |

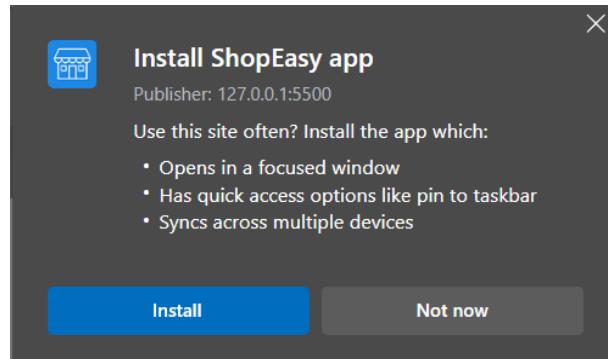
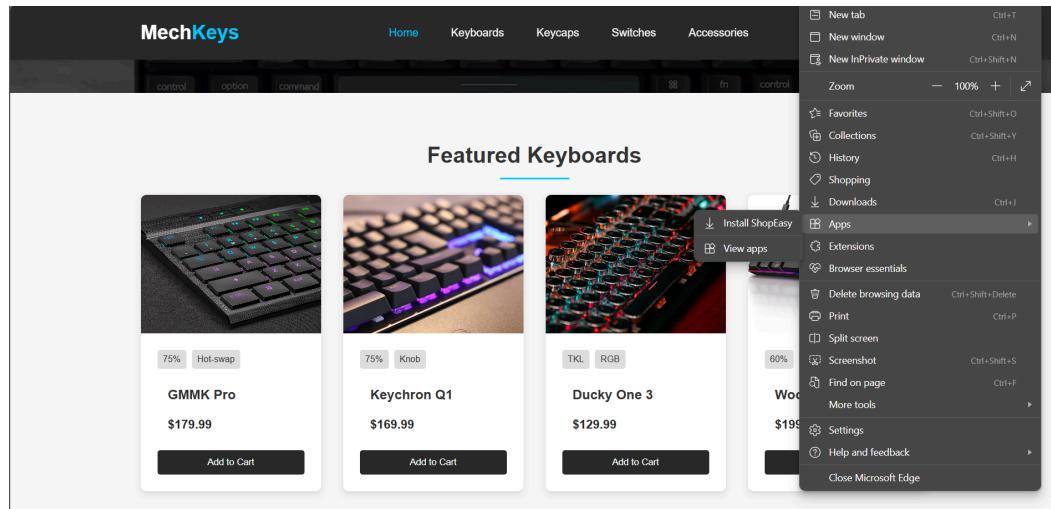
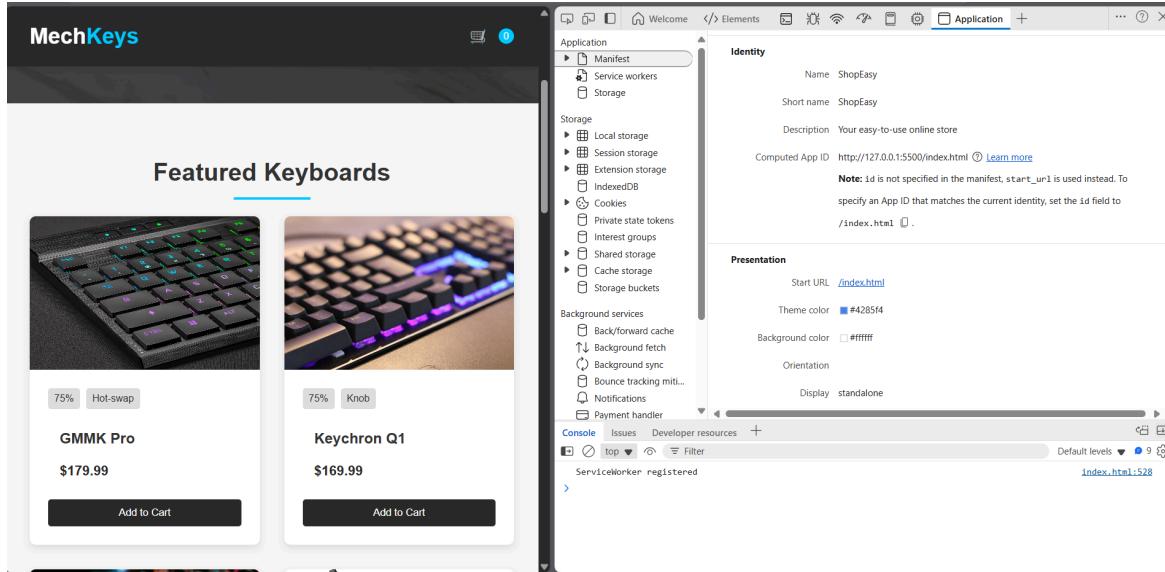
options > iOkENF1vlpHCT...

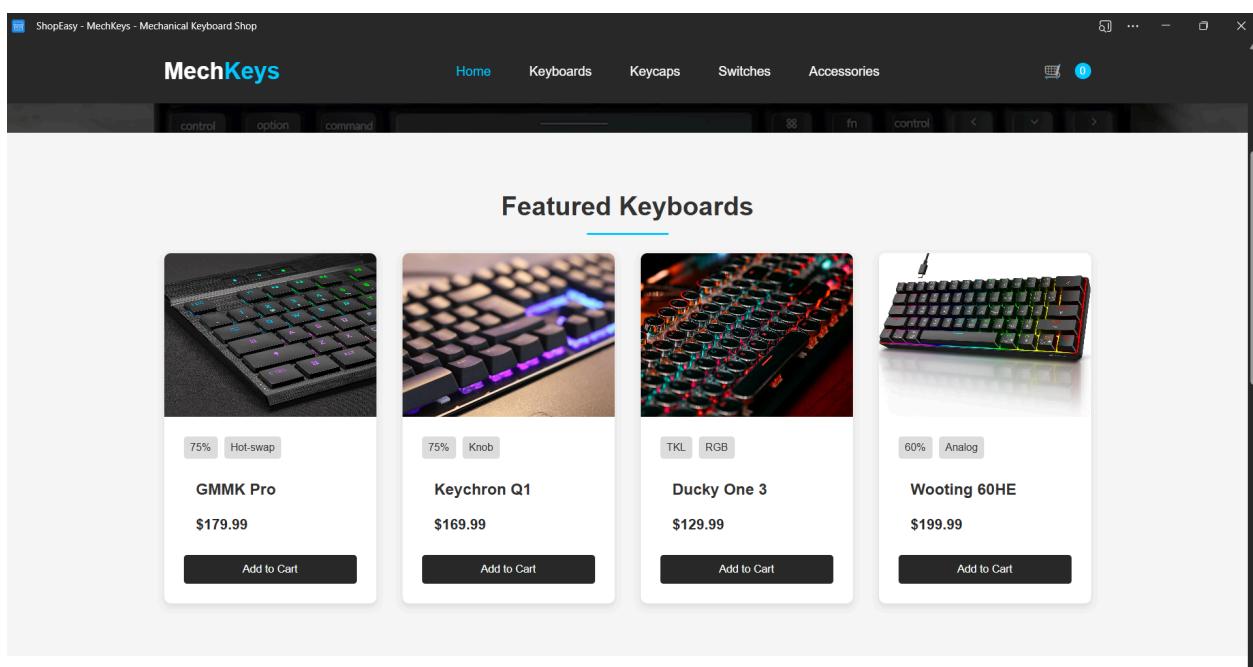
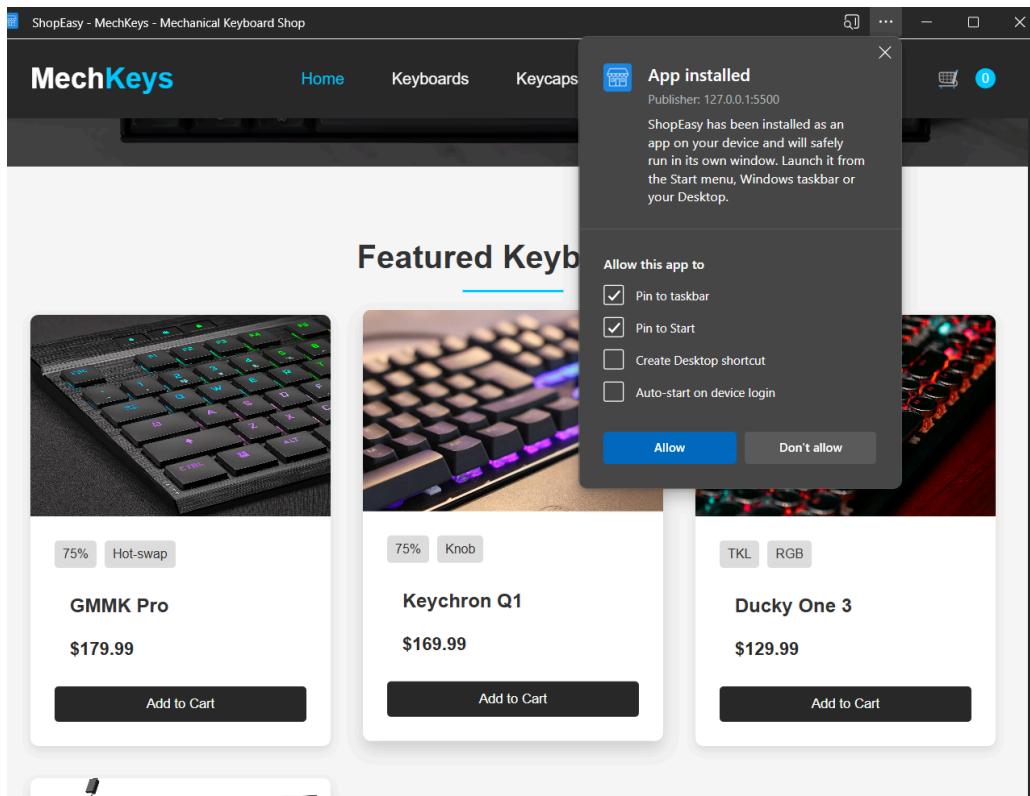
More in Google Cloud

| | | |
|------------------------|--|---|
| subscriptions | iOkENF1vlpHCTE00T7fR | ⋮ |
| + Add document | + Start collection | |
| 4jDEAT05xwH7JmQAMFLZ | | |
| bskLVe8CfDwICY0x4nDn | | |
| ⋮ iOkENF1vlpHCTE00T7fR | + Add field | ⋮ |
| rY1Qu5dh0mfUw4QXbAGM | category: "Music" | |
| yMPFhxQzqQ0jjYWt357A | logoPath: "assets/subscriptions/spotify.png" | |
| | name: "Spotify" | |
| | price: 1 | |
| | renewalDate: "23/3/2025" | |
| | renewalPeriod: "Quarterly" | |



PWA EXPERIMENT - 7





Experiment N0-8

Code:

```
// Change this to a new version to trigger an update
const CACHE_NAME = "pwa-cache-v2";

const ASSETS_TO_CACHE = [
  "./",
  "./index.html",
  "./styles.css",
  "./app.js",
  "./manifest.json",
  "./images/shop_icon.png",
];

// Install event - Cache important assets
self.addEventListener("install", (event) => {
  console.log("✓ Service Worker Installed");

  // This line forces the waiting service worker to become active
  self.skipWaiting();

  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");

  // This ensures the service worker takes control immediately
  event.waitUntil(clients.claim());

  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 - Modified to always check network first for HTML files
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);

  // Always get fresh HTML from network
  if (
    event.request.mode === "navigate" ||
    (event.request.method === "GET" &&
     event.request.headers.get("accept").includes("text/html"))
  ) {
    console.log("🔄 Getting fresh HTML for:", url.pathname);

    event.respondWith(
      fetch(event.request).catch(() => caches.match("./index.html"))
    );
  } else {
    // For non-HTML resources, use cache-first strategy
    event.respondWith(
      caches
        .match(event.request)
        .then((response) => {
          return (
            response ||
            fetch(event.request).then((fetchResponse) => {
              // Save new resources in cache
              const responseClone = fetchResponse.clone();
              caches
                .open(CACHE_NAME)
                .then((cache) => cache.put(event.request, responseClone));
            })
          );
        })
    );
  }
});
```

```

        return fetchResponse;
    })
    );
}
)
.catch(() => {
// Fallback for offline
if (event.request.url.includes(".html")) {
return caches.match("./index.html");
}
})
);
}

});

// Rest of your code remains the same
self.addEventListener("sync", (event) => {
if (event.tag === "sync-cart") {
event.waitUntil(syncCart());
}
});
}

self.addEventListener("push", (event) => {
const options = {
body: "This is a push notification!",
icon: "/images/shop_icon.png",
badge: "/images/badge.png",
};
event.waitUntil(self.registration.showNotification("New Message!", options));
});

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
? event.data.json()
:{ title: "New Notification", body: "Check out our latest products!" };

```

```

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));
});

```

The screenshot shows the Chrome DevTools interface with the Application tab selected. On the left, the Storage section is expanded, showing Local storage, Session storage, Extension storage (IndexedDB), Cookies, Shared storage, Cache storage, and Storage buckets. The Service workers section is also expanded, showing a manifest file. The main panel displays the service worker for the URL <http://127.0.0.1:5500/>. The source is listed as `service-worker.js`, and it was received on 13/4/2025, 10:39:27 am. The status is shown as activated and running. Under the Push section, there is a text input field containing "Test push message from DevTools." and a "Push" button. Under Sync, there is a text input field containing "test-tag-from-devtools" and a "Sync" button. Under Periodic sync, there is a text input field containing "test-tag-from-devtools" and a "Periodic sync" button. The Update Cycle section shows three entries: #131 Install, #131 Wait, and #131 Activate, with the Activate entry having a progress bar indicating completion. Below the main panel, a separate window or tab shows a list of network requests and events:

- Service Worker Installed
- Caching assets
- Fetching: <chrome-extension://gppongmhjkpfnbhagpmjfkannfbllamg/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-4.0.3&t=1650000000>
- Fetching: <https://images.unsplash.com/photo-1553062407-98eeb64c6a62?ixlib=rb-4.0.3&t=1650000000>

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

- Fetching: <https://images.unsplash.com/photo-1579586337278-3befd40fd17a?ixlib=rb-4.0.3&t=1650000000>
- Fetching: <https://images.unsplash.com/photo-1608043152269-423dbba4e7e1?ixlib=rb-4.0.3&t=1650000000>
- Fetching: <http://127.0.0.1:5500/PWA/app.js>
- Fetching: <http://127.0.0.1:5500/PWA/manifest.json>

Screenshot of the Network tab in the Chrome DevTools Performance panel, showing a request for database.json.

Request Headers:

- Accept: */*
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.5
- Connection: keep-alive
- Host: 127.0.0.1:5500
- Referer: http://127.0.0.1:5500/index.html

Response Headers:

- Status: 200 OK
- Version: HTTP/1.1
- Transferred: 0 B (2.24 kB size)
- Referrer Policy: strict-origin-when-cross-origin
- DNS Resolution: System

Raw Response Headers:

```
Accept-Ranges: bytes
Access-Control-Allow-Credentials: true
Cache-Control: public, max-age=0
Connection: keep-alive
Content-Length: 2243
Content-Type: application/json; charset=UTF-8
Date: Sun, 13 Apr 2025 07:10:33 GMT
ETag: W/"8c3-1962d9e25aa"
Keep-Alive: timeout=5
Last-Modified: Sun, 13 Apr 2025 05:27:41 GMT
Vary: Origin
```

Timing Details:

- 1 request
- 2.24 kB / 0 B transferred
- Finish: 248 ms
- DOMContentLoaded: 41 ms

EXPERIMENT NO-9

```
// Change this to a new version to trigger an update
const CACHE_NAME = "pwa-cache-v2";

const ASSETS_TO_CACHE = [
  "./",
  "./index.html",
  "./styles.css",
  "./app.js",
  "./manifest.json",
  "./images/shop_icon.png",
];
;

// Install event - Cache important assets
self.addEventListener("install", (event) => {
  console.log("✓ Service Worker Installed");

  // This line forces the waiting service worker to become active
  self.skipWaiting();

  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");

  // This ensures the service worker takes control immediately
  event.waitUntil(clients.claim());

  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 - Modified to always check network first for HTML files
self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);

  // Always get fresh HTML from network
  if (
    event.request.mode === "navigate" ||
    (event.request.method === "GET" &&
     event.request.headers.get("accept").includes("text/html"))
  ) {
    console.log("🔄 Getting fresh HTML for:", url.pathname);

    event.respondWith(
      fetch(event.request).catch(() => caches.match("./index.html"))
    );
  } else {
    // For non-HTML resources, use cache-first strategy
    event.respondWith(
      caches
        .match(event.request)
        .then((response) => {
          return (
            response ||
            fetch(event.request).then((fetchResponse) => {
              // Save new resources in cache
              const responseClone = fetchResponse.clone();
              caches
                .open(CACHE_NAME)
                .then((cache) => cache.put(event.request, responseClone));
            })
          );
        })
    );
  }
});
```

```

        return fetchResponse;
    })
    );
})
)
.catch(() => {
// Fallback for offline
if (event.request.url.includes(".html")) {
return caches.match("./index.html");
}
})
);
}

});

// Rest of your code remains the same
self.addEventListener("sync", (event) => {
if (event.tag === "sync-cart") {
event.waitUntil(syncCart());
}
});
}

self.addEventListener("push", (event) => {
const options = {
body: "This is a push notification!",
icon: "/images/shop_icon.png",
badge: "/images/badge.png",
};
event.waitUntil(self.registration.showNotification("New Message!", options));
});

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
}

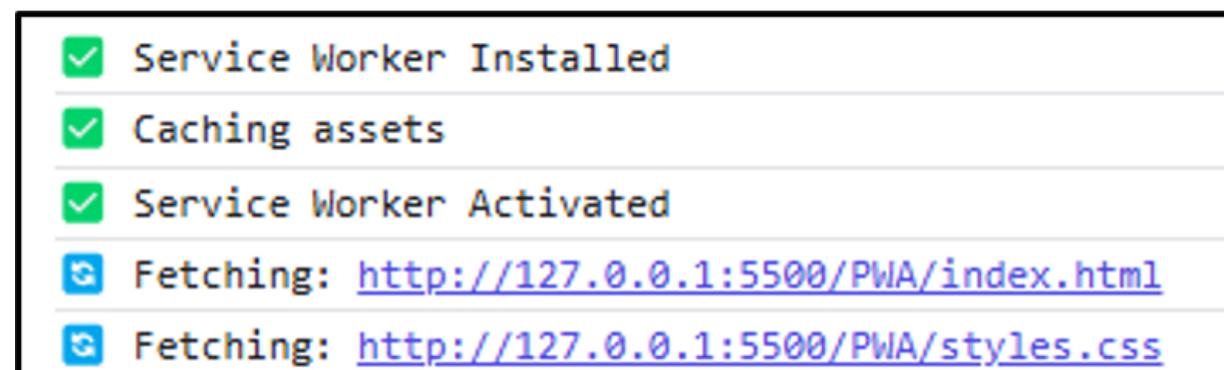
// Push event - Handle push notifications
self.addEventListener("push", (event) => {
const data = event.data
? event.data.json()
:{ title: "New Notification", body: "Check out our latest products!" };

```

```

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));
});

```



```

Fetching: chrome-extension://gpoongmhjkpfnbhagomjfkannfbllampm/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
⌚ Fetching: http://127.0.0.1:5500/PWA/manifest.json
✓ Service Worker registered successfully: → ServiceWorkerRegistration {installing: null, waiting: null, active: ServiceWorker, navi
⌚ Fetching: http://127.0.0.1:5500/PWA/images/shop_icon144.png
⌚ Fetching: http://127.0.0.1:5500/PWA/images/hero-hg.jpg
⌚ Fetching: chrome-extension://gpoongmhjkpfnbhagomjfkannfbllampm/js/js.js
⌚ Fetching: chrome-extension://gpoongmhjkpfnbhagomjfkannfbllampm/js/dom.js

```

Console Issues Application Developer resources +

Application

- Manifest
- Service workers
- Storage

Storage

- Local storage
- Session storage
- Extension storage
- IndexedDB
- Cookies
- Private state tokens
- Interest groups
- Shared storage
- Cache storage
- Storage buckets

Background services

- Back/forward cache
- Background fetch
- Background sync
- ...

Source [service-worker.js](#)
Received 13/4/2025, 1:20:04 pm

Status ● #149 activated and is running Stop

Clients <http://127.0.0.1:5500/index.html>

Push `'meathod': 'pushMessage', "message": "Hello this is tarun"` Push

Sync `test-tag-from-devtools` Sync

Periodic sync `test-tag-from-devtools` Periodic sync

Update Cycle

| Version | Update Activity | Timeline |
|---------|-----------------|-------------|
| ► #149 | Install | |
| ► #149 | Wait | |
| ► #149 | Activate | ███████████ |

Service workers from other origins
[See all registrations](#)

Service Worker: Push notification received:

► `{meathod: 'pushMessage', message: 'Hello this is tarun '}`

[service-worker.js:123](#)

Experiment No-10

The screenshot shows a GitHub repository page for 'pwa10'. The repository is public and has one commit from Tarun Sharma. The commit details are as follows:

| File | Action | Time |
|-------------------|--------|---------------|
| images | add | 3 minutes ago |
| app.js | add | 3 minutes ago |
| database.json | add | 3 minutes ago |
| index.html | add | 3 minutes ago |
| manifest.json | add | 3 minutes ago |
| service-worker.js | add | 3 minutes ago |
| styles.css | add | 3 minutes ago |

The repository has 1 branch and 0 tags. The 'About' section indicates no description, website, or topics provided. The 'Activity' section shows 0 stars, 1 watching, and 0 forks. There are no releases or packages published. A deployment named 'github-pages' was made 1 minute ago.

<https://tarun207.github.io/pwa10/>

The screenshot shows the GitHub Pages preview interface. It displays the heading 'GitHub Pages' and a brief description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.' Below this, it shows the live site URL 'Your site is live at <https://tarun207.github.io/pwa10/>' and the deployment information 'Last deployed by  Tarun207 4 minutes ago'. There are 'Visit site' and '...' buttons at the bottom right.

<https://tarun207.github.io/pwa10/>

MechKeys

Home Keyboards Keycaps Switches Accessories

Elevate Your Typing Experience

Premium mechanical keyboards, keycaps, and accessories for enthusiasts and professionals

Shop Collection

Featured Keyboards



<https://tarun207.github.io/pwa10/>

MechKeys

Home Keyboards Keycaps Switches Accessories

Featured Keyboards



75% Hot-swap

GMMK Pro

\$179.99

Add to Cart

75% Knob

Keychron Q1

\$169.99

Add to Cart

TKL RGB

Ducky One 3

\$129.99

Add to Cart

60% Analog

Wooting 60HE

\$199.99

Add to Cart

Experiment No-11

The screenshot shows the Lighthouse tool's configuration interface. At the top, there are various icons and a "Welcome" button. Below that, a message says "The Lighthouse tool provides links to content hosted on third-party websites." with a "Don't show again" button. A "Show more" button is also present. In the center, there is a "Generate a Lighthouse report" button with a person icon. To its right is a "Analyze page load" button. On the left, under "Mode", "Navigation (Default)" is selected. Under "Device", "Mobile" is selected. Under "Categories", all checkboxes are checked: Performance, Accessibility, Best practices, and SEO. There is also a "Learn more" link.

The screenshot shows the Lighthouse tool's generated report interface. At the top, it displays the same header and message as the configuration screen. Below that, it shows a report for "2:06:55 pm - 127.0.0.1:5500" at "http://127.0.0.1:5500/index.html". It features four large circular performance metrics: "Performance" (100), "Accessibility" (93), "Best Practices" (75), and "SEO" (100). Below these, there is a large green circle with a "100" rating labeled "Performance". A note below states: "Values are estimated and may vary. The [performance score](#) is [calculated](#) directly from these metrics. [See calculator.](#)" At the bottom, there is a legend: a red triangle for 0-49, an orange square for 50-89, and a green circle for 90-100. To the right, there is a preview of a mobile device displaying a keyboard advertisement for "MechKeys".

MechKeys

Elevate Your Typing Experience

Premium mechanical keyboards, keycaps, and accessories for enthusiasts and professionals

Shop Collection

Featured Keyboards

METRICS

- First Contentful Paint **1.0 s**
- Largest Contentful Paint **1.8 s**
- Total Blocking Time **50 ms**
- Cumulative Layout Shift **0**
- Speed Index **1.0 s**

View Treemap

Show audits relevant to: FCP ICP TRT

MechKeys

Elevate Your Typing Experience

Premium mechanical keyboards, keycaps, and accessories for enthusiasts and professionals

Shop Collection

Featured Keyboards

Audit Details

- Minify CSS — Potential savings of 7 KB
- Minify JavaScript — Potential savings of 216 KB
- Serve Images in next-gen formats — Potential savings of 163 KB
- Serve static assets with an efficient cache policy — 6 resources found
- Eliminate render-blocking resources — Potential savings of 0 ms
- Properly size images — Potential savings of 840 KB
- Reduce unused CSS — Potential savings of 65 KB
- Reduce unused JavaScript — Potential savings of 255 KB
- Avoid long main-thread tasks — 3 long tasks found
- Avoid chaining critical requests — 1 chain found

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)

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

The Lighthouse tool provides links to content hosted on third-party websites. [Don't show again](#)

2:06:55 pm - 127.0.0.1:5500 [🔗](#)

http://127.0.0.1:5500/index.html

100 93 75 100

Best Practices

GENERAL

- ⚠️ Uses third-party cookies — 30 cookies found
- ⚠️ Issues were logged in the [Issues](#) panel in Chrome Devtools

USER EXPERIENCE

- ⚠️ Displays images with incorrect aspect ratio

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks

The Lighthouse tool provides links to content hosted on third-party websites. [Don't show again](#)

2:06:55 pm - 127.0.0.1:5500 [🔗](#)

http://127.0.0.1:5500/index.html

100 93 75 100

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)

- Structured data is valid

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

PASSED AUDITS (8)

NOT APPLICABLE (2)