Experiment No. 11

Aim : To study and implement deployment of your app to GitHub Pages.

Theory :

Step 1: Creating a flutter web project.
To create a flutter web project, you have to be in one of the following flutter channels : beta, dev or master. Change your channel and upgrade it, if you were in stable and then proceed to the next step.
Enable support for flutter-web in terminal/command-prompt:

```
$ flutter config --enable-web
```

Now, create your flutter project like you usually do and it will have web support. If you want to add web support to an existing project you can use the following command inside the directory:

```
$ flutter create .
```

After creating your flutter project, you can check if you have the web directory. It shows that your project supports flutter-web.
The sources for the examples are at the end of this page.
Now, you can also publish it on GitHub.
Step 2: Making your flutter-web build.
You can make a release build for the flutter-web using the command:

```
$ flutter build web --release
```

Now, you will have a new directory named build and you will find your web build in it, like this.

Now if you look inside that web folder, you will find the build files. As the dart code trans compiles into javascript code with HTML and CSS, the starting point of the build naturally, is the index.html .
Step 3: Publishing the build to GitHub.
You can now start a new repository or clone a repository you already have in a new folder/location .

This is how the cloned repository will look like (mine is just empty).
Then rename it to web and paste it into flutter-project/build/ directory, before really building the web build.

Now, that you have done everything, you can edit your flutter code just like you want, anytime from the flutter-project and build it. Then the changes can be committed and pushed from flutter-project/build/web . This reduces our worry to handle the build files to host and maintain two different repositories, one for the flutter project and the other exclusively for hosting.
tip: You can add build directory to your .gitignore file in your flutter project folder to avoid confusion and repetition.
Step 4: Hosting it on GitHub Pages.

After you have successfully pushed the build files you got from the flutter project, go to the repository page. Then, navigate to the settings, you will find a title “GitHub Pages”. Select your source as main branch and save.

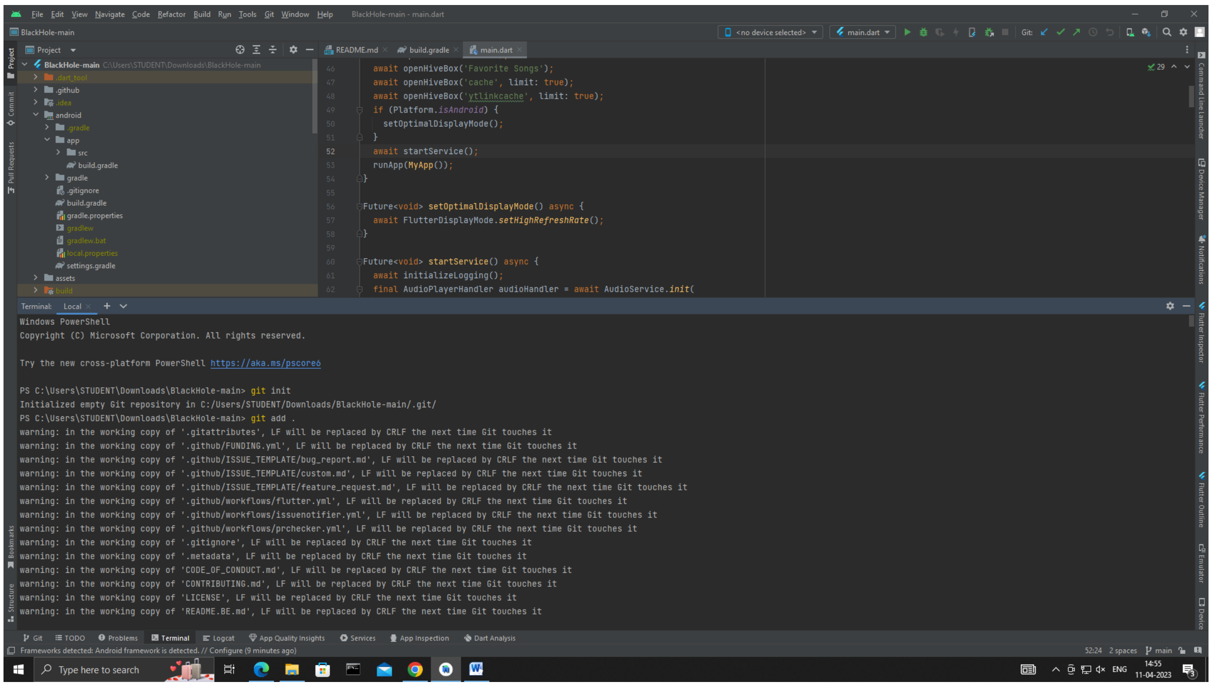
Code :

```
import …
Future<void> main() async {
 WidgetsFlutterBinding.ensureInitialized();
 Paint.enableDithering = true;

 if (Platform.isWindows || Platform.isLinux || Platform.isMacOS) {
   await Hive.initFlutter('BlackHole');
 } else {
   await Hive.initFlutter();
 }
 await openHiveBox('settings');
 await openHiveBox('downloads');
 await openHiveBox('stats');
 await openHiveBox('Favorite Songs');
 await openHiveBox('cache', limit: true);
 await openHiveBox('ytlinkcache', limit: true);
 if (Platform.isAndroid) {
   setOptimalDisplayMode();
 }
 await startService();
 runApp(MyApp());
}

Future<void> setOptimalDisplayMode() async {
 await FlutterDisplayMode.setHighRefreshRate();
}

Future<void> startService() async {
 await initializeLogging();
 final AudioPlayerHandler audioHandler = await AudioService.init(
   builder: () => AudioPlayerHandlerImpl(),
   config: AudioServiceConfig(
     androidNotificationChannelId: 'com.shadow.blackhole.channel.audio',
     androidNotificationChannelName: 'BlackHole',
     androidNotificationIcon: 'drawable/ic_stat_music_note',
     androidShowNotificationBadge: true,
     androidStopForegroundOnPause: false,
     // Hive.box('settings').get('stopServiceOnPause', defaultValue: true) as
bool,
     notificationColor: Colors.grey[900],
   ),
 );
 GetIt.I.registerSingleton<AudioPlayerHandler>(audioHandler);
 GetIt.I.registerSingleton<MyTheme>(MyTheme());
}

Future<void> openHiveBox(String boxName, {bool limit = false}) async {
 final box = await Hive.openBox(boxName).onError((error, stackTrace) async {
   Logger.root.severe('Failed to open $boxName Box', error, stackTrace);
   final Directory dir = await getApplicationDocumentsDirectory();
   final String dirPath = dir.path;
   File dbFile = File('$dirPath/$boxName.hive');
   File lockFile = File('$dirPath/$boxName.lock');
   if (Platform.isWindows || Platform.isLinux || Platform.isMacOS) {
     dbFile = File('$dirPath/BlackHole/$boxName.hive');
```

```dart
      lockFile = File('$dirPath/BlackHole/$boxName.lock');
    }
    await dbFile.delete();
    await lockFile.delete();
    await Hive.openBox(boxName);
    throw 'Failed to open $boxName Box\nError: $error';
  });
  // clear box if it grows large
  if (limit && box.length > 500) {
    box.clear();
  }
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();

  static _MyAppState of(BuildContext context) =>
      context.findAncestorStateOfType<_MyAppState>()!;
}

class _MyAppState extends State<MyApp> {
  Locale _locale = const Locale('en', '');
  late StreamSubscription _intentTextStreamSubscription;
  late StreamSubscription _intentDataStreamSubscription;
  final GlobalKey<NavigatorState> navigatorKey = GlobalKey<NavigatorState>();

  @override
  void dispose() {
    _intentTextStreamSubscription.cancel();
    _intentDataStreamSubscription.cancel();
    super.dispose();
  }

  @override
  void initState() {
    super.initState();
    final String systemLangCode = Platform.localeName.substring(0, 2);
    if (ConstantCodes.languageCodes.values.contains(systemLangCode)) {
      _locale = Locale(systemLangCode);
    } else {
      final String lang =
          Hive.box('settings').get('lang', defaultValue: 'English') as String;
      _locale = Locale(ConstantCodes.languageCodes[lang] ?? 'en');
    }

    AppTheme.currentTheme.addListener(() {
      setState(() {});
    });

    // For sharing or opening urls/text coming from outside the app while the
app is in the memory
    _intentTextStreamSubscription =
ReceiveSharingIntent.getTextStream().listen(
      (String value) {
        Logger.root.info('Received intent on stream: $value');
        handleSharedText(value, navigatorKey);
      },
      onError: (err) {
        Logger.root.severe('ERROR in getTextStream', err);
      },
    );
```

```
   // For sharing files coming from outside the app while the app is in the
memory
   _intentDataStreamSubscription =
       ReceiveSharingIntent.getMediaStream().listen(
     (List<SharedMediaFile> value) {
       if (value.isNotEmpty) {
         for (final file in value) {
           if (file.path.endsWith('.json')) {
             final List playlistNames = Hive.box('settings')
                     .get('playlistNames')
                     ?.toList() as List? ??
                 ['Favorite Songs'];
             importFilePlaylist(
               null,
               playlistNames,
               path: file.path,
               pickFile: false,
             ).then(
               (value) => navigatorKey.currentState?.pushNamed('/playlists'),
             );
           }
         }
       }
     },
     onError: (err) {
       Logger.root.severe('ERROR in getDataStream', err);
     },
   );

    void setLocale(Locale value) {
   setState(() {
     _locale = value;
   });
 }

 Widget initialFuntion() {
   return Hive.box('settings').get('userId') != null
       ? HomePage()
       : AuthScreen();
 }

 @override
 Widget build(BuildContext context) {
   SystemChrome.setSystemUIOverlayStyle(
     SystemUiOverlayStyle(
       statusBarColor: Colors.transparent,
       systemNavigationBarColor: AppTheme.themeMode == ThemeMode.dark
           ? Colors.black38
           : Colors.white,
       statusBarIconBrightness: AppTheme.themeMode == ThemeMode.dark
           ? Brightness.light
           : Brightness.dark,
       systemNavigationBarIconBrightness: AppTheme.themeMode ==
ThemeMode.dark
           ? Brightness.light
           : Brightness.dark,
     ),
   );
   SystemChrome.setPreferredOrientations([
     DeviceOrientation.portraitUp,
     DeviceOrientation.portraitDown,
     DeviceOrientation.landscapeLeft,
     DeviceOrientation.landscapeRight,
   ]);
   return MaterialApp(
     title: 'BlackHole',
     restorationScopeId: 'blackhole',
     debugShowCheckedModeBanner: false,
     themeMode: AppTheme.themeMode,
     theme: AppTheme.lightTheme(
       context: context,
     ),
     darkTheme: AppTheme.darkTheme(
       context: context,
     ),
     locale: _locale,
     localizationsDelegates: const [
       AppLocalizations.delegate,
       GlobalMaterialLocalizations.delegate,
       GlobalWidgetsLocalizations.delegate,
       GlobalCupertinoLocalizations.delegate,
     ],
     supportedLocales: ConstantCodes.languageCodes.entries
         .map((languageCode) => Locale(languageCode.value, ''))
         .toList(),
     routes: {
       '/': (context) => initialFuntion(),
       '/pref': (context) => const PrefScreen(),
       '/setting': (context) => const SettingPage(),
       '/about': (context) => AboutScreen(),
       '/playlists': (context) => PlaylistScreen(),
       '/nowplaying': (context) => NowPlaying(),
       '/recent': (context) => RecentlyPlayed(),
       '/downloads': (context) => const Downloads(),
       '/stats': (context) => const Stats(),
     },
     navigatorKey: navigatorKey,
     onGenerateRoute: (RouteSettings settings) {
       if (settings.name == '/player') {
         return PageRouteBuilder(
           opaque: false,
           pageBuilder: (_, __, ___) => const PlayScreen(),
         );
       }
       return HandleRoute.handleRoute(settings.name);
     },
   );
 }
}
```

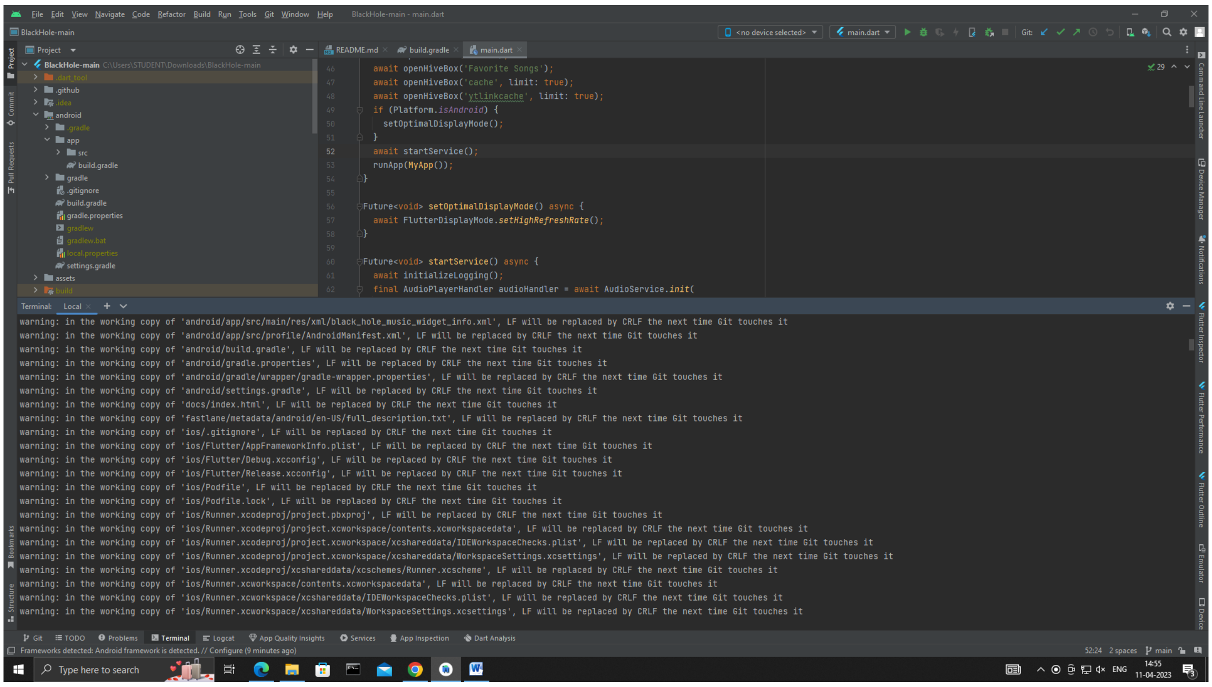
Output :-



```
  await openHiveBox('Favorite Songs');
  await openHiveBox('cache', limit: true);
  await openHiveBox('ytlinkcache', limit: true);
  if (Platform.isAndroid) {
    setOptimalDisplayMode();
  }
  await startService();
  runApp(MyApp());
}

Future<void> setOptimalDisplayMode() async {
  await FlutterDisplayMode.setHighRefreshRate();
}

Future<void> startService() async {
  await initializeLogging();
  final AudioPlayerHandler audioHandler = await AudioService.init(
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\STUDENT\Downloads\BlackHole-main> git init
Initialized empty Git repository in C:/Users/STUDENT/Downloads/BlackHole-main/.git/
PS C:\Users\STUDENT\Downloads\BlackHole-main> git add .
warning: in the working copy of '.gitattributes', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/FUNDING.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/ISSUE_TEMPLATE/bug_report.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/ISSUE_TEMPLATE/custom.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/ISSUE_TEMPLATE/feature_request.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/workflows/flutter.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/workflows/issuenotifier.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.github/workflows/prchecker.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.metadata', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'CODE_OF_CONDUCT.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'CONTRIBUTING.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'LICENSE', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'README.BE.md', LF will be replaced by CRLF the next time Git touches it
```

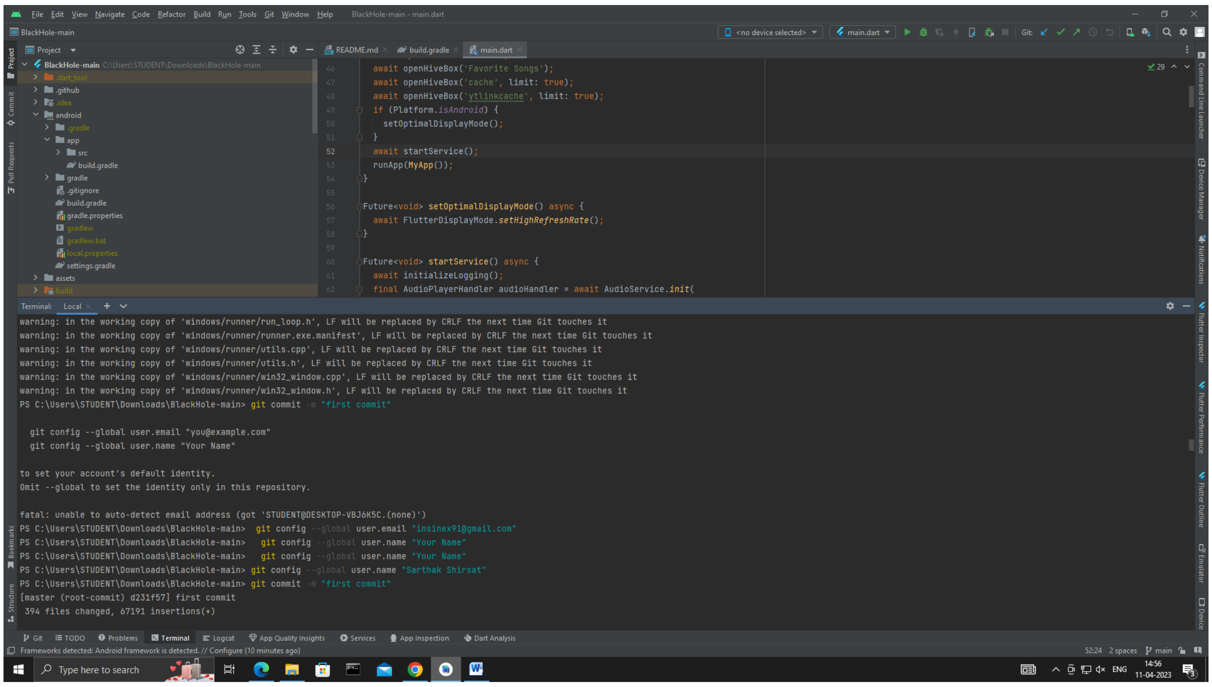
Frameworks detected: Android framework is detected. // Configure (9 minutes ago)



```dart
  await openHiveBox('Favorite Songs');
  await openHiveBox('cache', limit: true);
  await openHiveBox('ytlinkcache', limit: true);
  if (Platform.isAndroid) {
    setOptimalDisplayMode();
  }
  await startService();
  runApp(MyApp());
}

Future<void> setOptimalDisplayMode() async {
  await FlutterDisplayMode.setHighRefreshRate();
}

Future<void> startService() async {
  await initializeLogging();
  final AudioPlayerHandler audioHandler = await AudioService.init(
```

```
warning: in the working copy of 'android/app/src/main/res/xml/black_hole_music_widget_info.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'android/app/src/profile/AndroidManifest.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'android/build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'android/gradle.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'android/gradle/wrapper/gradle-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'android/settings.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'docs/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'fastlane/metadata/android/en-US/full_description.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Flutter/AppFrameworkInfo.plist', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Flutter/Debug.xcconfig', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Flutter/Release.xcconfig', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Podfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Podfile.lock', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcodeproj/project.pbxproj', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcodeproj/project.xcworkspace/contents.xcworkspacedata', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcodeproj/project.xcworkspace/xcshareddata/IDEWorkspaceChecks.plist', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcodeproj/project.xcworkspace/xcshareddata/WorkspaceSettings.xcsettings', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcodeproj/xcshareddata/xcschemes/Runner.xcscheme', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcworkspace/contents.xcworkspacedata', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcworkspace/xcshareddata/IDEWorkspaceChecks.plist', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ios/Runner.xcworkspace/xcshareddata/WorkspaceSettings.xcsettings', LF will be replaced by CRLF the next time Git touches it
```

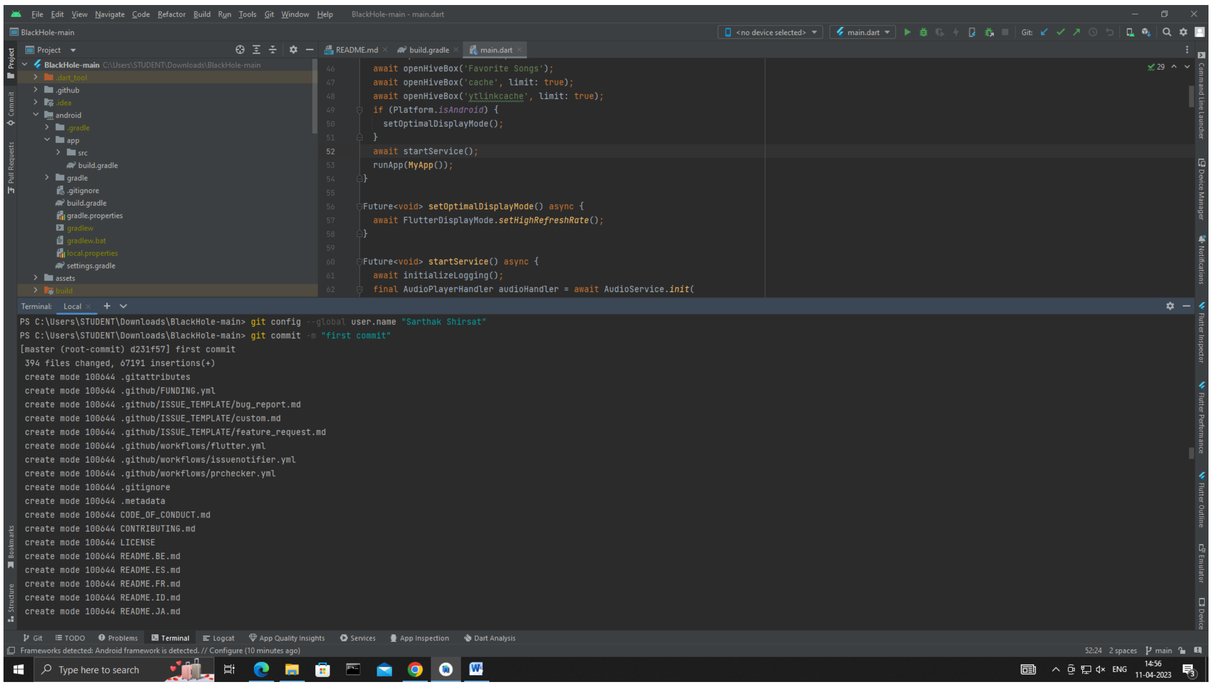
Frameworks detected: Android framework is detected. // Configure (9 minutes ago)



```dart
  await openHiveBox('Favorite Songs');
  await openHiveBox('cache', limit: true);
  await openHiveBox('ytlinkcache', limit: true);
  if (Platform.isAndroid) {
    setOptimalDisplayMode();
  }
  await startService();
  runApp(MyApp());
}

Future<void> setOptimalDisplayMode() async {
  await FlutterDisplayMode.setHighRefreshRate();
}

Future<void> startService() async {
  await initializeLogging();
  final AudioPlayerHandler audioHandler = await AudioService.init(
```

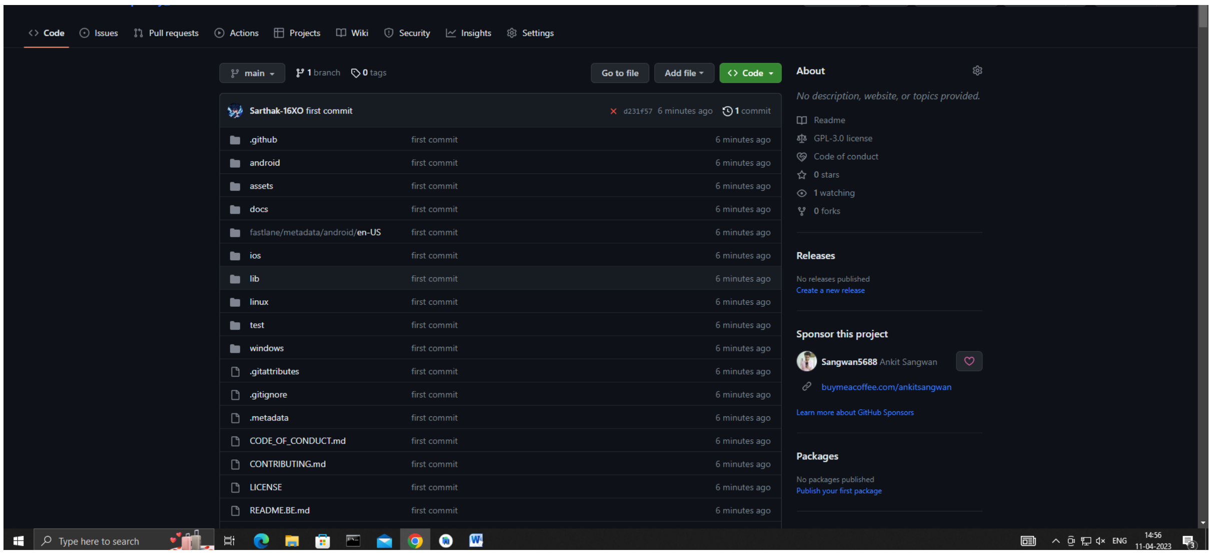
```
warning: in the working copy of 'windows/runner/run_loop.h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/runner/runner.exe.manifest', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/runner/utils.cpp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/runner/utils.h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/runner/win32_window.cpp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'windows/runner/win32_window.h', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\STUDENT\Downloads\BlackHole-main> git commit -m "first commit"

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'STUDENT@DESKTOP-VBJ6K5C.(none)')
PS C:\Users\STUDENT\Downloads\BlackHole-main>  git config --global user.email "insinex91@gmail.com"
PS C:\Users\STUDENT\Downloads\BlackHole-main>   git config --global user.name "Your Name"
PS C:\Users\STUDENT\Downloads\BlackHole-main>   git config --global user.name "Your Name"
PS C:\Users\STUDENT\Downloads\BlackHole-main> git config --global user.name "Sarthak Shirsat"
PS C:\Users\STUDENT\Downloads\BlackHole-main> git commit -m "first commit"
[master (root-commit) d231f57] first commit
 394 files changed, 67191 insertions(+)
```

Frameworks detected: Android framework is detected. // Configure (10 minutes ago)



```
await openHiveBox('Favorite Songs');
await openHiveBox('cache', limit: true);
await openHiveBox('ytlinkcache', limit: true);
if (Platform.isAndroid) {
  setOptimalDisplayMode();
}
await startService();
runApp(MyApp());
}

Future<void> setOptimalDisplayMode() async {
await FlutterDisplayMode.setHighRefreshRate();
}

Future<void> startService() async {
await initializeLogging();
final AudioPlayerHandler audioHandler = await AudioService.init(
```

```
PS C:\Users\STUDENT\Downloads\BlackHole-main> git config --global user.name "Sarthak Shirsat"
PS C:\Users\STUDENT\Downloads\BlackHole-main> git commit -m "first commit"
[master (root-commit) d231f57] first commit
 394 files changed, 67191 insertions(+)
 create mode 100644 .gitattributes
 create mode 100644 .github/FUNDING.yml
 create mode 100644 .github/ISSUE_TEMPLATE/bug_report.md
 create mode 100644 .github/ISSUE_TEMPLATE/custom.md
 create mode 100644 .github/ISSUE_TEMPLATE/feature_request.md
 create mode 100644 .github/workflows/flutter.yml
 create mode 100644 .github/workflows/issuenotifier.yml
 create mode 100644 .github/workflows/prchecker.yml
 create mode 100644 .gitignore
 create mode 100644 .metadata
 create mode 100644 CODE_OF_CONDUCT.md
 create mode 100644 CONTRIBUTING.md
 create mode 100644 LICENSE
 create mode 100644 README.BE.md
 create mode 100644 README.ES.md
 create mode 100644 README.FR.md
 create mode 100644 README.ID.md
 create mode 100644 README.JA.md
```



Sarthak-16XO first commit d231f57 6 minutes ago 1 commit

.github	first commit	6 minutes ago
android	first commit	6 minutes ago
assets	first commit	6 minutes ago
docs	first commit	6 minutes ago
fastlane/metadata/android/en-US	first commit	6 minutes ago
ios	first commit	6 minutes ago
lib	first commit	6 minutes ago
linux	first commit	6 minutes ago
test	first commit	6 minutes ago
windows	first commit	6 minutes ago
.gitattributes	first commit	6 minutes ago
.gitignore	first commit	6 minutes ago
.metadata	first commit	6 minutes ago
CODE_OF_CONDUCT.md	first commit	6 minutes ago
CONTRIBUTING.md	first commit	6 minutes ago
LICENSE	first commit	6 minutes ago
README.BE.md	first commit	6 minutes ago

About

No description, website, or topics provided.

Readme
GPL-3.0 license
Code of conduct
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Sponsor this project

Sangwan5688 Ankit Sangwan
buymeacoffee.com/ankitsangwan
Learn more about GitHub Sponsors

Packages

No packages published
Publish your first package

Github Link : https://github.com/yash1501-arch/blogapp

Conclusion : Hence we have successfully tested and deployed production ready Flutter App on Android platform.