How To Set Up and Run the KMP Project

Table of contents

Table of contents **1**
Important Note **2**
Troubleshooting **2**
System Requirements **3**
- Minimum System Requirements 3
- Explanation 3
- Latest Xcode Compatible Devices 4

Setup KMP Environment **5**
IDEs Involved **5**
Install Fleet **6**
Open The Project In Fleet **8**
Run The Project In Fleet **10**
Instructions for Connecting to a Local Server **12**

Important Note

This document provides instructions for running the Kotlin Multiplatform (KMP) project from the codebase.

For instructions on installing the Android application and iOS application, please refer to the file titled **"How To Install the Android App and the iOS App.pdf"** located in the root directory of the codebase.

Troubleshooting

Although Kotlin Multiplatform (KMP) is a promising technology, it's still in its early stages. Even the IDE named Fleet recommended by JetBrains, the company behind KMP, is still in preview mode. This indicates that various challenges may occur during the development environment setup.

If issues occur during setup, please contact the author of the project to schedule a live debugging session.

However, the instructions provided here are designed to be comprehensive to ensure the setup goes smoothly without additional assistance. For example, this document explains the minimum system requirements in depth and highlights the usage of the KMP setup assistant, **Project Preflight**, to address common issues.

System Requirements

Minimum System Requirements

Devices listed at the following URL are sufficient for this project as of December 2024: https://support.apple.com/en-us/105113 (Accessed: 1 Dec 2024).

Explanation

The KMP development environment integrates both iOS and Android development tools. While the Android development environment is relatively flexible, iOS development requires macOS. For iOS, it is recommended to download and use the latest version of Xcode, as issues in older versions are typically resolved in the latest release.

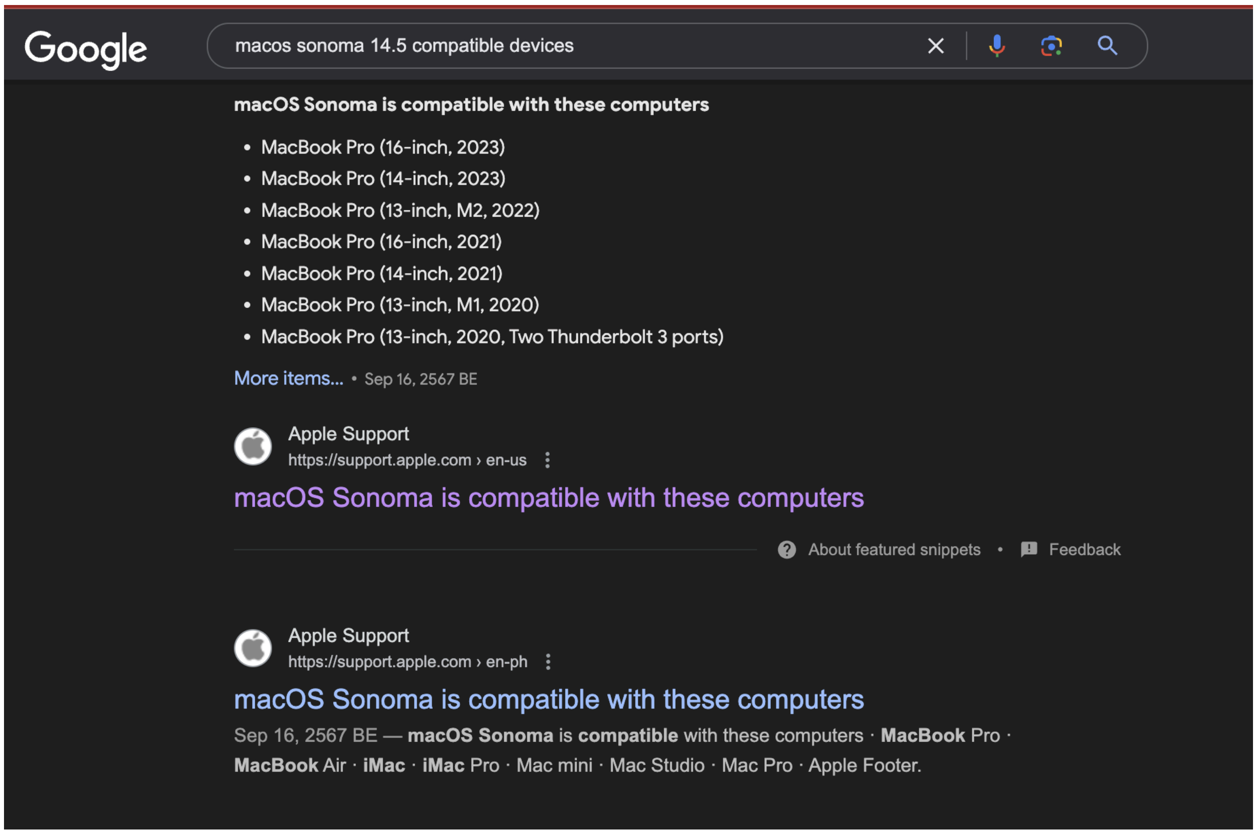
At the time of this project, Xcode 16.1 was the latest version. This version is compatible with all devices listed at the following URL: https://support.apple.com/en-us/105113 (Accessed: 1 Dec 2024).

If Xcode 16.1 is no longer the latest version, please refer to the section titled "Latest Xcode Compatible Devices" for guidance on identifying devices compatible with the latest Xcode version.

Latest Xcode Compatible Devices

1. To determine which macOS version is required for the latest Xcode, refer to the "Minimum requirements and supported SDKs" section at the following URL: https://developer.apple.com/support/xcode/ (Accessed on: 1 Dec 2024).

2. Copy the macOS version and search for "[macOS version] compatible devices." For example, a search for "macOS Sonoma 14.5 compatible devices" might yield results similar to those shown below:



macOS Sonoma is compatible with these computers

- MacBook Pro (16-inch, 2023)
- MacBook Pro (14-inch, 2023)
- MacBook Pro (13-inch, M2, 2022)
- MacBook Pro (16-inch, 2021)
- MacBook Pro (14-inch, 2021)
- MacBook Pro (13-inch, M1, 2020)
- MacBook Pro (13-inch, 2020, Two Thunderbolt 3 ports)

More items... • Sep 16, 2567 BE

Apple Support
https://support.apple.com › en-us
macOS Sonoma is compatible with these computers

About featured snippets • Feedback

Apple Support
https://support.apple.com › en-ph
macOS Sonoma is compatible with these computers
Sep 16, 2567 BE — **macOS Sonoma** is **compatible** with these computers · **MacBook** Pro · **MacBook** Air · **iMac** · **iMac** Pro · Mac mini · Mac Studio · Mac Pro · Apple Footer.

Figure 1. Search Result For "macos sonoma 14.5 compatible devices"

3. The first link in the example above directs to compatible devices for macOS Sonoma 14.5. The URL for this example is as follows: https://support.apple.com/en-us/105113 (Accessed on: 1 Dec 2024).

Setup KMP Environment

To set up the Kotlin Multiplatform (KMP) environment, refer to the following article from JetBrains: https://www.jetbrains.com/help/kotlin-multiplatform-dev/multiplatform-setup.html (Accessed on: 1 Dec 2024).

The setup process includes the following components:

- Java development environment
- Android development environment
- iOS development environment
- Kotlin Multiplatform environment setup
- Installation of the KMP plugin in Android Studio

IDEs Involved

Kotlin Multiplatform (KMP) development involves the use of three integrated development environments (IDEs):

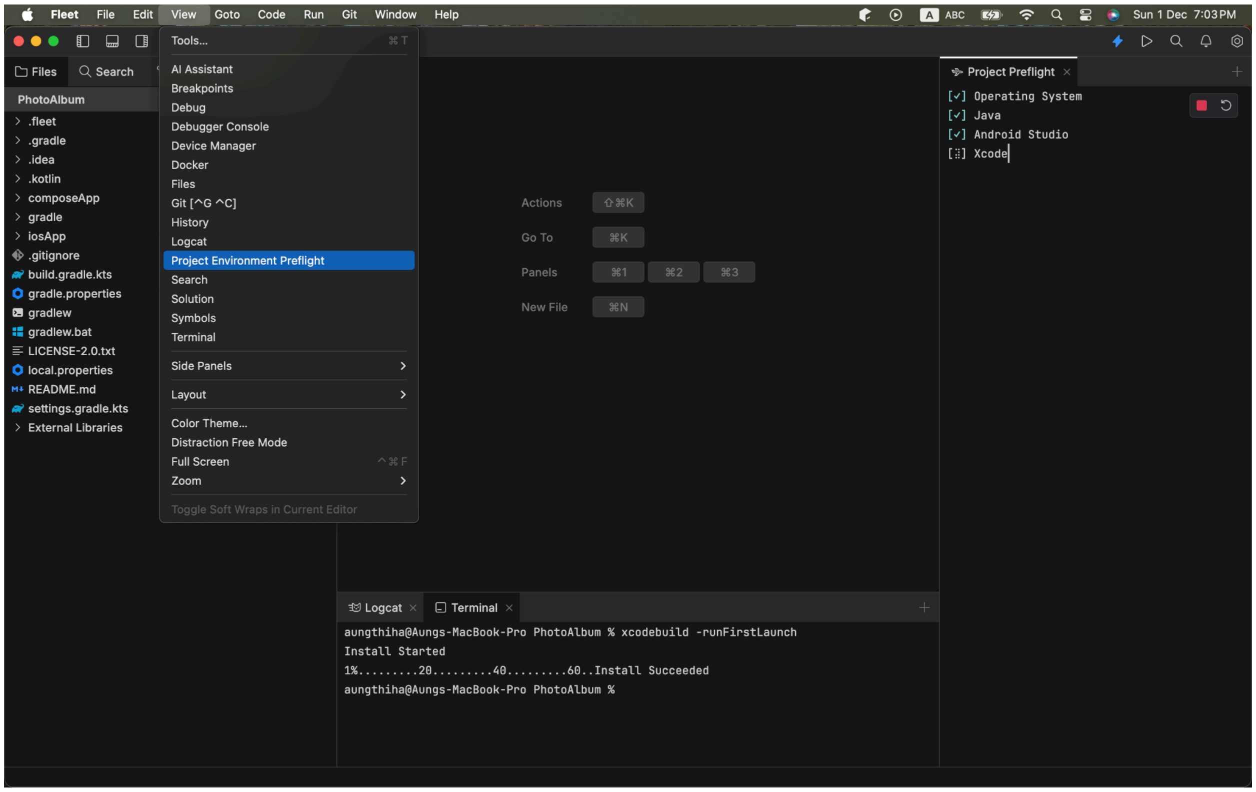
1. Android Studio for Android development
2. Xcode for iOS development
3. Fleet for Kotlin Multiplatform development

While Fleet serves as the primary IDE for development, Xcode and Android Studio are essential for compiling and releasing platform-specific applications.

Install Fleet

Fleet is the recommended IDE for Kotlin Multiplatform (KMP) development. For detailed installation instructions, refer to the following URL: https://www.jetbrains.com/help/kotlin-multiplatform-dev/fleet.html#get-started-with-fleet (Accessed on: 1 Dec 2024).

Fleet includes the **Project Preflight** assistant, which helps identify and troubleshoot setup issues. This assistant appears in the right pane of the Fleet interface. Figure 2 demonstrates how to open the **Project Preflight** window to begin troubleshooting potential setup problems.

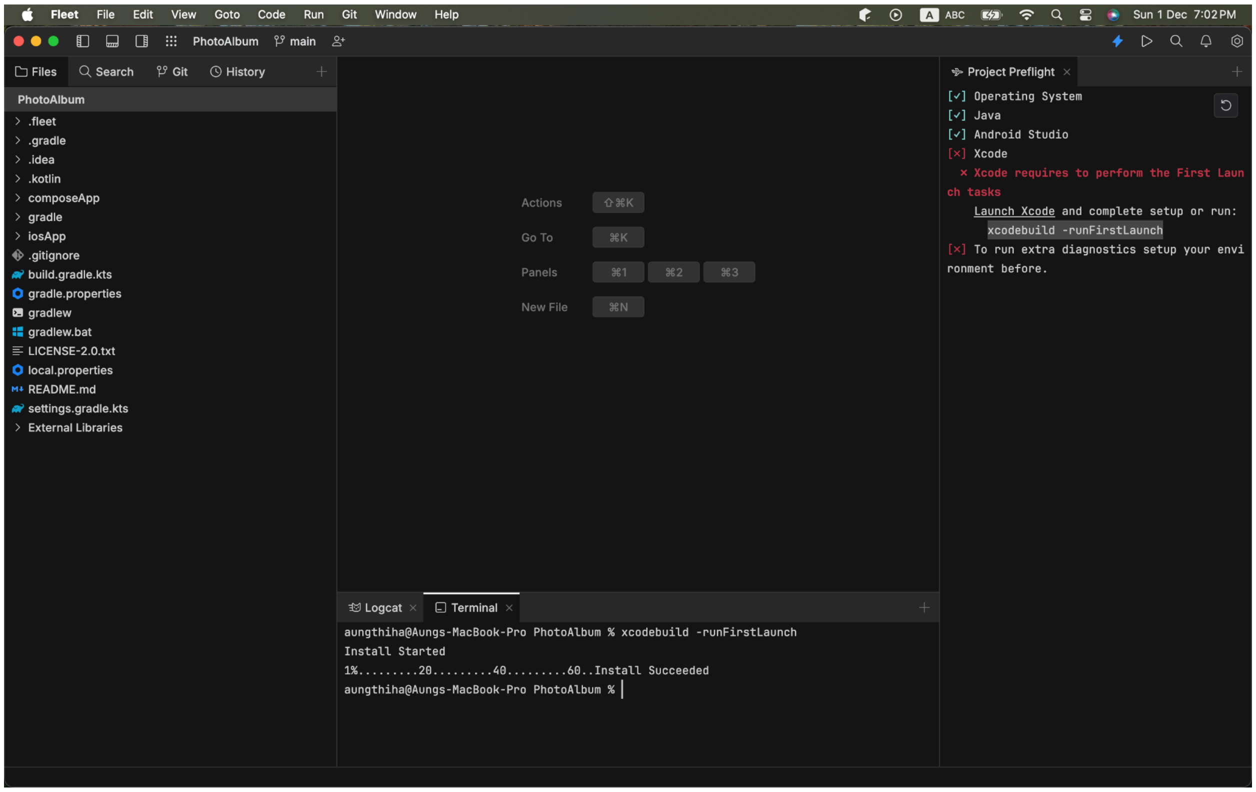


```
[✓] Operating System
[✓] Java
[✓] Android Studio
[⠿] Xcode
```

```
aungthiha@Aungs-MacBook-Pro PhotoAlbum % xcodebuild -runFirstLaunch
Install Started
1%.........20.........40.........60..Install Succeeded
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 2. Opening Project Environment Preflight

After opening, the **Project Preflight** pane provides insights into setup issues. Figure 3 shows an example of a common setup error detected in the **Project Preflight** pane, caused by Xcode, an IDE required for iOS development.



```
[✓] Operating System
[✓] Java
[✓] Android Studio
[×] Xcode
  × Xcode requires to perform the First Launch tasks
    Launch Xcode and complete setup or run:
      xcodebuild -runFirstLaunch
[×] To run extra diagnostics setup your environment before.
```

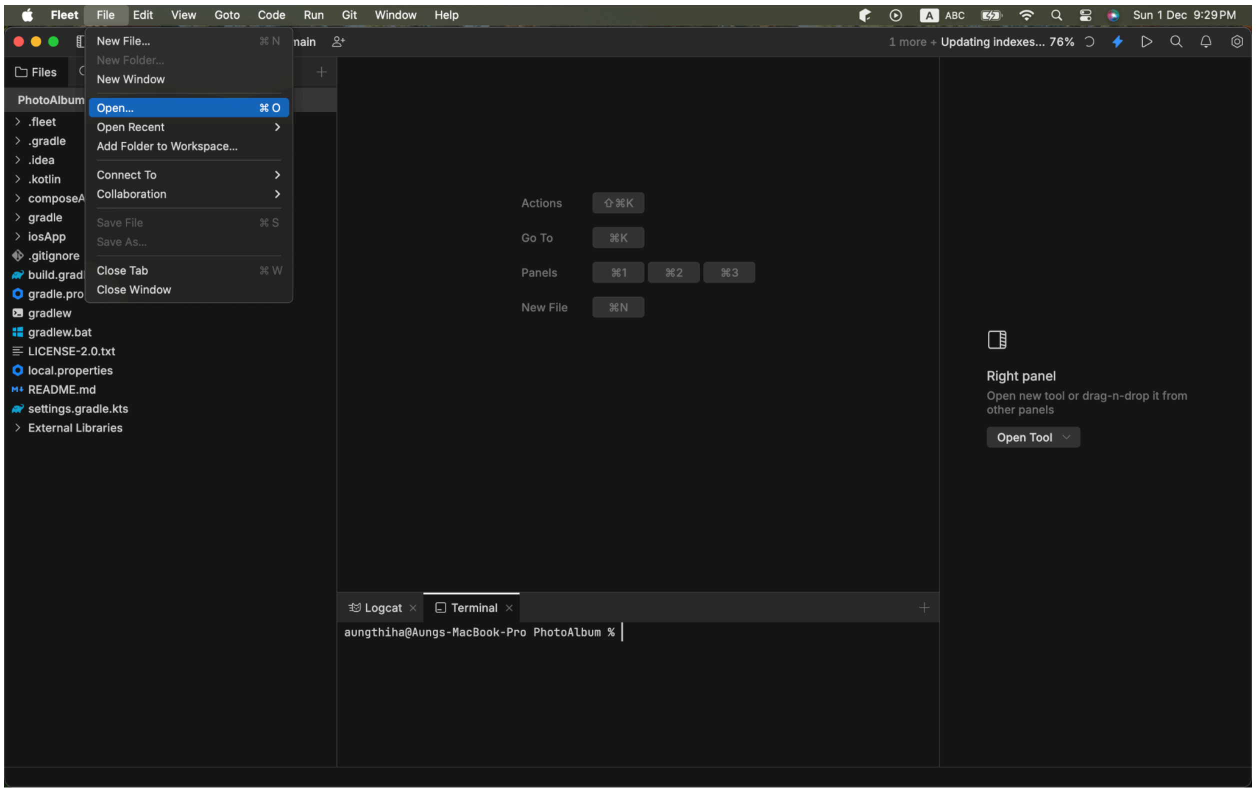
```
aungthiha@Aungs-MacBook-Pro PhotoAlbum % xcodebuild -runFirstLaunch
Install Started
1%.........20.........40.........60..Install Succeeded
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 3. Example of a KMP Setup Error

Open The Project In Fleet

To open the project in Fleet, follow these steps:

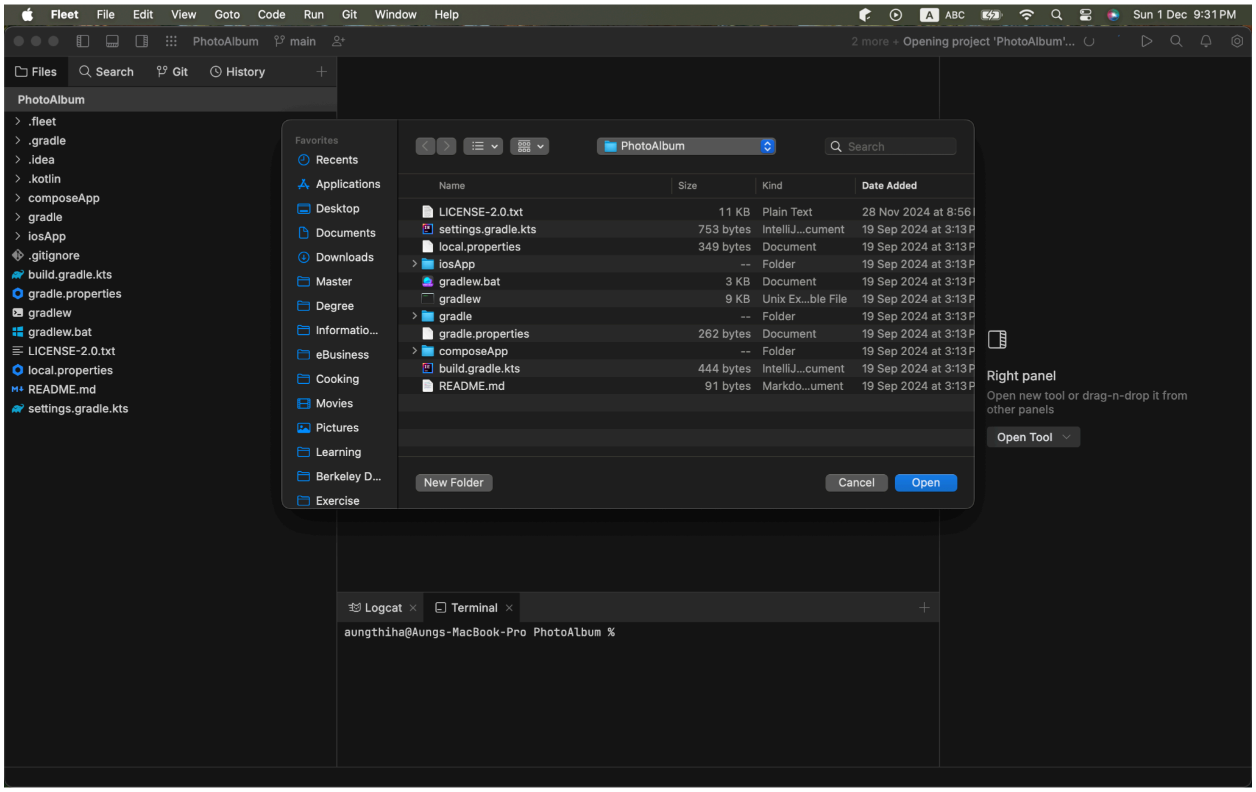
1. Click **File** and then select **Open** as shown in Figure 4.



```
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 4. Open a Project in Fleet

2. Navigate to the directory containing the project files, and then click **Open** as illustrated in Figure 5.



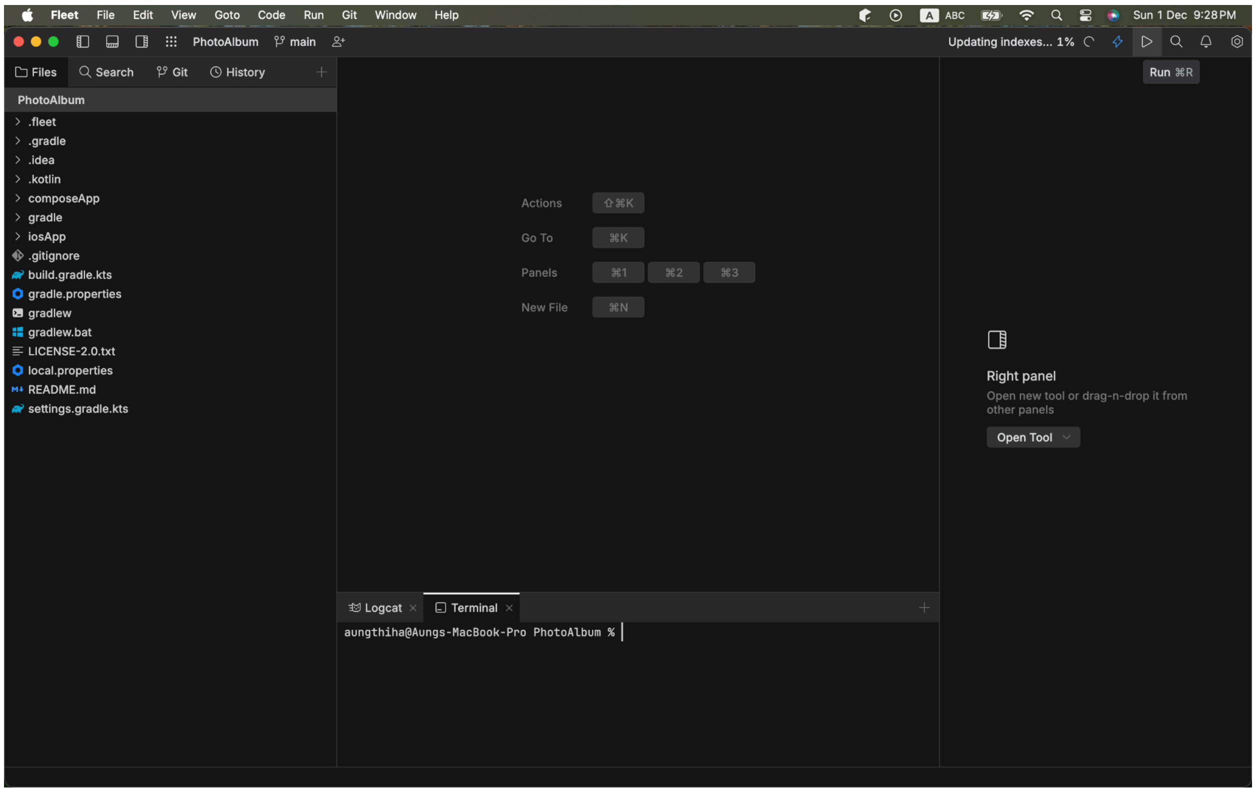
Name	Size	Kind	Date Added
LICENSE-2.0.txt	11 KB	Plain Text	28 Nov 2024 at 8:56
settings.gradle.kts	753 bytes	IntelliJ...cument	19 Sep 2024 at 3:13 P
local.properties	349 bytes	Document	19 Sep 2024 at 3:13 P
iosApp	--	Folder	19 Sep 2024 at 3:13 P
gradlew.bat	3 KB	Document	19 Sep 2024 at 3:13 P
gradlew	9 KB	Unix Ex...ble File	19 Sep 2024 at 3:13 P
gradle	--	Folder	19 Sep 2024 at 3:13 P
gradle.properties	262 bytes	Document	19 Sep 2024 at 3:13 P
composeApp	--	Folder	19 Sep 2024 at 3:13 P
build.gradle.kts	444 bytes	IntelliJ...cument	19 Sep 2024 at 3:13 P
README.md	91 bytes	Markdo...ument	19 Sep 2024 at 3:13 P

```
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 5. Locating and Opening the Project Directory

Run The Project In Fleet

1. **Indexing Files**
   When the project is opened in Fleet for the first time, the IDE may require up to 10 minutes to index all files in the project. The indexing progress is displayed in the top-right corner, as shown in Figure 6. Once indexing is complete, the project can be run by clicking the arrow-like triangle icon in the top-right corner.

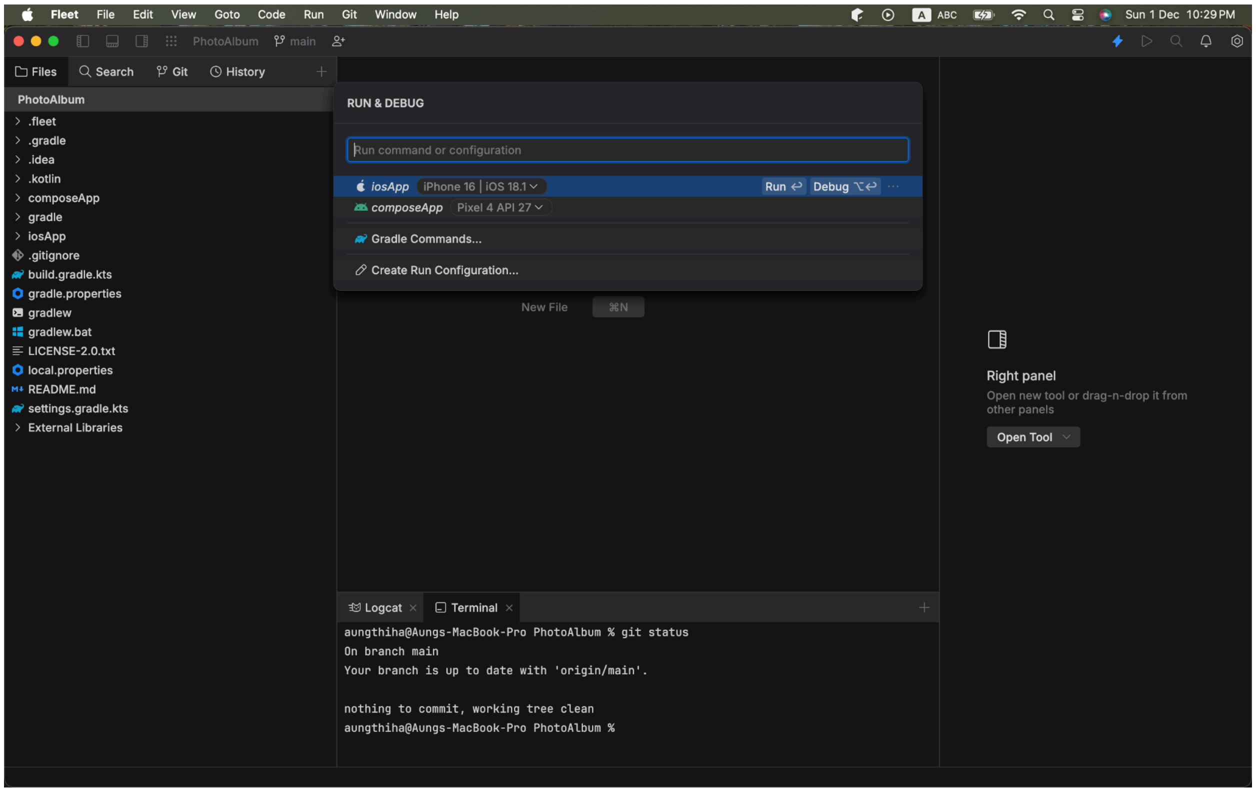


```
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 6. Indexing Files in Fleet for the First Time

2. **Selecting the Platform**

   After indexing is complete and the arrow-like triangle icon in the top-right corner is clicked, select the desired platform for the build. As shown in Figure 7, choose **iosApp** to build the iOS application or **composeApp** to build the Android application.



```
aungthiha@Aungs-MacBook-Pro PhotoAlbum % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 7. Selecting a Platform and Device

3. **Selecting a Device**

   The dropdown toggles beside **iosApp** and **composeApp** allow for device selection. If no devices are connected, the dropdown will display available Android emulators and iOS simulators.

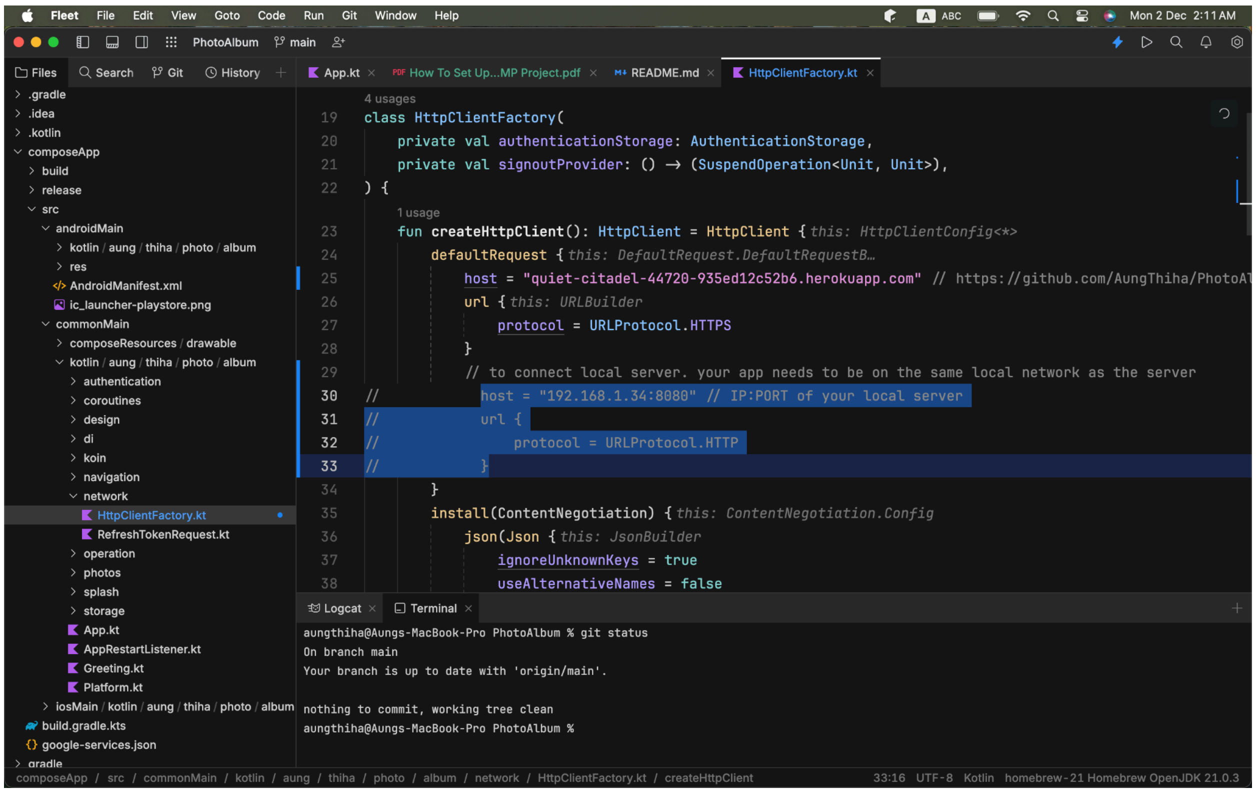
4. **Running the Application**

   Selecting **iosApp** or **composeApp** initiates the build process, which may take several minutes. Once completed, the application is launched on the chosen device.

Instructions for Connecting to a Local Server

Important Note: The server the app is configured to connect to supports secure HTTPS connections. If the configuration is changed to connect to a different server, photos may not appear in the mobile application because Coil, the image loading library, does not support unsecured HTTP connections without custom configuration. Instructions for connecting to a local server are included for cases where switching to a different server is necessary for evaluation.

Line 25 in Figure 8 shows the app is configured to connect to the server hosted on Heroku. The server is the backend app described in Appendix G in the final report. The file shown in Figure 8 is **HttpClientFactory.kt**, located in the **commonMain** folder within the KMP project (**composeApp/src/commonMain/kotlin/aung/thiha/photo/album/network/HttpClientFactory.kt**).


Fleet File Edit View Goto Code Run Git Window Help
Mon 2 Dec 2:11 AM
PhotoAlbum
main
Files
Search
Git
History
App.kt
How To Set Up...MP Project.pdf
README.md
HttpClientFactory.kt
.gradle
.idea
.kotlin
composeApp
build
release
src
androidMain
kotlin / aung / thiha / photo / album
res
AndroidManifest.xml
ic_launcher-playstore.png
commonMain
composeResources / drawable
kotlin / aung / thiha / photo / album
authentication
coroutines
design
di
koin
navigation
network
HttpClientFactory.kt
RefreshTokenRequest.kt
operation
photos
splash
storage
App.kt
AppRestartListener.kt
Greeting.kt
Platform.kt
iosMain / kotlin / aung / thiha / photo / album
build.gradle.kts
google-services.json
gradle
Logcat
Terminal
composeApp / src / commonMain / kotlin / aung / thiha / photo / album / network / HttpClientFactory.kt / createHttpClient
33:16 UTF-8 Kotlin homebrew-21 Homebrew OpenJDK 21.0.3

```
      4 usages
19    class HttpClientFactory(
20        private val authenticationStorage: AuthenticationStorage,
21        private val signoutProvider: () → (SuspendOperation<Unit, Unit>),
22    ) {
          1 usage
23        fun createHttpClient(): HttpClient = HttpClient { this: HttpClientConfig<*>
24            defaultRequest { this: DefaultRequest.DefaultRequestB…
25                host = "quiet-citadel-44720-935ed12c52b6.herokuapp.com" // https://github.com/AungThiha/PhotoA
26                url { this: URLBuilder
27                    protocol = URLProtocol.HTTPS
28                }
29                // to connect local server. your app needs to be on the same local network as the server
30    //            host = "192.168.1.34:8080" // IP:PORT of your local server
31    //            url {
32    //                protocol = URLProtocol.HTTP
33    //            }
34            }
35            install(ContentNegotiation) { this: ContentNegotiation.Config
36                json(Json { this: JsonBuilder
37                    ignoreUnknownKeys = true
38                    useAlternativeNames = false
```

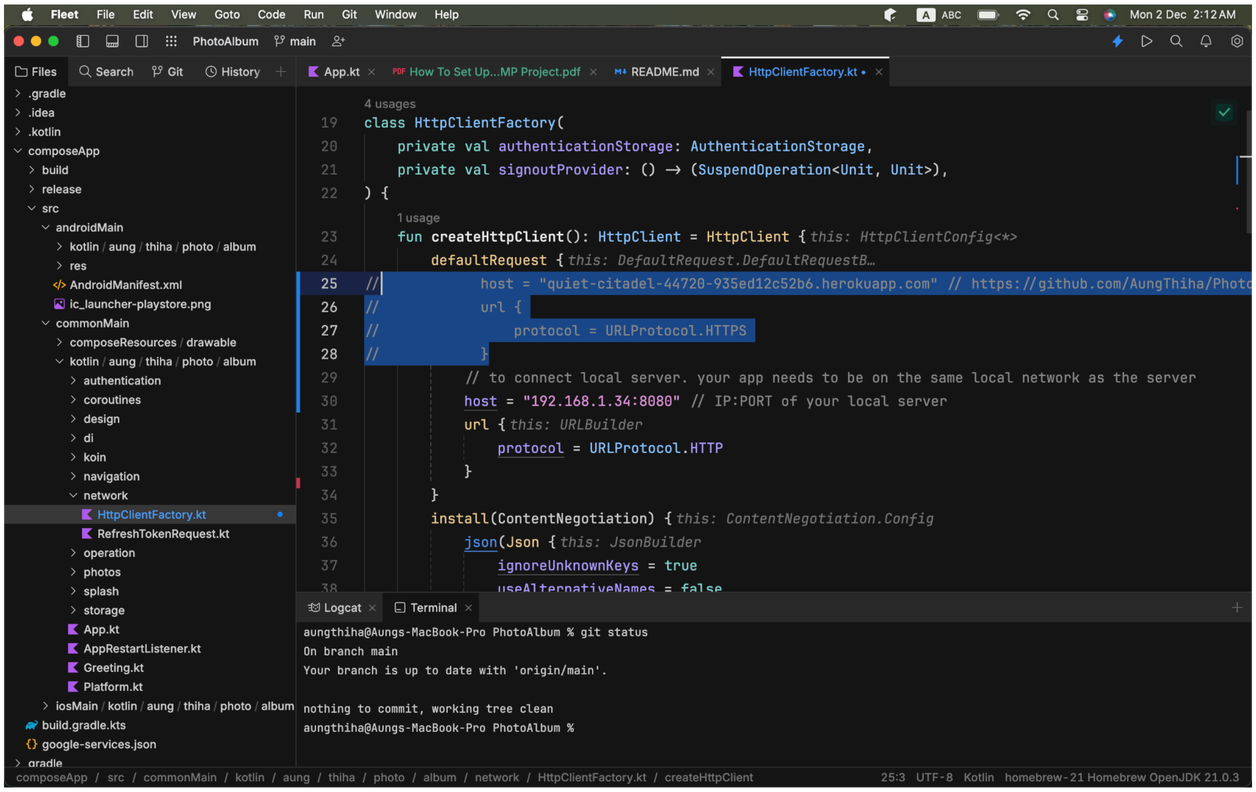
```
aungthiha@Aungs-MacBook-Pro PhotoAlbum % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 8. Connecting to the Server on Heroku

To connect the application to a local server, the following changes are required:

1. Comment out the code from lines 25 to 28.
2. Uncomment the code from lines 30 to 33.
3. Update the host value at line 30 to the `[IP]:[PORT]` of your local server.
4. Ensure that both the mobile application and the local server are running on the same local network.


Fleet File Edit View Goto Code Run Git Window Help
Mon 2 Dec 2:12 AM
PhotoAlbum
main
Files Search Git History
App.kt
How To Set Up...MP Project.pdf
README.md
HttpClientFactory.kt
.gradle
.idea
.kotlin
composeApp
build
release
src
androidMain
kotlin / aung / thiha / photo / album
res
AndroidManifest.xml
ic_launcher-playstore.png
commonMain
composeResources / drawable
kotlin / aung / thiha / photo / album
authentication
coroutines
design
di
koin
navigation
network
HttpClientFactory.kt
RefreshTokenRequest.kt
operation
photos
splash
storage
App.kt
AppRestartListener.kt
Greeting.kt
Platform.kt
iosMain / kotlin / aung / thiha / photo / album
build.gradle.kts
google-services.json
gradle
Logcat
Terminal
composeApp / src / commonMain / kotlin / aung / thiha / photo / album / network / HttpClientFactory.kt / createHttpClient
25:3 UTF-8 Kotlin homebrew-21 Homebrew OpenJDK 21.0.3

```kotlin
   4 usages
19 class HttpClientFactory(
20     private val authenticationStorage: AuthenticationStorage,
21     private val signoutProvider: () -> (SuspendOperation<Unit, Unit>),
22 ) {
       1 usage
23     fun createHttpClient(): HttpClient = HttpClient { this: HttpClientConfig<*>
24         defaultRequest { this: DefaultRequest.DefaultRequestB…
25 //              host = "quiet-citadel-44720-935ed12c52b6.herokuapp.com"  // https://github.com/AungThiha/Photo
26 //              url {
27 //                  protocol = URLProtocol.HTTPS
28 //              }
29             // to connect local server. your app needs to be on the same local network as the server
30             host = "192.168.1.34:8080" // IP:PORT of your local server
31             url { this: URLBuilder
32                 protocol = URLProtocol.HTTP
33             }
34         }
35         install(ContentNegotiation) { this: ContentNegotiation.Config
36             json(Json { this: JsonBuilder
37                 ignoreUnknownKeys = true
38                 useAlternativeNames = false
```

```
aungthiha@Aungs-MacBook-Pro PhotoAlbum % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
aungthiha@Aungs-MacBook-Pro PhotoAlbum %
```

Figure 9. Connecting to Local Server