NOAH APP

Table Of Contents

Table Of Contents 1
Prerequisite 2
Setup Project 2
Change App Icon 2
Archive and Export IPA Files to App Store Connect 3
Codebase 7
For Future Enhancement 9

Prerequisite

- Clone Noah Project into your local
- XCode
- Apple Developer account (Must register under company / legit organization not personal due to App Store Guideline 3.1.5 Cryptocurrencies (i))

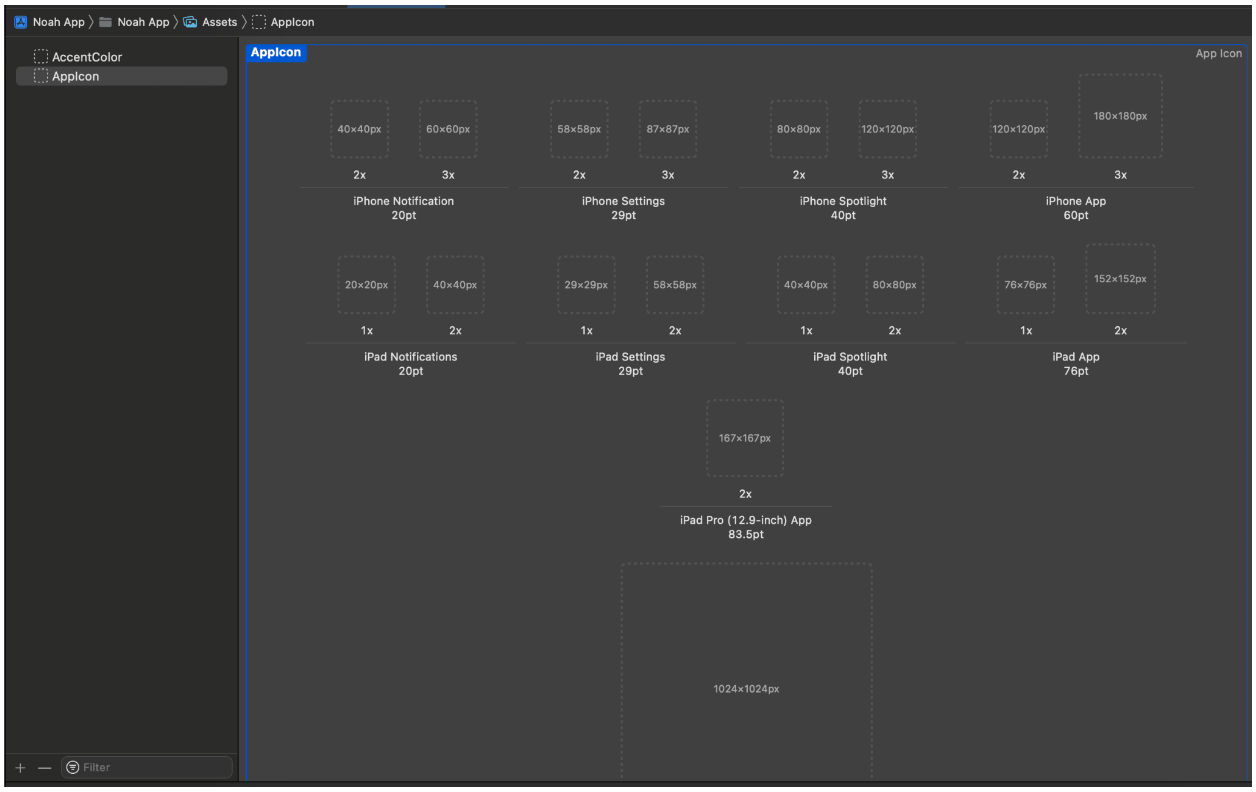
Setup Project

Open Noah App Project with XCode. In XCode, click **XCode** > **Settings…** > **Account Tab** and add Apple ID with AppStore Connect Team.

Once done, go to **Noah App** > **Noah Apps under Targets** > **Signing and capabilities** > Tick **automatically manage signing** and choose **your team**. Now, you can start to run the app

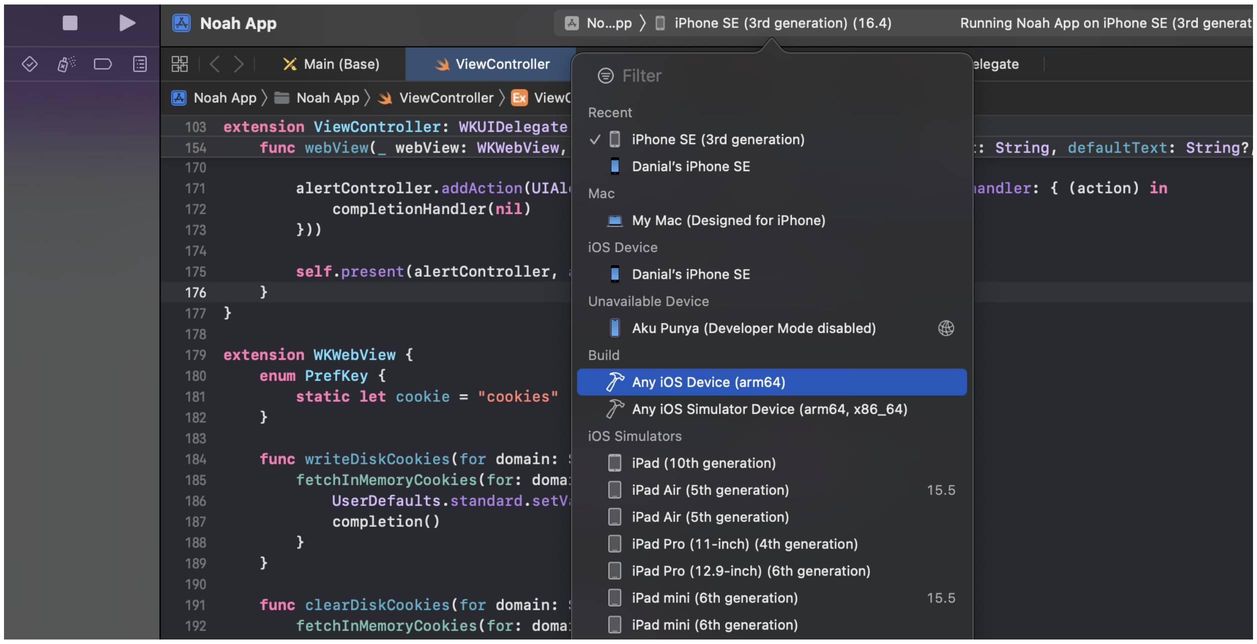
Change App Icon

Click **Assets** > **AppIcon**. Upload Icon based on the size given:



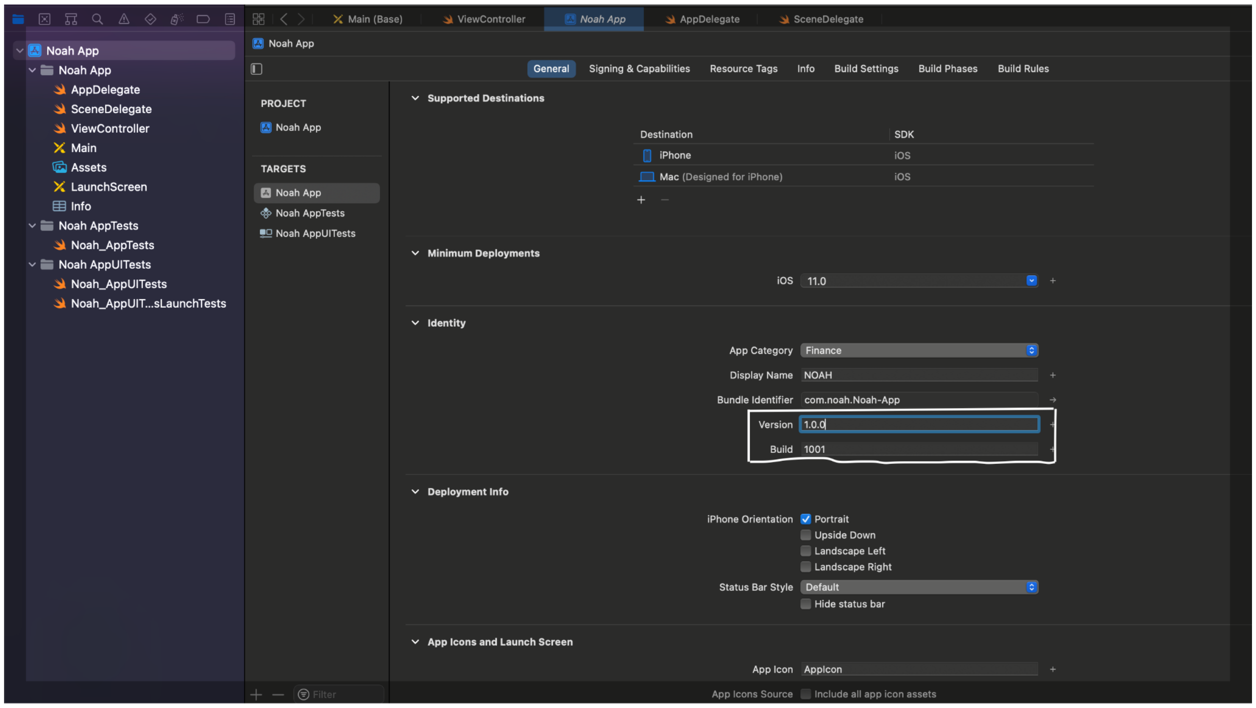
Archive and Export IPA Files to App Store Connect

To start off, in Xcode, at the top, you need to set your simulator to Any iOS device.


Noah App
No...pp 〉 iPhone SE (3rd generation) (16.4)
Running Noah App on iPhone SE (3rd generat
Main (Base)
ViewController
elegate
Noah App 〉 Noah App 〉 ViewController 〉 ViewC
Filter
Recent
✓ iPhone SE (3rd generation)
Danial's iPhone SE
Mac
My Mac (Designed for iPhone)
iOS Device
Danial's iPhone SE
Unavailable Device
Aku Punya (Developer Mode disabled)
Build
Any iOS Device (arm64)
Any iOS Simulator Device (arm64, x86_64)
iOS Simulators
iPad (10th generation)
iPad Air (5th generation) 15.5
iPad Air (5th generation)
iPad Pro (11-inch) (4th generation)
iPad Pro (12.9-inch) (6th generation)
iPad mini (6th generation) 15.5
iPad mini (6th generation)

```
extension ViewController: WKUIDelegate
    func webView(_ webView: WKWebView,                              : String, defaultText: String?

        alertController.addAction(UIAl                              andler: { (action) in
            completionHandler(nil)
        }))

        self.present(alertController,
    }
}

extension WKWebView {
    enum PrefKey {
        static let cookie = "cookies"
    }

    func writeDiskCookies(for domain:
        fetchInMemoryCookies(for: doma
            UserDefaults.standard.setV
            completion()
        }
    }

    func clearDiskCookies(for domain:
        fetchInMemoryCookies(for: doma
```

Navigate to your project's settings. Under iOS (or the target you want to build your app for) > Identity, you'll want to increment the Build number. For example, if the Build number was 1, you'll want to set it to 2.



Supported Destinations

Destination	SDK
iPhone	iOS
Mac (Designed for iPhone)	iOS

Minimum Deployments

iOS 11.0

Identity

App Category: Finance
Display Name: NOAH
Bundle Identifier: com.noah.Noah-App
Version: 1.0.0
Build: 1001

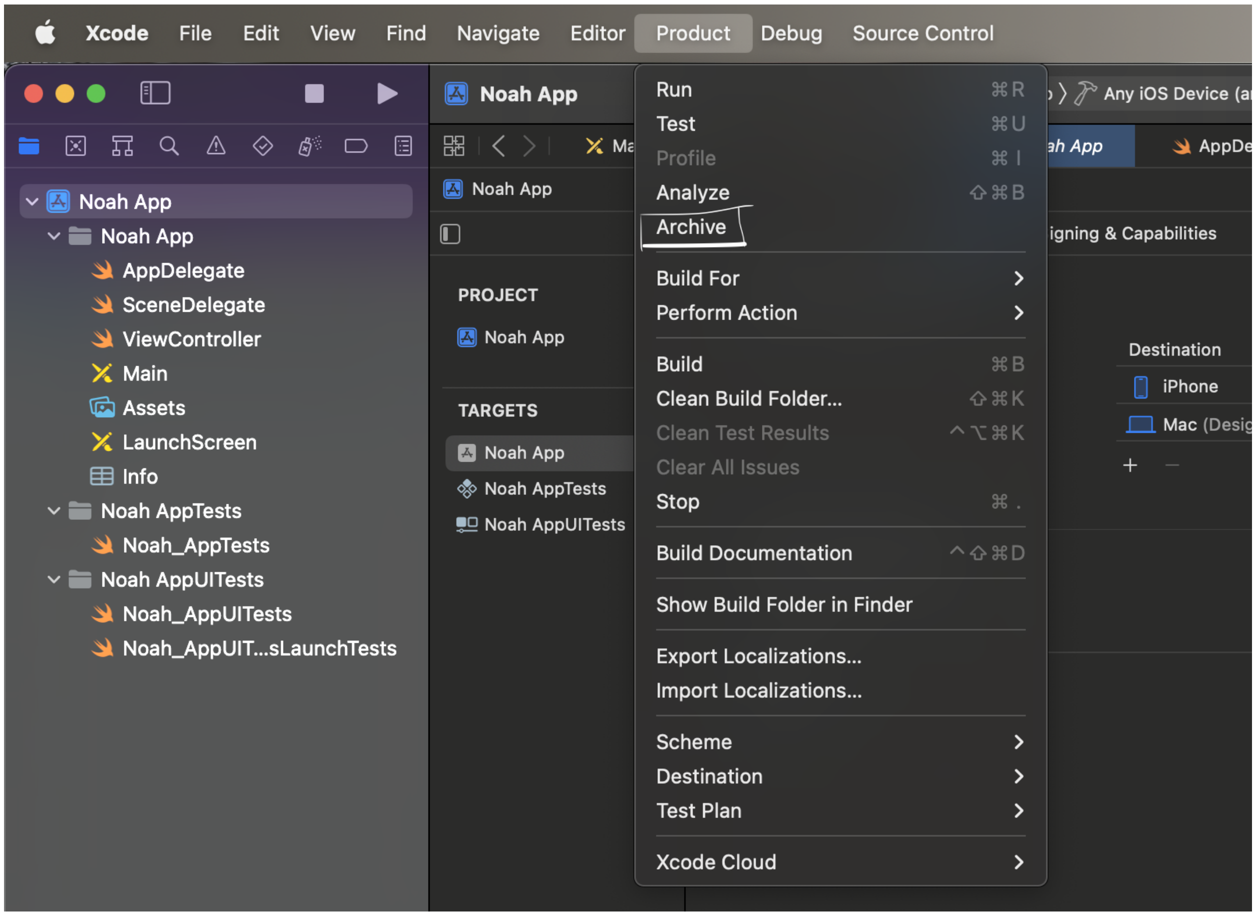
Deployment Info

iPhone Orientation: [✓ Portrait] Upside Down, Landscape Left, Landscape Right
Status Bar Style: Default
Hide status bar

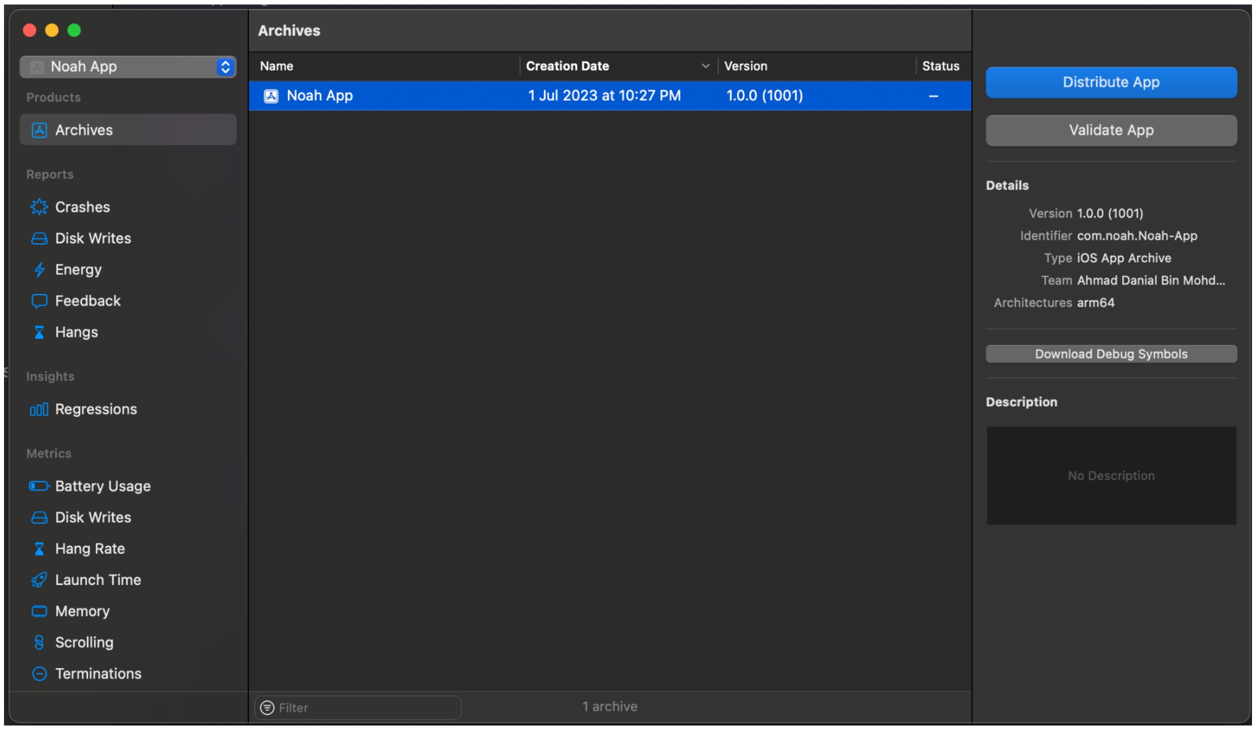
App Icons and Launch Screen

App Icon: AppIcon
App Icons Source: Include all app icon assets

Then, in the top menu, under Product, click on Archive.



After the archive is completed, a window will popup and you'll be able to Validate App. Once it's validated, you can Distribute App and it will upload it to App Store Connect, where you'll be able to share it with Beta testers or upload it directly to the App Store.



Name	Creation Date	Version	Status
Noah App	1 Jul 2023 at 10:27 PM	1.0.0 (1001)	–

Details

Version	1.0.0 (1001)
Identifier	com.noah.Noah-App
Type	iOS App Archive
Team	Ahmad Danial Bin Mohd...
Architectures	arm64

Description

No Description

Codebase

Since everything is run on webview. We can add features without needing to touch native code or go through the app submission.

Here are some important code snippets:

```
func writeDiskCookies(for domain: String, completion: @escaping () ->
()) {
        fetchInMemoryCookies(for: domain) { data in
            UserDefaults.standard.setValue(data, forKey:
PrefKey.cookie)
            completion()
        }
    }
```

Above code is to set user cookies during login and store the data into local storage to keep the user session.

```
func clearDiskCookies(for domain: String) {
        fetchInMemoryCookies(for: domain) { data in
            UserDefaults.standard.removeObject(forKey:
PrefKey.cookie)
        }
    }
```

Above code is to delete user cookies during logout and redirect users to the login screen.

```
func loadDiskCookies(for domain: String, completion: @escaping () ->
()) {
        if let diskCookie = UserDefaults.standard.dictionary(forKey:
(PrefKey.cookie)) {
            fetchInMemoryCookies(for: domain) { freshCookie in

                let mergedCookie = diskCookie.merging(freshCookie) {
(_, new) in new }

                for (_, cookieConfig) in mergedCookie {
                    let cookie = cookieConfig as! Dictionary<String,
Any>

                    var expire : Any? = nil

                    if let expireTime = cookie["Expires"] as? Double
{
                        expire = Date(timeIntervalSinceNow:
expireTime)
                    }

                    let newCookie = HTTPCookie(properties: [
                        .domain: cookie["Domain"] as Any,
                        .path: cookie["Path"] as Any,
                        .name: cookie["Name"] as Any,
                        .value: cookie["Value"] as Any,
                        .secure: cookie["Secure"] as Any,
                        .expires: expire as Any
                    ])


self.configuration.websiteDataStore.httpCookieStore.setCookie(newCook
ie!)
                }

                completion()
            }

        } else {
            completion()
        }
    }
```

Above code is to load user cookies and inject into webview.

For Future Enhancement

if want to add communication between web and native, we can use **WKScriptMessageHandler**

```
self.webView.configuration.userContentController.add(<#T##scriptMessa
geHandler: WKScriptMessageHandler##WKScriptMessageHandler#>, name:
<#T##String#>)
```

Above snippet is to set a listener for the Javascript Callback handler.

```
func userContentController(_ userContentController:
WKUserContentController, didReceive message: WKScriptMessage) {
        //add process here
    }
```

Above snippet is to process when an app receives a Javascript Callback handler.

```
self.webView.evaluateJavaScript("javascript:
acceptDeviceTokenForNotification('\(deviceToken)')")
```

Above snippet is to pass data from the app to javascript (web).