| Activity No. 10 Building with Expo | |
|---|---|
| **Course Code:** CPE 026 | **Program:** Computer Engineering |
| **Course Title:** Emerging Technologies 3 | **Date Performed:** 11/27/2024 |
| **Section:** CPE41S8 | **Date Submitted:** 11/30/2024 |
| **Name:**<br>Joshua L. Alferos<br>Christian Ed B. Efa | **Instructor:** Engr. Roman Richard |

### 1. Objective(s)

This activity aims to introduce Expo tool for students to deploy their applications for iOS or A

### 2. Intended Learning Outcomes

After this module, the students should be able to:
- Demonstrate the deployment of an application built in previous modules.

### 3. Discussion

Discussion contents are found in the module.

📄 8.1 How to Start Building and Publishing an Application

📄 8.2 Pros and Cons of Building with Expo

🔗 8.3 App Stores Best Practices

EAS Build allows you to build a ready-to-submit binary of your app for the Google Play Store or Apple App Store. The contents of this activity is found in the official documentation of Expo for building https://docs.expo.dev/build/setup/ React Native applications. Feel free to read the corresponding documents for more details.

### 4. Materials and Equipment

To properly perform this activity, the student must have:
- Nodejs LTS
- Visual Studio Code
- Emulator/Similar for Android/iOS

### 5. Procedure

#### Step 0: PRE-REQUISITES
EAS Build is a new and rapidly evolving service. Before you set out to create a build for your project we recommend consulting the limitations page and the other prerequisites below.

https://docs.expo.dev/build-reference/limitations/

1. Choose the application you want to build (You have 3 choices: the To-Do List App, the Messaging App, or your Final Project App)
2. Create an Expo User account. EAS Build is available to anyone with an Expo account, regardless of whether you pay for EAS or use the free tier. You can sign up at https://expo.dev/signup

**Step 1: Install the latest EAS CLI**
EAS CLI is the command-line app that you will use to interact with EAS services from your terminal. To install it, run the command:

```
npm install -g eas-cli
```

You can also use the above command to check if a new version of EAS CLI is available. We encourage you to always stay up to date with the latest version.
We recommend using npm instead of yarn for global package installations. You may alternatively use `npx eas-cli@latest`, just remember to use that instead of eas whenever it's called for in the documentation.

**Step 2: Log in to your Expo Account**
If you are already signed in to an Expo account using Expo CLI, you can skip the steps described in this section. If you are not, run the following command to log in:
```
eas login
```

You can check whether you are logged in by running **eas whoami**.

**Step 3: Configure the Project**
To configure an Android or an iOS project for EAS Build, run the following command:

```
eas build:configure
```

To learn more about what happens behind the scenes, see build configuration process reference.
Additional configuration may be required for some scenarios:
- Are you migrating an Expo managed app from "expo build"? Learn about the differences.
- Does your app code depend on environment variables? Add them to your build configuration.
- Is your project inside of a monorepo? Follow these instructions.
- Does your app depend on specific versions of tools like Node, Yam, npm, CocoaPods, or Xcode? Specify these versions in your build configuration.

**Step 4: Run a Build**
For this activity, **you are not required to publish your application for app stores**. Only to build for Android Emulator/device. Optionally, you may also build for iOS simulator.

The easiest way to try out EAS Build is to create a build that you can run on your Android device/emulator or iOS Simulator. It's quicker than uploading it to a store, and you don't need

store developer membership accounts.

- If you'd like to try this, read about:
  - [Creating an installable APK for Android](#)
  - [Creating a simulator build for iOS](#)

## Step 5: Wait for Build to Complete
By default, the eas build command will wait for your build to complete, but you can interrupt it if you prefer not to wait. Monitor the progress and read the logs by following the link to the build details page that EAS CLI prompts once the build process gets started. You can also find this page by visiting your build dashboard or running the following command:
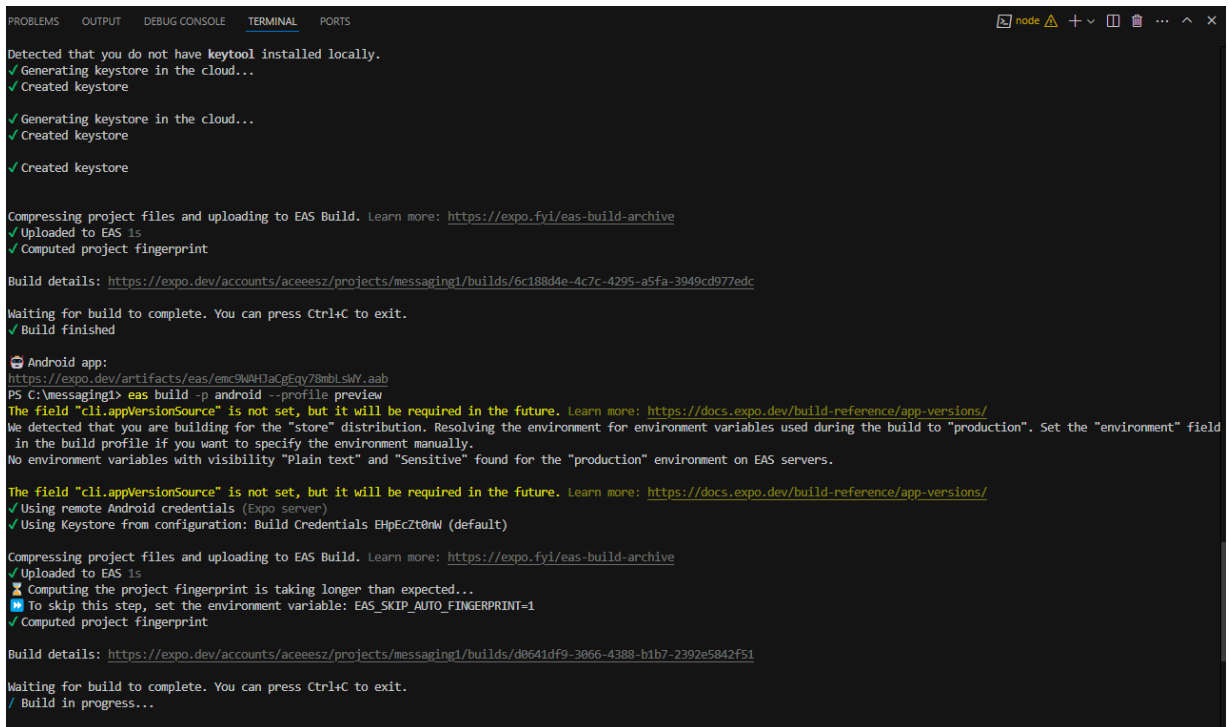
```
eas build:list
```

## Step 6: Deploy the Build
Congratulations on making it to this step! Now, you can open your emulator and install the application.

### Install and run the app
You will only be able to install the app directly to your Android device/iOS Simulator if you explicitly built it for that purpose. If you built for app store distribution, you will need to upload to an app store and then install it from there (for example, from Apple's TestFlight app).

## 6. Output

| 7. Supplementary Activity |
| --- |

**ILO1: Demonstrate the deployment of an application built in previous modules**
Choose one of the applications we have built in the class. Follow the above steps to configure your project, build and install in your emulator. Show that the app is working.

Output requirement:
1. Screenshots of per step requirement for your chosen app.

2. Short video showing the installation and use of your app on your emulator / phone.

**ALFEROS**

**Step 1: Install the latest EAS CLI**

```
PS C:\Users\admin\Documents\Emtech3-HOA\awesomeProject> npm install -g
eas-cli
npm warn deprecated inflight@1.0.6: This module is not supported, and
```

**Step 2: Log in to your Expo Account**

```
PS C:\Users\admin\Documents\Emtech3-HOA\awesomeProject> eas login
You are already logged in as jlalferos.
√ Do you want to continue? ... yes
Log in to EAS with email or username (exit and run eas login --help to
see other login options)
√ Email or username ... jlalferos
√ Password ... ***************
Logged in
```

**Step 3: Configure the Project**

```
PS C:\Users\admin\Documents\Emtech3-HOA\awesomeProject> eas build:configure
EAS project not configured.
√ Would you like to automatically create an EAS project for @jlalferos/awesomeProject? ... yes
√ Created @jlalferos/awesomeProject: https://expo.dev/accounts/jlalferos/projects/awesomeProject on Expo
√ Linked local project to EAS project f9bcf1b5-6dd4-48eb-a69f-6daf62c1efd0
💡 The following process will configure your iOS and/or Android project to be compatible with EAS Build. Th

√ Which platforms would you like to configure for EAS Build? » Android

√ Generated eas.json. Learn more: https://docs.expo.dev/build-reference/eas-json/

🎉 Your project is ready to build.

- Run eas build when you are ready to create your first build.
- Once the build is completed, run eas submit to upload the app to app stores.
- Learn more about other capabilities of EAS Build: https://docs.expo.dev/build/introduction
PS C:\Users\admin\Documents\Emtech3-HOA\awesomeProject>
```

**Step 4: Run a Build**

```json
{
  "build": {
    "preview": {
      "android": {
        "buildType": "apk"
      }
    },
    "preview2": {
      "android": {
        "gradleCommand": ":app:assembleRelease"
      }
    },
    "preview3": {
      "developmentClient": true
    },
    "preview4": {
      "distribution": "internal"
    },
    "production": {}
  }
}
```

## Step 5: Wait for Build to Complete

```
PS C:\Users\admin\Documents\Emtech3-HOA\awesomeProject> eas build -p android --profile preview
The field "cli.appVersionSource" is not set, but it will be required in the future. Learn more: https://docs.exp
We detected that you are building for the "store" distribution. Resolving the environment for environment variab
u want to specify the environment manually.
No environment variables with visibility "Plain text" and "Sensitive" found for the "production" environment on

📝 Android application id Learn more: https://expo.fyi/android-package
√ What would you like your Android application id to be? ... com.jlalferos.awesomeProject
The field "cli.appVersionSource" is not set, but it will be required in the future. Learn more: https://docs.exp
√ Using remote Android credentials (Expo server)
√ Generate a new Android Keystore? ... yes
Detected that you do not have keytool installed locally.
√ Generating keystore in the cloud...
√ Created keystore

Compressing project files and uploading to EAS Build. Learn more: https://expo.fyi/eas-build-archive
√ Uploaded to EAS 1s
/ Computing project fingerprint
```

```
Compressing project files and uploading to EAS Build. Learn more: https://expo.fyi/eas-build-archive
✓ Uploaded to EAS 1s
⧗ Computing the project fingerprint is taking longer than expected...
⏩ To skip this step, set the environment variable: EAS_SKIP_AUTO_FINGERPRINT=1
✓ Computed project fingerprint

Build details: https://expo.dev/accounts/jlalferos/projects/awesomeProject/builds/21363ec4-8082-46fd-94f0-bbb9428e5a36

Waiting for build to complete. You can press Ctrl+C to exit.
- Build queued...
```

## Step 6: Deploy the Build

```
Build details: https://expo.dev/accounts/jlalferos/projects/awesomeProject/builds/dcfedfc5-bf29-4011-a579-0f688f68609d

Waiting for build to complete. You can press Ctrl+C to exit.
✓ Build finished

🤖 Android app:
https://expo.dev/artifacts/eas/thMNOXuSD8TX12hYosog8t.apk
```

## Install and run the app
https://drive.google.com/drive/folders/17zZL-WASQgCuRYhiqC9CvbCiJfbkWlAb?usp=sharing

## EFA
## Step 1: Install the latest EAS CLI

```
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\messaging1> npm install -g eas-cli
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by
 a key value, which is much more comprehensive and powerful.
npm warn deprecated rimraf@2.4.5: Rimraf versions prior to v4 are no longer supported
npm warn deprecated @oclif/screen@3.0.8: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated glob@6.0.4: Glob versions prior to v9 are no longer supported
npm warn deprecated @xmldom/xmldom@0.7.13: this version is no longer supported, please update to at least 0.8.*

changed 446 packages in 32s
```
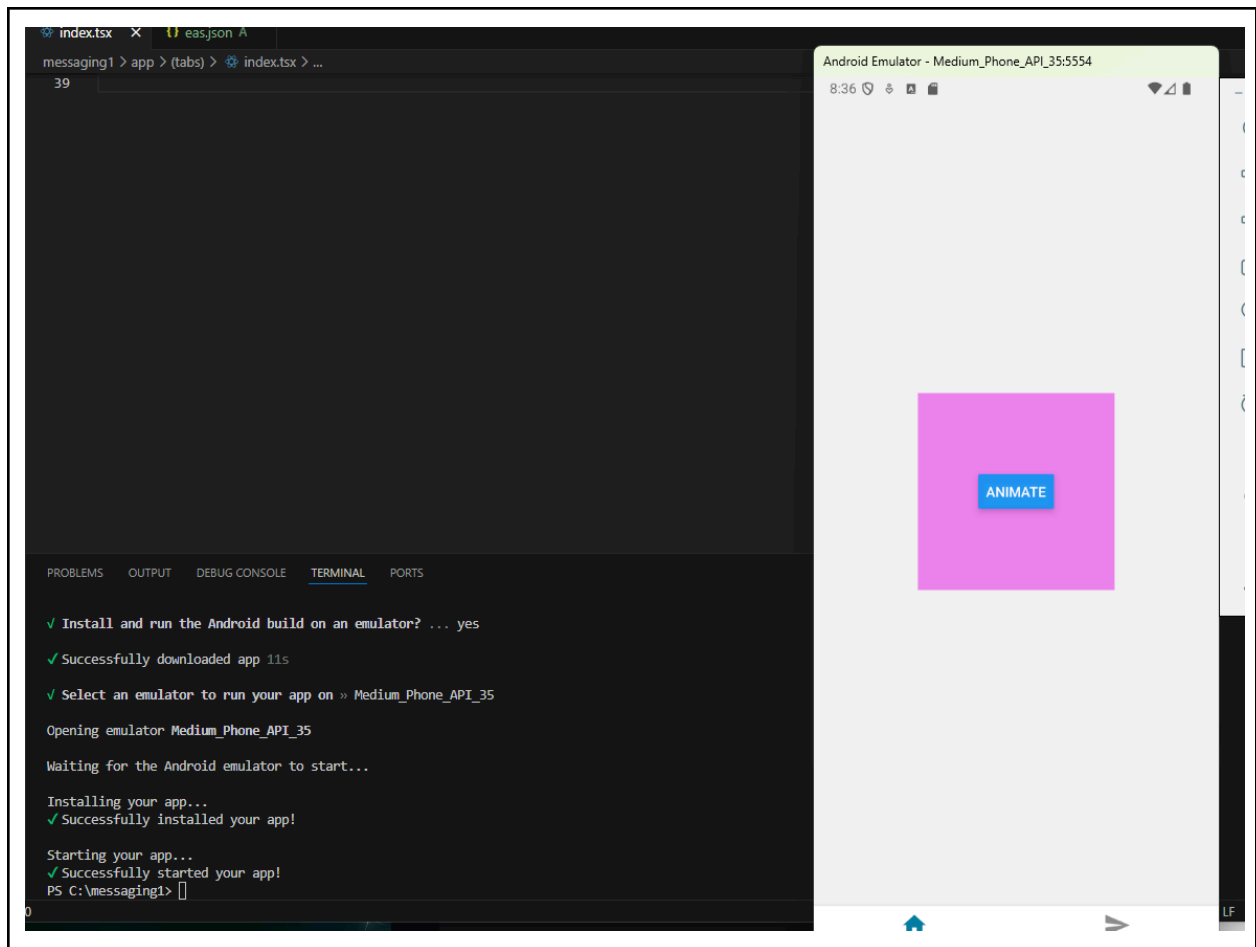
## Step 2: Log in to your Expo Account

```
- Learn more about other capabiliti
PS C:\messaging1> eas build:list

Builds for @aceeesz/messaging1:
```
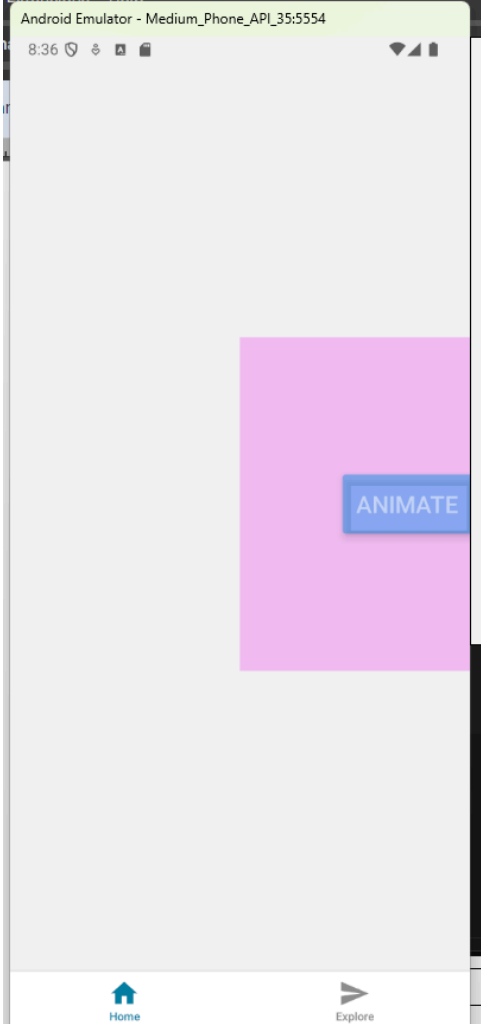
## Step 3: Build

```
build details: https://expo.dev/accounts/aceeesz/projects/messaging

 Waiting for build to complete. You can press Ctrl+C to exit.
 - Build in progress...
```

ANIMATE

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

√ Install and run the Android build on an emulator? ... yes

√ Successfully downloaded app 11s

√ Select an emulator to run your app on » Medium_Phone_API_35

Opening emulator Medium_Phone_API_35

Waiting for the Android emulator to start...

Installing your app...
√ Successfully installed your app!

Starting your app...
√ Successfully started your app!
PS C:\messaging1>

[VIDEO LINK](#)

| 8. Conclusion |
| --- |

**ALFEROS**
In this activity, we explored the use of Expo's EAS Build to deploy applications for iOS or Android. Following the procedure, we learned to configure their projects, build applications, and install it on emulators or devices. Through this activity, we gained practical experience in preparing applications for real-world usage, laying the foundation for future app development projects.

**EFA**
This task was aimed to guide us on how to Demonstrate the deployment of an application built in previous modules. In different OS such as IOS and ANDROID. IOS consists of IPA while Android requires the APK file.