

Let's walk through **how to run a basic “Hello World” app in React Native** — from scratch — on your PC (Windows), for Android.

✓ I'll Guide You Through Both — Choose One

A OPTION 1: Using React Native CLI (Full Android + Native Setup)

🔗 Prerequisites (Already Done or In Progress for You):

- ✓ Node.js installed
 - ✓ JDK (Java) installed and `JAVA_HOME` set
 - ✓ Android Studio installed + SDK + emulator
 - ✓ Android SDK path set in `ANDROID_HOME` or `local.properties`
-

📁 Step-by-Step: Create and Run Hello World

1. Create the app

```
npx @react-native-community/cli init MyFirstApp
cd MyFirstApp
```

2. Replace `App.tsx` with Hello World

Open `App.tsx` and paste:

```
import React from 'react';
import { SafeAreaView, Text, StyleSheet } from 'react-native';

const App = () => {
  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.text}>Hello, World! 🙌</Text>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f5f5f5',
  },
});
```

```
    text: {
      fontSize: 24,
      color: '#333',
    },
  });

export default App;
```

3. Start Metro bundler (in Terminal 1)

```
npx react-native start
```

Leave this terminal open.

4. Start your emulator (or connect a device)

- Launch an emulator from Android Studio
- OR plug in your Android phone with USB debugging enabled

5. Build and run the app (in Terminal 2)

```
npx react-native run-android
```

☑ If everything is set up, you'll see "Hello, World!" on your Android screen.

Step 1: download and install Android Studio :-

← → ↻ 🔍 developer.android.com/studio?utm_source=chatgpt.com

🐾 Developers Essentials ▾ Design & Plan ▾ Develop ▾

ANDROID STUDIO

Download IDE guides Gemini in Android Studio Gradle build guides

Android Studio

The official IDE for Android app development now accelerates your productivity with Gemini in Android Studio, your AI-powered coding companion.

Download Android Studio Narwhal 3 Feature Drop ↓

🔧 Step 1: Install Android Studio (if not done)

Download and install:

🔗 <https://developer.android.com/studio>

During installation, **make sure you check:**

- Android SDK
- Android SDK Platform-Tools
- Android Virtual Device (AVD) Manager

🔧 Step 2: Add Android & Java to Environment Variables

🔑 A. Add Android SDK and Platform Tools to PATH

1. Find your Android SDK path (default is usually):

C:\Users\

2. Add the following to your **System Environment Variables** → **PATH**:

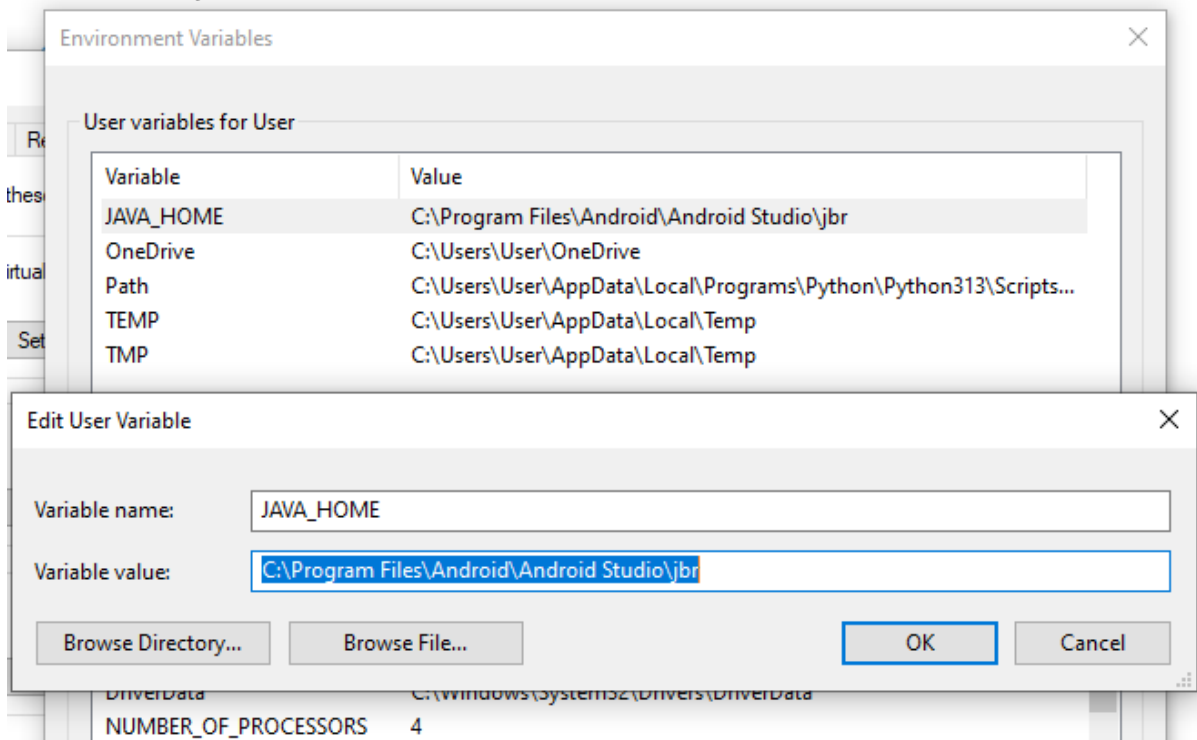
C:\Users\

C:\Users\

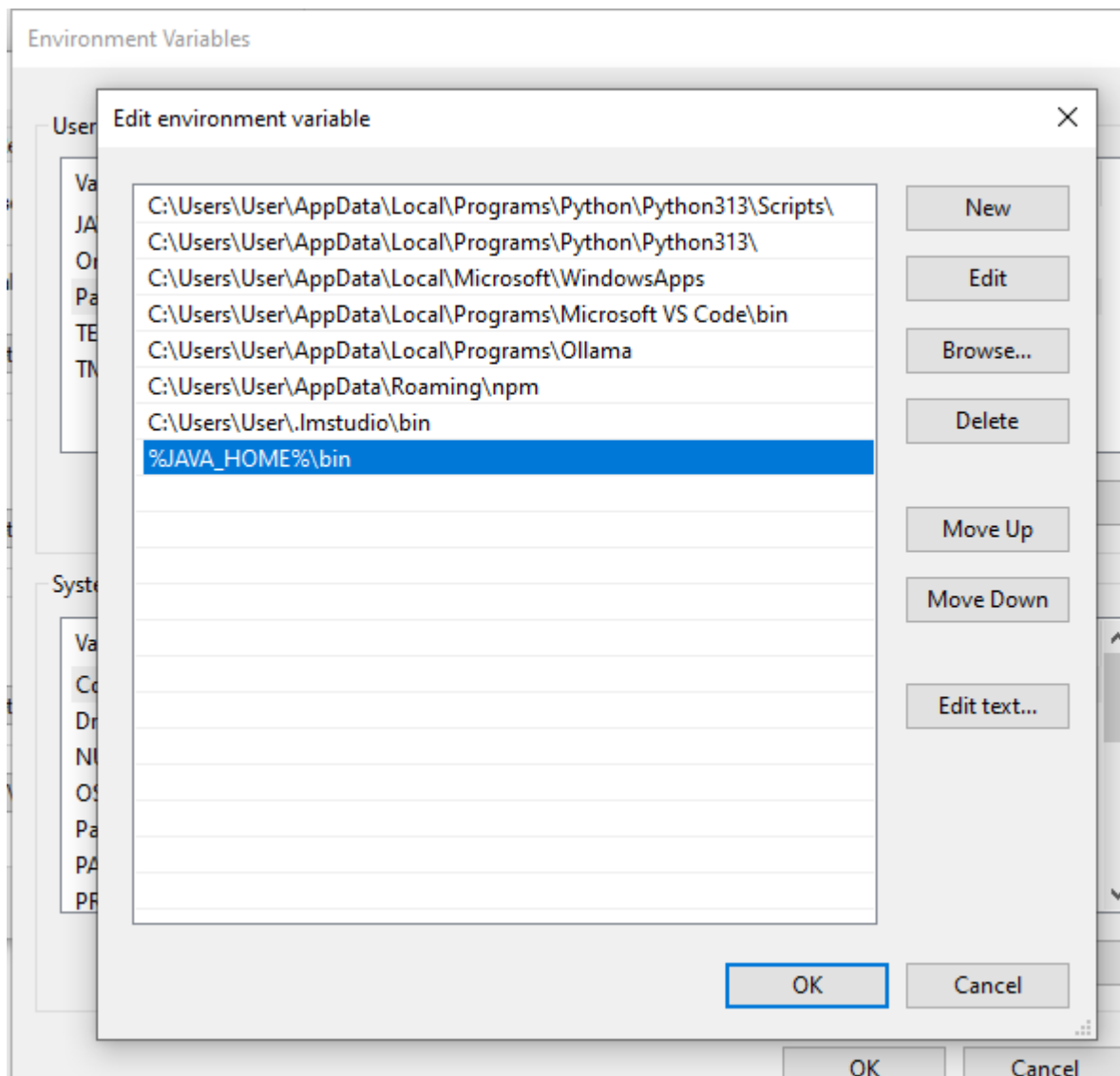
🔑 B. Set JAVA_HOME

1. Open Android Studio → File > Project Structure → check the path to the JDK.
 - Usually something like:
 - C:\Program Files\Android\Android Studio\jbr
2. Add this path as a new environment variable:
 - Variable name: JAVA_HOME
 - Variable value: C:\Program Files\Android\Android Studio\jbr

more and protected.



3. Also add %JAVA_HOME%\bin to your **PATH**.



Step 3: Create and Start an Emulator

1. Open Android Studio
2. Go to:
3. Tools > Device Manager
4. Click **“Create Device”**
5. Choose a phone (e.g., Pixel 5) and a system image (like Android 13)
6. Finish setup and then click **Start**

Once it's running, go back to your project folder and run:

Open command prompt :-

```
npx @react-native-community/cli init MyFirstApp
```

Cd MYFirstApp

```
npx react-native run-android
```

you will see following after running this command:-

```
F:\mobile\MyFirstApp>npx react-native run-android
info A dev server is already running for this project on port 8081.
'"adb"' is not recognized as an internal or external command,
operable program or batch file.
info Launching emulator...
error Failed to launch emulator. Reason: No emulators found as an output of `e
warn Please launch an emulator manually or connect a device. Otherwise app may
info Installing the app...
Downloading https://services.gradle.org/distributions/gradle-8.14.3-bin.zip
.....10%.....20%.....30%.....40%.....5
.....80%.....90%.....100%

Welcome to Gradle 8.14.3!

Here are the highlights of this release:
- Java 24 support
- GraalVM Native Image toolchain selection
- Enhancements to test reporting
- Build Authoring improvements

For more details see https://docs.gradle.org/8.14.3/release-notes.html

Starting a Gradle Daemon (subsequent builds will be faster)
<-----> 0% INITIALIZING [2m 35s]
> IDLE
> IDLE
> IDLE
> Evaluating settings > :gradle-plugin:settings-plugin:compileKotlin
```

Optional: Run Doctor

```
npx react-native doctor
```

This will check and guide you on any remaining setup problems.

Set `ANDROID_HOME` Environment Variable (recommended)

🔍 Step-by-step:

1. Find your Android SDK path

Default location is usually:

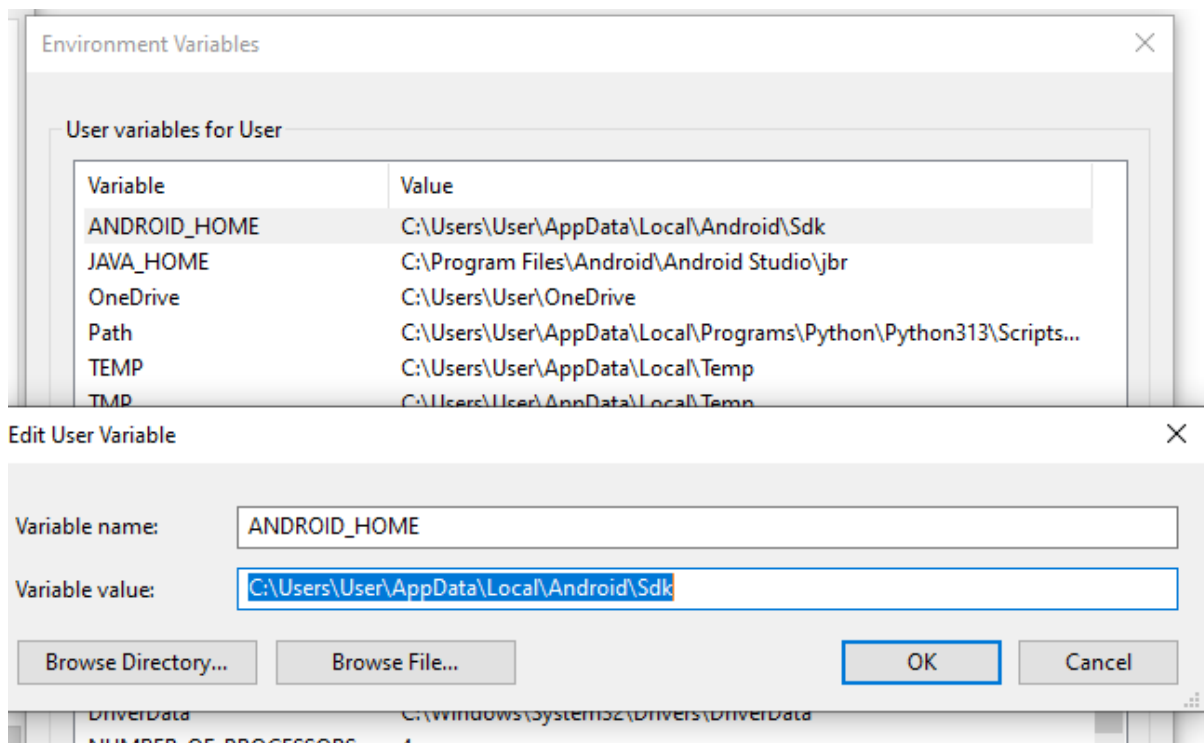
```
C:\Users\\AppData\Local\Android\Sdk
```

To confirm:

- Open Android Studio
- Go to **File > Settings > Appearance & Behavior > System Settings > Android SDK or tools and sdk manager**
- Note the SDK path listed at the top

2. Set `ANDROID_HOME` Environment Variable

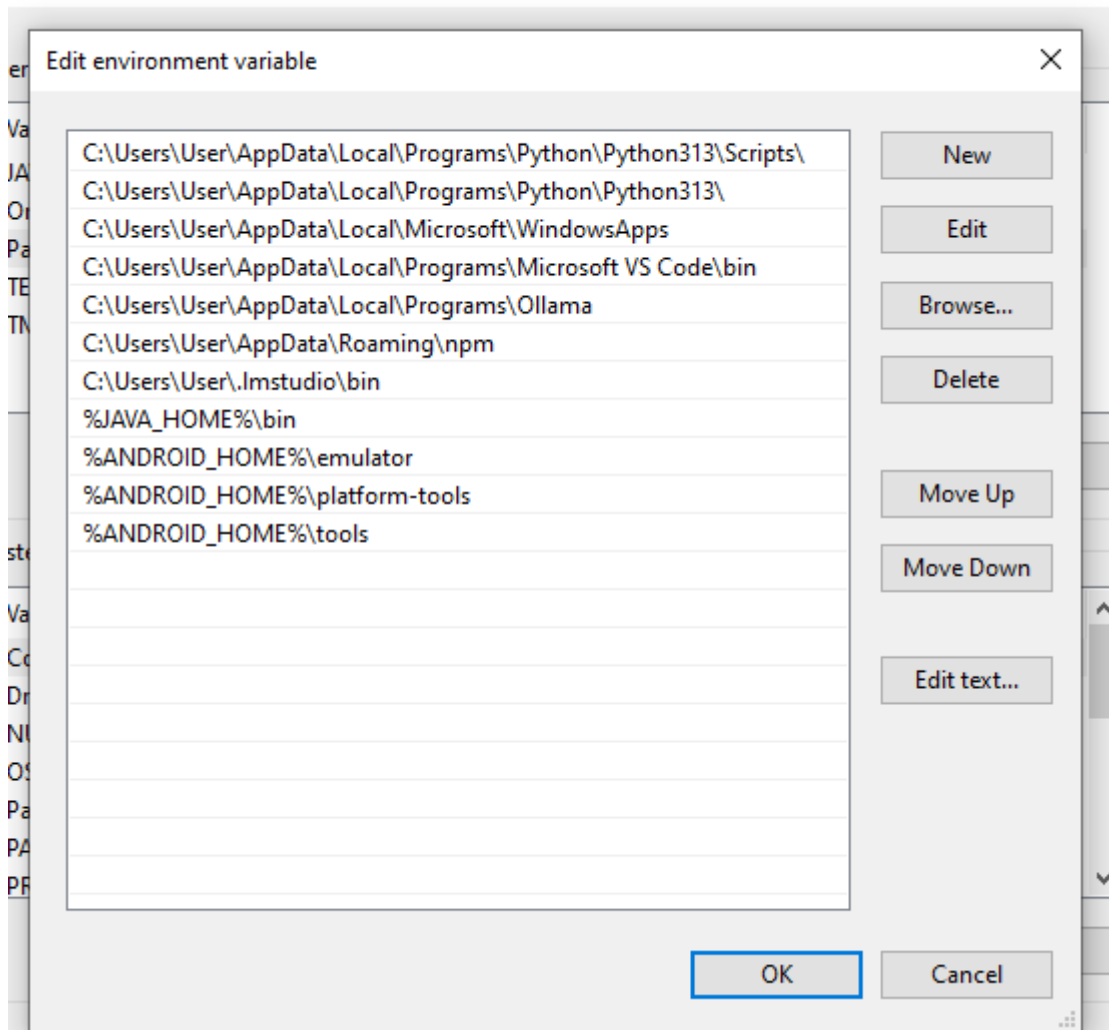
1. Open **System Environment Variables**
 - Press `Win + S` → search: *"environment variables"*
2. Under **System Variables**, click **New**:
 - **Name:** `ANDROID_HOME`
 - **Value:** `C:\Users\\AppData\Local\Android\Sdk`



3. Edit your `Path` variable, and add these **3 lines**:
4. `%ANDROID_HOME%\platform-tools`

5. %ANDROID_HOME%\emulator
6. %ANDROID_HOME%\tools

Environment Variables



7. Click OK to save everything

3. Restart your terminal

To apply the new env vars.

✔ Option 2: Add SDK path directly in local.properties (easy & quick)

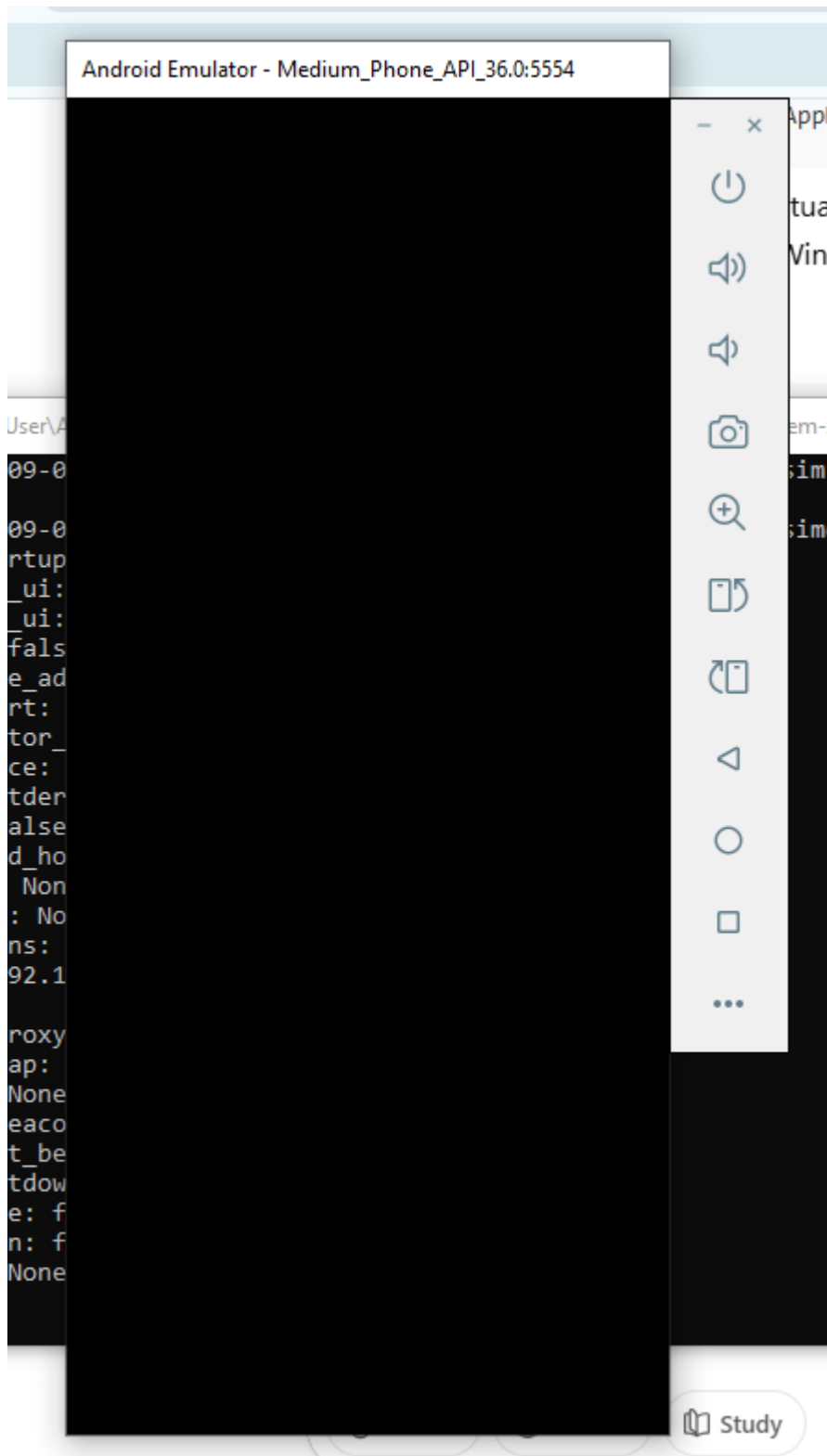
1. Go to your project folder:

```
F:\mobile\MyFirstApp\android\
```

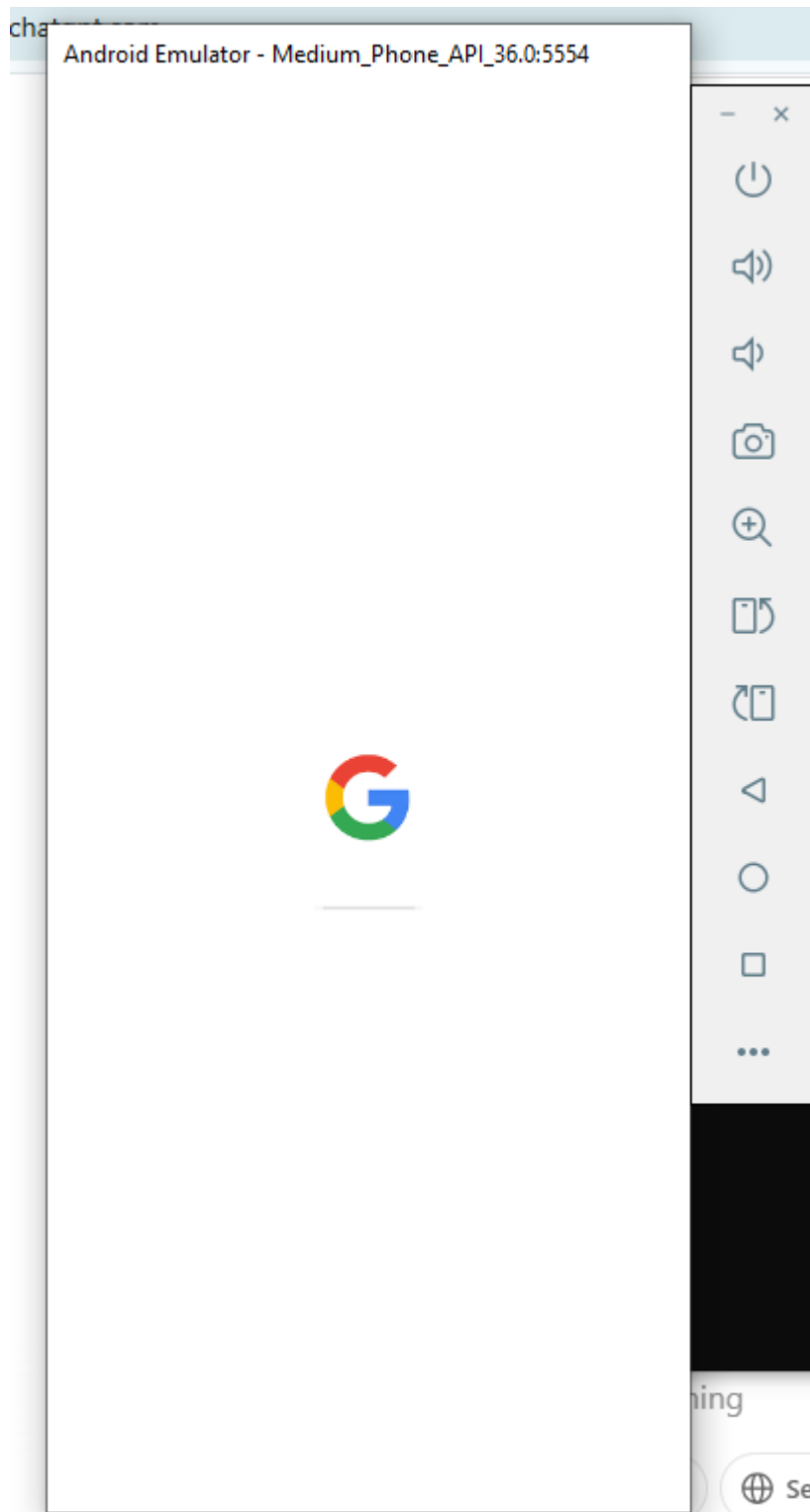
```
F:\mobile\MyFirstApp>npx react-native run-android
info A dev server is already running for this project on port 8081.
info Launching emulator...
error Failed to launch emulator. Reason: It took too long to start and c
6.0. You can try starting the emulator manually from the terminal with:
/emulator @Medium_Phone_API_36.0.
warn Please launch an emulator manually or connect a device. Otherwise a
info Installing the app...

> Configure project :app
Checking the license for package NDK (Side by side) 27.1.12297006 in C:\
License for package NDK (Side by side) 27.1.12297006 accepted.
Preparing "Install NDK (Side by side) 27.1.12297006 v.27.1.12297006".
<-----> 0% CONFIGURING [1m 5s]
> root project > :app
```

Finally you will see:-



And you will see gradually like this :-



And final output will be in cmd you will see :-

```
\app\build\intermediates\cxx\Debug\1c2ov4u5\obj\x86_64\libreactnative.so' failed. Doing a slower copy instead.

> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Medium_Phone_API_36.0(AVD) - 16' for :app:debug
Installed on 1 device.

[Incubating] Problems report is available at: file:///F:/mobile/MyFirstApp/android/build/reports/problems/problems-report.html

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.14.3/userguide/command_line_interface.html#sec:command_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 14m 57s
82 actionable tasks: 72 executed, 10 up-to-date
info Connecting to the development server...
8081
info Starting the app...

Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.myfirstapp/.MainActivity }
```

Android Emulator - Medium_Phone_API_36.0:5554

Hello, World! 🖐️



⚠️ Open debugger to view warnings.

Search (

Note in my MyFirstApp Folder in my **App.tsx file** I have written this Code :-

```
import React from 'react';
import { SafeAreaView, Text, StyleSheet } from 'react-native';
```

```
const App = (): JSX.Element => {
  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.text}>Hello, World! 🌍</Text>
    </SafeAreaView>
  );
};
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#e5e5e5',
  },
  text: {
    fontSize: 24,
    color: '#333',
  },
});
```

```
export default App;
```

Project Folder Structure :-

This PC > Anigrfx-02 (F:) > mobile > MyFirstApp

	Name
	.bundle
	__tests__
	android
	ios
	node_modules
larke	.eslintrc
	.gitignore
	.prettierrc
	.watchmanconfig
	app
	App
	babel.config
	Gemfile
	index
	jest.config
	metro.config
	package
	package-lock
	README
	tsconfig