Lab 3: React Native To-Do List Application

Mouli Naidu Lukalapu
Instructor: Flavio Esposito
Course: Web Technologies
GitHub: https://github.com/mouli86/web-tech-lab3

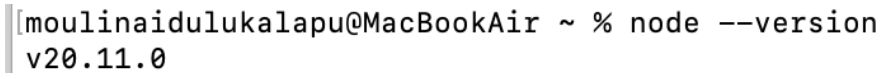
November 18, 2024

Task 1: Set Up the Dev Environment

In this task, you will set up your development environment to build React Native applications.

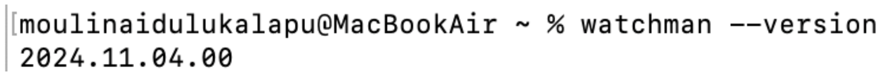
1.4 Step 1: Install Node.js and Watchman

1. **Install Node.js:**



```
moulinaidulukalapu@MacBookAir ~ % node --version
v20.11.0
```

2. **Install Watchman (Optional for macOS/Linux):**



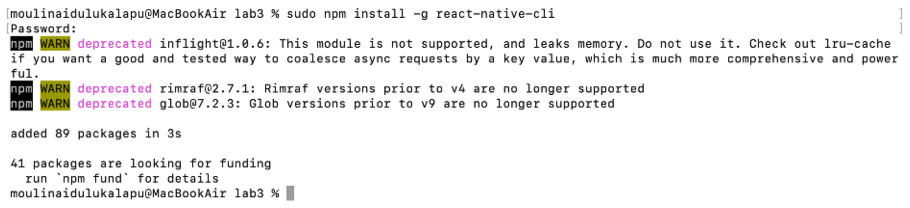
```
moulinaidulukalapu@MacBookAir ~ % watchman --version
2024.11.04.00
```

1.5 Step 2: Install React Native CLI

The React Native CLI (Command Line Interface) allows you to create new React Native projects and run them easily.

- Open your terminal and run:

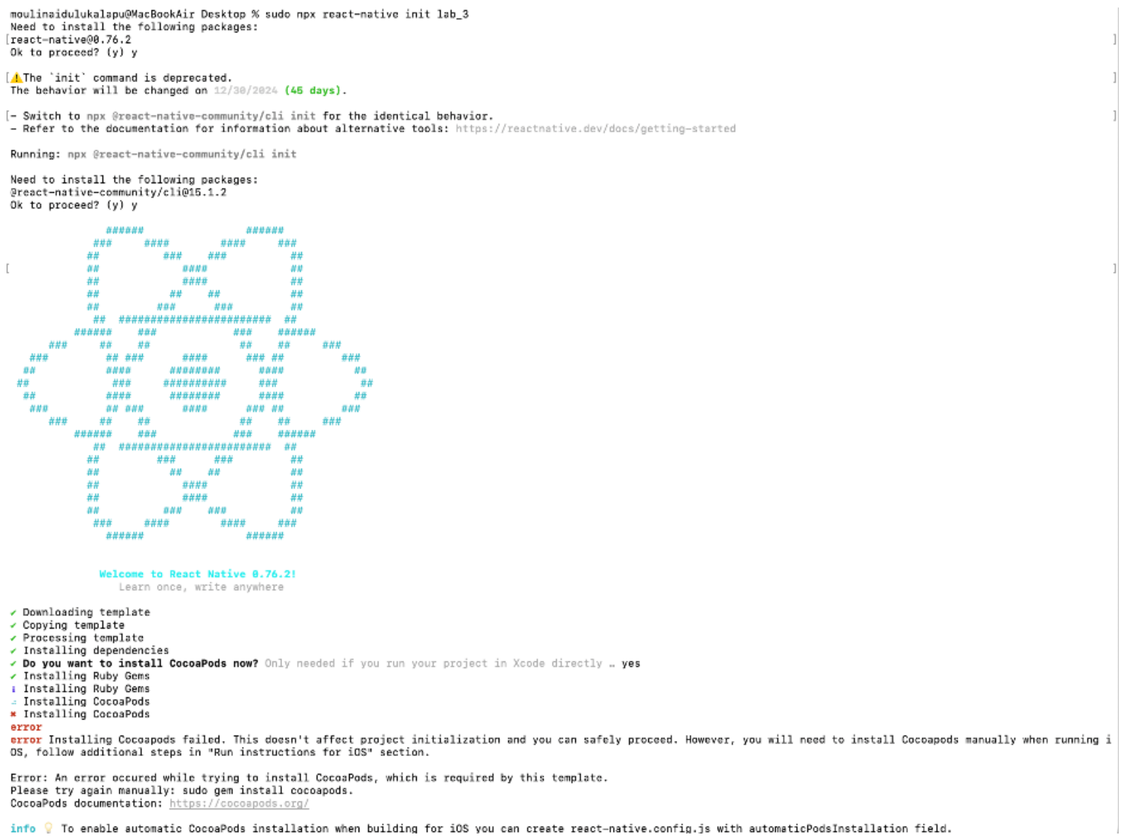
```
npm install -g react-native-cli
```



```
moulinaidulukalapu@MacBookAir lab3 % sudo npm install -g react-native-cli
Password:
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 power
ful.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 89 packages in 3s

41 packages are looking for funding
  run `npm fund` for details
moulinaidulukalapu@MacBookAir lab3 %
```

- Note: If the global ‘react-native-cli‘ package has been deprecated, you can use the local version with ‘npx‘:

```
npx react-native init YourProjectName
```

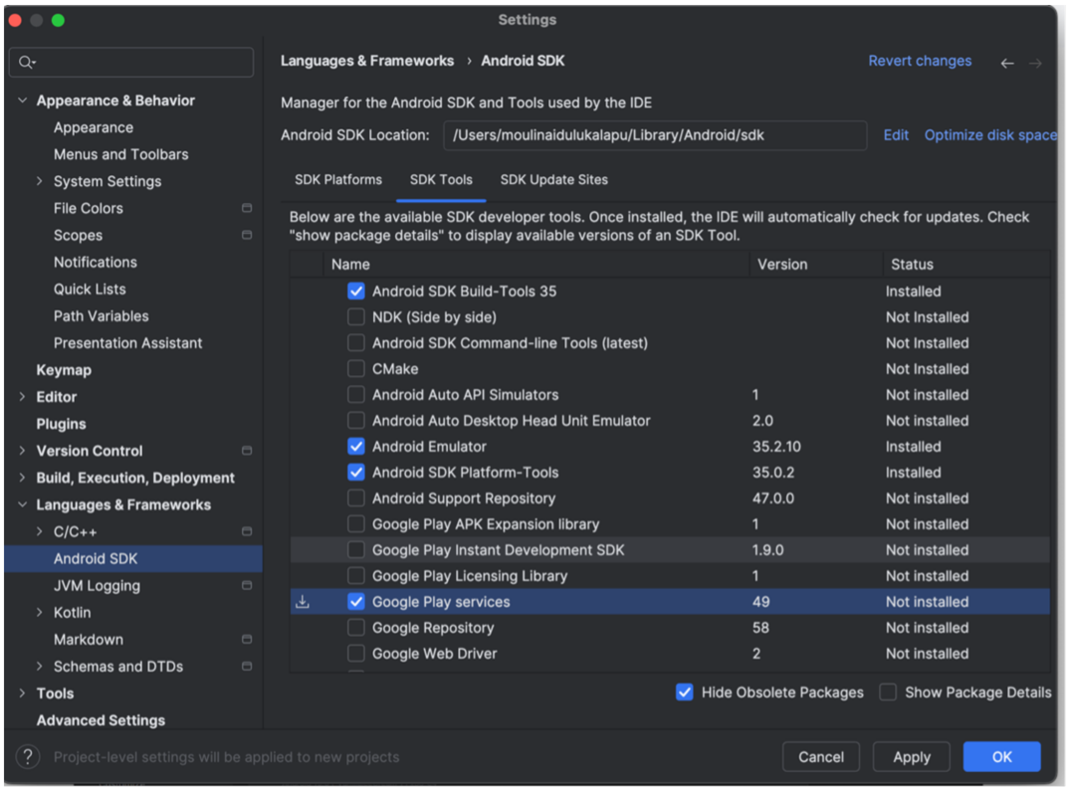


```
moulinaidulukalapu@MacBookAir Desktop % sudo npx react-native init lab_3
Need to install the following packages:
react-native@0.76.2
Ok to proceed? (y) y

⚠The `init` command is deprecated.
The behavior will be changed on 12/30/2024 (45 days).

- Switch to npx @react-native-community/cli init for the identical behavior.
- Refer to the documentation for information about alternative tools: https://reactnative.dev/docs/getting-started

Running: npx @react-native-community/cli init

Need to install the following packages:
@react-native-community/cli@15.1.2
Ok to proceed? (y) y

               ######                ######
             ###     ####        ####     ###
            ##          ###    ###          ##
            ##             ####             ##
            ##             ####             ##
            ##           ##    ##           ##
            ##         ###      ###         ##
             ##  ########################  ##
          ######    ###            ###    ######
      ###     ##    ##              ##    ##     ###
   ###         ## ###      ####      ### ##         ###
  ##           ####      ########      ####           ##
 ##             ###     ##########     ###             ##
  ##           ####      ########      ####           ##
   ###         ## ###      ####      ### ##         ###
      ###     ##    ##              ##    ##     ###
          ######    ###            ###    ######
             ##  ########################  ##
            ##         ###      ###         ##
            ##           ##    ##           ##
            ##             ####             ##
            ##             ####             ##
            ##          ###    ###          ##
             ###     ####        ####     ###
               ######                ######


              Welcome to React Native 0.76.2!
                 Learn once, write anywhere

✔ Downloading template
✔ Copying template
✔ Processing template
✔ Installing dependencies
✔ Do you want to install CocoaPods now? Only needed if you run your project in Xcode directly … yes
✔ Installing Ruby Gems
⠇ Installing Ruby Gems
⠴ Installing CocoaPods
✖ Installing CocoaPods
error
error Installing Cocoapods failed. This doesn't affect project initialization and you can safely proceed. However, you will need to install Cocoapods manually when running i
OS, follow additional steps in "Run instructions for iOS" section.

Error: An error occured while trying to install CocoaPods, which is required by this template.
Please try again manually: sudo gem install cocoapods.
CocoaPods documentation: https://cocoapods.org/

info 💡 To enable automatic CocoaPods installation when building for iOS you can create react-native.config.js with automaticPodsInstallation field.
```

1.6 Step 3: Set Up Android Studio (or Xcode for iOS)

For Android Users:

(a) Enable Recommended SDK Tools under the SDK Tools tab:
   - Android SDK Build-Tools (install the latest version)
   - Android SDK Platform-Tools
   - Android Emulator
   - Google Play Services (if your app needs Google services)



Languages & Frameworks › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: /Users/moulinaidulukalapu/Library/Android/sdk

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
✓ Android SDK Build-Tools 35		Installed
☐ NDK (Side by side)		Not Installed
☐ Android SDK Command-line Tools (latest)		Not Installed
☐ CMake		Not Installed
☐ Android Auto API Simulators	1	Not installed
☐ Android Auto Desktop Head Unit Emulator	2.0	Not installed
✓ Android Emulator	35.2.10	Installed
✓ Android SDK Platform-Tools	35.0.2	Installed
☐ Android Support Repository	47.0.0	Not installed
☐ Google Play APK Expansion library	1	Not installed
☐ Google Play Instant Development SDK	1.9.0	Not installed
☐ Google Play Licensing Library	1	Not installed
✓ Google Play services	49	Not installed
☐ Google Repository	58	Not installed
☐ Google Web Driver	2	Not installed

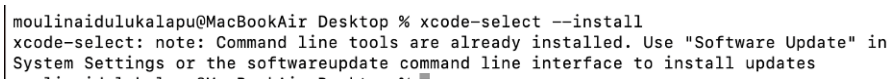
✓ Hide Obsolete Packages ☐ Show Package Details

Project-level settings will be applied to new projects

For iOS Users:

- Ensure Xcode is installed from the App Store.
- Install Xcode Command Line Tools:

```
xcode-select --install
```

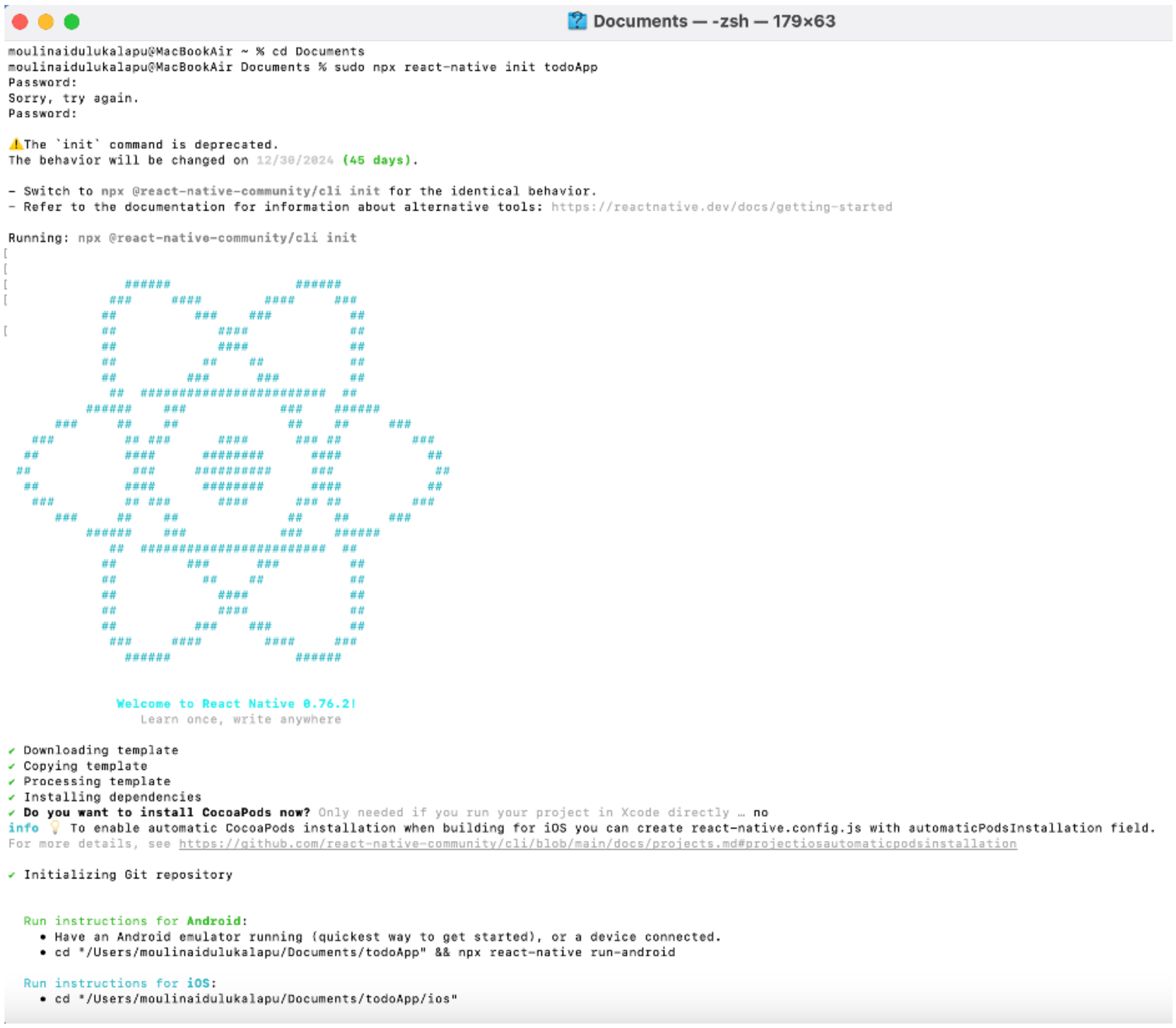


```
moulinaidulukalapu@MacBookAir Desktop % xcode-select --install
xcode-select: note: Command line tools are already installed. Use "Software Update" in
System Settings or the softwareupdate command line interface to install updates
```

1.7 Step 4: Create a New React Native Project

1. Open your terminal and run:

```
npx react-native init YourProjectName
cd YourProjectName
```

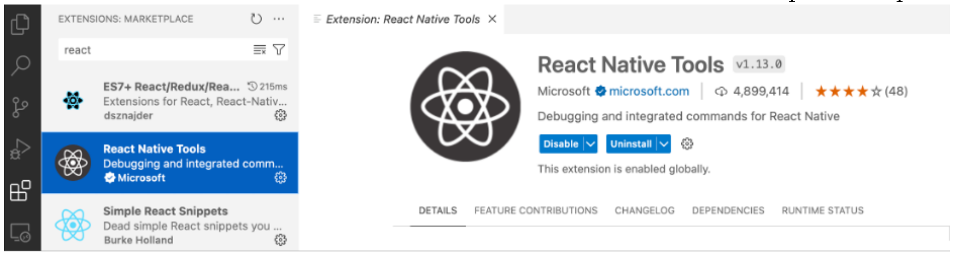


```
Documents — -zsh — 179×63
moulinaidulukalapu@MacBookAir ~ % cd Documents
moulinaidulukalapu@MacBookAir Documents % sudo npx react-native init todoApp
Password:
Sorry, try again.
Password:

⚠️The `init` command is deprecated.
The behavior will be changed on 12/30/2024 (45 days).

- Switch to npx @react-native-community/cli init for the identical behavior.
- Refer to the documentation for information about alternative tools: https://reactnative.dev/docs/getting-started

Running: npx @react-native-community/cli init
[
[
[              ######                ######
[            ###     ####        ####     ###
            ##          ###    ###          ##
[           ##             ####             ##
            ##             ####             ##
            ##           ##    ##           ##
            ##         ###      ###         ##
             ##  ########################  ##
          ######    ###            ###    ######
      ###     ##    ##              ##    ##     ###
   ###         ## ###      ####      ### ##         ###
  ##           ####      ########      ####           ##
 ##             ###     ##########     ###             ##
  ##           ####      ########      ####           ##
   ###         ## ###      ####      ### ##         ###
      ###     ##    ##              ##    ##     ###
          ######    ###            ###    ######
             ##  ########################  ##
            ##         ###      ###         ##
            ##           ##    ##           ##
            ##             ####             ##
            ##             ####             ##
            ##          ###    ###          ##
             ###     ####        ####     ###
               ######                ######


              Welcome to React Native 0.76.2!
                 Learn once, write anywhere

✔ Downloading template
✔ Copying template
✔ Processing template
✔ Installing dependencies
✔ Do you want to install CocoaPods now? Only needed if you run your project in Xcode directly … no
info 💡 To enable automatic CocoaPods installation when building for iOS you can create react-native.config.js with automaticPodsInstallation field.
For more details, see https://github.com/react-native-community/cli/blob/main/docs/projects.md#projectiosautomaticpodsinstallation

✔ Initializing Git repository


  Run instructions for Android:
    • Have an Android emulator running (quickest way to get started), or a device connected.
    • cd "/Users/moulinaidulukalapu/Documents/todoApp" && npx react-native run-android

  Run instructions for iOS:
    • cd "/Users/moulinaidulukalapu/Documents/todoApp/ios"
```

Figure 1: Creating a new React Native project

1.8 Step 5: Open the Project in Visual Studio Code

1. Install the React Native Tools extension in VS Code for a better development experience.



EXTENSIONS: MARKETPLACE

react

ES7+ React/Redux/Rea... 215ms
Extensions for React, React-Nativ...
dsznajder

React Native Tools
Debugging and integrated comm...
Microsoft

Simple React Snippets
Dead simple React snippets you ...
Burke Holland

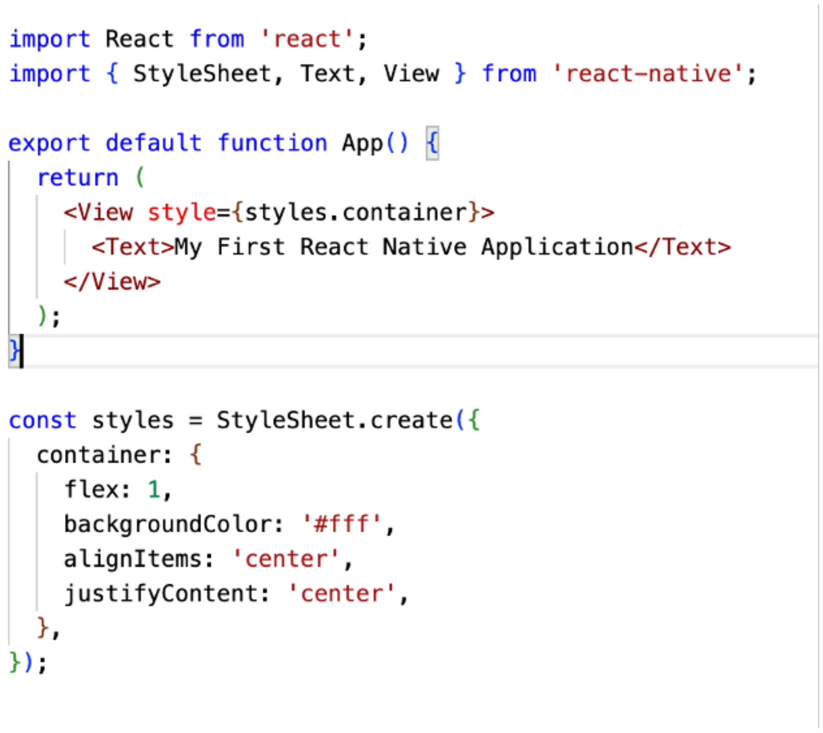
Extension: React Native Tools

React Native Tools v1.13.0
Microsoft microsoft.com | 4,899,414 | ★★★★☆ (48)
Debugging and integrated commands for React Native
Disable | Uninstall
This extension is enabled globally.

DETAILS | FEATURE CONTRIBUTIONS | CHANGELOG | DEPENDENCIES | RUNTIME STATUS

Figure 2: Installing React Native Tools extension

2. Open App.js and modify the content to display ”My First React Native Application”.



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

export default function App() {
  return (
    <View style={styles.container}>
      <Text>My First React Native Application</Text>
    </View>
  );
}

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

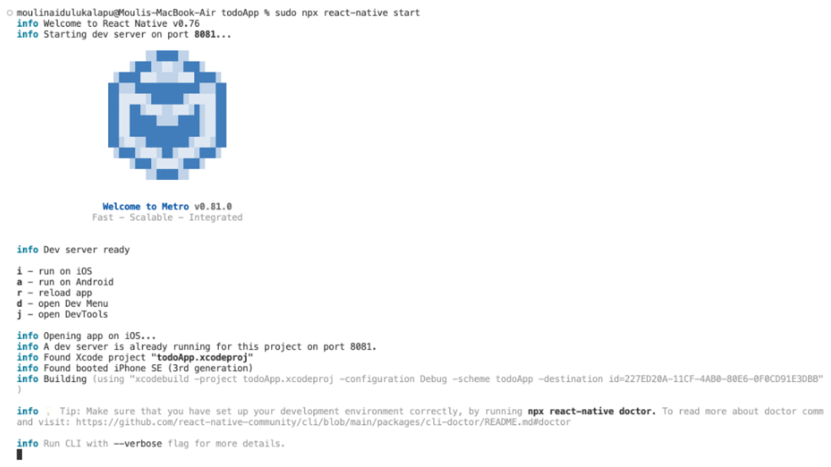
Figure 3: Modifying App.js

1.9 Step 6: Start the Metro Bundler

The Metro Bundler is a JavaScript bundler specifically for React Native. It watches your files and serves the appropriate JavaScript code.

- In your terminal, run:

```
npx react-native start
```



```
○ moulinaidulukalapu@Moulis-MacBook-Air todoApp % sudo npx react-native start
  info Welcome to React Native v0.76
  info Starting dev server on port 8081...

                Welcome to Metro v0.81.0
              Fast - Scalable - Integrated

  info Dev server ready

  i - run on iOS
  a - run on Android
  r - reload app
  d - open Dev Menu
  j - open DevTools

  info Opening app on iOS...
  info A dev server is already running for this project on port 8081.
  info Found Xcode project "todoApp.xcodeproj"
  info Found booted iPhone SE (3rd generation)
  info Building (using "xcodebuild -project todoApp.xcodeproj -configuration Debug -scheme todoApp -destination id=227ED20A-11CF-4AB0-80E6-0F0CD91E3DBB"
  )

  info   Tip: Make sure that you have set up your development environment correctly, by running npx react-native doctor. To read more about doctor comm
  and visit: https://github.com/react-native-community/cli/blob/main/packages/cli-doctor/README.md#doctor

  info Run CLI with --verbose flag for more details.
```

Figure 4: Starting the Metro Bundler

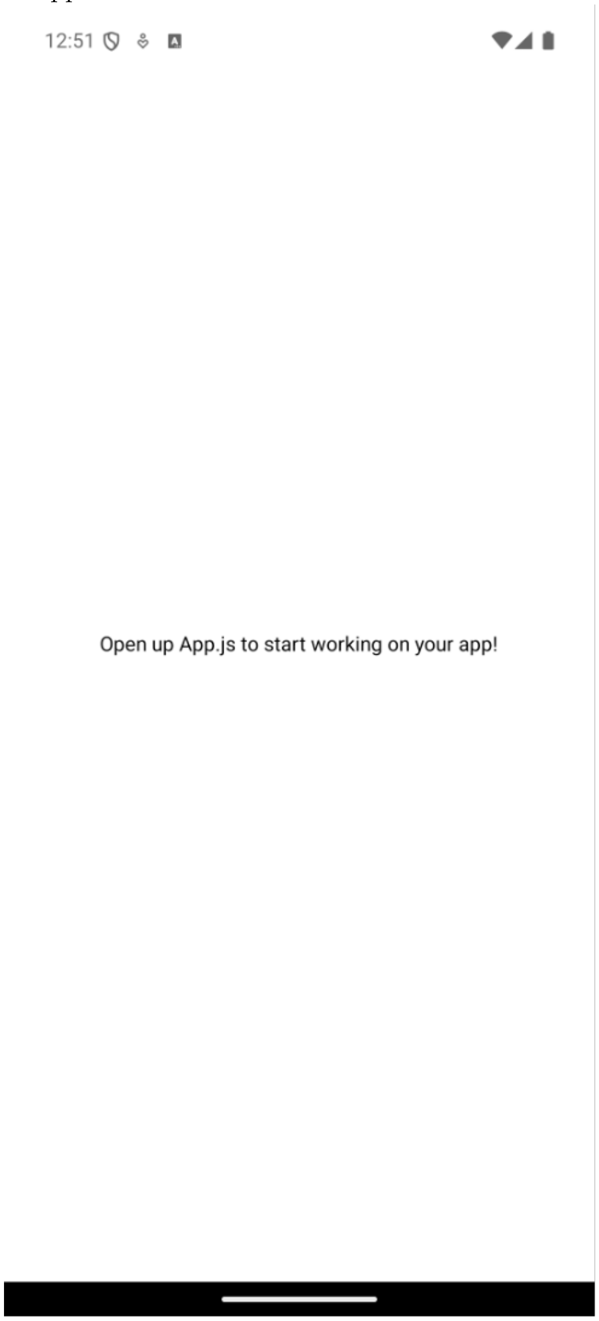
1.10 Step 7: Run Your App on an Emulator or Device

For Android:

1. Ensure an emulator is running (you can create one in Android Studio's AVD Manager) or connect a physical device with USB debugging enabled.
2. Run:

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

3. This compiles your app and installs it on the connected Android device or emulator.



Open up App.js to start working on your app!

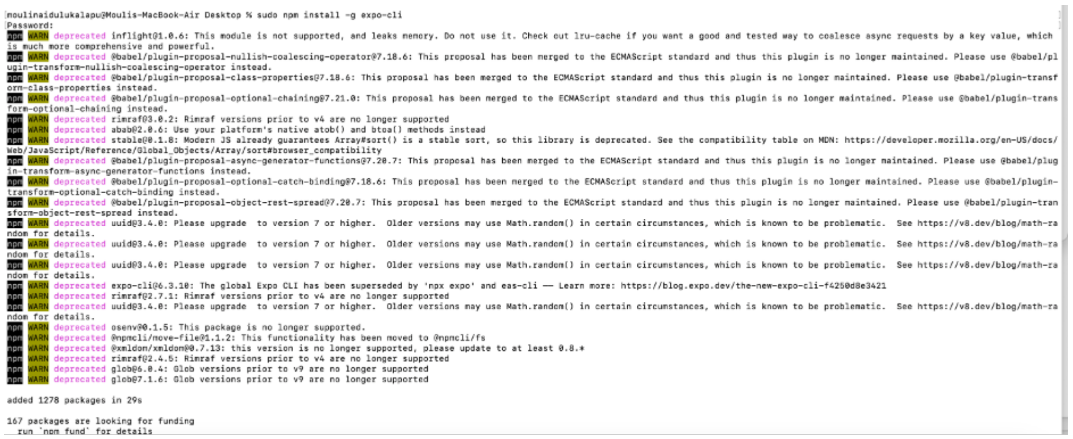
Figure 5: Running the app on an Android device

1.11 Step 8: Run Your App on a Mobile Device Using Expo

1. Install and create a new Expo project:

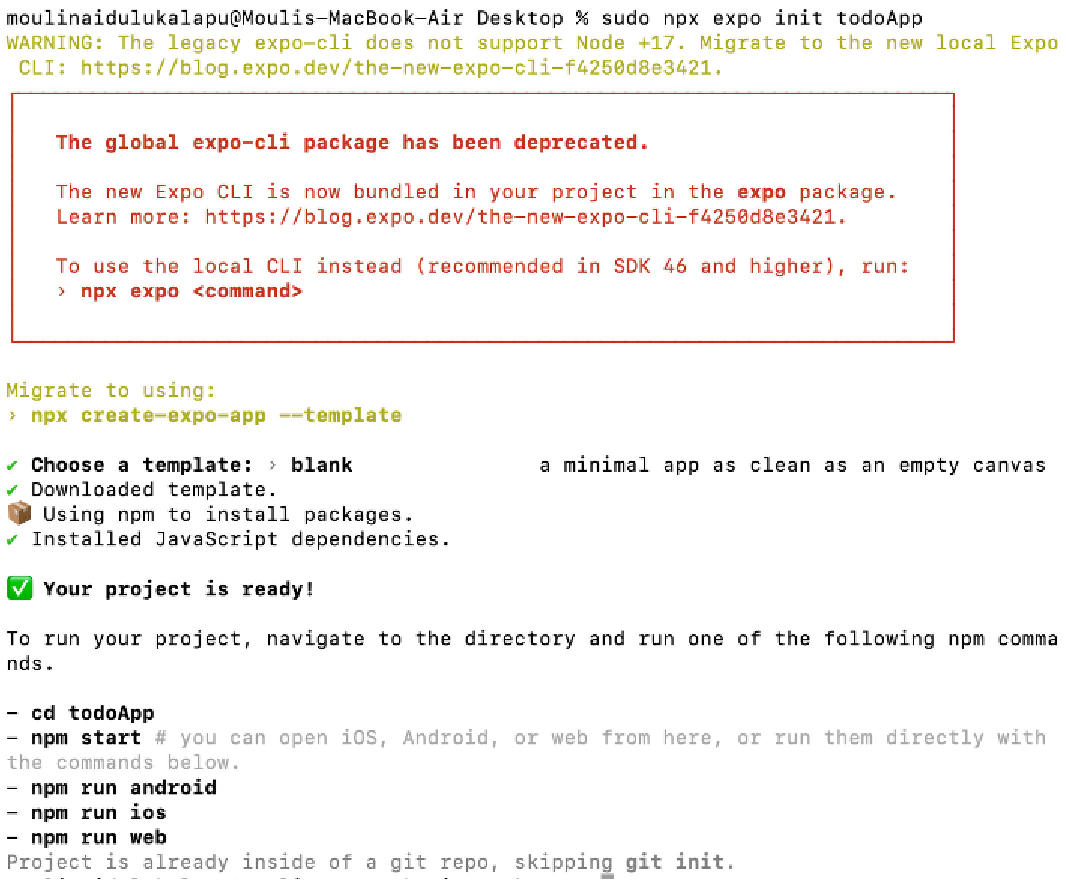
```
npm install -g expo-cli
npx expo init YourProjectName
cd YourProjectName
npx expo start
```

2. Connect Your Device:
   - Ensure your mobile device is connected to the same Wi-Fi network as your development machine.
3. Open the Expo Go App:
   - Install the Expo Go app from the App Store (iOS) or Google Play Store (Android).
4. Scan the QR Code:
   - In the Expo developer tools opened in your browser, you will see a QR code.
   - Use the Expo Go app to scan the QR code.
5. Run the App:
   - Once the QR code is scanned, your React Native app will start running on your mobile device.



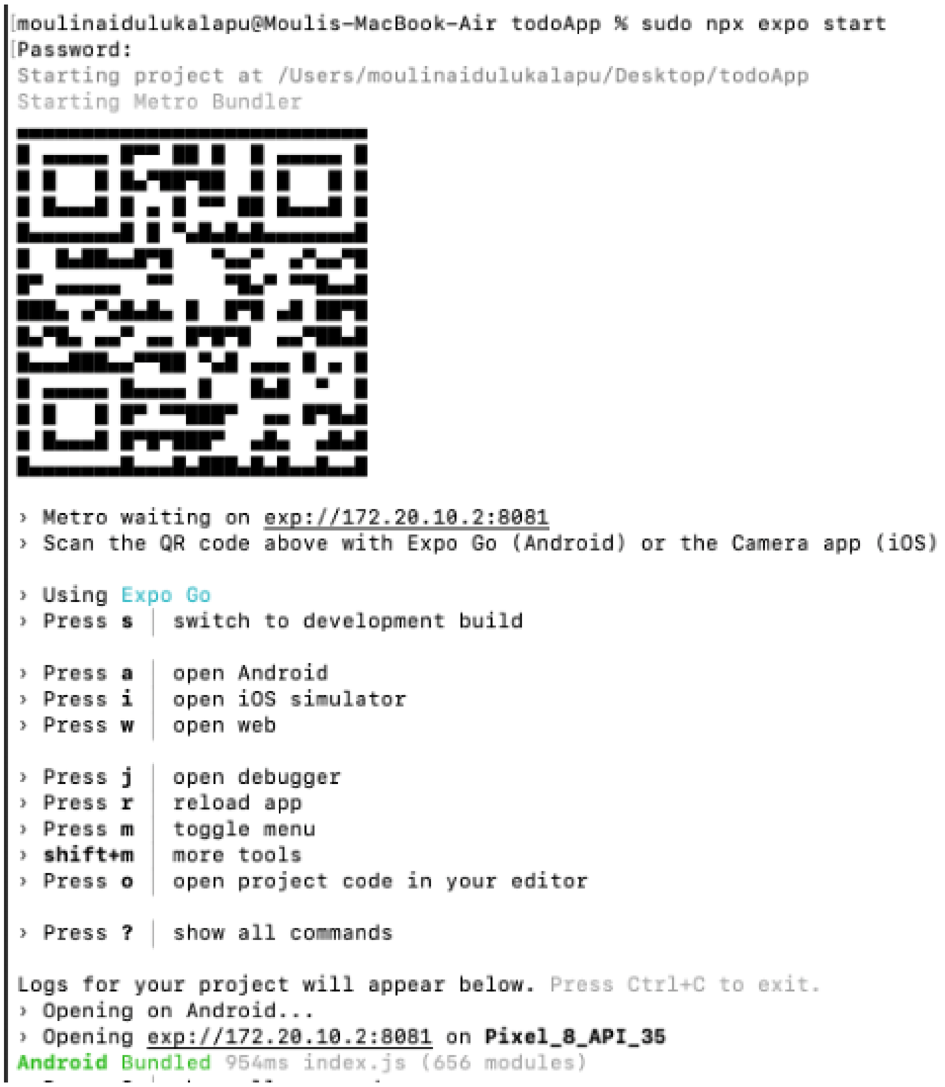
```
moulinaidulukalapu@Moulis-MacBook-Air Desktop % sudo npm install -g expo-cli
Password:
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which
is much more comprehensive and powerful.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/pl
ugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transf
orm-class-properties instead.
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-trans
form-optional-chaining instead.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/
Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated @babel/plugin-proposal-async-generator-functions@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plug
in-transform-async-generator-functions instead.
npm WARN deprecated @babel/plugin-proposal-optional-catch-binding@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-
transform-optional-catch-binding instead.
npm WARN deprecated @babel/plugin-proposal-object-rest-spread@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-tran
sform-object-rest-spread instead.
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-ra
ndom for details.
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-ra
ndom for details.
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-ra
ndom for details.
npm WARN deprecated expo-cli@6.3.10: The global Expo CLI has been superseded by 'npx expo' and eas-cli — Learn more: https://blog.expo.dev/the-new-expo-cli-f4250d8e3421
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic.  See https://v8.dev/blog/math-ra
ndom for details.
npm WARN deprecated osenv@0.1.5: This package is no longer supported.
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated @xmldom/xmldom@0.7.13: this version is no longer supported, please update to at least 0.8.*
npm WARN deprecated rimraf@2.4.5: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@6.0.4: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.1.6: Glob versions prior to v9 are no longer supported

added 1278 packages in 29s

167 packages are looking for funding
  run `npm fund` for details
```

Figure 6: Installing Expo CLI



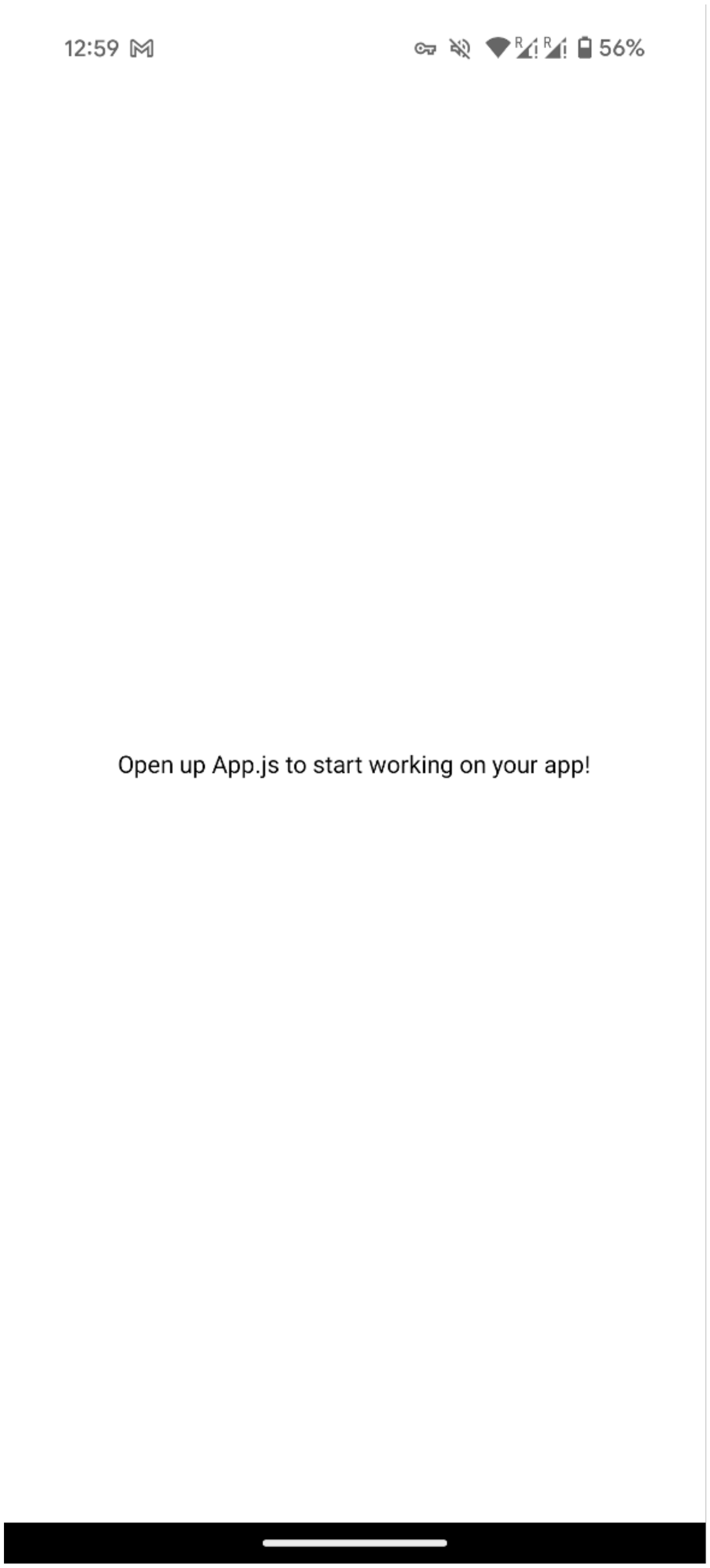
```
moulinaidulukalapu@Moulis-MacBook-Air Desktop % sudo npx expo init todoApp
WARNING: The legacy expo-cli does not support Node +17. Migrate to the new local Expo
 CLI: https://blog.expo.dev/the-new-expo-cli-f4250d8e3421.

  The global expo-cli package has been deprecated.

  The new Expo CLI is now bundled in your project in the expo package.
  Learn more: https://blog.expo.dev/the-new-expo-cli-f4250d8e3421.

  To use the local CLI instead (recommended in SDK 46 and higher), run:
  › npx expo <command>


Migrate to using:
› npx create-expo-app --template

✔ Choose a template: › blank               a minimal app as clean as an empty canvas
✔ Downloaded template.
📦 Using npm to install packages.
✔ Installed JavaScript dependencies.

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm comma
nds.

- cd todoApp
- npm start # you can open iOS, Android, or web from here, or run them directly with
the commands below.
- npm run android
- npm run ios
- npm run web
Project is already inside of a git repo, skipping git init.
```

Figure 7: Creating a new Expo project



```
moulinaidulukalapu@Moulis-MacBook-Air todoApp % sudo npx expo start
Password:
Starting project at /Users/moulinaidulukalapu/Desktop/todoApp
Starting Metro Bundler

› Metro waiting on exp://172.20.10.2:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Using Expo Go
› Press s │ switch to development build

› Press a │ open Android
› Press i │ open iOS simulator
› Press w │ open web

› Press j │ open debugger
› Press r │ reload app
› Press m │ toggle menu
› shift+m │ more tools
› Press o │ open project code in your editor

› Press ? │ show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
› Opening on Android...
› Opening exp://172.20.10.2:8081 on Pixel_8_API_35
Android Bundled 954ms index.js (656 modules)
```

Figure 8: Starting the Expo development server



Open up App.js to start working on your app!

Figure 9: Running a physical device

Task 1

Provide detailed answers to the following questions, including any necessary screenshots:

1. **Screenshots of Your App (5 Points)**
   - Attach screenshots of your app running on an emulator and on a physical Android or iOS device.
   - Describe any differences you observed between running the app on an emulator versus a physical device.
     A: I observed some differences when running todo app between on an emulator(Expo) and on physical device. While the emulator provided a convenient environment for testing and debugging, I noticed some lag, while loading the app and while showing animations, as it simulates hardware. On the other hand, the physical device offered smoother performance and more responsiveness over emulator Although these differences weren't significant in my small app, I think these would have impact on performance on large applications requiring intensive hardware utilization

2. **Setting Up an Emulator**
   - Explain the steps you followed to set up an emulator in Android Studio or Xcode.
     A:I have opened Android Studio Preferences Then, I navigated to Appearance and Behavior - System Settings - Android SDK. In the SDK tab, I made sure to install the correct SDK version, along with the necessary packages like SDK tools, platform tools, and build tools.

     Next, I created a Virtual Device (AVD) by going to Tools - AVD Manager, and clicked Create Virtual Device. I selected a device model (MEDIUMPHONEAP) and clicked Next. After that, I chose a system image that matched my preferred Android version, then clicked Next and clicked Finish to create the AVD.
     Once the AVD was created, I could see it listed in the AVD Manager. I clicked the Play button next to my virtual device to launch the emulator.

   - Discuss any challenges you faced during the setup and how you overcame them.
     I faced an issue with the SDK not having the proper permissions while setting up the Android emulator. To fix it, I ensured that the SDK directories had the correct permissions. I ran the command sudo chmod -R 777 /to/android-sdk/ in the terminal to grant required permissions

3. **Running the App on a Physical Device Using Expo**
   - Describe how you connected your physical device to run the app using Expo.
     A: I have installed the Expo Go app on my physical device and then initialized the React Native app with Expo CLI using the command 'npx expo init simpleToDoApp'. To run application I connected the physical device via USB and scanned the QR code generated by the Expo CLI.
   - Include any troubleshooting steps if you encountered issues.
     I had to run the command with sudo to ensure the necessary permissions were granted for the process to complete successfully.

4. **Comparison of Emulator vs. Physical Device (10 Points)**
   - Compare and contrast using an emulator versus a physical device for React Native development and Discuss the advantages and disadvantages of each option.
     A:Using an emulator for React Native development offers convenience, as it allows quick testing without needing a physical device. It's easy to set up and supports debugging tools, but performance can be slower due to the extra layer of simulation, and some device-specific features like sensors may not work accurately. On the other hand, a physical device provides a more realistic testing environment, with smoother performance and better handling of hardware features like GPS or the camera. However, it requires more setup, such as connecting the device via USB or configuring wireless debugging. Both options have their pros and cons depending on the app's complexity and the need for precise testing.

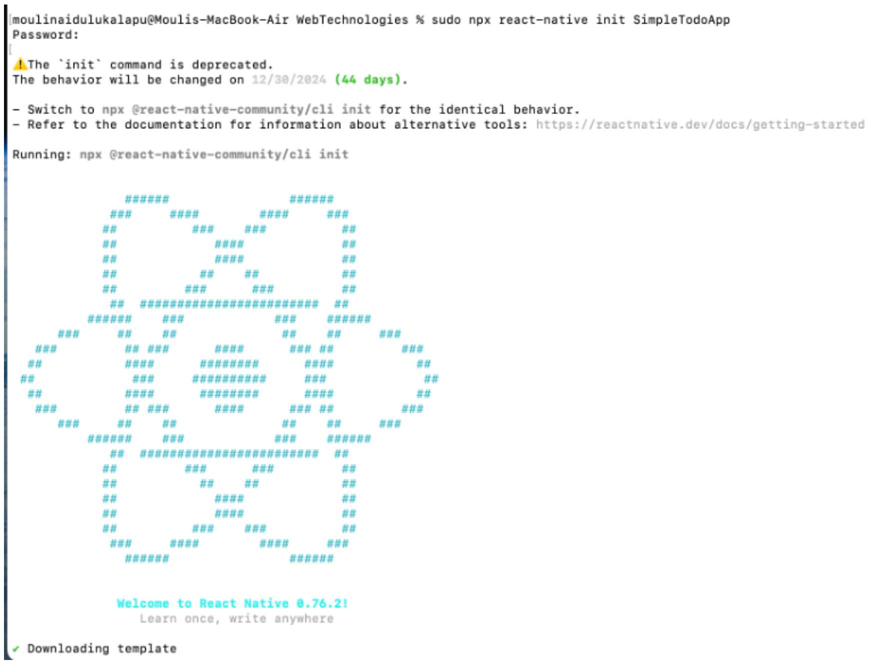
5. **Troubleshooting a Common Error (5 Points)**
   - Identify a common error you encountered when starting your React Native app. Note that it is very unlikely that everyone will get the same error here.
   - Explain the cause of the error and the steps you took to resolve it.
     When starting my React Native app, I encountered an error related to outdated devDependencies, specifically the React Native CLI version. Updating the CLI version in package.json resolved the issue, and the app started without errors after npm install.

Task 2: Building a Simple To-Do List App

2.2 Step 1: Set Up the Project

1. Create and navigate to the new project:

```
npx react-native init SimpleTodoApp
cd SimpleTodoApp
```



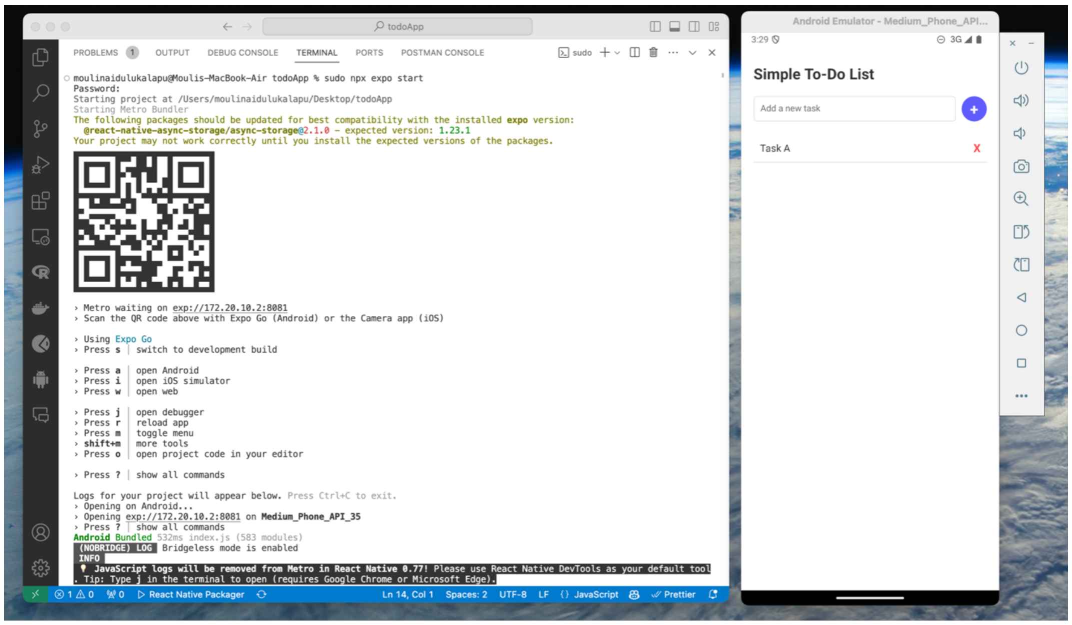
```
moulinaidulukalapu@Moulis-MacBook-Air WebTechnologies % sudo npx react-native init SimpleTodoApp
Password:

⚠️The `init` command is deprecated.
The behavior will be changed on 12/30/2024 (44 days).

- Switch to npx @react-native-community/cli init for the identical behavior.
- Refer to the documentation for information about alternative tools: https://reactnative.dev/docs/getting-started

Running: npx @react-native-community/cli init


               ######                ######
             ###     ####        ####     ###
            ##          ###    ###          ##
            ##             ####             ##
            ##             ####             ##
            ##           ##    ##           ##
            ##         ###      ###         ##
             ##  ########################  ##
          ######    ###            ###    ######
      ###     ##    ##              ##    ##     ###
   ###         ## ###      ####      ### ##         ###
  ##           ####      ########      ####           ##
 ##             ###     ##########     ###             ##
  ##           ####      ########      ####           ##
   ###         ## ###      ####      ### ##         ###
      ###     ##    ##              ##    ##     ###
          ######    ###            ###    ######
             ##  ########################  ##
            ##         ###      ###         ##
            ##           ##    ##           ##
            ##             ####             ##
            ##             ####             ##
            ##          ###    ###          ##
             ###     ####        ####     ###
               ######                ######


                  Welcome to React Native 0.76.2!
                     Learn once, write anywhere

✔ Downloading template
```

Figure 10: Setting up the project

2.6 Step 4: Running the App

1. In your terminal, run:

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



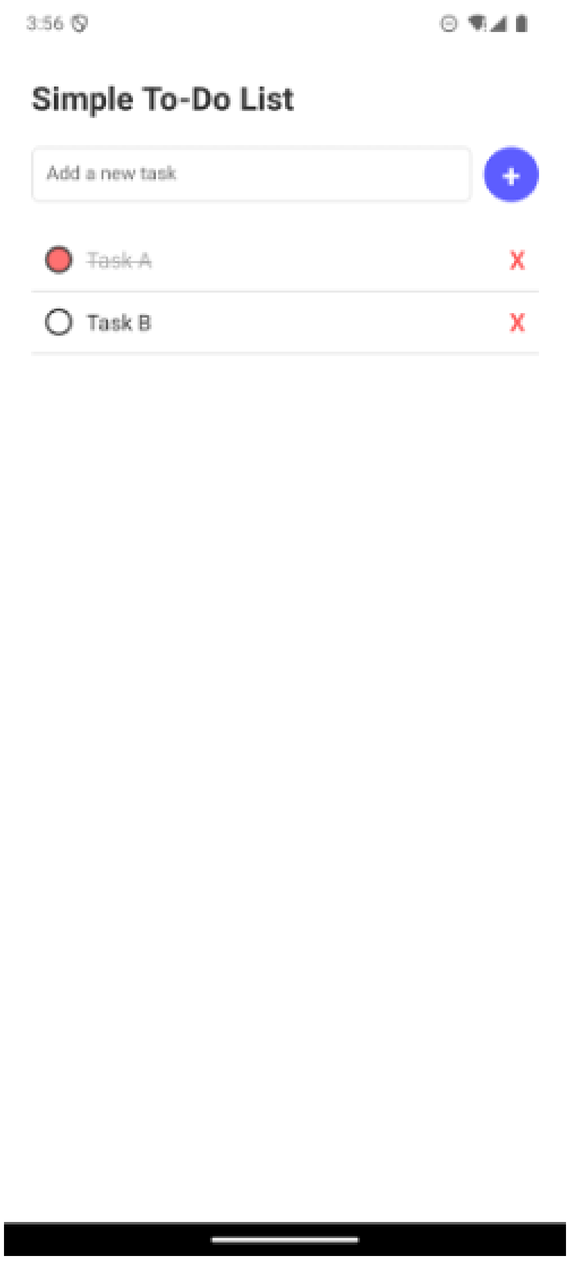
```
moulinaidulukalapu@Moulis-MacBook-Air todoApp % sudo npx expo start
Password:
Starting project at /Users/moulinaidulukalapu/Desktop/todoApp
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
  @react-native-async-storage/async-storage@2.1.0 - expected version: 1.23.1
Your project may not work correctly until you install the expected versions of the packages.

› Metro waiting on exp://172.20.10.2:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Using Expo Go
› Press s │ switch to development build

› Press a │ open Android
› Press i │ open iOS simulator
› Press w │ open web

› Press j │ open debugger
› Press r │ reload app
› Press m │ toggle menu
› shift+m │ more tools
› Press o │ open project code in your editor

› Press ? │ show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
› Opening on Android...
› Opening exp://172.20.10.2:8081 on Medium_Phone_API_35
› Press ? │ show all commands
Android Bundled 532ms index.js (583 modules)
 (NOBRIDGE) LOG  Bridgeless mode is enabled
 INFO
 💡 JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool
. Tip: Type j in the terminal to open (requires Google Chrome or Microsoft Edge).
```

Figure 11: Running the app

2.7 Task 2 Submission

Provide detailed answers to the following questions, including any necessary screenshots:

1. **Mark Tasks as Complete**
   - Add a toggle function that allows users to mark tasks as completed.
   - Style completed tasks differently, such as displaying strikethrough text or changing the text color.

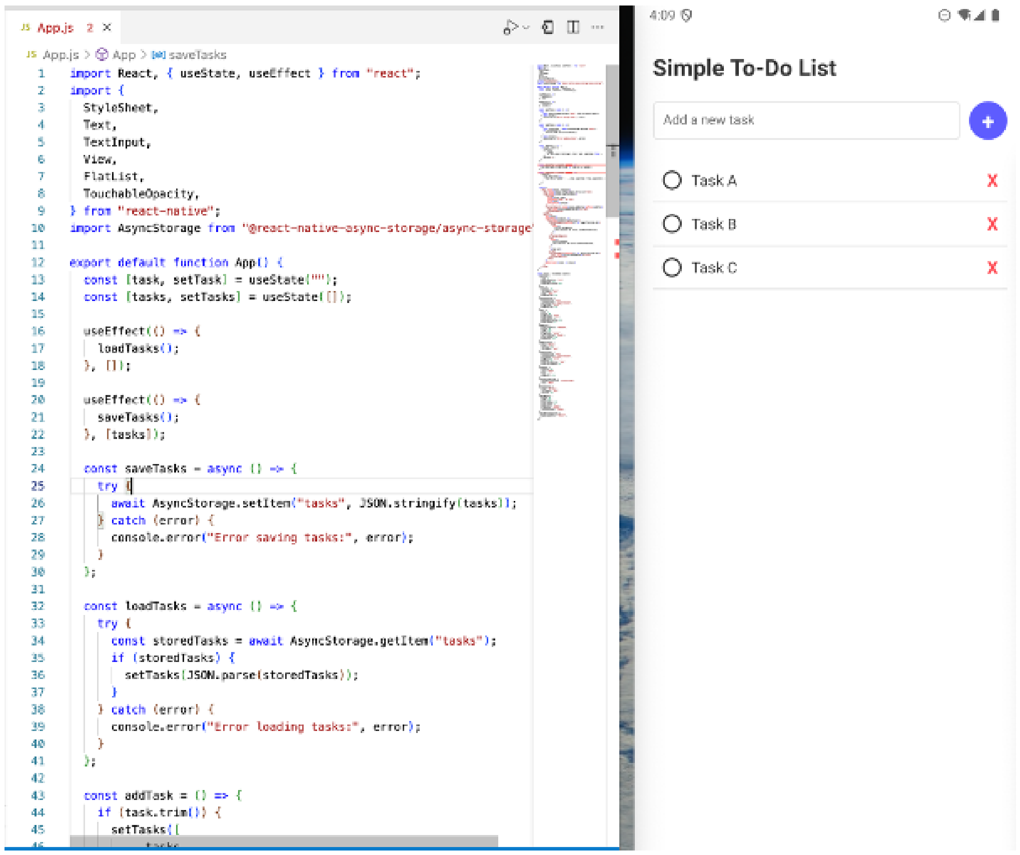


- Explain how you updated the state to reflect the completion status of tasks.

  A: For updating state, the toggleTask function is responsible for marking tasks as complete or incomplete. For this toggleTask iterates through the tasks array and checks if the item.id matches the task ID passed to the function. If they match, a new object is created with the spread operator (...item) to ensure immutability. The completed property of the new object is toggled (!item.completed) based on the current state. The entire tasks array is then updated using setTasks with the mapped array containing the modified task object.
  To show difference between checked and unchecked tasks, task text is styled conditionally using the taskTextCompleted style when completed is true, indicating the task's status.

2. **Persist Data Using AsyncStorage**

   - Implement data persistence so that tasks are saved even after the app is closed.
   - Use AsyncStorage to store and retrieve the tasks list.



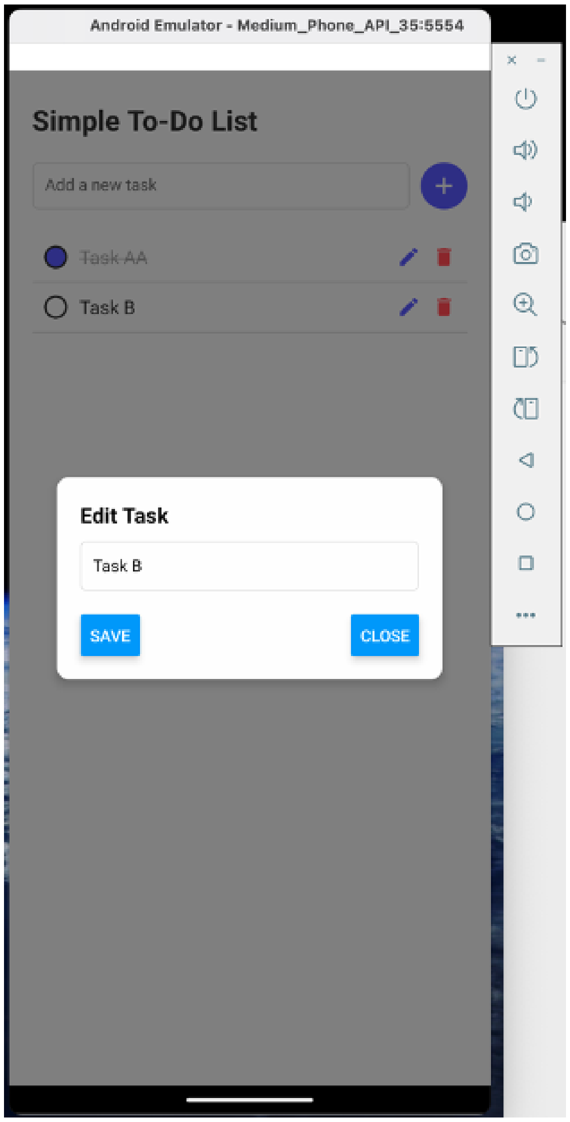
```
import React, { useState, useEffect } from "react";
import {
  StyleSheet,
  Text,
  TextInput,
  View,
  FlatList,
  TouchableOpacity,
} from "react-native";
import AsyncStorage from "@react-native-async-storage/async-storage'

export default function App() {
  const [task, setTask] = useState("");
  const [tasks, setTasks] = useState([]);

  useEffect(() => {
    loadTasks();
  }, []);

  useEffect(() => {
    saveTasks();
  }, [tasks]);

  const saveTasks = async () => {
    try {
      await AsyncStorage.setItem("tasks", JSON.stringify(tasks));
    } catch (error) {
      console.error("Error saving tasks:", error);
    }
  };

  const loadTasks = async () => {
    try {
      const storedTasks = await AsyncStorage.getItem("tasks");
      if (storedTasks) {
        setTasks(JSON.parse(storedTasks));
      }
    } catch (error) {
      console.error("Error loading tasks:", error);
    }
  };

  const addTask = () => {
    if (task.trim()) {
      setTasks([
        tasks
```

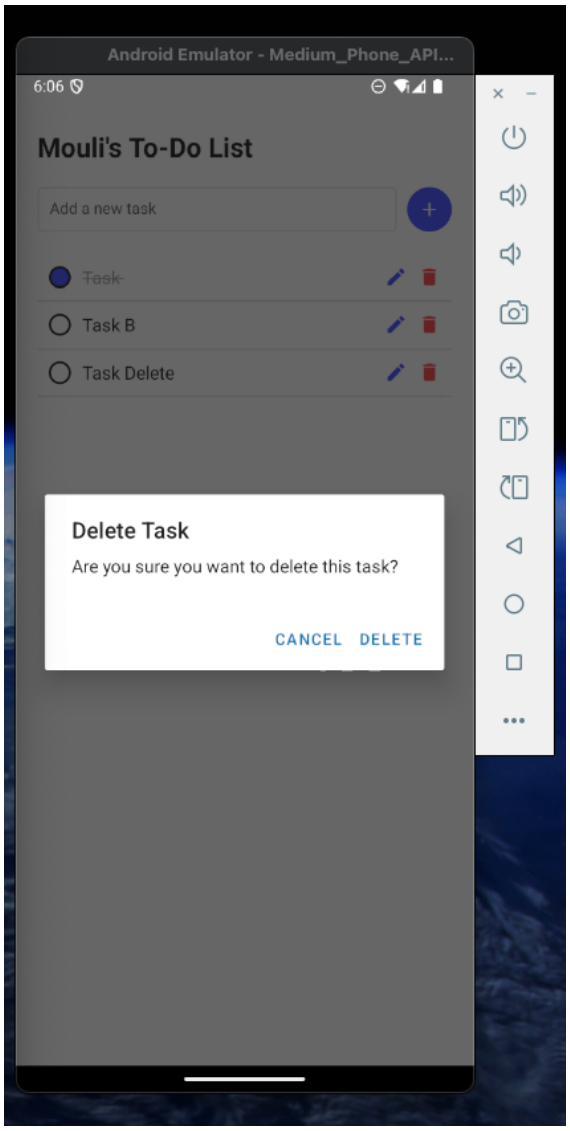
AsyncStorage ensures that the task list is saved and retrieved even after the app is closed. The saveTasks function takes the current tasks state as input, converts it into a JSON string using JSON.stringify, and stores it in the device's storage using AsyncStorage.setItem("tasks", JSON.stringify(tasks)). This process ensures the data is in a format suitable for persistent storage.

On the other hand , the loadTasks function retrieves the stored tasks from AsyncStorage using AsyncStorage.getItem("tasks"). If the tasks are successfully fetched, the function parses the JSON string back into a JavaScript object using JSON.parse and updates the tasks state via setTasks.

3. **Edit Tasks**

   - Allow users to tap on a task to edit its content.
   - Implement an update function that modifies the task in the state array.


Android Emulator - Medium_Phone_API_35:5554
Simple To-Do List
Add a new task
+
~~Task AA~~
Task B
Edit Task
Task B
SAVE
CLOSE



Delete Task

Are you sure you want to delete this task?

CANCEL DELETE

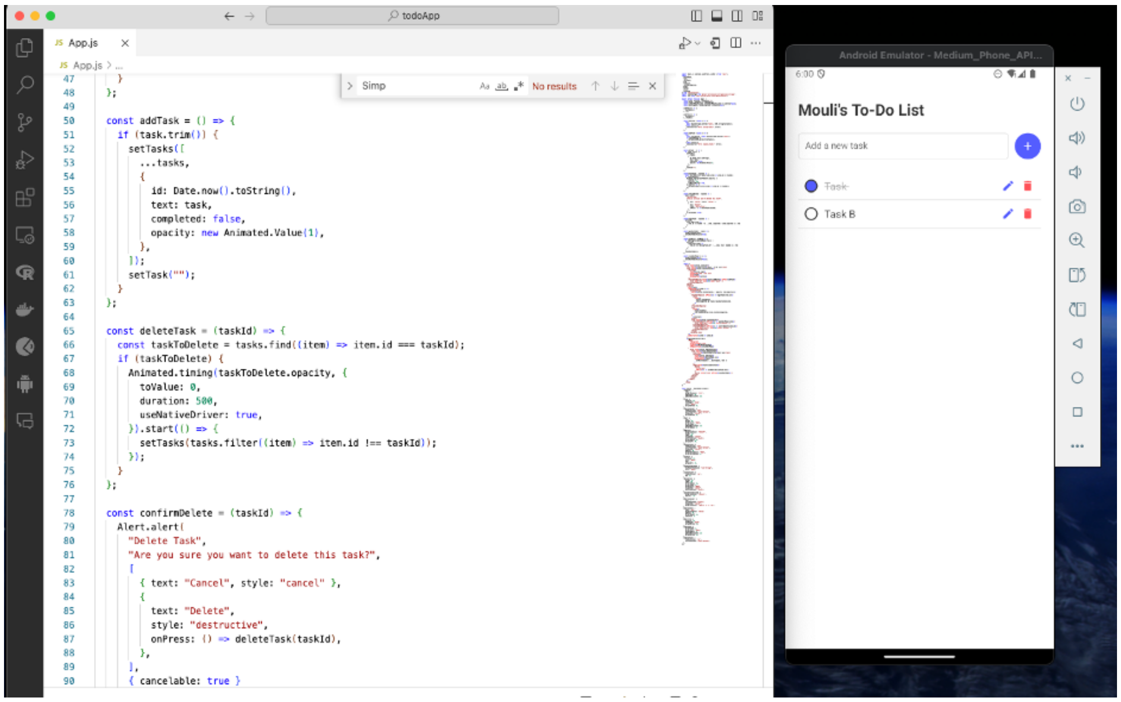
- Explain how you managed the UI for editing tasks.

  A: When user selects edit icon, the modal is conditionally rendered based on the value of isEditModalVisible and includes an input field pre-populated with the task's current text. When user selects Save button it triggers saveEdit method for saving the changes and when close is selected closeEditModal to close the prompt without making any changes.

  The saveEdit function checks if editingTask and newName are valid and the it iterates through tasks with the matching ID and updates the task with the new name using the map method. The spread operator creates a copy of the task to prevent direct mutation, and the text property is updated with the new name. The modified array is set back into the tasks state and openEditModal function takes a task as an argument, sets the editingTask state, and sets isEditModalVisible to true to display the edit modal.

4. **Add Animations**

   - Use the Animated API from React Native to add visual effects when adding or deleting tasks.



```
    }
  };

  const addTask = () => {
    if (task.trim()) {
      setTasks([
        ...tasks,
        {
          id: Date.now().toString(),
          text: task,
          completed: false,
          opacity: new Animated.Value(1),
        },
      ]);
      setTask("");
    }
  };

  const deleteTask = (taskId) => {
    const taskToDelete = tasks.find((item) => item.id === taskId);
    if (taskToDelete) {
      Animated.timing(taskToDelete.opacity, {
        toValue: 0,
        duration: 500,
        useNativeDriver: true,
      }).start(() => {
        setTasks(tasks.filter((item) => item.id !== taskId));
      });
    }
  };

  const confirmDelete = (taskId) => {
    Alert.alert(
      "Delete Task",
      "Are you sure you want to delete this task?",
      [
        { text: "Cancel", style: "cancel" },
        {
          text: "Delete",
          style: "destructive",
          onPress: () => deleteTask(taskId),
        },
      ],
      { cancelable: true }
```

- Describe the animations you implemented and how they enhance user experience.

  A: I have implemented an opacity fade-out animation using the Animated component from React Native. Each task object in the tasks async array has an opacity property initialized with a new Animated.Value(1), representing the starting opacity (fully visible). When the delete operation is called, the deleteTask function uses Animated.timing to animate the opacity of the task to be deleted from 1 (fully visible) to 0 (invisible) over a duration of 500 milliseconds. The animation is started using .start(), and inside the callback, the task is filtered out of the tasks state using setTasks.

Use of Generative AI Tools

I used ChatGPT to assist me in fixing styling issues and troubleshooting problems related to running my React Native app on a physical device. It helped me resolve issues with SDK permissions and provided guidance on setting up and running the app successfully.