

React Native Development Report

Priyal Dharmendra Patel

November 2024

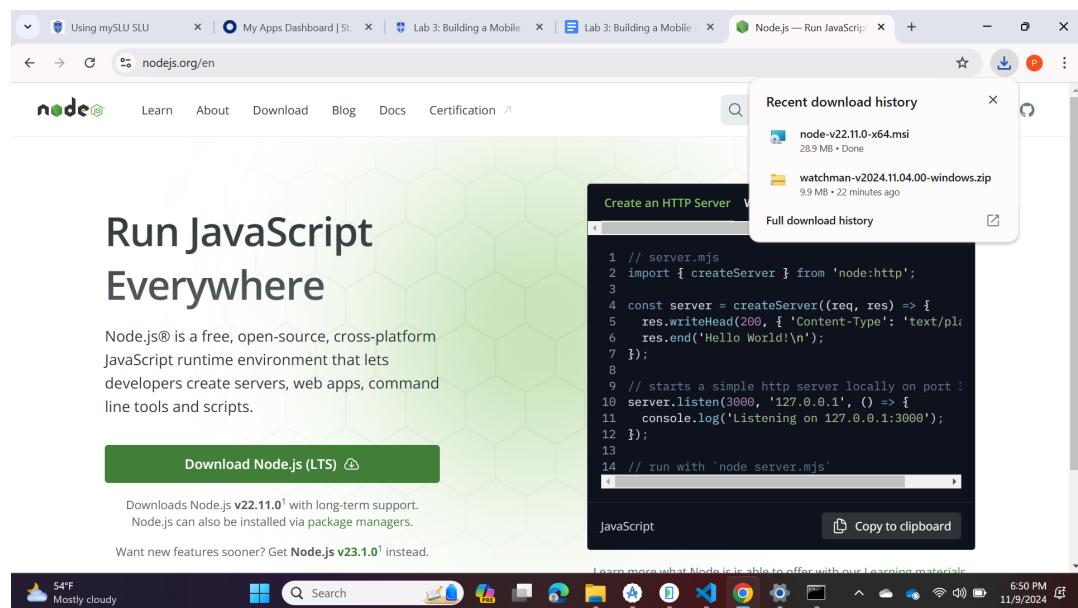
Github:

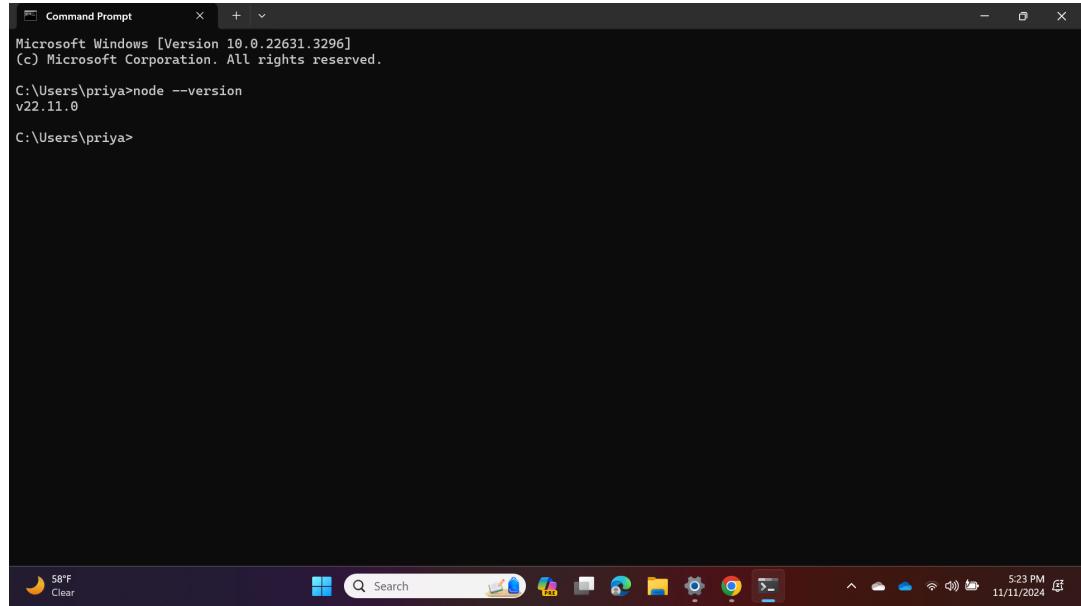
<https://github.com/mepripri/SimpleTodoApp>.

Task 1: Set Up the Development Environment

Step 1: Install Node.js and Watchman

- Install Node.js:





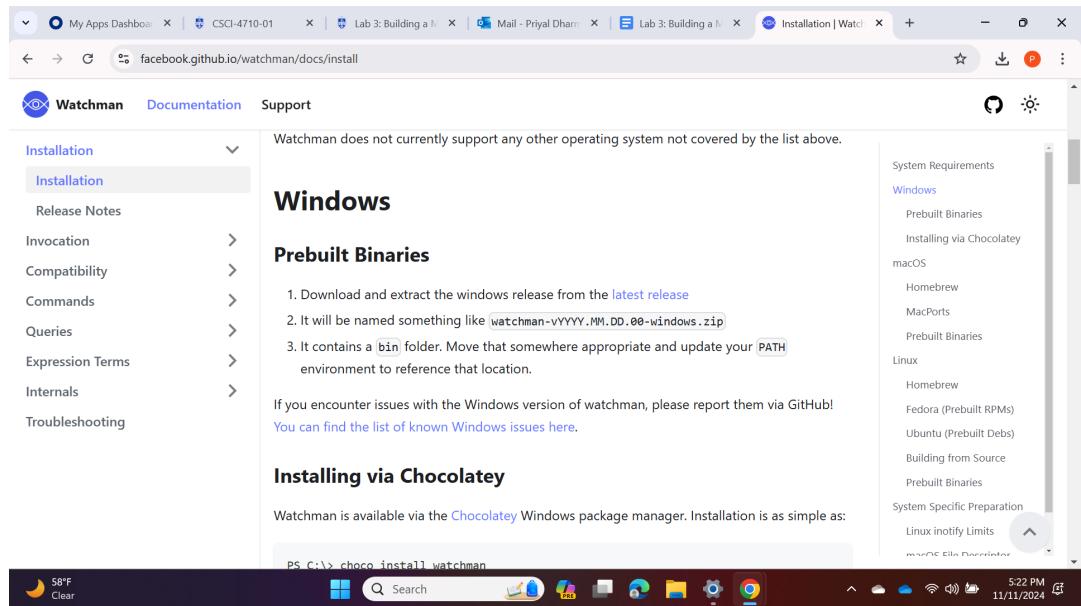
```
Command Prompt      + - X
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\priya>node --version
v22.11.0

C:\Users\priya>
```

58°F Clear 5:23 PM 11/11/2024

- **Install Watchman (Optional for macOS/Linux):**



My Apps Dashboard | CSCI-4710-01 | Lab 3: Building a ... | Mail - Priyal Dham... | Lab 3: Building a ... | Installation | Watchman | +

facebook.github.io/watchman/docs/install

Watchman Documentation Support

Installation

Windows

Prebuilt Binaries

1. Download and extract the windows release from the [latest release](#)
2. It will be named something like `watchman-vYYYY.MM.DD.00-windows.zip`
3. It contains a `bin` folder. Move that somewhere appropriate and update your `PATH` environment to reference that location.

If you encounter issues with the Windows version of watchman, please report them via GitHub! You can find the list of known Windows issues [here](#).

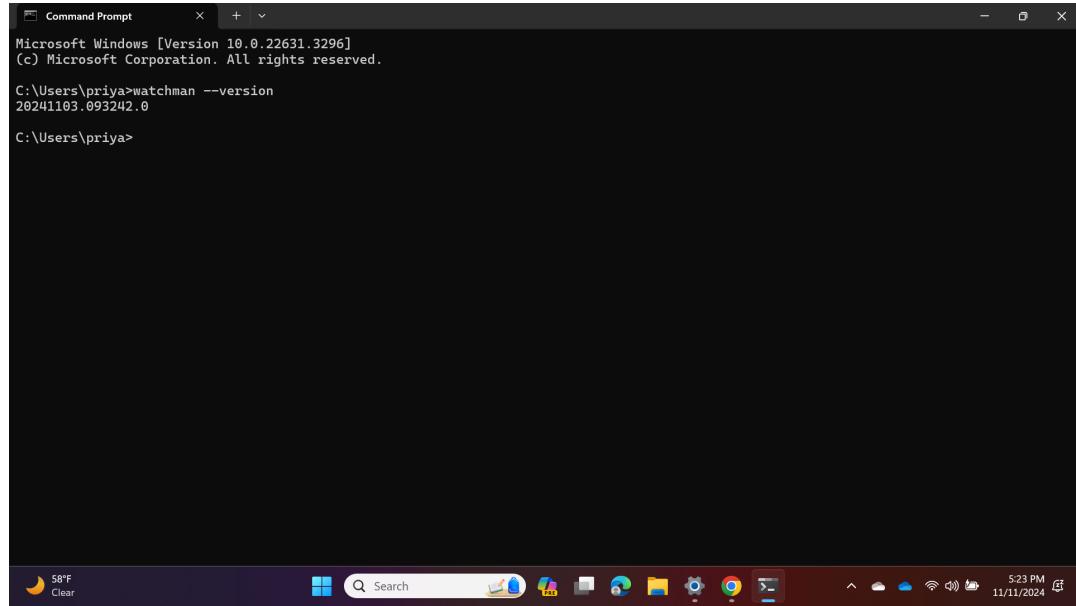
Installing via Chocolatey

Watchman is available via the [Chocolatey](#) Windows package manager. Installation is as simple as:

```
PS C:\> choco install watchman
```

System Requirements
Windows
Prebuilt Binaries
Installing via Chocolatey
macOS
Homebrew
MacPorts
Prebuilt Binaries
Linux
Homebrew
Fedora (Prebuilt RPMs)
Ubuntu (Prebuilt Debs)
Building from Source
Prebuilt Binaries
System Specific Preparation
Linux inotify Limits

58°F Clear 5:22 PM 11/11/2024



```
Command Prompt      X | + | V
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\priya>watchman --version
20241103.093242.0

C:\Users\priya>
```

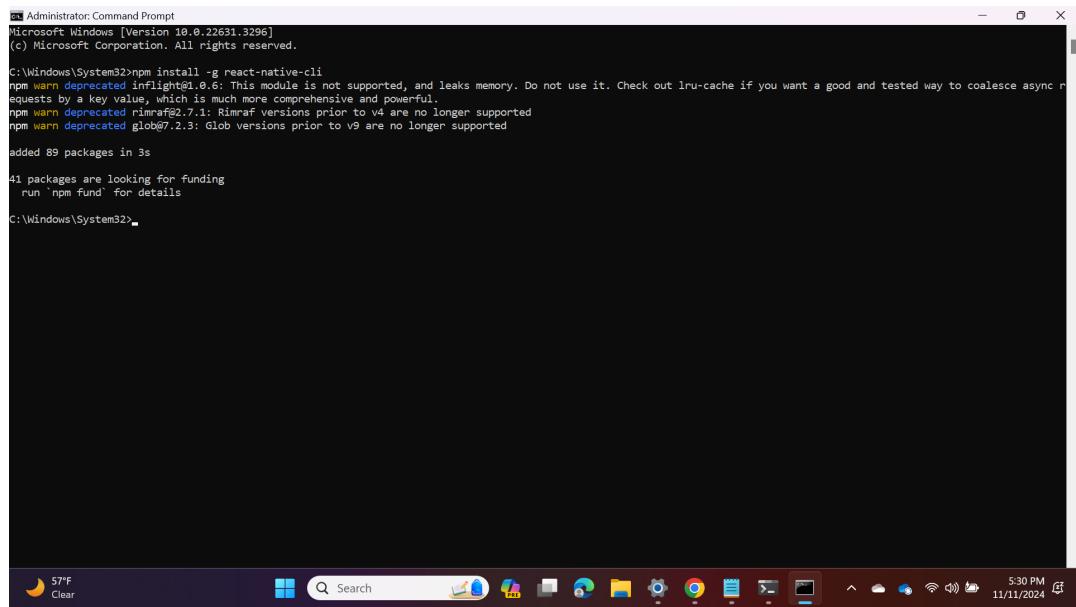
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The title bar includes standard window controls (minimize, maximize, close). The window content displays the command "watchman --version" and its output, which is the version number "20241103.093242.0". The prompt "C:\Users\priya>" is visible at the bottom. The taskbar at the bottom of the screen shows various pinned icons and the system tray with the date and time "11/11/2024 5:23 PM".

Step 2: Install React Native CLI

The React Native CLI allows you to create and manage React Native projects.

- Open your terminal and run:

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



```
Administrator: Command Prompt      X | + | V
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>npm install -g react-native-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 asynchronous requests by a key value, which is much more comprehensive and powerful.
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

C:\Windows\System32>
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The title bar includes standard window controls. The window content displays the command "npm install -g react-native-cli" and its output, which includes several warning messages about deprecated modules like "inflight", "rimraf", and "glob". The prompt "C:\Windows\System32>" is visible at the bottom. The taskbar at the bottom of the screen shows various pinned icons and the system tray with the date and time "11/11/2024 5:30 PM".

- If the global package is deprecated, use the local version with:

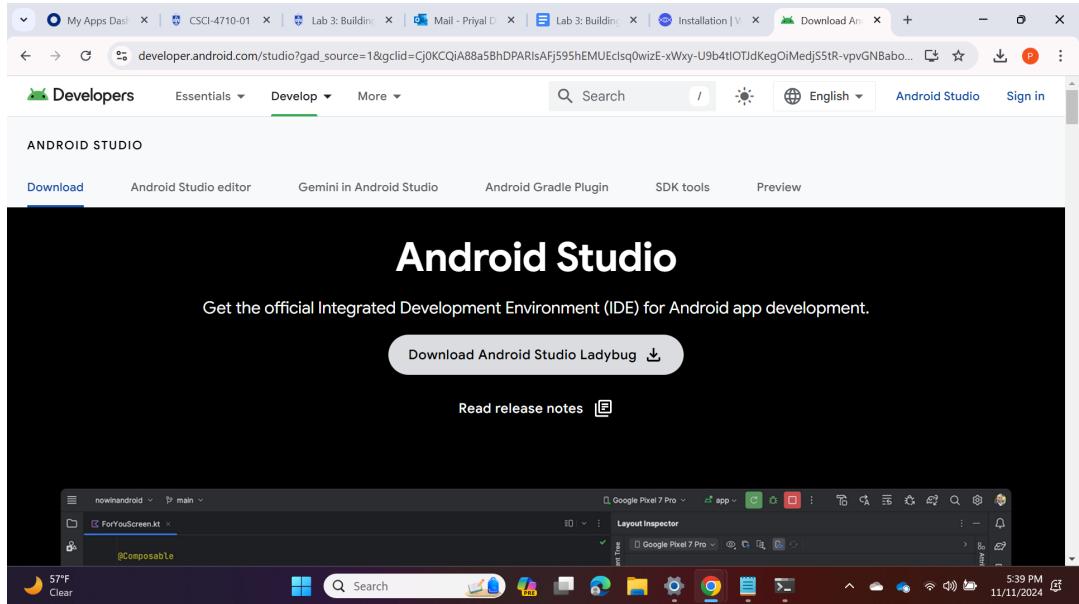
```
npx react-native init YourProjectName
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command entered is "npx react-native init todo". The output shows a large ASCII art logo of a tree made of hashtags (#) and spaces. Below the logo, the text "Welcome to React Native 0.76.1!" and "Learn once, write anywhere" is displayed. The command prompt then lists the steps of the initialization process: "Downloading template", "Copying template", "Processing template", "Installing dependencies", and "Initializing Git repository". The taskbar at the bottom of the screen shows various icons for applications like File Explorer, Edge, and Google Chrome.

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

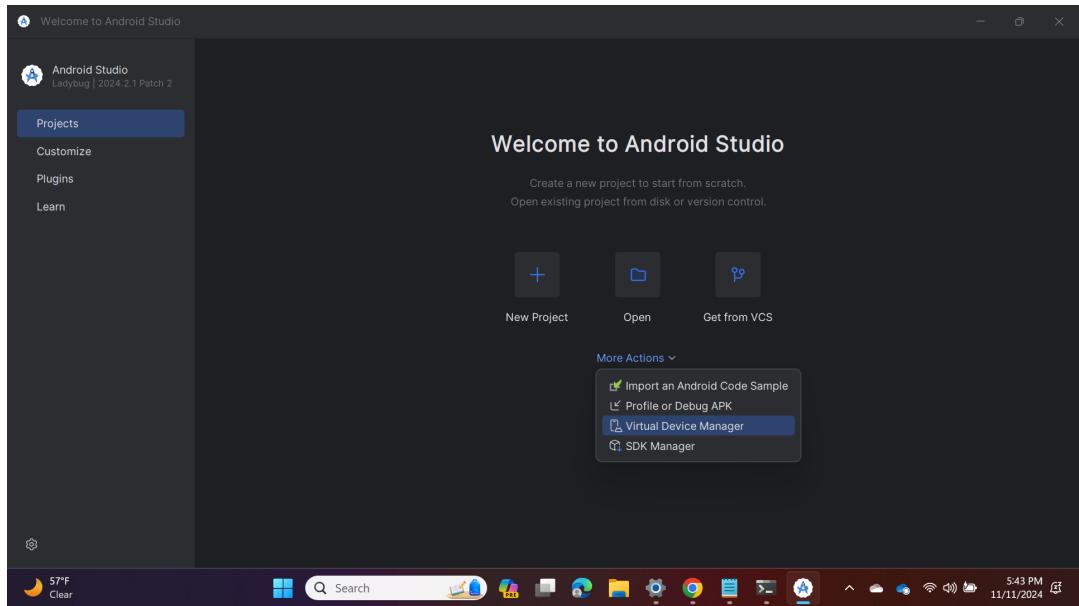
For Android Users:

1. Download and install Android Studio from <https://developer.android.com/studio>.



1. Configure Android Studio:

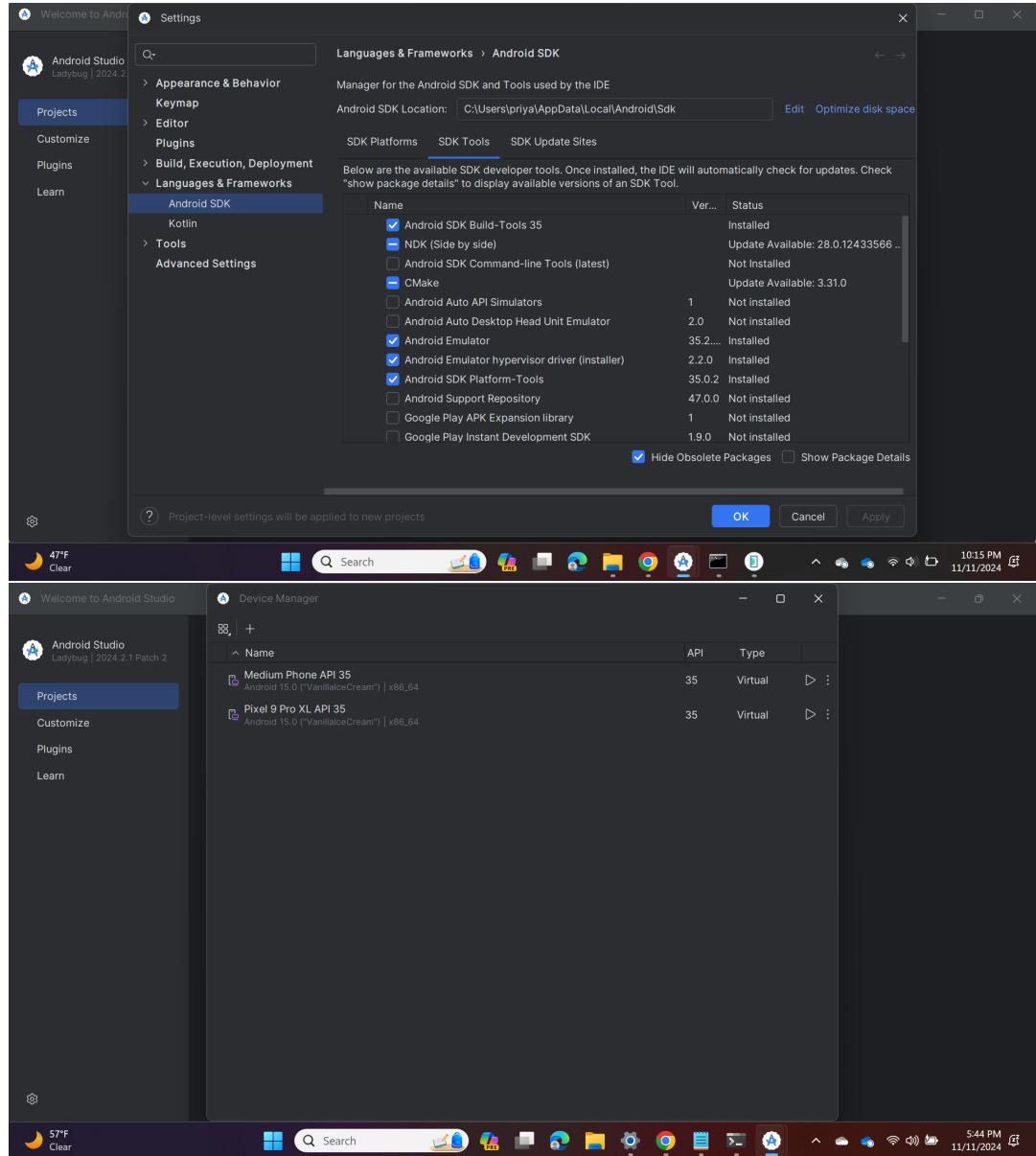
Preferences → Appearance & Behavior → System Settings → Android SDK



1. Enable the following recommended SDK tools:

- Android SDK Build-Tools (latest version)

- Android SDK Platform-Tools
- Android Emulator
- Google Play Services (if required)



Step 4: Create a New React Native Project

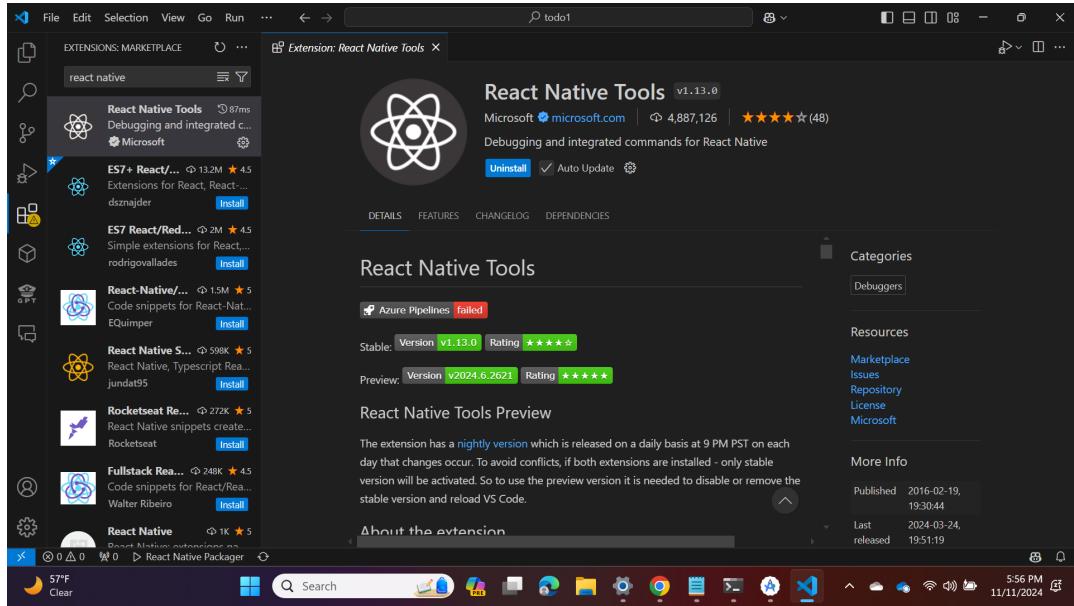
- Run the following commands:

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

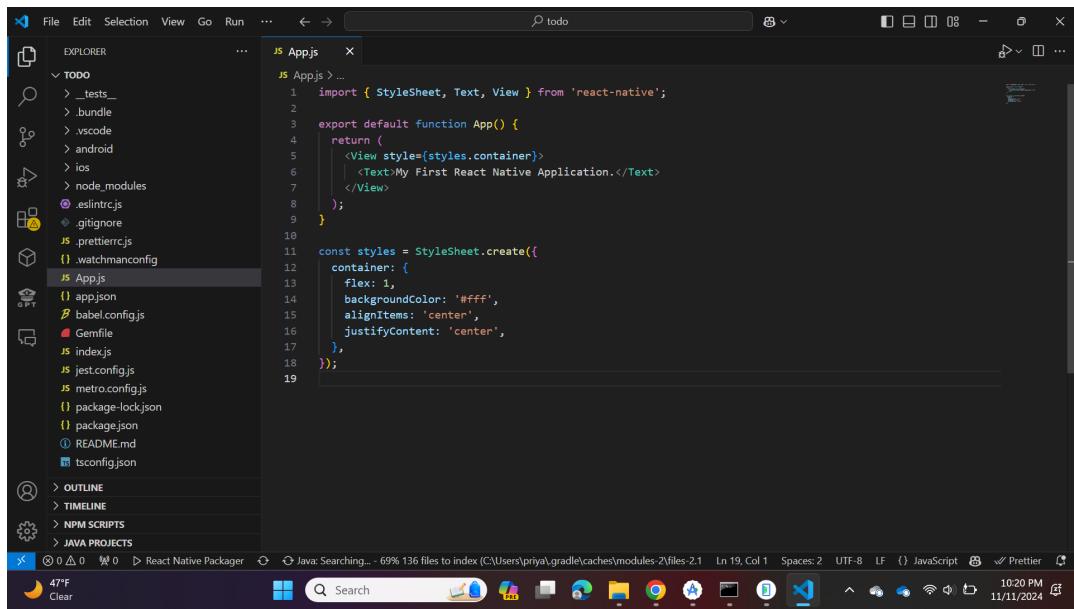
- This creates a new React Native project.

Step 5: Open the Project in Visual Studio Code

- Launch Visual Studio Code.
 - Open the project folder via **File > Open Folder**.
 - Install the React Native Tools extension for enhanced development experience.



- Modify App.js to display "My First React Native Application".



Step 6: Start the Metro Bundler

- Run the following command in the terminal:

```
npx react-native start
```

```

Administrator: C:\WINDOWS\system32\cmd.exe
C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoApp\todo>npx react-native start
(node:5620) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
info Welcome to React Native v0.76
info Starting dev server on port 8081...

Welcome to Metro v0.81.0
Fast - Scalable - Integrated

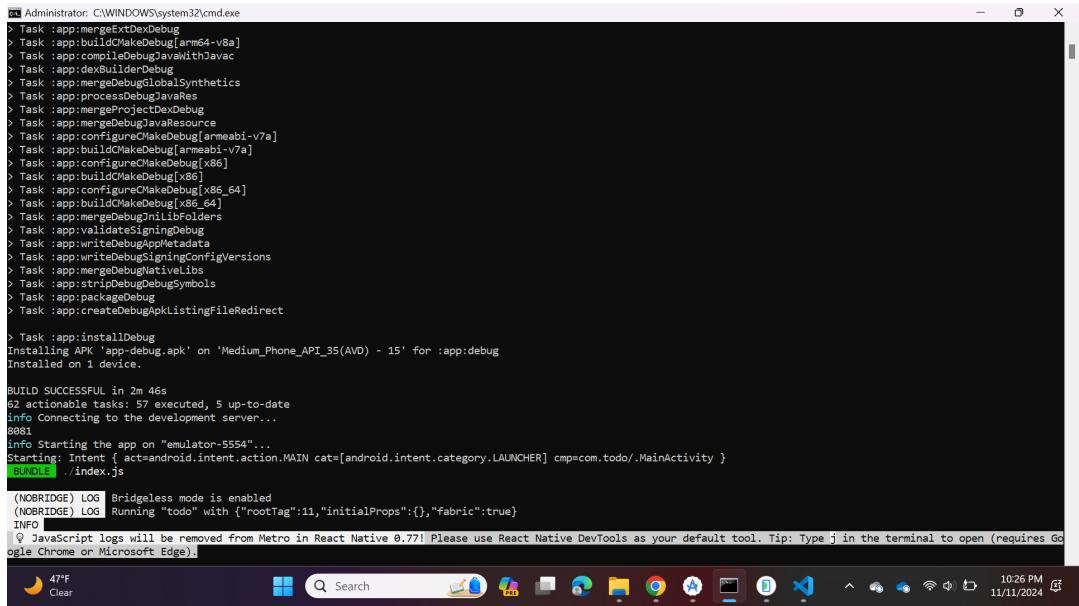
Waiting for Watchman 'query' (10s)...
Waiting for Watchman 'query' (30s)...
Watchman 'query' finished after 46.6s.
info Dev server ready

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

info Opening app on Android...
info A dev server is already running for this project on port 8081.
info Installing the app...
Starting a Gradle Daemon, 2 incompatible and 6 stopped Daemons could not be reused, use --status for details
47°F Clear 10:26 PM 11/11/2024

Administrator: C:\WINDOWS\system32\cmd.exe
> Task :gradle-plugin:shared:classes UP-TO-DATE
> Task :gradle-plugin:shared:jar
> Task :gradle-plugin:settings:plugin:compileKotlin UP-TO-DATE
> Task :gradle-plugin:settings:plugin:compileJava NO-SOURCE
> Task :gradle-plugin:settings:plugin:pluginDescriptors UP-TO-DATE
> Task :gradle-plugin:settings:plugin:processResources
> Task :gradle-plugin:settings:plugin:resources
> Task :gradle-plugin:react-native:gradle-plugin:checkKotlinGradlePluginConfigurationErrors
> Task :gradle-plugin:react-native:gradle-plugin:compileKotlin UP-TO-DATE
> Task :gradle-plugin:react-native:gradle-plugin:compileJava NO-SOURCE
> Task :gradle-plugin:react-native:gradle-plugin:pluginDescriptors UP-TO-DATE
> Task :gradle-plugin:react-native:gradle-plugin:processResources
> Task :gradle-plugin:react-native:gradle-plugin:resources
> Task :gradle-plugin:react-native:gradle-plugin:jar
> Task :app:checkKotlinGradlePluginConfigurationErrors
> Task :app:generateAutoLinkingNewArchitectureFiles
> Task :app:generateAutoLinkingPackageList
> Task :app:generateCodegenSchemaFromJavaScript SKIPPED
> Task :app:generateCodegenArtifactsFromSchema SKIPPED
> Task :app:preBuild
> Task :app:preDebugBuild
> Task :app:generateDebugBuildConfig
> Task :app:checkDebugAarMetadata
> Task :app:generateDebugResValues
> Task :app:mapDebugSourcesSetPaths
> Task :app:processDebugManifest
> Task :app:packageDebugResources
> Task :app:processDebugCompatibleScreenManifests
> Task :app:extractDebugLinksDebug
> Task :app:parseDebugLocalResources
> Task :app:mergeDebugResources
> Task :app:processDebugMainManifest
> Task :app:processDebugManifest
> Task :app:processDebugManifestForPackage
> Task :app:javaPreCompileDebug
> Task :app:mergeDebugShaders
> Task :app:compileDebugShaders NO-SOURCE
> Task :app:generateDebugAssets UP-TO-DATE
> Task :app:mergeDebugAssets
47°F Clear 10:26 PM 11/11/2024

```



```
Administrator: C:\WINDOWS\system32\cmd.exe
> Task :app:mergeExeDebug
> Task :app:buildIdMakeDebug[arm64-v8a]
> Task :app:compileDebugJavaWithJavac
> Task :app:dexBuilderDebug
> Task :app:mergeDebugGlobalSynthetics
> Task :app:processDebugJavaRes
> Task :app:mergeProjectExeDebug
> Task :app:copyDebugResources
> Task :app:configureMakeDebug[armeabi-v7a]
> Task :app:buildIdMakeDebug[armeabi-v7a]
> Task :app:configureMakeDebug[x86]
> Task :app:buildIdMakeDebug[x86]
> Task :app:configureMakeDebug[x86_64]
> Task :app:buildIdMakeDebug[x86_64]
> Task :app:mergeDebugJniLibFolders
> Task :app:validateSigningDebug
> Task :app:writeDebugAppMetadata
> Task :app:writeDebugSigningConfigVersions
> Task :app:mergeDebugNativeLibs
> Task :app:stripDebugDebugSymbols
> Task :app:packageDebug
> Task :app:createDebugApkListingFileRedirect
> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Medium_Phone_API_35(AVD) - 15' for ':app:debug'
Installed 1 device.

BUILD SUCCESSFUL in 2m 46s
62 actionable tasks: 57 executed, 5 up-to-date
Info Connecting to the development server...
9081
Info Starting the app on "emulator-5554"...
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.todo/.MainActivity }
BUNDLE ./index.js

(NOBRIDGE) LOG Bridgeless mode is enabled
(NOBRIDGE) LOG Running "todo" with {"rootTag":11,"initialProps":{},"Fabric":true}
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).]

10:26 PM 11/11/2024
```

- This starts the Metro Bundler, which monitors your files and serves JavaScript code.

Step 7: Run Your App on an Emulator or Device

For Android:

1. Ensure an emulator is running via Android Studio's AVD Manager, or connect a physical device with USB debugging enabled.

```

Administrator: C:\WINDOWS\system32\cmd.exe
> Task :app:mergeExeDebug
> Task :app:buildId[MakeDebug[armeabi-v8a]]
> Task :app:compileDebugJavaWithJavac
> Task :app:dexBuilderDebug
> Task :app:mergeDebugGlobalSynthetics
> Task :app:processDebugJavaRes
> Task :app:mergeProjectExeDebug
> Task :app:copyDebugResources
> Task :app:configureMakeDebug[armeabi-v7]
> Task :app:buildId[MakeDebug[armeabi-v7]]
> Task :app:buildId[MakeDebug[x86]]
> Task :app:buildId[MakeDebug[x86_64]]
> Task :app:buildId[MakeDebug[x86_64]]
> Task :app:mergeDebugJniLibFolders
> Task :app:validateSigningDebug
> Task :app:writeDebugAppMetadata
> Task :app:writeDebugSigningConfigVersion
> Task :app:mergeDebugNativeLibs
> Task :app:stripDebugDebugSymbols
> Task :app:packageDebug
> Task :app:createDebugApkListingFileRedirect
> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Medium_Phone_API_35(AVD) - 15' for :app:debug
Installed 1 device.

BUILD SUCCESSFUL in 2m 46s
62 actionable tasks: 57 executed, 5 up-to-date
Info Connecting to the development server...
8981
Info Starting the app on "emulator-5554"...
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.todo/.MainActivity }
BUNDLE ./index.js

(NOBRIDGE) LOG Bridgeless mode is enabled
(NOBRIDGE) LOG Running "todo" with {"rootTag":11,"initialProps":{},"fabric":true}
INFO
JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool
ogle Chrome or Microsoft Edge.

```

1. Run:

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

```

Administrator: C:\WINDOWS\system32\cmd.exe
> Task :app:buildId[MakeDebug[x86_64]]
> Task :app:mergeDebugNativeLibs UP-TO-DATE
> Task :app:stripDebugDebugSymbols UP-TO-DATE
> Task :app:validateSigningDebug UP-TO-DATE
> Task :app:writeDebugAppMetadata UP-TO-DATE
> Task :app:writeDebugSigningConfigVersions UP-TO-DATE
> Task :app:packageDebug UP-TO-DATE
> Task :app:createDebugApkListingFileRedirect UP-TO-DATE

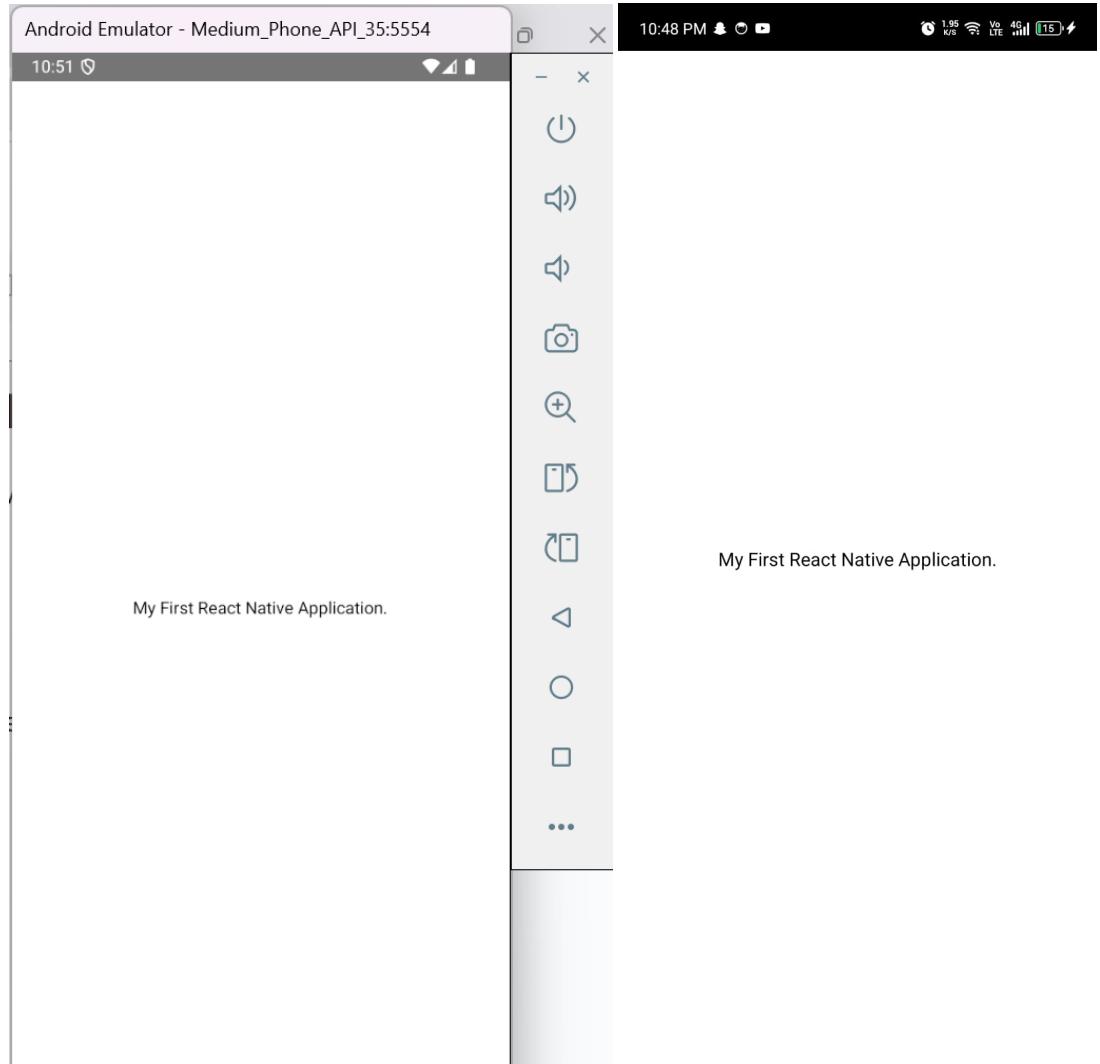
> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Medium_Phone_API_35(AVD) - 15' for :app:debug
Installing APK 'app-debug.apk' on 'Infinix X6731 - 14' for :app:debug
Installed on 2 devices.

BUILD SUCCESSFUL in 16s
62 actionable tasks: 13 executed, 49 up-to-date
Info Connecting to the development server...
8981
Info Starting the app on "10619313AK014335"...
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.todo/.MainActivity }
Info Connecting to the development server...
Info Starting the app on "emulator-5554"...
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.todo/.MainActivity }
BUNDLE ./index.js

(NOBRIDGE) LOG Bridgeless mode is enabled
(NOBRIDGE) LOG Running "todo" with {"rootTag":11,"initialProps":{},"fabric":true}
INFO
JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool
ogle Chrome or Microsoft Edge.
(NOBRIDGE) LOG Bridgeless mode is enabled
(NOBRIDGE) LOG Running "todo" with {"rootTag":21,"initialProps":{},"fabric":true}
(NOBRIDGE) LOG Running "todo" with {"rootTag":31,"initialProps":{},"fabric":true}
INFO
JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool
ogle Chrome or Microsoft Edge.

```

1. This installs and launches the app on the selected device.



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

1. Install Expo CLI and create a new project:

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

```

Administrator: Command Prompt
C:\Windows\System32>cd C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI
C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI>npx expo init todo
WARNING: The legacy expo-cli does not support Node v17. Migrate to the new local Expo CLI: https://blog.expo.dev/the-new-expo-cli-f4250d8e3421.
(node:22932) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)

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 Yarn to install packages. Pass --npm to use npm instead.
✓ Installed JavaScript dependencies.

 Your project is ready!

To run your project, navigate to the directory and run one of the following yarn commands.

- cd todo
- yarn start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- yarn android
- yarn ios # requires an iOS device or macOS for access to an iOS simulator
- yarn web

C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI>

48°F Clear Search 10:57 PM 11/11/2024
Administrator: Command Prompt
C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI>cd todo
C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI\todo>npx expo start
Starting project at C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI\todo
(node:25084) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
Port 8081 is being used by another process.
Use port 8082 instead? ... yes
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
  expo@51.0.38 - expected version ~51.0.39
Your project may not work correctly until you install the expected versions of the packages.

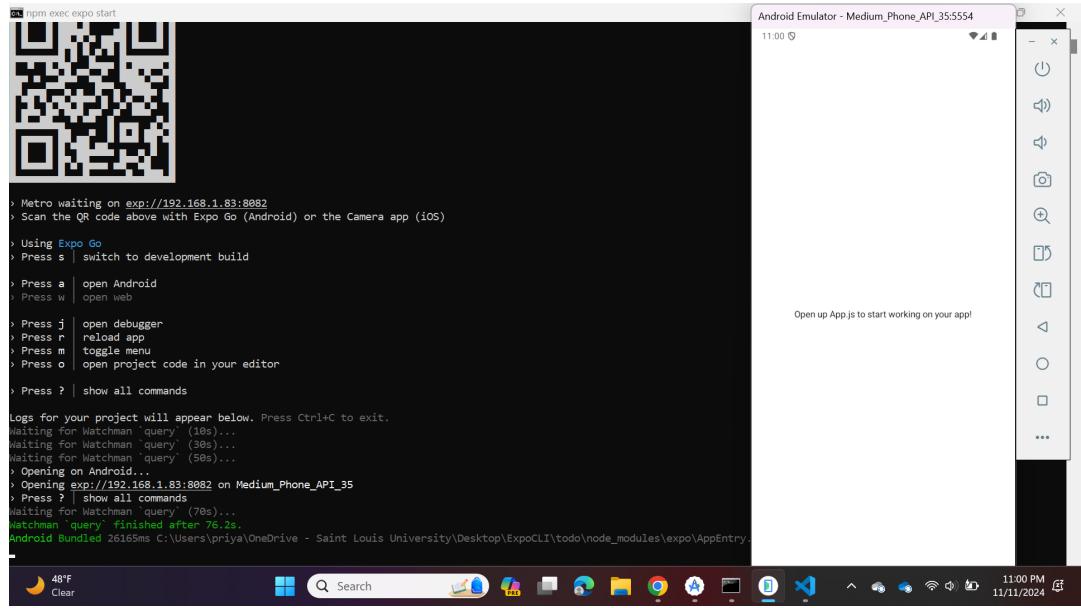
QR code:

> Metro waiting on exp://192.168.1.83:8082
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

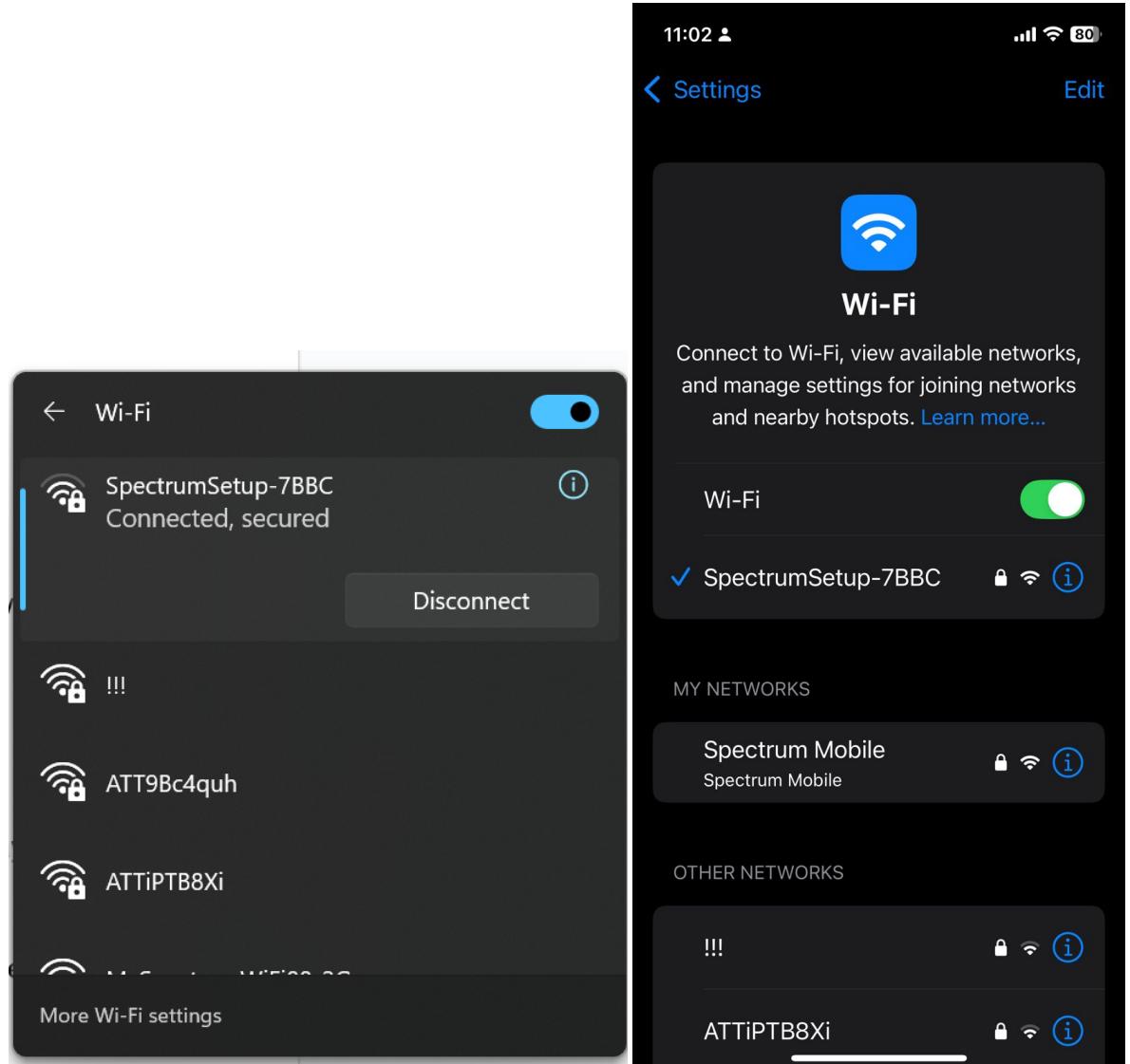
Using Expo Go
> Press s | switch to development build
> Press a | open Android
> Press w | open web
> Press j | open debugger
> Press r | reload app
> Press m | toggle menu

48°F Clear 10:58 PM 11/11/2024

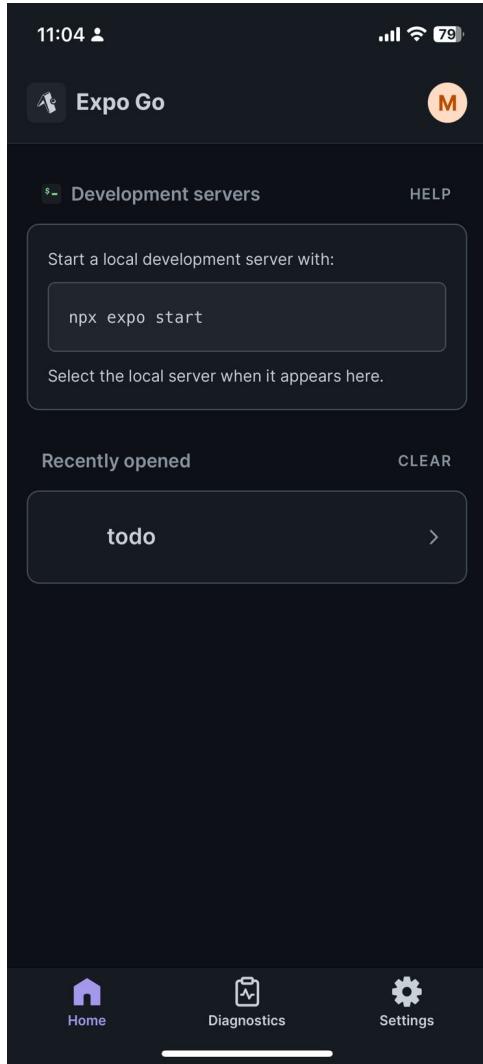
```



1. Connect your mobile device to the same Wi-Fi network.



1. Install the Expo Go app from the App Store or Google Play Store.



1. Scan the QR code provided in the Expo developer tools to launch the app.

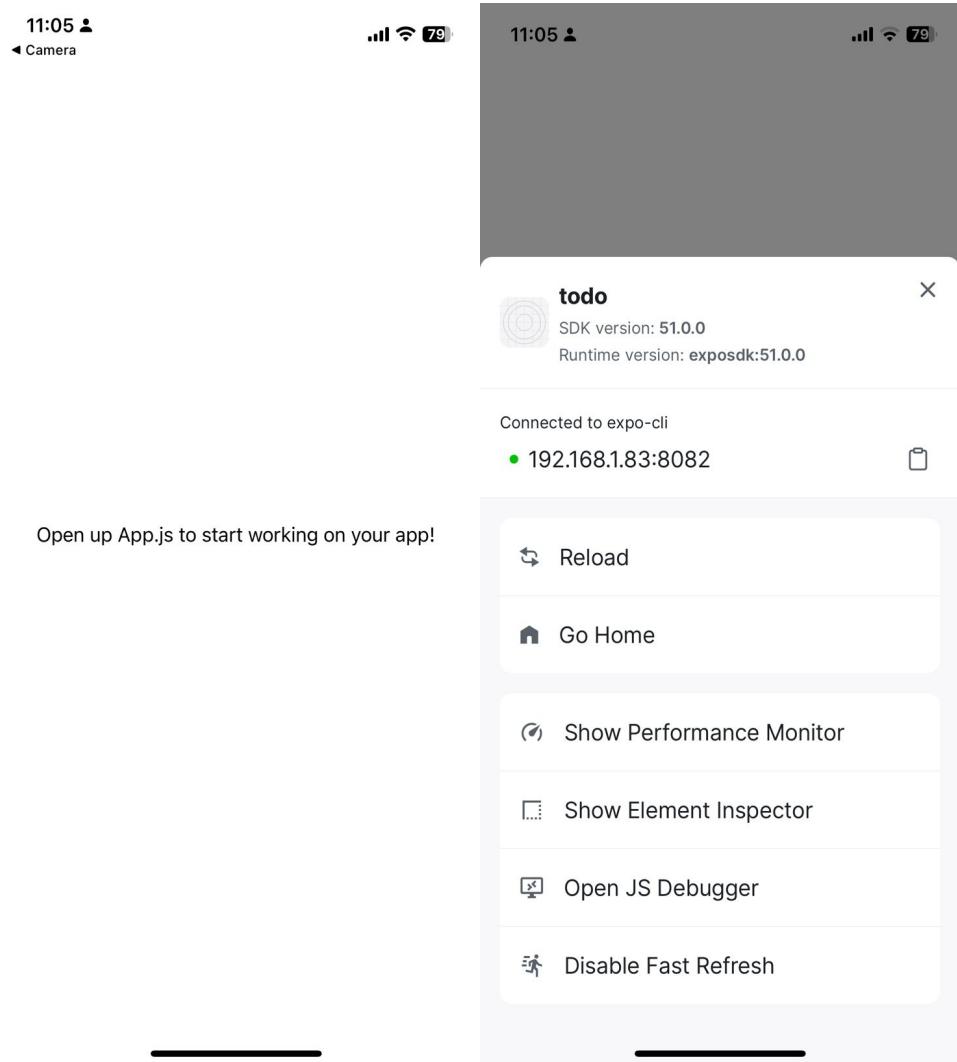
```
cd npm exec expo start

> Metro waiting on exp://192.168.1.83:8082
> 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 w | open web
> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> Press o | open project code in your editor
> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
Waiting for Watchman 'query' (10s)... .
Waiting for Watchman 'query' (30s)... .
Waiting for Watchman 'query' (50s)... .
> Opening on Android.
> Opening exp://192.168.1.83:8082 on Medium_Phone_API_35
> Press ? | show all commands
Waiting for Watchman 'query' (70s)... .
watchman 'query' finished after 76.2s.
Android Bundled 26165ms C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI\todo\node_modules\expo\AppEntry.js (648 modules)
iOS Bundled 3913ms C:\Users\priya\OneDrive - Saint Louis University\Desktop\ExpoCLI\todo\node_modules\expo\AppEntry.js (645 modules)

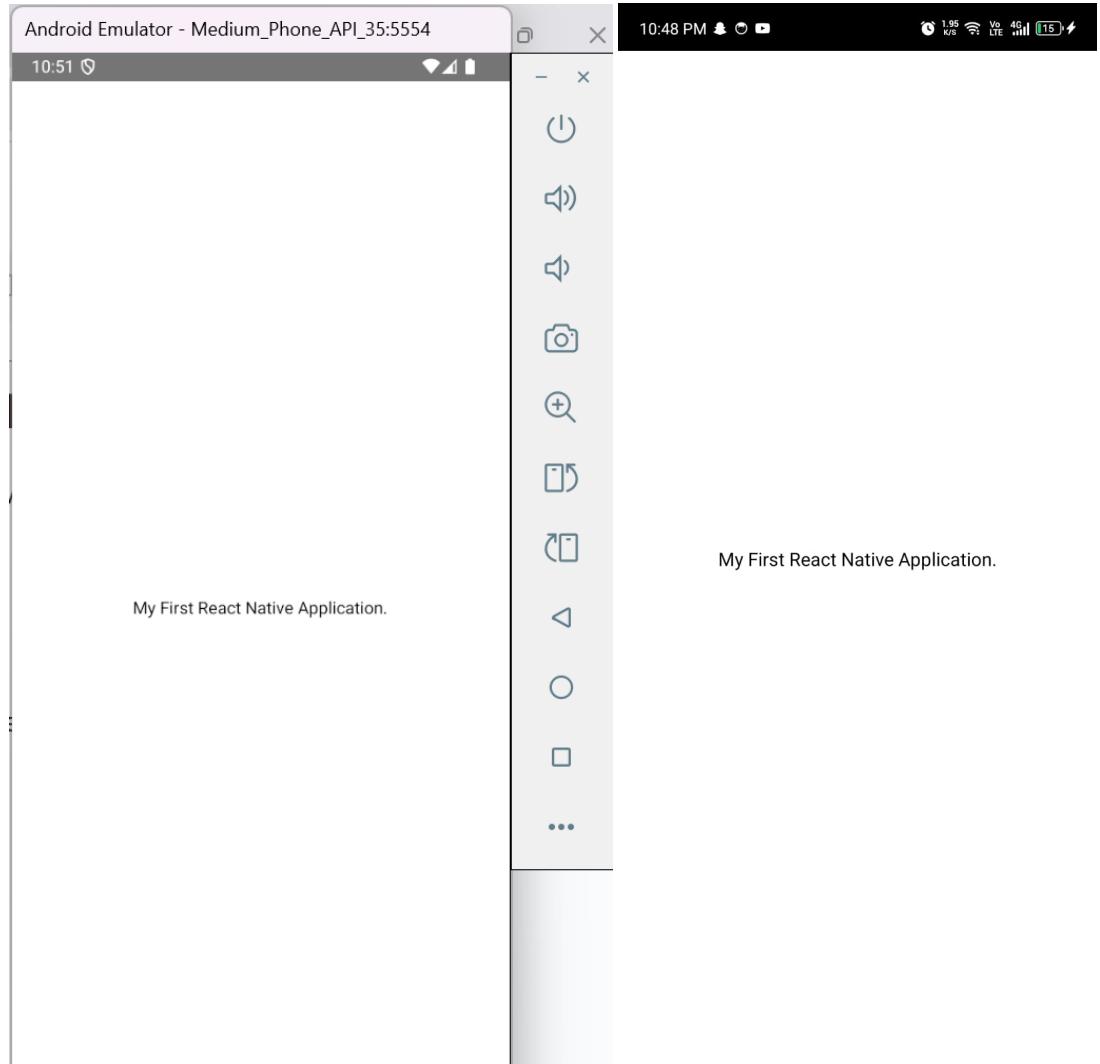
48°F Clear 11:05 PM 11/11/2024
```



What to Submit for Task 1 (Total 40 Points)

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.

When running the app on an emulator, it felt slower compared to a physical device, which was much faster. The graphics on the emulator sometimes lagged or looked slightly different, while the physical device displayed the UI more accurately. Also, using the emulator meant relying on a mouse and keyboard for gestures, but the physical device allowed for real touch interactions, making it more intuitive.

2. Setting Up an Emulator (10 Points)

- Explain the steps you followed to set up an emulator in Android Studio or Xcode.
 - I opened Android Studio and went to the AVD Manager.
 - I created a virtual device by selecting a phone model (e.g., Pixel 4).
 - Then, I downloaded the Android version I needed and set the emulator's settings like RAM and storage.
 - After that, I clicked the play button to launch the emulator.
- Discuss any challenges you faced during the setup and how you overcame them.
 - The emulator was really slow.
 - I also had trouble with a specific Android version.

3. Running the App on a Physical Device Using Expo (10 Points)

- Describe how you connected your physical device to run the app using Expo.
 - I downloaded the Expo Go app on my phone.
 - I made sure my phone and computer were on the same Wi-Fi.
 - I started the app on my computer using expo start and scanned the QR code with my phone's Expo Go app.
- Include any troubleshooting steps if you encounter issues.
 - Once, my phone wasn't on the same Wi-Fi as my computer, so I connected them to the same network, and it worked fine.
 - Sometimes, the QR code wouldn't work, so I copied the link from the terminal and opened it manually in the Expo Go app.

4. Comparison of Emulator vs. Physical Device (10 Points)

- Compare and contrast using an emulator versus a physical device for React Native development.
 - Emulator:
 - * Slower and sometimes lags.
 - * Simulates touch with a mouse, which isn't natural.
 - * Easy to set up and test on different virtual devices.

- Physical Device:
 - * Faster and shows real-world performance.
 - * Lets you test real touch gestures.
 - * Takes more effort to set up but gives accurate results.
- Discuss the advantages and disadvantages of each option.
 - Advantages:
 - * Easy to set up and use.
 - * Runs faster and shows exactly how users will experience the app.
 - * Can test on lots of "fake" devices with different screen sizes.
 - Disadvantages:
 - * Slower and sometimes laggy.
 - * Takes more effort to set up.
 - * Harder to test on different types of devices unless you have several phones.

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.

I faced dependencies error where my jdk wasn't compatible with the running react-native version. I had to run npx react-native doctor to find the loose dependencies.

- Explain the cause of the error and the steps you took to resolve it.
- This caused due to incompatible jdk version. To solve this I download jdk-17 restarted the computer and ran the run-android command again to start the app.

Task 2: Building a Simple To-Do List App

Step 1: Set Up the Project

- Create a new project:

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

- Open the project in VS Code:

code .

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a project structure with files like `TODO`, `_tests_`, `.bundle`, `.vscode`, `android`, `ios`, `node_modules`, `.eslintrc.js`, `.gitignore`, `.prettierrc.js`, and `.watchmanconfig`.
- Code Editor:** The main editor window displays the `App.js` file with the following code:

```
JS App.js x
JS App.js > todo
1 import { StyleSheet, Text, View } from 'react-native';
2
3 export default function App() {
4   return (
5     <View style={styles.container}>
6       <Text>My First React Native Application.</Text>
7     </View>
8   );
9 }
10 const styles = StyleSheet.create({
11   container: {
12     flex: 1,
13     backgroundColor: '#fff',
14     alignItems: 'center',
15     justifyContent: 'center',
16   },
17 });
18
19 }
```
- Status Bar:** At the bottom, it shows "Java: Ready", "Ln 17, Col 5", "Spaces: 2", "UTF-8", "LF", "JavaScript", "Prettier", "106 PM", and the date "11/13/2024".

Step 2: Create the Basic To-Do List Structure

Replace the content of `App.js` with a basic structure for the To-Do list.

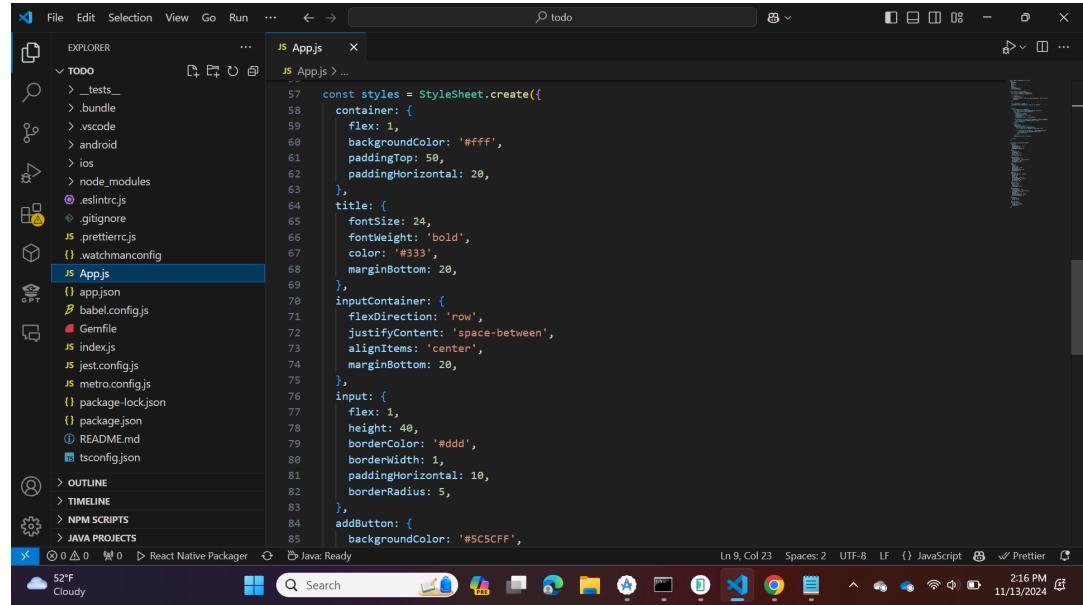
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a project structure with files like `TODO`, `_tests_`, `.bundle`, `.vscode`, `android`, `ios`, `node_modules`, `.eslintrc.js`, `.gitignore`, `.prettierrc.js`, and `.watchmanconfig`.
- Code Editor:** The main editor window displays the `App.js` file with the following code:

```
JS App.js ...
JS App.js > ...
11 export default function App() {
12   const [task, setTask] = useState('');
13   const [tasks, setTasks] = useState([]);
14
15   const addTask = () => {
16     if (task.trim()) {
17       setTasks([...tasks, {id: Date.now().toString(), text: task}]);
18       setTask('');
19     }
20   };
21
22   const deleteTask = taskId => {
23     setTasks(tasks.filter(item => item.id !== taskId));
24   };
25
26   return (
27     <View style={styles.container}>
28       <Text style={styles.title}>Simple To-Do List</Text>
29       <View style={styles.inputContainer}>
30         <TextInput
31           style={styles.input}
32           placeholder="Add a new task"
33           value={task}
34           onChange={text => setTask(text)}
35         />
36         <TouchableOpacity style={styles.addButton} onPress={addTask}>
37           <Text style={styles.addButtonText}>+</Text>
38         </TouchableOpacity>
39       </View>
40       <FlatList
41         data={tasks}
42         keyExtractor={(item) => item.id}
43         renderItem={({item}) =>
44           <Text style={styles.item}>{item.text}</Text>
45         }
46       />
47     </View>
48   );
49 }
```
- Status Bar:** At the bottom, it shows "Cloudy", "Java: Ready", "Ln 9, Col 23", "Spaces: 2", "UTF-8", "LF", "JavaScript", "Prettier", "2:13 PM", and the date "11/13/2024".

Step 3: Add Styles for the To-Do List

Add personalized styles to the bottom of App.js to improve the UI.



The screenshot shows the Visual Studio Code interface with the 'App.js' file open in the editor. The file contains code for a to-do list application. At the bottom of the code, there is a block of CSS-like styles:

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    paddingTop: 50,
    paddingHorizontal: 20,
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    color: '#333',
    marginBottom: 20,
  },
  inputContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    marginBottom: 20,
  },
  input: {
    flex: 1,
    height: 40,
    border: '1px solid #ddd',
    borderWidth: 1,
    padding: 10,
    borderRadius: 5,
  },
  addButton: {
    backgroundColor: '#5C5CFF',
    color: 'white',
    padding: 10,
    border: '1px solid #5C5CFF',
    borderRadius: 5,
    width: 50px,
  }
});
```

The 'EXPLORER' sidebar on the left shows the project structure, including files like 'app.json', 'babel.config.js', 'Gemfile', 'index.js', 'jest.config.js', 'metro.config.js', 'package-lock.json', 'package.json', 'README.md', and 'tsconfig.json'. The 'TODO' folder is also visible.

Step 4: Running the App

- Run the app with:

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

```

Administrator: C:\WINDOWS\system32\cmd.exe
> Task :app:processDebugJavaRes UP-TO-DATE
> Task :app:mergeDebugJavaResource UP-TO-DATE
> Task :app:checkDebugDuplicateClasses UP-TO-DATE
> Task :app:mergeExtDebug UP-TO-DATE
> Task :app:mergeLibDebug UP-TO-DATE
> Task :app:mergeProjectDebug UP-TO-DATE
> Task :app:configureCmakeDebug[armv7a]
> Task :app:configureCmakeDebug[armeabi-v8a]
> Task :app:configureCmakeDebug[mips64-v7a]
Task :app:buildId[MakeDebug armv7a]
> Task :app:configureCmakeDebug[x86]
> Task :app:buildId[MakeDebug x86]
> Task :app:configureCmakeDebug[x86_64]
> Task :app:buildId[MakeDebug x86_64]
> Task :app:mergeDebugJniLibFolders UP-TO-DATE
> Task :app:mergeDebugNativeLibs UP-TO-DATE
> Task :app:stripDebugDebugSymbols UP-TO-DATE
> Task :app:validateSigningDebug UP-TO-DATE
> Task :app:writeDebugAppMetadata UP-TO-DATE
> Task :app:writeDebugSigningConfigVersions UP-TO-DATE
> Task :app:packageDebug UP-TO-DATE
> Task :app:createDebugApkListingFileRedirect UP-TO-DATE

> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Medium_Phone_API_35(AVD) - 15' for :app:debug
Installed 1 device.

BUILD SUCCESSFUL in 32s
62 actionable tasks: 13 executed, 49 up-to-date
Info Connecting to the development server...
Info Starting the app on "emulator-5554"...
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.todo/.MainActivity }
DIDLE // index.js

(NOBRIDGE) LOG Bridgeless mode is enabled
(NOBRIDGE) LOG Running "todo" with {"rootTag":11,"initialProps":{},"fabric":true}
INFO
JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool
(google Chrome or Microsoft Edge).]

Android Emulator - Medium_Phone_API_35:5554
2:21 9
Simple To-Do List
Add a new task

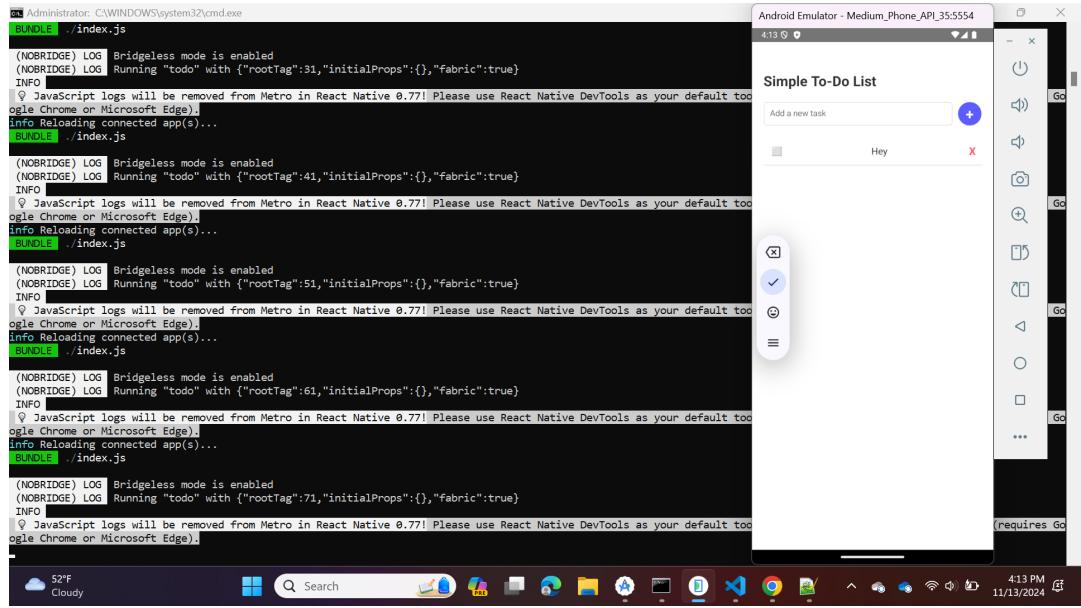
```

- This compiles and launches the app on your selected platform.

Features to Implement

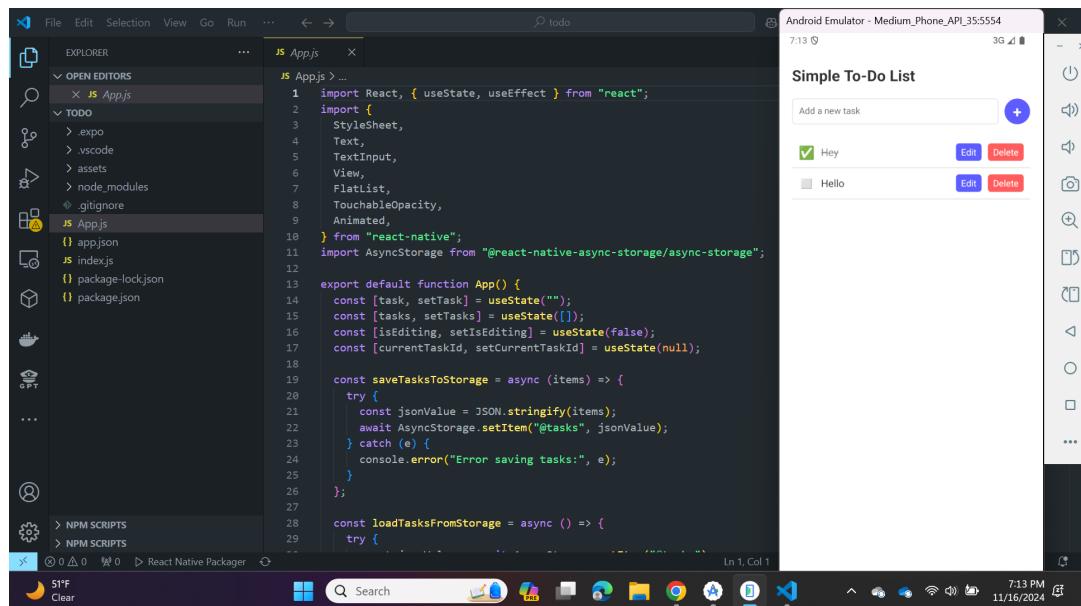
1. Mark Tasks as Complete (15 Points):

- Add a toggle function to mark tasks as completed.
- Style completed tasks with strikethrough text or color changes.



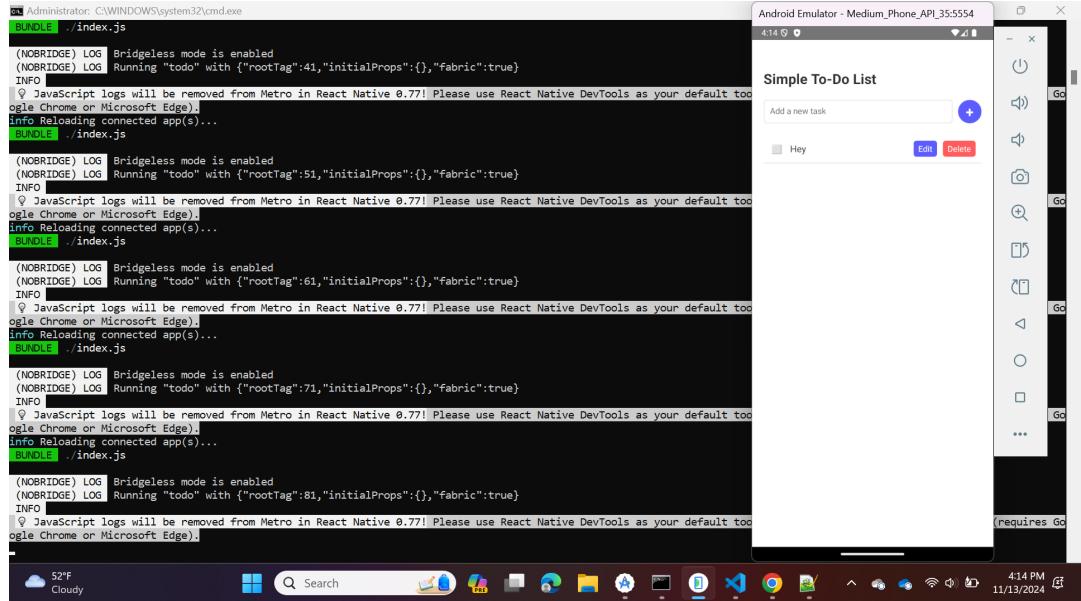
1. Persist Data Using AsyncStorage (15 Points):

- Use AsyncStorage to save and retrieve tasks, ensuring persistence after app closure.



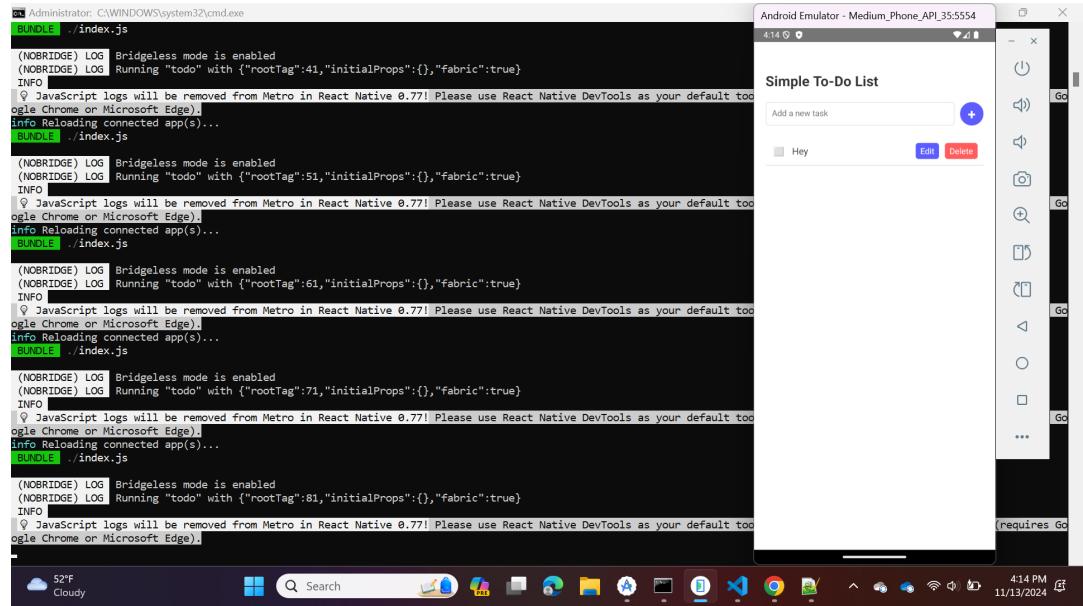
1. Edit Tasks (10 Points):

- Allow users to tap on tasks to edit their content.
- Implement a function to update tasks in the state array.



1. Add Animations (10 Points):

- Use the Animated API for visual effects when adding or deleting tasks.
- Describe how animations enhance the user experience.



- Changed UI Styling:

