

Task 1 (Total 40 Points):

(A) Screenshots of Your App (5 Points).

```
Last login: Sun Nov  3 13:42:45 on console
(base) pateljeet@MacBookPro ~ % brew install watchman
zsh: command not found: brew
(base) pateljeet@MacBookPro ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for "sudo" access (which may request your password)...
Password:
=> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/_brew
/usr/local/Homebrew
=> The following existing directories will be made group writable:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/share
/usr/local/share/doc
/usr/local/share/man
/usr/local/share/man/man1
=> The following existing directories will have their owner set to pateljeet:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/share
/usr/local/share/doc
/usr/local/share/man
/usr/local/share/man/man1
=> The following existing directories will have their group set to admin:
/usr/local/bin
/usr/local/include
/usr/local/lib
/usr/local/share
/usr/local/share/doc
/usr/local/share/man
/usr/local/share/man/man1
=> The following new directories will be created:
/usr/local/bin
/usr/local/sbin
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Caskroom
/usr/local/Frameworks

Press RETURN/ENTER to continue or any other key to abort:
=> /usr/bin/sudo /bin/chmod u=rwx /usr/local/bin/usr/local/include /usr/local/lib /usr/local/share /usr/local/doc /usr/local/share/man /usr/local/share/man/man1
=> /usr/bin/sudo /bin/chmod g=rwx /usr/local/include /usr/local/lib /usr/local/share /usr/local/doc /usr/local/share/doc /usr/local/share/man /usr/local/share/man/man1
=> /usr/bin/sudo /bin/chmod o=rwx /usr/local/include /usr/local/lib /usr/local/share /usr/local/include /usr/local/lib /usr/local/share /usr/local/doc /usr/local/share/man /usr/local/share/man/man1
=> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/bin /usr/local/include /usr/local/lib /usr/local/share /usr/local/doc /usr/local/share/man /usr/local/share/man/man1
=> /usr/bin/sudo /bin/mkdir -p /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked
/usr/local/Cellar /usr/local/Caskroom /usr/local/Frameworks
=> /usr/bin/sudo /bin/chmod u=rwx /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked
=> /usr/bin/sudo /bin/chmod g=rwx /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked
=> /usr/bin/sudo /bin/chmod o=rwx /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked
=> /usr/bin/sudo /bin/chgrp admin /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked
=> /usr/bin/sudo /bin/chgrp admin /usr/local/etc /usr/local/sbin /usr/local/var /usr/local/opt /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew /usr/local/var/homebrew/linked

/linked /usr/local/Cellar /usr/local/Caskroom /usr/local/Frameworks
=> /usr/bin/sudo /bin/mkdir -p /usr/local/Homebrew
=> /usr/bin/sudo /usr/sbin/chown -R pateljeet:admin /usr/local/Homebrew
=> /usr/bin/sudo /bin/chmod g=rwx /Users/pateljeet/Library/Caches/Homebrew
=> /usr/bin/sudo /bin/chmod g=rwx /Users/pateljeet/Library/Caches/Homebrew
=> /usr/bin/sudo /bin/chmod g=rwx /Users/pateljeet/Library/Caches/Homebrew
=> Downloading and installing Homebrew...
remote: Enumerating objects: 285222, done.
remote: Counting objects: 100% (140/140), done.
remote: Compressing objects: 100% (107/107), done.
remote: Total 285222 (delta 35), reused 126 (delta 28), pack-reused 285882 (from 1)
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 65 (delta 33), reused 33 (delta 33), pack-reused 22 (from 1)
  Updating Homebrew...
=> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:e02b387d80f10c835df1511536b0b5debBe35f8967c7e68c9942af046023209
#####
##### 100.0%
=> Pouring portable-ruby-3.3.5.e1-capitan.bottle.tar.gz
=> Installation successful!

=> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

=> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

=> Next steps:
- Run these commands in your terminal to add Homebrew to your PATH:
  echo >> /Users/pateljeet/.zprofile
  echo 'eval "$(/usr/local/bin/brew shellenv)"' >> /Users/pateljeet/.zprofile
  eval '$(/usr/local/bin/brew shellenv)'
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

(base) pateljeet@MacBookPro ~ % brew install watchman
=> Downloading https://ghcr.io/v2/homebrew/core/watchman/manifests/2024.10.28.0
#####
##### 100.0%
=> Fetching dependencies for watchman: icu@0.97, xz@1.4, zstd, boost@1.76.0, double-conversion@3.2.0, fmt@8.1.0, gflags@2.2.2, glog@0.3.7, ca-certificates@2024.09.13
=> Downloading https://ghcr.io/v2/homebrew/core/icu@0.97/manifests/76.1.1
#####
##### 100.0%
=> Fetching icu@0.97
=> Downloading https://ghcr.io/v2/homebrew/core/icu@0.97/blobs/sha256:7ea20264
#####
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/1.10.0-1
#####
##### 100.0%
=> Fetching xz@1.4
=> Downloading https://ghcr.io/v2/homebrew/core/xz/1.4/blobs/sha256:75cb294d26d3
#####
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/zstd/manifests/1.5.6
#####
##### 100.0%
=> Fetching zstd@1.5.6
=> Downloading https://ghcr.io/v2/homebrew/core/zstd/blobs/sha256:eb298fe6b57
#####
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/boost/manifests/1.86.0_2
#####
##### 100.0%
```

```

==> Downloading https://ghcr.io/v2/homebrew/core/python/3.13/blobs/sha256:44239b
#####
Fetching https://ghcr.io/v2/homebrew/core/watchman/blobs/sha256:97bece127c
#####
Installing dependency for watchman: icu@0@76, xz, lz4, zstd, boost, double-conversion, fmt, gflags, glog, ca-certificates, openssl@3, libevent, libsodium, snappy, folly, fizz, wangle, xxhash, fbthrift, fb303, edencommon, pcre2, mpfr, readline, sqlite and python@3.13
==> Installing watchman dependency: icu@0@76
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/c105f8980d3ac0ff0253b47492e0865020580406ee75fa52509cf5b251b4d633--icu@0@76-76.1.1.bottle_manifest.json
  Pouring icu@0@76-76.1.1.sonoma.bottle.tar.gz
  /usr/local/Cellar/icu@0@76/76.1.1: 277 files, 79.9MB
==> Installing watchman dependency: xz
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.6.3
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/e68799185cc17334188ba286ba8c3df8587c9fd8375647d85f7157b62a979599--xz-5.6.3.bottle_manifest.json
  Pouring xz-5.6.3.tar.gz
  /usr/local/Cellar/xz/5.6.3: 96 files, 1.7MB
==> Installing watchman dependency: lz4
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/manifests/1.10.0-1
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/8e11e90eb21a0e60f199af9d80e011e3693c77dd353b2477579d95c8471a5802--lz4-1.10.0-1.bottle_manifest.json
  Pouring lz4-1.10.0.sequoia.bottle.1.tar.gz
  /usr/local/Cellar/lz4/1.10.0: 24 files, 663.9KB
==> Downloading https://ghcr.io/v2/homebrew/core/zstd/manifests/1.5.6
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/29403e0df540d8aea0e750ac135ec9ef44fc5e6b6df69170ed602acabf0ffb--zstd-1.5.6.bottle_manifest.json
  Pouring zstd-1.5.6.sequoia.bottle.tar.gz
  /usr/local/Cellar/zstd/1.5.6: 32 files, 2.3MB
==> Installing watchman dependency: boost
==> Downloading https://ghcr.io/v2/homebrew/core/boost/manifests/1.86.0_2
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/c57d0cd93a80a3629c8344f740e1cc2b4f4bd4ba4e95ed9415993c3d0153cb6--boost-1.86.0_2.bottle_manifest.json
  Pouring boost-1.86.0_2.tar.gz
  /usr/local/Cellar/boost/1.86.0_2: 16.225 files, 521.6MB
==> Installing watchman dependency: double-conversion
==> Downloading https://ghcr.io/v2/homebrew/core/double-conversion/manifests/3.3
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/3f6c03611d69daaddcd2d467993bc4e75cc54a55e29800fd5f383411c445a5b4--double-conversion-3.3.0.bottle_manifest.json
  Pouring double-conversion-3.3.0.sonoma.bottle.tar.gz
  /usr/local/Cellar/double-conversion/3.3.0: 27 files, 246.4KB
==> Installing watchman dependency: glog
==> Downloading https://ghcr.io/v2/homebrew/core/glog/manifests/0.6.0
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/cf089458479cf6cac0b693117f82c8cdbdf37b82474afbf5fc6e22a48193739f--fmt-11.0.2.bottle_manifest.json
  Pouring fmt-11.0.2.sonoma.bottle.tar.gz
  /usr/local/Cellar/fmt/11.0.2: 29 files, 1MB
==> Installing watchman dependency: gflags
==> Downloading https://ghcr.io/v2/homebrew/core/gflags/manifests/2.2.2-2
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/9683aedd51c2c61698e3e3e99a2ac77c776893acd679bcd1690ffd98fe8a2e2--gflags-2.2.2-2.bottle_manifest.json
  Pouring gflags-2.2.2-2.sonoma.bottle.2.tar.gz
  /usr/local/Cellar/gflags/2.2.2: 27 files, 721.2KB
==> Installing watchman dependency: glog
==> Downloading https://ghcr.io/v2/homebrew/core/glog/manifests/0.6.0
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/ce2b2dbc995f80003a0f9198fb41eb0d881df95aaaff8b5bb14ffdc7867599e--glog-0.6.0.bottle_manifest.json
  Pouring glog-0.6.0.sonoma.bottle.tar.gz
  /usr/local/Cellar/glog/0.6.0: 1 file, 386.4KB
  Installing watchman dependency: ca-certificates
  Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2024-
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/338dad07c2ff7c822cdca7c417944521589856741c0fb7d7a7f07b8a18d7fb7e05--ca-certificates-2024-09-24.bottle_manifest.json
  Pouring ca-certificates-2024-09-24.all.bottle.tar.gz
  Regenerating CA certificate bundle from keychain, this may take a while...
  /usr/local/Cellar/ca-certificates/2024-09-24: 4 files, 237.4KB
  Downloading https://ghcr.io/v2/homebrew/core/openssl@3/manifests/3.4.0
Already downloaded: /Users/pateljeet/Library/Caches/Homebrew/downloads/a8a169d38329b14f967d497d0ea77eee6dc5444175093c7b26026b357e8173e--openssl@3-3.4.0.bottle_manifest.json
  Pouring openssl@3-3.4.0.sonoma.bottle.tar.gz
  /usr/local/Cellar/openssl@3/3.4.0: 7,236 files, 33.8MB

  /usr/local/Cellar/python@3.13/3.13.0_1: 3,253 files, 60.9MB
==> Installing watchman
  Pouring watchman--2024.10.28.00.sonoma.bottle.tar.gz
  /usr/local/Cellar/watchman/2024.10.28.00: 26 files, 13.5MB
  Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
  Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
  (base) pateljeet@MacBookPro ~ % watchman --version
2024.10.28.00
  (base) pateljeet@MacBookPro ~ % npm install -g react-native-cli
npm warn deprecated inflight@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 Glob@2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated rimraf@2.7: Rimraf versions prior to v4 are no longer supported
added 89 packages in 4s

41 packages are looking for funding
  run 'npm fund' for details
  (base) pateljeet@MacBookPro ~ % 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
  (base) pateljeet@MacBookPro ~ % npx react-native init SimpleTodoApp
  This will walk you through creating a new React Native project in /Users/pateljeet/SimpleTodoApp
  Installing react-native...
  Consider installing yarn to make this faster: https://yarnpkg.com
  npm warn deprecated inflight@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@0.2: Rimraf versions prior to v4 are no longer supported
  npm warn deprecated glob@2.3: Glob versions prior to v9 are no longer supported
  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-transform-class-properties instead.
  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/plugin-in-operator instead.
  npm warn deprecated @babel/plugin-proposal-optimal-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-transform-optimal-chaining instead.
  npm warn deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported

  added 502 packages, and audited 503 packages in 29s

  30 packages are looking for funding
    run 'npm fund' for details

  found 0 vulnerabilities
  /usr/local/lib/node_modules/react-native-cli/index.js:302
    cli.init(root, projectName);
    ^

  TypeError: cli.init is not a function
    at run (/usr/local/lib/node_modules/react-native-cli/index.js:302:7)
    at createProject (/usr/local/lib/node_modules/react-native-cli/index.js:249:3)
    at init (/usr/local/lib/node_modules/react-native-cli/index.js:200:5)
    at Object. (/usr/local/lib/node_modules/react-native-cli/index.js:153:7)
    at Module._compile (node:internal/modules/js/loader:1546:14)
    at Object.Module._extensions..js (node:internal/modules/helpers:102:32)
    at Module.load (node:internal/modules/loader:80:32)
    at Function._load (node:internal/modules/cjs/loader:1128:12)
    at TracingChannel.traceSync (node:diagnostics_channel:315:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:218:24)

  Node.js v22.11.0
  (base) pateljeet@MacBookPro SimpleTodoApp % npx react-native start

```

```

-transform=async-generator-functions 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-trans
over object-rest-spread instead
npm warn deprecated glob@7.1.6: 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
npm warn deprecated glob@7.1.6: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.1.6: this version is no longer supported, please update to at least 0.8.*
added 451 packages, and audited 1076 packages in 11s
135 packages are looking for funding
  run 'npm fund' for details

3 moderate severity vulnerabilities

To address all issues, run:
  npm audit fix

Run 'npm audit' for details.
(base) pateljeet@MacBookPro SimpleTodoApp % npx expo start
Starting project at /Users/pateljeet/SimpleTodoApp
(node:7378) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
Starting Metro Bundler...
The following packages should be updated for best compatibility with the installed expo version:
  react-native@0.76.1 - expected version: 0.74.5
Your project may not work correctly until you install the expected versions of the packages.


> Metro waiting on exp://192.168.1.44:8081
> 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
> Press o | open project code in your editor

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
> Reloading apps
warn No apps connected. Sending "reload" to all React Native apps failed. Make sure your app is running in the simulator or on a phone connected via USB.

```

localhost:8081

To address all issues (including breaking changes), run:
 npm audit fix --force
 Run 'npm audit' for details.
 (base) pateljeet@Jeeets-MacBook-Pro SimpleApp % npm run web
> simpleapp@0.0.0 web
> expo start --web
Starting project at /Users/pateljeet/SimpleApp
(node:4278) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
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.



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

```

> Metro waiting on exp://172.20.10.10:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
> Web is waiting on http://localhost:8081

> 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
> Press o | open project code in your editor

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
Web Bundled 4425ms node_modules/expo/AppEntry.js (214 modules)

```

The screenshot shows the Expo developer tools interface. On the left, the project structure is visible with files like App.js, package.json, and README.md. The main area displays the code for App.js:

```

1 import React, { useState } from 'react';
2 import { StyleSheet, Text, View, Button } from 'react-native';
3
4 export default function App() {
5   const [showText, setShowText] = useState(true);
6
7   const toggleText = () => {
8     setShowText(!showText);
9   };
10
11   return (
12     <View style={styles.container}>
13       [<showText && <Text style={styles.text}>Hello, This my first native application !</Text>]
14         <Button title="Toggle Text" onPress={toggleText} />
15       </View>
16     );
17   }
18
19 const styles = StyleSheet.create({
20   container: {
21     flex: 1,
22     justifyContent: 'center',
23     alignItems: 'center',
24     backgroundColor: '#ffff',
25   },
26   text: {
27     fontSize: 20,
28     marginBottom: 20,
29   },
30 });
31

```

The right side shows a preview of the app running on an iPhone 15 Pro. The screen displays the text "Hello, This my first native application !" and a button labeled "Toggle Text". Below the preview are controls for font scaling and device selection.

B) Describe any differences you observed between running the app on an emulator versus a physical device.

When running the app on an emulator and a physical device, you may notice the following differences:

- Performance:** The physical device often runs the app more smoothly, with faster load times and fewer lags, especially with animations. Emulators may be slower, particularly on machines with limited resources.
- User Interaction:** The physical device allows real touch gestures, like swipes and pinches, whereas the emulator mimics touch with mouse clicks, which isn't as precise.
- Hardware-Specific Features:** Certain features, like camera or sensors, may not work as expected on an emulator but function correctly on a physical device.
- Screen Resolution and Display:** Physical devices show the actual display quality, including color and brightness. Emulators approximate this, but physical devices may reveal UI issues related to screen size and aspect ratio more accurately.

(2) Setting Up an Emulator (10 Points):

A) Explain the steps you followed to set up an emulator in Android Studio or Xcode.

Step 1: Install Xcode

1. **Download Xcode** from the [Mac App Store](#).
2. **Open Xcode** once installed and follow any initial setup instructions.

Step 2: Set Command Line Tools (recommended for SDK tools)

1. In Xcode, go to **Xcode > Preferences** in the top menu.
2. Click on the **Locations** tab.
3. Under **Command Line Tools**, select the version of Xcode you installed (e.g., Xcode 13 or Xcode 14). This will enable SDK and build tools for development.

Step 3: Open the Simulator

1. You can open the **Simulator** directly from Xcode:
 - o Go to **Xcode > Open Developer Tool > Simulator** from the top menu.

Step 4: Set Up a New iOS Simulated Device

1. In Xcode, go to **Window > Devices and Simulators** in the top menu.
2. Click the **Simulators** tab to view existing devices or create a new one.
3. Press the + button in the bottom left to add a new simulator.
4. Click **Create** to add the simulator to your list.

Step 5: Start the Simulator

1. In the **Devices and Simulators** window, select your newly created simulator.
2. Click **Start** to launch it or open it by selecting the device from Xcode's toolbar when running a project.

3. The Simulator will open a window emulating the iOS device, allowing you to test applications as if they were running on an actual iPhone or iPad.

B) Discuss any challenges you faced during the setup and how you overcame them.

- **SDK Version Compatibility:** Sometimes, the latest SDK versions may not be compatible with certain versions of React Native. To solve this, select a stable or recommended SDK version.
- **Slow Performance:** Emulators can be slow, especially on machines with limited memory. Increasing RAM allocation for the emulator in AVD settings can help.
- **Device-Specific Issues:** Some devices or system images may have unique bugs or crashes. Switching to another device model or Android version often resolves these issues.

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

A) Describe how you connected your physical device to run the app using Expo:

- As I have uploaded the Screen shot of how I did install the Expo in my Mac Terminal Using command **npm install -g expo-cli** This opens Expo Developer Tools in your browser, displaying a QR code.
- Then after I have installed **Expo Go App** from the Apple Store, Open Expo Go and Scan the QR code which is given from terminal tool.
- The app should load on your device. Any code changes will automatically reload in Expo Go.

B) Include any troubleshooting steps if you encountered issues.

- **Network Issues:** Ensure both your computer and mobile device are connected to the same Wi-Fi network. Expo requires both to be on the same network to communicate.
- **App Not Loading or Crashing:** Restarting the Expo server and clearing the cache (expo start -c) can resolve this.

2:56

◀ Code Scanner



Hello, This my first native application !

[Toggle Text](#)

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

Aspect	Emulator	Physical Device
Performance	Slower, especially on lower-end machines.	Faster, showing real-world app performance.
Touch Accuracy	Uses mouse clicks to simulate touch.	Actual touch gestures, more natural interactions.
Hardware Access	Limited access (e.g., camera, GPS).	Full access to all hardware features.
Screen Quality	Approximate colors and resolution.	True screen quality and color display
Testing Scope	Easy to test on multiple simulated devices.	Limited to the specific physical device.

Advantages vs. Disadvantages:

Advantages of Emulator:

- Quick to test on various Android/iOS versions and screen sizes without needing multiple devices.
- Simulated environment allows rapid testing without using a physical device.

Advantages of Physical Device:

- Provides an accurate representation of app behavior in real-world usage.
- Allows for testing of hardware-dependent features and shows true display quality.

Disadvantages of Emulator

- **Slower Performance:** Emulators run slower, especially on low-end machines.
- **Limited Hardware Access:** Can't fully simulate features like camera, GPS, or sensors.
- **Inaccurate Display:** Colors and resolution may differ from physical devices.

Disadvantages of Physical Device

- **Single Device Testing:** Limited to one device's screen size and OS version.
- **Network Dependency:** Requires the device and computer to be on the same Wi-Fi network.

(5) Troubleshooting a Common Error (5 Points).

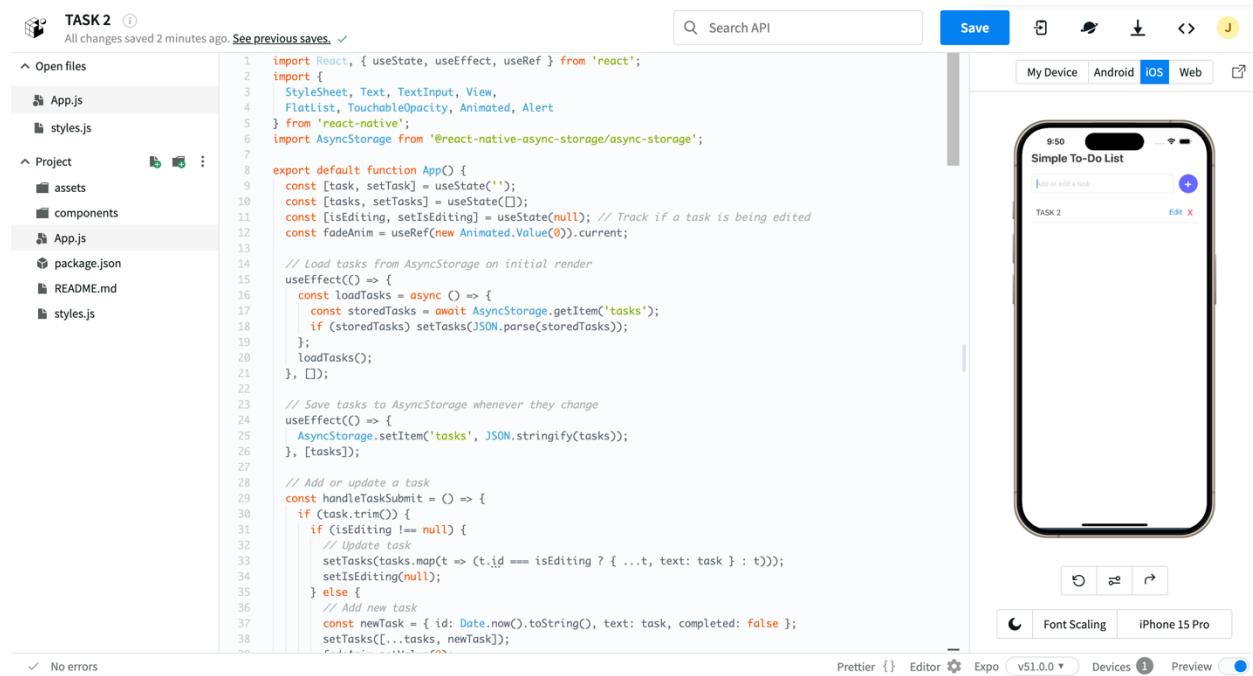
A) Identify a common error you encountered when starting your React Native app. Note that it is very unlikely that everyone will get the same error here.

I am facing common error while working with React Native is error “**Unable to load script**” from terminal often occurs when the Metro Bundler, which serves the JavaScript code to the app, fails to connect with the Android emulator or physical device. This is usually because I guess network issue. Kind of same error while **connecting the Development server**. This happens when the app can’t connect to the Metro Bundler, often due to network issues or incorrect configuration.

To Resolve this error:

- Ensure that both the device/emulator and the development computer are on the same network.
- The device/emulator and the development server must be on the same network, and any disconnection can cause this error.
- Cache or other configuration problems can prevent the bundler from properly serving the script.

Task 2: Building a Simple To-Do List App (60 Points).



The screenshot shows the Expo developer tools interface. On the left, the file structure is visible with files like App.js, styles.js, assets, components, and package.json. The main area contains the code for App.js:

```
1 import React, { useState, useEffect, useRef } from 'react';
2 import {
3   StyleSheet, Text, TextInput, View,
4   FlatList, TouchableOpacity, Animated, Alert
5 } from 'react-native';
6 import AsyncStorage from '@react-native-async-storage/async-storage';
7
8 export default function App() {
9   const [task, setTask] = useState('');
10  const [tasks, setTasks] = useState([]);
11  const [isEditing, setIsEditing] = useState(null); // Track if a task is being edited
12  const fadeAnim = useRef(new Animated.Value(0)).current;
13
14  // Load tasks from AsyncStorage on initial render
15  useEffect(() => {
16    const loadTasks = async () => {
17      const storedTasks = await AsyncStorage.getItem('tasks');
18      if (storedTasks) setTasks(JSON.parse(storedTasks));
19    };
20    loadTasks();
21  }, []);
22
23  // Save tasks to AsyncStorage whenever they change
24  useEffect(() => {
25    AsyncStorage.setItem('tasks', JSON.stringify(tasks));
26  }, [tasks]);
27
28  // Add or update a task
29  const handleTaskSubmit = () => {
30    if (task.trim()) {
31      if (isEditing !== null) {
32        // Update task
33        setTasks(tasks.map(t => (t.id === isEditing ? { ...t, text: task } : t)));
34        setIsEditing(null);
35      } else {
36        // Add new task
37        const newTask = { id: Date.now().toString(), text: task, completed: false };
38        setTasks([...tasks, newTask]);
39      }
40    }
41  };
42
43  // Delete a task
44  const handleTaskDelete = (id) => {
45    setTasks(tasks.filter(t => t.id !== id));
46  };
47
48  // Render tasks
49  const TaskItem = ({ task, isEditing, handleTaskDelete }) => {
50    const [text, onChangeText] = useState(task);
51    const [isFocused, onFocus] = useState(false);
52    const [isSubmitting, setSubmitting] = useState(false);
53
54    const handleTextChange = (text) => {
55      if (!isSubmitting) {
56        setSubmitting(true);
57        setTimeout(() => {
58          setSubmitting(false);
59          if (isEditing) {
60            setTasks([
61              ...tasks.slice(0, isEditing),
62              { ...task, text },
63              ...tasks.slice(isEditing + 1)
64            ]);
65          } else {
66            setTasks([
67              ...tasks.slice(0, -1),
68              { ...task, completed: true },
69              ...tasks.slice(-1)
70            ]);
71          }
72        }, 500);
73      }
74    };
75
76    return (
77      <View style={styles.taskContainer}>
78        <Text>{task}</Text>
79        <Text>Edit</Text>
80        <Text>Delete</Text>
81      </View>
82    );
83  };
84
85  // Render the list of tasks
86  const TaskList = ({ tasks }) => {
87    return (
88      <FlatList
89        data={tasks}
90        keyExtractor={(task) => task.id}
91        renderItem={({ task }) => (
92          <TaskItem task={task} isEditing={isEditing} handleTaskDelete={handleTaskDelete} />
93        )}
94      />
95    );
96  };
97
98  // Render the form
99  const Form = ({ placeholder, value, onChangeText, isEditing, handleTextChange }) => {
100    return (
101      <View style={styles.formContainer}>
102        <Text>{placeholder}</Text>
103        <TextInput
104          value={value}
105          onChangeText={onChangeText}
106          style={styles.input}
107          ref={fadeAnim}
108          blurOnSubmit
109          blurOnExit
110          onFocus={onFocus}
111          onBlur={onFocus}
112        />
113        <Text>{isEditing ? 'Edit' : 'Add'}</Text>
114      </View>
115    );
116  };
117
118  // Render the entire app
119  return (
120    <View style={styles.container}>
121      <Form placeholder="Add a task" value={task} onChangeText={handleTextChange} isEditing={false} />
122      <TaskList tasks={tasks} />
123    </View>
124  );
125}
```

The right side of the interface shows a preview of the iPhone 15 Pro displaying a simple To-Do List application with one task: "TASK 2".

Explanation of Features Implemented:

1. Add New Tasks:

- The handleTaskSubmit function handles adding new tasks or updating existing tasks. Tasks are only added if the input is not empty.

2. Update Existing Tasks:

- Users can edit a task by pressing the "Edit" button. This populates the input field with the selected task, and the "+" button changes to a "✓" to indicate an update.

3. Delete Tasks:

- Tasks can be deleted with a confirmation alert using the "X" button.

4. Scrollable Task List:

- The FlatList component is used to render the task list, allowing it to be scrollable.

5. User-Friendly Interface:

- The interface is intuitive, with clear buttons for adding, editing, and deleting tasks. It also includes animations for task additions and deletions, enhancing the user experience.

Explanation of Code:

State Management:

- useState hooks manage the input field (task) and the list of tasks (tasks).
- useEffect is used to load tasks from AsyncStorage when the app starts and save tasks whenever they are modified.

Adding a Task:

- addTask function adds a new task with a unique ID and resets the input field to an empty string.

Deleting a Task:

- deleteTask function filters out the specified task and updates the state, removing it from the list.

Toggling Task Completion:

- `toggleTaskCompletion` function toggles the completed state of a task and applies a strikethrough style for completed tasks.

Rendering the List:

- `FlatList` is used for efficient rendering of the task list. Each task displays its text and a delete button, and tasks can be marked as completed by tapping on them.

The screenshot shows the React Native code editor interface. On the left, the project structure is visible with files like App.js, styles.js, assets, components, package.json, README.md, and .gitignore. The main area displays the contents of App.js. The code uses useState, useEffect, useRef, and AsyncStorage to manage tasks. It loads tasks from AsyncStorage on initial render, saves them back whenever they change, and handles adding or updating a task. The right side shows a preview of the iOS application on an iPhone 15 Pro. The app has a simple design with a header 'Simple To-Do List' and a list item 'FIRST TASK' with a checkmark. The bottom of the screen includes navigation icons for back, forward, and search.

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

export default function App() {
  const [task, setTask] = useState('');
  const [tasks, setTasks] = useState([]);
  const [isEditing, setIsEditing] = useState(null); // Track if a task is being edited
  const fadeAnim = useRef(new Animated.Value(0)).current;

  // Load tasks from AsyncStorage on initial render
  useEffect(() => {
    const loadTasks = async () => {
      const storedTasks = await AsyncStorage.getItem('tasks');
      if (storedTasks) setTasks(JSON.parse(storedTasks));
    };
    loadTasks();
  }, []);

  // Save tasks to AsyncStorage whenever they change
  useEffect(() => {
    AsyncStorage.setItem('tasks', JSON.stringify(tasks));
  }, [tasks]);

  // Add or update a task
  const handleTaskSubmit = () => {
    if (task.trim()) {
      if (isEditing !== null) {
        // Update task
        setTasks(tasks.map(t => (t.id === isEditing ? { ...t, text: task } : t)));
        setIsEditing(null);
      } else {
        // Add new task
        const newTask = { id: Date.now().toString(), text: task, completed: false };
        setTasks([...tasks, newTask]);
      }
    }
  };
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 10,
  },
  input: {
    height: 40,
    width: '100%',
    margin: 10,
  },
  list: {
    width: '100%',
  },
  item: {
    padding: 10,
    border-bottom: 1px solid #ccc,
  },
  itemText: {
    margin: 0,
  },
  itemDelete: {
    color: 'red',
    margin-left: 10,
  },
});
```

TASK 2 ⓘ
All changes saved 11 minutes ago. [See previous saves](#) ✓

Open files

- App.js
- styles.js

Project

- assets
- components
- App.js
- package.json
- README.md
- styles.js

```

1 import React, { useState, useEffect, useRef } from 'react';
2 import {
3   StyleSheet, Text, TextInput, View,
4   FlatList, TouchableOpacity, Animated, Alert
5 } from 'react-native';
6 import AsyncStorage from '@react-native-async-storage/async-storage';
7
8 export default function App() {
9   const [task, setTask] = useState('');
10  const [tasks, setTasks] = useState([]);
11  const [isEditing, setIsEditing] = useState(null); // Track if a task is being edited
12  const fadeAnim = useRef(new Animated.Value(0)).current;
13
14  // Load tasks from AsyncStorage on initial render
15  useEffect(() => {
16    const loadTasks = async () => {
17      const storedTasks = await AsyncStorage.getItem('tasks');
18      if (storedTasks) setTasks(JSON.parse(storedTasks));
19    };
20    loadTasks();
21  }, []);
22
23  // Save tasks to AsyncStorage whenever they change
24  useEffect(() => {
25    AsyncStorage.setItem('tasks', JSON.stringify(tasks));
26  }, [tasks]);
27
28  // Add or update a task
29  const handleTaskSubmit = () => {
30    if (task.trim()) {
31      if (isEditing !== null) {
32        // Update task
33        setTasks(tasks.map(t => (t.id === isEditing ? { ...t, text: task } : t)));
34        setIsEditing(null);
35      } else {
36        // Add new task
37        const newTask = { id: Date.now().toString(), text: task, completed: false };
38        setTasks([...tasks, newTask]);
39      }
40    }
41  }

```

Save Search API

My Device Android iOS Web

Font Scaling iPhone 15 Pro

No errors

GitHub Link to my Repo: https://github.com/mrpateljeet/LAB3_ASS_WEB