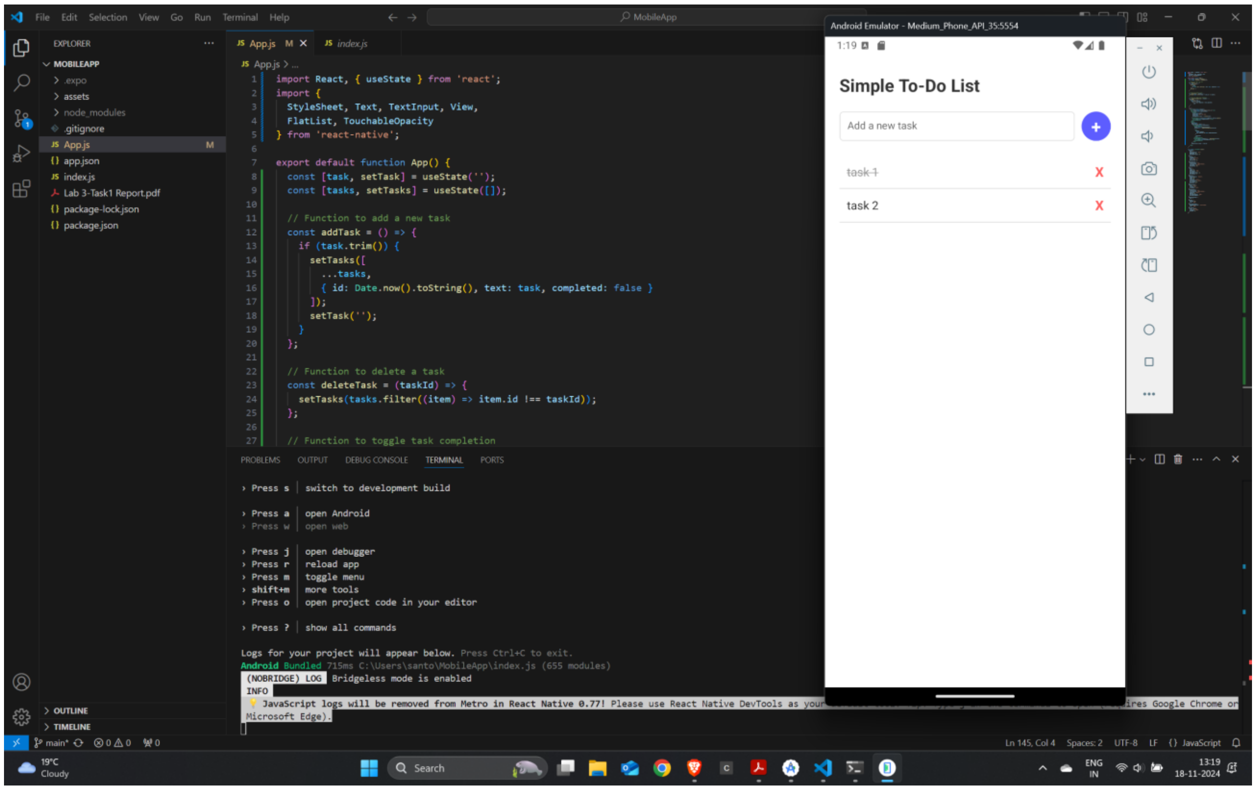
Lab 3

Sivalakshmi Chaitanya Koppuravuri Venkata 001286953

Task 2 (60 points):

(a) Mark Tasks as Complete (15 Points):



```javascript
import React, { useState } from 'react';
import {
  StyleSheet, Text, TextInput, View,
  FlatList, TouchableOpacity
} from 'react-native';

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

  // Function to add a new task
  const addTask = () => {
    if (task.trim()) {
      setTasks([
        ...tasks,
        { id: Date.now().toString(), text: task, completed: false }
      ]);
      setTask('');
    }
  };

  // Function to delete a task
  const deleteTask = (taskId) => {
    setTasks(tasks.filter((item) => item.id !== taskId));
  };

  // Function to toggle task completion
```

```
› 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
› 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.
Android Bundled 715ms C:\Users\santo\MobileApp\index.js (655 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 ires Google Chrome or
Microsoft Edge).
```

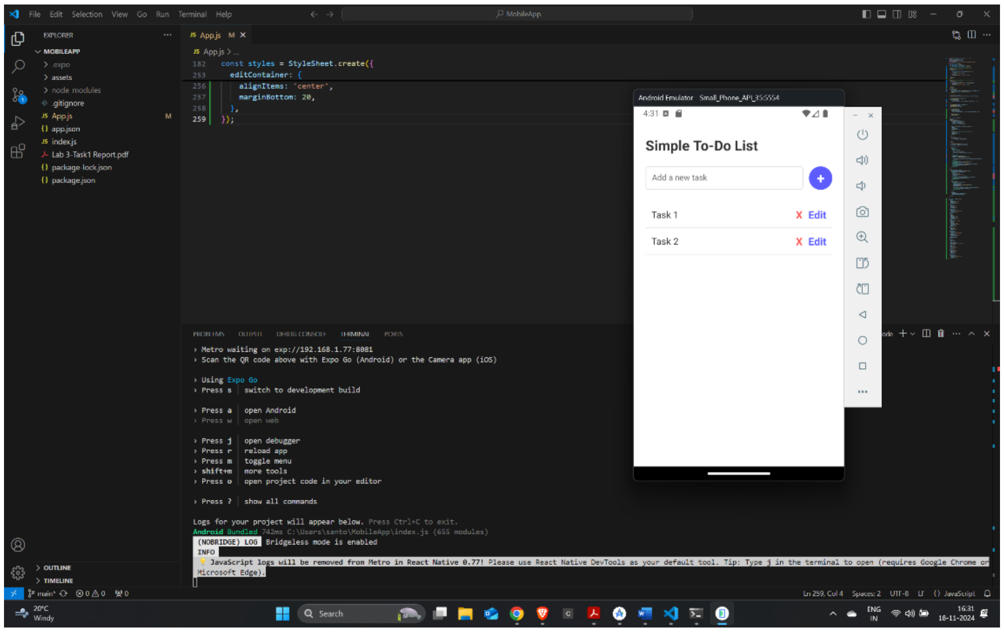
When adding a new task, we now initialize it with a completed property set to false.

The toggleTaskCompletion function uses map to iterate over the tasks. If a task's id matches the taskId passed to the function, it toggles the completed status.

Added a function to let a task is completed by clicking on the task. We used a function to strike it so that we can know that it is completed.

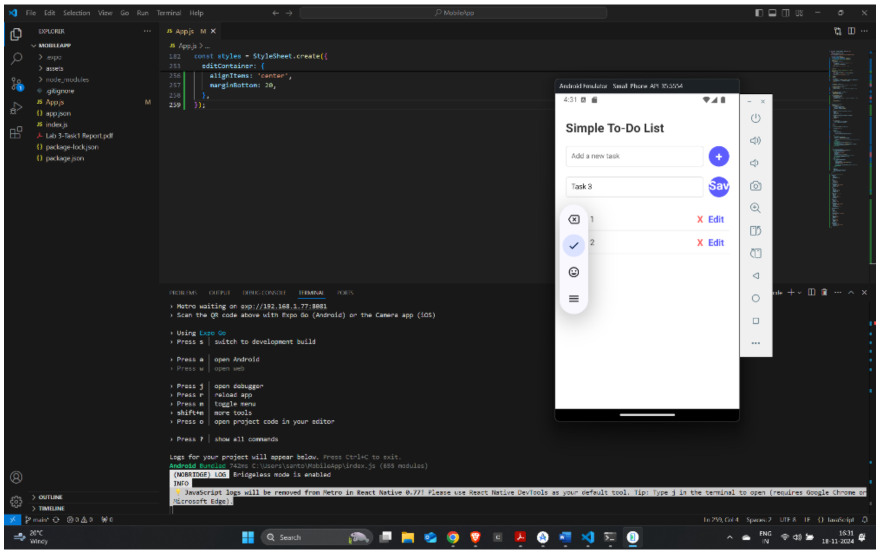
(b) This task is done with (c)

(c) **Edit Tasks (10 Points):**



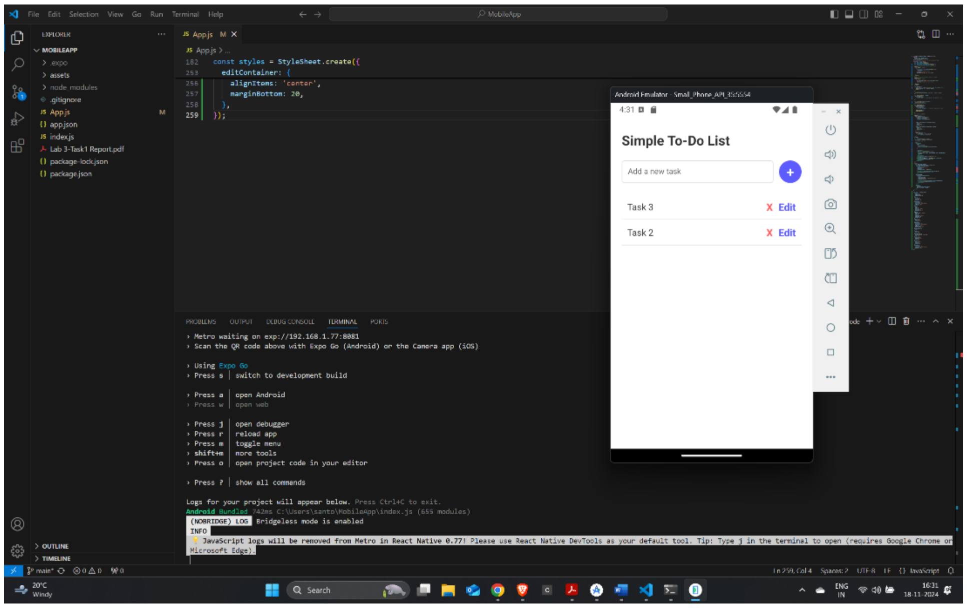
```
const styles = StyleSheet.create({
  editContainer: {
    alignItems: 'center',
    marginBottom: 20,
  },
});
```

```
› Metro waiting on exp://192.168.1.77:8081
› 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 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.
Android Bundled 742ms C:\Users\santo\MobileApp\index.js (655 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).
```



```
const styles = StyleSheet.create({
  editContainer: {
    alignItems: 'center',
    marginBottom: 20,
  },
});
```

```
› Metro waiting on exp://192.168.1.77:8081
› 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 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.
Android Bundled 742ms C:\Users\santo\MobileApp\index.js (655 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).
```



```
const styles = StyleSheet.create({
  editContainer: {
    alignItems: 'center',
    marginBottom: 20,
  },
});
```

```
› Metro waiting on exp://192.168.1.77:8081
› 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 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.
Android Bundled 742ms C:\Users\santo\MobileApp\index.js (655 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).
```

Resulting User Interface Flow:

- **Task List View**: The user sees a list of tasks with an "Edit" button next to each task.
- **Editing Mode**: When the user clicks "Edit", an input field appears with the current task's text, allowing the user to modify it. The "Save" button also appears to save the changes.
- **Save Changes**: After editing, the user clicks "Save", and the task in the list is updated with the new text. The UI then returns to the normal task list view without the editing fields.

(d) **Add Animations (10 Points):**

Animation Details:

1. **Adding a Task**:
   - **opacity**: The task gradually fades in using Animated.timing.
   - **translateY**: The task slides up from below, starting from 50 and ending at 0.
   - Both animations have a duration of 500ms to provide a smooth effect.
2. **Deleting a Task**:
   - While the deletion animation is not fully implemented in the above code, you can extend the deleteTask function to animate the task fading and sliding out before removing it. This could be done by applying an opacity reduction and a sliding motion down, and then removing the task from the state after the animation completes.