

Student: Krishna Rao

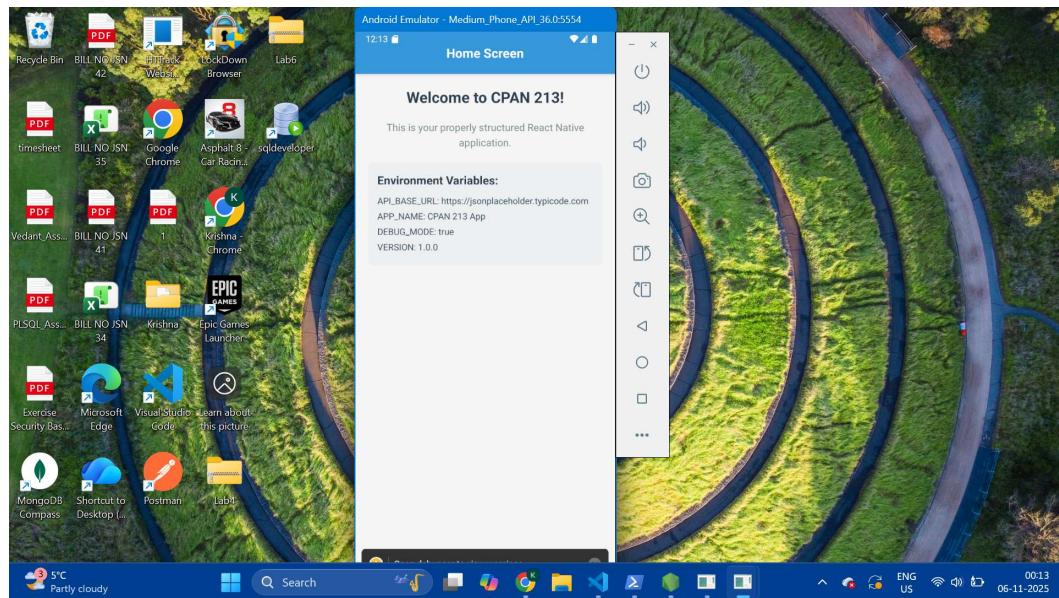
Date: November 6, 2025

1. Overview

This lab focused on setting up a complete React Native environment, including Android Studio, Node.js, ESLint, and Prettier configuration. I successfully created and ran a sample project named "Lab2." The app displayed environment variables on the home screen, confirming that the setup was working correctly.

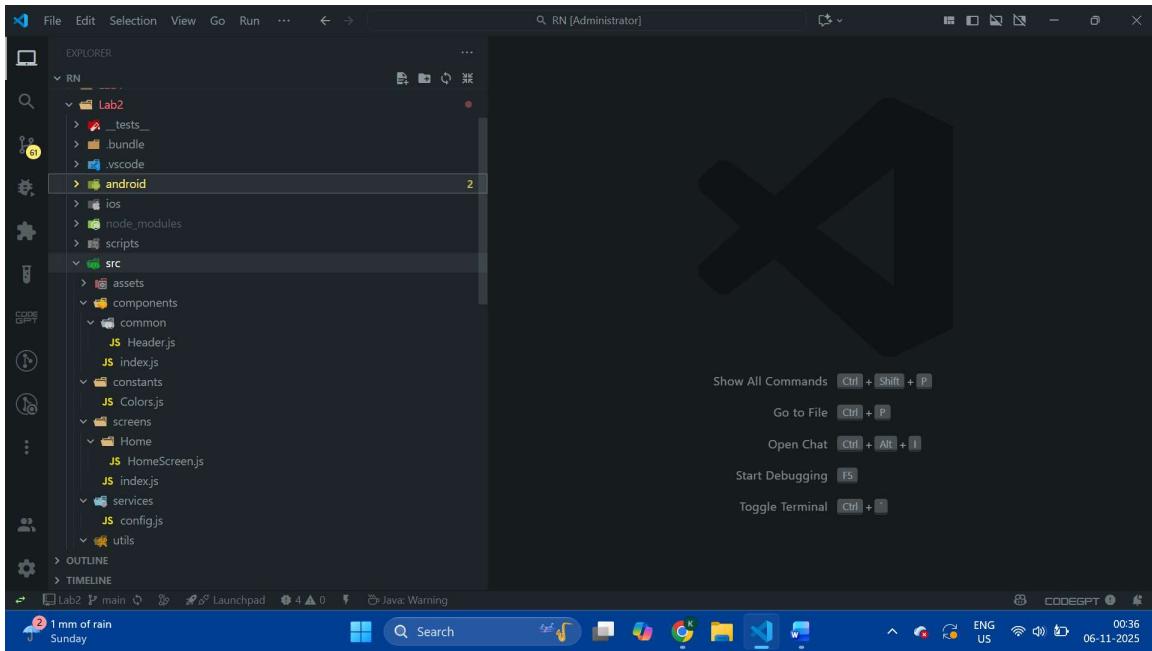
2. Screenshot of Working Application

Below is a screenshot showing the app running successfully on the Android Emulator:



3. Key Configurations

- Project Folder Structure: Organized under `src/` with separate folders for components, screens, and services.
- ESLint & Prettier: Configured to maintain consistent code quality and formatting.
- Environment Variables: Stored inside `*.env` and accessed using `react-native-config`.
- VS Code Settings: Enabled auto-format and debugging integration for Android emulator testing.



4. Issues Faced and Solutions

Issue	Description	Solution
Parsing error: Config returned typeof string	ESLint failed to read the configuration file correctly because `eslintrc.js` was returning a string instead of an object.	Updated `eslintrc.js` to properly export a JavaScript object using `module.exports = { ... }`, added correct plugins and rules, and re-ran `npm run lint`.
Parsing error: No Babel config file detected	The Babel configuration was missing or incorrectly formatted, causing the linter to fail.	Fixed by ensuring a valid `babel.config.js` file existed with the proper preset: `module:metro-react-native-babel-preset`.

5. Learnings and Takeaways

Through this lab, I learned how to properly configure and structure a React Native project for development. I understood how to resolve ESLint and Babel parsing errors, confirm device boot status before installation, and test environment variables successfully. This setup ensures smooth future development for upcoming labs.