**Student Name:**

**Student No:**

**Unit Code:**

**Unit:**

**Lecturer:**

# Introduction

A hybrid application development has its pros, with one of its major advantage being the ability to write code that targets multiple platforms with a single codebase. Different users have diverse preferences and taste in how they like to consume applications with some going for the android experience, others for the IOS, not forgetting the browser experience. Therefore, it might be fruitful for organizations to develop applications that target multiple platforms as this enhances efficiency, and can ultimately increase revenue depending on the application’s purpose. The following report aims to elaborate the process taken in development of an hybrid applications using the Cordova platform. The report will describe how to run and test the application, and elaborate on the challenges faced during the development process with the steps taken to try and solve them.

# Testing

The application has a client side and a server side. The client side is running on port 8000 while the server on port 5000. One needs to run the cordova run command to target different platforms.

## Browser

To test the application in browser, run the *cordova run browser* command. This runs the program in port 8000. One can manually navigate to the root html file by following the following link [local host link](http://localhost:8000/index.html). Additionally, one is required to start the server which can be done by navigating to the server folder in the terminal and executing the *npm run dev* command.

## Android

The android platform has been added onto the application. One needs to run the *cordova build android* application to first build the project and generate the APK file. The paths to the gradle bin, and android software development kit should be added appropriately in the environment variables to build the project successfully. The project’s target sdk and compile versions are 31. Additionally, the project uses gradle 7.1 distribution to build and test. To run the application after successful build, one needs to run *cordova run android* command. The application should be installed and launched on the emulator or mobile phone used to debug. Furthermore, one needs to launch the server by navigating to the server folder in the terminal and running either *npm run start* or *npm run dev*. The *npm run dev* starts the server and connects to the mongoDB database. The difference between the two is that npm run dev watches the server for changes which is helpful during development.

# Features Done

## 1. QR code Object Scanning

This feature entails capturing of data object representing a book and storing to the local storage as shown below.

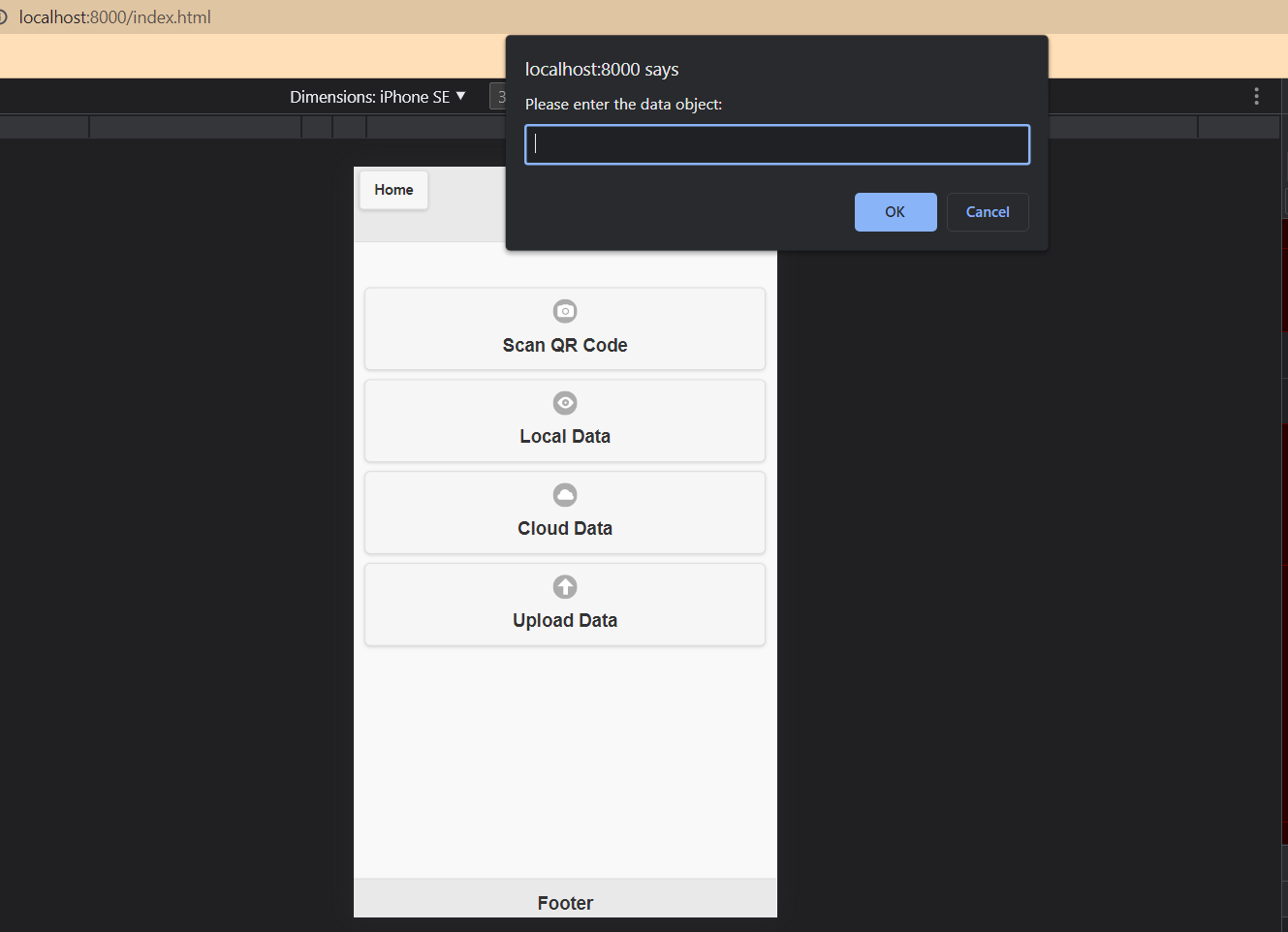


Figure Book object input

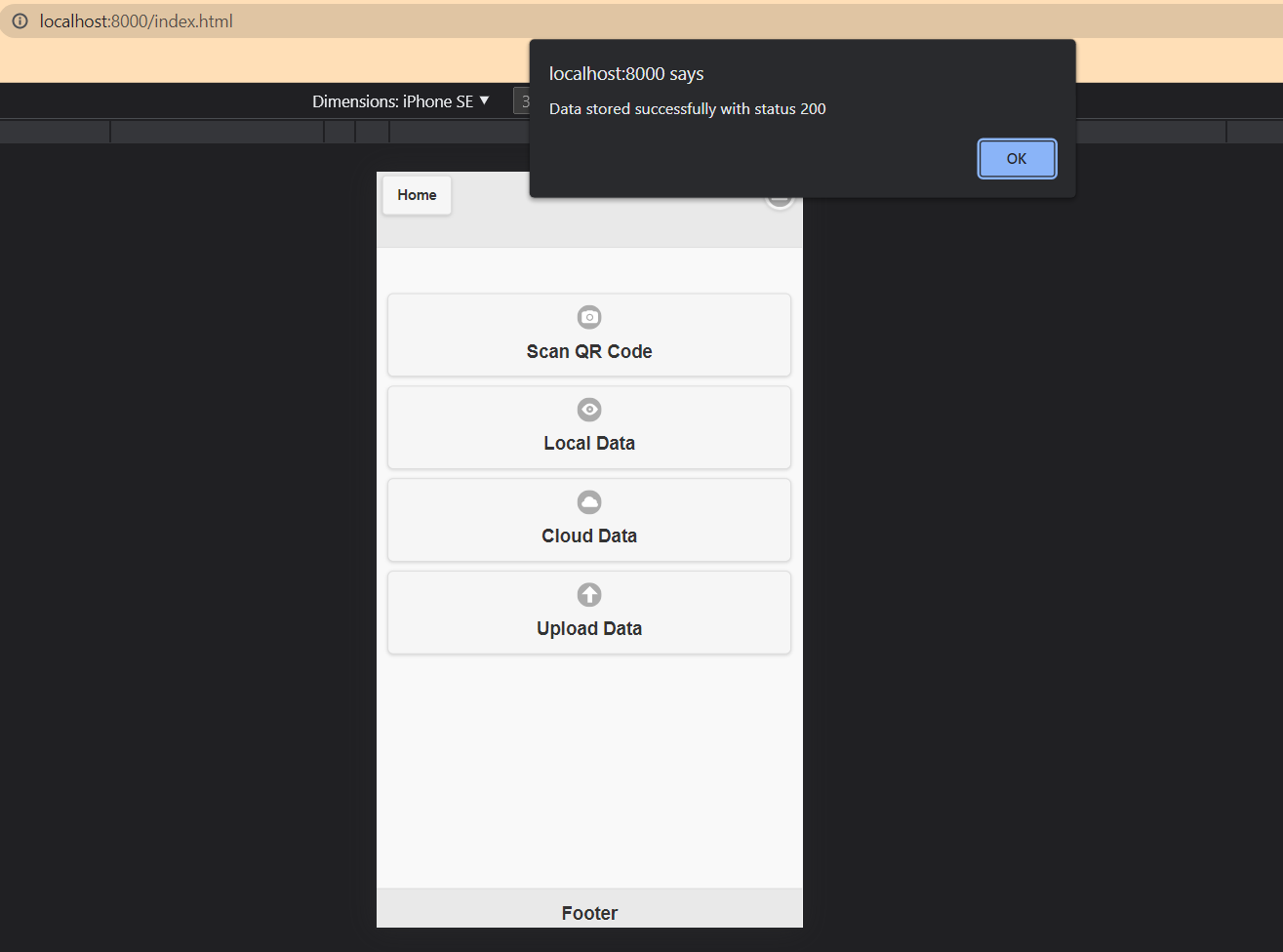


Figure Successful local storage store

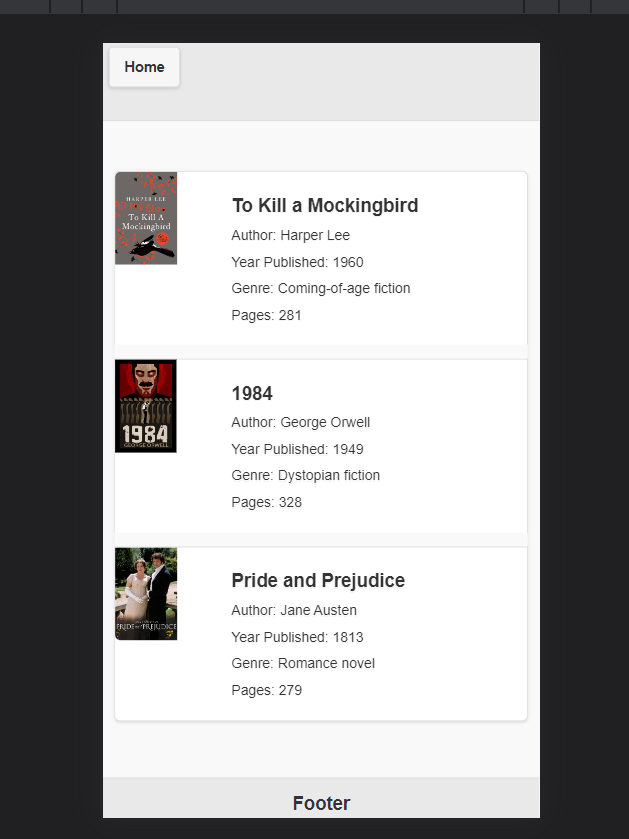


Figure Local storage data display

## 2. Upload data to cloud

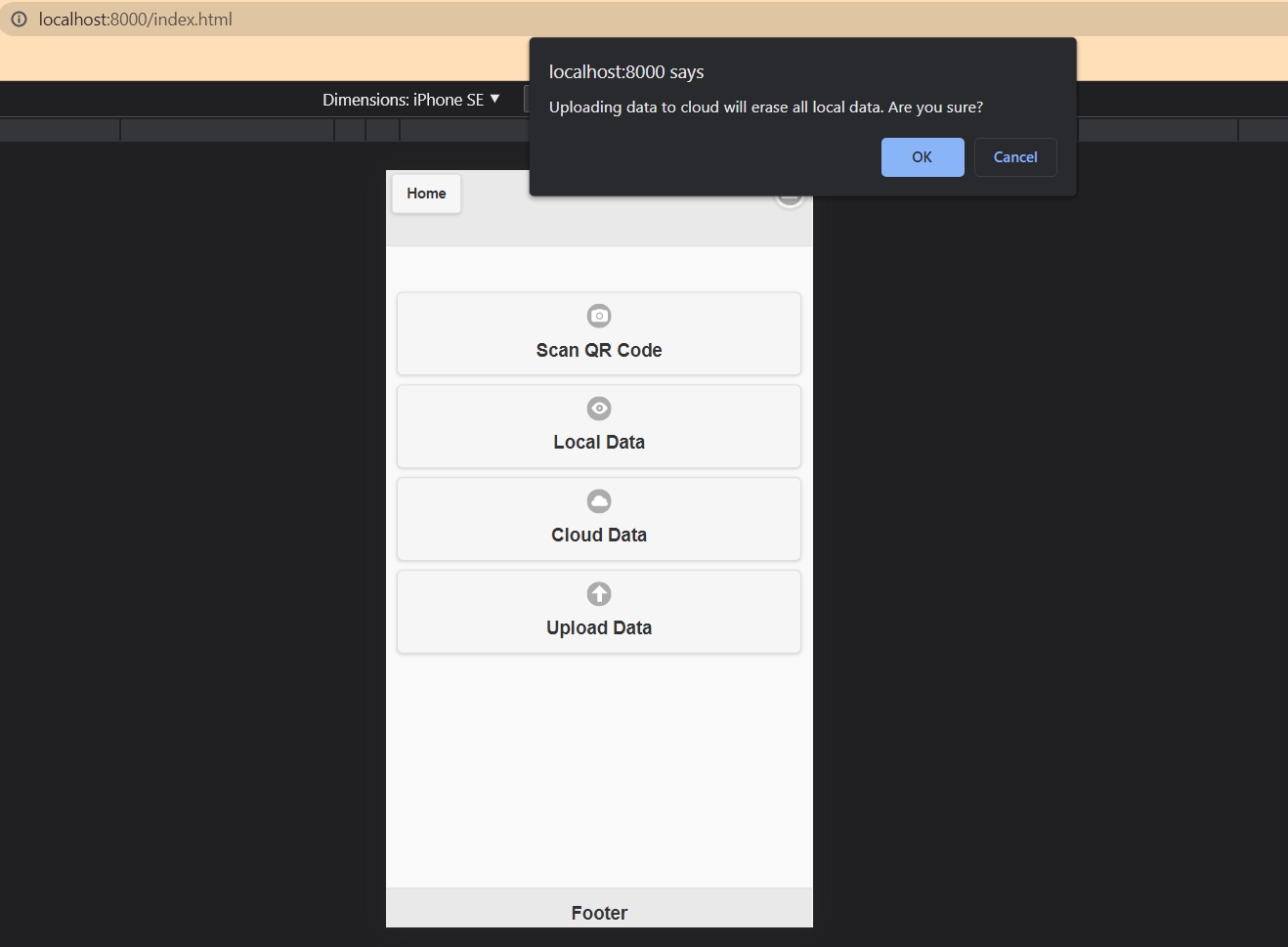


Figure Confirmation before upload

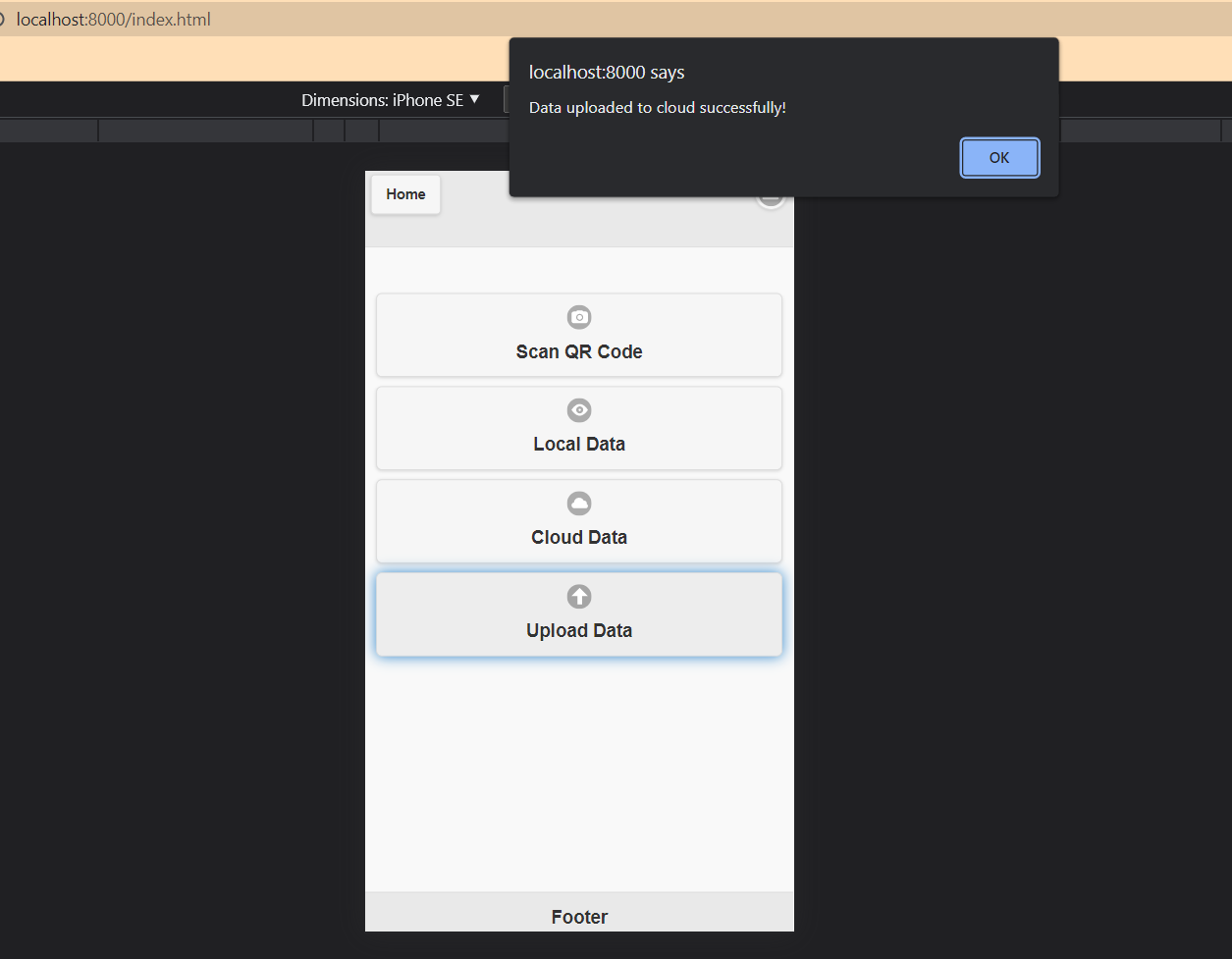


Figure On successful upload message

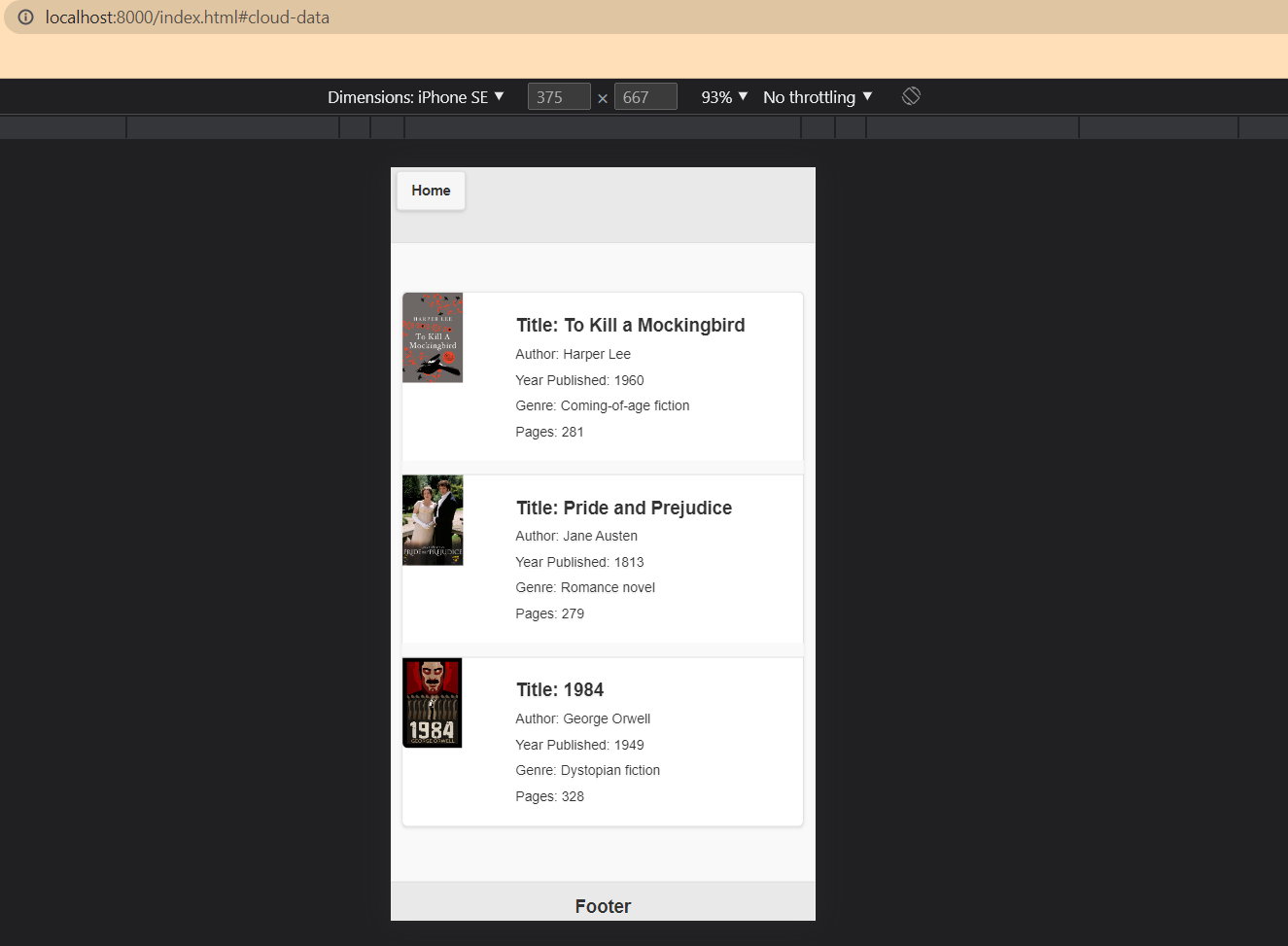


Figure Cloud data display

## 3. Delete cloud and local data

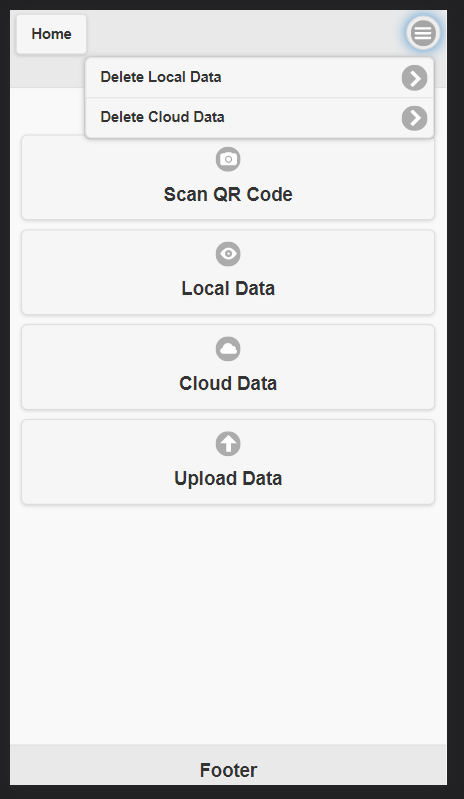


Figure Menu system

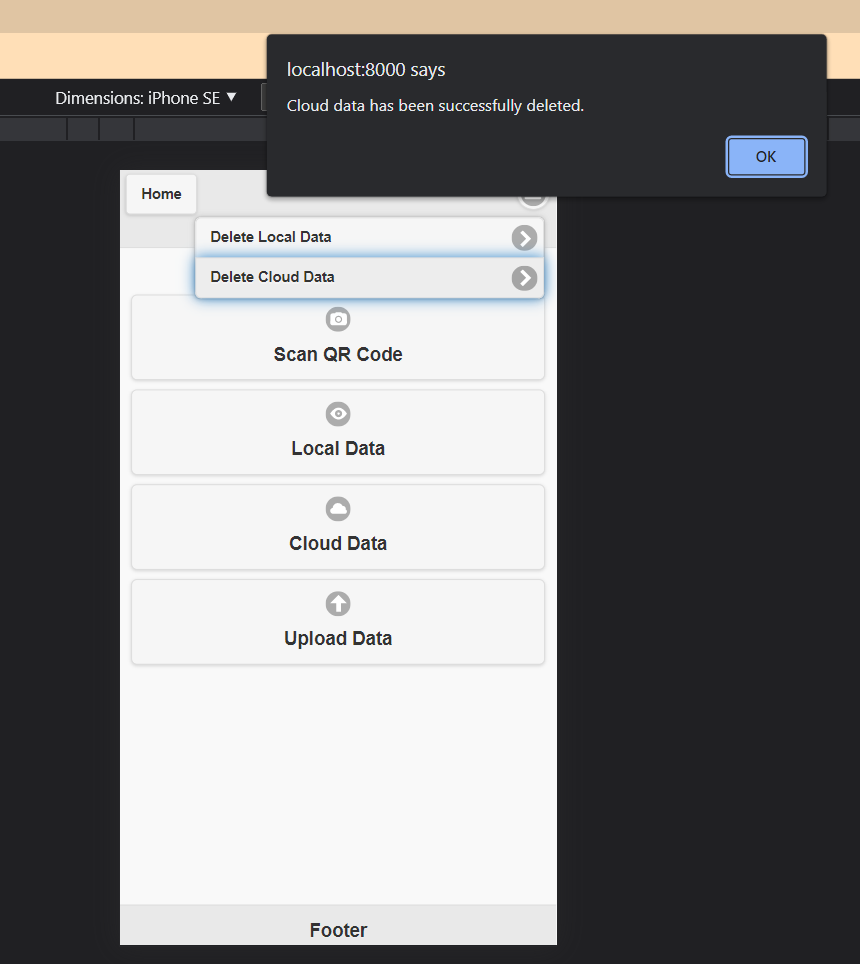


Figure Successful cloud data deletion

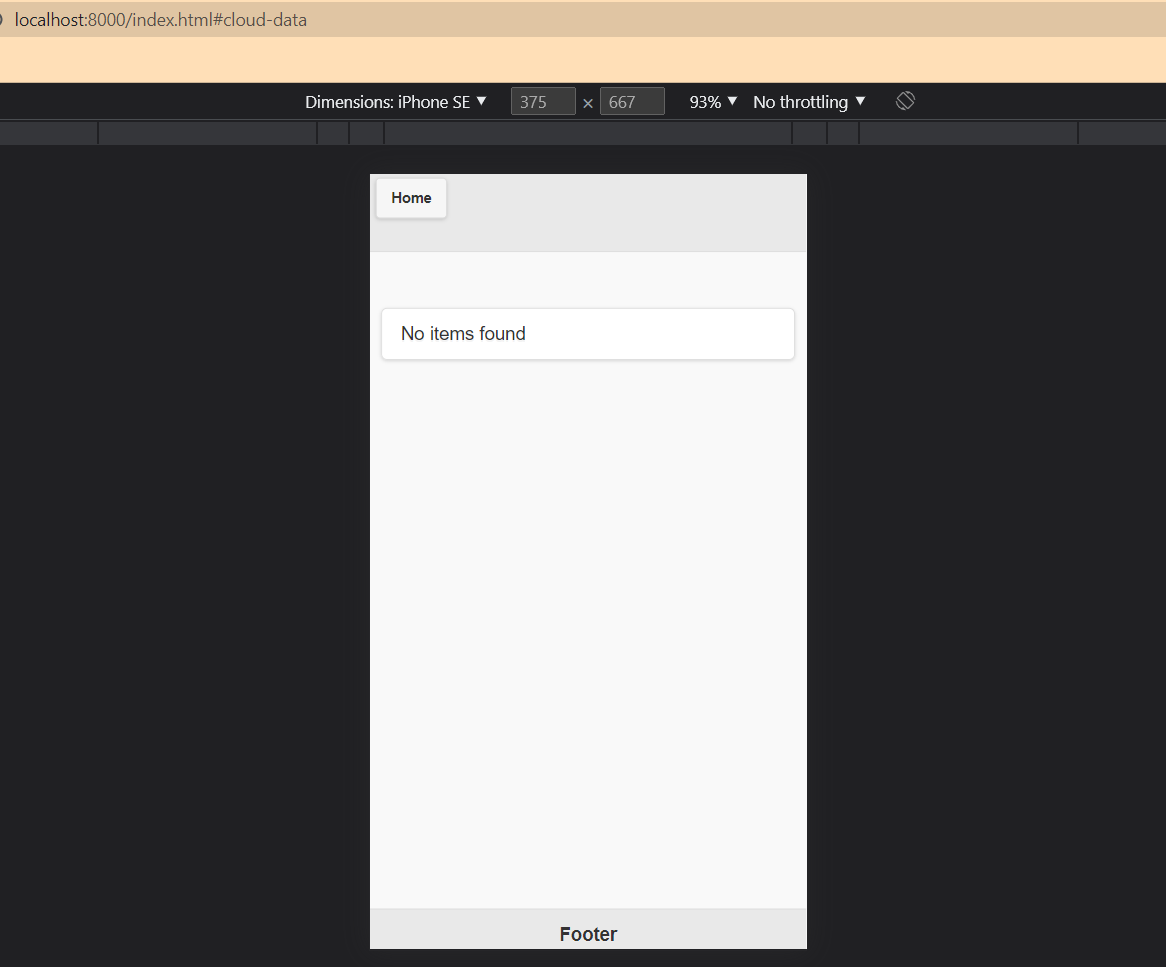


Figure Cloud data list empty

## 4. Smart device testing

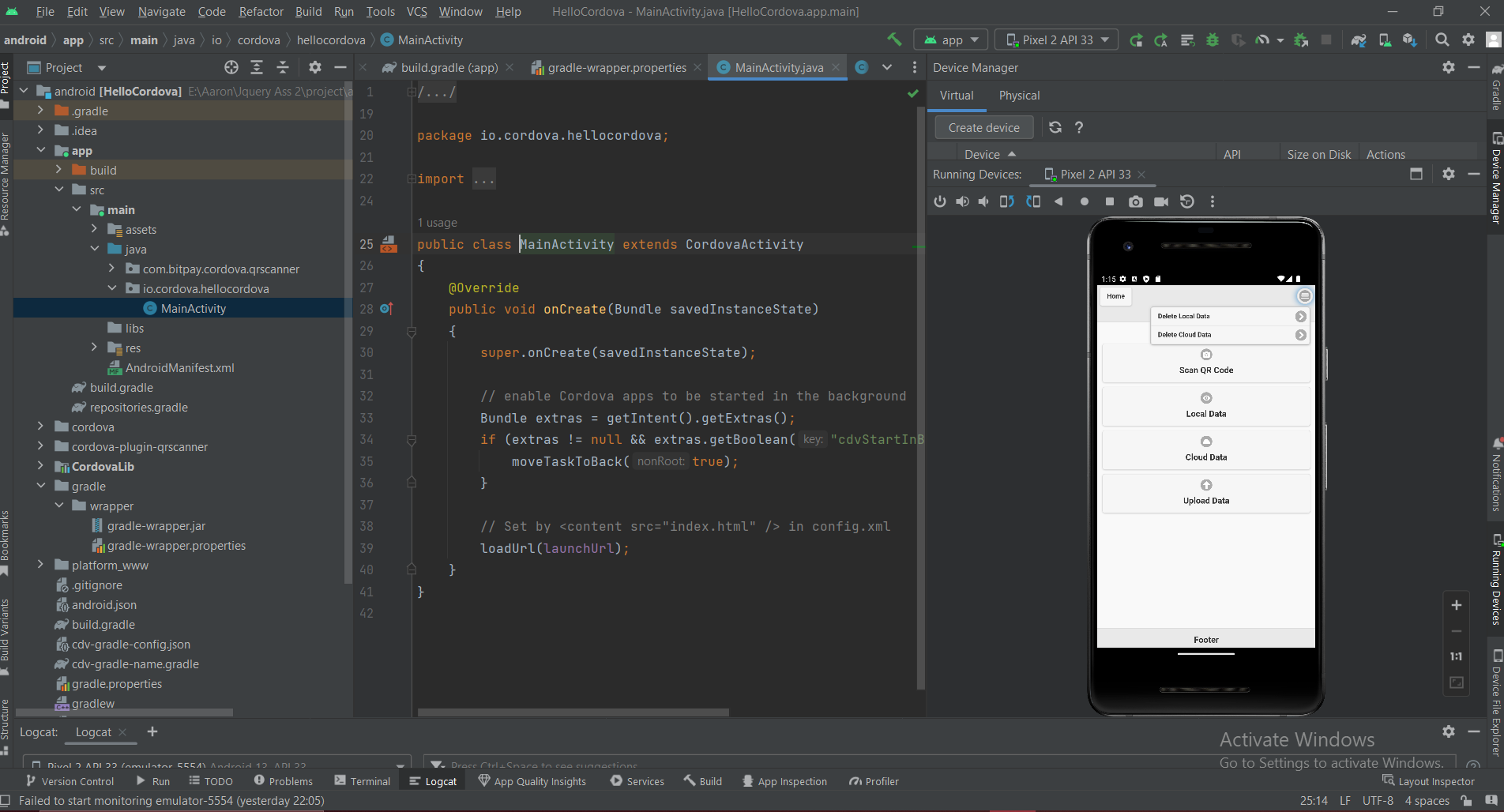


Figure App in android studio emulator