

# Installation Manual

The system's frontend is created in flutter/dart and all major backend services are making use of AWS services, specifically:

- AppSync
- DynamoDB
- Lambda
- Cognito
- IAM
- Amplify

This does not include location services which are making use of the Google Places API. The monorepo is managed by nx and the nx-rocks plugin.

## Perquisites

### Flutter (v3.05)

Download and install flutter (at least v3.05) for your operating system. Note you will need to unzip and set the path variable yourself.

Other dependencies such as android studio will need to be downloaded but the website as well as the "flutter doctor" command will assist you.

Link: <https://docs.flutter.dev/get-started/install/macos>

#### **flutter doctor**

Doctor summary (to see all details, run flutter doctor -v):

```
[✓] Flutter (Channel stable, 3.0.5, on macOS 12.4 21F79 darwin-arm, locale en-ZA)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Xcode - develop for iOS and macOS (Xcode 13.4.1)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.2)
[✓] VS Code (version 1.69.2)
[✓] Connected device (2 available)
[✓] HTTP Host Availability
```

• No issues found!

Example of what flutter doctor should output

## AWS Account

You will need an AWS developer account (note if this is your first time registering an account you are able to access certain feature for free for one year).

Link: <https://aws.amazon.com/>

Link: [AWS Free Tier](#)

## AWS Cli (v2.4.9)

Download and install the AWS Cli, this is used for a variety of AWS services. There are multiple installation options for each OS, use whichever one is preferred for you.

Link: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

## AWS Amplify Cli

Download and install the AWS Cli, this is use for managing your amplify app. Note the CLI can also be installed via npm.

Function	createChatResolver	No Change	awscloudformation
Function	sendMessageResolver	No Change	awscloudformation
Function	getConsumerMessagesResolver	No Change	awscloudformation

? Are you sure you want to continue? (Y/n) ☐

Example of Amplify Push

Link: <https://docs.amplify.aws/cli/start/install/>

## Google Places API

## Git & GitHub

A GitHub account is required as well as the latest version of git.

Link: <https://github.com/>

Link: <https://git-scm.com/downloads>

## NPM (v8.12.1)

Download and install NPM. NPM is used by Nx and Npx.

Link: <https://nodejs.org/en/download/>

## NX & Nx-Rocks

Download and install Nx as well as the nx-rocks plugin. Nx is used to manage the mono-repo along with the nx-rocks plugin to allow for flutter functionality.

Link: <https://www.npmjs.com/package/npx>

Link: <https://github.com/tinesoft/nxrocks/tree/develop/packages/nx-flutter>

## Installation

### 1. Clone the repository

Clone the repository using the following command:

git clone <https://github.com/COS301-SE-2022/ReverseHand.git>

### 2. Install all dependencies

In the root directory run the following commands

npm i

nx run-many --target=get --all

Theses commands will install all the npm as well as all the flutter dependencies.

### 3. Set Up AWS Services

1. Ensure you are in an AWS region that supports Amplify e.g. Ireland.
2. Create a new amplify project.
3. Navigate to the libs/amplify and pull the app.

```
Function    getConsumerMessagesResolver    No Change    awscloudformation

Local changes detected.
Pulling changes from the cloud will override your local changes.
? Are you sure you would like to continue? (y/N) █
```

Example of Amplify Pull

4. Navigate to the libs/amplify folder and run amplify push.

```
Function    getConsumerMessagesResolver    No Change    awscloudformation

? Are you sure you want to continue? (Y/n) █
```

Example of Amplify Push

5. Add the google API key to AWS secrets and link it to the existing lambda function that was pushed earlier.
6. Create two tables in dynamoDB one for users and one for adverts/bids/chat.
  1. Set up AWS Cogntio such that there are no cognito attributes.  
There must be two suer groups (Tradesman and Consumer).
  2. Enable the admin API.

### Deployment/Running

- To run the app since everything is on the cloud and serverless no backend needs to be started.
- Ensure a valid emulator running (either an iOS emulator running version 15 or Android minSDK 21)
- To run the app in the root directory run the following command  
nx run app:run
- If the app prints the message Amplify Successfully Configured 🎉 then you know that the app has successfully connected to the backend.

```
> nx run app:run

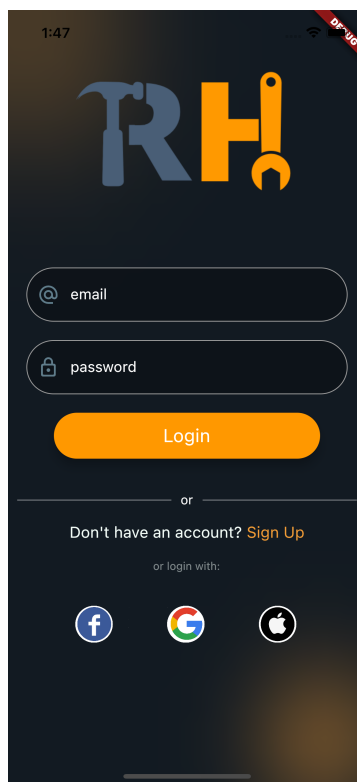
Launching lib/main.dart on iPhone 13 Pro in debug mode...
Running Xcode build...
  └─Compiling, linking and signing...                    3,2s
Xcode build done.                                         19,6s
Syncing files to device iPhone 13 Pro...                 381ms

Flutter run key commands.
r Hot reload. 🍻🍻🍻
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

🔥 Running with sound null safety 🔥

An Observatory debugger and profiler on iPhone 13 Pro is available at: http://127.0.0.1:59792/cj-4JFk79i4=/
The Flutter DevTools debugger and profiler on iPhone 13 Pro is available at: http://127.0.0.1:9101?uri=http://127.0.0.1:59792/cj-4JFk79i4=/
flutter: Amplify Successfully Configured 🎉
```

Example of running the command



Landing page that should be displayed.

## Useful Extensions for VSCode

- [Awesome Flutter Snippets](#)
- [Dart Language Extension](#)
- [Flutter](#)