

# Welcome to **SKILLZUPP** **POS**

We're so glad to have you join the team! This guide is meant to provide you with specific information to help you get off to a great and productive start with our software.

## BACKEND SETUP 2

1 . Firebase 2

2 . Configs Setup 2

3 . Environment Setup 3

Deployment

4

## APP SETUP

2

Flutter Setup

1 Flutter commands

2 Configs Setup 3

Deploy and Build apks

4

## HOW TO USE BACKEND

3

Modules 1 Inventory 2

Stalls 3 Reports

4

## HOW TO USE APP

4

Setting Device ID

1

User Authentication

POS

4

2 Device Inventory

3

# BACKEND SETUP

## 1 . Firebase

Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

→ Create a Firebase account. [Here](#)

→ Click on Add project to create a one.

◆ Enter name of your project

◆ Try to pick a short and precise name

◆ Choose your location (Pick the nearest one to your location)

◆ Click Continue to complete

→ Once done click on Database on the navigation Drawer

→ Click on Authentication

- ◆ Select Sign-in method and enable email sign in.
- Click on Database to enable firestore database
- Click on Storage and enable it
- Click on Hosting and enable it.
- Go to firebase Project
- ◆ Open 'Project settings'
- ◆ At 'Your apps', click at the web icon.
- ◆ Copy only the content of the var 'config'

## 2 . Configs Setup

Extract you zip file to your desired location. You will find Zip file named Backed\_Script, go ahead and extract that. Once done you will be able to identify two specific files

1. Backend
2. Frontend

### Backend :-

- Save and replicate the Firebase Admin SDK in these files.
  - ◆ /backend/service-accounts/localhost.json
  - ◆ /backend/service-accounts/test.json
  - ◆ /backend/service-accounts/development.json
  - ◆ /backend/service-accounts/production.json
- Configure the SMTP credentials, this is important in order to be able to send user invitations via emails, you need the SMTP credentials configured.
  - ◆ Open '/backend/config/<environment>.json
  - ◆ Locate the 'email' section.

- ◆ Configure the SMTP credentials. [Here](#)

#### Frontend :-

→ Configure Internationalization (i18n).

- ◆ For each locale at 'frontend/src/i18n/<locale>.js'
  - Go to entities section
  - Add the labels for the entity and its field.
  - Go to roles section
  - Replace the object name for the label → Copy the contents at the of the var 'config'.
- ◆ /frontend/src/config/localhost.js
- ◆ /frontend/src/config/development.js
- ◆ /frontend/src/config/production.js

**Refer to the documentation on the downloaded file  
For a more intense overview**

## 3 . Environment Setup

The backend is build with

1. Vue js -: A progressing JavaScript Framework, [here](#)
2. Element UI -: A vue 2.0 based component library for developers and designers
3. Node JS.

→ Download and install NodeJS. [here](#)

→ Install version 8.11 of NodeJS.

→ Update NPM

◆ Run : npm install -g npm

→ Install firebase-tools globally

◆ Run : npm install -g firebase-tools.

→ Login at firebase-cli and add your project.

◆ Run : firebase login

#### **Backend :-**

→ Open Backend file

→ Run : npm install

→ To run locally

→ Run : npm run start

#### **Frontend :-**

→ Open Frontend file

→ Run : npm install

→ To run locally

→ Run : npm run start

**Refer to the documentation on the downloaded file  
For a more intense overview**

## **APP SETUP**

### **Flutter setup**

→ Install Android Studio

- ◆ Android Studio offers a complete, integrated IDE experience for Flutter
- ◆ Download Android Studio, version [3.0](#) or Later. → Install the Flutter and Dart plugins

- ◆ Start Android Studio
- ◆ Open plugin preferences (File -> Settings -> Plugins)
- ◆ Select Marketplace
- ◆ Choose Flutter plugin and click install
- ◆ Click yes when prompted to install the Dart plugin.
- ◆ Click Restart when prompted.

## Flutter Commands

- flutter doctor.
- Flutter clean.
- Flutter get packages

Checkout this blog for a better understanding of flutter commands [her e](#)

This is how you create your first Flutter app [her e](#)

## Configs Setups

1. Extract and open App\_Script on your desired location
2. Open pubspec.yalm and edit line 1 and 2 with your name and description.
3. Go to /android/app/src/main/java and change package name
4. Go to /android/app/src/main and open Android Manifest and change package
5. Go to /android/app/src/profile and open Android Manifest and change package
6. Go to /android/app/src and open build.gradle and change package name
7. Go to /android/app and add google service file here
8. Go to /assets/images/ and replace app\_logo.png and logo.png to your respective icon logos
9. Go to /lib/Utils/ and open config.dart and change
  - a. app\_name
  - b. App\_url
  - c. App\_contact
  - d. App\_po\_box
  - e. App\_language
  - f. App\_currency
  - g. Language
  - h. Terms\_url
  - i. App\_title\_name
10. Go to /lib/Utils/ and open language.dart
  - a. Create a new var language object of your choice.
  - b. Add your object language to skeleton\_language\_objects

## **Deploy and Build apks**

- Adding a launcher icon [here](#)
- Signing the app [her e](#)
- Enabling Proguard [her e](#)
- Reviewing the app in manifest [her e](#)
- Review build configs [her e](#)
- Build the app for release [her e](#)
- Build an app bundle [her e](#)
- How to test app bundle [her e](#)

## **HOW TO USE APP**

**Modules**

**Inventory**

**Stalls**

**Report**

## **HOW TO USE App**

**Device Setup**

**Authentication**



Inventory

POS

