YCD Farmer Guide - Complete Setup Guide

This guide will help you set up and run the YCD Farmer Guide application on your own computer.

Table of Contents

---

1. Prerequisites

Install the following software on your computer:

Required Software

Software	Version	Download Link
Node.js	v18+ (LTS recommended)	https://nodejs.org/
npm	Comes with Node.js	-
Git	Latest	https://git-scm.com/
PostgreSQL	v14+	https://www.postgresql.org/download/
PostGIS	v3+	Usually bundled with PostgreSQL installer

Optional (for mobile development)

Software	Purpose	Download Link
Android Studio	Android Emulator	https://developer.android.com/studio
Expo Go App	Run on physical device	App Store / Play Store

Install Global npm Packages

```
npm install -g expo-cli eas-cli sequelize-cli
```

---

2. Clone the Repository

```
git clone https://github.com/laurentjoel78/YCDPlantApp-v2.git
cd YCDPlantApp-v2
```

Or if you have the project folder, just navigate to it:

```
cd path/to/YCD_App
```

---

3. Database Setup

Option A: Local PostgreSQL Database

Step 1: Install PostgreSQL with PostGIS

Windows:

1. Download PostgreSQL from https://www.postgresql.org/download/windows/
2. Run the installer and follow the setup wizard
3. **IMPORTANT:** Remember the password you set for the `postgres` user
4. When the installer finishes, it will open **Stack Builder**
5. In Stack Builder, select your PostgreSQL installation
6. Expand "Spatial Extensions" and check **PostGIS**
7. Complete the PostGIS installation

Mac:

```
brew install postgresql postgis
brew services start postgresql
```

Linux (Ubuntu/Debian):

```
sudo apt update
sudo apt install postgresql postgresql-contrib postgis
sudo systemctl start postgresql
```

Step 2: Create the Database

Choose ONE of the following methods:

Method 1: Using pgAdmin (Recommended for Windows - GUI)

1. **Open pgAdmin** (installed with PostgreSQL)
2. In the left panel, expand "Servers" → right-click "PostgreSQL" → "Connect"
3. Enter your postgres password

Create the Database: 4. Right-click "Databases" → "Create" → "Database..." 5. In "Database" field, enter: `ycd_farmer_guide` 6. Click "Save"

Enable Extensions: 7. Click on `ycd_farmer_guide` database to select it 8. Go to menu: **Tools** → **Query Tool** 9. In the query editor, paste and run these commands ONE BY ONE:

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

Click the ▶ Run button (or press F5)

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Click the ▶ Run button (or press F5)

10. You should see "Query returned successfully" for each command

---

Method 2: Using Windows Command Prompt

1. Open **Command Prompt** (search "cmd" in Start menu)
2. Navigate to PostgreSQL bin folder:

```
cd "C:\Program Files\PostgreSQL\16\bin"
```

(Replace `16` with your PostgreSQL version number)

3. Connect to PostgreSQL:

```
psql -U postgres
```

Enter your postgres password when prompted.

4. Run these commands in the PostgreSQL shell:

```
CREATE DATABASE ycd_farmer_guide;
\c ycd_farmer_guide
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
CREATE EXTENSION IF NOT EXISTS postgis;
\q
```

Method 3: Using Windows PowerShell

1. Open **PowerShell** (search "PowerShell" in Start menu)
2. Navigate to PostgreSQL bin folder:

```
cd "C:\Program Files\PostgreSQL\16\bin"
```

(Replace `16` with your PostgreSQL version number)

3. Connect to PostgreSQL:

```
.\psql.exe -U postgres
```

Enter your postgres password when prompted.

4. Run these commands in the PostgreSQL shell:

```
CREATE DATABASE ycd_farmer_guide;
\c ycd_farmer_guide
```

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
CREATE EXTENSION IF NOT EXISTS postgis;
\q
```

Method 4: Add PostgreSQL to PATH (Optional but Recommended)

This lets you run `psql` from anywhere without navigating to the bin folder.

Windows 10/11:

1. Search "Environment Variables" in Start menu
2. Click "Edit the system environment variables"
3. Click "Environment Variables..." button
4. Under "System variables", find "Path" and click "Edit"
5. Click "New" and add: `C:\Program Files\PostgreSQL\16\bin` (Replace `16` with your version)
6. Click "OK" on all dialogs
7. **Restart PowerShell/Command Prompt**

Now you can run from anywhere:

```
psql -U postgres
```

Option B: Use Neon (Cloud PostgreSQL - Free Tier)

This is the **easiest option** - no local installation required!

1. Go to https://neon.tech/
2. Create a free account
3. Create a new project named "ycd-farmer-guide"
4. Copy the connection string (it looks like: `postgresql://user:password@host/database?sslmode=require`)
5. PostGIS and UUID extensions are usually pre-installed

Advantages of Neon:

- No local PostgreSQL installation needed
- Free tier is generous (enough for development)
- Already configured for production use
- Accessible from anywhere

---

4. Backend Setup

Step 1: Navigate to Backend Folder

```
cd backend
```

Step 2: Install Dependencies

```
npm install
```

Step 3: Create the .env File

Create a file named `.env` in the `backend` folder with the following content:

```
# ==========================================
# DATABASE CONFIGURATION
# ==========================================
# Option 1: Use DATABASE_URL (recommended for Neon/cloud databases)
DATABASE_URL=postgresql://username:password@hostname:5432/ycd_farmer_guide?sslmode=require

# Option 2: Use individual settings (for local PostgreSQL)
# DB_HOST=localhost
# DB_PORT=5432
# DB_NAME=ycd_farmer_guide
# DB_USER=postgres
# DB_PASSWORD=your_postgres_password
# DB_DIALECT=postgres

# ==========================================
# SERVER CONFIGURATION
# ==========================================
PORT=3000
NODE_ENV=development

# ==========================================
# JWT AUTHENTICATION
# ==========================================
# Generate a random secret: run `node -e
"console.log(require('crypto').randomBytes(64).toString('hex'))"`
JWT_SECRET=your_super_secret_jwt_key_here_make_it_long_and_random
JWT_EXPIRES_IN=7d

# ==========================================
# EMAIL SERVICE (Brevo/Sendinblue)
# ==========================================
# Create free account at https://www.brevo.com/
BREVO_API_KEY=your_brevo_api_key
EMAIL_FROM=noreply@yourdomain.com
EMAIL_FROM_NAME=YCD Farmer Guide

# For development without email service:
USE_MOCK_EMAIL=true
AUTO_VERIFY_EMAILS=true

# ==========================================
# WEATHER API (OpenWeatherMap)
# ==========================================
# Create free account at https://openweathermap.org/api
OPENWEATHER_API_KEY=your_openweather_api_key

# ==========================================
# AI/ML SERVICES (Optional)
```

```
# ==========================================
# Groq API for AI features - https://console.groq.com/
GROQ_API_KEY=your_groq_api_key

# ==========================================
# FILE UPLOAD (Cloudinary - Optional)
# ==========================================
# Create free account at https://cloudinary.com/
CLOUDINARY_CLOUD_NAME=your_cloud_name
CLOUDINARY_API_KEY=your_api_key
CLOUDINARY_API_SECRET=your_api_secret

# ==========================================
# PAYMENT (Optional - for production)
# ==========================================
# PAYMENT_PROVIDER=stripe
# STRIPE_SECRET_KEY=your_stripe_secret_key

# ==========================================
# LOGGING
# ==========================================
LOG_LEVEL=info
```

Step 4: Get Your API Keys

Required Keys:

1. **JWT_SECRET** (Required)
    - Generate one by running:

```
node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"
```

2. **OpenWeatherMap API Key** (Free)
    - Go to https://openweathermap.org/api
    - Create an account
    - Go to "API Keys" section
    - Copy your API key

Optional Keys (can skip for basic development):

3. **Brevo Email API Key**
    - Go to https://www.brevo.com/
    - Create account → SMTP & API → API Keys
    - If skipping, set `USE_MOCK_EMAIL=true`
4. **Groq API Key** (for AI features)
    - Go to https://console.groq.com/
    - Create account → API Keys
5. **Cloudinary** (for image uploads)

- Go to https://cloudinary.com/
- Dashboard shows your credentials

Step 5: Run Database Migrations

```
# Make sure you're in the backend folder
cd backend

# Run all migrations
npx sequelize-cli db:migrate

# (Optional) Seed initial data
npx sequelize-cli db:seed:all
```

Step 6: Verify Backend Setup

```
# Start the backend server
npm start
```

You should see:

```
🚀 Starting YCD Farmer Guide Backend...
Database connection has been established successfully
Server is running on port 3000
```

---

5. Frontend Setup

Step 1: Navigate to Frontend Folder

```
# From the project root
cd frontend
```

Step 2: Install Dependencies

```
npm install
```

Step 3: Configure API URL

Edit `frontend/app.json` and update the `apiUrl` to match your backend:

```
{
  "expo": {
    "extra": {
      "apiUrl": "http://YOUR_LOCAL_IP:3000/api"
    }
```

```
  }
}
```

Finding your local IP address:

Windows:

```
ipconfig
# Look for "IPv4 Address" under your network adapter
```

Mac/Linux:

```
ifconfig
# Or
ip addr
# Look for your local IP (usually starts with 192.168.x.x or 10.x.x.x)
```

Step 4: Update API Service (for local development)

Create/edit `frontend/src/config/api.ts`:

```
// For local development on same network
export const API_BASE_URL = "http://YOUR_LOCAL_IP:3000/api";

// For production (Railway deployment)
// export const API_BASE_URL = 'https://your-app.up.railway.app/api';
```

---

6. Running the Application

Terminal 1: Start Backend

```
cd backend
npm start
```

Terminal 2: Start Frontend

```
cd frontend
npx expo start
```

This will show a QR code and options:

- Press `a` to open on Android emulator
- Press `i` to open on iOS simulator (Mac only)
- Press `w` to open in web browser
- Scan QR code with Expo Go app on your phone

---

7. Expo Account & Building

Step 1: Create Expo Account

1. Go to https://expo.dev/
2. Click "Sign Up"
3. Create your account

Step 2: Login via CLI

```
npx expo login
# Enter your Expo username and password
```

Step 3: Configure EAS (Expo Application Services)

```
cd frontend

# Initialize EAS for your project
eas init

# This will create/update eas.json and link to your Expo account
```

Step 4: Update app.json with Your Project ID

After running `eas init`, update `frontend/app.json`:

```
{
  "expo": {
    "name": "YCD Farmer Guide",
    "slug": "ycd-farmer-guide",
    "owner": "your_expo_username",
    "extra": {
      "eas": {
        "projectId": "your-project-id-from-expo"
      }
    }
  }
}
```

Step 5: Build the App

For Android APK (for testing):

```
eas build --platform android --profile preview
```

For Android App Bundle (for Play Store):

```
eas build --platform android --profile production
```

For iOS (requires Apple Developer account):

```
eas build --platform ios --profile production
```

---

8. Running on Emulator

Android Emulator (Android Studio)

1. **Install Android Studio**
   - Download from https://developer.android.com/studio
   - During setup, include "Android Virtual Device"
2. **Create Virtual Device**
   - Open Android Studio
   - Go to: Tools → Device Manager
   - Click "Create Device"
   - Choose a phone (e.g., Pixel 6)
   - Select a system image (API 33 or higher recommended)
   - Finish setup
3. **Start Emulator**
   - In Device Manager, click the Play button next to your device
4. **Run App on Emulator**

```
cd frontend
npx expo start
# Press 'a' to open on Android
```

iOS Simulator (Mac Only)

1. Install Xcode from App Store
2. Open Xcode → Preferences → Locations → Select Command Line Tools
3. Run:

```
cd frontend
npx expo start
# Press 'i' to open on iOS Simulator
```

---

9. Running on Physical Device (Expo Go)

Step 1: Install Expo Go

- **Android:** Download "Expo Go" from Google Play Store

- **iOS:** Download "Expo Go" from App Store

Step 2: Connect to Same Network

IMPORTANT: Your phone and computer must be on the same WiFi network!

Step 3: Configure Backend for Network Access

Update backend to listen on all interfaces (already configured in this project):

```
// backend/src/index.js
server.listen(PORT, "0.0.0.0", () => {
  // This allows connections from other devices on the network
});
```

Step 4: Update Frontend API URL

In `frontend/app.json`, set the API URL to your computer's local IP:

```
{
  "expo": {
    "extra": {
      "apiUrl": "http://192.168.1.100:3000/api"
    }
  }
}
```

Replace `192.168.1.100` with your actual local IP address.

Step 5: Start Both Servers

Terminal 1 (Backend):

```
cd backend
npm start
```

Terminal 2 (Frontend):

```
cd frontend
npx expo start
```

Step 6: Scan QR Code

1. Open Expo Go on your phone
2. Scan the QR code shown in the terminal
3. The app will load on your device

---

10. Troubleshooting

Common Issues

"Cannot connect to backend"

1. Check that backend is running: `npm start` in backend folder
2. Verify your local IP is correct
3. Make sure phone and computer are on same WiFi
4. Check firewall settings - allow Node.js through firewall

Windows Firewall:

```
Control Panel → System and Security → Windows Firewall → Allow an app through
→ Find Node.js and allow both Private and Public
```

"Database connection failed"

1. Check PostgreSQL is running:

```
# Windows
net start postgresql-x64-14

# Mac
brew services start postgresql

# Linux
sudo systemctl start postgresql
```

2. Verify database exists:

```
psql -U postgres -l
```

3. Check .env credentials match your database setup

"Migration failed"

1. Make sure PostGIS extension is installed:

```
CREATE EXTENSION IF NOT EXISTS postgis;
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

2. Run migrations with verbose output:

```
npx sequelize-cli db:migrate --debug
```

"Expo Go can't find the app"

1. Try tunnel mode instead of LAN:

```
npx expo start --tunnel
```

2. This uses ngrok to create a public URL (slower but works across networks)

"Build failed on EAS"

1. Make sure you're logged in:

   ```
   npx expo whoami
   ```

2. Check your app.json has correct projectId

3. View build logs on https://expo.dev/

Quick Start Commands Summary

```
# Clone and setup
git clone https://github.com/laurentjoel78/YCDPlantApp-v2.git
cd YCDPlantApp-v2

# Backend setup
cd backend
npm install
# Create .env file with your keys
npx sequelize-cli db:migrate
npm start

# Frontend setup (new terminal)
cd frontend
npm install
npx expo start

# Build APK
cd frontend
eas build --platform android --profile preview
```

Environment Variables Quick Reference

Variable	Required	Description	Where to Get
DATABASE_URL	Yes	PostgreSQL connection string	Neon.tech or local setup
JWT_SECRET	Yes	Secret for token signing	Generate with crypto
OPENWEATHER_API_KEY	Yes	Weather data	openweathermap.org
BREVO_API_KEY	No*	Email service	brevo.com
GROQ_API_KEY	No	AI features	console.groq.com
CLOUDINARY_*	No	Image uploads	cloudinary.com

*Set `USE_MOCK_EMAIL=true` if not using email service

Support

If you encounter issues:

1. Check this guide's troubleshooting section
2. Review console/terminal error messages
3. Check the project's GitHub issues page

Good luck! 🌱