# Ags-AI User Guide: Simple Step-by-Step Instructions for Everyone

This guide is written specifically for farm managers, estate supervisors, and agronomists who need to understand their soil and leaf test results without getting bogged down in technical jargon. Think of it like a cooking recipe - we’ll tell you exactly what ingredients you need, what each step does, and what your final dish should look like. No computer science degree required!

## 1. What Exactly Is Ags-AI and Why Should You Care?

Imagine you’re a chef checking if your kitchen has enough ingredients to make a great meal. Ags-AI does the same thing for your palm oil plantation - it examines your soil and leaf test results and compares them against the “perfect recipe” established by palm oil experts.

**What it does in simple terms:** - Takes your lab test results (the numbers from soil samples and leaf samples) - Compares them to the ideal levels set by the Malaysian Palm Oil Board (MPOB) - these are like the recommended measurements in a cooking recipe - Shows you exactly where your plantation’s “ingredients” are too low, just right, or maybe even too high - Creates a professional report that you can share with your team or bosses

**Who this helps most:** - Estate managers who need to make fertilizer decisions - Agronomists tracking plantation health - Farm supervisors explaining problems to owners - Anyone who wants to understand nutrient problems without studying chemistry

**What you’ll get:** - Easy-to-read tables (no complicated graphs or formulas) - Color-coded warnings (green = good, yellow = watch carefully, red = action needed now) - A PDF report that looks professional enough to share in meetings

**Real-world example:** If your soil test shows potassium levels are 20% below what’s recommended, Ags-AI will flag this as “Critical” and help you calculate how much fertilizer to add to fix it.

## 2. What Can You Actually Do With This App?

Ags-AI isn’t just another computer program - it’s a practical tool that gives you actionable insights about your plantation’s health. Here’s what you can accomplish:

### Main Features:

1. **Upload Your Test Results**
   * Drag and drop soil test files from your lab
   * Add leaf analysis results from plant tissue tests
   * Works with files from different labs or even simple spreadsheets you create yourself
2. **Get a “Nutrient Health Check” Report**
   * See all your nutrients in one easy table
   * Problems are listed from most serious to least serious (so you tackle the big issues first)
   * Each nutrient shows: your current level, the ideal level, and how far off you are
3. **Understand Severity Levels at a Glance**
   * **Balanced (Green)**: Your levels are very close to perfect - like having just the right amount of sugar in a recipe
   * **Low (Yellow)**: There’s a small shortage - like running low on cooking oil; not an emergency but worth watching
   * **Critical (Red)**: Big problem that needs immediate attention - like discovering you’re completely out of salt mid-recipe
4. **Create Professional Reports**
   * Generate a PDF that looks exactly like what you see on screen
   * Perfect for sharing with fertilizer suppliers, farm owners, or regulatory bodies
   * Includes all your data plus the expert analysis
5. **Share With Your Team**
   * Run it on your office computer for local use
   * Put it online so remote team members can access it
   * Package it as a portable app for different locations

**What this saves you:** Instead of spending hours comparing numbers manually or paying consultants, you get instant analysis that tells you exactly which nutrients need attention and how urgently.

## 3. What You Need Before You Start

Don’t worry - you don’t need fancy equipment or computer skills. Here’s your shopping list:

### Essential Items (You Must Have These):

* **A Windows Computer**: Windows 10 or 11 works perfectly. Any modern office computer will do.
* **A Web Browser**: Google Chrome, Microsoft Edge, or Firefox. (You probably already have one of these.)
* **Your Test Data**: Soil and/or leaf test results from your lab. These are usually PDF files or Excel spreadsheets.

### Optional Software (Only if running locally):

* **Python 3.10 or newer**: This is free software that runs the app. Think of it as the engine that makes the car go.
* **Your Lab Files**: Bring your actual test results to see real analysis

**Cost**: Everything is free except your computer (which you already have). No subscriptions, no hidden fees.

**Pro tip**: The app includes sample files in the “json” folder, so you can practice with fake data before using your real results.

## 3.1 Getting the Ags-AI Software

Before you can run Ags-AI, you need to get the application files on your computer. You have two main options:

### Option A: Download the Ready-to-Use Package (Easiest for Most Users)

**Perfect for:** Users who just want to run the app without making changes

**How to get it:** 1. Visit the Ags-AI GitHub page: <https://github.com/Markishime/Ags-AI> 2. Look for the “Releases” section (usually on the right side or under the repository description) 3. Download the latest release (usually a ZIP file named something like “Ags-AI-v1.0.2.zip”) 4. Extract/unzip the downloaded file to a folder on your computer (like Desktop or Documents)

**What you’ll get:** A complete, ready-to-run package with all files included.

### Option B: Clone the Repository (For Advanced Users or Developers)

**Perfect for:** Users who want the latest code, plan to contribute changes, or use Git version control

**What you’ll need:** Git installed on your computer (free download from git-scm.com)

**How to clone:** 1. Open PowerShell or Command Prompt on your computer 2. Navigate to where you want to place the project (e.g., cd Desktop) 3. Run this command: git clone https://github.com/Markishime/Ags-AI.git 4. Wait for the download to complete

**What happens:** This creates a new folder called “Ags-AI” with all the application files.

### Option C: Fork the Repository (For Contributing or Customizing)

**Perfect for:** Users who want to modify the code or contribute improvements back to the project

**How to fork:** 1. Go to <https://github.com/Markishime/Ags-AI> 2. Click the “Fork” button in the top-right corner of the page 3. GitHub will create a copy of the repository under your own GitHub account 4. Clone your forked version using the clone instructions above

**What this gives you:** Your own copy of the project that you can modify freely.

### Which Option Should You Choose?

* **Choose Option A (Download)** if you just want to use Ags-AI as-is
* **Choose Option B (Clone)** if you want the latest version and are comfortable with Git
* **Choose Option C (Fork)** if you plan to make changes or contribute to the project

**Next step:** Once you have the files on your computer (using any of the above options), proceed to Section 5 below.

## 5. Getting Started: Running the App on Your Computer (Beginner-Friendly Method)

This is like setting up a new kitchen appliance. We’ll go through it step by step, and it should take about 15-20 minutes the first time. Don’t rush - we’ll explain why each step matters.

### Step 1: Get Python (The App’s Engine)

**What this does:** Python is the software that makes Ags-AI work, like gasoline makes a car run.

**How to do it:** 1. Open your web browser and go to python.org 2. Click “Downloads” and choose “Python 3.10.x” for Windows 3. Download and run the installer 4. **Important**: When asked, check the box that says “Add Python to PATH” - this tells Windows where to find Python 5. Click “Install Now” and wait for it to finish

**What to expect:** You’ll see a success message. Close the installer when done.

**Common mistake:** Forgetting to check “Add Python to PATH” - if you do this, the later steps won’t work.

### Step 2: Open the Ags-AI Folder

**What this does:** This gets you to the app’s files, like opening the box that contains your new tool.

**How to do it:** 1. Find the Ags-AI folder you downloaded (usually on your Desktop or Downloads) 2. Right-click inside the folder (not on a file) 3. Choose “Open in Terminal” or “Open PowerShell window here” - This opens a black command window (don’t be scared - it’s just a text-based way to give instructions to your computer)

**What to expect:** A window that looks like old computer screens from movies, with text like “C:-AI>”

### Step 3: Create a Safe Workspace (Virtual Environment)

**What this does:** This creates a “clean room” for the app so it doesn’t interfere with other programs on your computer. Like having a separate kitchen for cooking to avoid mixing ingredients with cleaning supplies.

**How to do it:** 1. In the PowerShell window, type exactly: python -m venv .venv 2. Press Enter and wait (it might take a few seconds)

**What happens next:** Activate the workspace by typing: .\.venv\Scripts\activate **What to expect:** Your command prompt will change to show (.venv) at the beginning. This means you’re now in the safe workspace.

**Why this matters:** Without this step, the app might conflict with other software on your computer.

### Step 4: Install the App’s Helper Tools

**What this does:** Downloads and installs all the free tools the app needs to work, like gathering ingredients before cooking.

**How to do it:** 1. Make sure you’re still in the activated environment (you should see (.venv) in the prompt) 2. Type: pip install -r requirements.txt 3. Press Enter and wait (this might take 2-5 minutes)

**What to expect:** You’ll see text scrolling by as it downloads things. At the end, you should see “Successfully installed…” messages.

**Common issue:** If this fails, try updating pip first: pip install -U pip

### Step 5: Set Up Basic Settings (Optional but Recommended)

**What this does:** Creates a settings file that tells the app how to behave.

**How to do it:** 1. In the Ags-AI folder, find the file called .streamlit/secrets.example.toml 2. Make a copy of it and rename the copy to .streamlit/secrets.toml 3. Open the new file in Notepad and add these lines: [app] environment = "production" log\_level = "INFO"

**What this does:** Tells the app it’s ready for real use and to show helpful messages.

**Note:** You can skip this for now and come back to it later if you want advanced features like passwords.

### Step 6: Launch the App

**What this does:** Starts the app and opens it in your web browser.

**How to do it:** 1. Make sure you’re still in the activated environment ((.venv) visible) 2. Type: streamlit run app.py 3. Press Enter

**What to expect:** - Your web browser will automatically open to http://localhost:8501 - If it doesn’t open automatically, copy that web address from the PowerShell window and paste it into your browser - You’ll see the Ags-AI welcome page

**Important:** Keep the PowerShell window open while using the app. Closing it stops the app.

**Success check:** You should see the Ags-AI interface in your browser, just like a website but running on your computer.

## 6. Uploading Your Test Data

Now that the app is running, let’s add your actual data. This is where the real analysis happens.

### Step-by-Step Upload Process:

1. **Find the Upload Section**
   * Look for an “Upload” tab or button in the app (usually at the top or side)
   * Click it to go to the upload page
2. **Select Your Files**
   * Click “Browse” or “Choose Files”
   * Find your soil and/or leaf test files
   * You can select multiple files at once
   * **Practice tip:** Try the example files in the “json” folder first to see how it works
3. **Start the Analysis**
   * Click “Upload” or “Analyze”
   * Wait while the app processes your data (usually just a few seconds)

### What the App Does With Your Data:

* **Reads your numbers:** Extracts all the nutrient values from your lab reports
* **Calculates averages:** If you have multiple tests, it combines them intelligently
* **Compares to standards:** Checks each nutrient against MPOB recommendations
* **Identifies gaps:** Shows where you’re above, at, or below ideal levels

### File Format Tips:

* **Works with:** PDF lab reports, Excel spreadsheets, CSV files, JSON data
* **Units matter:** Make sure your file uses standard units (we’ll explain these in Section 10)
* **Missing data:** If a nutrient isn’t tested, it shows “N/A” - that’s normal, the app just skips it

### What to Expect:

* A progress message while uploading
* Success confirmation when complete
* Automatic redirect to results page
* No errors if your file format is supported

### Common Upload Issues:

* **File too large:** Check size limits in CONFIGURATION.md
* **Wrong format:** Try saving as CSV or Excel first
* **Test with samples:** Always try the example files first to make sure everything works

## 7. Understanding Your Results (The Important Part!)

This is where Ags-AI does its magic. Instead of staring at confusing lab numbers, you get clear answers about your plantation’s health.

### The Main Report: Nutrient Gap Analysis

This is the star of the show - a table that ranks all your nutrient problems from most urgent to least urgent.

### What Each Column Means:

1. **Nutrient Name** (like “Soil pH” or “Leaf Potassium”)
   * The specific nutrient being measured
   * Grouped by soil nutrients and leaf nutrients
2. **Your Average Value**
   * What your tests actually measured
   * If you uploaded multiple samples, this is the combined average
3. **MPOB Recommended Minimum**
   * The “target” level set by palm oil experts
   * Think of this as the passing grade for healthy palms
4. **Gap (Difference)**
   * How far you are from the target
   * Positive numbers = you’re above target (usually good)
   * Negative numbers = you’re below target (needs attention)
5. **Percent Gap**
   * The gap expressed as a percentage
   * Makes it easy to compare different nutrients
6. **Severity Level** (Color-Coded)
   * **Balanced (Green):** Gap is 5% or less - you’re in the safe zone
   * **Low (Yellow):** Gap is 5-15% - minor issue, monitor but not urgent
   * **Critical (Red):** Gap over 15% - major problem requiring immediate action

### How the Table Is Organized:

* **Sorted by priority:** Critical problems appear first, then Low, then Balanced
* **Within each severity:** Largest gaps at the top (biggest problems first)
* **Color coding:** Makes it easy to spot issues at a glance

### Real-World Example:

Imagine your soil potassium shows: - Your average: 0.8 meq/100g - MPOB minimum: 1.0 meq/100g - Gap: -0.2 (20% below target) - Severity: Critical (Red)

This means you need to add potassium fertilizer soon to prevent yield losses.

### What to Look For:

* **Focus on negatives first:** Deficiencies hurt production more than excesses waste money
* **Critical items:** These are your top priorities for the next fertilizer application
* **Trends:** Compare results across seasons to see if problems are getting better or worse

### Understanding the Numbers:

* **Small gaps (Balanced):** Your plantation is well-nourished for these nutrients
* **Medium gaps (Low):** Watch these nutrients - they might need attention soon
* **Large gaps (Critical):** These are costing you money in lost production

## 8. Creating and Sharing PDF Reports

Once you have results you want to save or share, the PDF export makes it professional.

### How to Create a PDF:

1. **Navigate to Results**
   * Make sure you’re on the page showing your analysis tables
2. **Find the Export Button**
   * Look for “Export PDF”, “Download PDF”, or “Generate Report”
   * Usually located near the top or bottom of the results page
3. **Generate the Report**
   * Click the button
   * Wait a moment while it creates the file
   * Your browser will automatically download the PDF

### What the PDF Contains:

* **Exact copy:** Everything you see on screen, including colors and formatting
* **Professional layout:** Clean tables that look good in presentations
* **All data:** Your averages, MPOB standards, gaps, and severity levels
* **Timestamp:** Shows when the report was generated

### How to Use the PDF:

* **Email to team:** Share findings with fertilizer suppliers or farm managers
* **Archive records:** Keep historical records of your nutrient status
* **Meeting presentations:** Professional-looking document for stakeholders
* **Regulatory compliance:** Some areas require documented nutrient management

### PDF Tips:

* **File name:** Automatically includes date and time
* **File size:** Usually small (under 1MB) so easy to email
* **Compatibility:** Opens in any PDF reader (free ones available)
* **Printing:** Looks good when printed in color

### Troubleshooting PDF Issues:

* **Doesn’t match screen:** Refresh the page and try again
* **Colors look wrong:** Make sure your browser isn’t in grayscale mode
* **File corrupted:** Clear browser cache and regenerate

## 9. Basic Settings (Keep It Simple for Now)

Most users can skip this section initially. The app works fine with default settings.

### The Settings File:

* **Location:** .streamlit/secrets.toml (the file we created earlier)
* **Purpose:** Controls app behavior and optional features

### Minimal Settings for Local Use:

[app]  
environment = "production"  
log\_level = "INFO"

**What these mean:** - environment = "production": Tells the app this is real use (not testing) - log\_level = "INFO": Shows helpful messages but not too much detail

### Advanced Features (Optional):

These require additional setup and are mainly for larger teams:

1. **Password Protection**
   * Uses Google’s Firebase service (free)
   * Requires Firebase project setup
2. **AI Document Reading**
   * Helps read messy or handwritten lab reports
   * Uses Google Document AI (has costs)
3. **Email Alerts**
   * Automatically sends reports to team members
   * Requires email server settings

### Important Security Notes:

* **Never share secrets.toml** with real passwords or API keys
* **Use the example file** as your template
* **Keep backups** of your settings
* **Test changes** on a copy first

## 10. Sharing the App With Your Team

Choose the method that fits your team’s needs and technical comfort level.

### Option A: Local Sharing (Best for Small Teams)

**Perfect for:** Office environments, same building users

**How it works:** 1. Run the app on one computer following Section 4 2. Other team members open web browsers on the same network 3. They visit: http://[computer-name]:8501 (the app will show the exact address)

**Advantages:** - No internet required - Full control over data - Easy to update

**Requirements:** All users on same local network

### Option B: Online Sharing via Streamlit Cloud (Easiest)

**Perfect for:** Remote teams, different locations

**Setup steps:** 1. **Create GitHub account** (free at github.com) 2. **Upload your Ags-AI folder** to GitHub 3. **Go to share.streamlit.io** 4. **Connect your GitHub** and select the repository 5. **Configure the app:** - Main file: app.py - Python version: 3.10 - Requirements file: requirements.txt 6. **Add your secrets** in their secrets section 7. **Deploy** and get your shareable web link

**Advantages:** - Works from anywhere with internet - No installation for users - Automatic updates possible

**Cost:** Free tier available, paid plans for heavy use

### Option C: Docker Container (For IT Teams)

**Perfect for:** Company servers, consistent deployment

**What you’ll need:** Ask your IT department for help with this method

**Basic steps:** 1. **Install Docker** (free software) 2. **Open PowerShell in the Ags-AI folder** 3. **Build container:** docker build -t ags-ai:latest . 4. **Run container:** docker run --rm -p 8501:8501 -v ${PWD}\.streamlit:/app/.streamlit ags-ai:latest 5. **Access at:** http://localhost:8501

**Advantages:** - Consistent across different computers - Professional deployment - Easy to backup and restore

**Note:** Requires Docker knowledge or IT support

### Choosing the Right Method:

* **Solo use:** Local installation (Section 4)
* **Small team, same office:** Local sharing
* **Remote access needed:** Streamlit Cloud
* **Company-wide deployment:** Docker with IT help

## 11. Understanding Your Nutrients (What Each One Means)

Don’t worry about memorizing chemistry - here’s what matters for palm oil production:

### Soil Nutrients (What Your Soil Provides):

1. **pH Level** (Soil Acidity/Basic-ness)
   * **What it measures:** How acidic or basic your soil is (scale from 0-14)
   * **Why it matters:** Extreme pH prevents plants from absorbing nutrients
   * **Ideal range:** 4.5-6.5 for oil palms
   * **Problems:** Too acidic (low pH) locks up nutrients; too basic (high pH) causes deficiencies
   * **Real impact:** Poor pH can reduce yields by 20-30%
2. **Available Phosphorus (P)**
   * **What it does:** Helps young roots grow strong and healthy
   * **Critical for:** Early plant establishment and fruit development
   * **Deficiency signs:** Stunted growth, poor root systems
   * **Sources:** Rock phosphate, superphosphate fertilizers
3. **Exchangeable Potassium (K)**
   * **What it does:** Controls water movement in plants, improves oil quality
   * **Critical for:** Drought resistance and fruit filling
   * **Deficiency signs:** Leaf yellowing, reduced bunch weight
   * **Sources:** Muriate of potash, sulfate of potash
4. **Calcium (Ca) & Magnesium (Mg)**
   * **What they do:** Build strong cell walls, help create chlorophyll (green color)
   * **Critical for:** Overall plant strength and photosynthesis
   * **Deficiency signs:** Leaf tip burning, poor fruit development
   * **Balance matters:** Too much of one can block absorption of the other
5. **Organic Carbon (OC)**
   * **What it measures:** Soil health and organic matter content
   * **Why it matters:** Healthy soil holds water and nutrients better
   * **Ideal level:** Above 2-3% for good palm growth
   * **Improvement:** Add compost, manure, or practice conservation tillage
6. **Cation Exchange Capacity (CEC)**
   * **What it measures:** Soil’s ability to hold onto nutrients
   * **Why it matters:** High CEC means less fertilizer leaching
   * **Good levels:** Above 10-15 meq/100g for palms
   * **Affected by:** Clay content and organic matter

### Leaf Nutrients (What’s Actually in Your Plants):

1. **Nitrogen (N)**
   * **What it does:** Fuel for growth and green leaf production
   * **Critical for:** Leaf expansion and overall vigor
   * **Deficiency signs:** Yellow leaves, slow growth
   * **Excess signs:** Dark green leaves, excessive vegetative growth
2. **Phosphorus (P) & Potassium (K)**
   * **Same as soil versions but measured in leaves**
   * **Shows:** Whether nutrients are actually reaching the plant
   * **Important:** Leaf levels tell the real story of plant nutrition
3. **Magnesium (Mg) & Calcium (Ca)**
   * **Same critical roles as in soil**
   * **Leaf testing:** Confirms soil nutrients are being absorbed
4. **Boron (B)**
   * **What it does:** Essential for flower production and fruit set
   * **Critical for:** Preventing “chicken neck” in bunches
   * **Deficiency signs:** Deformed bunches, reduced yields
   * **Small amounts needed:** But critical when missing
5. **Copper (Cu) & Zinc (Zn)**
   * **What they do:** Help with enzyme functions and disease resistance
   * **Critical for:** Healthy metabolism and stress tolerance
   * **Deficiency signs:** Stunted growth, increased disease susceptibility

### Key Takeaways:

* **Focus on Critical gaps first** - these hurt production most
* **Balance is important** - too much of one nutrient can cause problems
* **Regular testing pays off** - catch problems before they reduce yields
* **Work with suppliers** - they can recommend specific fertilizers for your gaps

## 12. Fixing Common Problems

Even experienced users run into issues. Here’s how to solve the most common ones:

### App Won’t Start:

**Symptoms:** Error messages when running streamlit run app.py

**Solutions to try:** 1. **Check environment:** Make sure you see (.venv) in the command prompt 2. **Reactivate environment:** Type .\.venv\Scripts\activate again 3. **Update pip:** pip install -U pip 4. **Reinstall requirements:** pip install -r requirements.txt 5. **Check Python version:** python --version should show 3.10 or higher

### Upload Fails:

**Symptoms:** Error when trying to upload files

**Solutions:** 1. **Check file size:** Look in CONFIGURATION.md for limits 2. **Try sample files:** Use files from json/ folder to test 3. **Check format:** Save as CSV or Excel if having issues 4. **Browser cache:** Clear cache and try again

### Missing Data in Tables:

**Symptoms:** Some nutrients show “N/A” or are blank

**Normal cause:** Your lab didn’t test for all nutrients **Solution:** This is expected - the app analyzes what’s available

### PDF Looks Wrong:

**Symptoms:** Colors missing or layout broken

**Solutions:** 1. **Refresh page:** Reload the results page 2. **Regenerate PDF:** Try export again 3. **Check browser:** Try a different browser like Chrome

### Slow Performance:

**Symptoms:** App takes long to load or analyze

**Solutions:** 1. **Close other programs:** Free up computer memory 2. **Restart app:** Stop and restart with streamlit run app.py 3. **Check internet:** Some features need internet connection

### General Troubleshooting Steps:

1. **Restart everything:** Close PowerShell, reopen, reactivate environment, restart app
2. **Check CHANGELOG.md:** See if there are known issues or updates
3. **Test with samples:** Always verify with the example files first
4. **Update app:** Download latest version if problems persist

## 13. Common Questions Answered

### Do I need internet to use Ags-AI?

* **Local installation:** No internet required for basic use
* **Online features:** Internet needed for Streamlit Cloud or advanced features like AI document reading
* **Data privacy:** Your files stay on your computer unless you choose online sharing

### Can I add new nutrients to track?

* **Yes, but with limits:** The app can handle additional nutrients you upload
* **MPOB comparisons:** Work best for standard nutrients (most common ones are already included)
* **Custom standards:** You can add your own reference values if needed

### Is my data secure and private?

* **Local use:** Completely private - data never leaves your computer
* **Online sharing:** Use password protection and avoid uploading sensitive files
* **Firebase option:** Encrypted storage if you enable authentication
* **Best practice:** Don’t upload real farm data to public online versions

### How often should I test my soil and leaves?

* **Soil:** Every 2-3 years for established plantations, annually for new plantings
* **Leaves:** Every 6 months during active growth, annually minimum
* **After changes:** Test after major fertilizer applications or environmental changes

### Can I customize the severity levels?

* **Standard levels:** 5% (Balanced), 5-15% (Low), >15% (Critical) work well for most users
* **Customizable:** Advanced users can modify thresholds in the code
* **MPOB based:** Current levels are recommended by palm oil experts

### What if my lab uses different units?

* **Unit conversion:** The app handles most standard units automatically
* **Check results:** Always verify the first few analyses against manual calculations
* **Contact lab:** Ask for standard units if possible (mg/kg, %, meq/100g)

## 14. Sharing the App Setup With Others

When handing over Ags-AI to team members or other locations:

### Complete Package Checklist:

* ☐ Main application files (full folder)
* ☐ This user guide (docs/UserGuide.md)
* ☐ Example data files (json/ folder with samples)
* ☐ Blank settings template (.streamlit/secrets.example.toml)
* ☐ Version information (check CHANGELOG.md)
* ☐ Installation instructions (this guide)

### Packaging Methods:

1. **Zip file:** Use the included script or git archive --format zip --output Ags-AI-v1.0.2.zip HEAD
2. **USB drive:** Copy entire folder for offline transfer
3. **Network share:** Place on company server for team access

### What to Include in Handover:

* Installation guide (Section 4)
* Basic usage instructions (Sections 5-7)
* Troubleshooting tips (Section 11)
* Contact information for support

### Security Considerations:

* Remove any real API keys or passwords from shared files
* Use example secrets file, not the working one
* Document any custom configurations separately

## 15. Updating the App to New Versions

Keep your Ags-AI current for the latest features and bug fixes.

### How to Update:

1. **Download new version:** Get latest files from GitHub or provided source
2. **Backup your settings:** Save your .streamlit/secrets.toml file
3. **Replace files:** Copy new version over old (keep your settings file)
4. **Reactivate environment:** .\.venv\Scripts\activate
5. **Update requirements:** pip install -r requirements.txt
6. **Test run:** streamlit run app.py
7. **Check changelog:** Read CHANGELOG.md for new features

### What Updates Might Include:

* New nutrients or analysis features
* Better file format support
* Security improvements
* Bug fixes
* Performance improvements

### When to Update:

* When upload features stop working
* After major lab changes
* When new nutrients become important
* Every 3-6 months for regular maintenance

## 16. Quick Reference: Key Terms Explained

### Essential Terms:

* **MPOB:** Malaysian Palm Oil Board - the expert organization that sets nutrient standards for palm oil plantations
* **Severity Levels:** How serious a nutrient gap is:
  + Balanced (≤5% difference): Good, no action needed
  + Low (5-15% difference): Monitor, might need attention soon
  + Critical (>15% difference): Urgent, fix immediately
* **Absolute Percent Gap:** The size of the difference between your levels and ideal levels, shown as a percentage
* **Cation Exchange Capacity (CEC):** How well your soil holds nutrients (like a sponge holding water)

### Technical Terms Made Simple:

* **Virtual Environment:** A separate workspace for the app (like a clean kitchen)
* **Dependencies:** Helper programs the app needs (like ingredients for a recipe)
* **API Keys:** Passwords for online services (keep these secret!)
* **Repository:** A storage location for code files (like a digital filing cabinet)

## 17. Where to Find Things in the App

### Key Files and Folders:

* **app.py:** The main file that starts the application
* **modules/results.py:** Code that creates the analysis tables
* **utils/pdf\_utils.py:** Code that generates PDF reports
* **utils/analysis\_engine.py:** The brain that does the nutrient calculations
* **.streamlit/secrets.toml:** Your settings and configuration file
* **json/ folder:** Sample data files for testing
* **requirements.txt:** List of helper programs needed

### Documentation Files:

* **README.md:** Basic project information
* **DEPLOYMENT.md:** Advanced installation options
* **CONFIGURATION.md:** Technical settings guide
* **CHANGELOG.md:** List of changes in each version
* **API.md:** Technical details for developers

## Quick Command Reference

### First-Time Setup:

python -m venv .venv  
.\.venv\Scripts\activate  
pip install -r requirements.txt  
streamlit run app.py

### Docker Deployment (Advanced):

docker build -t ags-ai:latest .  
docker run --rm -p 8501:8501 -v ${PWD}\.streamlit:/app/.streamlit ags-ai:latest

### Creating Distribution Package:

git archive --format zip --output Ags-AI-v1.0.2.zip HEAD