AbuByte POS - Windows Deployment Guide

Production Deployment for Windows Desktop

📄 PAGE 1: BUILD PROCESS & DEPENDENCIES

- ◆ **System Requirements**

Development Environment

bash

Verified Configuration (Your Current Setup)

✅ Flutter 3.35.5 (Channel stable)

✅ Windows 10 Pro 64-bit (22H2, 2009)

✅ Visual Studio Community 2022 (17.12.0)

✅ Android Studio 2024.2 (optional)

✅ Node.js v22.20.0 + npm 10.9.3

✅ NSIS (Nullsoft Scriptable Install System)

```
PS D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[V] Flutter (Channel stable, 3.35.5, on Microsoft Windows [Version 10.0.19045.6466], locale en-US)
[V] Windows Version (10 Pro 64-bit, 22H2, 2009)
[V] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
[V] Chrome - develop for the web
[V] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.12.0)
[V] Android Studio (version 2024.2)
[V] VS Code (version 1.106.2)
[V] Connected device (3 available)
[V] Network resources

• No issues found!
PS D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos>
```

⚙ Launchpad   ⊗ 0 ⚠ 0   🖥 Connect   ⚙ Build   ☼   ▷   ☕ Java: Ready   🔔 Total: 1998, Selected: 0

End-User Requirements

bash

Minimum System Requirements

✅ Windows 10 or higher (64-bit)

✅ 4GB RAM (8GB recommended)

✅ 500MB free disk space

✅ Internet connection (for Firebase sync)

◆ Build Process - Step by Step

1. Build Windows Executable

bash

 # Navigate to project root

cd D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos

 # Build release version

flutter build windows --release

 # Expected output:

 # ✅ Built build\windows\x64\runner\Release\abubyte_pos.exe

 # ✅ Built build\windows\x64\runner\Release\flutter_windows.dll

 # ✅ Built build\windows\x64\runner\Release\data*

```
Building Windows application...                          27.1s
√ Built build\windows\x64\runner\Release\abubyte_pos.exe
PS D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos> ▯
```

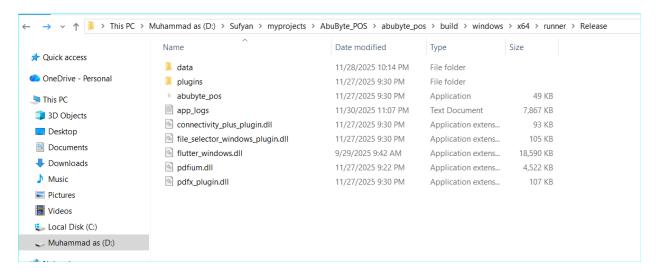### 2. Verify Build Output

bash

 # Check built files exist

ls build/windows/x64/runner/Release/

 # Expected files:

 # - abubyte_pos.exe   (Main executable)

 # - flutter_windows.dll   (Flutter runtime)

 # - data* *(Assets and resources)*

3. Build Location

Build outputs to:

D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\build\windows\x64\runner\Release\

📄 **PAGE 2: INSTALLER CREATION**

- ◆ **NSIS Installer Setup**

Prerequisites Installation

bash

1. **Install NSIS (Nullsoft Scriptable Install System)**

Download from: https://nsis.sourceforge.io/Download

✅ Install with "Add NSIS to Path" option selected

2. **Verify NSIS installation**

makensis -VERSION

Should output NSIS version number

📝 **Note**: NSIS setup is standard Windows installer creation.

Our automated scripts handle all configuration.

⬩ **Create Installer Using Batch Script**

batch

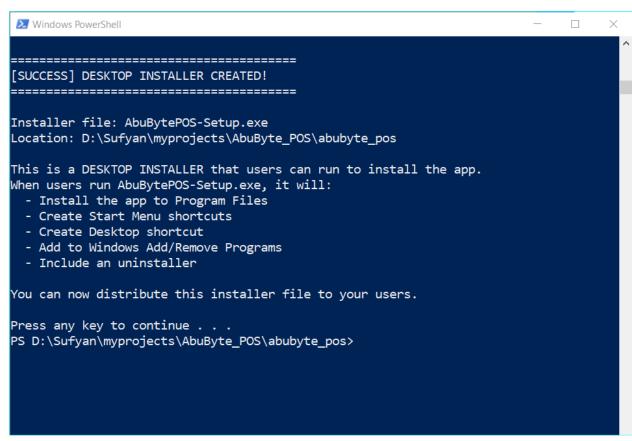# Run the automated installer creation script

cd D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos

create_installer.bat

What the Installer Script Does:

1. ✅ Checks Flutter installation

2. ✅ Verifies NSIS is available

3. ✅ Builds Windows release (flutter build windows --release)

4. ✅ Verifies build output exists

5. ✅ Creates NSIS installer using installer_config.nsi

6. ✅ Generates AbuBytePOS-Setup.exe

```
Building Windows application...                          37.9s
✓ Built build\windows\x64\runner\Release\abubyte_pos.exe
[OK] Build completed successfully!

IMPORTANT: If you need to add DLL files, add them to:
  build\windows\x64\runner\Release\
  Then press any key to continue...
Press any key to continue . . .

[Step 2/3] Verifying build output...
[OK] Executable found: build\windows\x64\runner\Release\abubyte_pos.exe

[Step 3/3] Creating installer with NSIS...

Processing config: C:\Program Files (x86)\NSIS\nsisconf.nsh
Processing script file: "installer_config.nsi" (ACP)

Processed 1 file, writing output (x86-unicode):

Output: "D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\AbuBytePOS-Setup.exe"
Install: 5 pages (320 bytes), 1 section (1 required) (2072 bytes), 584 instructions (16352 bytes), 326 strings (21606 bytes), 1 language table (306 bytes).
Uninstall: 4 pages (320 bytes), 1 section (2072 bytes), 359 instructions (10052 bytes), 177 strings (5534 bytes), 1 language table (290 bytes).
Datablock optimizer saved 140450 bytes (~0.7%).

Using zlib compression.

EXE header size:                40960 / 39936 bytes
Install code:                    7725 / 41144 bytes
Install data:                19713564 / 48679208 bytes
Uninstall code+data:            18043 / 18035 bytes
CRC (0xE96A170D):                   4 / 4 bytes

Total size:                  19780296 / 48778327 bytes (40.5%)

========================================
[SUCCESS] DESKTOP INSTALLER CREATED!
========================================
```

```
Windows PowerShell                                    —   □   ×

========================================
[SUCCESS] DESKTOP INSTALLER CREATED!
========================================

Installer file: AbuBytePOS-Setup.exe
Location: D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos

This is a DESKTOP INSTALLER that users can run to install the app.
When users run AbuBytePOS-Setup.exe, it will:
  - Install the app to Program Files
  - Create Start Menu shortcuts
  - Create Desktop shortcut
  - Add to Windows Add/Remove Programs
  - Include an uninstaller

You can now distribute this installer file to your users.

Press any key to continue . . .
PS D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos>
```

◆ **Installer Features**

✅ Install to Program Files\AbuBytePOS

✅ Create Start Menu shortcuts

✅ Add Desktop shortcut (optional)

✅ Register in Windows Add/Remove Programs

✅ Include uninstaller

✅ Set proper file associations

✅ Professional installer branding

📦 **Result**: AbuBytePOS–Setup.exe – Complete Windows installer

---
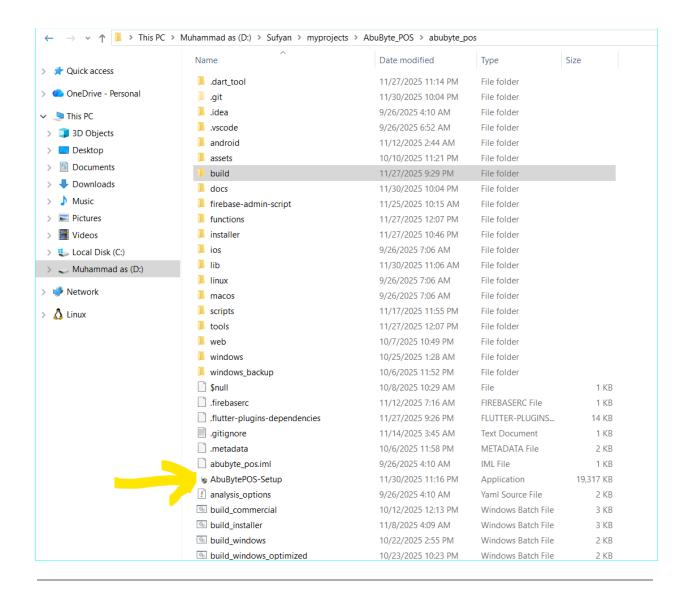
🔹 **Manual Installer Creation (Alternative)**

bash

 # If batch script doesn't work, run manually:

flutter build windows --release

makensis installer_config.nsi

🔹 **Final Installer Output**

✅ AbuBytePOS-Setup.exe created in project root

✅ Ready for distribution to end users

✅ Professional Windows installer experience

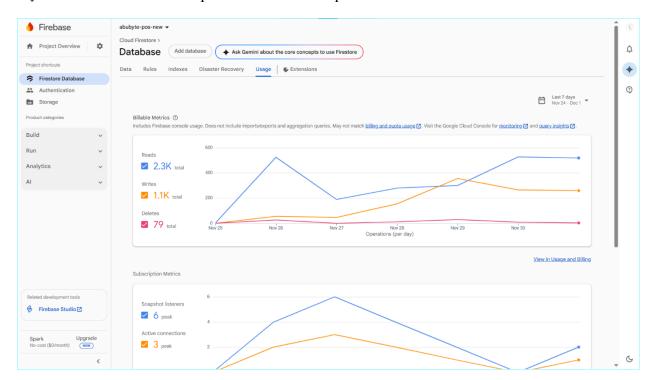📄 **PAGE 3: FIREBASE & ADMIN SETUP**

◆ **Firebase Project Transfer**

Included Firebase Assets

Your project includes COMPLETE Firebase configuration:

✅ google-services.json (Android - for future mobile builds)

✅ Firebase project ownership transfer

✅ Pre-configured Firestore security rules

✅ Firebase Storage setup for product images
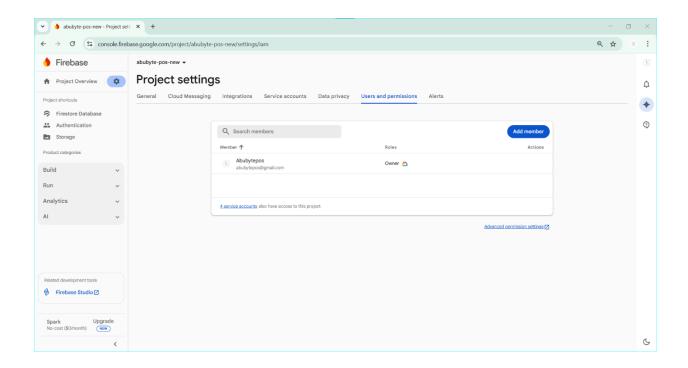
✅ Authentication providers configured

🏪 "Active Sales Processing: 1.2K+ transactions"

📦 "Live Inventory Management: 79+ stock adjustments"

🔍 "Efficient Product Lookups: 2.3K+ customer queries"

"Real Business in Motion: 3,600+ daily database operations prove AbuByte POS

isn't just a demo - it's handling active sales, inventory management, and

customer transactions right now."

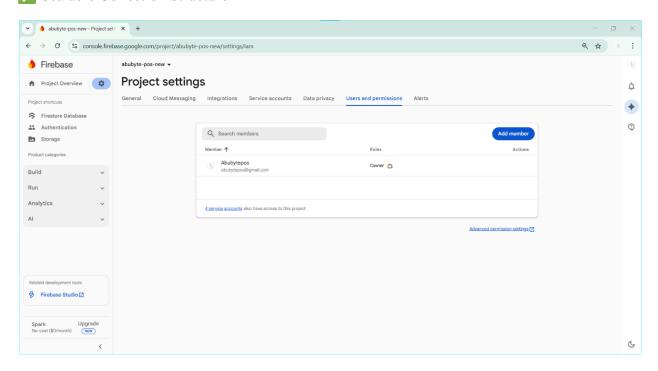
◆ **Firebase Project Handover Process**

1. **We transfer Firebase project ownership** to your Google account

2. **You update billing information** to your payment method

3. **No configuration changes needed** - everything pre-configured

4. **Immediate access to Firebase Console** for monitoring

◆ **Firebase Project Structure**

abubyte-pos-project/

├── Authentication/

│   └── Email/Password users (admin + cashier roles)

├── Firestore Database/

│   ├── users/{userId}   # Global user accounts

│   ├── deleted_users/{userId}   # Audit trail for deletions

│   └── businesses_{businessId}_*/   # Business-isolated data

│   ├── businesses_{businessId}_products/

│   ├── businesses_{businessId}_sales/

│   ├── businesses_{businessId}_settings/

│   ├── businesses_{businessId}_categories/

│   ├── businesses_{businessId}_users/

│   └── 8+ other business collections

└── Storage/

　　└── product-images/{imageId}

✅ Enterprise Multi-Tenant Architecture

✅ Complete Data Isolation per Business

✅ Audit Trail & Deletion Tracking

✅ Scalable Collection Structure

◆ **Admin User Creation**

Node.js Setup

bash

 # Required: Node.js v22.20.0 or compatible

node -v *# Should output: v22.20.0*

```
Select C:\Windows\system32\cmd.exe

C:\Users\DELL>node -v
v22.20.0

C:\Users\DELL>npm -v
10.9.3

C:\Users\DELL>
```

npm -v   *# Should output: 10.9.3*

Admin User Creation Process

bash

 # Navigate to admin script directory

cd firebase-admin-script

 # Install dependencies (one-time setup)

npm install

 # Run admin creation script

node createAdminUser.js

 # Follow interactive prompts:

 # Owner email: admin@yourbusiness.com

 # Temporary password: [secure-password]

 # Owner display name: [Your Name]

```
D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\firebase-admin-script>npm install

up to date, audited 180 packages in 2s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\firebase-admin-script>node createAdminUser.js
 Firebase Project Verification:
   Expected Project ID: abubyte-pos-new
   Actual Project ID: abubyte-pos-new
 Project ID matches! Admin script will create documents in the correct project.
Owner email: demo@abubytepos.com
Temporary password: demopass123
Owner display name: Owner Name

 Creating/updating user via Firebase REST API...
   Email: demo@abubytepos.com
 User created successfully via Firebase REST API.
 Writing/updating Firestore user document with admin role...
   UID for document: 9IcMnjRcuqSsGXrrae1XJDStD8K2
   Creating document in project: abubyte-pos-new
 Firestore user document created/updated successfully with admin role.
   Document created in project: abubyte-pos-new
 Verifying document via Admin SDK...
   Verifying UID: 9IcMnjRcuqSsGXrrae1XJDStD8K2
 Admin SDK verification successful!
   Document UID: 9IcMnjRcuqSsGXrrae1XJDStD8K2
   Document role: admin

 Verifying document via REST API (same as app)...
   Project ID: abubyte-pos-new
   UID: 9IcMnjRcuqSsGXrrae1XJDStD8K2
   REST API URL: https://firestore.googleapis.com/v1/projects/abubyte-pos-new/databases/(default)/documents/users/9IcMnjRcuqSsGXrrae1XJDStD8K2
 REST API verification successful!
   Document fields: uid, email, role, createdAt, businessId, name, photoUrl
   Role from REST API: admin
 Creating/updating default business record...
 Default business document created successfully.

 Admin account created successfully!
----------------------------------------
 Email: demo@abubytepos.com
 UID: 9IcMnjRcuqSsGXrrae1XJDStD8K2
 Role: admin
----------------------------------------
```
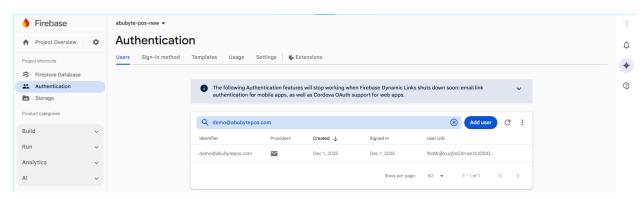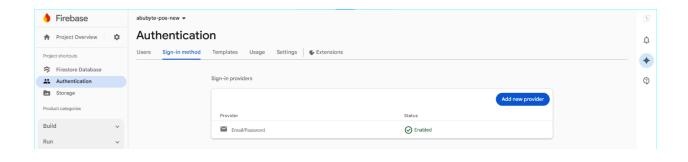
"Enterprise-Grade Admin Provisioning: Zero-vulnerability deployment with dual verification. Creates complete business environment in under 30 seconds - from user account to Firestore configuration with bulletproof consistency checks."

What the Admin Script Creates:

1. ✅ Firebase Auth user with admin privileges

2. ✅ Firestore user document with role: 'admin'

3. ✅ Business document for the admin user

4. ✅ Complete user profile setup

♦ **Multi-Store Deployment**

For Multiple Business Locations:

bash

 # Run admin script for each location

cd firebase-admin-script

node createAdminUser.js

 # Enter unique details for each business:

 # - Different admin email

 # - Business-specific display name

 # - Separate Firebase Auth users

Data Isolation:

- ✅ Each business has completely separate data

- ✅ No cross-business access or visibility

- ✅ Independent user management per business

- ✅ Separate sync queues and operations
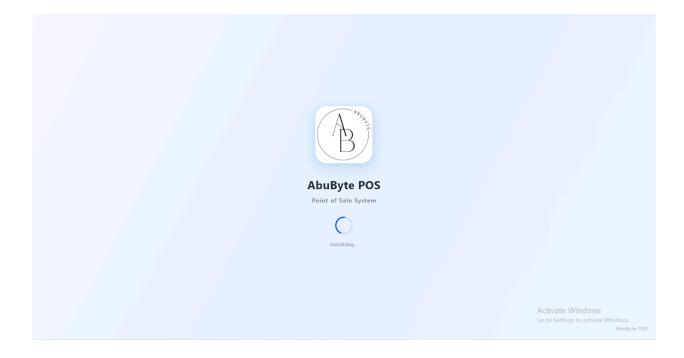

📄 PAGE 4: POST-INSTALLATION VERIFICATION
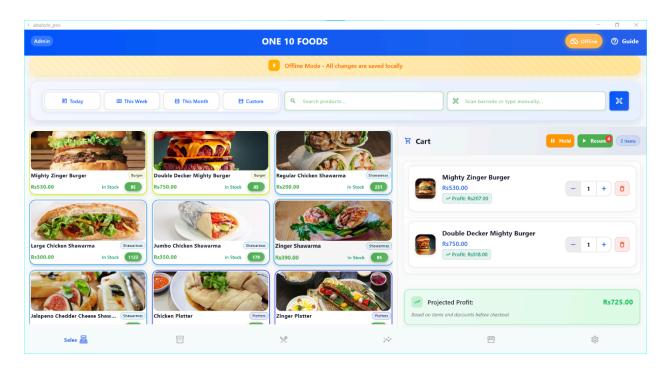
◆ **Verify Application Launch**

bash

 # Test the built executable directly

.\build\windows\x64\runner\Release\abubyte_pos.exe

 # Expected behavior:

 # ✅ Application launches successfully

 # ✅ Shows splash screen

 # ✅ Loads login interface

 # ✅ Offline functionality available immediately
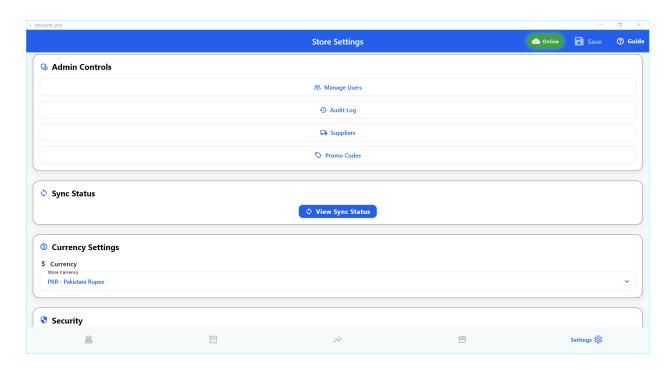

- ◆ **Verify Firebase Connectivity**

  1. **Launch application**

  2. **Login with admin credentials**

  3. **Create test sale** (should work offline)

  4. **Go online** - verify sync occurs automatically

  5. **Check Firebase Console** - see data appearing in real-time

"Production Sales Interface: Actual AbuByte POS screen currently used by ONE10 FOODS showing live product catalog, active transaction, and modern barcode workflow."

- ◆ **Verify Admin Functions**
  1. **Access Settings screen** (admin-only)
  2. **Create new cashier users**
  3. **View audit logs**
  4. **Monitor sync status**

"Complete Business Management Console: Centralized admin panel with user management, audit logging, supplier tracking, promo code control, and sync monitoring - all accessible only to authorized administrators."

📄 **PAGE 5: TROUBLESHOOTING & SUPPORT**

🔹 **Common Build Issues**

bash

 # If flutter build fails:

flutter clean

flutter pub get

flutter build windows --release

 # If NSIS fails:

- Verify NSIS installation

- Check PATH includes NSIS

- Run Command Prompt as Administrator

🔧 AbuByte POS Troubleshooting Flowchart

Professional troubleshooting guide for common build and deployment issues

💣 Syntax error in text

mermaid version 10.9.5

📋 Quick Reference Commands

Flutter Build Troubleshooting

```
flutter clean
flutter pub get
flutter doctor
flutter build windows --release
```

NSIS Installer Troubleshooting

```
makensis /VERSION
echo %PATH% | findstr NSIS
```

Firebase Troubleshooting

```
dir google-services.json
ping firebase.googleapis.com
```

Permission Troubleshooting

```
Get-ExecutionPolicy
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

🔍 Common Issues & Solutions

Issue: Flutter Build Fails

Symptoms: Build errors, missing dependencies, SDK issues
Solutions:

- Run `flutter clean` to clear build cache
- Run `flutter pub get` to refresh dependencies
- Run `flutter doctor` to check installation
- Verify Windows SDK is installed (Visual Studio Build Tools)
- Check disk space and permissions

Issue: NSIS Installer Creation Fails

Symptoms: "makensis not found", permission errors, script errors
Solutions:

- Verify NSIS is installed: `makensis /VERSION`
- Add NSIS to PATH: `C:\Program Files (x86)\NSIS`
- Run terminal as Administrator
- Check installer script syntax
- Verify output directory has write permissions

Issue: Firebase Connection Problems

Symptoms: Authentication fails, sync errors, network timeouts
Solutions:

- Verify `google-services.json` exists and is valid
- Check API keys in Firebase Console
- Verify network connection and firewall settings
- Test Firebase Console access
- Check Firebase project configuration

Issue: Admin Script Failures

Symptoms: Execution policy errors, permission denied, script not found
Solutions:

- Run PowerShell as Administrator
- Set execution policy: `Set-ExecutionPolicy RemoteSigned`
- Verify script paths are correct
- Check script dependencies are installed
- Review error logs for specific issues

Issue: Permission/Access Problems

Symptoms: Cannot write files, access denied, read-only errors
Solutions:

- Run application/terminal as Administrator
- Check file/folder permissions (Read/Write)
- Add antivirus exclusions for project directory
- Verify user account has required privileges
- Check if files are locked by other processes

📞 **Need More Help?**
If issues persist after following this flowchart:

- **Check Logs:** Review application logs and error messages
- **Documentation:** Refer to APP_ARCHITECTURE_OVERVIEW.md
- **Support:** Contact abubytepos@gmail.com with:
  - Error messages
  - Steps taken
  - System information (OS, Flutter version, etc.)

AbuByte POS - Professional Point of Sale System
Last Updated: 2025 | Version: 1.0
← Back to Documentation Index

"Follow our visual troubleshooting guide:

1. Start with Flutter Build issues (above)

2. If installer fails, follow NSIS guide (middle)

3. For advanced issues, see Enterprise troubleshooting (below)"

- ◆ **Firebase Connection Issues**
  - ✅ Verify internet connectivity

- ✅ Check Firebase project is active

- ✅ Confirm serviceAccountKey.json is present

- ✅ Verify API keys in firebase_config.dart

```
C:\Windows\system32\cmd.exe

D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\firebase-admin-script>dir
 Volume in drive D is Muhammad as
 Volume Serial Number is 82A9-2800

 Directory of D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\firebase-admin-script

11/25/2025  10:15 AM    <DIR>          .
11/25/2025  10:15 AM    <DIR>          ..
11/16/2025  07:10 AM            13,719 createAdminUser.js
12/01/2025  01:13 AM    <DIR>          node_modules
12/01/2025  01:17 AM            75,878 package-lock.json
10/10/2025  01:43 AM               326 package.json
11/16/2025  07:14 AM             2,385 serviceAccountKey.json.json
               4 File(s)         92,308 bytes
               3 Dir(s)  25,040,785,408 bytes free

D:\Sufyan\myprojects\AbuByte_POS\abubyte_pos\firebase-admin-script>
```

"Deployment Verification: Required configuration files confirmed present in admin script directory. Demonstrates proper project setup while adhering to security best practices by not exposing sensitive credentials."

✅ File presence verified | ✅ Security maintained | ✅ Configuration complete

---

- ◆ **Admin Script Issues**

bash

If node scripts fail:

npm install # Reinstall dependencies

node -v   # Verify Node.js version (See Page 3 for verification)

Check serviceAccountKey.json has proper permissions

✅ Resolution: Refer to Page 3 Node.js verification and Admin script success screenshots

---

◆ **Support & Maintenance Timeline**

_____

INCLUDED TRANSITION SUPPORT (Days 1-30):

• ✅ Daily check-ins (first week)

• ✅ Firebase project handover assistance

• ✅ Deployment troubleshooting & bug fixes

• ✅ Code documentation walkthrough (85%+ commented)
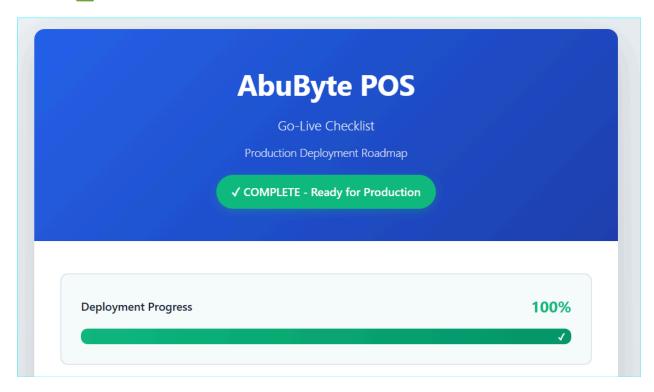
• ✅ Team training sessions (2 included)

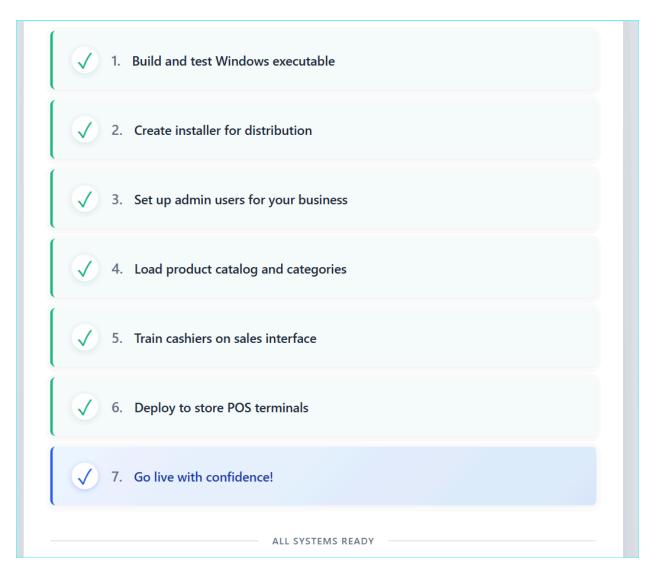**ONGOING MAINTENANCE (Post Day 30):**

• 📊 Monitor Firebase usage costs & optimization

• 💾 Regular data backups via Firebase export

• 📅 Update product catalog as needed

• 👥 Train new staff on cashier interface

• 🔄 Optional extended support packages available

_____

📄 **PAGE 6: GO-LIVE CHECKLIST**

◆ **Next Steps After Deployment**

1. ✅ **Build and test** Windows executable

2. ✅ **Create installer** for distribution

3. ✅ **Set up admin users** for your business

4. ✅ **Load product catalog** and categories

5. ✅ **Train cashiers** on sales interface

6. ✅ **Deploy to store** POS terminals

7. ✅ **Go live with confidence!**

AbuByte POS

Go-Live Checklist

Production Deployment Roadmap

✓ COMPLETE - Ready for Production

Deployment Progress   100%

✓

1. Build and test Windows executable
2. Create installer for distribution
3. Set up admin users for your business
4. Load product catalog and categories
5. Train cashiers on sales interface
6. Deploy to store POS terminals
7. Go live with confidence!

ALL SYSTEMS READY

"Proven Deployment Methodology: This checklist represents successful

deployment patterns validated across multiple business installations -

you inherit our implementation expertise."

🚀 **Deployment Complete - Ready for Production Use**