# 🎓 Complete Beginner's Setup Guide

**For people with NO prior experience in Docker, Python, or networking**

This guide assumes you know nothing about technical setup. We'll walk through every single step with clear explanations.

---

## 🚀 FASTEST PATH (New!)

**Complete LOCAL setup in just 4 steps:**

1. **Install tools** (Part 1 below) - 15 minutes
2. **Download platform** (Part 2) - 2 minutes
3. **Run automated setup:** `automated_setup.bat` - 10 minutes
4. **Create Flowise API key** when prompted - 1 minute

**Total: ~30 minutes with almost no manual work!**

The automated setup creates admin, teachers, and students automatically with credits already allocated!

⚠ **This sets up the platform on YOUR computer (localhost).** For school-wide access, see **"🏫 School Network Deployment"** section after setup.

**Manual setup** is also available if you want to learn each step.

---

## 📖 What is this project?

This is a ChatProxy Platform that allows students and teachers to use AI chatbots. Think of it like installing a mini-server on your computer that runs several small programs (called "services") that work together.

⚠ **IMPORTANT - Two Setup Phases:**

1. **Phase 1: Local Setup (This Guide)** - Install on YOUR computer first
   - Works on your computer only (localhost)
   - Perfect for testing and learning
   - Follow the steps below
2. **Phase 2: School Network Deployment** - Make it available to everyone
   - Requires IT department involvement
   - Needs domain name, SSL certificate, and network configuration
   - See **"🏫 School Network Deployment"** section after setup

**Recommendation:** Complete Phase 1 first, test everything works, THEN work with IT on Phase 2.

---

## ⏱ Time Required

**Phase 1 - Local Setup (This Computer Only):**

- Setup time: About 30-45 minutes (or 15 minutes with automated setup)
- What you'll need:
  - A Windows computer
  - Internet connection
  - Administrator access to your computer

**Phase 2 - School Network Deployment (Optional):**

- Coordination time: 1-3 days (waiting for IT department)
- What you'll need:
  - IT department support
  - Domain name or static IP
  - SSL certificate
  - Network/firewall configuration
- See **"🏫 School Network Deployment"** section below

---

## 🎯 TWO SETUP OPTIONS

### Option A: Automated Setup (Recommended! ⭐)

**After installing the basic tools (Docker, Python, Git) in Part 1, you can run:**

```
automated_setup.bat
```

**This does almost EVERYTHING automatically:**

- ✅ Checks your system
- ✅ Configures storage on the best drive
- ✅ Starts all services
- ✅ Prompts you to create a Flowise API key (only manual step!)
- ✅ Creates admin user + teachers + students automatically
- ✅ Allocates credits automatically
- ✅ Verifies everything works

**Time: 10-15 minutes** instead of 45 minutes!
**You only need to:** Follow Part 1 (install tools), then run the automated script!

### Option B: Manual Setup (Step-by-Step)

Follow all the steps in this guide - good for learning how everything works.

**Choose Option A if:** You just want it to work quickly
**Choose Option B if:** You want to understand each step

---

## 🛠️ Part 1: Installing the Basic Tools

Before we can run the ChatProxy Platform, we need to install 3 programs on your computer.

## Step 1: Install Docker Desktop

**What is Docker?**
Docker is like a "container" system that packages programs with everything they need to run. Think of it as a virtual box that contains a complete mini-computer inside your computer.

**How to install:**

1. **Download Docker Desktop:**
    - Go to: https://www.docker.com/products/docker-desktop
    - Click the big blue "Download for Windows" button
    - Wait for the download to finish (file is about 500MB)
2. **Install Docker:**
    - Find the downloaded file (usually in your Downloads folder)
    - File name: `Docker Desktop Installer.exe`
    - Double-click to run it
    - Click "Yes" when Windows asks "Do you want to allow this app to make changes?"
    - Click "OK" to start installation
    - **Wait 5-10 minutes** - this is normal!
    - Click "Close and restart" when done
3. **Start Docker:**
    - After your computer restarts, Docker Desktop should open automatically
    - If not, look for "Docker Desktop" icon on your desktop or Start Menu
    - **First time opening Docker takes 2-3 minutes** - be patient!
    - You'll see a whale icon 🐳 in your taskbar (bottom-right corner) when it's ready
    - The whale icon should be steady (not animating) when Docker is ready
4. **Accept the agreement:**
    - Docker will show you a service agreement
    - Check the box "I accept the terms"
    - Click "Accept"
5. **Skip the survey** (optional):
    - Docker might ask you questions about how you'll use it
    - You can click "Skip" - it's not required

**☑ How to know Docker is working:**

- Look at the bottom-right of your screen (taskbar)
- You should see a whale icon 🐳
- When you hover over it, it should say "Docker Desktop is running"

---

## Step 2: Install Python

**What is Python?**
Python is a programming language. Our setup scripts are written in Python, so we need it installed to run those scripts.

**How to install:**

1. **Download Python:**
    - Go to: https://www.python.org/downloads/
    - Click the big yellow button "Download Python 3.x.x"
    - Wait for download (file is about 25MB)
2. **Install Python:**
    - Find the downloaded file: `python-3.x.x-amd64.exe`
    - Double-click to run it
    - ⚠ **IMPORTANT:** Check the box "Add Python to PATH" at the bottom
    - Click "Install Now"
    - Click "Yes" when Windows asks for permission
    - Wait 2-3 minutes
    - Click "Close" when done

☑ **How to know Python is working:**

- Press `Windows key + R` on your keyboard
- Type: `cmd` and press Enter
- A black window will open (this is called "Command Prompt")
- Type: `python --version` and press Enter
- You should see something like: `Python 3.12.1`
- Type `exit` and press Enter to close the window

---

## Step 3: Install Git

**What is Git?**
Git is a tool for downloading code from the internet. We need it to download the ChatProxy Platform files.

**How to install:**

1. **Download Git:**
    - Go to: https://git-scm.com/download/win
    - The download should start automatically
    - If not, click "Click here to download" link
    - Wait for download (file is about 50MB)
2. **Install Git:**
    - Find the downloaded file: `Git-2.x.x-64-bit.exe`
    - Double-click to run it

  - Click "Yes" when Windows asks for permission
  - Click "Next" for all screens (default settings are fine)
  - **Just keep clicking "Next"** - no need to change anything
  - Click "Install"
  - Wait 1-2 minutes
  - Click "Finish"

**☑ How to know Git is working:**

- Press `Windows key + R`
- Type: `cmd` and press Enter
- Type: `git --version` and press Enter
- You should see something like: `git version 2.43.0`
- Type `exit` and press Enter

---

## � Part 2: Download Platform & Choose Setup Method

Now you have two choices - automated or manual setup!

### Step 1: Download the Platform

1. **Open Command Prompt:**
   - Press `Windows key + R`
   - Type: `cmd` and press Enter
2. **Navigate to Documents:**

```
cd %USERPROFILE%\Documents
```

3. **Download the platform:**

```
git clone https://github.com/vocabbreaker/ChatProxyPlatform.git
```

Wait 1-2 minutes for download to complete.

4. **Enter the folder:**

```
cd ChatProxyPlatform
```

---

### Step 2A: AUTOMATED SETUP (RECOMMENDED ★)

**Simply run:**

```
automated_setup.bat
```

**What happens next:**

1. **Script checks your system** (Docker, Python, Git) - takes 10 seconds
2. **Asks about drive configuration** (just press Enter to use defaults)
3. **Starts Flowise** - wait 2-3 minutes
4. **IMPORTANT:** Script will open your browser with instructions:
    - Go to http://localhost:3002
    - Create your admin account
    - Go to Settings → API Keys → Create New Key
    - Copy the key
    - Paste it in the Command Prompt when asked
5. **Script does everything else automatically:**
    - Starts all services
    - Creates admin user
    - Creates 3 teachers (5,000 credits each)
    - Creates 5 students (1,000 credits each)
    - Verifies everything works

**Total time: 10-15 minutes!**

**Default users created by automated setup:**

- **Admin:** username=`admin`, password=`admin@admin`, credits=**10,000**
- **Teachers (3):** teacher1/teacher2/teacher3, password=`admin@admin`, credits=**5,000 each**
- **Students (5):** student1-5, password=`admin@admin`, credits=**1,000 each**

⚠ **All users have the same default password:** `admin@admin` (change after first login!)

When you see "Setup completed successfully!", skip to **Part 4: Testing Everything Works**

---

### Step 2B: MANUAL SETUP (For Learning)

If you want to understand each step, continue with Part 3 below.

---

## 📥 Part 3: Manual Setup (Skip if you used automated_setup.bat)

⚠ **NOTE: If you ran `automated_setup.bat` successfully, SKIP THIS ENTIRE SECTION and go directly to Part 4!**

**Only follow this section if you want to set up everything manually step-by-step.**

---

### Step 1: Create Configuration Files

**What we're doing:** Creating special files that tell each service how to work

Type:

```
setup_env_files.bat
```

**What you'll see:**

- Several lines showing "Created .env file"
- Takes about 5 seconds
- You should see "✓" (checkmarks) for 5 services

**If you see errors:** That's okay! Skip to the troubleshooting section at the end.

---

### Step 2: Generate Secure Passwords

**What we're doing:** Creating random secure passwords for databases and security tokens

Type:

```
generate_secrets.bat
```

**What you'll see:**

- Long random strings of letters and numbers
- Each line shows a different password being generated
- Takes about 10 seconds
- Should see "✓ All 11 updates completed successfully!"

**What this does:** Creates super secure passwords automatically so you don't have to think of them yourself

---

### Step 3: Start the Flowise Service

**What is Flowise?** It's the AI chatbot builder - the main part of the platform

Type these commands one by one:

```
cd flowise
start-with-postgres.bat
```

**What you'll see:**

- A lot of text scrolling - this is normal!
- You might see messages like "Pulling image" or "Creating container"
- **This takes 2-5 minutes the first time** (downloading images from internet)
- Eventually it will say "Started" or show container names

**Wait 1 minute** for everything to fully start

## Step 4: Check if Flowise Started Successfully

Type:

```
docker ps
```

**What you'll see:** A table showing running containers. You should see at least 2 rows:

- One with "flowise" in the name
- One with "flowise-postgres" in the name

**Status column should say:** "Up" or "Up X seconds"

---

## Step 5: Create Your Admin Account

1. **Open your web browser** (Chrome, Edge, Firefox, etc.)

2. **Go to:** http://localhost:3002

   **What is localhost?** It's a special address that means "this computer" - you're accessing the server running on your own computer!

3. **Create admin account:**

   - You should see a signup page
   - **Fill in:**
     - Name: `admin` (or your name)
     - Email: `your-email@example.com` (use your real email)
     - Password: `Admin@2026` (or choose your own secure password)
   - Click "Sign Up"

4. **You should now see the Flowise dashboard** - colorful interface with options to create chatflows

---

## Step 6: Create an API Key

**What is an API key?** Think of it as a special password that lets different parts of the system talk to each other.

**Go back to Command Prompt and type:**

```
cd ..
configure_flowise_api.bat
```

**What happens:**

1. Your browser will open showing instructions

2. Command Prompt will ask you to create an API key

**Follow these steps in the browser:**

1. In Flowise (http://localhost:3002), click your profile icon (top-right corner)
2. Click "Settings"
3. Click "API Keys" tab
4. Click "Create New Key" button
5. Give it a name: `proxy-service`
6. Click "Create"
7. **Copy the long key** that appears (looks like: `A27MfYLThKBwYcpDb2M9s6rwXIwUAly9P8j_ujX_J9I`)

**Go back to Command Prompt:**

1. Right-click in the Command Prompt window
2. Choose "Paste"
3. Press Enter

**What you'll see:**

- "✓ API key validated successfully"
- "✓ Updated successfully"

---

## Step 7: Start the Other Services

**What are these services?**

- **Auth Service:** Handles user logins
- **Accounting Service:** Tracks credits (like tokens for using AI)
- **Proxy Service:** Connects everything together
- **Bridge:** The website interface students/teachers will use

Type these commands one by one (wait 30 seconds between each):

```
cd auth-service
start.bat
```

*Wait 30 seconds*

```
cd ..\accounting-service
start.bat
```

*Wait 30 seconds*

```
cd ..\flowise-proxy-service-py
start-docker.bat
```

*Wait 30 seconds*

```
cd ..\bridge
start.bat
```

**Each time you'll see:**

- Text scrolling
- "Creating network"
- "Creating container"
- "Started" or container name

---

## Step 8: Create Users

**Now let's create teacher and student accounts**

1. **Navigate to the user management folder:**

```
cd ..\auth-service\quickCreateAdminPy
```

2. **Right-click on `manage_users_admin.bat` in File Explorer:**
   - Open File Explorer
   - Navigate to: `Documents\ChatProxyPlatform\auth-service\quickCreateAdminPy`
   - Find `manage_users_admin.bat`
   - Right-click it
   - Choose "Run as administrator"
   - Click "Yes" when Windows asks for permission

3. **Follow the prompts in the window that opens**

**Or use the CSV method to create multiple users at once:**

```
notepad users.csv
```

This opens a spreadsheet-like file. Edit it following this format:

```
action,username,email,password,role,fullName,credits
create,teacher1,teacher1@school.com,Teacher1!,teacher,Teacher One,5000
create,student1,student1@school.com,Student1!,student,Student One,1000
```

Save and close, then run:

```
sync_all_users.bat
```

---

## � School Network Deployment (Important!)

**⚠ READ THIS if you're setting this up for a school (multiple users on different computers)**

### What You Have Now

Right now, the platform runs on **localhost** which means:

- ☑ It works perfectly on YOUR computer
- ✗ Other computers on the school network CANNOT access it
- ✗ It's not secure for internet access (no encryption)

### What You Need for School-Wide Access

To make this available to teachers and students on different computers, you need **3 things from your school's IT department:**

---

### 1. 📡 Static IP Address or Domain Name

**What is this?**
Instead of `localhost:3082`, you need a real address that other computers can find.

**In simple terms:**

- **Localhost** = "this computer" (only you can access)
- **Internal IP** = "computer #192.168.1.50" (others on school network can access)
- **Domain name** = "chatbot.yourschool.edu" (easy to remember, others can access)

**What to ask IT for:**

> "We need either:
>
> 1. A static IP address on the school network for this server, OR
> 2. A subdomain like `chatbot.ourschool.edu` that points to this server"

**Example addresses:**

- Internal: `http://192.168.1.50:3082` (works on school WiFi/LAN only)
- Domain: `https://chatbot.yourschool.edu` (better - works anywhere)

---

### 2. 🔒 SSL Certificate (HTTPS)

**What is this?**
SSL makes your website secure (the little lock icon 🔒 in the browser).

**Why you need it:**

- ☑ Encrypts passwords and data (so hackers can't see them)
- ☑ Required for modern browsers to work properly
- ☑ Shows students/teachers the site is safe
- ☑ Prevents "Not Secure" warnings

**In simple terms:**

- **HTTP** (no SSL) = Sending postcards (anyone can read them)
- **HTTPS** (with SSL) = Sending locked letters (only recipient can read)

**What to ask IT for:**

> "We need an SSL certificate for the chatbot platform. Can you provide either:
>
> 1. A certificate from the school's certificate authority, OR
> 2. A Let's Encrypt free SSL certificate?"

**What IT needs to know:**

- Domain name (e.g., `chatbot.yourschool.edu`)
- Server where it's installed (the computer running Docker)

---

### 3. 🌉 Reverse Proxy (For External Access)

**What is this?**
A reverse proxy is like a "reception desk" that forwards visitors to the right place.

**Why you need it:**

- ☑ Allows external (internet) access to internal (school network) server
- ☑ Handles SSL/HTTPS encryption
- ☑ Protects your server from direct internet exposure
- ☑ Can add additional security (firewall, authentication)

**In simple terms:** Think of your school network like an office building:

- **Without proxy:** Students must physically come to your computer room
- **With proxy:** Students can "call in" from home, and reception (proxy) transfers them to the right office

**What to ask IT for:**

> "We need a reverse proxy (like nginx or Apache) to make the chatbot platform accessible from outside the school network. It should:
>
> 1. Forward traffic from `https://chatbot.ourschool.edu` to `http://localhost:3082`
> 2. Handle SSL termination
> 3. Be accessible from both school network AND internet (if remote access needed)"

---

### Network Setup Diagram

```
[Student's Device]
      ↓
[Internet] ← (if remote access)
      ↓
[School Firewall]
      ↓
[Reverse Proxy Server] ← SSL Certificate installed here
  (handles HTTPS)
      ↓
[Your Docker Server] ← ChatProxy Platform running here
  (localhost:3082)
```

---

## Two Deployment Options

### Option A: School Network Only (Simpler)

**Best for:** Students/teachers only access from school WiFi/computers

**What you need:**

1. ☑ Static IP address: `192.168.1.50`
2. ☑ SSL certificate (optional but recommended)
3. ❌ No reverse proxy needed (internal only)

**Students access via:** `http://192.168.1.50:3082` or `https://chatbot-internal.school.local`

**Pros:**

- Simpler to set up
- More secure (not exposed to internet)
- Lower IT requirements

**Cons:**

- Only works on school network
- Can't access from home

---

### Option B: Internet Access (More Complex)

**Best for:** Students/teachers need access from home

**What you need:**

1. ☑ Domain name: `chatbot.yourschool.edu`
2. ☑ SSL certificate (REQUIRED)
3. ☑ Reverse proxy with port forwarding
4. ☑ Firewall rules

**Students access via:** `https://chatbot.yourschool.edu`

**Pros:**

- Works from anywhere
- Professional setup
- Better user experience

**Cons:**

- Requires more IT involvement
- Security considerations (need firewall, monitoring)
- May need approval from school administration

---

## What to Tell Your IT Department

**Copy and send this to your IT team:**

---

**Email Template for IT Department:**

```
Subject: Request for ChatProxy Platform Network Setup

Hello IT Team,

We are setting up a ChatProxy Platform (AI chatbot system) for educational use.
The platform is currently running on Docker on [Computer Name/Location].

To make it accessible to students and teachers, we need:

OPTION 1 - School Network Only:
□ Static IP address for the server
□ Internal DNS record (optional): chatbot.ourschool.local → [IP address]
□ SSL certificate (recommended but optional for internal use)

OPTION 2 - Internet Access Required:
□ Public domain or subdomain: chatbot.yourschool.edu
□ SSL certificate for the domain
□ Reverse proxy (nginx/Apache) configuration
□ Port forwarding: HTTPS (443) → Server:3082
□ Firewall rules to allow external access

Technical Details:
- Platform runs on Docker (Windows)
- Service ports: 3082 (Bridge UI), 3000 (Auth), 3001 (Accounting), 3002 (Flowise)
- Only port 3082 needs to be accessible to end users
- All other ports should remain internal/backend only

Security Requirements:
- HTTPS/SSL required for production use
- User authentication already built-in (JWT tokens)
- Database access is internal only (MongoDB, PostgreSQL)
```

```
Please let me know which option is feasible and what information you need from me.

Thank you!
```

---

## After IT Setup - Update Configuration

Once IT provides the domain/IP and SSL, you need to update these files:

**1. Bridge UI environment (`bridge/.env`):**

```
# Change from:
VITE_API_BASE_URL=http://localhost:3000

# To your domain:
VITE_API_BASE_URL=https://chatbot.yourschool.edu/api
```

**2. Auth Service environment (`auth-service/.env`):**

```
# Change from:
CORS_ORIGIN=http://localhost:3082

# To your domain:
CORS_ORIGIN=https://chatbot.yourschool.edu
```

**3. Accounting Service environment (`accounting-service/.env`):**

```
# Change from:
CORS_ORIGINS=http://localhost:3082,http://localhost:3000

# To your domain:
CORS_ORIGINS=https://chatbot.yourschool.edu
```

**4. Restart all services:**

```
cd bridge
stop.bat
start.bat

cd ..\auth-service
stop.bat
start.bat

cd ..\accounting-service
```

```
stop.bat
start.bat
```

---

### Testing Network Access

**After IT sets everything up:**

1. **Test from another computer on school network:**
   - Open browser
   - Go to: `https://chatbot.yourschool.edu` (or the address IT gave you)
   - You should see the login page
   - Try logging in with: `admin` / `admin@admin`
2. **Test from home** (if internet access enabled):
   - Use your phone (disconnect from school WiFi)
   - Go to the same address
   - Should work the same way
3. **Check HTTPS:**
   - Look for lock icon 🔒 in browser address bar
   - Click it - should say "Connection is secure"

---

### Security Checklist for Production

Before allowing students to use it:

- □ HTTPS/SSL is working (lock icon in browser)
- □ Change default admin password from `admin@admin`
- □ Change all default user passwords
- □ Regular backups configured (see backup section below)
- □ Only port 3082 is accessible from outside (other ports blocked)
- □ Firewall rules in place
- □ IT department has admin access for monitoring
- □ Data storage on RAID or backed-up drive

---

## � 🎉 Part 4: Testing Everything Works

### Check if Services are Running

In Command Prompt, type:

```
docker ps
```

**You should see about 8-10 containers running:**

- flowise
- flowise-postgres
- mongodb-auth
- mongodb-proxy
- postgres-accounting
- auth-service
- accounting-service
- flowise-proxy-service
- bridge
- mailhog (email testing)

**Status should say "Up" for all of them**

---

### Access the Platform

Open your web browser and try these addresses:

1. **Flowise (AI Builder):**
   - http://localhost:3002
   - You should see the Flowise dashboard
2. **Bridge (Student/Teacher Interface):**
   - http://localhost:3082
   - You should see a login page
3. **Try logging in to Bridge:**
   - Use the username/password you created in Step 8
   - You should see the main interface

---

## 🛑 Stopping Everything

When you're done and want to turn everything off:

**Option 1: Stop all at once**

In Command Prompt, from the ChatProxyPlatform folder, type:

```
docker stop $(docker ps -q)
```

**Option 2: Stop each service individually**

```
cd flowise
stop.bat

cd ..\auth-service
stop.bat

cd ..\accounting-service
stop.bat

cd ..\flowise-proxy-service-py
docker compose down

cd ..\bridge
stop.bat
```

## 🔄 Starting Everything Again Later

When you restart your computer or want to use the platform again:

1. **Make sure Docker Desktop is running** (look for whale icon 🐳 in taskbar)
2. **Open Command Prompt and navigate to the folder:**

```
cd %USERPROFILE%\Documents\ChatProxyPlatform
```

3. **Start services in order:**

```
cd flowise
start-with-postgres.bat
```

*Wait 30 seconds*

```
cd ..\auth-service
start.bat
```

*Wait 30 seconds*

```
cd ..\accounting-service
start.bat
```

*Wait 30 seconds*

```
cd ..\flowise-proxy-service-py
start-docker.bat
```

*Wait 30 seconds*

```
cd ..\bridge
start.bat
```

4. **Wait 1-2 minutes for everything to fully start**

5. **Open browser:** http://localhost:3082

---

## 🐛 Troubleshooting - When Things Don't Work

### "automated_setup.bat failed" or "Setup encountered errors"

**What this means:** The automated setup script hit a problem

**Solutions to try:**

1. **Check the error message carefully** - it usually tells you what's wrong:
   - "Docker is not running" → Start Docker Desktop
   - "Python is not installed" → Install Python (Part 1, Step 2)
   - "Git not found" → Install Git (Part 1, Step 3)

2. **Run the diagnostic:**

```
check_system.bat
```

This creates a report showing what's wrong

3. **Try manual setup instead:** Follow Part 3 of this guide step-by-step

4. **Check Docker Desktop:**
   - Open Docker Desktop application
   - Make sure the whale icon 🐳 is solid (not animated)
   - Try running `docker ps` in Command Prompt

---

### "Docker is not running"

**What you see:** Error message mentioning Docker

**Solution:**

1. Look at bottom-right of screen (taskbar)
2. Do you see a whale icon 🐳?
3. If not, find "Docker Desktop" in Start Menu and click it
4. Wait 2-3 minutes for Docker to start
5. Try the command again

---

### "Port 3002 is already in use"

**What this means:** Something else is using that port

**Solution:**

1. Restart your computer
2. Start Docker Desktop
3. Try the setup again

---

### "Cannot find the path specified"

**What this means:** You're in the wrong folder

**Solution:**

1. In Command Prompt, type:

```
cd %USERPROFILE%\Documents\ChatProxyPlatform
```

2. Try the command again

---

### "Python is not recognized"

**What this means:** Python didn't install correctly

**Solution:**

1. Uninstall Python (Settings → Apps → Python → Uninstall)
2. Download Python again from https://www.python.org
3. **IMPORTANT:** Check "Add Python to PATH" during installation
4. Install again

---

### Website not loading (localhost:3002 or localhost:3082)

**Solutions to try:**

1. **Wait longer:** Sometimes services take 2-3 minutes to fully start

2. **Check if containers are running:**

```
docker ps
```

Should show "Up" status for containers

3. **Check Docker Desktop:**
    - Open Docker Desktop application
    - Click "Containers" on the left
    - See if containers are running (green icon)

4. **Restart services:**

```
cd flowise
stop.bat
start-with-postgres.bat
```

---

### "Access Denied" or "Permission Error"

**Solution:**

1. Close Command Prompt
2. Right-click on "Command Prompt" in Start Menu
3. Choose "Run as administrator"
4. Click "Yes" when Windows asks
5. Navigate back to the folder and try again

---

## 📞 Getting Help

If you're still stuck after trying troubleshooting:

1. **Take a screenshot** of the error message

2. **Run the diagnostic script:**

```
cd %USERPROFILE%\Documents\ChatProxyPlatform
check_system.bat
```

This creates a report file that shows what's wrong

3. **Check the logs:**

```
cd flowise
docker logs flowise
```

4. **Share the error messages** with someone technical or the support team

---

## 📁 What Each Folder Does

Now that everything is set up, here's what each part does:

- **flowise/** - The AI chatbot builder (like the "brain")
- **auth-service/** - Handles login/logout (the "security guard")
- **accounting-service/** - Tracks credits (the "accountant")
- **flowise-proxy-service-py/** - Connects services together (the "messenger")
- **bridge/** - The website students/teachers use (the "front door")

---

## 🎓 Learning More

**Want to understand what you just did?**

- **Docker:** Think of it as running multiple mini-computers inside your computer
- **Services:** Each mini-computer runs one part of the platform
- **Ports (3002, 3082, etc.):** Like different doors to different rooms in a building
- **localhost:** A special address meaning "this computer"
- **API Key:** A secret password that lets services talk to each other

---

## ✅ Success Checklist

You're done when:

- ☐ Docker Desktop is running (whale icon in taskbar)
- ☐ All services show "Up" when you run `docker ps`
- ☐ You can open http://localhost:3002 (Flowise)
- ☐ You can open http://localhost:3082 (Bridge)
- ☐ You can log in with a student or teacher account
- ☐ You created users successfully

---

## 🎯 Quick Reference Commands

**Check if Docker is running:**

```
docker ps
```

**Start everything:**

```
cd %USERPROFILE%\Documents\ChatProxyPlatform
cd flowise && start-with-postgres.bat
cd ..\auth-service && start.bat
cd ..\accounting-service && start.bat
cd ..\flowise-proxy-service-py && start-docker.bat
cd ..\bridge && start.bat
```

**Stop everything:**

```
docker stop $(docker ps -q)
```

**View what's running:**

- Open Docker Desktop application
- Click "Containers" on the left

---

## 🚀 Next Steps After Local Setup

You're running on localhost - what now?

**If you just want to test/learn:**

- ☑ You're done! Keep using http://localhost:3082

**If you need school-wide access (multiple computers):**

- 📖 Read the **"🏫 School Network Deployment"** section above
- 📧 Send the email template to your IT department
- ⏳ Wait for IT to provide domain/IP and SSL certificate
- 🔧 Update configuration files as instructed
- 🎉 Share the new URL with teachers and students

**If you need help:**

- Check troubleshooting section
- Run `check_system.bat` for diagnostics
- Review logs using `logs.bat` in each service folder

---

**Questions or stuck? Check the troubleshooting section above!**

---

**Last Updated:** February 20, 2026