Below is a **consolidated, end-to-end guide** to installing ERPNext on **Windows** using **WSL** (Windows Subsystem for Linux) and **Ubuntu**, in a way that avoids all the common pitfalls (Python 3.12 issues, Yarn version mismatch, MariaDB “Access denied,” 404 at root, etc.). Follow these steps carefully, and you’ll have ERPNext up and running without errors.

**CHECK OUT THIS CHATGPT LINK IF THERES ANY CONFUSIONS:** [**https://chatgpt.com/share/6785f01b-3fd8-8000-b747-325ae5887cb3**](https://chatgpt.com/share/6785f01b-3fd8-8000-b747-325ae5887cb3)

## **1. Install & Set Up WSL and Ubuntu**

Open **PowerShell** (as Administrator) and **enable WSL**:  
powershell  
Copy code  
wsl --install

1. **Restart** your computer if prompted.
2. Install **Ubuntu** from the **Microsoft Store** (if not already installed).
3. **Launch Ubuntu** and create a username/password when prompted.

## **2. Update Ubuntu & Install Basic Packages**

Inside Ubuntu (WSL):

**Update** packages:  
bash  
Copy code  
sudo apt update && sudo apt upgrade -y

**Install common prerequisites**:  
bash  
Copy code  
sudo apt install -y software-properties-common git curl

## **3. Install Python 3.10 (or 3.11) via Deadsnakes PPA**

ERPNext v14 doesn’t fully support Python 3.12 yet because of certain dependencies (e.g., RestrictedPython). So we’ll use Python 3.10.

**Add** the Deadsnakes PPA (if on Ubuntu 20.04 or older):  
bash  
Copy code  
sudo add-apt-repository ppa:deadsnakes/ppa

sudo apt update

**Install** Python 3.10 & dev tools:  
bash  
Copy code  
sudo apt install -y python3.10 python3.10-dev python3.10-venv python3-pip

Verify:

bash

Copy code

python3.10 --version

You should see something like Python 3.10.x.

## **4. Install Node.js, npm, and Classic Yarn (1.x)**

Frappe requires Node.js for asset building, and the **classic Yarn (1.x)** for package management.

**Install** Node.js & npm (from Ubuntu repos or NodeSource):  
bash  
Copy code  
sudo apt install -y nodejs npm

**Uninstall** any existing Yarn 2 or 3 (if installed):  
bash  
Copy code  
sudo apt remove yarn

sudo apt autoremove

**Install** Yarn 1.x (e.g., 1.22.x) via npm:  
bash  
Copy code  
sudo npm install --global yarn@1.22.19

yarn --version # Should be 1.22.x

## **5. Install MariaDB, Redis, wkhtmltopdf**

ERPNext relies on MariaDB (database), Redis (caching & queueing), and wkhtmltopdf (for PDF generation).

bash

Copy code

sudo apt install -y mariadb-server mariadb-client redis-server wkhtmltopdf

## **6. Configure MariaDB for Password Authentication**

By default on some Ubuntu versions, MariaDB root uses “socket authentication.” ERPNext expects a password.

**Log in** to MariaDB:  
bash  
Copy code  
sudo mariadb

**Switch** to the mysql DB:  
sql  
Copy code  
USE mysql;

**Set** a root password and enable native password plugin:  
sql  
Copy code  
ALTER USER 'root'@'localhost'

IDENTIFIED VIA mysql\_native\_password

USING PASSWORD('YOUR\_ROOT\_PASSWORD');

FLUSH PRIVILEGES;

EXIT;

(Optional) **Restart** MariaDB:  
bash  
Copy code  
sudo service mariadb restart

## **7. Create a Python Virtual Environment for ERPNext**

We’ll create a dedicated virtual environment using Python 3.10:

**Make** a folder for your ERPNext setup (optional but organized):  
bash  
Copy code  
mkdir -p ~/erpnext-setup

cd ~/erpnext-setup

**Create** the venv:  
bash  
Copy code  
python3.10 -m venv frappe-bench-env

**Activate** it:  
bash  
Copy code  
source frappe-bench-env/bin/activate

1. You should see (frappe-bench-env) in your prompt.

## **8. Install the Bench CLI (Inside Virtual Env)**

bash

Copy code

pip install frappe-bench

No sudo, because everything goes into your venv.

## **9. Initialize a New Frappe Bench**

Still in ~/erpnext-setup, with venv activated:

bash

Copy code

bench init frappe-bench --frappe-branch version-14

cd frappe-bench

This creates a bench folder named frappe-bench and installs the Frappe framework.

## **10. Get ERPNext App**

bash

Copy code

bench get-app erpnext --branch version-14

## **11. Create a New Site**

**Create** a site (e.g., my-site):  
bash  
Copy code  
bench new-site my-site

* + When prompted for **MySQL root password**, enter the password you set in Step #6.
  + Set an **Administrator password** for ERPNext.

**Install** ERPNext on this site:  
bash  
Copy code  
bench --site my-site install-app erpnext

## **12. (Optional) Make my-site the Default Site**

If you want to avoid 404 errors at localhost:8000/, tell the bench to use my-site by default:

bash

Copy code

bench use my-site

## **13. Start the Development Server**

bash

Copy code

bench start

You’ll see the processes spin up (redis\_queue, redis\_cache, web, watch, socketio, etc.).

**Access** ERPNext at:

* http://localhost:8000  
  (if you set a default site)
* or http://127.0.0.1:8000?site=my-site  
  (to specify site by query param)

**Login** with:

* **Username**: Administrator
* **Password**: (whatever you set)

# **That’s It!**

You now have a working **ERPNext v14** setup on **Windows (WSL Ubuntu)**.

### **Quick Checklist**

1. **WSL** + **Ubuntu** installed.
2. **Python 3.10** via Deadsnakes PPA.
3. **MariaDB** configured with native password.
4. **Node.js** + **npm** + **Yarn 1.x** installed.
5. Virtual environment + bench installed.
6. **Initialized bench** -> **Installed ERPNext** -> **Created site**.
7. **Bench start** -> Access from browser.

No more errors about Python 3.12, Yarn 2/3, or MariaDB root password. You’re all set to explore, develop, or evaluate ERPNext on Windows!