Summary

This guide provides **exact, project-specific steps** to set up your MT copy-trading app from scratch on Windows, covering all required **downloads**, **installations**, and **commands**. You'll install and configure **WSL 2**, **Docker Desktop**, **Node.js**, **npm**, **Git**, **Visual Studio Code**, and **Postman**, then clone your `MT` repo, configure environment files, and launch your containers with Docker Compose.

---

1. Prerequisites & Downloads

1.1 Enable Windows Subsystem for Linux (WSL 2)

1. **Open PowerShell as Administrator** (search "PowerShell" → right-click → Run as administrator).
2. **Install WSL and set version 2**:

```
wsl --install                          # Installs WSL2 and Ubuntu by default
([Issues with Docker Desktop and WSL 2 Integration]
(https://forums.docker.com/t/issues-with-docker-desktop-and-wsl-2-
integration/141865?utm_source=chatgpt.com))
wsl --set-default-version 2           # Ensures all new distros use WSL2
([Issues with Docker Desktop and WSL 2 Integration]
(https://forums.docker.com/t/issues-with-docker-desktop-and-wsl-2-
integration/141865?utm_source=chatgpt.com))
```

3. **Restart Windows** to apply changes.
4. **Verify** by running:

```
wsl -l -v                              # Lists distributions and WSL versions
([iot-plug-and-play-bridge/pnpbridge/docs/build_deploy.md at master]
(https://docs.microsoft.com/es-es/azure/iot-pnp/howto-build-deploy-extend-pnp-
bridge?utm_source=chatgpt.com))
```

1.2 Install Docker Desktop with WSL 2 Backend

1. **Download Docker Desktop for Windows (AMD64)** from: https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe?utm_source=chatgpt (Install Docker Desktop on Windows)
2. **Run the installer**, ensuring "Use WSL 2 based engine" is selected when prompted (Docker Desktop WSL 2 backend on Windows).
3. **Launch Docker Desktop** from the Start menu and wait for the icon to show "Docker is running."

1.3 Install Node.js & npm (LTS)

1. **Download Node.js LTS installer** for Windows (x64) from: https://nodejs.org/en/ (Node.js — Run JavaScript Everywhere)
2. **Run the installer**, accept defaults, and ensure both **Node.js** and **npm** are installed.
3. **Verify** versions:

```
node -v                                # e.g., v22.x.x  ([Download Node.js]
(https://nodejs.org/en/download?utm_source=chatgpt.com))
npm -v                                 # e.g., 10.x.x
```

> *Alternative via winget:*
>
> ```
> winget install OpenJS.NodeJS.LTS     # Installs Node.js LTS  ([Installing Node.js via package manager](https://nodejs.org/en/download/package-manager/all?utm_source=chatgpt.com))
> ```

1.4 Install Git for Windows

1. **Download Git for Windows (x64)** from: https://git-scm.com/download/win (Downloading Package - Git)
2. **Run the installer**, choose "Use Git from the Windows Command Prompt," and accept defaults.
3. **Verify**:

```
git --version                        # e.g., git version 2.49.0  ([Downloads - Git](https://git-scm.com/downloads?utm_source=chatgpt.com))
```

1.5 Install Visual Studio Code

1. **Download VS Code User Installer (x64)** for Windows from: https://code.visualstudio.com/download (Download Visual Studio Code - Mac, Linux, Windows)
2. **Run the installer**, accept defaults, and add **"Open with Code"** options.
3. **Launch** VS Code and install recommended extensions (e.g., ESLint, Prettier).

1.6 Install Postman

1. **Download Postman for Windows** from: https://www.postman.com/downloads/ (Download Postman | Get Started for Free)
2. **Run** the `.exe` installer and follow prompts.
3. **Launch** Postman and optionally sign up/sign in.

---

2. Project Setup Commands

2.1 Clone Your MT Repository

In PowerShell (not WSL) at your Documents folder:

```
cd "C:\Users\dell mve\Documents"        # navigate to Documents
git clone https://github.com/talha0pse/MT.git  # clone MT repo  ([Git](https://git-scm.com/?utm_source=chatgpt.com))
cd MT                                  # enter project folder
```

2.2 Configure Environment Variables

Backend

1. In `MT/backend/`, create a file named `.env` with:

```
MONGO_URI="mongodb://localhost:27017/mt_db"
JWT_SECRET="mySuperSecretKey"
PORT=5000
```

2. **Save** the file.

Frontend

1. In `MT/frontend/`, create `.env.local` with:

   ```
   REACT_APP_API_URL="http://localhost:5000"
   ```

2. **Save** the file.

2.3 Build & Run with Docker Compose

Back in `MT/` root:

```
docker compose up --build              # builds images & starts containers  ([Docker Desktop WSL 2 backend on Windows](https://docs.docker.com/desktop/features/wsl/?utm_source=chatgpt.com))
```

- **Backend** at http://localhost:5000
- **Frontend** at http://localhost:3000

To stop and clean up:

```
docker compose down                    # stops and removes containers & network ([Docker Desktop WSL 2 backend on Windows](https://docs.docker.com/desktop/features/wsl/?utm_source=chatgpt.com))
```

---

3. Memory & Interview Tips

3.1 Mnemonics & Chunking

- **F-W-C-R-C** for Dockerfiles: **FROM**, **WORKDIR**, **COPY**, **RUN**, **CMD** .
- **"Clone-Configure-Compose"** flow for project setup.

3.2 Active Recall & Rehearsal

- **Practice** writing each Dockerfile and Compose file on paper.
- **Quiz** yourself on installation steps using flashcards.

3.3 Interview-Ready Explanation

> ***"To containerize MT, I first enabled WSL 2 via `wsl --install` and installed Docker Desktop with WSL 2 integration. Then I installed Node.js LTS, Git, VS Code, and Postman. After cloning the `MT` repo, I created `.env` files in backend and frontend, installed dependencies via Docker multi-stage builds, and orchestrated services in `docker-compose.yml`. Finally, I launched everything with `docker compose up --build`, yielding a consistent, portable development environment."***

This concise answer demonstrates both your **practical execution** and **conceptual understanding** of environment setup and containerization.