

Shoonya Project – Environment Setup & Reproducibility Guide

This document is a **single source of truth** for recreating the Shoonya project environment on **any machine** (Windows / EC2 / Linux / new PC / remote server) and resuming work **exactly from the same stable state**.

Follow this step-by-step. Do not skip steps.

1. Design Principles (Why this works)

- Python version is pinned (3.9.x)
- Dependencies are pinned (`requirements.txt`)
- Broker SDK is bundled as a wheel (no external dependency risk)
- Virtual environment is isolated (`venv/`)
- Setup is idempotent (safe to re-run)
- Windows, Linux, EC2 all behave the same logically

Time does NOT break Python environments. Changes do.

2. Required Python Version

Supported versions

- Python 3.9.x (recommended, production-aligned)
- Python 3.10.x (allowed but not used currently)
- Python 3.8 or lower (blocked)
- Python 3.11+ (not supported for this project)

Why Python 3.9?

- Matches EC2 production
 - Stable ecosystem
 - All required wheels available
 - Predictable runtime behavior
-

3. Project Directory Structure (Final)

```
shoonya/
├── setup.py
└── pyproject.toml
```

```
├── README_SETUP_ENVIRONMENT.md  
├── requirements/  
│   ├── requirements.txt  
│   └── NorenRestApi-0.0.30-py2.py3-none-any.whl  
└── shoonya_bot/  
    └── venv/                      # created by setup.py
```

 Never rename `venv/` or create `.venv/` in this project.

4. requirements.txt (Frozen & Approved)

This file defines **intentional direct dependencies only**.

```
# Core utilities  
cachetools==6.2.4  
cachelib==0.13.0  
certifi==2025.11.12  
charset-normalizer==3.4.4  
colorama==0.4.6  
idna==3.11  
six==1.17.0  
tabulate==0.9.0  
tzdata==2025.3  
pytz==2025.2  
PyYAML==6.0.3  
  
# Networking / HTTP  
requests==2.32.5  
urllib3==2.6.2  
websocket-client==1.9.0  
websockets==15.0.1  
  
# Scheduling / OTP  
schedule==1.2.0  
pyotp==2.9.0  
python-dateutil==2.9.0.post0  
python-dotenv==1.2.1  
  
# Data / Math  
numpy==2.0.2  
pandas==2.3.3  
scipy==1.13.1  
  
# Web / API stack
```

```
Flask==3.0.0
Flask-Session==0.8.0
Werkzeug==3.1.4
Jinja2==3.1.6
MarkupSafe==3.0.3
click==8.1.8
gunicorn==23.0.0
waitress==2.1.2

# Shoonya / Noren API broker SDK (pure-python wheel)
./requirements/NorenRestApi-0.0.30-py2.py3-none-any.whl
```

5. setup.py – What it does

`setup.py` is the **only supported way** to initialize the project.

It performs:

1. Validates Python version (3.9 only)
2. Blocks running inside an active venv
3. Creates `venv/` if missing
4. Upgrades pip safely (Windows-compatible)
5. Installs all dependencies from `requirements.txt`
6. Reuses venv on re-run (does NOT recreate)

 Do not manually create a virtual environment.

6. Windows Setup (Fresh Machine)

Step 1: Install Python 3.9

- Download: **Python 3.9.13 (64-bit)** Windows installer
- During install:
 - Check **Add Python to PATH**
 - Install for current user

Verify:

```
py -3.9 --version
```

Step 2: Run setup.py (ONE command)

From PowerShell:

```
cd C:\path\to\shoonya  
py -3.9 setup.py
```

Expected output: - Python version OK - venv created or reused - All packages installed - Setup complete

Step 3: Activate venv

```
venv\Scripts\Activate.ps1
```

Verify:

```
python --version    # must show 3.9.x  
pip list
```

Step 4: VS Code Configuration (IMPORTANT)

1. Open project folder in VS Code
2. **Ctrl + Shift + P**
3. **Python: Select Interpreter**
4. Choose:

```
.\venv\Scripts\python.exe
```

After this: - ► Run button uses correct Python - Terminal auto-uses venv

7. EC2 / Linux Setup

Step 1: Install Python 3.9

On Amazon Linux:

```
sudo yum install python39
```

Verify:

```
python3.9 --version
```

Step 2: Run setup

```
cd /home/ec2-user/shoonya  
python3.9 setup.py
```

Step 3: Activate venv

```
source venv/bin/activate
```

Verify:

```
python --version  
pip list
```

8. Verification Checklist (MANDATORY)

After setup, the following **must be true**:

```
Python 3.9.13  
numpy 2.0.2  
pandas 2.3.3  
scipy 1.13.1  
Flask 3.0.0  
NorenRestApi 0.0.30
```

If these match → environment is correct.

9. What NOT to do (Critical Rules)

- ✗ Do NOT install Python 3.11+
✗ Do NOT edit venv manually

- ✗ Do NOT add transitive deps to requirements.txt
 - ✗ Do NOT run setup.py inside an active venv
 - ✗ Do NOT mix `venv` and `.venv`
 - ✗ Do NOT rely on system Python
-

10. Re-running setup.py

- Safe to re-run
- Will NOT recreate venv
- Will reuse existing environment

Optional hard-lock can be added with `.setup_done` marker.

11. Disaster Recovery (New PC / New EC2)

1. Install Python 3.9
2. Clone / copy project
3. Run:

```
py -3.9 setup.py
```

4. Activate venv
5. Resume work

❤️ Total recovery time: ~5–10 minutes

12. Final Guarantee

If you follow **this document exactly**, you will:

- Resume at the same environment state
- Avoid dependency drift
- Match EC2 production behavior
- Be immune to PyPI changes
- Be immune to OS changes

This document is **intentionally verbose** so future-you never gets stuck.

✓ Environment is now officially frozen and reproducible.