**Nasdaq100 CFD Trading Signal Bot** project, grouped by **technology/tool**, including all Git/GitHub commands and their purpose.

**🐍 Python**

Used as the core programming language for the bot.

* Built a script to detect **Fair Value Gaps (FVGs)** using candlestick logic.
* Used Python to process 15-minute candle data.
* Triggered Telegram alerts when signals were found.
* Logged signal data to a **CSV** file (fvg\_signals.csv).
* Modular design for future upgrades (Order Blocks, Liquidity Zones, RR calc, etc.).

**📦 Python Libraries Used**

**tvdatafeed**

* Fetched **live 15-minute candles** for **CDT (Nasdaq100 CFD)** from **TradingView**.
* Used TradingView login credentials for access.

**pandas**

* Processed and filtered historical price data.
* Managed CSV reading/writing for signal history.

**requests**

* Sent real-time **Telegram messages** using the Telegram Bot API.

**python-telegram-bot *(from previous RSI bot, not used in this FVG bot)***

* Was part of your earlier RSI Telegram bot hosted on Render.

**💬 Telegram Bot API**

* Created a bot using **@BotFather**.
* Retrieved **bot token** and **chat ID**.
* Sent FVG signals like:

yaml

CopyEdit

📉 FVG Detected!

Time: 2025-06-30 13:45:00

Type: Bullish FVG

**📂 File Structure**

plaintext

CopyEdit

nasdaq\_fvg\_bot.py # Main bot logic

fvg\_signals.csv # Stores FVG signal history

README.md # Project documentation

requirements.txt # Python dependencies

**💾 CSV Logging**

* Detected signals are saved with timestamp and type.
* Avoids duplicates by checking for existing timestamps.
* Useful for reviewing signal history and future backtesting.

**💻 Visual Studio Code (VS Code)**

* Used as your **IDE (Integrated Development Environment)** to:
  + Write and edit code
  + Run scripts using Terminal (Ctrl + ~)
  + Manage Git commands via integrated terminal

**🐙 Git & GitHub**

Used for version control and publishing the project publicly.

**✅ Purpose:**

* Track code changes
* Roll back mistakes
* Publish work to GitHub
* Make the project portfolio-ready

**🧾 Git Commands Used and Their Purpose:**

| **Command** | **Purpose** |
| --- | --- |
| git init | Initializes a new Git repository in the folder. |
| git add . | Stages all changes (new files, edits, deletions) for commit. |
| git add filename | Stages a specific file (e.g., requirements.txt) for commit. |
| git commit -m "message" | Saves a snapshot of the current changes with a message. |
| git remote add origin <repo-url> | Links your local Git repo to a GitHub repo. |
| git branch -M main | Renames the default branch to main. |
| git push -u origin main | Pushes the first commit and sets the upstream branch. |
| git push | Pushes changes to GitHub (after first setup). |
| git pull origin main --rebase | Pulls changes from GitHub and reapplies your changes on top (used when your local repo is behind the remote). |

**📄 README.md (Documentation)**

* Explained the purpose and features of the bot.
* Listed technologies used and how to run the project.
* Displayed on GitHub repo homepage.
* Updated either:
  + Locally via VS Code → git commit & push
  + Or via GitHub Web UI → Pencil icon → Edit → Commit changes

**🌐 Render Platform (from your earlier bot)**

While not directly used in this project, your **RSI-based bot** was deployed there and used:

* requirements.txt (also extended here)
* Python background worker setup
* Telegram alerts based on RSI logic