About

Dive into Drapt's history, from the beginning to where it currently stands.
Quickly navigate to the section which interests you:

What is Drapt?

Drapt

Drapt is a modular portfolio analytics platform for portfolio risk and performance analysis - built for clarity, speed, and most importantly to educate.

Empowering Portfolio Managers and Analysts alike, it demystifies quantitative portfolio analysis in an intuitive way, making it accessible for all levels of technical proficiency, from students who have never heard of Python to seasoned quants.

Powered by high-performance servers and optimised computation pipelines, Drapt makes it easier than ever to understand a portfolio's current position amidst the inherent chaos of financial markets.

```
drapt init
> initialising...
> Done! Welcome to Drapt.
```

Early vision...

November, 2024

Drapt began its story as DRAP-T, an acronym for *Dynamic Risk Analysis and Performance Tool*, in November 2024.

The inspiration for starting the project was my desire to incorporate quantitative risk analytics into the NEFS Investment Fund, as around 1 month into being an analyst I realised that The Fund relied on qualitative risk analysis.

Until the revamp, Drapt was being developed in a very inefficient manner, with the following tech stack:

- Python (backend)
  - Flask - used for backend routing, deployed using a test server. Unconventional.
  - NumPy, Pandas, yfinance - quite typical for processing data.
  - Sqlite3 - hosted locally, not very modular as I had to create custom database managers in Python.
- HTML/CSS/JS (frontend)
  - HTML and CSS - manually written, literally from the ground up. No Tailwind CSS or DaisyUI like is being used today.
  - JS - base JS, including only one package: Chart.js

Milestone #1

End of November, 2024

The first version of DRAP-T was not the prettiest, or the most functional...

Figure 1: First functional DRAP-T prototype (Nov 2024)

The first prototype did not allow for portfolio creation, and it only had three features:

- A returns histogram with no additive value.
- A cumulative returns chart, though I mistakenly used a simple sum of returns instead of compounding them - I hadn't discovered log returns yet.
- A low-utility Monte Carlo simulation visualisation which 'forecasted' future portfolio returns, without taking into account asset correlations.

```
def milestone_1():
  fix_cumulative_returns = False
  return "The prototype which started it all.."
```

Milestone #2

February, 2025

This was the final version before the current revamp. It helped take my team in the WBSS Investment Challenge 2024/2025 to the finals, it was really quite fantastic.

Fixes were made, styles were changed, and the platform became a whole lot more functional. I mean, *I would hope so,* it took 3 months.

Additionally, DRAP-T was renamed to **Drapt Analytics**, to reflect the clean interface and professional usability of the platform.

From a high level, improvements and feature changes included:

- Changed the style, from that odd green and gray to a more professional blue on white.
- Fixed cumulative returns (using log returns now instead of percentage returns).
- Added rolling volatility analysis (Figure 2), this really was a great feature.
- Improved the asset correlation matrix (Figure 3).

And a few more features.

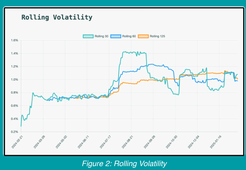

Figure 2: Rolling Volatility

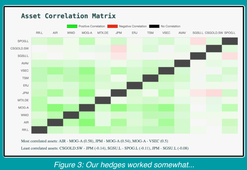

Figure 3: Our hedges worked somewhat...

To expand on my previous sentence mentioning the WBSS Investment Challenge, my team **Notts for Profit** managed to finish top 5 out of 80+ teams across the United Kingdom. We met some heavy competition, from brilliant teams from Warwick, UCL and Imperial, however as a team full of first-year undergraduates we really put Nottingham on the map.

Team Notts for Profit, representing NEFSIF on the 17th floor of The Shard (Feb, 2025)

```
def milestone_two():
  return "Refactor, restyle, and rethink risk"
```

Where it currently stands

Present

At the end of May 2025, I decided to revamp Drapt Analytics, rebuilding it using a modern React frontend (using TailwindCSS and DaisyUI), as well as upgrading the backend.

As well as this, there was a (yet another) name change, with the platform now called **Drapt**.

Drapt is still in motion, slowly accelerating, but it's already become something far greater than I imagined when I started. Thanks for reading - and welcome aboard.

You may ask, what's next?

- Month 1 → UI Completion + UX Polish
  - Refine visuals, motion and responsiveness.
  - Finalise component library (e.g., metrics, cards, layouts).
  - Add joy: empty states, tooltips, microcopy.
  - Wrap up with confidence.
- Month 2 → Backend Build + API Design
  - FastAPI backend: user, auth, risk/return APIs.
  - Define clear schema: portfolios, metrics, time series.
  - Connect frontend to live/mock endpoints.
  - Handle auth logic and state management.
- Month 3 → Infrastructure, Optimisation, Internal Deployment
  - Dockerise frontend + backend.
  - Deploy staging instance (Render/Fly.io).
  - Add basic analytics/logging.
  - Run Drapt internally at NEFS with dummy logins.
- Month 4 → Documentation, Rollout, Expansion
  - Create onboarding flows + logic.
  - Finalise README, tech spaces, admin tools.
  - Prep outreach: PDF one-pager, demo videos, live microsite.
  - Integrate feedback, and do a "V1.0" cut.

Szymon Kopyciński

Founder and Lead Developer.

Acknowledgments

Drapt wouldn't be possible without the incredible open-source tools and resources that supported its development:

- React - the foundation of the frontend.
- Tailwind CSS & DaisyUI - for styling and UI speed.
- Python, Pandas, NumPy, FastAPI - for data wrangling, logic, and the rest of the backend.
- My mentors at NEFS, The Fund, and the WBSS challenge team - for believing in this vision early on.
- And finally, my parents, for their unconditional support throughout my studies.