

KrakenR

**AI-Driven Financial Dashboard with GPT-3.5-turbo and
Real-Time Analysis**

Programming tools in Data Science

19.12.2024

Nathanaël , Luca, Steve, Yanis, Fernanda, Hassan



Content

01

Motivations & Objetives

02

App presentation

03

Functions

04

Pain Point

05

Further development

06

Live Demo



Motivations



01



We enjoy finance and created a Shiny app to track crypto prices and news, blending our interests with technology

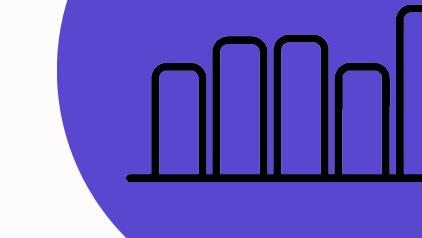
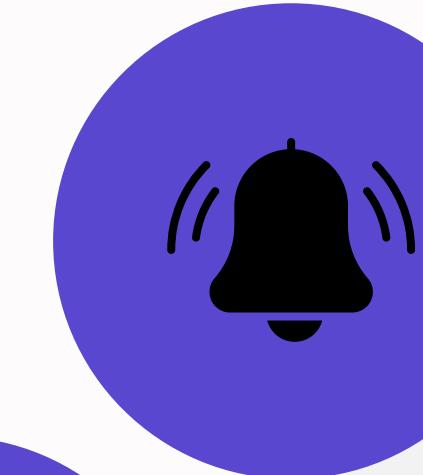
Our interest in Kraken's ecosystem inspired us to explore their API and create an app that connects users to real-time market data and updates



02



Main features

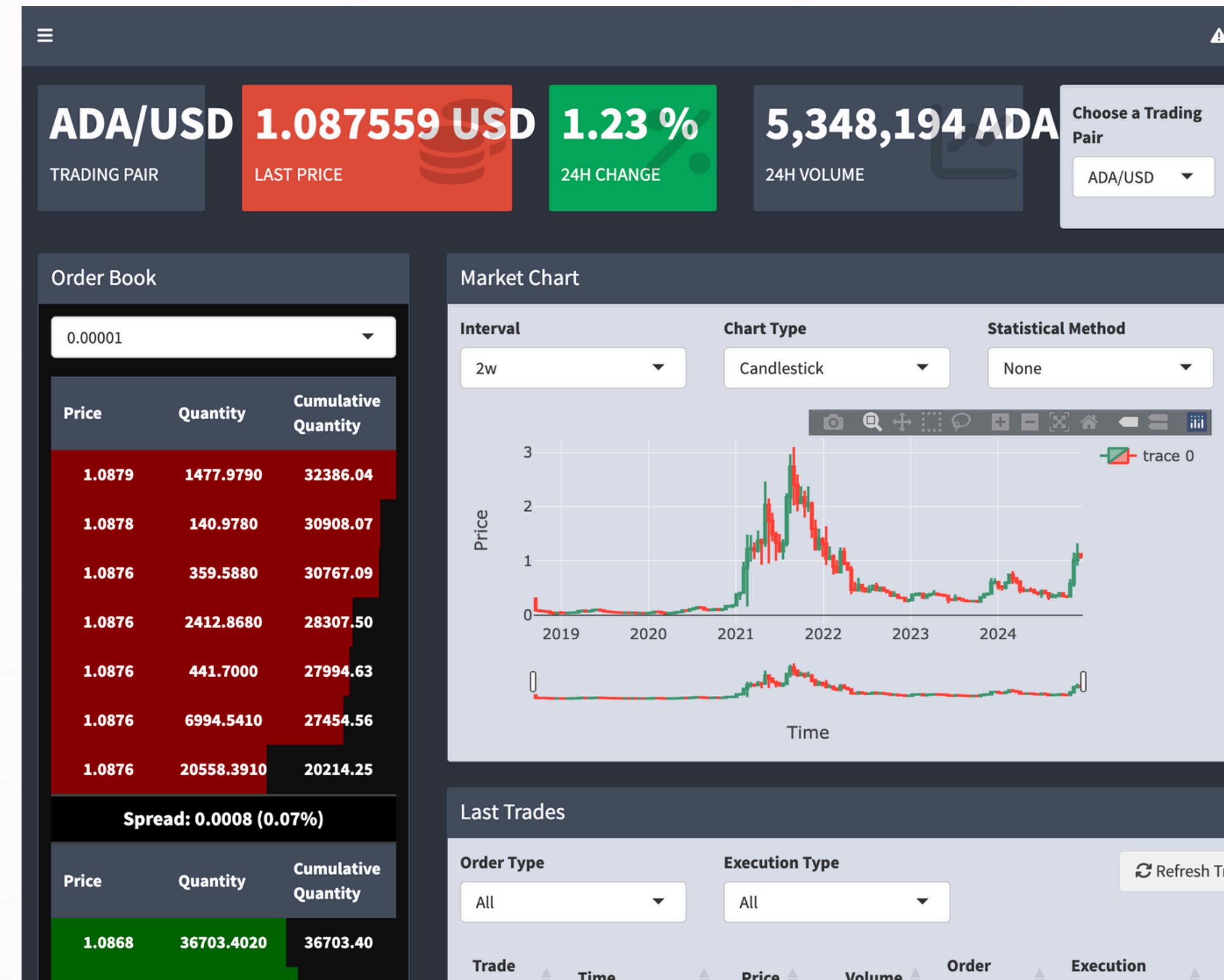
-  **Near real-time data**
Retrieve live market data and visualize it with graphs
-  **News feed**
Stay updated with the latest market news and trends
-  **Dashboard**
Apply statistical methods directly to your graphs
-  **User Alert**
Set Alert to receive real time notification on the price



Dashboard

01

Market Chart with statistical methods overview



02

Bid/Ask Table with real time dynamic bar plots

03

Last place header and daily change volume

04

Last trade table

News

01

Dynamic news card pagination

02

Possibility to read articles directly within the app

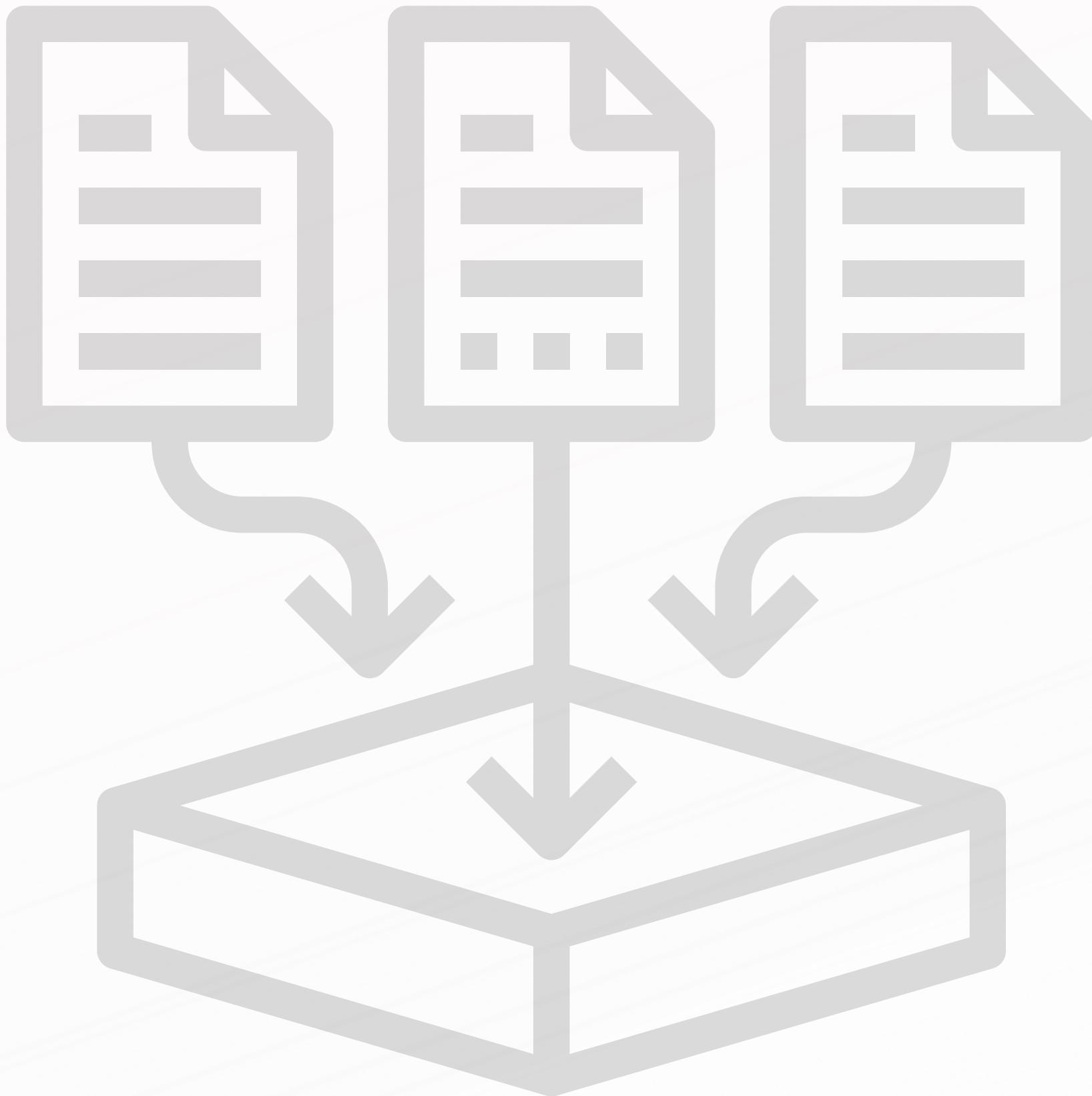
Near real time data

01

Fully dynamic app, no hard-coded assets or data.

02

Automatically updates when Kraken adds or removes assets, always aligned with Kraken's latest offerings.



Functions : API information

	<i>Description</i>
<i>getStatus</i>	The <i>getStatus</i> function retrieves the current system status of the Kraken exchange. It returns either the operational status, the timestamp, or both
<i>getTime</i>	The <i>getTime</i> function retrieves the current server time from the Kraken exchange. The function returns a POSIXct object or a formatted character string

Functions : Assets

Description

getAssets

The *getAssets* function fetches detailed asset information from the Kraken API, either for all assets or specified ones by their abbreviations

getPairs

The *getPairs* function retrieves information about tradable asset pairs from the Kraken API. The function returns a structured data frame

getTickers

The *getTickers* function retrieves detailed ticker information for specific trading pairs or all available pairs from the Kraken API. The function returns a structured data frame

Functions : Market Data

Description

getOB

The *getOB* function retrieves recent order book data for a specific trading pair from the Kraken API. The function returns a structured data frame containing bid and ask orders, sorted by price in ascending order

getOHLC

The *getOHLC* function retrieves recent OHLC (Open, High, Low, Close) data for a specific trading pair from the Kraken API. The function returns a structured data frame

getSpreads

The *getSpreads* function gets recent spread data for a specific trading pair from the Kraken API. The function returns a structured data frame

getTrades

The *getTrades* function retrieves recent trade data for a specific trading pair from the Kraken API. The function returns a structured data frame

Pain points

- 01- Collaboration
- 02- API calls
- 03- Interface Design
- 04- Feature complexity
- 05- CRAN and Shinyapps integration



Further developments

01

Expand the scope of the project to include other markets.

02

Enhanced the news feed to include a function permitting to more accurately filter information.

03

Performance improvement : The more features are added on the app the slower it runs.



Demo

