**Projektzusammenfassung: Modulares Crypto-Analyse-, Backtesting- und Reporting-Framework**

**1. Projektziel**

Entwicklung eines robusten, skalierbaren Python-Frameworks für die wissenschaftliche Analyse, Simulation und Visualisierung von Kryptowährungs-Handelsstrategien. Fokus auf Wiederholbarkeit, Automatisierung, Skalierbarkeit und verständliches Reporting.

**2. Kernfeatures und Module**

**2.1 Datenimport & Aufbereitung**

* Import von Marktdaten (z.B. BTCUSDT) im Parquet-Format, beliebige Zeitintervalle (Stunden, später Minuten)
* Automatische Berechnung technischer Indikatoren: EMA (20, 50, 200), RSI, MACD, Bollinger Bands, ADX, ATR, etc.
* Klassifizierung von Marktregimen (z.B. Uptrend, Downtrend, Seitwärts, High Volatility, Breakout, Overbought/Oversold)

**2.2 Candlestick Pattern Erkennung**

* Implementierung von 10 Candlestick-Formationen, z.B.:
  + Bullish/Bearish Engulfing
  + Morning Star, Evening Star
  + Hammer, Shooting Star
  + Inside Bar, Doji, Piercing Line, Three White Soldiers

**2.3 Backtesting & Strategie-Simulation**

* Walkforward-Backtesting mit Rolling-Fenstern: 6 Monate In-Sample, 1 Monat Out-of-Sample
* Parallele Ausführung aller Fenster via concurrent.futures.ProcessPoolExecutor
* Für jedes Fenster: Auswahl der besten Strategie basierend auf In-Sample Performance
* Dynamisches Kapitalmanagement:
  + Startkapital 10.000 €
  + Maximaler Einsatz pro Trade 20 % des verfügbaren Kapitals
  + Stop-Loss bei 5 % Verlust, Take-Profit bei 20 % Gewinn
  + Kapital wird nach jedem Trade aktualisiert
  + Automatischer Stop bei Kapitalverlust auf 0

**2.4 Trade-Management**

* Pro Trade Speicherung von:
  + Trade-ID, Asset, Parquet-Quelle, Strategie, Marktregime
  + Entry/Exit Zeit und Preis
  + Kapital-Einsatz
  + Absoluter und prozentualer Gewinn/Verlust
  + Kapital nach dem Trade

**2.5 Visualisierung**

* Interaktive Plotly-Charts mit Subplots:
  + Oberes Diagramm: Candlestick Chart mit Entry- und Exit-Dreiecken
    - Gefüllte Dreiecke für Entry, ungefüllte für Exit, farblich grün/rot nach Profit/Loss
    - Trade-ID als Text mit schwarzem Kreis umrandet
    - Verbundene Entry-Exit Punkte mit gestrichelten Linien
    - Hover-Infos erweitern Anzeige um Kapital-Einsatz und Gewinn/Verlust
  + Unteres Diagramm: Kapitalverlauf über Zeit mit Trade-IDs und Gewinn-Hover
* Export der Charts als HTML und optional PNG

**2.6 Reporting**

* Vollständige Excel-Reports:
  + Zusammenfassung der Performance pro Walkforward-Fenster
  + Detailierte Trade-Listen aller OOS Trades
* Terminal-Ausgabe der wichtigsten KPIs und Performance-Übersichten

**3. Erweiterungsmöglichkeiten & Workflow**

* Flexible Integration neuer Candlestick Patterns und Indikatoren
* Anpassbare Kapital- und Risikomanagement-Parameter
* Erweiterung auf andere Assets und Zeitauflösungen (z.B. 1-Minuten-Daten)
* Portfolio-Backtests mit Multi-Asset-Handling
* Automatisierte Optimierung und Machine Learning zur Strategieverbesserung
* Dashboard-Integration (Dash, Streamlit) und PDF-Reporting

**4. Technische Eckdaten**

* Python 3.11, virtuelle Umgebung (.venv)
* Pandas, NumPy, Plotly, TQDM, Dateutil, Openpyxl, concurrent.futures
* Parallelisierte Berechnung für effiziente Backtests
* Modularer Codeaufbau mit klarer Trennung von Daten, Analyse, Visualisierung, Reporting

**5. Bisheriger Stand & Nächste Schritte**

* Vollständiges Walkforward-Backtesting mit Kapitalmanagement und interaktiver Visualisierung realisiert
* Robuste Fehlerbehandlung, skalierbare Darstellung, sauberes Reporting implementiert
* Nächste Themen:
  + Umsetzung mit feinerer Zeitauflösung (Minutendaten)
  + Erweiterte Strategieparameter und Position-Sizing
  + Integration Portfolio-Backtest & Risikoanalyse
  + Automatisierte Optimierung (z.B. Grid Search, Optuna)
  + Dashboard mit interaktiven Filtermöglichkeiten